STOCK RANKING WITH MARKET MICROSTRUCTURE, NEWS AND TECHNICAL INDICATORS

by

LIANG WANG

(Under the Direction of Khaled Rasheed)

ABSTRACT

Using machine learning techniques to assist financial decision making surged in several areas in the past decade. The availability of high-frequency data enriches the forecasting models with features from market microstructure. Text mining introduces count, tonality and sentiments of financial buzz into machine learned equity price models. In this research, we first conduct a comprehensive survey of the most recent developments of financial text mining techniques. We organize and summarize financial text mining techniques in six aspects: news source selection, text preprocessing, document alignment and labeling, time series preprocessing, forecasting algorithm, and performance evaluation. We list available configuration choices in tables for each design aspect and highlight the performance comparison of different alternatives available in the literature. The survey is finished with a summary of most recent developments in this area and some suggestions on possible future research directions. In Chapter 3, we demonstrate a new stock forecasting perspective that directly learns the stocks' relative performance with a ranking algorithm. We argue that the traditional regress-then-rank approach casts the portfolio selection practice into an unnecessarily hard problem and show that ranking algorithms outperform the neural network regressor significantly in terms of both ranking quality and simulated profit on

out-of-sample testing data. More specifically, with testing data gathered from the ShenZhen stock exchange, LambdaMART scored an NDCG of 82.725 and 0.054% in return per position, while neural network return regressor can only get 10.998 in NDCG and its averaged return per position is -0.237%. With simulated trading under rigorous constraints of transaction costs and order execution price, we demonstrate that the ranker can be used to build highly profitable portfolios.

INDEX WORDS: Data Mining, Ranking, Stock Forecasting, News Analysis, High Frequency Trading

STOCK RANKING WITH MARKET MICROSTRUCTURE, NEWS AND TECHNICAL INDICATORS

by

LIANG WANG

B.S., University of Electronic Science and Technology of China, China, 2008

A Thesis Submitted to the Graduate Faculty of The University of Georgia in Partial Fulfillment of the Requirements for the Degree

MASTER OF SCIENCE

ATHENS, GEORGIA

2018

STOCK RANKING WITH MARKET MICROSTRUCTURE, NEWS AND TECHNICAL INDICATORS

by

LIANG WANG

Major Professor:	Khaled Rasheed
Committee:	Frederick Maier
	Janine E. Aronson

Electronic Version Approved:

Suzanne Barbour
Dean of the Graduate School
The University of Georgia
May 2018

TABLE OF CONTENTS

Page

CHAPTER

CHAPTER 1

INTRODUCTION

Using machine learning techniques to assist financial decision making surged in several areas in the past decade. Text mining introduces count, tonality and sentiments of financial buzz into machine learned equity price models (Nardo, Petracco-Giudici, & Naltsidis, 2016). Deep learning methods introduce extra layers of feature abstraction, such as trend extraction and public attentiveness detection, that mimic the human decision-making process (Hu, Liu, Bian, Liu, & Liu, 2018). The availability of high-frequency data has driven researchers to explore data at finer granularity and examine the dynamic details about price formation (Cont, 2011). These recent developments, together with many previously published works in financial forecasting, typically take a two-step regress-then-rank approach, which builds regression or classification models that predict future returns, and then pick the investment targets from the stocks with relatively higher predicted yield.

In this research, we aim to improve financial investment decision making in Chinese markets with innovative data mining methods. The thesis is composed of two main chapters. In Chapter 2, we perform a comprehensive review of the most recent developments of financial text mining techniques. We organize and summarize text mining techniques for financial forecasting in six aspects: news source selection, text preprocessing, document alignment and labeling, time

series preprocessing, forecasting algorithm, and performance evaluation. We list available configuration choices in tables for each design aspect and highlight the performance comparison of different alternatives available in the literature. At a high level, we observe that the mainstream in financial text mining is using support vector machines to build return or volatility forecasters with text features extracted from financial websites. From recent publications, we also notice the gradual transition from using traditional financial websites to contemporary social media data, and more interests in analyzing the news sentiment or count with deep learning methods.

In Chapter 3, we propose an innovative approach of using ranking algorithms to assist portfolio selection based on news, technical indicators and features extracted from market microstructure. Most traditional financial data mining architectures used the regress-then-rank approach, in which the algorithm is configured to learn the exact price or return of each individual stock, and the ranking is derived from the predicted returns. In contrast, our approach uses LambdaMART to learn the ranking directly with a Normalized Discounted Cumulative Gains (NDCG) augmented binomial log-likelihood cost function. Compared to the traditional price regressors proxied by neural networks, our results show that LambdaMART stock ranker performs better than neural networks in terms of both ranking quality and simulated profits. We assess the ranking quality on out-of-sample testing dataset in terms of NDCG, whose value is in the range [0, 100], the higher the better. With testing data gathered from the ShenZhen stock exchange, LambdaMART scored an NDCG of 82.725 and 0.054% in return per position, while

neural network return regressor can only get 10.998 in NDCG and its averaged return per position is -0.237%. Additionally, since LambdaMART builds an ensemble of regression trees, the built model is interpretable and we can compare the relative importance of features. We discovered that market microstructure features are of most importance to stock ranking, followed by past price, technical indicators and news. By simulating the trading under rigorous constraints of transaction costs and order execution price, we also demonstrate that the ranker can be used to build highly profitable portfolios for real investments.

CHAPTER 2

LITERATURE REVIEW:

SURVEYING STOCK MARKET FORECASTING TECHNIQUES – TEXT MINING METHODS[1]

2.1 Introduction

With the ever-increasing power of computer hardware, more and more complex machine learning techniques have been applied to forecasting the financial markets. The work by (Atsalakis & Valavanis, 2009) surveyed various soft-computing methods that learn the association between numerical features, such as stock price, fundamental variables and technical indicators, with the future price. In this survey, we focus on the literature that relates financial news to stock prices. Financial text mining is a relatively new and burgeoning branch that did not arouse much attention until 1998 when Wüthrich et al. published their seminal paper (Wüthrich, Permunetilleke, et al., 1998). Their system forecasted the change direction of the DJIA index with articles collected from the Wall Street Journal and reported a simulated profit of around 7.5% over a three-month period. This groundbreaking result led to a lot of research in using textual data to predict the stock markets in the past decade. We observe profound developments in this field, including the fast-growing interest in analyzing short messages from social media; the widespread implementation of sentiment analysis; the emerging perspective of using just the

[1] To be submitted to Journal of Forecasting

volume of textual data to forecast the markets; and the expanding variety of learning algorithms that range from traditional ones, such as Support Vector Machines and Naïve Bayesian Classifiers, to contemporary ones like deep neural networks.

Financial text mining learns the relationship between a series of textual data and the future stock prices (Lavrenko et al., 2000a), while traditional stock time series forecasting is mainly concerned with the linkage between a series of numerical quantities (such as past price, fundamental variables, technical indicators) and the future stock prices. This key difference brings about several challenges to the architecture of forecasting systems. For example, the unstructured textual data needs to be represented in a machine-friendly form; the alignment between features (news articles) and labels (price series) is ambiguous and requires optimization or domain knowledge; and it is unintuitive to integrate features from texts and numerical variables, among other things. In the literature, the reported system architectures are drastically different from each other and they used "often incomparable criteria for performance measurements" (Bozic, Chalup, & Seese, 2012). This makes the evolution and the state-of-the-art techniques in this field obscure. For new researchers, it is hard to get an overall picture of the available system architectures in financial text mining.

This research offers a comprehensive review of available papers published in the period from 1998 to 2017, which covers the time from the earliest known research in this area to the present. We examine each system design in six stages, i.e. *news source selection*, *text preprocessing, document alignment and labeling, time series preprocessing*, *forecasting*

methodology and *performance evaluation metrics*. Methods for each stage are summarized in tables to demonstrate the building blocks of textual stock forecasting systems. At a high level, we observed three clusters of methods: frequency-based text mining, sentiment analysis and message volume analysis. We also provide suggestions on possible future search. The main body of this paper is organized according to the six stages of system design, followed by a discussion of recent developments and conclusions.

2.2 News sources

The development of the Internet has made numerous news sources a few clicks away. News sources are dramatically different in quality, including content creditability, level of noise, and timeliness. For example, compared to companies' annual reports, financial buzz on Twitter updates more frequently, and tends to be noisier and less trustworthy. Therefore, choosing the news source is of critical importance to the system performance. The choice should be made jointly with other system design aspects, such as the alignment between features and return labels, and the trading strategy. Table 1 lists the news sources that have been used in the literature.

Table 2.1 List of news sources

News Source	Article
Commercial Software	***PRNewswire***: (Luss & d'Aspremont, 2012; Luss & d'Aspremont, 2009; Mittermayer, 2004; Mittermayer & Knolmayer, 2006) ***Reuters Market 3000 Extra***: (Fung, Yu, & Lam, 2002) ***Thomson Financial Web Service***: (Takahashi, Takahashi, Takahashi, & Tsuda, 2006) ***Bloomberg Professional Service***: (C. Robertson, S. Geva, & R. C. Wolff, 2007; C. S. Robertson, S. Geva, & R. C. Wolff, 2007) ***Others***: (Li et al., 2011)

Companies' Financial Reports/Announcements	(Groth & Muntermann, 2011; Hagenau, Liebmann, & Neumann, 2013; Lee, Lin, Kao, & Chen, 2010; Lin, Lee, Kao, & Chen, 2011; Wang, Huang, & Wang, 2012)
Online Forum	(Thomas & Sycara, 2000; D. D. Wu, Zheng, & Olson, 2014; Y. Zhang, Swanson, & Prombutr, 2012)
News Feeds	(C.-J. Huang, Liao, Yang, Chang, & Luo, 2010; Tang, Yang, & Zhou, 2009)
Manually Collected	(Zhai, Hsu, & Halgamuge, 2007)
Online Corpora and Archives	(Cohen-Charash, Scherbaum, Kammeyer-Mueller, & Staw, 2013; Han, 2012; Lavrenko et al., 2000a, 2000b; Li et al., 2014; Yu, Jan, Debenham, & Simoff, 2006)
Social Media	***Twitter***: (Bollen, Mao, & Zeng, 2011; Bouktif & Awad, 2013; Han, 2012; Makrehchi, Shah, & Liao, 2013; H. Mao, Counts, & Bollen, 2011; Y. Mao, Wei, & Wang, 2013; Mittal & Goel, 2012; Oliveira, Cortez, & Areal, 2013b; Porshnev, Redkin, & Shevchenko, 2013; Rao & Srivastava, 2012a, 2012b; Ruiz, Hristidis, Castillo, Gionis, & Jaimes, 2012; Smailović, Grčar, Lavrač, & Žnidaršič, 2013; Sprenger, Tumasjan, Sandner, & Welpe, 2013; Wolfram, 2011; X. Zhang, Fuehres, & Gloor, 2011) ***LiveJournal***: (Gilbert & Karahalios, 2010) ***StockTwits***: (Oh & Sheng, 2011; Oliveira, Cortez, & Areal, 2013a; F. Xu, 2012) ***WeBlog***: (Kharratzadeh & Coates, 2012) ***Weibo***: (Zhou, Shi, Sun, Qu, & Shi, 2013)
News websites	***Yahoo Finance***: (Dondio, 2013; Li, Deng, Wang, & Dong, 2010; Schumaker & Chen, 2006, 2008, 2009a, 2009b, 2010; Schumaker, Zhang, Huang, & Chen, 2012) ***Reuters***: (Aase, 2011; Hagenau, Hauser, Liebmann, & Neumann, 2013; H. Mao et al., 2011; Rachlin & Last, 2006; Rachlin, Last, Alberg, & Kandel, 2007; Xie, Passonneau, Wu, & Creamer, 2013) ***Forbes***: (H. Mao et al., 2011; Rachlin & Last, 2006; Rachlin et al., 2007) ***Wall Street Journal:*** (Lu, Chen, Chen, Hung, & Li, 2010; H. Mao et al., 2011; Wüthrich, Cho, et al., 1998; Wüthrich, Permunetilleke, et al., 1998) ***CNN-Money:*** (H. Mao et al., 2011) ***CNBC***: (H. Mao et al., 2011) ***Bloomberg***: (H. Mao et al., 2011) ***BusinessWeek***: (H. Mao et al., 2011) ***Financial Times***: (H. Mao et al., 2011) ***Others***: (Dange, Argiddi, & Apte, 2012; Gunduz & Cataltepe, 2013; Junqué de Fortuny, De Smedt, Martens, & Daelemans, 2014; Liang, 2005; Liang & Chen, 2005; Liang, Chen, He, & Chen, 2013; Pinto & Asnani, 2011; Thanh & Meesad, 2014; Vanipriya & Reddy, 2014; Xue, Xiong, Zhu, Wu, & Chen, 2013)

The coverage of research works using news websites is 34%, the highest among other sources. We do not observe superior performance from systems built on news websites than on others in terms of simulated gain. We also observe a rapidly growing interest in exploring the potential of analyzing User Generated Contents (UGC) from social media since 2010. In 2013, Twitter became the most popular source in the literature, probably because of its well-designed APIs as well as its huge user base (Sprenger et al., 2013).

2.3 Text Preprocessing

The motivation of text preprocessing is to represent the raw text in machine-understandable features, such as counts, term frequencies and aggregated sentiment scores that represent the positive or negative feelings expressed in an article. Feature dimensionality reduction is carried out in feature extraction and selection where the goal is to keep enough semantic level of features to portray the meaning of a document accurately (Feldman & Sanger, 2006) while reduce dimensionality for better computational efficiency and forecast performance. Table 2 summarizes the available choices for features, feature extraction, feature selection and document representation from the literature.

Table 2.2 Document Preprocessing Specifications

Articles	**Features of Interest**	**Feature Extraction**	**Feature Selection**	**Document Representation**
(Wüthrich, Permunetilleke, et al., 1998)	Words Phrases	Stemming	Manually selected dictionary	TF-CDF
(Lavrenko et al., 2000a)	Words	N.A.	N.A.	Variations of TF&DF

(Thomas & Sycara, 2000)	Words	Existing package	N.A.	TF
(Gidófalvi & Elkan, 2001)	N.A.	Stemming	Mutual Information Stop words removal	N.A.
(Fung et al., 2002)	N.A.	Existing package[2]	N.A.	Variations of TF&DF
(Mittermayer, 2004)	Words	Stemming-Porter's	TF, DF Stop words removal	Binary
(Mittermayer & Knolmayer, 2006)	Words Phrases	N.A.	Manually selected dictionary TF, DF Chi-square Information Gain Odd's Ratio	TF-IDF
(Schumaker & Chen, 2006)	Words Noun Phrase Name Entities	N.A.	TF, DF Stop words removal	N.A.
(Schumaker & Chen, 2008)	Proper Nouns	Existing package[3]	TF, DF	Binary
(Schumaker & Chen, 2009b)	Words Noun Phrase Name Entities	Existing package[4]	TF, DF	Binary
(Schumaker & Chen, 2009a)	Proper Nouns	Existing package[5]	TF, DF	Binary
(Schumaker & Chen, 2010)	Proper Nouns	N.A.	TF, DF	Binary
(Schumaker et al., 2012)	Proper Nouns	N.A.	TF, DF	Binary Sentiment Score
(Yu et al., 2006)	Words Phrases	Stemming-Porter's	N.A.	TF-IDF
(Tang et al., 2009)	Words	Word Segmentation	TF, DF Manually selected dictionary Stop words removal	TF
(Zhai et al., 2007)	Concepts	N.A.	Stop words removal	TF-IDF
(C. Robertson et al., 2007)	Words	Stemming-Porter's	TF, DF Stop words removal Information Gain BM25/ADBM25	Binary
(Luss & d'Aspremont, 2012)	Words	Stemming	Manually selected dictionary	TF-IDF

[2] IBM Intelligent Miner
[3] Arizona Text Extractor system
[4] Arizona Text Extractor system
[5] Arizona Text Extractor system

(Luss & d'Aspremont, 2009)	Words	Stemming	Manually selected dictionary	TF-IDF
(C.-J. Huang et al., 2010)	Words	Existing package[6] Tokenize	TF, DF	Variations of TF&DF
(Kumar, Kumar, & Prasad, 2012)	Phrases	N.A.	Manually selected dictionary	TF-IDF
(Lin et al., 2011)	Words	Stemming-Porter's Tokenize	Remove Stop Words	TF-IDF
(Oh & Sheng, 2011)	Words	N.A.	N.A.	Sentiment Scores
(Wang et al., 2012)	Words	Tokenize	TF, DF	TF-IDF
(Li et al., 2011)	Words	Tokenize Segmentation	Stop words removal Chi-square	TF-IDF
(Li et al., 2010)	Words	Tokenize Segmentation	Stop words removal	Message Volume TF-IDF
(X. Zhang et al., 2011)	Words	N.A.	Manually selected dictionary	Message Volume
(Lu et al., 2010)	Words Bigrams Trigrams Parts of speech tags	N.A.	N.A.	N.A.
(Gilbert & Karahalios, 2010)	Words	Stemming	TF, DF Information Gain	Sentiment Scores
(Wolfram, 2011)	Words	Tokenize Stemming	TF, DF Stop words removal	TF TF-IDF
(Dange et al., 2012)	Words	N.A.	TF	Variations of TF&DF
(Pinto & Asnani, 2011)	Phrases	Stemming Existing package	Stop words removal	N.A.
(Groth & Muntermann, 2011)	Words	Stemming-Porter's Tokenize	Stop words removal TF, DF Information Gain Chi-square	TF-IDF
(Lee et al., 2010)	Words	Stemming-Porter's Tokenize	Stop words removal	Binary
(Takahashi et al., 2006)	Words	N.A.	Sent	N.A.
(Rachlin et al., 2007)	Words Phrases	Existing package[7]	Stop words removal	TF

[6] Chinese Knowledge and Information Processing(CKIP)
[7] GenEx

(Rachlin & Last, 2006)	Words Phrases	Existing package[8]	Stop words removal	TF
(Han, 2012)	Words	Stemming-Porter's	Stop words removal TF, DF	N.A.
(Liang & Chen, 2005)	Words Phrases	Existing Package	Manually selected dictionary	Variations of TF&DF
(Aase, 2011)	Words Bigrams	Tokenize Stemming	Stop words removal TF, DF	TF-IDF
(Mittal & Goel, 2012)	Words	N.A.	Sentiment Lexicon (POMS)	Sentiment Score
(H. Mao et al., 2011)	N.A.	N.A.	Sentiment Lexicon	Sentiment Scores Message Volume
(Bollen et al., 2011)	N.A.	N.A.	Stop words removal Sentiment Lexicon (POMS)	Sentiment Scores
(Dondio, 2013)	N.A.	N.A.	N.A.	Message Volume
(F. Xu, 2012)	Words Bigrams	Tokenize	N.A.	Binary
(Xue et al., 2013)	Words	Tokenize	Stop words removal TF, DF	Sentiment Score
(Ruiz et al., 2012)	Ticker Symbol Microblog features	N.A.	N.A	Microblog-specific graph representation
(Rao & Srivastava, 2012a)	N.A.	N.A.	N.A.	Sentiment Scores Message volume
(Rao & Srivastava, 2012b)	N.A.	N.A.	Stop words removal	Sentiment Scores Message volume
(Xie et al., 2013)	Semantic Frames Semantic Trees	N.A.	N.A.	Frequency-based semantic frames representation Semantic trees
(Xue et al., 2013)	Words	Tokenize	Sentiment lexicon Stop words removal	Sentiment Scores
(D. D. Wu et al., 2014)	Characters (Chinese) Words	Tokenize	DF POS tags (only keep adjectives)	Binary representation Sentiment Scores
(Zhou et al., 2013)	Words	N.A.	Sentiment lexicon	TF
(Hagenau, Hauser, et al., 2013)	Words	Stemming-Porter's	Tonality	Aggregated and normalized tonality value
(Bouktif & Awad, 2013)	N.A.	N.A.	N.A.	Sentiment scores
(Vanipriya & Reddy, 2014)	Words	N.A.	Sentiment lexicon	Sentiment scores

[8] GenEx

(Makrehchi et al., 2013)	Words Microblog features	N.A.	Sentiment Lexicon Manually selected dictionary	Sentiment Scores
(Thanh & Meesad, 2014)	Words	Tokenize Stemming-Existing package	Stop words removal Linear Support Vector Machine Weight	TF-IDF
(Porshnev et al., 2013)	Words	N.A.	Manually selected dictionary	Sentiment Scores
(Cohen-Charash et al., 2013)	Words	N.A.	Manually selected dictionary	Sentiment scores
(Sprenger et al., 2013)	Words	N.A.	N.A.	Sentiment Scores Message volume
(Oliveira et al., 2013b)	Words	N.A.	Sentiment Lexicon	Sentiment Scores
(Oliveira et al., 2013a)	Words	N.A.	Manually Selected Dictionary	TF
(Li et al., 2014)	Words	Tokenize	Stop words removal	TF-IDF
(Hagenau, Liebmann, et al., 2013)	Words N-Gram Noun-phrases Word combinations (extended 2-Gram with a word distance greater than zero)	Stemming-Porter's	Stop words removal Chi-Square BNS	TF-IDF
(Liang et al., 2013)	Words	Tokenize	Manually Selected Dictionary	TF-IDF
(Gunduz & Cataltepe, 2013)	Words	Stemming(Existing package)	Stop words removal Mutual Information	TF-IDF
(Y. Mao et al., 2013)	N.A.	N.A.	N.A.	Message Volume (volume Spikes)
(Smailović et al., 2013)	Words	Stemming-N-gram Tokenize	Stop words removal	TF-IDF Sentiment Scores
(Junqué de Fortuny et al., 2014)	Words	Stemming	Stop words removal	TF-IDF Sentiment Scores

The features of interest determine the level of abstraction in processing a document (Blostein, Zanibbi, Nagy, & Harrap, 2003). An appropriate level of abstraction is important because, for example, a term “General Electronics” gives more accurate information than the

words “general” and “electronics” in the context of stock market analysis, and “General Electronics quarterly result surpasses street expectation” can be represented as a tuple of <Actor = GE, Action = surpass, Object = expectation> using open information extraction, which may have more direct impact on stock price. Candidate linguistic features can be characters, words, phrases, N-grams, named entities, word vectors and paragraph vectors etc. (Fagan & Gencay, 2009). As shown in Table 2, 68% of the surveyed papers analyzed the document in the scale of words or phrases. Parts of speech tags were employed by (C.-J. Huang et al., 2010; Lu et al., 2010). And two papers (Makrehchi et al., 2013; Ruiz et al., 2012) used microblog-specific features, such as hash-tags and number of retweets, for a clear classification of the related stocks and the impact factor of a tweet. Schumaker and Chen compared the performance of various document representations, including words, noun phrases, proper nouns, named entities and verbs (Schumaker, 2009; Schumaker & Chen, 2009b). They reported the optimal representations under each performance metric, but no general agreement was found across all metrics. Representing the document in words, as their experiments showed, was the worst in most cases.

Feature extraction, as the name suggests, aims to extract linguistic features from the raw data. Depending on the features of interest, the results of feature extraction can be a set of characters, words, terms, etc. For languages written with spaces to delimitate word boundaries, stemming can help reducing words to their stem form. Stemming algorithms such as affix removal, table lookup, successor variety and Porter’s stemming algorithm have been used by various research (Jensen & Shen, 2008). For languages written without word boundaries, an

additional word segmentation step is necessary to tokenize streams of characters into words, phrases, or other meaningful elements (He, Tan, & Tan, 2003).

Feature selection draws the optimal subset from the extracted features. Ideally, an optimal feature subset should be compact in size and retain as much relevant information as possible. Most surveyed papers used either metrics thresholding or dictionary lookup for feature selection. Commonly used quantitative metrics include term frequency, document frequency, information gain, mutual information, chi-square, odds ratio, bi-normal separation (BNS) and term strength (Yang & Pedersen, 1997). Forman conducted a performance comparison of many feature selection metrics in text classification, and recommended BNS as the optimal one (Forman, 2003). Instead of these general-purpose filtering metrics, one can also design their own indicators to distinguish informative words from non-informative ones. Hagenau, Hauser, et al. calculated the tonality for each word based on associated market movement directions, and only kept those words with highest tonality for forecasting (Hagenau, Hauser, et al., 2013). Thanh and Meesad used coefficients in a trained linear SVM model to represent the relative importance of each word on the stock market (Thanh & Meesad, 2014). Besides thresholding on quantitative metrics, 19 papers selected features according to pre-defined or manually established dictionaries. They typically select words or terms that exist in a sentiment lexicon, which can be as simple as two words (bull and bear) (H. Mao et al., 2011) to hundreds of mood words and their derivations (Makrehchi et al., 2013) and complex lexicons in software packages (Bollen et al., 2011).

In the document representation stage, the selected features are translated to numerical forms that can be directly processed by the learning algorithm. Most papers represented features using Term Frequency Inverse Document Frequency (TF-IDF) or other variations of term and document frequency. Another common scheme is to use the binary representation to indicate the existence of each selected feature in a given article. Some recent research implemented aggressive dimensionality reduction and used a single sentiment score to represent a document, which could be aggregated even further.

2.4 Document Alignment and Labeling

Finding the alignment between text and return labels in financial text mining is challenging. The choice of lag from the document timestamp to the time for return calculation demonstrates the systems' perspective in market effectiveness. Efficient market hypothesis suggests that news will be immediately compounded into stock price (Malkiel & Fama, 1970), while other hypotheses such as behavioral finance (Shiller, 2003) and adaptive market hypothesis (A. W. Lo, 2004) suggest that temporary inefficiency may exist. Generally, there is no definitive guideline for the choice of optimal window of influence (Gidófalvi & Elkan, 2001).

Facing this difficulty, previous works either choose the window of influence with their own judgments or treat alignment tuning as an optimization problem and experiment with different time spans. As shown in Table 3, 72% of surveyed papers picked the alignment with past experience or domain knowledge. The selected lags range from zero (efficient market) to

long-term trends[9] in price charts which may last for months after the release of the news. Several papers compared the performance of different alignments. But due to the dramatically different system architectures and performance metrics, no optimal alignment could be concluded. Most research assigns the label according to the change direction in the stock price within the document's window of influence. (Xie et al., 2013) adopted multi-stage labeling with which each document is given two labels, one to indicate whether there is significant price change, the one for the direction of price movement. Manual labeling also exists in the literature, but with much less popularity.

Some recent papers take the sentiment analysis perspective and introduce a layer of sentimental abstraction between textual data and financial price data. These papers typically label each document with sentiments (attitudes or emotions) of the content and analyze the relationship between the aggregated sentiment scores and the stock prices. Some research used machine learning algorithms, such as SVM (D. D. Wu et al., 2014) and Naïve Bayesian Classifier (Rao & Srivastava, 2012a; Sprenger et al., 2013), to identify the overall sentiment of a document. Others used lexicon-based methods which only count the keywords in established sentiment lexicons. Instead of implementing a sentiment extractor from scratch, Bollen et al. and Schumaker et al. adopted existing sentiment extraction software package (Bollen et al., 2011; Schumaker et al., 2012). Some websites allow users to leave comments and give a rating of their

[9] In most cases, the trends are general bullish/bearish periods recognized by piecewise segmentation algorithms. The length of each period varies from days to months or even years.

feelings after reading the article. Y. Zhang et al. used such user-rated sentiments attached with the financial texts (Y. Zhang et al., 2012). Wu, Zheng and Olson compared the performance of SVM and lexicon-based approaches in extracting sentiments for financial forecasting, and found SVM was better with statistical significance (D. D. Wu et al., 2014).

Table 2.3 Document labeling and alignment with price series

Article	Document-Price Alignment	Document Labeling
(Schumaker & Chen, 2006)	Immediate	N.A.
(Schumaker & Chen, 2008)	Immediate	N.A.
(Schumaker & Chen, 2009b)	Immediate	N.A.
(Schumaker & Chen, 2009a)	Immediate	N.A.
(Schumaker & Chen, 2010)	Immediate	N.A.
(Schumaker et al., 2012)	Immediate	Sentiment analysis software (OpinionFinder)
(Li et al., 2014)	[0, 5min] [0, 10min] [0, 15min] [0, 20min] [0, 25min] [0, 30min]	According to stock movement directions
(Wolfram, 2011)	[0, 15min]	According to stock movement directions
(Mittermayer & Knolmayer, 2006)	[0, 15min]	According to stock movement directions
(Groth & Muntermann, 2011)	[0, 15min] [0, 30min]	According to stock movement directions
(Mittermayer, 2004)	[0, 1h]	According to stock movement directions
(Wüthrich, Permunetilleke, et al., 1998)	[0, next closing]	According to stock movement directions
(Thomas & Sycara, 2000)	[0, next closing]	According to stock movement directions
(Yu et al., 2006)	[0, next closing]	According to stock movement directions
(Zhai et al., 2007)	[0, next closing]	According to stock movement directions
(Kumar et al., 2012)	[0, next closing]	According to stock movement directions

(Lin et al., 2011)	[0, next closing]	According to stock movement directions
(Lee et al., 2010)	[0, next closing]	According to stock movement directions
(Aase, 2011)	[0, next closing]	According to stock movement directions Manual Clustering algorithm
(Oh & Sheng, 2011)	[0, next closing]	Sentiment classifier (J48)
(F. Xu, 2012)	[0, next closing]	Manual
(Gilbert & Karahalios, 2010)	[0, next closing]	Sentiment classifier (Decision Tree, Naïve Bayesian Classifier)
(H. Mao et al., 2011)	[0, next closing]	Sentiment score aggregation
(Tang et al., 2009)	[0, next closing]	N.A.
(C.-J. Huang et al., 2010)	[0, next closing]	N.A.
(Pinto & Asnani, 2011)	[0, next closing]	N.A.
(Zhou et al., 2013)	[0, next closing]	According to stock movement directions
(Thanh & Meesad, 2014)	[0, next closing]	According to stock movement directions
(Gunduz & Cataltepe, 2013)	[0, next closing]	According to stock movement directions
(Sprenger et al., 2013)	[0, next closing]	Sentiment classifier(Naïve Bayesian classifier) Sentiment score aggregation
(Oliveira et al., 2013a)	[0, next closing]	Sentiment score aggregation
(Oliveira et al., 2013b)	[0, next closing]	Sentiment score aggregation
(Cohen-Charash et al., 2013)	[0, next opening]	Sentiment score aggregation
(Hagenau, Liebmann, et al., 2013)	[0, next opening] [0, next closing]	According to stock movement directions
(Y. Zhang et al., 2012)	[0, next closing] [0, 2nd closing] [0, 3rd closing]	Labeled from news source
(X. Zhang et al., 2011)	[0, next closing] [0, 2nd closing] [0, 3rd closing]	N.A.
(Smailović et al., 2013)	[0, next closing] [0, 2nd closing] [0, 3rd closing]	Sentiment classifier(SVM)
(Xie et al., 2013)	[0, 2nd closing]	According to stock movement directions
(Vanipriya & Reddy, 2014)	[0, 2nd closing]	Sentiment score aggregation
(Porshnev et al., 2013)	[0, next closing] … [0, 7th closing]	Sentiment score aggregation

(Bouktif & Awad, 2013)	[0, next closing] … [0, 9th closing]	The retrieved data is already labeled
(D. D. Wu et al., 2014)	Rolling window (1-10 days)	Sentiment classifier (SVM) Sentiment aggregation
(Rao & Srivastava, 2012a)	[0, next month's closing]	Sentiment classifier(Naïve Bayesian Classifier)
(Rao & Srivastava, 2012b)	[0, next week's closing]	Sentiment classifier(Naïve Bayesian Classifier)
(Hagenau, Hauser, et al., 2013)	[0, 4 weeks] [0, 6 weeks] [0, 8 weeks] [0, 10 weeks] [0, 12 weeks]	Tonality aggregation
(Lavrenko et al., 2000a)	[0, end of the current trend]	According to stock movement directions
(Fung et al., 2002)	[0, end of the current trend]	According to stock movement directions
(Junqué de Fortuny et al., 2014)	[0, next closing](Pre-specified) [0, 4min] (Optimized)	According to stock movement directions
(C. Robertson et al., 2007)	[0, 5min] (Optimized)	According to stock movement directions
(Luss & d'Aspremont, 2009)	[0, 10min] (Optimized)	According to stock movement directions
(Li et al., 2011)	[0, 20min] (Optimized)	According to stock movement directions
(Gidófalvi & Elkan, 2001)	[0, 20min] (Optimized)	According to stock movement directions
(Ruiz et al., 2012)	[0, next closing] (Optimized)	N.A.
(Mittal & Goel, 2012)	[0, 3rd closing] (Optimized)	Sentiment aggregation
(Bollen et al., 2011)	[0, 3rd closing] (Optimized)	Sentiment analysis software (OpinionFinder) Sentiment aggregation
(Li et al., 2010)	[0, 20 days] (Optimized)	According to stock movement directions
(Luss & d'Aspremont, 2012)	N.A.	According to stock movement directions
(Dange et al., 2012)	N.A.	According to stock movement directions
(Takahashi et al., 2006)	N.A.	According to stock movement directions
(Rachlin et al., 2007)	N.A.	According to stock movement directions

(Rachlin & Last, 2006)	N.A.	According to stock movement directions
(Liang & Chen, 2005)	N.A.	According to stock movement directions
(Lu et al., 2010)	N.A.	Manual
(Xue et al., 2013)	N.A.	Sentiment aggregation
(Makrehchi et al., 2013)	N.A.	Sentiment aggregation

2.5 Stock Time Series Specifications and Preprocessing

In Atsalakis and Valavanis's survey about using soft-computing techniques to predict the market, they divided the forecasting targets into individual stocks and stock market indices, and discovered that most research chose to predict the indices (Atsalakis & Valavanis, 2009). In the text mining papers we reviewed, however, only 19% forecasted stock indices (Bollen et al., 2011; Cohen-Charash et al., 2013; Gunduz & Cataltepe, 2013; Hagenau, Hauser, et al., 2013; C.-J. Huang et al., 2010; Liang et al., 2013; Makrehchi et al., 2013; H. Mao et al., 2011; Mittal & Goel, 2012; Pinto & Asnani, 2011; Rao & Srivastava, 2012a, 2012b; Tang et al., 2009; D. D. Wu et al., 2014; Wüthrich, Permunetilleke, et al., 1998; Xie et al., 2013; X. Zhang et al., 2011). The rest focused on forecasting specific stocks.

Data frequency is also a noteworthy factor in financial system design. The majority (75%) of past research analyzed daily data (open, high, low, close, volume). The rest of the literature chose to process intraday data that was updated every 10 minutes (Gidófalvi & Elkan, 2001; Lavrenko et al., 2000a), 5 minutes (Luss & d'Aspremont, 2009), 1 minute (Groth & Muntermann, 2011; C. Robertson et al., 2007; Schumaker, 2009; Schumaker & Chen, 2006, 2008, 2009a, 2009b, 2010, 2011; Schumaker et al., 2012; Wolfram, 2011), 15 seconds (Mittermayer & Knolmayer, 2006) and 1 second (Aase, 2011; Li et al., 2011; Li et al., 2014;

Mittermayer, 2004). On the contrary, only one paper used market data that updated less frequently than once a day (Wang et al., 2012).

Table 2.4 List of surveyed stock markets

Stock Market	Articles
New York Stock Exchange (NYSE) and/or NASDAQ	(Bollen et al., 2011; Cohen-Charash et al., 2013; Dondio, 2013; Gidófalvi & Elkan, 2001; Gilbert & Karahalios, 2010; Kharratzadeh & Coates, 2012; Lee et al., 2010; Liang, 2005; Lin et al., 2011; Luss & d'Aspremont, 2012; Luss & d'Aspremont, 2009; Makrehchi et al., 2013; H. Mao et al., 2011; Y. Mao et al., 2013; Mittal & Goel, 2012; Mittermayer, 2004; Mittermayer & Knolmayer, 2006; Oh & Sheng, 2011; Oliveira et al., 2013a, 2013b; Pinto & Asnani, 2011; Porshnev et al., 2013; Rachlin & Last, 2006; Rachlin et al., 2007; Rao & Srivastava, 2012a, 2012b; C. Robertson et al., 2007; Ruiz et al., 2012; Schumaker & Chen, 2006, 2008, 2009a, 2009b, 2010; Schumaker et al., 2012; Smailović et al., 2013; Sprenger et al., 2013; Thomas & Sycara, 2000; Wang et al., 2012; Wolfram, 2011; Wüthrich, Permunetilleke, et al., 1998; Xie et al., 2013; F. Xu, 2012; X. Zhang et al., 2011)
American Stock Exchange (AMEX)	(Mittermayer, 2004)
Tokyo Stock Exchange	(Wüthrich, Permunetilleke, et al., 1998)
London Stock Exchange	(C. Robertson et al., 2007; Wüthrich, Permunetilleke, et al., 1998)
Hong Kong Stock Exchange	(Fung et al., 2002; Li et al., 2010; Li et al., 2011; Li et al., 2014; Wüthrich, Permunetilleke, et al., 1998)
Singapore Stock Exchange	(Wüthrich, Permunetilleke, et al., 1998)
Australian Stock Exchange	(C. Robertson et al., 2007; Yu et al., 2006; Zhai et al., 2007)
Shanghai Stock Exchange	(Liang et al., 2013; Tang et al., 2009; Wang et al., 2012; D. D. Wu et al., 2014; Xue et al., 2013; Zhou et al., 2013)
Taiwan Stock Exchange	(C.-J. Huang et al., 2010)
Indian Stock Exchange	(Dange et al., 2012; Vanipriya & Reddy, 2014)
Oslo Stock Exchange (Norwegian)	(Aase, 2011)
Chicago Board Options Exchange Market	(H. Mao et al., 2011)

Frankfurt Stock Exchange (German)	(Hagenau, Hauser, et al., 2013; Hagenau, Liebmann, et al., 2013)
Ho Chi Minh City Stock Exchange	(Makrehchi et al., 2013)
Istanbul Stock Exchange	(Gunduz & Cataltepe, 2013)
Euronext Brussels	(Junqué de Fortuny et al., 2014)

2.6 Forecasting Methodology and Learning Algorithm

The mainstream forecasting methodologies can be classified into three categories: traditional word frequency based text mining, sentiment analysis and message volume analysis (Li et al., 2010). The choice of methodology reveals the researchers' perspective on how textual information influences the financial markets. Traditional text mining makes the prediction based on linguistic features, which reflects the belief that the contents of news articles have a profound impact on the equity markets. On the contrary, message volume analysis overlooks the contents, and attributes price movements to the quantity of news articles. Sentiment analysis stands between the two perspectives and models the stock price on the abstract sentimental polarity of documents. Table 5 summarizes the forecasting methodologies and learning algorithms used in the literature.

2.6.1 Traditional word frequency-based text mining

More than 65%[10] of the surveyed articles adopted this classical approach which builds stock price models on linguistic feature vectors. For example, Wüthrich et al. harvested the articles from the Wall Street Journal and represented each document with the TF-IDF of a

[10] Table 1 presents selected paper. Some surveyed papers are not included in the table due to lack of uniqueness in system architecture. Most of the excluded papers use linguistical text mining approach and SVM.

manually selected pool of keywords (Wüthrich, Permunetilleke, et al., 1998). They used probabilistic rules as their learning algorithm to predict the movement direction of the DJIA index. The forecasting accuracy of their system was moderate (43.6% accurate in predicting positive, negative or neutral) compared with later research, but the simulated profit reached a groundbreaking 7.5% return rate over a three-month period. Despite the fact that they ignored the transaction costs in simulated trading, it was a significant first step in utilizing textual information in financial forecasting.

The most popular learning algorithm in text mining systems is the Support Vector Machine (SVM). Naïve Bayesian ranks the second, followed by k-nearest neighbor. Among all papers that implemented SVM, 61% of them used it as their only learning algorithm. This is probably because of its capability in handling very high dimensional data (Joachims, 1998). But interestingly, among the 9 papers that compared SVM with other learning algorithms, only two papers reported that SVM was superior to its comparing algorithms: k-nearest neighbor (Mittermayer & Knolmayer, 2006) and a hybrid of neural networks and naïve Bayesian classifier (Hagenau, Liebmann, et al., 2013). In the other 7 papers, SVM did not perform as well as decision tree (C. Robertson et al., 2007; F. Xu, 2012), k-nearest neighbor (Groth & Muntermann, 2011), maximum entropy classifier (Y. Zhang et al., 2012), Self-organizing Fuzzy Neural networks (Mittal & Goel, 2012), extreme learning machine (Li et al., 2014) and multiple data domain description (Xue et al., 2013).

Kernel selection is an important step in SVM configuration. In order to help reduce the number of computations required for training the SVM, many kernel functions have been designed to simplify the calculation of dot production between high-dimensional feature vectors (Boser, Guyon, & Vapnik, 1992). The most commonly used kernels are the following:

- Linear kernel: $K(\boldsymbol{x}, \boldsymbol{y}) = \boldsymbol{x}^T \boldsymbol{y}$
- Polynomial kernel: $K(\boldsymbol{x}, \boldsymbol{y}) = (\gamma \boldsymbol{x}^T \boldsymbol{y} + R)^{\boldsymbol{d}}$, $\gamma > 0$
- Gaussian Radial Basis Function (RBF) kernel: $K(\boldsymbol{x}, \boldsymbol{y}) = \exp(-\gamma \|\boldsymbol{x} - \boldsymbol{y}\|^2), \gamma > 0$
- Sigmoid kernel: $K(\boldsymbol{x}, \boldsymbol{y}) = \tanh(\alpha \boldsymbol{x}^T \boldsymbol{y} + c)$

In the financial forecasting literature, most systems adopt the linear kernel for its simplicity and good performance (Groth & Muntermann, 2011; Junqué de Fortuny et al., 2014; Mittermayer & Knolmayer, 2006; Rao & Srivastava, 2012b; Schumaker & Chen, 2006, 2008; Schumaker et al., 2012; Wolfram, 2011; Xie et al., 2013; Xue et al., 2013; Y. Zhang et al., 2012). Other kernels were also implemented but with less popularity, such as Gaussian (Mittermayer & Knolmayer, 2006; Wang et al., 2012; Zhai et al., 2007), sigmoid (Mittermayer & Knolmayer, 2006; Zhou et al., 2013) and polynomial (Mittermayer & Knolmayer, 2006; Zhai et al., 2007). The dominance of the linear kernel is in agreement with non-financial research of SVM: Hsu et al. suggested using a linear kernel rather than other kernels in situations where feature dimensionality is high (Hsu, Chang, & Lin, 2003). But contrary opinion also exists in the forecasting literature. Thanh and Meesad suggested that radial basis kernel “gives higher

accuracy and efficient processing time" compared to linear and polynomial kernels (Thanh & Meesad, 2014).

Some research used both textual data and numerical price time series for better forecasting performance. Support Vector Regression can take both numerically represented text and price data as features to forecast the exact stock price/index, and it was implemented by (Liang et al., 2013; Schumaker & Chen, 2006, 2008, 2009b; Schumaker et al., 2012; Tang et al., 2009). Schumaker et al. compared the performance of combining text and price features with using either one alone, and they were in support for this feature diversity. Multiple Kernel Learning (MKL) is also capable of processing both text and price data series by implementing dedicated kernels for each input series. MKL was used to forecast the price change direction of stocks in the NYSE with an accuracy of 71% (Luss & d'Aspremont, 2012; Luss & d'Aspremont, 2009). Li et al. compared MKL with other algorithms for processing both numerical and textual data, and showed that MKL was superior (Li et al., 2011).

2.6.2 Sentiment Analysis

Sentiment analysis is an individual field of study that analyzes people's attitudes, opinions and emotions from written language (Liu, 2012). As applied to financial forecasting, these systems generally represent an article or a collection of articles with a single sentiment score (optimistic, worried, etc.), and then analyze the correlation between the sentiment score and stock prices. Forecasting systems built on sentiment scores typically adopt news sources with large message volume, such as social media, probably to compensate for the loss of

information due to the aggressive dimensionality reduction. The most cited paper in this category is probably from Bollen et al. who forecasted both the exact index and movement directions of the DJIA index by analyzing twitter postings (Bollen et al., 2011). They used existing sentiment extraction packages to bucketize tweets into one of six moods (Calm, Alert, Sure, Vital, Kind, Happy) and then aggregate the count of tweets in each bucket to form six feature time series. Using self-organizing fuzzy neural networks, they built a forecasting model on these features and got an accuracy of 86.70% in predicting the movement direction and a MAPE of 1.79% in predicting the actual index. Some sentiment analysis forecasting systems use sentiment indices to reflect the overall sentiment of all documents retrieved within a time period (Gilbert & Karahalios, 2010; H. Mao et al., 2011; Oh & Sheng, 2011; Rao & Srivastava, 2012a, 2012b). The commonly used sentiment indices are the Bullishness Index and Agreement introduced by (Antweiler & Frank, 2004).

As shown in Table 2.5, many sentiment analysis forecasting systems made use of regression and correlation analysis to analyze the relationship between sentiment scores/indices and the stock price/return. Some applied granger causality analysis to further demonstrate the cause-and-effect relationships. Others used various machine learning algorithms, such as SVM, naïve Bayesian classifier, decision trees and fuzzy neural networks.

2.6.3 Message Volume Analysis

Message volume analysis builds stock price models purely on the number of news articles. As far as we know, this method was first applied by Thomas and Sycara who used the Genetic

Algorithm to learn trading rules based on message volume and trading volume (Thomas, 2003). The messages were collected from online discussion boards and the model was trained to maximize the excess return[11] on selected stocks in NASDAQ or NYSE. After integrating the trading rules with a maximum entropy learner, they reported an excess return of 19.26% over 200 consecutive trading days. Liu et al. forecasted the stock price change direction by identifying the days with a burst of news[12] (Li et al., 2010). In particular, they learned the association between trend reversal and abnormal message volume. The system's prediction accuracy was reported to be superior compared to word frequency-based text mining using SVM. Similar to sentiment analysis, most message volume analysis systems made use of regression models and correlation analysis. Some implemented SVM and decision trees.

2.7 Performance Measures

The forecasting performance has been reported in three ways in the literature: statistics of predicted stock price and index, accuracy of predicted change direction and simulated profit. The statistic measures about the predicted stock price and index include mean squared error (MSE), mean absolute percentage error (MAPE), root mean squared error (RMSE), the coefficient of determination (R^2) and correlation coefficient. About 80% of the articles measured their systems' performance in terms of accuracy, precision, recall and F-1 score in predicting the price change direction. For simulated profit, some papers used the Sharpe Ratio to demonstrate the system's

[11] Excess return as compared to buy and hold strategy.
[12] Defined as message volume $> \mu + 2\sigma$, where μ is the average of message volume in the past, and σ is the variance

profitability relative to its risk exposure (Luss & d'Aspremont, 2012). Others directly reported the simulated profit or return.

Table 2-5 Performance measures

Article	Statistics about Predicted Stock Price and Index	Accuracy in Direction Forecasting	Simulated Profit
(Wüthrich, Permunetilleke, et al., 1998)	N.A.	Accuracy	Return
(Lavrenko et al., 2000a)	N.A.	Recall, Precision	Return
(Thomas & Sycara, 2000)	N.A.	N.A.	Excess Return (compared to B&H)
(Gidófalvi & Elkan, 2001)	R^2	Accuracy, Precision, Recall	N.A.
(Fung et al., 2002)	R^2	N.A.	Return
(Mittermayer, 2004)	N.A.	Recall	Return
(Mittermayer & Knolmayer, 2006)	N.A.	Accuracy, F-1	Return
(Schumaker & Chen, 2006)	MSE	Accuracy	Return
(Schumaker & Chen, 2008)	N.A.	N.A.	Return
(Schumaker & Chen, 2009b)	MSE	Accuracy	Return
(Schumaker & Chen, 2009a)	MSE	Accuracy	Return
(Schumaker & Chen, 2010)	N.A.	N.A.	Return
(Schumaker et al., 2012)	MSE	Accuracy	Return
(Yu et al., 2006)	N.A.	Accuracy	N.A.
(Tang et al., 2009)	MAPE	Accuracy	N.A.
(Zhai et al., 2007)	N.A.	Accuracy	Profit
(C. Robertson et al., 2007)	N.A.	Accuracy, Recall	N.A.
(Luss & d'Aspremont, 2012)	N.A.	Accuracy	Sharpe Ratio
(Luss & d'Aspremont, 2009)	N.A.	Accuracy, Recall	Sharpe Ratio
(C.-J. Huang et al., 2010)	N.A.	Precision, Recall	N.A.
(Kumar et al., 2012)	N.A.	Accuracy, Precision, Recall	N.A.
(Lin et al., 2011)	N.A.	Accuracy	Return
(Oh & Sheng, 2011)	N.A.	Precision, Recall, F-1	N.A.
(Wang et al., 2012)	MAE, MAPE, RMSE	N.A.	Return
(Li et al., 2011)	N.A.	Accuracy	N.A.
(Li et al., 2010)	N.A.	Precision, Recall, F-1	N.A.
(X. Zhang et al., 2011)	CC	N.A.	N.A.
(Lu et al., 2010)	N.A.	Accuracy, Recall, Precision, F-1	N.A.

(Gilbert & Karahalios, 2010)	CC, Standard Deviation, t-value, p-value	N.A.	N.A.
(Wolfram, 2011)	MSE	N.A.	N.A.
(Dange et al., 2012)	N.A.	Accuracy	N.A.
(Pinto & Asnani, 2011)	N.A.	Accuracy	N.A.
(Groth & Muntermann, 2011)	N.A.	Accuracy, Precision, Recall, F-1	Return
(Y. Zhang et al., 2012)	R^2	Accuracy	Return
(Lee et al., 2010)	N.A.	Accuracy	Return
(Rachlin et al., 2007)	N.A.	Accuracy	Profit
(Han, 2012)	N.A.	Accuracy	N.A.
(Liang, 2005)	N.A.	Accuracy	N.A.
(Aase, 2011)	N.A.	Accuracy, Precision, Recall, F-1	Return
(Mittal & Goel, 2012)	MAPE	Accuracy	Profit
(H. Mao et al., 2011)	MAPE, CC, p-value	Accuracy	N.A.
(Bollen et al., 2011)	MAPE, CC, p-value, t-value	Accuracy	N.A.
(Dondio, 2013)	N.A.	Precision, Accuracy	N.A.
(F. Xu, 2012)	CC	Accuracy, Precision, Recall, F-1	N.A.
(Xue et al., 2013)	N.A.	Accuracy	N.A.
(Ruiz et al., 2012)	CC	N.A.	Return
(Rao & Srivastava, 2012a)	R^2, MaxAPE	Accuracy	N.A.
(Rao & Srivastava, 2012b)	R^2, MaxAPE	Accuracy	N.A.
(Xie et al., 2013)	N.A.	Matthews correlation coefficient (MCC)	N.A.
(D. D. Wu et al., 2014)	N.A.	Accuracy	N.A.
(Zhou et al., 2013)	N.A.	Accuracy	N.A.
(Hagenau, Hauser, et al., 2013)	N.A.	Accuracy	Return
(Bouktif & Awad, 2013)	N.A.	J-Index Precision Recall	N.A.
(Vanipriya & Reddy, 2014)	N.A.	Accuracy	N.A.
(Makrehchi et al., 2013)	N.A.	Precision Recall F-measure	N.A.
(Thanh & Meesad, 2014)	N.A.	Accuracy	N.A.
(Porshnev et al., 2013)	N.A.	Accuracy	N.A.
(Oliveira et al., 2013a)	RMSE MAPE	N.A.	N.A.
(Li et al., 2014)	N.A.	Accuracy	N.A.

(Hagenau, Liebmann, et al., 2013)	N.A.	Accuracy	Return
(Liang et al., 2013)	MAPE	N.A.	N.A.
(Gunduz & Cataltepe, 2013)	N.A.	Accuracy Precision Recall F-measure	N.A.
(Junqué de Fortuny et al., 2014)	N.A.	Accuracy AUC	Return Sharpe Ratio

2.8 Recent developments

So far, we have discussed three major methodologies: frequency-based text mining, sentiment analysis and message volume analysis. With developments in the past several years, some extensions to the existing system structures have attracted significant interests.

First, recent researchers have introduced some semantic-level features to the financial forecasting literature. Zhai et al. (Zhai et al., 2007) predicted the direction of an individual stock in the Australian Stock Exchange by extracting and concepts[13] which are semantic abstractions of linguistic features (Feldman & Sanger, 2006). Xie, Passonneau et al. compared the forecasting performance of four document representation schemes, i.e. Bag-of-Words, supervised latent Dirichlet allocation (sLDA), semantic tree, and semantic frame based features. They showed that representing documents in semantic tree and semantic frame based features give better forecasting accuracy (Xie et al., 2013). Xue, Xiong et al. applied an online LDA model to extract topics in article collections, and aggregated the sentiments of articles by extracted topics (Xue et al., 2013). Compared with word frequency and text sentiment, semantic-level features give the learning algorithm more information with regard to the meaning of the text, while it may also

[13] For example, Nokia and Motorola can be mapped to an abstract concept "Mobile Industry".

impose a heavier load of computation. However, the growth in computer hardware may mitigate this drawback, and make semantic-level analysis a promising research direction in financial forecasting.

Second, with increased financial text data availability, message volume analysis on social media and search engine data gains more popularity in works published recently. Mao, Wei et al. used anomaly detection techniques to recognize spikes in news volume and studied the association between such news volume anomaly and excess return (Y. Mao et al., 2013). Some search engines, such as Google, provide search volume data that shows the trend of public interests on user-specified keywords[14]. H.S. Moat et al. suggested that the Google search volume and the number of views and edits of companies' profile page on Wikipedia can assist investment decision making (Moat, Curme, Stanley, & Preis, 2014). H. Mao et al. compared the predictive power of Google search volume and twitter volume and concluded that, while "Google search volume is indeed predictive of financial indicators", Twitter is potentially more efficient in forecasting (H. Mao et al., 2011). Some social media rank the search terms or hashtags by their aggregated search or posting volumes. Zhou et al. used the "heat" of financial keywords on social media to predict the market movement direction and achieved an accuracy of 78.38% (Zhou et al., 2013).

[14] Open http://www.google.com/trends/, and search for terms like "DJIA". It will show a plot of interest over time respect to the term you searched.

Third, in contrast to the traditional way of attributing price changes to each document, recent publications mostly adopt the time-series model, in which the features are aggregated sentiments, message volumes or term frequencies of selected keywords across multiple documents, and return labels are aligned with these features at the lag of choice. N. Oliveira et al. built learning models on the sentiment and attention indicators from tweeter to predict the return of S&P 500 index, and they showed superior performance than models built without such features with statistical significance (Oliveira, Cortez, & Areal, 2017). Such model configuration also enables a wider variety of features to be processed together with text features. J.-L. Wu et al. combined technical analysis with sentiment analysis for predicting stock prices (J.-L. Wu, Su, Yu, & Chang, 2013). Fortuny and Smedt et al. discovered that the hybrid of term frequencies, sentiment features and technical indicators does not necessarily lead to superior performance (Junqué de Fortuny et al., 2014).

Finally, with the surging interests in deep learning, research in the past three years started to appreciate the extra level of abstraction brought by the "deep" hidden layers in neural networks. (Hu, Liu, Bian, Liu, & Liu, 2017) used extra neural network layers to detect the level of attention among audience based on trustworthiness and informativeness of news. (Ding, Zhang, Liu, & Duan, 2015) used convolutional neural networks to capture the consecutive occurrence of events (such as lawsuits, quarter releases) extracted from texts. (Akita, Yoshihara, Matsubara, & Uehara, 2016) implemented recurrent neural network in the Long Short Term Memory(LSTM) architecture to predict the closing stock price with news articles represented in paragraph vectors.

Similarly, (Xiong, Nichols, & Shen, 2015) used a LSTM network to predict the volatility of S&P 500 index with google search volume on a list of 25 selected keywords. All surveyed papers using deep learning methods exhibited either high accuracy or positive simulated profit on testing dataset.

2.9 Conclusions and Suggestions for Future Research

In this paper, we organize and summarize text mining techniques for financial forecasting in six aspects: news source selection, text preprocessing, document alignment and labeling, time series preprocessing, forecasting algorithm, and performance evaluation. We list available configuration choices in tables for each design aspect and highlight the performance comparison of different alternatives available in the literature. We hope this survey could provide an informative recap of the literature. In retrospect, we observe the following trends in the mainstream of financial text mining:

1. With improved availability of textual data from the web, we see a shifting interest from using texts collected from financial websites to using user generated contents on social media in the past 5 years.
2. For text representation, accumulating evidence suggests that traditional term-frequency based features aggregated on words and phrases are inferior to features extract from the semantic level. There are a variety of semantic level features. For example, (Ding et al., 2015) extracted “events” that describes impactful activities of companies, and (Akita et al., 2016) used paragraph vectors to cast words into numerical vectors which also encodes information about

the surrounding context. Another popular approach is to represent text with sentiment scores or just message volume count. The idea is to average the score calculated from all documents within a timespan, so as to cancel the noise in the contents of or traders' interpretations to each document.

3. Text features are typically aligned with labels calculated from the daily closing price of the related stock. It is noteworthy that even with increased data volume and frequency from the social media, most research still forecasts stock price movement in the scale of days.
4. The majority of papers investigate constituent stocks of developed markets, such as component stocks in the S&P 500, NASDAQ, etc. Among all surveyed articles, 75% used daily price data, while others mostly used intraday transaction data.
5. SVM is the dominating learning algorithm in financial text mining, yet 7 out of 9 papers that compared SVM with other algorithms suggested inferior performance from SVM. With the recent developments in deep learning and hardware computing power, most published works in the recent 3 years adopt deep recurrent neural networks or convolutional networks. These network structure can be interpreted in ways that resemble human decision-making process. Going forward, we expect to see more research exploring the profitability of using deep learning methods to process sentiment features.
6. For performance evaluations, most papers reported the accuracy of predicted price change direction. Simulated profits and stock price predictions have also been observed.

Most of the reviewed papers confirmed the profitability of financial text mining. However, there are still lots of unexplored areas in financial forecasting with textual analysis. Human traders do not only consider news for decision making. A variety of factors like technical analysis, fundamental quality of a company and market microstructure play important roles in traders' decisions as well. A promising future research direction is to analyze the relative importance of these factors and develop an agile model that adapts to market conditions. The research would target on learning the market sensitivity to news given the occurrence of special events or price patterns. For example, stocks that exhibits reversal patterns in technical analysis would be more sensitive to negative news. A good starting point would be building a ranker that operates on high-frequency data to either rank the stocks or rank the features groups to be used for forecasting.

Table 2-5 Forecasting methodology and core learning algorithms: column titles are abbreviated for space.

Articles	TM	SA	MVA	SVM	NB	DT	k-NN	NN	CA	GCA	ME	RA	WAR	SOFNN	Others
(Fung et al., 2002)	•			•											
(Mittermayer, 2004)	•			•											
(Mittermayer & Knolmayer, 2006)	•			•			•								Rocchio Algorithm
(Schumaker & Chen, 2006)	•			•											
(Schumaker & Chen, 2008)	•			•											
(Schumaker & Chen, 2009b)	•			•											
(Schumaker & Chen, 2009a)	•			•											
(Schumaker & Chen, 2010)	•			•											
(Schumaker et al., 2012)	•			•											
(Yu et al., 2006)	•			•											
(Tang et al., 2009)	•			•											
(Zhai et al., 2007)	•			•											
(C. Robertson et al., 2007)	•			•		•									
(Luss & d'Aspremont, 2012)	•			•											
(Luss & d'Aspremont, 2009)	•			•											
(Kumar et al., 2012)	•			•											
(Wang et al., 2012)	•			•											
(Li et al., 2011)	•			•											
(Thanh & Meesad, 2014)	•			•											
(Hagenau, Liebmann, et al., 2013)	•			•											
(Zhou et al., 2013)	•			•					•	•					
(Lu et al., 2010)	•			•								•			
(Wolfram, 2011)	•			•											
(Mittal & Goel, 2012)	•			•								•		•	
(Groth & Muntermann, 2011)	•			•		•	•	•							

(Y. Zhang et al., 2012)	•		•	•		•		•		Expectation Maximization; Kullback-Leibler Divergence; Probabilistic Indexing Model
(Lavrenko et al., 2000a)	•			•						
(Gidófalvi & Elkan, 2001)	•			•						
(Aase, 2011)	•			•						
(Gunduz & Cataltepe, 2013)	•			•						
(Dange et al., 2012)	•				•					
(Rachlin et al., 2007)	•				•					
(Rachlin & Last, 2006)	•				•					
(Pinto & Asnani, 2011)	•						•			
(Liang & Chen, 2005)	•						•			
(C.-J. Huang et al., 2010)	•								•	
(Wüthrich, Permunetilleke, et al., 1998)	•									Probabilistic Rules
(Lin et al., 2011)	•									Hierarchical Agglomerative Clustering and K-means Clustering
(Lee et al., 2010)	•									Hierarchical Agglomerative Clustering and K-means Clustering
(Li et al., 2014)	•									Extreme Learning Machine
(Thomas & Sycara, 2000)	•	•		•				•		Genetic Algorithm

(F. Xu, 2012)	•	•		•	•	•		•	•			
(Junqué de Fortuny et al., 2014)	•	•		•								
(Bouktif & Awad, 2013)		•			•							Ant Colony Optimization
(Xie et al., 2013)		•		•								
(Liang et al., 2013)		•		•								
(Xue et al., 2013)		•		•								Multiple Data Domain Description
(Oh & Sheng, 2011)		•		•		•						Cost Sensitive Classification; ZeroR
(Porshnev et al., 2013)		•		•			•					
(Vanipriya & Reddy, 2014)		•					•					
(Makrehchi et al., 2013)		•										Rocchio Classifier
(Gilbert & Karahalios, 2010)		•							•			
(Smailović et al., 2013)		•							•			
(H. Mao et al., 2011)		•						•	•			
(Bollen et al., 2011)		•							•		•	
(Rao & Srivastava, 2012a)		•						•	•	•		Exponential Smoothing; ARIMA; Seasonal ARIMA
(Rao & Srivastava, 2012b)		•		•					•	•		Exponential Smoothing; ARIMA; Seasonal ARIMA
(Cohen-Charash et al., 2013)		•										ARIMA
(Sprenger et al., 2013)		•	•					•				

(Oliveira et al., 2013b)	•	•					•
(Oliveira et al., 2013a)	•	•					•
(D. D. Wu et al., 2014)	•	•	•			•	
(Li et al., 2010)		•					
(X. Zhang et al., 2011)		•				•	
(Liang, 2005)		•			•		
(Dondio, 2013)		•		•			

CHAPTER 3

STOCK RANKING WITH MARKET MICROSTRUCTURE, TECHNICAL INDICATOR AND NEWS[15]

3.1 Introduction

Using machine learning techniques to assist financial decision making surged in several areas in the past decade. Text mining introduces count, tonality and sentiments of financial buzz into machine learned equity price models (Nardo et al., 2016). Deep learning methods introduce extra layers of feature abstraction, such as trend extraction and public attentiveness detection, that mimics human decision-making process (Hu et al., 2018). The availability of high-frequency data has driven researchers to explore data at finer granularity and examine the dynamic details about price formation (Cont, 2011). These recent developments, together with many previously published works in financial forecasting, typically take a two-step regress-then-rank approach, which builds regression or classification models that predicts future returns, and then make investment suggestions from the stocks with higher predicted yield.

In this research, we take a new perspective that directly learns the stocks' relative performance with a ranking algorithm. We argue that the traditional regress-then-rank approach casts the portfolio selection practice into an unnecessarily hard problem in the sense that traders typically pick their stocks without forming an accurate prediction of the target prices. LambdaRank is a group of ranking algorithms that have been proven successful in solving ranking problems on big data from the web (Christopher JC Burges, 2010). It uses gradient

[15] Submitted to World Congress in Computer Science, 2018. Currently under review.

descent learners, such as backpropagation neural networks (LeCun, Bottou, Orr, & Müller, 1998) and MART (Friedman, 2001), to model the labels that represent the relative order on a given set of instances. Since labels like this are inconsistent by nature[16], LambdaRank is designed to learn the probability that an instance should be ranked higher than another. A probabilistic model is built to optimize for an augmented cross entropy cost function that gives higher penalty to ranking mistakes on top performers. This skewed emphasis also fits naturally into the portfolio selection task, in the sense that traders care more about the accuracy on the predicted top performing stocks. We compare our approach with the traditional regress-then-rank approach by building a ranker and a neural network regressor on the same features. The result suggests that the ranker outperforms the neural network regressor significantly both in terms of ranking quality and simulated profit on out-of-sample testing data, and that the ranker can be used to build highly profitable portfolios after deduction of transaction costs.

Feature design is of great importance to the performance of a forecasting system. Ideally, the learning algorithm needs to get all factors that have impact on stock prices as features. Traditional traders make investment decisions based on past stock prices, fundamental variables (Graham & Dodd, 1934), technical rules (Murphy, 1999) and news. But since the invention of algorithmic traders, there is an increasing diversity in the factors that influence market participants' decision making. An automated trader is capable of extracting features from high-frequency order flow and transaction data and derive trading strategies from it (Aldridge, 2013). In order to empower our learners with features that may influence both human traders and algorithmic traders, we provide a wide spectrum of features to the learners, including past stock prices, technical indicators and rules, news and features to describe order flow and order book

[16] The ranking for a stock may float up or down due to changes in other stocks' performance, while its own features remain constant. Thus, the ranker may receive different labels for the same set of features.

dynamics. Since LambdaMART uses decision trees as base learners, we have the benefit to compare the relative importance of each feature based on its cumulative information gain from training instances. Our results suggest that features extracted from market microstructure are the most impactful features to our ranker, followed by current price, technical features and news.

3.2 Background and Related Works

Stock prediction has been a heated topic for several decades in the literature of finance and computer science. On the finance side, since Fama's papers that formally defined the efficient market hypothesis which is against market predictability (Fama, 1965, 1970), empirical evidence (De Bondt & Thaler, 1985) and new theories like behavioral finance (Shiller, 2003) and adaptive market hypothesis (A. Lo, 2004) were published to support at least temporary inefficiency and predictability.

Using computer algorithms to forecast stock price, return, risk and their composites emerged vaguely after 1990. The mainstream of this topic can be clustered by the underlying trading practices, including fundamental analysis, technical analysis, trading with news and high frequency trading.

Fundamental analysts believe in the intrinsic value of equities and make relatively long-term forecasts based on metrics reported in financial statements and macroeconomic variables. Previous research used decision trees (M.-C. Wu, Lin, & Lin, 2006), neural networks (Lam, 2004; Quah & Srinivasan, 1999) and kernel methods (C.-F. Huang, 2012; Ince & Trafalis, 2007) for predictions with fundamental variables.

Practitioners use rules built on top of technical indicators to assist their trading. Various indicators had been developed based on statistics of the past stock prices (Murphy, 1999). Most technical rules can be generalized as crossovers among the indicators, static thresholds and first

or second order derivate of indicators and stock prices. Previous research used various machine learning techniques such as neural networks (Kaastra & Boyd, 1996), genetic programming (Neely, Weller, & Dittmar, 1997) and fuzzy logic (Chang & Liu, 2008) to predict the market with strong evidence of excess returns on out-of-sample test. Another branch of technical analysis is charting (a.k.a. pattern study). Many studies were conducted to examine the profitability of price patterns, such as head-and-shoulder (Osler & Chang, 1995), double-bottom (A.W. Lo, Mamaysky, & Wang, 2002) and rounding bottoms (Zapranis & Tsinaslanidis, 2011). Financial text mining is a relatively new branch that did not get much attention until 1998 when Wüthrich et al. published their seminal paper (Wüthrich, Permunetilleke, et al., 1998). Their system forecasted the direction of the DJIA index using articles collected from the Wall Street Journal and reported a simulated profit of around 7.5% over a three-month period. The wide spread of social media such as Twitter and StockTwits deepens the impact of textual information on stock markets (Bollen et al., 2011; Oliveira et al., 2017). More recently, with the heated discussion about deep learning, several deep network structures were proposed for stock prediction with textual data (Akita et al., 2016; Ding et al., 2015; Hu et al., 2018; Xiong et al., 2015).

Driven by the accumulating evidence of decreased profitability from analyzing daily data (Kidd & Brorsen, 2004; Schulmeister, 2009), considerable recent literature has switched to higher data frequency for forecasting opportunities (Nelson, Pereira, & de Oliveira, 2017; Son, Noh, & Lee, 2012). High frequency trading, however, is not just applying forecasting frameworks described above to data of higher frequency, but a developing research area that challenges existing theories and trading practices (Aldridge, 2013). Many published works in this area attributed stock price changes to market events and microstructure, such as probability of

order arrival (Eisler, Bouchaud, & Kockelkoren, 2012), order flow imbalance (Cont, Kukanov, & Stoikov, 2014), share volume of various order types (Smith, Farmer, Gillemot, & Krishnamurthy, 2003) and gaps in order book (Farmer, Gillemot, Lillo, Mike, & Sen, 2004). These perspectives provide a more detailed picture of the dynamics in demand-supply and market's reaction in terms of stock prices (Bouchaud, Farmer, & Lillo, 2008). Using machine learning algorithms to model price changes with market microstructure features emerged recently (Kearns & Nevmyvaka, 2013), but published research under this topic is still very rare.

Stock ranking has been mentioned several times in the literature. Some used decision trees (Sorensen, Miller, & Ooi, 2000; Zhu, Philpotts, Sparks, & Stevenson, 2011), others used genetic programming (Becker, Fei, & Lester, 2007) and neural networks (Refenes, Azema-Barac, & Zapranis, 1993). However, existing literature either models the stock ranking indirectly (build models on stock return, then rank the stocks according to model predictions) or cast the ranking problem as an ordinal regression problem (Zhu et al., 2011), which complicates the original problem even more than regress-then-rank. In contrast, ranking algorithms, such as RankNet (C. Burges et al., 2005), LambdaRank (Christopher J Burges, Ragno, & Le, 2007) and LambdaMART (Christopher JC Burges, 2010), learn the ranking among instances directly. These algorithms have been proved successful in information retrieval, but there is very limited published work that applied ranking algorithms for portfolio building, if any.

Our research extends the current literature by applying LambdaMART to stock ranking with features from market microstructure, news, past stock price and technical analysis.

3.3 Research Methodology

LambdaMART is a hybrid of two techniques: LambdaRank and Multiple Additive Regression Trees (MART). It is an efficient and robust ranking algorithm mostly used for

information retrieval tasks, in which the algorithm builds a model to rank web results under a user query. As applied to our scenario, the trained model will rank all tradable stocks at a given time. MART's capability of handling missing values and its robustness to outliers make it the preferred learning algorithm for stock ranking and prediction. In this section, we clarify the derivation and rationale in the design of MART and LambdaRank in the hope to serve as a supplement in understanding these two algorithms.

3.3.1 MART

MART is a gradient boosting algorithm that approximates the chosen target function additively by building one regression tree at a time (Friedman, 2001). The final learned model is the weighted sum of all regression trees. More specifically, given the user specified iterations number M and regression tree terminal node count J, boosting builds M regression trees sequentially to form the additive model

$$F(\boldsymbol{x}) = \sum_{m=0}^{M} \beta_m f_m = \sum_{m=0}^{M} \beta_m f(\boldsymbol{x}; \boldsymbol{\gamma}_m) \tag{1}$$

where $\boldsymbol{x}$ is the input feature vector, $f(\boldsymbol{x}; \boldsymbol{\gamma}_m)$ is the m^{th} regression tree and β_m is the learned weight for the m^{th} tree . $\boldsymbol{\gamma}_m$ within $f(\boldsymbol{x}; \boldsymbol{\gamma}_m)$ fully parameterizes the tree splits (choice of feature and threshold on the chosen feature).

In each iteration, a new base learner is built to minimize the overall loss function using gradient descent method.

$$f_m(\boldsymbol{x}) = -\rho_m gradient_m(\boldsymbol{x}) = -\rho_m \frac{\partial L\big(F_{m-1}(\boldsymbol{x})\big)}{\partial F_{m-1}(\boldsymbol{x})} \tag{2}$$

Under the assumption that $F_{m-1}(\boldsymbol{x})$ is smooth and differentiable near each training sample $\boldsymbol{x}$, the parameters of $f_m(\boldsymbol{x})$ could be tuned, ideally to make $f_m(\boldsymbol{x})$'s output equivalent to $\partial L\big(F_{m-1}(\boldsymbol{x})\big)/\partial F_{m-1}(\boldsymbol{x})$, so that the model performs the gradient descent in the steepest

direction. ρ_m is an optional term for linear search included here for generality. However, under circumstances where no such parameters for $f_m(\boldsymbol{x})$ exist or it is infeasible to derive the solution due to computational cost, one could minimize the difference between the two. Thus, Eq. (2) could be represented as

$$f_m(\boldsymbol{x}) = -\rho_m h(\boldsymbol{x}; \boldsymbol{a}_m) \tag{3}$$

where $\boldsymbol{a}_m$ are the parameters that defines each base learner $h(\boldsymbol{x}; \boldsymbol{a}_m)$, and they can be obtained from the solution:

$$\boldsymbol{a}_m = arg\min_{\boldsymbol{a},\beta} \sum_{\boldsymbol{x}_i \in all\ training\ instances\ \boldsymbol{x}_1^N} [-gradient_m(\boldsymbol{x}) - \beta h(\boldsymbol{x}; \boldsymbol{a}_m)]^2 \tag{4}$$

In this way, MART reduces the complex cost function minimization problem to a stage-wise least-squares function minimization.

3.3.2 LambdaRank

LambdaRank is an efficient algorithm that could be used to learn a wide range of non-differentiable cost functions. It was originally designed to rank web documents regarding to their relevance to a given query. In information retrieval, the ranking quality evaluation functions are typically not smooth. For example, Discounted Cumulative Gain (DCG) is defined as

$$DCG_T = \sum_{i=1}^{T} \frac{2^{l_i} - 1}{\log(1+i)} \tag{5}$$

where T is the user specified truncation level, l_i is the label for the i^{th} document in the ranked list. Typically, $l_i \in \{0,1,2,3,4\}$ with 4 meaning very relevant, 0 meaning not relevant. It is easy to observe that the more relevant documents in the top T of the ranked result, the higher the DCG. $NDCG$ is a normalized version of DCG that measures the quality of a given ranking against the perfect ranking.

$$NDCG_T = \frac{DCT_T}{\max(DCT_T)} \tag{6}$$

where $\max(DCT_T)$ is DCG on the perfect ranking (ordered by l_i descending).

LambdaRank overcomes two major difficulties in learning this target cost function. First, since the target cost function is either flat or non-differentiable, gradient descent minimization would not work and other function minimization methods are non-trivial. A natural way is to find a smoothed cost function that approximates the target cost function well. LambdaRank adopts cross entropy on pairwise probability error as the smoothed cost function. More specifically, at a given time slice t, we have M tradable stocks characterized by feature vector $\boldsymbol{x_m}$, each with a label l_m. The ranking of a stock pair $\langle Stock_i, Stock_j \rangle$ is defined by their labels,

$$\begin{cases} Stock_i \rhd Stock_j, & if\ l_i > l_j \\ Stock_i = Stock_j, & if l_i = l_j \\ Stock_i \lhd Stock_j, & if\ l_i < l_j \end{cases}$$

where $Stock_i \rhd Stock_j$denotes $Stock_i$ should be ranked higher than $Stock_j$.The learning algorithm is then configured to learn the target probability of $Stock_i \rhd Stock_j$:

$$\bar{P}_{ij} = \frac{1}{2}(1 + l_{ij}) = \begin{cases} 1, & Stock_i \rhd Stock_j \\ 0.5, & Stock_i = Stock_j \\ 0, & Stock_i \lhd Stock_j \end{cases} \tag{7}$$

where $l_{ij} = l_i - l_j$. Each pair of stock features are fed into the model producing two outputs o_i and o_j. The estimated probability of $Stock_i \rhd Stock_j$ and corresponding cost function are formulated as

$$P_{ij} = \frac{1}{1 + e^{-\sigma(o_i - o_j)}} \tag{8}$$

$$C = -\bar{P}_{ij} \log P_{ij} - (1 - \bar{P}_{ij}) \log(1 - P_{ij}) \tag{9}$$

Plug $\bar{P}_{ij}$ and P_{ij} into Eq. 9 we can rewrite it to

$$C = \frac{1}{2}(1 - l_{ij})\sigma o_{ij} + \log(1 + e^{-\sigma o_{ij}}) \quad (10)$$

The second difficulty is a byproduct of this cost function formulation. Although it is now comfortingly smooth, it does not represent the target cost function $NDCG$ very well. For exmaple, imagine a list of 5 stocks labeled [4,3,2,1,0] and the model outputs [3,4,2,0,1], the smoothed cost for mis-ranking the first two stocks is the same as that of mis-ranking the last two stocks. However, in NDCG, the ranking mistakes at top positions are penalized more. This skewed attention in NDCG makes meaningful sense in IR as well as stock ranking, because we would prefer the ranker to spend more energy on learning the stocks on the head of return distribution, where it is profitable to take either a long or short position depending on the ranking critieria.

LambdaRank solves this problem by defining a "virtual gradient" on an "implicit cost function". To illustrate this concept, a refactorization of the gradient calculation needs to be carried out first. Suppose the learning algorithm approximates the function that maps training features $\boldsymbol{x}$ into labels $\boldsymbol{l}$ by tuning the parameters of a specified funtion (like in neural networks), the expected gradient of the cost function on each individual weight is given by

$$\sum_{\{i,j\} \in I} \frac{\partial C}{\partial w_k} = \sum_{\{i,j\} C \in I} \frac{\partial C}{\partial o_i}\frac{\partial o_i}{\partial w_k} + \frac{\partial C}{\partial o_j}\frac{\partial o_j}{\partial w_k} \quad (11)$$

where $\{i,j\}$ denotes all stock pairs in the same time slot that satisfies $Stock_i \rhd Stock_j$, so that $l_{ij} = 1$[17], I includes each such pair just once. w_k is the k^{th} weight in the modeled function. Gradients of the cost function at the outputs are

[17] In pairwise training, it makes sense to use only either $\langle Stock_i, Stock_j \rangle$ or $\langle Stock_j, Stock_i \rangle$ in the training set since they are essentially duplicates. Suppose $Stock_i \rhd Stock_j$, it simplfies the gradient calculation by picking $\langle Stock_i, Stock_j \rangle$ for all such pairs.

$$\frac{\partial C}{\partial o_i} = -\frac{\partial C}{\partial o_j} = \sigma\left(\frac{1}{2}(1 - l_{ij}) - \frac{1}{1 + e^{\sigma o_{ij}}}\right) = \frac{\sigma}{1 + e^{\sigma o_{ij}}} \tag{12}$$

Eq. 11 could be simplified into

$$\sum_{\{i,j\} \in I} \frac{\partial C}{\partial w_k} = \sum_{\{i,j\} \in I} \frac{\partial C}{\partial o_i}\left(\frac{\partial o_i}{\partial w_k} - \frac{\partial o_j}{\partial w_k}\right) = \sum_{\{i,j\} \in I} \lambda_{ij}\left(\frac{\partial o_i}{\partial w_k} - \frac{\partial o_j}{\partial w_k}\right) \tag{13}$$

where

$$\lambda_{ij} = \frac{\partial C}{\partial o_i} = \frac{\sigma}{1 + e^{\sigma o_{ij}}} \tag{14}$$

By defining the lambda as

$$\lambda_i = \sum_{j:\{i,j\} \in I} \lambda_{ij} - \sum_{j:\{j,i\} \in I} \lambda_{ij} \tag{15}$$

we can further simplify Eq. 13 into

$$\sum_{\{i,j\} \in I} \frac{\partial C}{\partial w_k} = \sum_i \lambda_i \left(\frac{\partial o_i}{\partial w_k}\right) \tag{16}$$

where the gradient of the cost function C with regard to a function parameter w_k, is refactored into the product of λ_i and the gradient of function output with regard to function parameter. Clearly, λ_i is the gradient of the cost function with regard to the function output.

To solve the problem where pairwise error probabilistic cost function does not approximate NDCG cost function well, LambdaRank augments the original gradient λ_{ij} by the delta of NDCG given by swapping the rank positions of two stocks.

$$\lambda_{ij} = \frac{\sigma}{1 + e^{\sigma o_{ij}}} |\Delta_{NDCG}| \tag{17}$$

With this augmented λ_{ij}, more weights are given to pairwise error at the top of the ranked list, which approximates the target cost function quite well and smoothness is inherited from the pairwise error probabilistic cost function. It can be interpreted as virtual gradients of an implicit

cost function defined at each model output. (Donmez, Svore, & Burges, 2009) show empirically that performing gradient descent with respect to λ_{ij} directly optimizes NDCG.

3.3.3 LambdaMART

With a well-defined gradient on ranking cost function from LambdaRank and a framework that breaks the complex function approximation problem down to stage-wise gradient approximation from MART, LambdaMART combines the merits from both algorithms for efficient and robust ranking optimization. Detailed algorithm pseudocode is listed in (Christopher JC Burges, 2010).

3.4 Research design

3.4.1 Data

3.4.1.1 Order Flow and Transactions Data

We have gathered the order flow and transaction data from Shenzhen stock exchange from 03/01/2017 to 07/28/2017. Order data is the finest description of trading activities in the market. It contains all orders received by the electronic trading platform, with details about timestamp, price (if it is limit order), volume and order type. Market order, limit order and cancellations are all included and listed as different order types in the data. In general, market orders are more aggressive because they are matched immediately with the best opposite order in the order book, while limit orders wait in the limit order book until an opposite order can be paired with it. Since there are a variety of market order types in the ShenZhen stock exchange, some equivalent to limit orders, we adopt the notation of effective market/limit orders in (Farmer et al., 2004) for brevity. Effect market order refers to orders that are filled immediately (partially or completely), while effective limit order refers to orders that wait in the limit order book for

some time. Transaction data contains trading details about filled orders, with fields specifying the timestamp, execution price, execution volume, buy order details, sell order details, etc.

The sheer volume of this data is 1.31 TB and it imposes great pressure on computation and storage, which is why the training data was not extended back longer. The data is available to all market participants in real-time at a cost, which is similar to the Level 2 market quotes in U.S. markets.

3.4.1.2 Aggregation and Feature Design

Orders and transactions arrive in irregular intervals. Since many price forecasters in the literature adopt the time-series model that aggregates features within fixed time spans, for a reasonable baseline and fair comparison, our ranker and price regressors are trained on aggregated features. Despite the information loss due to this aggregation, our results show that LambdaMART is capable of building profitable portfolios from such features. It is noteworthy that many continuous-time models had been built to couple with irregular spacing of high-frequency data (Engle, 2000), which is a future research direction with much potential. Features we use for ranking and price modeling roughly fall into four categories: order features, limit order book features, technical indicator and news.

The order flow data contains precious information about the market participants' trading strategy and their level of optimism. Previous research has shown that investors could be subject to waves of optimism and pessimism(Nardo et al., 2016), suggesting that the sentiment on a stock could persist in the near future. A straight-forward measure of the market sentiment is the ratio of effective market orders to effective limit orders. We calculate the market order ratio on both bid and ask orders in regard to both order count and order value. Andrew et al. (Andrew W. Lo, MacKinlay, & Zhang, 2002) found that the limit orders' execution time is sensitive to the

price difference between the order price and the security price. To further extend the granularity of our order sentiment feature, we bucketize the limit orders with regard to their price difference from the security price and calculate the ratio of each bucket to all orders. Hewlett (Hewlett, 2006) observed that market orders tend to arrive in clusters, which could be explained by some well-known order execution strategies like batch ordering, with which large orders are executed in small blocks to minimize the impact on the market (Almgren & Chriss, 2001). To capture the dynamics of order flows, we calculate the moving averages (MA) on the count and value of each order bucket and use the MA crossover signals as features to hint the beginning and end of an order cluster.

With complete order data, one can reconstruct the limit order book that accommodates all limit orders awaiting execution. Empirical studies have shown that the state of order book contains information about the future price movements. (Farmer et al., 2004) showed that large price fluctuations could be attributed to gaps in the order book. Bid-ask spread was observed to be associated to securities' return (Amihud & Mendelson, 1986). We use the gaps between each price level in the order book, as well as the bid-ask spread broken down to spread to bid and spread to ask as our order book features.

Practitioners use rules built on top of technical indicators to assist their trading. We adopt some crossover signals built on commonly used indicators such as Moving Averages (MA), Moving Average Convergence Divergence (MACD), Stochastic Oscillators (usually called KDJ indicators in Chinese markets) and Bollinger bands. Detailed calculations of each indicator and feature are listed in the appendix. It is noteworthy that some indicators, even basic ones like

moving average[18], are calculated differently in Chinese markets, and we find that features calculated in the market-specific way have higher association with stock ranking.

Financial news plays an important role in investors' sentiment formation, which may drive money flow and stock prices. We implement a polite web crawler[19] to retrieve financial news articles from five popular Chinese financial sites, namely, http://finance.qq.com/, http://finance.sina.com/, https://xueqiu.com/, http://www.caijing.com.cn/ and http://www.stockstar.com/. Retrieved news articles cover a wide range of news sources including official reports, general news about companies and market sectors, stock ratings and forum buzz. The crawled HTML webpages go through a pipeline of processors including text extraction, Chinese tokenization, deduplication and time stamping, stock and market sector classification. Text extraction is a much simpler task in Chinese webpages than in English ones because of the clear separation of Chinese text and HTML code written in English characters. A simple charset filtering suffices our needs. Chinese is written without spaces between words. Thus, in order to do text classification, we use a natural language processing package called ICTCLAS (H. Zhang) for word segmentation. Shingle hash deduplication is then performed on segmented N-grams to remove near duplicates from different sources. Related stocks of a given article are tagged by stock ticker matching (binary term frequency), and related market segments are identified with a naïve Bayesian classifier (Mitchell, 1997). With the contents of news articles classified and time stamped, we aggregate the count of news articles on each stock and its market segment at various time periods and use such features to give the learner a clue of dynamics in market attention and volatility.

[18] In the US markets, most practitioners regard the first few entries of MA as null because there is not enough data for the specified window, but most stock quote software in China shrink the window size when data is insufficient.

[19] We respect the Robots Exclusion Protocol and limit the rate we crawl each site. Since this crawler is just for research and study purpose, and we only do back testing, the requirement on crawling speed is low.

3.4.1.3 Label

The rate of change in stock price is the basis of labels used for the ranker and regressors. Typical price change rate of a stock at time t is calculated as

$$Return_t = \left(Price_{t+lag} - Price_t\right)/Price_t$$

However, at the data frequency we use, market dynamics can hardly be captured by just the stock price, which typically refers to the trading price of the last transaction that took place within a given time period. Due to gaps in the limit order book and sometimes large bid-ask spread, the closing prices at each second could flick up and down within the spread, introducing non-trivial volatility to the return label. For example, shifting price within a spread of 5 cents on a 10-dollar stock could cause 0.5% fluctuation in the calculated return, which is considered a good gain if the position is to be closed within a short period of time. However, such calculated return can hardly be realized because the order that triggers the last price flip could be infinitely small. To avoid such problem, we amend the return calculation into

$$Return_t = \frac{\left(VMP_{t+lag} - BestOppositeLimitOrderPrice_t\right)}{BestOppositeLimitOrderPrice_t}$$

where

$$VMP_{t+lag} = \sum_{transactions\ in\ [t,t+lag]} transaction\ price * transaction\ volume$$

The best opposite limit order price is a realistic price for opening a position, and VMP stands for volume weighted price that gives more weight to prices at which more shares are exchanged. We choose the lag to be 30 seconds empirically.

The calculated return for each stock at each second is used directly as labels for the neural network return regressor. For the ranker labels, all tradable stocks at each second are sorted on the calculated return, then grouped into 5 buckets with labels 4, 3, 2, 1, 0 respectively.

3.4.2 Model parameter tuning

We perform model parameter tuning with data from 03/01/2017 to 04/30/2017, which is further split into training (data before 03/31/2017), validation (between 04/01/2017 and 04/15/2017) and testing (after 04/16/2017). The tuning is performed in a greedy fashion, i.e. find the best performing value for a parameter by randomizing it while keeping all other parameters constant and iterate until all parameters are chosen. For LambdaMART, the key parameters are number of leaves per tree, learning rate and number of iterations (trees).

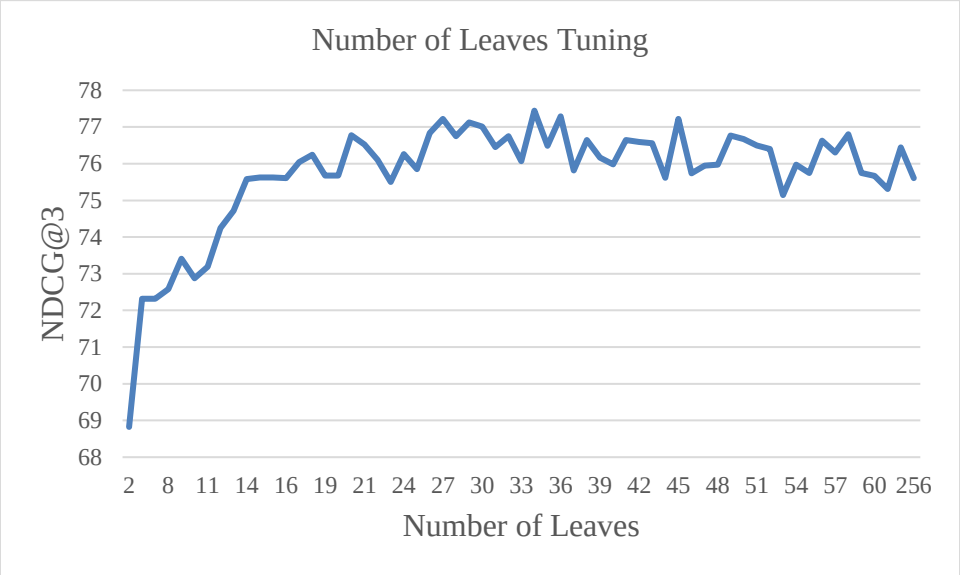

Figure 3-1 Tuning Result for Number of Leaves

The number of leaves per tree specifies the complexity of each weak learner in boosting. The optimal choice for this model parameter depends on the shape of the target function to be approximated. It could be as low as two, which refers to a stump with two directed arcs connecting the same splitting node to two separate leaf nodes. Hastie et. al. (Hastie, Tibshirani, & Friedman, 2009) showed that boosting with stumps performs better than with deeper trees for solving the nested sphere problem, in which the target function is an additive quadratic multivariable equation with no interaction among each variable. Since it is unclear what the

shape of our target function is, we varied the number of leaves from 2 to 256 and observed the following NDCG@3 on the testing dataset. It appears that increasing tree depth further than 30 does not benefit the overall model's performance.

The purpose of tuning the learning rate and number of trees is mostly to avoid overfitting and local optima. Ideally, we could just use a small learning rate and a large number of trees, then enforce early stopping rules to avoid overfitting. But due to the size of our data, learning rate being set too small may risk not getting an optimal model trained within a reasonable time. The following plot shows NDCG@3 on testing data from models trained to a maximum of 4000 trees without early stopping. Although learning rate at 0.15 achieved the highest NDCG@3 at iteration 1700, we prefer smaller learning rate 0.1 because it has smoother progress and the model reached comparable performance within reasonable iterations.

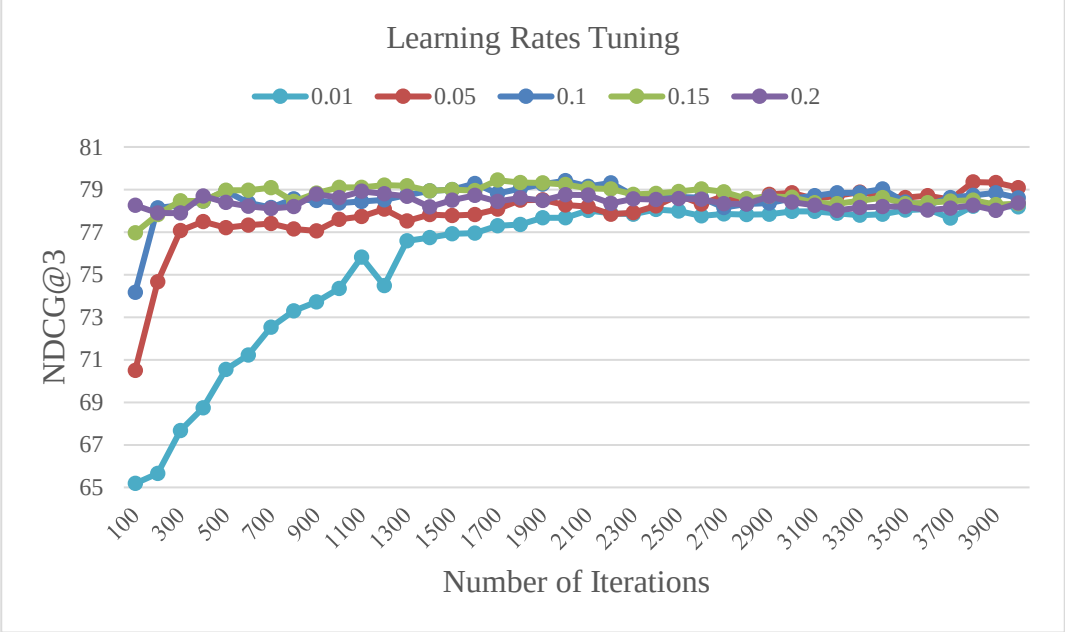

Figure 3-2 Tuning result for Learning Rate

With number of leaves and learning rate chosen, we set the number of trees to a relatively large value (5000), and apply early stopping to mitigate overfitting. Lodwich et al. (Lodwich,

Rangoni, & Breuel, 2009) compared several early stopping rules and suggested that Low Progress stopping criteria will more likely give better results given limited prior knowledge. Since LambdaMART seeks to maximize NDCG, we define the low progress rule as

$$P_k(t) = 1000\left(\frac{max_{t-k+1}^{t} NDCG}{avg_{t-k+1}^{t} NDCG} - 1\right) < \alpha$$

where our choice of k and α are 5 and 1, respectively.

For the neural network regressors, we need to decide the network structure as well as parameters related to alleviating local optima and overfitting, such as learning rate, momentum, weight decay, number of iterations and early stopping rule. Since we have enough data for a separate validation set, learning rate, momentum and weight decay are set to small positive values, and we mostly rely on early stopping to stop training from a large number of iterations. Low progress early stopping criteria is defined in the same way as (Prechelt, 1998).

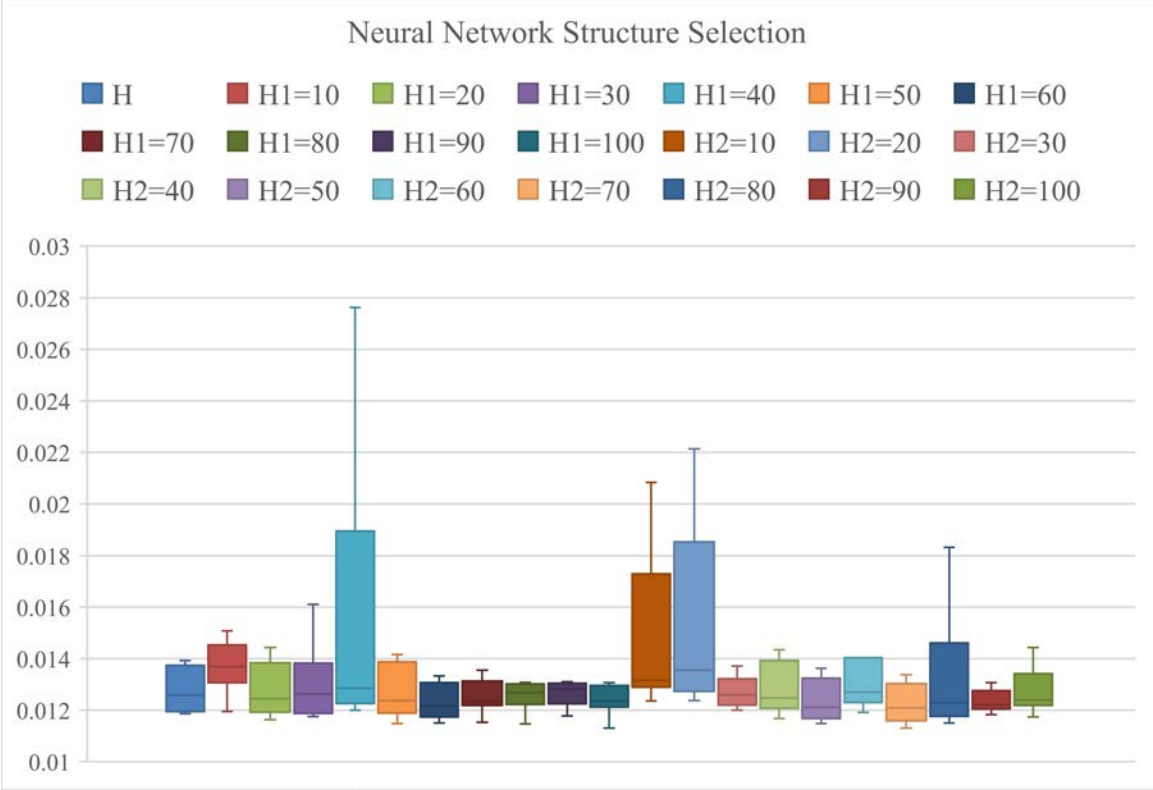

Figure 3-3 Tuning Neural Network Structure

There is no general guideline for network structure design. (Atsalakis & Valavanis, 2009) surveyed about a hundred papers that used neural networks for predicting stock price, return, risk and their composites. From their summarization, we observe that most designs use 1 to 2 hidden layers with no more than 60 nodes in each hidden layer. To find the structure that fits the problem we are trying to solve, we follow the observed guideline and experiment with 110 different network structures, including ten 1-hidden-layer networks with the number of hidden nodes spread from 10 to 100 at equal step size of 10, and one hundred 2-hidden-layer networks sampled in the same fashion. Test errors of each structure are clustered by number of hidden layers and number of hidden node at each layer, shown in the box plot in Figure 3-3. We observe that 2-hidden-layer networks with higher than 60 nodes in the first hidden layer perform better than 1-hidden-layer networks, and that 50 ~ 70 hidden nodes in the second hidden layer generally give better results than other choices. Our network structure is finalized as 61 input nodes, 100 nodes in the first hidden layer, 70 nodes in the second and 1 linear output node. Each layer is fully connected with the preceding one.

3.5 Results

After tuning the parameters for both the LambdaMART ranker and the Neural Network (NN) regressor on parameter tuning dataset (from 2017-03-01 to 2017-04-30), the learning algorithms are trained on our training dataset (from 2017-05-01 to 2017-06-30), followed by the system evaluation performed on out-of-sample test dataset (from 2017-07-01 to 2017-07-28). Ranker is trained to optimize NDCG of volume weighted return at a 30 seconds lag. NN regressor is trained to predict the same return directly. To better demonstrate the modeling power of NN, we also train a network to predict the exact price at 1-second lag. These two NN models

have the same structure and learning parameters, and they are denoted as return regressor and price regressor respectively. Features used for all three models are identical.

3.5.1 Metrics on Test Dataset

NDCG metrics are reported for ranking quality comparison. For the regressor, an extra ranking step needs to be taken prior to assessing its NDCG. We configure the evaluator to rank the stocks based on their predicted return, which can be viewed as investing in the most profitable stocks according to the learned model's prediction. NDCGs for both ranker and regressor are averaged across 312,202 evaluations in the test dataset (22 trading days, 14191 seconds per day). For the return regressor, we also report the regressor errors in basis points (4.8546 L1 score means 0.00048546 mean absolute error in predicting the return on testing dataset), averaged over 546.4 million evaluations (about 1750 tradable stocks per second per day). The reported error in price regressor is CNY. Note that the minimal tick for all stocks in the target market is 0.01 CNY, and an L1 score of 0.004 gets the regressor very close to the actual price. NDCG is not applicable to the price regressor because there is not much sense in ranking stocks on their prices.

Table 3-1 Metrics of Ranker and Regressors on Out-of-Sample Testing Dataset

	NDCG@1	**NDCG@2**	**NDCG@3**	**L1**	**L2**	**RMS**
Ranker	87.8221	85.0387	82.7252	-	-	-
Return Regressor	22.1763	18.3154	10.9981	4.8546	104.8171	10.2380
Price Regressor	-	-	-	0.0040	0.0002	0.0165

As shown in the table, the ranker outperforms the return regressor drastically in terms of NDCG. This significant ranking quality difference is also confirmed by simulated trading profit shown in the next section, suggesting that although the NN learners are capable of approximating

the stock price and return closely, the lack of ranking concept in the objective function makes the learner less reliable for constructing portfolios directly. More specifically, in LambdaMART, an error at the head of the return distribution is penalized more than an error of the same scale at the body and tail, but regressor models would treat such errors equally. The ranker may perform worse than regressors at the body and tail, but the ranking in the head is what matters the most in investment.

3.5.2 Ranked feature importance

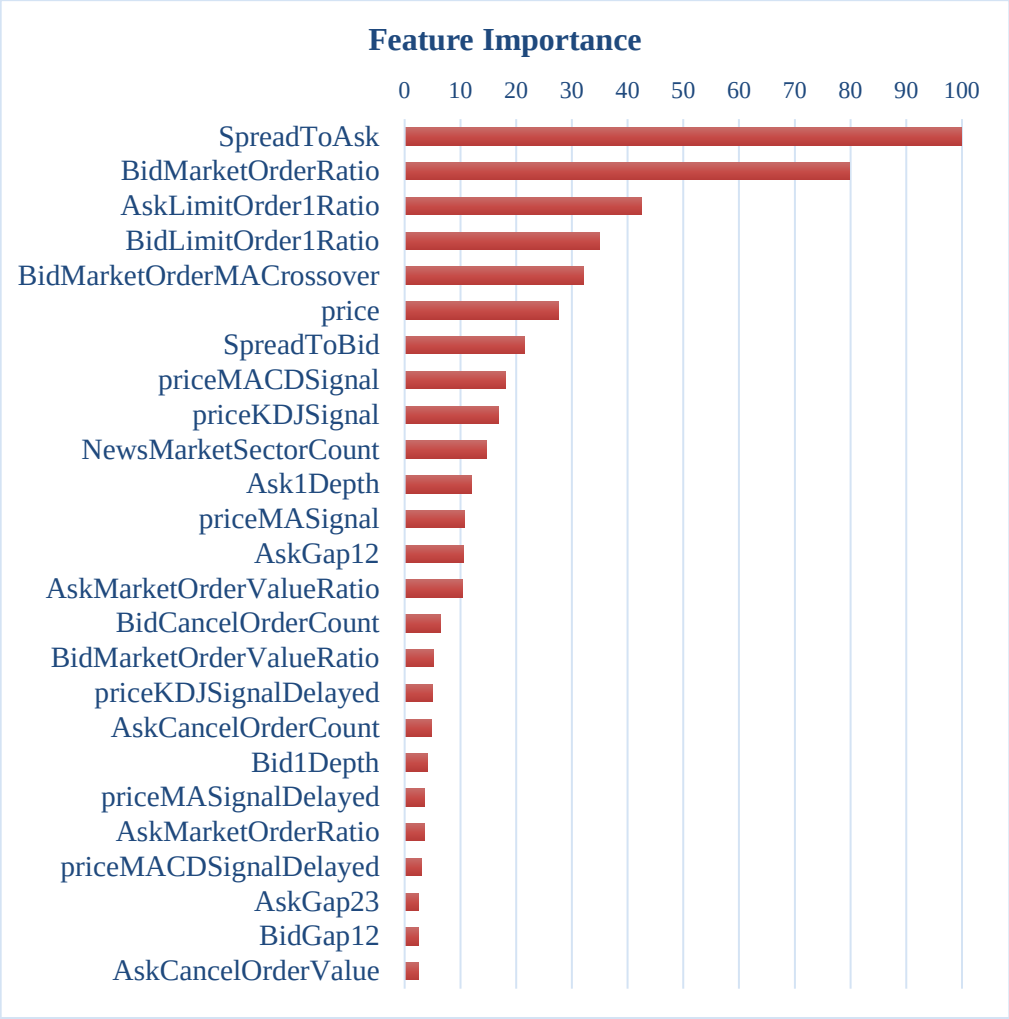

Figure 3-4 Relative Importance of Different Features

One benefit of decision trees over neural networks is their interpretability. It is easy to compare the relative importance of features by summing the information gain of each splitting feature used in internal nodes (Breiman, Friedman, Olshen, & Stone, 1984). For boosted decision trees, the importance measure is averaged over all trees for each feature (Hastie et al., 2009). The graph below shows the top 25 features in our ranker model ordered by importance. Our observations can be summarized in 3 aspects. First, bid-ask spread and order book sentiment features are most impactful in the learned ranking model. This suggests that features extracted from the order data and limit order book have more significant impact on short term return than technical analysis and news. Second, simple technical rules like crossover signals on MACD, KDJ and moving averages are associated with short term return at the data frequency we examine. Third, although news count per market sector is helpful in ranking the return, news count per stock is not selected as splitting feature in any tree within the ensemble.

3.5.3 Trading Simulation

To demonstrate the profitability of our trained system, we design a naïve trading system and calculate the simulated profit on the testing dataset. Since short positions are not supported in our target market, our trading rules only take long positions, defined as follows:

1. Divide the fund into three buckets.
2. Open a long position on the stock with optimal ranking for each unallocated fund bucket. Since there are 3 buckets, we could long the top 3 stocks at once, corresponding to NDCG@3 reported in the previous section.
    a. Orders are executed at optimal ask price in the limit order book.
    b. We assume there is enough ask quotes at the optimal ask price to use up all fund in the bucket, which is unrealistic for big funds. But since designing an order

execution strategy is out of the scope of this research, we make this assumption on both the ranker and regressor for brevity.

3. Close the position if the stock falls out of top 100 and there is no trading halt (H.-C. Xu, Zhang, & Liu, 2014). If a halt is imposed, the position will be cashed immediately after the halt is lifted.
    a. Orders are executed at optimal bid price in the limit order book.
    b. Same assumption is made as opening the position.

Transaction costs are deducted from the profit of each position. Stamp tax (0.1%), government fees (0.02%) and broker commissions (varies per broker from 0.00% to 0.1%) mount up to an estimated total of 0.23% on the closing value of the position. The following table shows the statistics about model profitability. Investment decisions made by the ranker are much better than those made by neural networks in terms of both absolute return and risk-adjusted return. Note that the average return per position from the neural networks model is not sufficient to cover transaction cost and the asset value approaches zero with the max drawdown over 1.

Table 3-2 Simulated Profits

	Average Return per position	**Return Standard Deviation**	**Return Sharpe Ratio**	**Return Max Drawdown**	**Average Positions Count per bucket**	**Average Position Duration**
LambdaMART	0.000537	0.003200	0.167813	0.575733	495.02 / day	30.57 seconds
Neural Networks	-0.002366	0.011839	-0.199848	1.022054	297.26 / day	47.57 seconds

The following plot shows a one-day cumulative gain of LambdaMART and Neural Networks, compared to the market index on simulated trading. The ranker portfolio performs

better than the overall market index, while neural network portfolio suffers from transaction costs and the asset value keeps regressing after a brief hike in the morning.

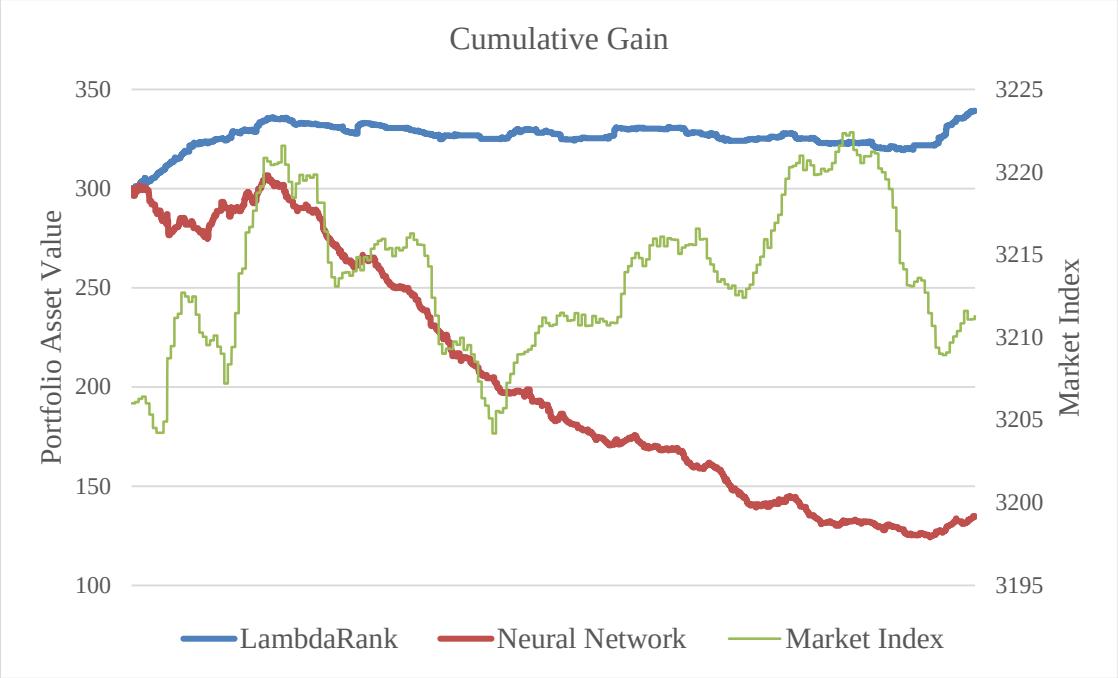

Figure 3-5 Cumulative Gain of Ranker, Neural Network and the Market Index

To further understand the reason why low regressor test error does not lead to good profitability in the NN model, we plot the outputs of the ranker, return regressor, price regressor, as well as the actual price of a stock in the following figure. We observe that both ranker and regressor have lag in their predictions: the ranker promotes this stock about 2 seconds after the beginning of the upward trend, whereas the regressor has an obvious 1 second lag in the sideways trend starting from time index 173. We attribute the poor profitability of the regressors to the lag in their predictions. Due to the rapid oscillating price movements and the lag in predicted prices from index 174 to 200, the price regressor forecasts the price movement in the wrong direction about 80% of the time, and the return regressor generates unprofitable buy signals for three times, each with a 2 seconds lag. Although the ranker output in the sideways period is volatile as well, it never pushes this stock to the top 3 positions.

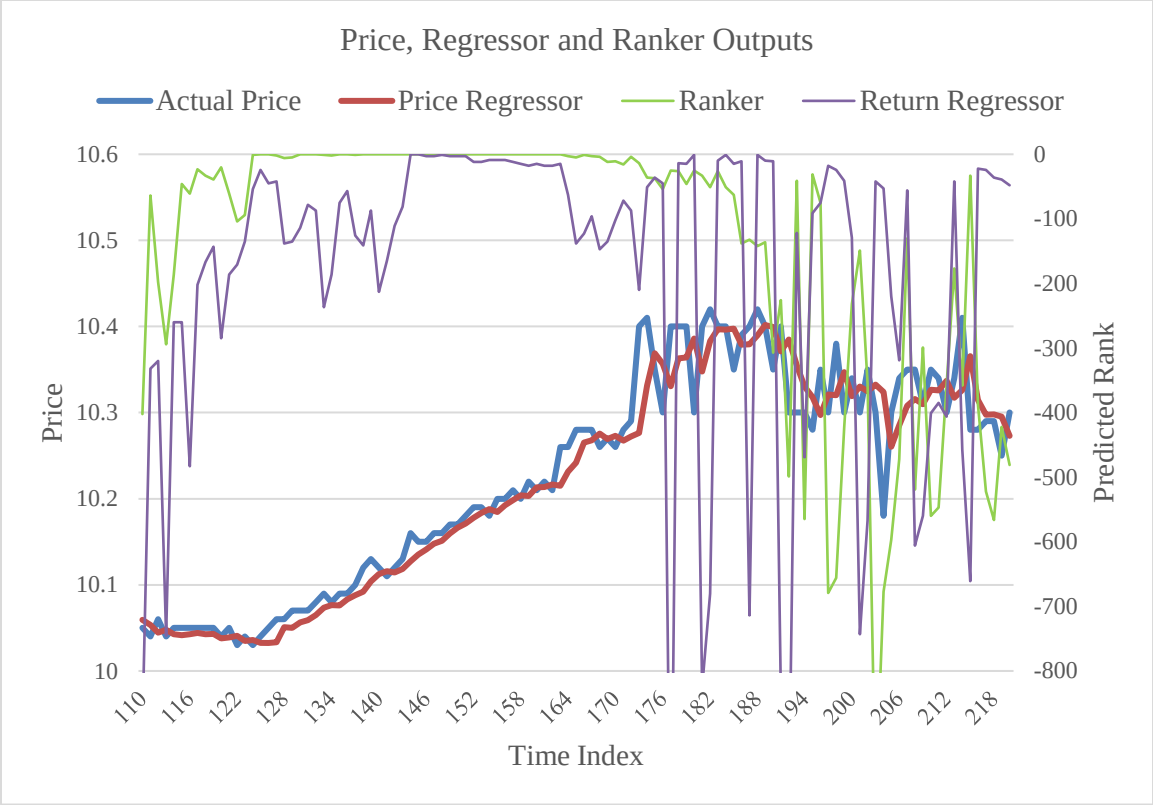

Figure 3-6 Price, regressor and ranker outputs for 000916 on 07/10/2017. The plotted data starts from 09:31:50AM (time index 110) and ends in 09:33:50AM. Left axis shows the price. Right axis shows the ranking in which 0 means best, the lower the worse. Our trading strategy opens a position when a stock is ranked to top 3 at a given time index and closes the position when it falls out of top 100. The ranker opened only 1 position at index 127 and closed at 187. Return regressor opened 4 positions in periods 145 - 161, 180 - 181, 183-184, 188-190.

3.6 Conclusion and Future Work

In this research, we design a stock ranking system that uses LambdaMART to predict the best-performing stocks in an intraday scale with data from ShenZhen stock exchange. We demonstrate that the ranking algorithm has significantly better performance in portfolio selection than neural network regressors in terms of ranking quality and simulated profits. Both the ranker

and the regressor are trained on features extracted from diverse trading practices in the hope that these features, collectively, could reflect a majority of factors involved in traders' decision making. The interpretability of tree learners gives us insight to our learned ranker model, and we observe that market microstructure features are the most impactful features in our ranker, followed by current price, technical features and news. Trading simulation is performed on out-of-sample test data under rigorous conditions with respect to trading price and transaction cost, and the results demonstrate strong profitability of the ranking algorithm.

Aligning terabytes of order and transaction data with online news for forecasting is an exciting area that is barely explored. We expect to continue our work in the following areas. First, although LambdaMART is quite efficient and robust, it is not straight forward to recognize price patterns which is an important factor in many traders' decision-making process. The pattern representation transformation widely used in deep learning is a good supplement to this shortage of LambdaMART. It can either be used as a pre-processing model that generates features to be consumed by the ranker or used as the base learner in the tree ensemble. Recent developments in high frequency financial econometric shed light on more features we could use for better ranking result (Aït-Sahalia & Jacod, 2014). And finally, order execution can be formulated into a constrained multi-objective optimization problem for which learning models would seek to optimize the execution price and certainty within the constraints of time and rapidly changing market microstructure.

CHAPTER 4

CONCLUSIONS

The primary contributions of this research can be summarized as follows. First, we did an up-to-date review of stock prediction with text mining techniques, which completes other surveys in the literature that focused on learning numerical features such as price and fundamental variables. We organize and summarize financial text mining techniques in six aspects: news source selection, text preprocessing, document alignment and labeling, time series preprocessing, forecasting algorithm, and performance evaluation. We list available configuration choices in tables for each design aspect and highlight the performance comparison of different alternatives available in the literature. The survey is finished with a summary of most recent developments in this area and some suggestions on possible future research directions. In particular, we find that using deep learning methods to learn news sentiments and message volume features gained increasing popularity in the past few years. For future research, we suggest exploring the relative importance among multiple market factors based on considerations of improved news data availability and lack of such comparison in the literature. In Chapter 3, we follow the trend observed in the survey to propose a new stock forecasting method that builds ranking models on news and market microstructure features. We compare the ranker with a fine-tuned neural network model which is commonly used for stock forecasting in the literature, and we show that the stock ranker based on LambdaMART yields significantly higher profits on out-of-sample testing data after deducting transaction costs. With data gathered from the ShenZhen stock exchange, LambdaMART scored an NDCG of 82.725 and 0.054% in return per position,

while neural network return regressor can only get 10.998 in NDCG and its averaged return per position is -0.237%. By simulating the trading under rigorous constraints of transaction costs and order execution price, we also demonstrate that the ranker can be used to build highly profitable portfolios for real investments. Third, the relative importance among features like market microstructure, technical analysis, news and past price was never reported in the literature, partly because most soft-computing methods are black-box models that are hard to interpret. We find that by assessing the average information gain associated with each feature across all weak learners, market microstructure features are of most importance to build the ranker that can be used to construct profitable portfolio.

REFERENCES

Aase, K.-G. (2011). *Text Mining of News Articles for Stock Price Predictions.* (Master of Science), Norwegian University of Science and Technology.

Aït-Sahalia, Y., & Jacod, J. (2014). *High-frequency financial econometrics*: Princeton University Press.

Akita, R., Yoshihara, A., Matsubara, T., & Uehara, K. (2016). *Deep learning for stock prediction using numerical and textual information.* Paper presented at the Computer and Information Science (ICIS), 2016 IEEE/ACIS 15th International Conference on.

Aldridge, I. (2013). *High-frequency trading: a practical guide to algorithmic strategies and trading systems* (Vol. 604): John Wiley & Sons.

Almgren, R., & Chriss, N. (2001). Optimal execution of portfolio transactions. *Journal of Risk, 3*, 5-40.

Amihud, Y., & Mendelson, H. (1986). Asset pricing and the bid-ask spread. *Journal of Financial Economics, 17*(2), 223-249.

Antweiler, W., & Frank, M. Z. (2004). Is All That Talk Just Noise? The Information Content of Internet Stock Message Boards. *Journal of Finance, 59*(3), 1259-1294. doi: 10.1111/j.1540-6261.2004.00662.x

Atsalakis, G. S., & Valavanis, K. P. (2009). Surveying stock market forecasting techniques–Part II: Soft computing methods. *Expert Systems with applications, 36*(3), 5932-5941.

Becker, Y. L., Fei, P., & Lester, A. M. (2007). Stock selection: An innovative application of genetic programming methodology *Genetic Programming Theory and Practice IV* (pp. 315-334):

Springer.

Blostein, D., Zanibbi, R., Nagy, G., & Harrap, R. (2003). *Document representations*.

Bollen, J., Mao, H., & Zeng, X. (2011). Twitter mood predicts the stock market. *Journal of Computational Science, 2*(1), 1-8.

Boser, B. E., Guyon, I. M., & Vapnik, V. N. (1992). *A training algorithm for optimal margin classifiers.* Paper presented at the Proceedings of the fifth annual workshop on Computational learning theory.

Bouchaud, J.-P., Farmer, J. D., & Lillo, F. (2008). How markets slowly digest changes in supply and demand. *arXiv preprint arXiv:0809.0822*.

Bouktif, S., & Awad, M. A. (2013). *Ant colony based approach to predict stock market movement from mood collected on Twitter.* Paper presented at the Proceedings of the 2013 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining.

Bozic, C., Chalup, S., & Seese, D. (2012). Application of Intelligent Systems for News Analytics. *Financial Decision Making Using Computational Intelligence*, 71-101.

Breiman, L., Friedman, J. H., Olshen, R. A., & Stone, C. J. (1984). *Classification and Regression Trees*. Monterey, CA: Wadsworth and Brooks.

Burges, C., Shaked, T., Renshaw, E., Lazier, A., Deeds, M., Hamilton, N., & Hullender, G. (2005). *Learning to rank using gradient descent.* Paper presented at the Proceedings of the 22nd international conference on Machine learning.

Burges, C. J. (2010). From ranknet to lambdarank to lambdamart: An overview. *Learning, 11*(23-581), 81.

Burges, C. J., Ragno, R., & Le, Q. V. (2007). *Learning to rank with nonsmooth cost functions.* Paper presented at the Advances in neural information processing systems.

Chang, P.-C., & Liu, C.-H. (2008). A TSK type fuzzy rule based system for stock price prediction. *Expert Systems with applications, 34*(1), 135-144.

Cohen-Charash, Y., Scherbaum, C. A., Kammeyer-Mueller, J. D., & Staw, B. M. (2013). Mood and the market: can press reports of investors' mood predict stock prices? *PloS one, 8*(8), e72031.

Cont, R. (2011). Statistical modeling of high-frequency financial data. *IEEE Signal Processing Magazine, 28*(5), 16-25.

Cont, R., Kukanov, A., & Stoikov, S. (2014). The price impact of order book events. *Journal of financial econometrics, 12*(1), 47-88.

Dange, S. N., Argiddi, R. V., & Apte, S. S. (2012). Financial Trading System using Combination of Textual and Numerical Data. *International Journal of Computer Applications, 51*, 36.

De Bondt, W. F. M., & Thaler, R. (1985). Does the stock market overreact? *Journal of finance*, 793-805.

Ding, X., Zhang, Y., Liu, T., & Duan, J. (2015). *Deep learning for event-driven stock prediction.* Paper presented at the Ijcai.

Dondio, P. (2013). *Stock Market Prediction without Sentiment Analysis: Using a Web-Traffic Based Classifier and User-Level Analysis.* Paper presented at the System Sciences (HICSS), 2013 46th Hawaii International Conference on.

Donmez, P., Svore, K. M., & Burges, C. J. (2009). *On the local optimality of LambdaRank.* Paper presented at the Proceedings of the 32nd international ACM SIGIR conference on Research and development in information retrieval.

Eisler, Z., Bouchaud, J.-P., & Kockelkoren, J. (2012). The price impact of order book events: market orders, limit orders and cancellations. *Quantitative finance, 12*(9), 1395-1419.

Engle, R. F. (2000). The econometrics of ultra-high-frequency data. *Econometrica, 68*(1), 1-22.

Fagan, S., & Gencay, R. (2009). An Introduction to Textual Econometrics. *Handbook of Empirical Economics and Finance*, 133.

Fama, E. F. (1965). The behavior of stock-market prices. *The journal of Business, 38*(1), 34-105.

Fama, E. F. (1970). Efficient capital markets: A review of theory and empirical work. *The Journal of Finance, 25*(2), 383-417.

Farmer, J. D., Gillemot, L., Lillo, F., Mike, S., & Sen, A. (2004). What really causes large price changes? *Quantitative finance, 4*(4), 383-397.

Feldman, R., & Sanger, J. (2006). *The text mining handbook: advanced approaches in analyzing unstructured data*: Cambridge University Press.

Forman, G. (2003). An extensive empirical study of feature selection metrics for text classification. *J. Mach. Learn. Res., 3*, 1289-1305.

Friedman, J. H. (2001). Greedy function approximation: a gradient boosting machine. *Annals of statistics*, 1189-1232.

Fung, G. P. C., Yu, J. X., & Lam, W. (2002). News sensitive stock trend prediction *Advances in Knowledge Discovery and Data Mining* (pp. 481-493): Springer.

Gidófalvi, G., & Elkan, C. (2001). Using news articles to predict stock price movements. *Department of Computer Science and Engineering, University of California, San Diego*.

Gilbert, E., & Karahalios, K. (2010). *Widespread Worry and the Stock Market.* Paper presented at the ICWSM.

Graham, B., & Dodd, D. L. (1934). *Security analysis: Principles and technique*: McGraw-Hill.

Groth, S. S., & Muntermann, J. (2011). An intraday market risk management approach based on textual analysis. *Decision Support Systems, 50*(4), 680-691.

Gunduz, H., & Cataltepe, Z. (2013). *Prediction of Istanbul Stock Exchange (ISE) Direction Based On News Articles.* Paper presented at the The Third International Conference on Digital Information Processing and Communications (ICDIPC2013).

Hagenau, M., Hauser, M., Liebmann, M., & Neumann, D. (2013). *Reading all the news at the same time: Predicting mid-term stock price developments based on news momentum.* Paper presented at the System Sciences (HICSS), 2013 46th Hawaii International Conference on.

Hagenau, M., Liebmann, M., & Neumann, D. (2013). Automated news reading: Stock price prediction based on financial news using context-capturing features. *Decision Support Systems*, *55*(3), 685-697.

Han, Z. (2012). *DATA AND TEXT MINING OF FINANCIAL MARKETS USING NEWS AND SOCIAL MEDIA.* (Master of Science), University of Manchester.

Hastie, T., Tibshirani, R., & Friedman, J. (2009). *The elements of statistical learning* (2nd ed.): Springer.

He, J., Tan, A. H., & Tan, C. L. (2003). On machine learning methods for Chinese document categorization. *Applied Intelligence, 18*(3), 311-322.

Hewlett, P. (2006). *Clustering of order arrivals, price impact and trade path optimisation.* Paper presented at the Workshop on Financial Modeling with Jump processes, Ecole Polytechnique.

Hsu, C.-W., Chang, C.-C., & Lin, C.-J. (2003). A practical guide to support vector classification. from https://www.cs.sfu.ca/people/Faculty/teaching/726/spring11/svmguide.pdf

Hu, Z., Liu, W., Bian, J., Liu, X., & Liu, T.-Y. (2017). Listening to Chaotic Whispers: A Deep Learning Framework for News-oriented Stock Trend Prediction. *arXiv preprint arXiv:1712.02136*.

Hu, Z., Liu, W., Bian, J., Liu, X., & Liu, T.-Y. (2018). *Listening to Chaotic Whispers: A Deep Learning Framework for News-oriented Stock Trend Prediction.* Paper presented at the Proceedings of the Eleventh ACM International Conference on Web Search and Data Mining.

Huang, C.-F. (2012). A hybrid stock selection model using genetic algorithms and support vector regression. *Applied Soft Computing, 12*(2), 807-818.

Huang, C.-J., Liao, J.-J., Yang, D.-X., Chang, T.-Y., & Luo, Y.-C. (2010). Realization of a news dissemination agent based on weighted association rules and text mining techniques. *Expert Systems with Applications, 37*(9), 6409-6413.

Ince, H., & Trafalis, T. B. (2007). Kernel principal component analysis and support vector machines for stock price prediction. *IIE Transactions, 39*(6), 629-637.

Jensen, R., & Shen, Q. (2008). *Computational intelligence and feature selection: rough and fuzzy approaches* (Vol. 8): Wiley-IEEE Press.

Joachims, T. (1998). Text categorization with support vector machines: Learning with many relevant features. *Machine learning: ECML-98*, 137-142.

Junqué de Fortuny, E., De Smedt, T., Martens, D., & Daelemans, W. (2014). Evaluating and understanding text-based stock price prediction models. *Information Processing & Management*.

Kaastra, I., & Boyd, M. (1996). Designing a neural network for forecasting financial and economic time series. *Neurocomputing*, *10*(3), 215-236.

Kearns, M., & Nevmyvaka, Y. (2013). Machine learning for market microstructure and high frequency trading. *High Frequency Trading: New Realities for Traders, Markets, and Regulators*.

Kharratzadeh, M., & Coates, M. (2012). *Weblog Analysis for Predicting Correlations in Stock Price Evolutions.* Paper presented at the ICWSM.

Kidd, W. V., & Brorsen, B. W. (2004). Why have the returns to technical analysis decreased? *Journal of Economics and Business, 56*(3), 159-176.

Kumar, R. B., Kumar, B. S., & Prasad, C. S. S. (2012). Financial News Classification using SVM. *International Journal of Scientific and Research Publications, 2*(3).

Lam, M. (2004). Neural network techniques for financial performance prediction: integrating fundamental and technical analysis. *Decision support systems, 37*(4), 567-581.

Lavrenko, V., Schmill, M., Lawrie, D., Ogilvie, P., Jensen, D., & Allan, J. (2000a). *Language models for financial news recommendation.* Paper presented at the Proceedings of the ninth international conference on Information and knowledge management.

Lavrenko, V., Schmill, M., Lawrie, D., Ogilvie, P., Jensen, D., & Allan, J. (2000b). *Mining of concurrent text and time series.* Paper presented at the KDD-2000 Workshop on Text Mining.

LeCun, Y., Bottou, L., Orr, G. B., & Müller, K.-R. (1998). Efficient backprop *Neural networks: Tricks of the trade* (pp. 9-50): Springer.

Lee, A. J., Lin, M.-C., Kao, R.-T., & Chen, K.-T. (2010). *An Effective Clustering Approach to Stock Market Prediction.* Paper presented at the PACIS.

Li, X., Deng, X., Wang, F., & Dong, K. (2010). *Empirical analysis: News impact on stock prices based on news density.* Paper presented at the Data Mining Workshops (ICDMW), 2010 IEEE International Conference on.

Li, X., Wang, C., Dong, J., Wang, F., Deng, X., & Zhu, S. (2011). *Improving stock market prediction by integrating both market news and stock prices.* Paper presented at the

Database and Expert Systems Applications.

Li, X., Xie, H., Wang, R., Cai, Y., Cao, J., Wang, F., . . . Deng, X. (2014). Empirical analysis: stock market prediction via extreme learning machine. *Neural Computing and Applications*, 1-12.

Liang, X. (2005). Impacts of internet stock news on stock markets based on neural networks *Advances in Neural Networks–ISNN 2005* (pp. 897-903): Springer.

Liang, X., & Chen, R.-C. (2005). *Mining Stock News in Cyberworid Based on Natural Language Processing and Neural Networks.* Paper presented at the Neural Networks and Brain, 2005. ICNN&B'05. International Conference on.

Liang, X., Chen, R.-C., He, Y., & Chen, Y. (2013). Associating stock prices with web financial information time series based on support vector regression. *Neurocomputing, 115*, 142-149.

Lin, M.-C., Lee, A. J., Kao, R.-T., & Chen, K.-T. (2011). Stock price movement prediction using representative prototypes of financial reports. *ACM Transactions on Management Information Systems (TMIS), 2*(3), 19.

Liu, B. (2012). Sentiment analysis and opinion mining. *Synthesis Lectures on Human Language Technologies, 5*(1), 1-167.

Lo, A. (2004). The adaptive markets hypothesis: Market efficiency from an evolutionary perspective. *Journal of Portfolio Management, Forthcoming*.

Lo, A. W. (2004). The adaptive markets hypothesis. *The Journal of Portfolio Management, 30*(5), 15-29.

Lo, A. W., MacKinlay, A. C., & Zhang, J. (2002). Econometric models of limit-order executions. *Journal of Financial Economics, 65*(1), 31-71. doi: https://doi.org/10.1016/S0304-405X(02)00134-4

Lo, A. W., Mamaysky, H., & Wang, J. (2002). Foundations of technical analysis: Computational algorithms, statistical inference, and empirical implementation. *The Journal of Finance, 55*(4), 1705-1770.

Lodwich, A., Rangoni, Y., & Breuel, T. (2009). *Evaluation of robustness and performance of early stopping rules with multi layer perceptrons.* Paper presented at the Neural Networks, 2009. IJCNN 2009. International Joint Conference on.

Lu, H.-M., Chen, H., Chen, T.-J., Hung, M.-W., & Li, S.-H. (2010). Financial text mining: supporting decision making using Web 2.0 content. *IEEE Intelligent Systems, 25*(2).

Luss, R., & d'Aspremont, A. (2012). Predicting abnormal returns from news using text classification. *Quantitative Finance*(ahead-of-print), 1-14.

Luss, R., & d'Aspremont, A. (2009). Predicting Abnormal Returns From News Using Text Classification.

Makrehchi, M., Shah, S., & Liao, W. (2013). *Stock Prediction Using Event-based Sentiment Analysis.* Paper presented at the Web Intelligence (WI) and Intelligent Agent Technologies (IAT), 2013 IEEE/WIC/ACM International Joint Conferences on.

Malkiel, B. G., & Fama, E. F. (1970). Efficient Capital Markets: A Review Of Theory And Empirical Work. *The Journal of Finance, 25*(2), 383-417.

Mao, H., Counts, S., & Bollen, J. (2011). Predicting financial markets: Comparing survey, news, twitter and search engine data. *arXiv preprint arXiv:1112.1051*.

Mao, Y., Wei, W., & Wang, B. (2013). *Twitter volume spikes: analysis and application in stock trading.* Paper presented at the Proceedings of the 7th Workshop on Social Network Mining and Analysis.

Mitchell, T. M. (1997). *Machine Learning*: McGraw-Hill, Inc.

Mittal, A., & Goel, A. (2012). Stock Prediction Using Twitter Sentiment Analysis: Stanford. edu. Retrieved on September.

Mittermayer, M.-A. (2004). *Forecasting intraday stock price trends with text mining techniques.* Paper presented at the System Sciences, 2004. Proceedings of the 37th Annual Hawaii International Conference on.

Mittermayer, M.-A., & Knolmayer, G. F. (2006). *NEWSCATS: A news categorization and trading system.* Paper presented at the Data Mining, 2006. ICDM'06. Sixth International Conference on.

Moat, H. S., Curme, C., Stanley, H. E., & Preis, T. (2014). Anticipating Stock Market Movements with Google and Wikipedia *Nonlinear Phenomena in Complex Systems: From Nano to Macro Scale* (pp. 47-59): Springer.

Murphy, J. J. (1999). *Technical analysis of the financial markets: A comprehensive guide to trading methods and applications*: Penguin.

Nardo, M., Petracco-Giudici, M., & Naltsidis, M. (2016). Walking down wall street with a tablet: A survey of stock market predictions using the web. *Journal of Economic Surveys, 30*(2), 356-369.

Neely, C., Weller, P., & Dittmar, R. (1997). Is technical analysis in the foreign exchange market profitable? A genetic programming approach. *Journal of financial and Quantitative Analysis, 32*(4), 405-426.

Nelson, D. M., Pereira, A. C., & de Oliveira, R. A. (2017). *Stock market's price movement prediction with LSTM neural networks.* Paper presented at the Neural Networks (IJCNN), 2017 International Joint Conference on.

Oh, C., & Sheng, O. (2011). *Investigating Predictive Power of Stock Micro Blog Sentiment in*

Forecasting Future Stock Price Directional Movement. Paper presented at the ICIS.

Oliveira, N., Cortez, P., & Areal, N. (2013a). On the Predictability of Stock Market Behavior Using StockTwits Sentiment and Posting Volume *Progress in Artificial Intelligence* (pp. 355-365): Springer.

Oliveira, N., Cortez, P., & Areal, N. (2013b). *Some experiments on modeling stock market behavior using investor sentiment analysis and posting volume from Twitter.* Paper presented at the Proceedings of the 3rd International Conference on Web Intelligence, Mining and Semantics.

Oliveira, N., Cortez, P., & Areal, N. (2017). The impact of microblogging data for stock market prediction: using Twitter to predict returns, volatility, trading volume and survey sentiment indices. *Expert Systems with applications, 73*, 125-144.

Osler, C., & Chang, P. (1995). Head and shoulders: Not just a flaky pattern. *FRB of New York staff report*(4).

Pinto, M. V., & Asnani, K. (2011). Stock Price Prediction Using Quotes and Financial News. *International Journal of Soft Computing, 1*.

Porshnev, A., Redkin, I., & Shevchenko, A. (2013). *Machine Learning in Prediction of Stock Market Indicators Based on Historical Data and Data from Twitter Sentiment Analysis.* Paper presented at the Data Mining Workshops (ICDMW), 2013 IEEE 13th International Conference on.

Prechelt, L. (1998). Automatic early stopping using cross validation: quantifying the criteria. *Neural Networks, 11*(4), 761-767.

Quah, T.-S., & Srinivasan, B. (1999). Improving returns on stock investment through neural network selection. *Expert Systems with applications, 17*(4), 295-301.

Rachlin, G., & Last, M. (2006). Predicting stock trends with time series Data Mining and Web Content Mining *Advances in Web Intelligence and Data Mining* (pp. 181-190): Springer.

Rachlin, G., Last, M., Alberg, D., & Kandel, A. (2007). *ADMIRAL: A data mining based financial trading system.* Paper presented at the Computational Intelligence and Data Mining, 2007. CIDM 2007. IEEE Symposium on.

Rao, T., & Srivastava, S. (2012a). *Analyzing Stock Market Movements Using Twitter Sentiment Analysis.* Paper presented at the Proceedings of the 2012 International Conference on Advances in Social Networks Analysis and Mining (ASONAM 2012).

Rao, T., & Srivastava, S. (2012b). Twitter Sentiment Analysis: How To Hedge Your Bets In The Stock Markets. *arXiv preprint arXiv:1212.1107*.

Refenes, A., Azema-Barac, M., & Zapranis, A. (1993). *Stock ranking: Neural networks vs multiple linear regression.* Paper presented at the Neural Networks, 1993., IEEE International Conference on.

Robertson, C., Geva, S., & Wolff, R. C. (2007). *Can the Content of Public News be used to Forecast Abnormal Stock Market Behaviour?* Paper presented at the Data Mining, 2007. ICDM 2007. Seventh IEEE International Conference on.

Robertson, C. S., Geva, S., & Wolff, R. C. (2007). *News aware volatility forecasting: Is the content of news important?* Paper presented at the Proceedings of the sixth Australasian conference on Data mining and analytics-Volume 70.

Ruiz, E. J., Hristidis, V., Castillo, C., Gionis, A., & Jaimes, A. (2012). *Correlating financial time series with micro-blogging activity.* Paper presented at the Proceedings of the fifth ACM international conference on Web search and data mining.

Schulmeister, S. (2009). Profitability of technical stock trading: Has it moved from daily to

intraday data? *Review of Financial Economics, 18*(4), 190-201.

Schumaker, R. P. (2009). *Analyzing representational schemes of financial news articles.* Paper presented at the The Third China Summer Workshop on Information Systems.

Schumaker, R. P., & Chen, H. (2006). *Textual Analysis of Stock Market Prediction Using Financial News.* Paper presented at the Americas Conference on Information Systems.

Schumaker, R. P., & Chen, H. (2008). Evaluating a news-aware quantitative trader: The effect of momentum and contrarian stock selection strategies. *Journal of the American Society for Information Science and technology, 59*(2), 247-255.

Schumaker, R. P., & Chen, H. (2009a). A quantitative stock prediction system based on financial news. *Information Processing & Management, 45*(5), 571-583.

Schumaker, R. P., & Chen, H. (2009b). Textual analysis of stock market prediction using breaking financial news: The AZFin text system. *ACM Transactions on Information Systems (TOIS), 27*(2), 12.

Schumaker, R. P., & Chen, H. (2010). A discrete stock price prediction engine based on financial news. *Computer, 43*(1), 51-56.

Schumaker, R. P., & Chen, H. (2011). Predicting Stock Price Movement from Financial News Articles. *Information Systems for Global Financial Markets: Emerging Developments and Effects*, 96.

Schumaker, R. P., Zhang, Y., Huang, C.-N., & Chen, H. (2012). Evaluating sentiment in financial news articles. *Decision Support Systems, 53*(3), 458-464.

Shiller, R. J. (2003). From efficient markets theory to behavioral finance. *The Journal of Economic Perspectives, 17*(1), 83-104.

Smailović, J., Grčar, M., Lavrač, N., & Žnidaršič, M. (2013). Predictive Sentiment Analysis of

Tweets: A Stock Market Application *Human-Computer Interaction and Knowledge Discovery in Complex, Unstructured, Big Data* (pp. 77-88): Springer.

Smith, E., Farmer, J. D., Gillemot, L. s., & Krishnamurthy, S. (2003). Statistical theory of the continuous double auction. *Quantitative finance, 3*(6), 481-514.

Son, Y., Noh, D.-j., & Lee, J. (2012). Forecasting trends of high-frequency KOSPI200 index data using learning classifiers. *Expert Systems with Applications*.

Sorensen, E. H., Miller, K. L., & Ooi, C. K. (2000). The decision tree approach to stock selection. *The Journal of Portfolio Management, 27*(1), 42-52.

Sprenger, T. O., Tumasjan, A., Sandner, P. G., & Welpe, I. M. (2013). Tweets and Trades: the Information Content of Stock Microblogs. *European Financial Management*. doi: 10.1111/j.1468-036X.2013.12007.x

Takahashi, S., Takahashi, M., Takahashi, H., & Tsuda, K. (2006). *Analysis of stock price return using textual data and numerical data through text mining.* Paper presented at the Knowledge-Based Intelligent Information and Engineering Systems.

Tang, X., Yang, C., & Zhou, J. (2009). *Stock price forecasting by combining news mining and time series analysis.* Paper presented at the Web Intelligence and Intelligent Agent Technologies, 2009. WI-IAT'09. IEEE/WIC/ACM International Joint Conferences on.

Thanh, H. T., & Meesad, P. (2014). Stock Market Trend Prediction Based on Text Mining of Corporate Web and Time Series Data. *Journal ref: Journal of Advanced Computational Intelligence and Intelligent Informatics, 18*(1), 22-31.

Thomas, J. D. (2003). *News and trading rules.* Carnegie Mellon University.

Thomas, J. D., & Sycara, K. (2000). Integrating genetic algorithms and text learning for financial prediction. *Data Mining with Evolutionary Algorithms*, 72-75.

Vanipriya, C., & Reddy, K. T. (2014). *Indian Stock Market Predictor System.* Paper presented at the ICT and Critical Infrastructure: Proceedings of the 48th Annual Convention of Computer Society of India-Vol II.

Wang, B., Huang, H., & Wang, X. (2012). A novel text mining approach to financial time series forecasting. *Neurocomputing, 83*, 136-145.

Wolfram, M. S. A. (2011). *Modelling the Stock Market using Twitter.* (Master of Science), University of Edinburgh.

Wu, D. D., Zheng, L., & Olson, D. L. (2014). A Decision Support Approach for Online Stock Forum Sentiment Analysis.

Wu, J.-L., Su, C.-C., Yu, L.-C., & Chang, P.-C. (2013). Stock Price Predication using Combinational Features from Sentimental Analysis of Stock News and Technical Analysis of Trading Information. *International Proceedings of Economics Development & Research, 55*.

Wu, M.-C., Lin, S.-Y., & Lin, C.-H. (2006). An effective application of decision tree to stock trading. *Expert Systems with applications, 31*(2), 270-274.

Wüthrich, B., Cho, V., Leung, S., Permunetilleke, D., Sankaran, K., & Zhang, J. (1998). *Daily stock market forecast from textual web data.* Paper presented at the Systems, Man, and Cybernetics, 1998. 1998 IEEE International Conference on.

Wüthrich, B., Permunetilleke, D., Leung, S., Lam, W., Cho, V., & Zhang, J. (1998). Daily prediction of major stock indices from textual www data. *HKIE Transactions, 5*(3), 151-156.

Xie, B., Passonneau, R. J., Wu, L., & Creamer, G. G. (2013). *Semantic frames to predict stock price movement.* Paper presented at the Proceedings of the 51st Annual Meeting of the

Association for Computational Linguistics.

Xiong, R., Nichols, E. P., & Shen, Y. (2015). Deep learning stock volatility with google domestic trends. *arXiv preprint arXiv:1512.04916*.

Xu, F. (2012). *Data Mining in Social Media for Stock Market Prediction.* (Master of Science), Dalhousie University.

Xu, H.-C., Zhang, W., & Liu, Y.-F. (2014). Short-term market reaction after trading halts in Chinese stock market. *Physica A: Statistical Mechanics and its Applications, 401*, 103-111.

Xue, L., Xiong, Y., Zhu, Y., Wu, J., & Chen, Z. (2013). Stock Trend Prediction by Classifying Aggregative Web Topic-Opinion *Advances in Knowledge Discovery and Data Mining* (pp. 173-184): Springer.

Yang, Y., & Pedersen, J. O. (1997). *A comparative study on feature selection in text categorization.* Paper presented at the MACHINE LEARNING-INTERNATIONAL WORKSHOP THEN CONFERENCE-.

Yu, T., Jan, T., Debenham, J., & Simoff, S. (2006). *Classify unexpected news impacts to stock price by incorporating time series analysis into support vector machine.* Paper presented at the Neural Networks, 2006. IJCNN'06. International Joint Conference on.

Zapranis, A., & Tsinaslanidis, P. E. (2011). A novel, rule-based technical pattern identification mechanism: Identifying and evaluating saucers and resistant levels in the US stock market. *Expert Systems with Applications*.

Zhai, Y., Hsu, A., & Halgamuge, S. K. (2007). Combining news and technical indicators in daily stock price trends prediction *Advances in Neural Networks–ISNN 2007* (pp. 1087-1096): Springer.

Zhang, H. ICTCLAS 2016. from http://ictclas.nlpir.org/

Zhang, X., Fuehres, H., & Gloor, P. A. (2011). Predicting stock market indicators through twitter "I hope it is not as bad as I fear". *Procedia-Social and Behavioral Sciences, 26*, 55-62.

Zhang, Y., Swanson, P. E., & Prombutr, W. (2012). Measuring Effects On Stock Returns Of Sentiment Indexes Created From Stock Message Boards. *Journal of Financial Research, 35*(1), 79-114.

Zhou, S., Shi, X., Sun, Y., Qu, W., & Shi, Y. (2013). Stock Market Prediction Using Heat of Related Keywords on Micro Blog. *Journal of Software Engineering and Applications, 6*, 37.

Zhu, M., Philpotts, D., Sparks, R., & Stevenson, M. J. (2011). A hybrid approach to combining CART and logistic regression for stock ranking. *Journal of Portfolio Management, 38*(1), 100.