
Faculty of Business and International Finance

# Bachelor's Thesis

in the Degree Program Internationales Finanzmanagement

leading to the Academic Degree of
Bachelor of Science (B.Sc.)

**Natural Language based Financial Forecasting**

presented by:

**Dennis Thumm**

Starting Date: 15.05.2019
Closing Date: 15.10.2019

First Examiner: Prof. Dr. Mathias Engel
Second Examiner: M.A. Andreas Schneider

# Acknowledgement

The following individuals, in no particular order, have been tremendously supportive throughout the duration of this thesis. The author would like to take this opportunity to express his gratitude.

Manuel Kleinknecht and Michael Bloss of Nuertingen-Geislingen University for providing a first insight into financial engineering in the course of their lectures and Michael Bloss in particular for establishing contact to Andreas Schneider.

Professor Mathias Engel of Nuertingen-Geislingen University for the introduction of Machine Learning and Data Mining within his lecture. Furthermore, for the openness, continuos support and feedback that enabled this project.

Andreas Schneider of IBM for his voluntary commitment and constant encouragement. The help through his real-world experience especially fascilitated the modelling.

# Table of Contents

# List of Abbreviations

| | | |
|---|---|---|
| ML | = | Machine Learning |
| NLP | = | Natural Language Processing |
| NLFF | = | Natural Language based Financial Forecasting |
| SPX | = | Standard & Poor's 500 Stock Index |
| CBOE | = | Chicago Board Options Exchange |
| VIX | = | CBOE Volatility Index |
| AAII | = | American Association of Individual Investors |
| SEC | = | United States Securities and Exchange Commission |
| EDGAR | = | Electronic Data Gathering, Analysis, and Retrieval |
| LSTM | = | Long Short-Term Memory |
| RNN | = | Recurrent Neural Network |
| ASCII | = | American Standard Code for Information Interchange |
| HTML | = | Hypertext Markup Language |
| XBRL | = | eXtensible Business Reporting Language |
| XML | = | Extensible Markup Language |
| TM | = | Text Mining |
| TF-IDF | = | Term Frequency-Inverse Document Frequency |
| SVM | = | Support Vector Machine |
| ANN | = | Artificial Neural Network |
| k-NN | = | k-Nearest Neighbour |
| MAE | = | Mean Absolute Error |
| AdaGrad | = | Adaptive Gradient Algorithm |
| RMSProp | = | Root Mean Square Propagation |
| MSE | = | Mean Squared Error |
| MAPE | = | Mean Absolute Percentage Error |
| RMSE | = | Root Mean Squared Error |
| CV | = | Cross-Validation |
| CPCV | = | Combinatorial Purged Cross-Validation |
| AGPT | = | Average Percentage Gain Per Transaction |

# Table of Figures

# Table of Tables

# 1 Introduction

## 1.1 Overview on the Issue

In the course of time, machine learning (ML) methods have been applied to the field of finance.

The versatile application possibilities of this technology include the evaluation of enterprises, identification of similar firms, prediction of financial figures, optimization of portfolios and automated trading.

In the past, financial time series forecasting relied on classical standard econometric models like multivariate linear regression that do not learn. ML not only offers non-linear functions that learn patterns in a high-dimensional space but also allows to gain additional knowledge (Prado 2018, p. 15).

This thesis faces the concern of which ML techniques suit for financial forecasting. More precisely the research issues how price and volatility development prediction can be modelled with information gained from natural language processing (NLP).

## 1.2 Objective and Methodical Procedure

The aim of this thesis is to proof the suitability of natural language based forecasting for price and volatility development by an experiment. In other words, the hypothesis of this work is that the development of financial figures as price and volatility can be predicted with ML modelling. Thereby, open source libraries for ML will be accessed in a Python development environment. The input data is sourced from data science platforms and providers.

## 1.3 Structuring

After this short introduction into the topic this paper will give an overview on the theoretical background of ML techniques and financial forecasting. Afterwards, the methodology for the modelling will be described. The core of the thesis is the development of a model which is divided into the following steps:

1. Data Scraping
2. Data Preparation
3. Data Exploration
4. Model Development
5. Model Implementation

6. Model Management

The first three chapters are typically the most arduous since a lot of the models success depends on the quality and quantity of data. The additional strains in this particular case derive from the noisy unstructured nature of financial textual data. The exploration should also give an insight into the characteristics of the data. In order to limit the efforts, one might refer to pre-processed data streams. Part of the development will be the choice of ML techniques. At the implementation, there will be a general trade-off between network complexity, accuracy and temporal restrictions. The last step of management primarily focusses on the improvement of performance. This is going to be followed by an evaluation and argumentation of the results. Finally, this paper concludes with recommendation for further studies and tries to give an outlook on upcoming development.

## 1.4 Current State of Research

Although the roots of ML go back decades the recent rise in particular of deep learning is due to the increase in computational power and data (Marcus 2018, p. 2). The research field of natural language based financial forecasting (NLFF) established with the soaring amount of papers published. That is due to the exponentially increasing generated amount of user content by social media websites (Xing et al. 2018, p. 50).

Text mining for market prediction is positioned at the intersection of linguistics, ML and behavioural economics (Nassirtoussi et al. 2014, p. 7654). The concept of readability as an indicator for interpretability has been introduced (Loughran and McDonald 2016, p. 1196). Identifying suitable feature selection method remains an open problem, ontologies need to be employed for financial domain and the use of sophisticated technologies is suggested (Kumar and Ravi 2016, p. 144). Recently, long-term stock index forecasting based on text mining of regulatory disclosures outperformed baseline predictions (Feuerriegel and Gordon 2018, p. 27).

## 1.5 Central Question and Working Hypothesis

A first issue of the experiment setting is which financial asset or figure should be chosen to predict. The model will try to forecast the price development of Standard & Poor's 500 stock index (SPX) and the corresponding Chicago Board Options Exchange (CBOE) Volatility Index (VIX). The data used for the forecast will be gathered from the American Association of Individual Investors (AAII) and files from the United States Securities and Exchange Commission (SEC). The range of prediction will refer to the frequency of data which is weekly for AAII sentiment and annually for Form 10-K or rather quarterly for

Form 10-Q filings from the SEC's Electronic Data Gathering, Analysis, and Retrieval (EDGAR) database. For the prediction model a long short-term memory (LSTM) recurrent neural network (RNN) is going to be used. The reasons for the above chosen experiment setting will be explained in depth in the background and methodology section.
The working hypothesis proofs by an experiment whether it is possible to forecast the weekly and quarterly price movement of the SPX and VIX rate with AAII sentiment data and textual information gained from SEC Form 10-K and 10-Q filings. Hence, the following research issue derives:

> Is it possible to partially predict weekly and quarterly development of SPX price and VIX rate with LSTM using AAII sentiment data and information gained from Form 10-K and Form 10-Q filings?

# 2 Background

This chapter describes the theoretical and methodical background of NLFF. Due to the interdisciplinary nature of this field relevant basics are described and applied methods highlighted.

## 2.1 Financial Forecasting

In developing a robust comprehension it's essential to review foundational concepts. Firstly, the absolute hypothesis of efficient markets is found not to be completely accurate. This is in contrast to the suggested idea that markets are completely random and are not predictable. There are many markets where the predictability is plausible and viable and such markets are termed as weakly efficient. Markets efficiency is correlated with information availability and a market is only strongly efficient when all information is completely available. Financial forecasting tries to use unrevealed information that isn't reflected in the price yet (Nassirtoussi et al. 2014, p. 7654).

Cognitive and behavioural economists look at price as a purely perceived value rather than a derivative of the production cost. Markets participants have cognitive biases such as overconfidence, overreaction, representive bias, information bias, and various other predictable human errors. Inverstor's behaviour can be shaped by whether they feel optimistic (bullish) or pessimistic (bearish) about future market value (Nassirtoussi et al. 2014, p. 7654).

In an effort to reconcile efficient market hypothesis with behavioral finance the adaptive market hypothesis was established. In effect, the linear dependence of stock returns varies over time but nonlinear dependence is strong thoughout (Nassirtoussi et al. 2014, p. 7654).

Technical analysis relies on the assumption that historic market movements are bound to repeat themselves. Most technical analysts don't back their observations up with anything more than stating that patterns exist (Nassirtoussi et al. 2014, p. 7655).

On the other hand, fundamental data available from different sources is used to make assumptions. Fundamental data is usually of an unstructured nature and it remains to be a challenge to make the best use of it efficiently through computing (Nassirtoussi et al. 2014, p. 7655).

## 2.2 Financial Data

| Fundamental Data | Market Data | Analytics | Alternative Data |
|---|---|---|---|
| • Assets<br>• Liabilities<br>• Sales<br>• Costs/earnings<br>• Macro variables<br>• . . . | • Price/yield/implied volatility<br>• Volume<br>• Dividend/coupons<br>• Open interest<br>• Quotes/cancellations<br>• Aggressor side<br>• . . . | • Analyst recommendations<br>• Credit ratings<br>• Earnings expectations<br>• News sentiment<br>• . . . | • Satellite/CCTV images<br>• Google searches<br>• Twitter/chats<br>• Metadata<br>• . . . |

**Figure 1: Types of Financial Data (Prado 2018, p.24)**

The table above shows the four essential types of financial data, ordered from left to right in terms of increasing diversity (Prado 2018, p. 23).

### 2.2.1 Market Data

The forecast models will refer to market data as price and volatility. The VIX measures the market's expectation of future volatility. It's based on options of the SPX, considered the leading indicator of broad U.S. stock market. The VIX is recognized as the world's premier gauge on U.S. equity market volatility. It estimates expected volatility by aggregating the weighted price of SPX puts and calls over a wide range of strike prices. Specifically, the prices used to calculate VIX values are midpoints of real-time SPX option bid/ask price quotations. It's used as a barometer for market uncertainty, providing market participants and observers with a measurement of constant, 30-day expected volatility of the broad U.S. stock market. It's not directly tradable, but the VIX methodology provides a script for replicating volatility exposure with a portfolio of SPX options. Historical Data is provided on the CBOE website.[1]

### 2.2.2 Analytics and Corporate Disclosures

For the experiment setting, analytics as sentiment and textual information will be used. Analytics can be seen as derivative data and thus the signals are extracted from a raw source. However, analytics may be costly, the methodology used in their production may be biased or opaque, and one will not be the sole consumer (Prado 2018, p. 25).

The AAII Investor Sentiment Survey is a weekly survey of its members which asks if they are "Bullish", "Bearish", or "Neutral" on the stock market over the next six months. AAII first conducted this survey in 1987 via standard mail. In 2000, the survey was moved

[1] http://www.cboe.com/products/vix-index-volatility/vix-options-and-futures/vix-index/vix-historical-data (Accessed 4th September 2019)

to AAII's website. Pre-processed data public available on Quandl also includes the SPX weekly price.[2]

**Table 1** Financial texts from different sources and examples

| Type | Characters | Example |
|---|---|---|
| Corporate disclosures | Long length, Subjective tone, Low frequency | Apple Quarter Reports: . . . We are pleased to report third quarter results that reflect stronger customer demand and business performance than we anticipated at the start of the quarter, said Tim Cook, Apple's CEO. . . |
| Financial reports | Long length, Objective tone, Low frequency | Quamnet Portal: Gold prices went through a week of uncertainty due to mixed economic data. First there were weak retail sales data, which led gold prices to surge, yet investors remained uncertain how the data will affect the upcoming decision of the Federal Reserve. . . |
| Professional periodicals | Variable length, Objective tone, Mid frequency | Financial Times: The US Consumer Product Safety Commission issued a formal recall notice for 1 million Samsung Galaxy Note 7 smartphones on Thursday, after nearly a hundred reports of overheating batteries. . . |
| Aggregated news | Mid length, Variable tone, Variable frequency | Yahoo! Finance: Indonesians Declare $8.9 Billion of Singapore Assets for Tax. . . A positive ruling, should remove the uncertainty that may be hampering more participation, said Euben Paracuelles, a Singapore-based economist with Nomura Holdings Inc., in a report Friday. . . |
| Message boards | Short length, Objective tone, High frequency | Amazon's Board: The fact is. . . The value of the company increases because the leader (Bezos) is identified as a commodity with a version for what the future may hold. He will now be a public figure until the day he dies. That is value |
| Social media | Short length, Subjective tone, High frequency | Twitter: $AAPL is loosing customers. everybody is buying android phones! $GOOG |

**Figure 2: Financial Texts from different Sources and Examples (Xing et al. 2018, p. 59)**

[2] https://www.quandl.com/data/AAII/AAII_SENTIMENT (Accessed 4th September 2019)

The Table in Figure 2 categorizes financial text sources into six main groups according to length, subjectivity and the frequency of updates (Xing et al. 2018, p. 58).
The Form 10-K and Form 10-Q used for text analysis belong to the group of corporate disclosures. A 10-K is a comprehensive report, filed annualy by a publicly traded company about its financial performance and is required by the SEC. The report contains much more detail than a company's annual report, which is sent to its shareholders before an annual meeting to elect company directors. Because of the depth and nature of the information they contain, 10-Ks are fairly long and tend to be complicated. But investors need to understand that this is one of the most comprehensive and most important documents a public company can publish on a yearly basis. The more information they can gather from the 10-K, the more they can understand about the company. The SEC requires companies to publish 10-K forms so investors have fundamental information about companies so they can make informed investment decisions. This form gives a clearer picture of everything a company does and what kinds of risks it faces. The 10-K includes five distinct sections:[3]

- **Business** provides an overview of the company's main operations, including its products and services (i.e., how it makes money).
- **Risk factors** outline any and all risks the company faces or may face in the future. The risks are typically listed in order of importance.
- **Selected financial data** details specific financial information about the company over the last five years. This section presents more of a near-term view of the company's recent performance.
- **Management's discussion and analysis** of financial condition and results of operations gives the company an opportunity to explain its business results from the previous fiscal year. Also known as MD&A, this section is where the company can tell its story in own words.
- **Financial statements and supplementary data** include the company's audited financial statements including the income statement, balance sheets, and statement of cash flows. A letter from the company's independent auditor certifying the scope of their review is also included in this section.

A 10-K filing also includes signed letters from the company's chief executive officer and chief financial officer. In it, the executives swear under oath that the information included

[3] https://www.investopedia.com/terms/1/10-k.asp (Accessed 5th September 2019)

in the 10-K is accurate. These letters became a requirement after several high-profile cases involving accounting fraud following the dot-com bust.

Notably, 10-K filings are public information and readily available through a number of sources. In fact, the vast majority of companies include them in the Investor Relations section of their website.

Form 10-Q must be submitted to the SEC on a quarterly basis. Unlike the 10-K, the information in the 10-Q is usually unaudited. The company is only required to file it three times a year as the 10-K is filed in the forth quarter. There are two parts to a 10-Q filing:[4]

- The first part contains relevant financial information covering the period. This includes condensed financial statements, management discussion and analysis on the financial condition of the entity, disclosures regarding market risk, and internal controls.
- The second part contains all other pertinent information. This includes legal proceedings, unregistered sales of equity securities, the use of proceeds from the sale of unregistered sales of equity, and defaults upon senior securities. The company disclosures any other information – including the use of exhibits – in this section.

The 10-Q provides a window into the financial health of the company. Investors can use the form to see what changes are taking place within the corporation even before it files its quarterly earnings. Some areas of interest to investors that are commonly visible in the 10-Q include change of working capital and/or accounts receivables, factors affecting a company's inventory, share buybacks, and even any legal risks that a company faces.

## 2.3 Sentiment and Emotional Analysis

In the context of Text Mining (TM) emotional sentiment preserved in text can be detected through specialized semantic analysis. Opinion mining is mainly based on identifying positive and negative words and processing text with the purpose of classifying its emotional stance as positive or negative (Nassirtoussi et al. 2014, p. 7655).

## 2.4 Text Mining

According to Das TM is the large-scale, automated processing of plain text language in digital form to extract data that is converted into useful quantitive or qualitive information (Das 2013, p. 4).

[4] https://www.investopedia.com/terms/1/10q.asp (Accessed 9th September 2019)

Kumar and Ravi categorized text mining tasks into five categories as depicted in Figure 3.

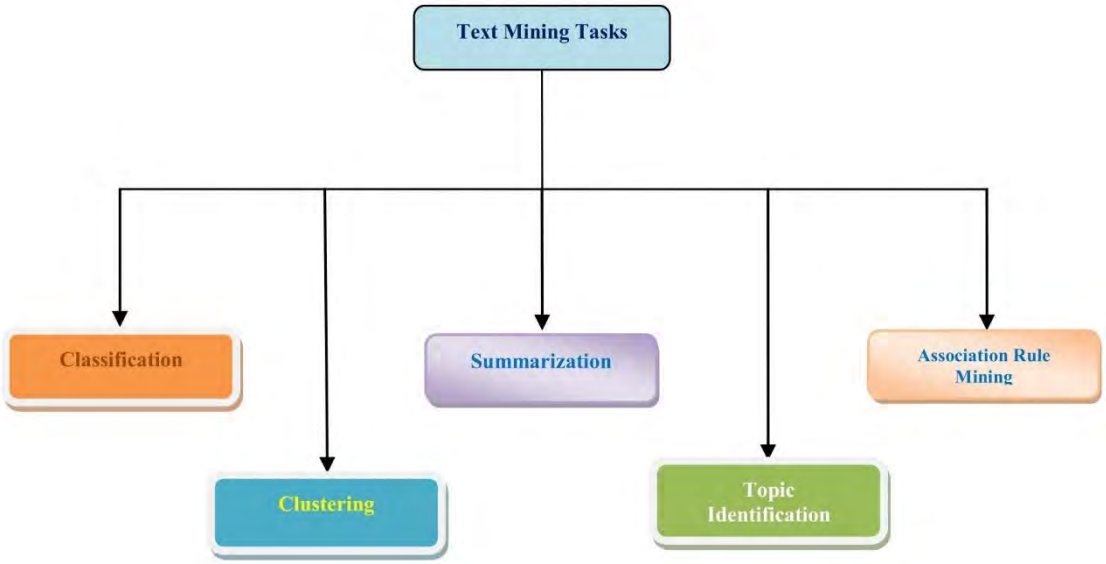


**Figure 3: Primary Tasks of Text Mining (Kumar and Ravi 2016, p. 129)**

A general process of textual analysis includes three steps, namely, harvest text, clean and parse text, and analyse text. The textual data is mostly stored in format file such as txt, xml and pdf which are easy for processing (Guo et al. 2016, p. 154).

### 2.4.1 Text Mining for Financial Market Prediction

Nassirtoussi et al. give a generic overview on the components of TM for financial market prediction.

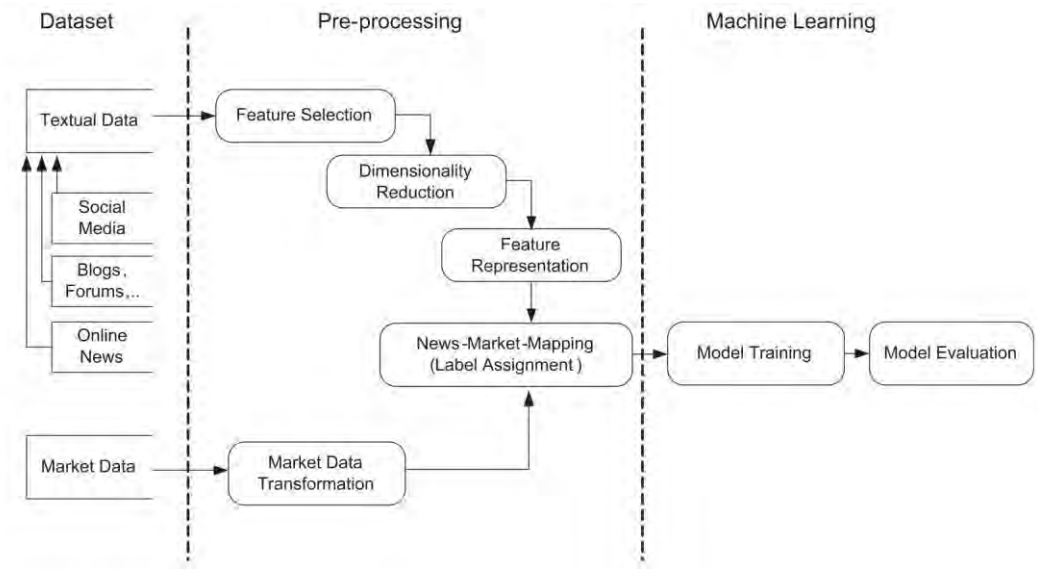


**Figure 4: Generic common System Components Diagram (Nassirtoussi et al. 2014, p. 7656)**

Preprocessing is the transformation of unstructured text into a representative format that is structured and can be processed by ML (Nassirtoussi et al. 2014, p. 7659). This is required because most text is created and stored in a way that only humans can understand (Guo et al. 2016, p. 154). High-quality of preprocessing is always yielded superior results (Kumar and Ravi 2016, p. 129).

Feature selection is crucial to avoid meaningless output (Nassirtoussi et al. 2014, p. 7659). Bag-of-words is the most common technique. It's essentially breaking the text up into words and considering each of the bags as a feature. The order and co-occurrence of words are completely ignored. (Nassirtoussi et al. 2014, p. 7659).

Dimensionality-reduction increases the efficiency of most of the learning algorithms by limiting the number of features. The most common approach is setting a minimum occurrence limit. Next common approach is using a predefined dictionary of some sort to replace them with a category name or number (Nassirtoussi et al. 2014, p. 7659).

Each feature needs to be represented by a numeric value so that it can be processed by ML algorithms. This assigned numeric value acts like a score or a weight. The most basic technique is a Boolean or a binary representation whereby two values like 0 and 1 represent the absence or presence of a feature. The next most common technique is the Term Frequency-Inverse Document Frequency (TF-IDF). This value increases proportionally to the number of times a word appears in the document, but is offset by the frequency of the word in the corpus, to balance out the general popularity of some words (Nassirtoussi et al. 2014, p. 7660).

### 2.4.2 Master Dictionary

In addition to providing a master word list, the dictionary used to create the input data set includes statistics for word frequencies in all 10-K/Q's from 1994-2018. The dictionary reports counts, proportion of total, average proportion per document, standard deviation of proportion per document, document count (i.e., number of documents containing at least one occurence of the word), eight sentiment category identifiers, Harvard Word List identifier, number of syllables, and source for each word. The sentiment categories are: negative, positive, uncertainty, litigious, modal, constraining. Modal words are flagged as 1, 2 or 3, with 1 = Strong Modal, 2 = Moderate Modal, and 3 = Weak Modal. The other sentiment words are flagged with a number indicating the year in which they were added to the list.[5]

[5] https://sraf.nd.edu/textual-analysis/resources/ (Accessed 14th October 2019)

### 2.4.3 File Summaries

The models use a file containing sentiment counts, file size and other measurements for all 10-K and 10-Q filings generally stated as 10-X filings from 1994 to 2018. This file contains a header record with labels and is comma delimited[6]. Each record reports:

1. CIK – the SEC Central Index Key.
2. FILING_DATE – the filing date (YYYYMMDD) for the form.
3. FYE – fiscal-year-end as reported in the filing.
4. FORM_TYPE – the specific from type (e.g., 10-K, 10-K/A, 10-Q405, etc.).
5. FILE_NAME – the local file name for the filing.
6. SIC – the four digit SIC reported in the header of the filing. If this number does not appear in the header, then the primary web page for all filings from that firm at EDGAR is parsed in an attempt to identify the SIC number. If all of these methods fail, an SIC of -99 is assigned.
7. FFInd – the Fama-French 48 industry classification based on the SIC number. All missing SIC's are assigned to the miscellaneous category.
8. N_Words – the count of all words, where a word is any token appearing in the Master Dictionary.
9. N_Unique – the number of words occurring at least once in the document.
10. A sequence of sentiment counts – negative, positive, uncertainty, litigious, weak modal, moderate modal, strong modal, constraining.
11. N_Negation – a count of cases where the negation occurs within four or fewer words from a word identified as positive. Negation words are no, none, neither, never, nobody (see Gunel Totie, 1991, Negation in Speech and Writing). Thus the net positive words is the positive word count minus the count for Negation. Although the technique seems reasonable, most important cases of negation are sufficiently subtle that most algorithms will not pick them up.
12. GrossFileSize – the total number of characters in the original filing.
13. NetFileSize – the total number of characters in the filing after the Stage One Parse.
14. ASCIIEncodedChars – the total number of American Standard Code for Information Interchange (ASCII) Encoded characters (e.g., &).
15. HTMLChars – the total number of characters attributable to Hypertext Markup Language (HTML) encoding.

[6] https://drive.google.com/file/d/12YQ3bczd3-G94eSpqawbA1hwF0Jzs_jB/view?usp=sharing (Accessed 9th September 2019)

16. XBRLChars – the total number of characters attributable to eXtensible Business Reporting Language (XBRL) encoding.
17. XMLChars – the total number of characters attributable to Extensible Markup Language (XML) encoding.
18. N_Tables – number of tables in the filing.
19. N_Exhibits – number of exhibits in the filing.

## 2.5 Machine Learning

While the lexicon-based approach for classification was already described it's also possible to train software to classify text or recognize pattern with ML method (Guo et al. 2016, p. 155). Figure 5 shows the most popular classification methodologies employed by accounting and finance researchers.

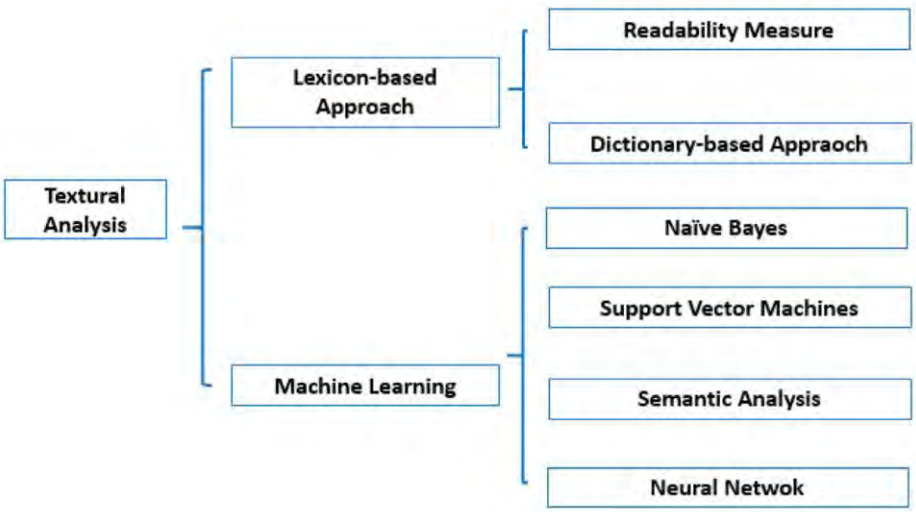


**Figure 5: General Method for Textual Analysis (Guo et al. 2016, p. 155)**

### 2.5.1 Machine Learning Techniques

These algorithms are also amongst those used for prediction. A predictive model whose development is based on both input and output data is considered as supervised learning. The aim of supervised ML is to build a model that makes predictions based on evidence in the presence of uncertainty. A supervised learning algorithm takes a known set of input data and known responses to the data (output) and trains a model to generate reasonable predictions for the response to new data. Supervised learning uses classification and regression techniques to develop predictive models.

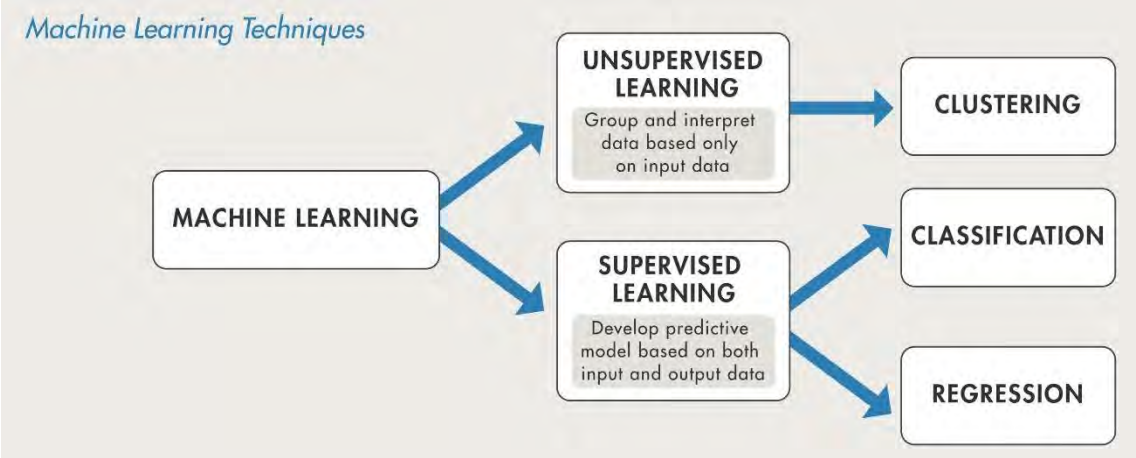


**Figure 6: Machine Learning Techniques (MathWorks, p. 4)**

In financial forecasting context classification algorithms use the input data to learn classify an output usually in terms of the movement of the market in classes such as Up, Down and Steady. However, for exact value predictions regression analysis is used (Nassirtoussi et al. 2014, p. 7661).

Regression models are particulary preferred since one can explicitly observe the impact of each factor included and analyse the importance of variables by dropping them out (Xing et al. 2018, p. 62).

### 2.5.2 Machine Learning Algorithms

Figure 7 is attempted to provide a summary of the ML algorithms.

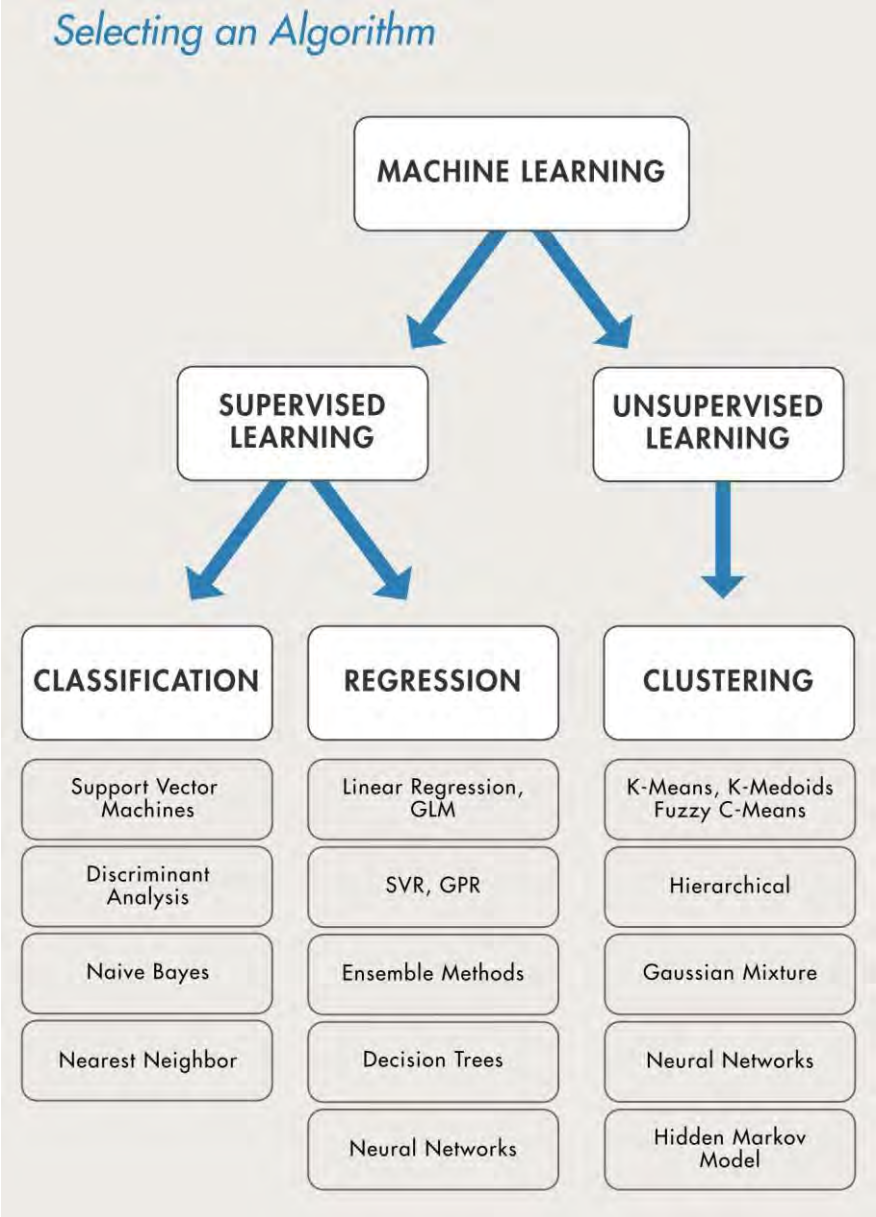


**Figure 7: Selecting an Algorithm (MathWorks, p. 7)**

Support Vector Machines (SVM) have been extensively and successfully used as a textual classification and sentiment learning approach while some other approaches like Artificial Neural Networks (ANN), k-Nearest Neighbour (k-NN) have rarely been considered in the text mining literature for market prediction. Research results of empirical SVM and ANN regarding document-level sentiment analysis show that ANN can produce superior or at least comparable results to SVM's (Nassirtoussi et al. 2014, p. 7663).

Considering the volume and quality of data available, overly complicated models generally have a poor performance. However, one drawback of linear models is that they rely

on strong hypotheses, for example, a Gaussian distribution of dependent variables, which does not always stand up in real world cases. In many studies, the features generated from the texts are combined with numerical data to form a robust data stream for prediction. In this case an ensamble method can be used to manipulate the combination either on a feature level or a decision level (Xing et al. 2018, p. 62).

### 2.5.3 Machine Learning for Natural Language based Financial Forecasting

It is still an open question as to what category of algorithms is especially appropriate for NLFF.

| Reference | Feature formatting | Model type | Implementation |
|---|---|---|---|
| Wuthrich et al. (1998) | Number of tuples occurrences | Naïve Bayes & Association rules | Experimentally tuned k-NN |
| Lavrenko et al. (2000) | Trend possibility distribution | n-gram language model | Conditional probability maximization |
| Fung et al. (2003) | Tf-idf weighted key words | Support vector machine | Split-and-merge segmentation |
| Antweiler and Frank (2004) | Text classification | Regressions | Variable & Lag tuning |
| Das and Chen (2007) | Lexicon occurrences | Classifiers voting | Discriminant values |
| Tetlock et al. (2008) | Lexicon based sentiment score | Regressions | Ordinary least square & Dependent variables |
| Schumaker and Chen (2009) | Binary representation | Support vector regression | Sequential minimal optimization |
| Bollen et al. (2011) | Temporal mood indicator | Self-organized fuzzy neural network | Online learning |
| Chan and Franklin (2011) | Textual information database | Inference engine | Multiple decision tree classifiers |
| Groth and Muntermann (2011) | Labelled lexicon occurrences | Ensemble learning | NB, k-NN, NN, SVM with tuning |
| Ruiz et al. (2012) | Graph features | Vector autoregression | Least square regression |
| Schumaker et al. (2012) | Proper Nouns | Support vector regression | Sequential minimal optimization |
| Si et al. (2013) | Topic based sentiment score | Vector autoregression | Least square regression |
| Si et al. (2014) | Lexicon based sentiment score | Vector autoregression | Least square regression |
| Li et al. (2014b) | Tf-idf weighted key/senti words | Support vector regression | Sequential minimal optimization |
| Ding et al. (2015) | Sequence of event embeddings | Convolutional neural network | Margin loss minimization |
| Nofer and Hinz (2015) | Weighted Social Mood Index | Vector autoregression | Minimum information criterion |
| Nguyen et al. (2015) | Topic model parameters | Support vector machine | Linear kernel soft margin |
| Yoshihara et al. (2016) | Temporal news embeddings | Deep belief network | Greedy layer-wise training |

**Figure 8: Algorithms involved and the Implementation Details (Xing et al. 2018, p. 63)**

From the table in Figure 8, mainstream algorithms can be placed into four categories: regressions, prohabilistic inferences, and neural networks, or a hybrid of them.
Selecting a ML algorithm is a process of trial and error. It's also a trade-off between special characteristics of the algorithms, such as:

- Speed of training
- Memory usage
- Predictive accuracy on new data
- Transparency or interpretability (how easily one can understand the reasons an algorithm makes its predictions).

### 2.5.4 Long Short-Term Memory Networks

The models of this study use LSTM neural networks. A general neural network consists of three types of layers: input layer, hidden layer and output layer. Each layer also consists of multiple units. The decision rule makes the units in the subsequent layers digest all information from the training sample by weighting all units of current layer (Guo et al. 2016, p. 163).

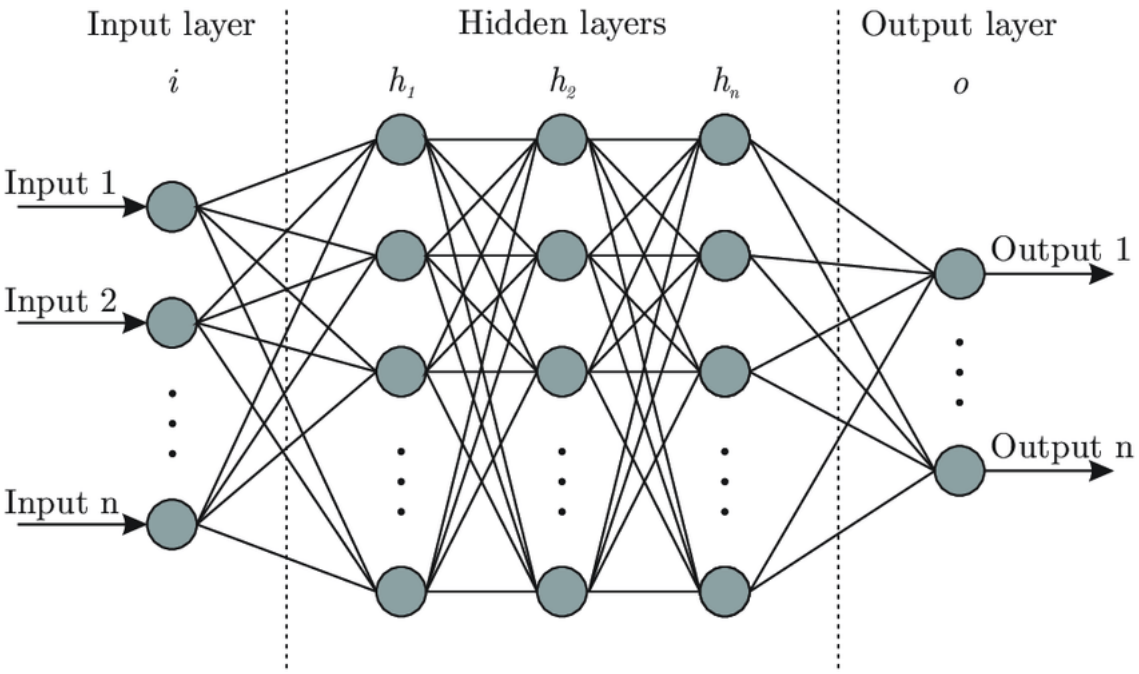


**Figure 9: Artificial Neural Network Structure**

Source: https://www.researchgate.net/figure/Artificial-neural-network-architecture-ANN-i-h-1-h-2-h-n-o_fig1_321259051 (Accessed 14th October 2019)

In practice, ANN seems to be a powerful data-driven, self-adaptive, flexible computational tool that can capture nonlinear and complex underlying characteristics of any phys-

ical process with a high degree of accuracy. It can handle large amount of data sets, implicitly detect complex nonlinear relationships between dependent and independent variables and even detect all possible interactions between predictor variables. Studies found that lexicon based approach predict short-term returns that are quickly reversed while ANN predicts larger and more persistant returns (Guo et al. 2016, p. 166).

LSTM networks are a special kind of RNN, capable of learning long-term dependencies. All RNNs have the form of a chain of repeating modules of neural networks. In standard RNNs, this repeating module will have very simple structure, such as single tanh layer.

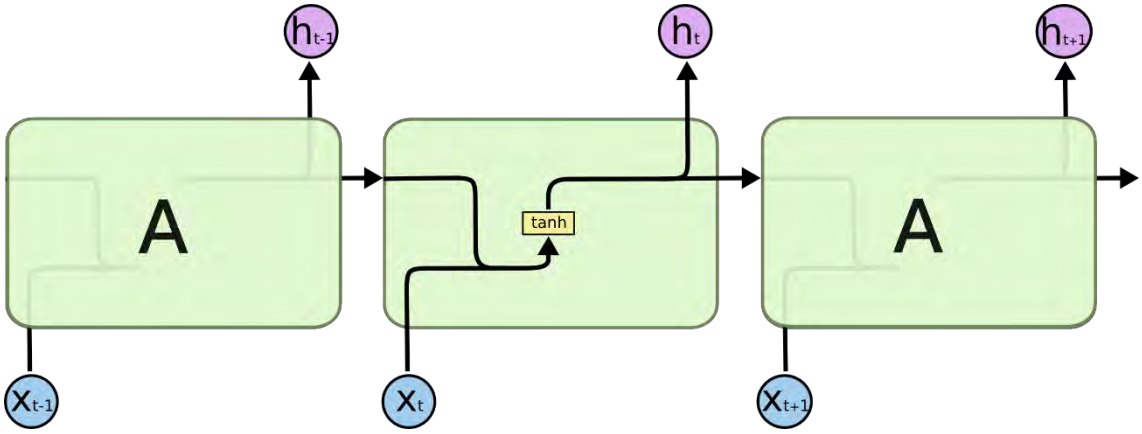


**Figure 10: The Repeating Module in a standard RNN contains a Single Layer**

Source: https://colah.github.io/posts/2015-08-Understanding-LSTMs/img/LSTM3-SimpleRNN.png (Accessed 14th October 2019)

LSTMs also have this chain like structure, but the repeating module has a different structure. Instead of having a single neural network layer, there are four, interacting in a very special way.

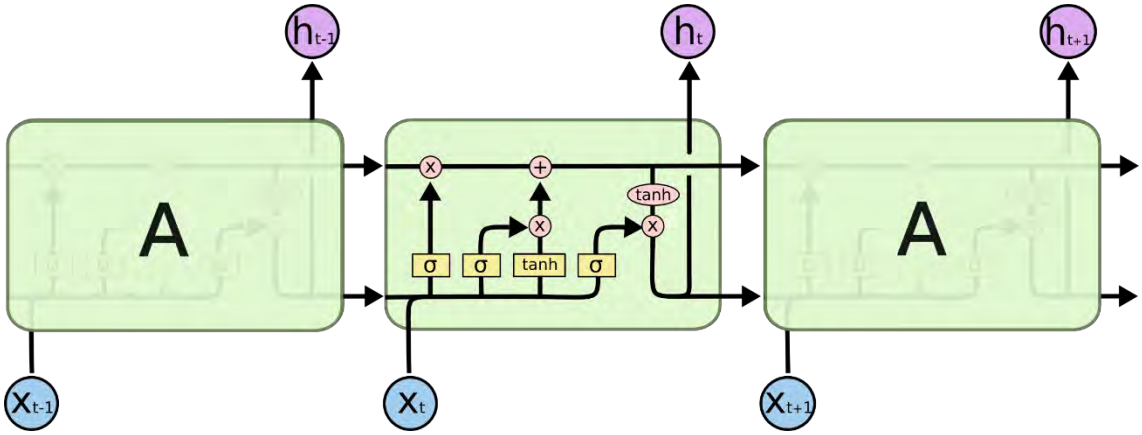


**Figure 11: The Repeating Module in a LSTM contains four Interacting Layers.**

Source: https://colah.github.io/posts/2015-08-Understanding-LSTMs/img/LSTM3-chain.png (Accessed 14th October 2019)

In the diagram, each line carries an entire vector, from the output of one node to the inputs of others. The pink circles represent pointwise operations, like vector addition, while the yellow boxes are learned neural network layers. Lines merging denote concatenation, while a line forking denotes its content being copied and the copies going to different locations.

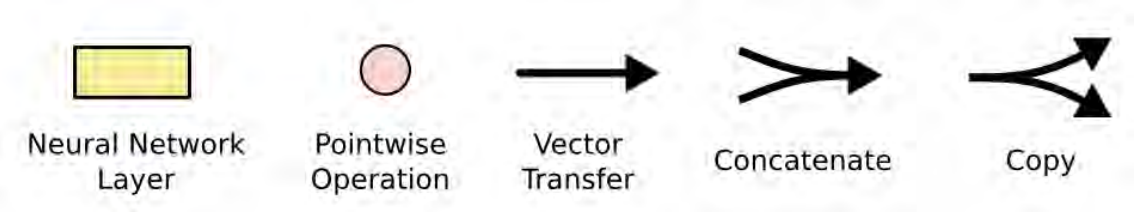


**Figure 12: Notation of LSTM Diagram**

Source: https://colah.github.io/posts/2015-08-Understanding-LSTMs/img/LSTM2-notation.png (Accessed 14th October 2019)

There's a detailed step-by-step walk through LSTM essay for the interested reader.[7]
Since LSTMs are able to recognize and evaluate patterns autonomously they belong to deep learning algorithms. LSTM demonstrated to be a good fit for pattern recognition problem in sequence data which are inherently difficult to paralyse.

[7] https://colah.github.io/posts/2015-08-Understanding-LSTMs/ (Accessed 14th October 2019)

# 3 Methodology

This chapter describes the development of the forecasting model. Since the data scraping was already described in the Background chapter this section starts with the data preparation. The models are coded in Python because it's open source and has a rich suite of pre-coded routines to facilitate development. To make sure that the modules are compatible with the Python version embedded the models are scripted in Jupyter Notebook available on the Anaconda platform. For example, NumPy and SciPy provide scientific tools, Matplotlib facilitates plotting, pandas is used for data structures and analysis, and NLTK provides a natural language toolkit. The advantage of these packages is that many times complex problems can be solved in a very small number of statements. For sharing and discussing reasons, these files have been uploaded into a IBM Watson Studio project.[8] In total, four models are developed:

1. AAII_LSTM.ipynb
2. AAII_VIX_LSTM.ipynb
3. LM_10X_LSTM.ipynb
4. LM_10X_VIX_LSTM.ipynb

The first two models refer to the AAII sentiment data set available on Quandl. The last two models use the Loughran and McDonald 10X file summaries. Model 1 and 3 predict the development of the SPX while model 2 and 4 forecast the VIX development.

## 3.1 Data Preperation

Success in the later phases is dependent on what occurs in earlier phases. It is not surprising, therefore, that poor data quality will lead to poor model performance.

Data preparation is one of the most important and often time-consuming phases of a data science project. In fact, it is estimated that data preparation ususally takes 50-70% of a project's time and effort, leaving very little time to uncover new insights and deliver them.[9]

Data preparation involves cleaning the data and reshaping it into a usable form for performing data science. Examples of common data preparation activities includes dealing with non-standard, unstructured or inconsistent data and combining data from different sources and formats.

[8] https://dataplatform.cloud.ibm.com/projects/22ebb973-531a-484e-b119-f67d435437e9/assets?context=wdp

[9] https://public.dhe.ibm.com/software/data/sw-library/analytics/data-science-lifecycle/ (Accessed 9th October 2019)

The pre-requiste Python packages are: Keras, sklearn, LSTM, Pandas, Seaborne, Matplotlib, and re. The setup also includes setting a random seed to make results reproducible, as all the random numbers generated will allways be the same.

### 3.1.1 Data Cleaning

The first step is reading the csv file. Once it's imported one drops the not required columns. The AAII_Sentiment file is given in descending date order. For the forecasting the data set must be sorted by dates in ascending order and set as an index. For the LM_10X file the date format must be changed first and can then be set as an index.
The AAII_Sentiment file already contains historic SPX price data for model 1 while this needs to be added to the LM_10X file for model 3.
Since there're two csv files for VIX data these have to be concated first. Afterwards the VIX file is merged by date column through inner join with the AAII_Sentiment for model 2 and LM_10X file respectively for model 4.

### 3.1.2 Preprocessing

Finally, the step after any analysis is converting the data to make it digestible for the LSTM model. Which means converting it from a timeseries data into supervised sequence, a 3D array as such with normalized variables for which MinMaxScaler is used. The timelag is seven days and number of features is four (three sentiment categories plus SPX or VIX rate) for model 1 and 2. For model 3 and 4 timelag is twelve weeks which is one quarter of a year and number of features is twelve (eight sentiment counts and three additional textual measurements plus SPX or VIX rate). The target value is the close price of SPX or VIX in each case. This very important step happens in the scripts just before fitting the model parameters.

## 3.2 Data Exploration

Once the data is in right format to work with, one can conduct the next step in the data analysis process: data exploration. This initial exploration of the dataset is critical because it helps illuminate previously unknown patterns, relationships, or other actionable findings.
Data visualisations are commonly used to quickly view relevant features of datasets and identify variables that are likely to result in interesting observations. By displaying data graphically - for example, through scatter plots or bar charts – users can see if two or more variables correlate and determine if they are good candidates for more in-depth analysis.

### 3.2.1 Time Series Plot

A plot of the time series gives an insight to the whole data where you can eyeball some small patterns in the data.

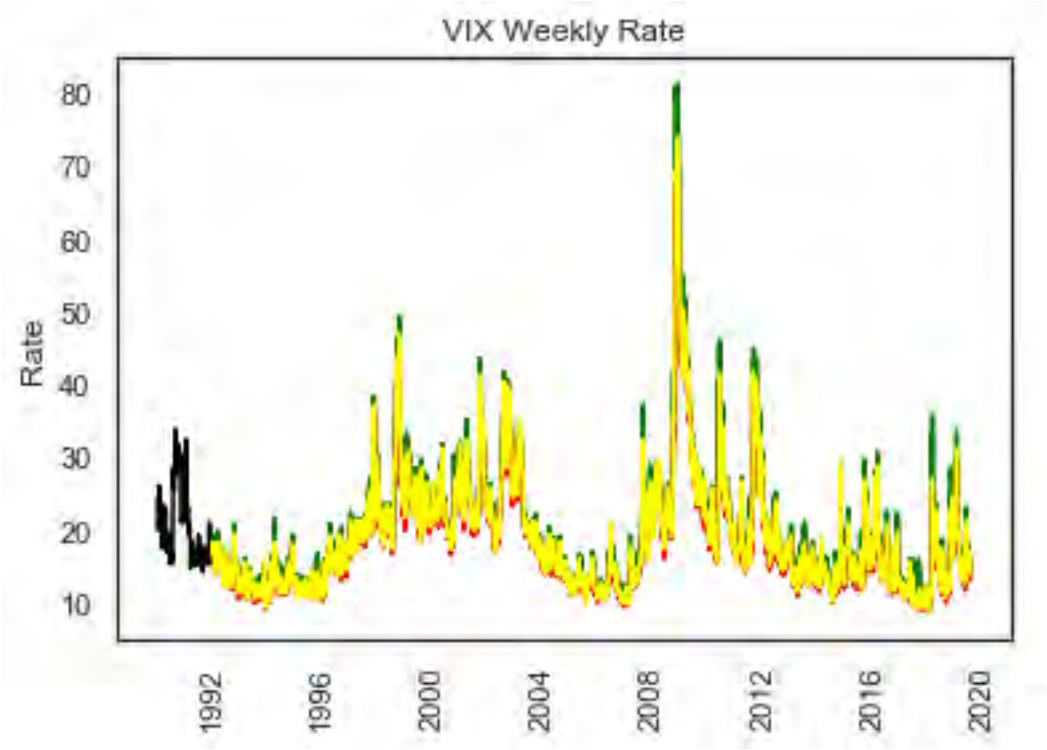


**Figure 13: VIX Weekly Rate (black = Close, green = High, red = Low, yellow = Open)**

The VIX time series fluctuates between certain barriers with ups and down. This development is similar to the LM_10X and AAII_Sentiment time series. The first one counts sentiment categories while the second portions the sentiment into bullish, neutral and bearish with the total sum being 1.


**Figure 14: SPX Weekly Price (black = Close, green = High, red = Low)**

On the other hand, the SPX has an overall upward trend. This is important to notice for the later train test split. If the split is done amongst the time axis the train set contains lower values of the beginning while the test set contains higher values of the end. This non-stationary nature is one of the largest challenges in financial forecasting and will be faced in the model management.

### 3.2.2 Description

The describe function gives a more detailed insight into key figures as count, mean, standard deviation, minimum, and maximum. Also the median should be checked since it's the statistically seen cleaner metric than the mean.

Although the AAII_Sentiment has a higher frequency (weekly) and longer history (since 1987) the LM_10X (quarterly and since 1993) has multiple more time stemps. This is due to the fact that AAII_Sentiment collects data for the stock market in general while the LM_10X involves all companies listed. For the SPX it's 1665 AAII_Sentiment counts compared to 1,022,593 LM_10X counts and for the VIX it's 1508 AAII_Sentiment counts compared to 1,022,399 LM_10X counts. These differences between sentiment counts for SPX and VIX might occur due to different data availability. Data availability and quality is crucial in ML and especially deep learning algorithms as LSTM are data

thirsty. As a thumb rule, the more data is available the better the results for the models will be.

### 3.2.3 Correlation Plot

A correlation plot gives a clearer indication of the features that may be more important than others. Which then helps derive an insight into how the LSTM predicts.

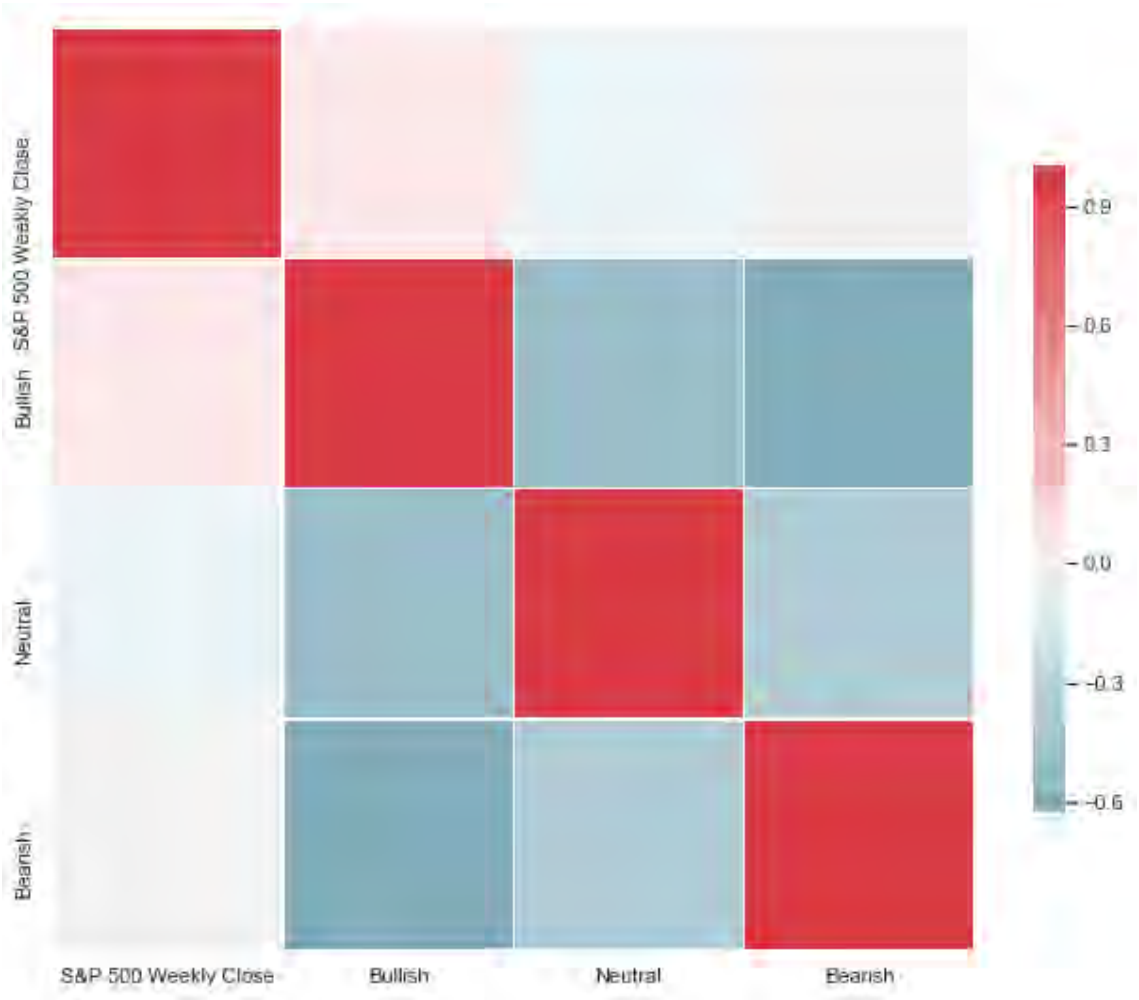


**Figure 15: AAII_Sentiment & SPX Correlation Plot**

In model 1 the AAII_Sentiment data does not have notable correlations with the SPX as all of them occur on the second decimal place.

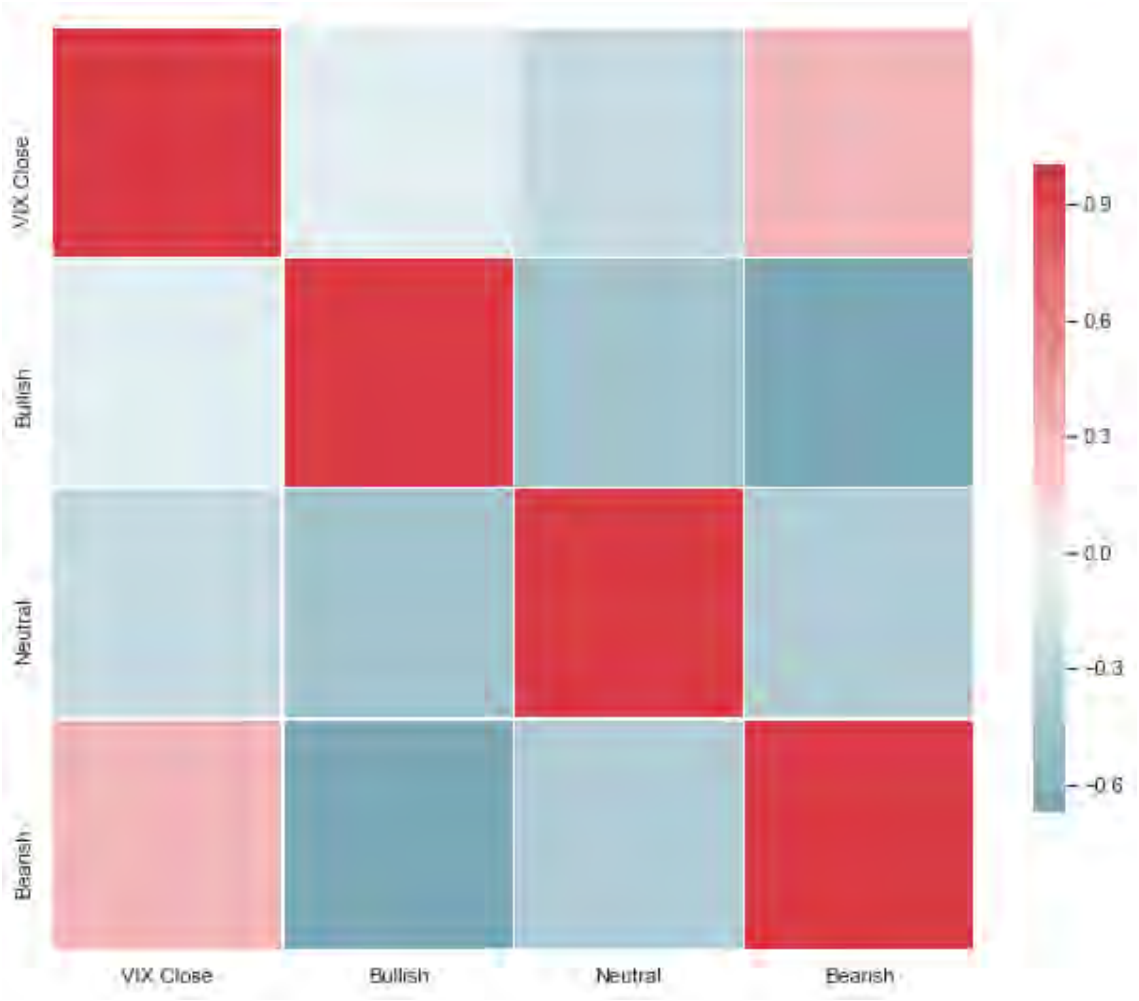


**Figure 16: AAII_Sentiment & VIX Correlation Plot**

On the contrary, in model 2 the AAII_Sentiment data has weak correlations with the VIX. The strongest being Bearish correlating the Close at 0.32. In both models it's surprising that the correlations between SPX or VIX and Neutral are the most negative.

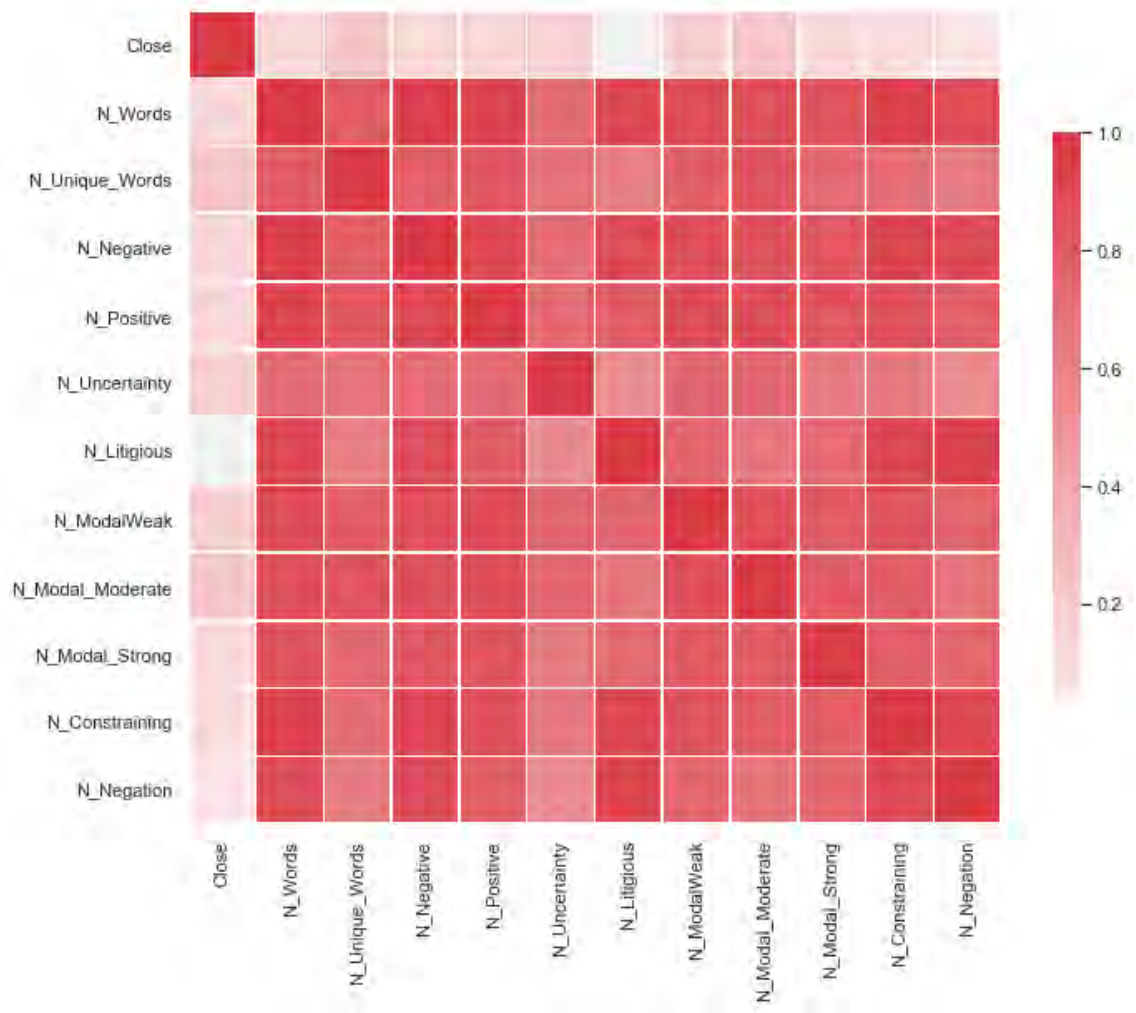


**Figure 17: LM_10X & SPX Correlation Plot**

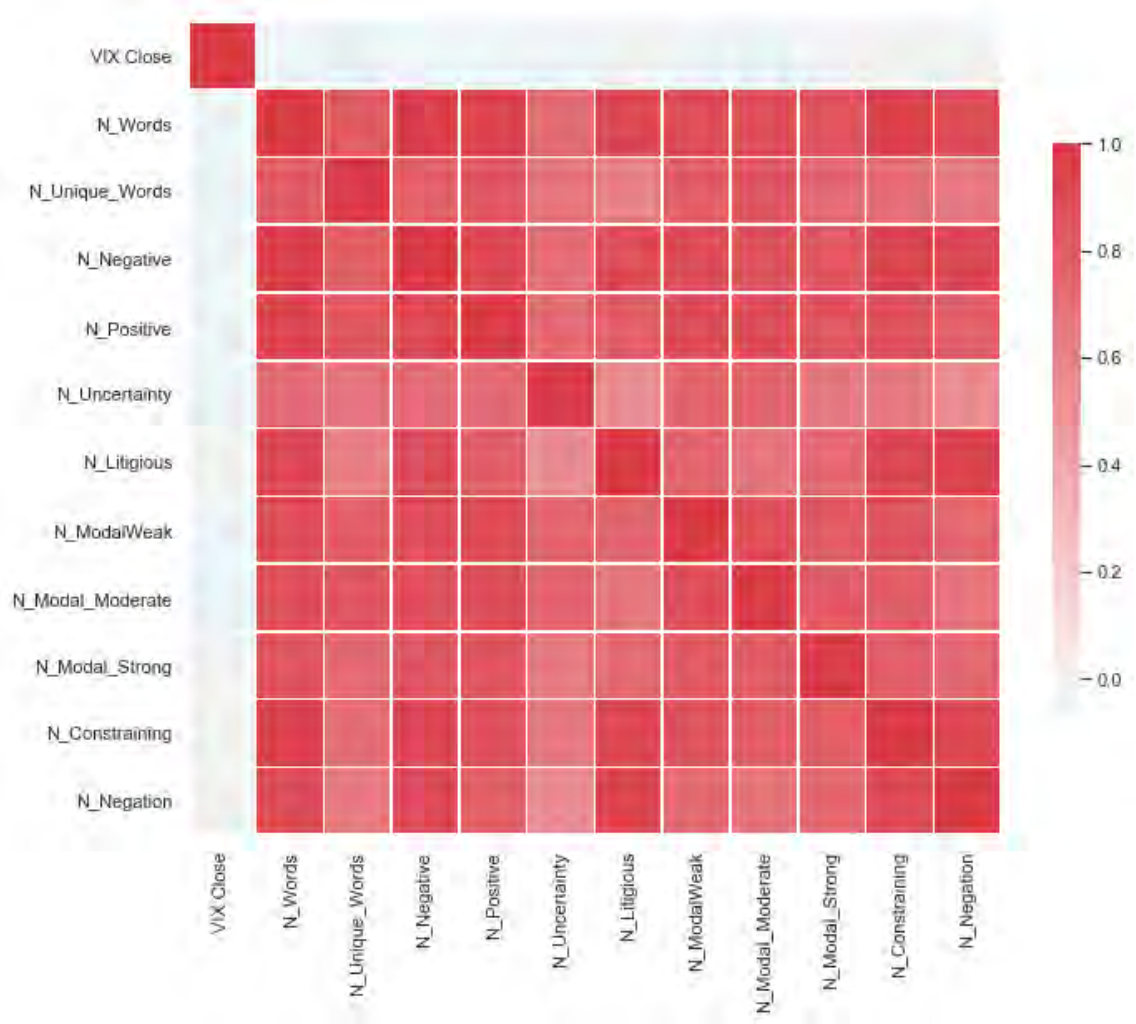


**Figure 18: LM_10X & VIX Correlation Plot**

These finding are reversed by model 3 and 4. In model 3 the LM_10X data has weak correlation with the SPX while the correlations with the VIX in model 4 only occur in the second decimal place. Furthermore, all the features of the LM_10X data have positive correlations with the SPX in model 3 while all those with the VIX in model 4 are negative. In both models N_Unique_Words has the strongest correlation: 0.24 with the SPX in model 3 and -0.06 with the VIX in model 4.

## 3.3 Model Development

Model development is usually conducted in multiple iterations. Typically, data scientists run several models using default parameters and then fine-tune the parameters or revert to the data preperation phase for manipulations required by their model of choice.

### 3.3.1 Design Network

First the Keras Sequential model is set which is a linear stack of layers. Afterwards the LSTM with five layers is added. The Dense of one is added since we target a single output value. The model is compiled with loss function Mean Absolute Error (MAE) and Adam optimizer which is derived from adaptive moment estimation.

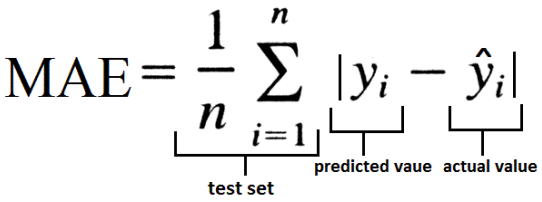


**Figure 19: Mean Absolute Error**

https://www.datavedas.com/wp-content/uploads/2018/04/image017-300x105.png (Accessed 14th October 2019)

In short, Adam combines the advantages of the Adaptive Gradient Algorithm (AdaGrad) and Root Mean Square Propagation (RMSProp) which makes it a popular algorithm in the field of deep learning because it achieves good results fast. For futher details on how it works and its attractive benefits please see the paper of Kingma and Ba.[10] As metrics Mean Squared Error (MSE) and Mean Absolute Percentage Error (MAPE) are used to further check for the optimal epoch size. The MSE is the stricter metric compared to MAE because it punishes strong deviations harder. The MAPE as a relative metric makes the models more comparable.

$$\mathrm{MSE} = \frac{1}{n}\sum_{i=1}^{n}\left(x_i - \hat{x}_i\right)^2$$

$$\mathrm{MAPE} = \frac{1}{n}\sum_{i=1}^{n}\left|\frac{x_i - \hat{x}_i}{x_i}\right|$$

**Figure 20: Mean Squared Error and Mean Absolute Percentage Error (Xing et al. 2018, p. 65)**

Also, in order to make the results comparable to the MAE the Root Mean Squared Error (RMSE) is calculated while predicting.

Since the LSTM aims to learn the sequence of patterns it doesn't shuffle input patterns during training.

[10] https://arxiv.org/abs/1412.6980 (Accessed 14th October 2019)

### 3.3.2 Fit Network

The empirical advices for the train test split range from 80% train and 20% test to 1/3 train and 2/3 test. The models use 70% in sample train data and 30% out of sample test data. The model parameters are trained on 50 epochs, the validation split is 0.2 and batch size is set as 3% of the number of train days. One epoch is when an entire dataset is passed forward and backward through an ANN only once. Note that the epoch size should be fit for each model individually but for comparison reasons 50 epochs seemed to be a good compromise. The validation split defines the fraction of training data to be used as validation data. The batch size is the total number of training examples present in a single batch. Since one epoch is too big to pass into an ANN at once it's necessary to divide the dataset into number of batches.

## 3.4 Model Implementation

For a predictive model the implementation involves using a testing set that's independent of the training set but follows the same probability distribution and has known outcome. The testing set is used to evaluate the model so it can be refined as needed.

## 3.5 Model Management

During deployment and integration of modelling results, model management must be a continuous process to ensure optimal performance over time. Improving a model means increasing its accuracy and predictive power and preventing overfitting. A model is overfitted when it can't distinguish between data and noise. Model improvement involves feature engineering (feature selection and transformation) and hyperparameter tuning.

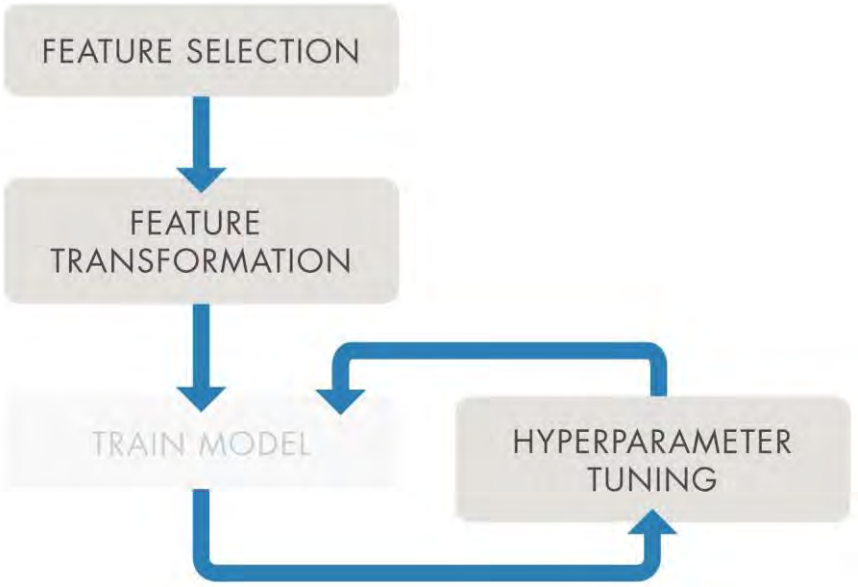


**Figure 21: Improving Models (MathWorks, p. 16)**

Feature selection is identifying the most relevant features, or variables, that provide the best predictive power in modelling data. This could mean adding variables to the model or removing variables that do not improve model performance.

Feature transformation means turning existing features into new features using techniques such as principal component analysis, nonnegative matrix factorisation, and factor analysis.

Hyperparameter tuning describes the process of identifying the set of parameters that provides the best model. Hyperparameters control how a ML algorithm fits the model to the data.

### 3.5.1 Hyperparameter Tuning

One way of improving the model is deepening it by adjusting the number of its layers. The original LSTM models are compromised of a single hidden LSTM layer followed by a standard feedforward output layer. The stacked LSTM is an extension to these models that has multiple hidden LSTM layers where each layer contains multiple memory cells. Given that LSTM operate on sequence data, it means that the addition of layers adds levels of abstraction of input observations over time. In effect, chunking observations over time or representing the problem at different time scales. There's an online tutorial on how to implement it as well.[11]

[11] https://machinelearningmastery.com/stacked-long-short-term-memory-networks/ (Accessed 9th October 2019)

Another approach is adjusting the number of units. The number of hidden units is a direct representation of the learning capacity of a neural network. It reflects the number of learned parameters. One could change the value experimentally and rerun the program to see how it affects the training accuracy. Using more units makes it more likely to perfectly memorize the complete training set. Although, it will take longer, and one runs the risk of overfitting.

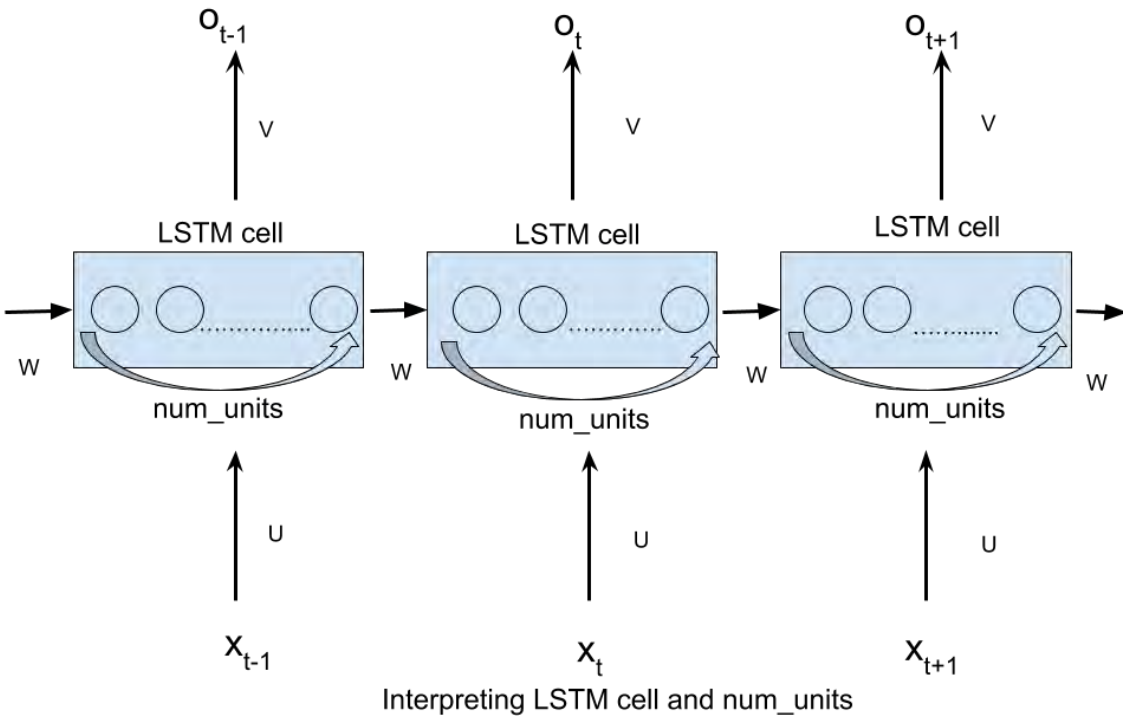


**Figure 22: Interpreting LSTM cell and num_units**

Source: https://raw.githubusercontent.com/jasdeep06/jasdeep06.github.io/master/posts/Understanding-LSTM-in-Tensorflow-MNIST/images/num_units.png (Accessed 10th October 2019)

The LSTMs of the models are stateless. The benefit of LSTM networks is their ability to maintain state and learn a sequence. The key to understand the difference between stateful and stateless LSTMs is when internal state is reset.

In the stateless LSTM configuration, internal state is reset after each training batch or each batch when making predictions. In the stateful LSTM configuration, internal state is only reset when the reset_state() function is called.

In consequence, it may be possible to simulate a stateful LSTM with a stateless LSTM using a larger batch size. As already mentioned before, the batch size defines the frequency of updating weights. For big time series as in financial markets the lookback window length is crucial. Consequently, Bayesian Optimisation might be an option.

It's suggested that the chosen LSTM configuration is focused more on learning input-output pairs rather than dependencies within sequence. That's why the stateless LSTM may outperform the stateful configuration and the shuffling of samples won't make a large difference to the stateless LSTM.[12]

Moreover, the number of epochs offers another opportunity of tuning. In order to find the fit number of epochs, it's helpful to plot the history of fitting the network. The optimal number of epochs is when the training and testing graph reach their minimum. By the moment of an increase on validation data the network is likely overfitted.


**Figure 23: Spot Overfitting by Number of Epochs**

Source: https://i.stack.imgur.com/pJU0X.png (Accessed 14th October 2019)

To sum it up, hyperparameter tuning is an essential step in fitting an ML algortihm. When this isn't done properly, the algorithm is likely to overfit, and live performance will disappoint (Prado 2018, p. 129). The simplest way of optimization is manual try and error of different configurations. Certainly, this approach fastly reaches its boarders in complex modelling. Automation and minimisation of training iterations can be achieved through stochastic methods as the alredy mentioned Bayesian Optimisation. Prado recommends the purged k-fold cross-validation (CV).

### 3.5.2 Dropout and Backtesting

As the improvement of predictive power and performance have been faced overfitting remains an accompanying issue. Models suffer from overfitting when they capture spurious patterns that won't occur in the future, leading to less accurate predictions. On the opposite, underfitting is when they fail to capture relevant patterns and again leading to

[12] https://machinelearningmastery.com/stateful-stateless-lstm-time-series-forecasting-python/ (Accessed 10th October 2019)

less accurate predictions. One way of preventing this issue is the penalisation of network complexity.

A single model can be used to simulate having a large number of different network architectures by randomly dropping out nodes during training. This is called dropout and offers a very computionally cheap and remarkably effective regularisation method to reduce overfitting and improve generalisation error in deep neural networks of all kinds. During training, some number of layer outputs are randomly ignored or dropped out. This has the effect of making the layer look-like and be treated-like a layer with a different number of nodes and connectivity to the prior layer. In effect, each update to a layer druing training is performed with different view of the configured layer. Dropout has the effect of making the training process noisy, forcing nodes within a layer to prohabilistically take on more or less responsibility for the inputs. This conceptualisation suggests that perhaps dropout breaks-up situations where network layers co-adapt to correct mistakes from prior layers, in turn making the model more robust. Dropout simulates a sparse activation from a given layer, which interestingly, in turn, encourages the network to actually learn a sparse representation as a side-effect. As such, it may be used as an alternative to activity regularisation for encouraging sparse representations in autoencoder models. Because the output of a layer under dropout are randomly subsampled, it has the effect of reducing the capacity or thinning the network during training. As such, a wider network, e.g. more nodes, may be required when using dropout.[13]

A backtest evaluates out-of-sample the performance using past observations. It's called walk-forward (WF) if past observations are used to simulate the historical performance, as if it had been run in the past. In a broader sense, past observations can be used to simulate scenarios that did not happen in the past (Prado 2018, p. 161). The main drawback of the WF and CV method namely is that those schemes test a single path. This is addressed by the combinatorial purged cross-validation (CPCV). Given a number of backtest paths targeted by the researcher, CPCV generates the precise number of combinations of training/testing sets needed to generate those paths, while purging training observations that contain leaked information (Prado 2018, p. 163). For a large enough number of paths, CPCV could make the variance of the backtest so small as to make the probalility of a false discovery negligible (Prado 2018, p. 167). For detailed reasons why backtest overfitting may be the most important open problem in all of mathematical finance please see Chapter 11 of Prado.

[13] http://jmlr.org/papers/v15/srivastava14a.html (Accessed 14th October 2019)

# 4 Evaluation & Argumentation

In this section, an estimation of expected outcome and the evaluation of observed results is addressed. In addition, the comparability and added value are discussed.

## 4.1 Expected Outcome

There are three commonly acknowledged measurements to evaluate forecasting results.

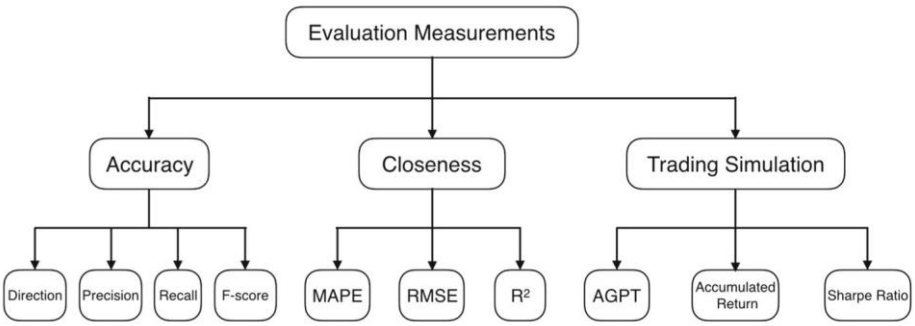


**Figure 24: Taxonomy of Measurements reported (Xing et al. 2018, p. 66)**

For directional accuracy, where the forecasting is simply represented in a binary up/down, the rate should at least differ significantly from 50% or improve on a benchmark method to be more convincing (Xing et al. 2018, p. 65).

Closeness measurement between the forecasted time series and the corresponding real world time series should demonstrate a reduction of errors which are used as metrics (Xing et al. 2018, p. 65).

Trading simulation results can be measured by specific metrics for the profits as average percentage gain per transaction (AGPT), accumulated profit for a certain period, profit ratio, or portfolio performance. Attention should be given to the deduction of transaction costs as this makes results more comparable (Xing et al. 2018, p. 65).

Former studies showed that changes in volatility are better predicted than changes in close price (Atkins et al. 2018, p. 130). The models might also examine if the opposed correlation between SPX and VIX rates has an impact on the forecasting results.

In total, it's not expected to score a holistic accordance but the hit ratio should be significant enough to be of predictive power.

Nevertheless, the results depend in principle on the accurateness of underlying financial principles.

## 4.2 Observed Results

| | Model 1: AAII_LSTM | Model 2: AAII_VIX_LSTM | Model 3: LM_10X_LSTM | Model 4: LM_10X_VIX_ LSTM |
|---|---|---|---|---|
| MSE | 21394.66 | 7.92 | 41225.07 | 0.28 |
| RMSE | 146.27 | 2.81 | 203.04 | 0.53 |
| MAE | 111.16 | 1.99 | 124.03 | 0.36 |
| MAPE | 5.06 | 11.72 | 5.46 | 2.32 |

**Table 1: Observed Results of Metrics**

As already mentioned the MAPE is used to compare the models. Model 4 scores the best results with MAPE of 2.32% while model 2 peforms worst with 11.72%. Since both models predict the VIX development no statement on whether volatility or stocks are better predicted can be done. However, the difference in performance between these two models is striking while the results of model 1 and 3 are very close with 5.06% and 5.46%. Note that the results vary with each execution but the overall observation remains the same. Thus, the models 3 and 4 which use the LM_10X data seem to have the better overall performance. These findings support the key concern of this paper that financial forecasting based on qualitive textual data adds value to classical forecasting methods solely based on quantitive data as financial ratios. Moreover, these findings indicate that textual information proves to be a long-term indicator. When evaluating these results, one should also keep in mind that the LM_10X file summaries is many times larger. Not only does it include more counts in general, it has also more features. The AAII_Sentiment consists of only three categories while the LM_10X contains eight sentiment counts and three additional textual measurements. Therewith, the common rule of thumb that more data leads to better results would be sustained.

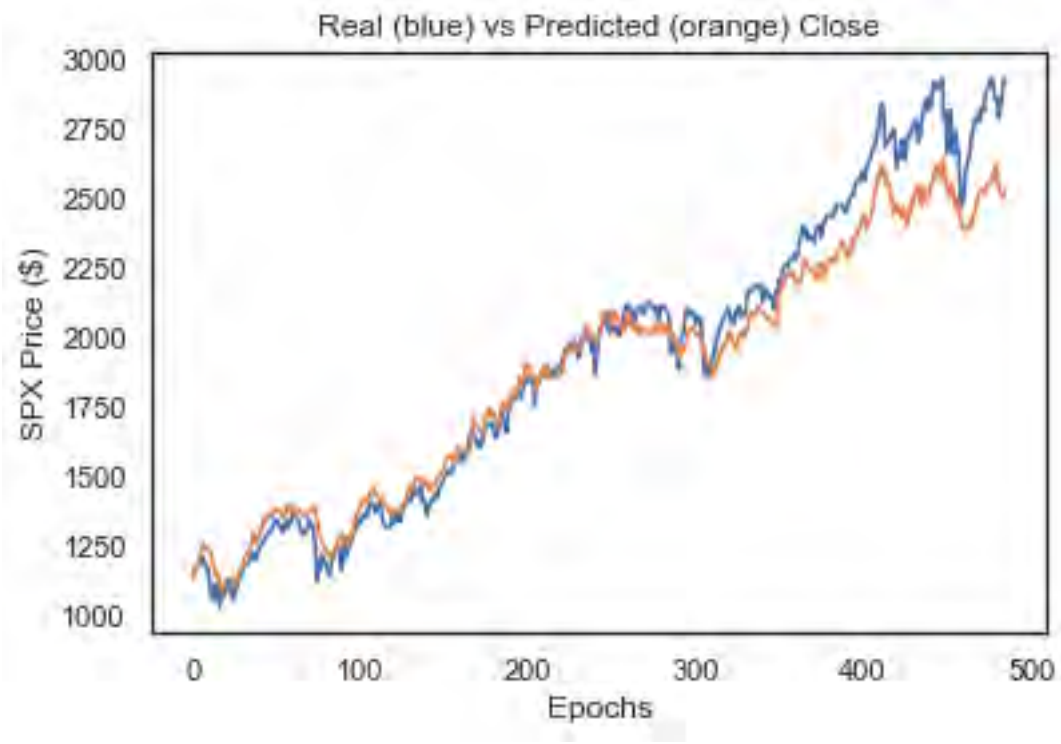


**Figure 25: Model 1 AAII_LSTM Performance Plot**

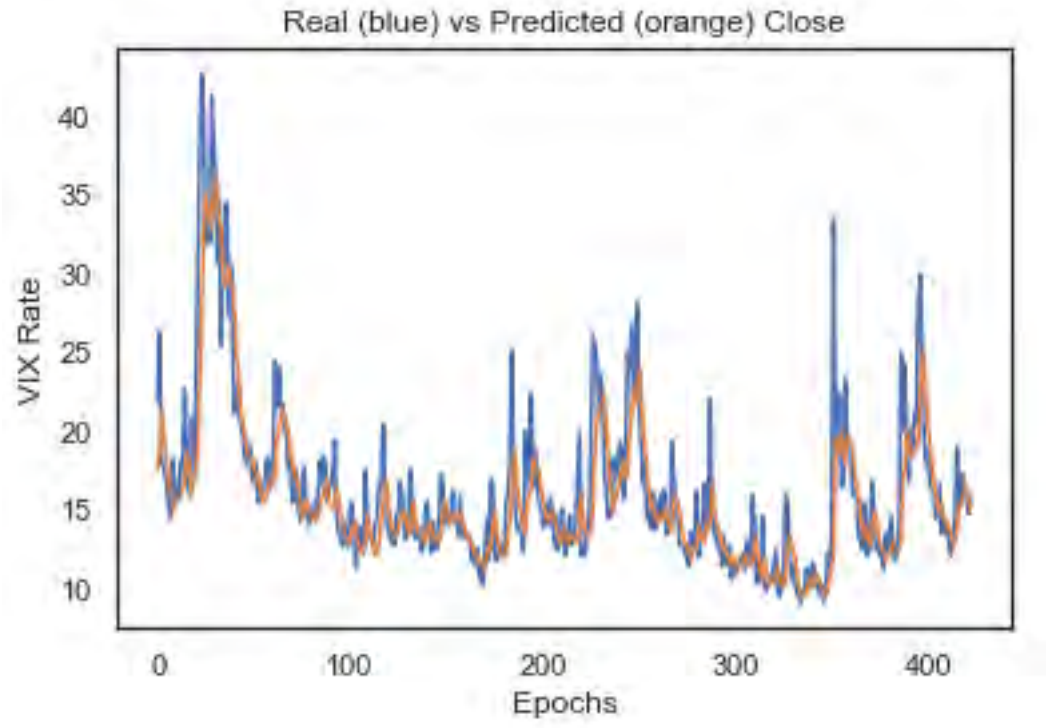


**Figure 26: Model 2 AAII_VIX_LSTM Performance Plot**

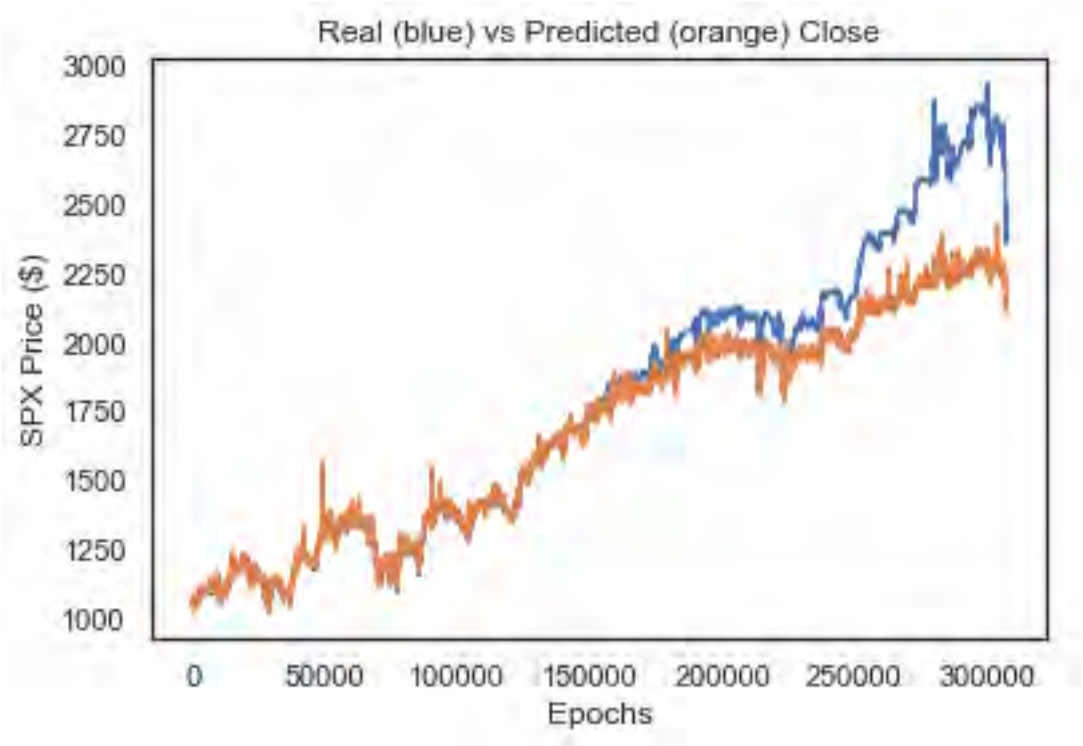


**Figure 27: Model 3 LM_10X_LSTM Performance Plot**

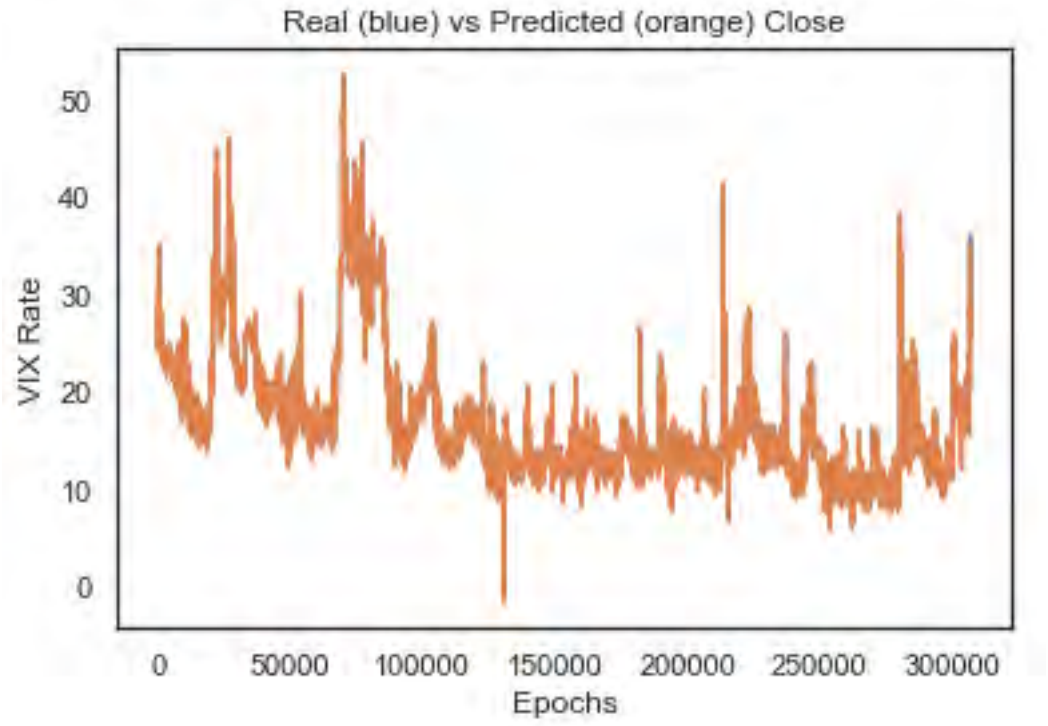


**Figure 28: Model 4 LM_10X_VIX_LSTM Performance Plot**

The graphics of model 2 and 4 don't seem to have increasing deviations between real and predicted close price with increasing epochs. On the opposite, this is in evidence the case with model 1 and 3. Towards the end the predicted prices stay lower then the real ones, which indicates a more pessimistic prediction. Hence, the concern of the non-stationary nature of the SPX is confirmed. Apart from the introduced measures as the shuffling of samples and the CPCV backtesting a higher frequency of retrainings might be another approach. It should also be taken into consideration that MinMaxScaler is problematic in the context of non-stationary distribution. This estimator scales and translates each feature individually such that it is in the given range of the training set[14].

## 4.3 Baseline Model

For reasons of statistical significance, it's recommended to compare the model results with a benchmark.

In order to provide evidence for informative value of the models, two different data sets are used where sentiment is gained from surveys (AAII) and corporate disclosures (Form 10-K/Q).

As mentioned in the ML chapter 2.4, numerous studies already compared different model types. Therefore, the models of this paper solely focus on LSTM which is considered as most promising. Nevertheless, the issue of a suiting baseline model for this context is faced. The Python statsmodel library offers a broad range of classical linear time series forecasting methods. Still, the baseline model must be tuned to this specific problem and its parameters configured and grid searched. In this setting, the vector autoregression moving-average with exogenous regressions (VARMAX) seems to be the best fit[15].

[14] https://scikit-learn.org/stable/modules/generated/sklearn.preprocessing.MinMaxScaler.html (Accessed 11th October 2019)

[15] https://www.statsmodels.org/dev/examples/notebooks/generated/statespace_varmax.html (Accessed 11th October 2019)

# 5 Conclusion and Future Directions

In financial forecasting the nature of data keeps changing and programs need to adapt. In this thesis models to demonstrate the capability of NLFF with ML were developed. The background of theories and techniques was described.

The models used sentiment gained from surveys or textual data and LSTM neural networks to predict SPX price and volatility measured by accuracy. Thereby, the central question and hypothesis of this paper are affirmed. The models showed that it's indeed possible to predict both price of stock index and volatility development with sentiment and other textual information. In addition, it's possible to forecast over short-term weekly range as well as long-term quarterly range.

The figure below provides a summary of the topics concerning NLFF.

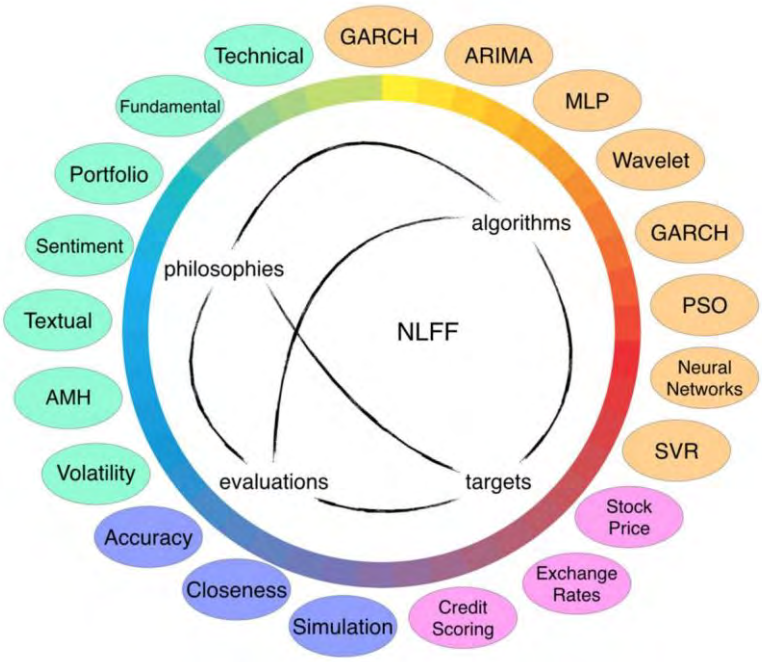


**Figure 29: Topics concerning NLFF (Xing et al. 2018, p. 68)**

## 5.1 Validity of Results

Even with the weak correlations given reasonable models were created. Correlation doesn't mean causation and it's possible that sentiment is then a reaction to the price. This is more likely the case with the AAII sentiment data than the LM_10X file summaries. However, this could happen in patterns that the LSTM could then help make predictions as suggested by the models of this study.

In terms of reproducibility, the models need to be retrained for every new prediction with new data. Past results don't guarantee for future forecasting but the models achieve a manageable uncertainty.

Furthermore, the information of the LM_10X file summaries were applied to the whole SPX. In detail, it might be more accurate to select the summaries of a single company and try to predict the corresponding stock and its volatility.

Accuracy measurements weren't used because a very high directional accuracy might work well in most cases, but at the same time they won't cover frailty to black swan events. The method that suffers a huge loss in a single transaction can illustrate no profability in trading simulation (Xing et al. 2018, p. 66).

## 5.2 Future Directions

With the rise of available data and progress of enhanced ML new analytics from non-traditional sources as the internet become possible. As a result, the impact of this information can be determined. Furthermore, an own expectation for development is possible. The combination with other information allows a deeper and more robust understanding of potential risks and opportunities. As such, the models could serve for application in research and risk management.

In efforts to summarize and compare models the introduced measurements of section 4.1 should be covered. Also, some metrics are not always measuring independent effects and these measures should be unified (Xing et al. 2018, p. 69).

The aim of this thesis is the issue of NLFF and thus didn't research on how to use these forecast. Nevertheless, it would be the next step to develop trading strategies and simulate them. One way to test the real profability is a paper trading system. Paper trading is a test state where the system sells and buys but does not exchange money[16]. This minimum risk test method has the benefit of being able to run in real time, as if it was a real trading system. On this occasion, it would be interesting to implement a reinforcement learning algorithm for the trading system that surrounds the predictive model. Depending on its results it could adjust and potentially improve on an ongoing basis. An increase of resources as computational power would allow more sophisticated model engineering.

With online predictive models more short-term intervals as daily or even hourly could be considered. In doing so, real-time algorithms will modify the key variables stored with the model each time a new batch of data comes in. For this reason, online models have a

[16] https://www.investopedia.com/terms/p/papertrade.asp (Accessed 13th October 2019)

very good adaptability, which is necessary for monitoring fast-changing markets. In addition, the short optimum time window requires a quick response in time as well. While online methods to reduce algorithm complexity for numerical data have been discussed there's a leakage for online NLP techniques (Xing et al. 2018, p. 69).

Since NLFF is a relative young field of studies there's a lot to be done in domain specific resource building. The master dictionary by Loughran and McDonald used to create the 10X file summaries leads in this direction.

# 6 Bibliography

Atkins, Adam; Niranjan, Mahesan; Gerding, Enrico (2018): Financial news predicts stock market volatility better than close price. In *The Journal of Finance and Data Science* 4 (2), pp. 120–137. DOI: 10.1016/j.jfds.2018.02.002.

Das, Sanjiv Ranjan (2013): Text and Context: Language Analytics in Finance. In *FNT in Finance* 8 (3), pp. 145–261. DOI: 10.1561/0500000045.

Feuerriegel, Stefan; Gordon, Julius (2018): Long-term stock index forecasting based on text mining of regulatory disclosures. In *Decision Support Systems* (112), pp. 88–97. DOI: 10.1016/j.dss.2018.06.008.

Guo, Li; Shi, Feng; Tu, Jun (2016): Textual analysis and machine leaning: Crack unstructured data in finance and accounting. In *The Journal of Finance and Data Science* 2 (3), pp. 153–170. DOI: 10.1016/j.jfds.2017.02.001.

Kumar, B. Shravan; Ravi, Vadlamani (2016): A survey of the applications of text mining in financial domain. In *Knowledge-Based Systems* 114, pp. 128–147. DOI: 10.1016/j.knosys.2016.10.003.

Loughran, Tim; McDonald, Bill (2016): Textual Analysis in Accounting and Finance: A Survey. In *Journal of Accounting Research* 54 (4), pp. 1187–1230. DOI: 10.1111/1475-679X.12123.

Marcus, Gary (2018): Deep Learning: A Critical Appraisal. New York University, New York City. Available online at https://arxiv.org/abs/1801.00631.

MathWorks: Applying Supervised Learning.

MathWorks: Introducing Machine Learning.

Nassirtoussi, Arman K.; Aghabozorgi, Saeed; Wah, Teh Y.; Ngo, David C. L. (2014): Text mining for market prediction: A systematic review. In *Expert Systems with Applications* 41 (16), pp. 7653–7670. DOI: 10.1016/j.eswa.2014.06.009.

Prado, Marcos L. de (2018): Advances in Financial Machine Learning. Hoboken, New Jersey: Wiley.

Xing, Frank Z.; Cambria, Erik; Welsch, Roy E. (2018): Natural language based financial forecasting: a survey. In *Artif Intell Rev* 50 (1), pp. 49–73. DOI: 10.1007/s10462-017-9588-9.

# Appendices

## Appendix 1: Source Code

The complete source code can be found at the following url:

https://github.com/thummd/NLFF