Application of machine learning techniques for stock market prediction

by

Bin Weng

A dissertation submitted to the Graduate Faculty of
Auburn University
in partial fulfillment of the
requirements for the Degree of
Doctor of Philosophy

Auburn, Alabama
May 6, 2017

Keywords: Machining learning, Feature selection, Dimensional reduction, Visual data mining, Stock market, Social media

Approved by

Fadel Megahed, Chair, Assistant Professor of Industrial and Systems Engineering
John Evans, Professor of Industrial and Systems Engineering
Jorge Valenzuela, Professor of Industrial and Systems Engineering
Aleksandr Vinel, Assistant Professor of Industrial and Systems Engineering

Abstract

Stock market prediction has attracted much attention from academia as well as business. Due to the non-linear, volatile and complex nature of the market, it is quite difficult to predict. As the stock markets grow bigger, more investors pay attention to develop a systematic approach to predict the stock market. Since the stock market is very sensitive to the external information, the performance of previous prediction systems is limited by merely considering the traditional stock data. New forms of collective intelligence have emerged with the rise of the Internet (e.g. Google Trends, Wikipedia, etc.). The changes on these platforms will significantly affect the stock market. In addition, both the financial news sentiment and volumes are believed to have impact on the stock price. In this study, disparate data sources are used to generate a prediction model along with a comparison of different machine learning methods. Besides historical data directly from the stock market, numbers of external data sources are also considered as inputs to the model. The goal of this study is to develop and evaluate a decision making system that could be used to predict stocks' short term movement, trend, and price. We took advantage of the open source API and public economic database which allow us to explore the hidden information among these platforms. The prediction models are compared and evaluated using machine learning techniques, such as neural network, support vector regression and boosted tree. Numbers of case studies are performed to evaluate the performance of the prediction system. From the case studies, several results were obtained: (1) the use of external data sources along with traditional metrics leads to improve the prediction performance; (2) the prediction models benefit from the feature selection and dimensional reduction techniques. (3) The prediction performance dominates the related works. Finally, a decision support system is provided to assist investors in making trading decision on any stocks.

Acknowledgments

First, I would like to express my sincere thanks to my advisor Dr. Fadel Megahed for his continuous support of my Ph.D. study. His guidance helped me in the research of applying machine learning techniques and writing journal papers in a professional way. I would like to mention that it was very difficult to write my first research paper. Dr. Fadel encouraged and motivated me to achieve this goal. The paper finally published in a high ranking journal. In addition, I would like to thank my committee members, Dr. John Evans, Dr. Jorge Valenzuela, and Dr. Alex Vinel. They helped me a lot and provide me insightful comments in my research, even though in my daily life. Especially, I would deeply appreciate to Dr. Vinel for his help after Dr. Fadel moved to Miami University during the last year of my Ph.D. study. Moreover, they enrich my Ph.D. study at Auburn. With special thanks to Dr. James Barth and Dr. Waldyn Martinez for their help in my research. They provided me very insightful comments and significantly enhanced my papers. Also, I would like to thank all my co-authors of my journal papers, Chen Li, Yao-te Tsai, Mohamed Ahmed, Xing Wang, and Lin Lu for their kindly help. I would like thank all my friends who always supported me. Last, I would like to thank my respected parents for supporting and encouraging me all the time. Without them, I could not have finished my Ph.D. study.

Table of Contents

List of Figures

List of Tables

Chapter 1

Problem Description and Significance

Stock market prediction, which has the capacity to reap large profits if done wisely, has attracted much attention from academia and business. Due to the non-linear, volatile and complex nature of the stock market, it is quite difficult to predict. The question remains:"To what extent can the past history of a common stock's price be used to make meaningful predictions concerning the future price of the stock?" (Fama, 1965). In my dissertation, the goal is to examine the important and potential factors/predictors that could drive the stock market and develop a set of models to predict the short-term stock movement and price. Early research on stock market prediction was based on the Efficient Market Hypothesis (EMH) (Fama, 1965) and the random walk theory (Cootner, 1964; Fama, Fisher, Jensen, & Roll, 1969; Fama, 1991, 1995).These early models suggested that the stock price cannot be predicted with >50% accuracy.There has been an increasing number of studies (see e.g.,Malkiel, 2003; Smith, 2003; Nofsinger, 2005; Prechter Jr & Parker, 2007; Bollen, Mao, & Zeng, 2011) that provide evidence contrary to what is suggested by the EMH and random walk hypotheses. These studies show that the stock market can be predicted to some degree and therefore, questioning the EMH's underlying assumptions. Many within the business community also highlight Warren Buffet's ability to consistently beat the S&P index (Kiersz, 2015; Loomis, 2012) as an additional proof that the market can be predicted with an accuracy rate that exceeds the 50% threshold. Data mining,intersection of artificial intelligence, machine learning, statistics and database system, has been extensively studied for the prediction of financial markets. For the target of forecasting the stock movement, previous studies did not exceed an accuracy of 83% (see e.g., Lawrence, 1997,Vu, Chang, Ha, & Collier, 2012). And an efficient prediction system of predicting the short-term price does

not exist. Therefore, different discussions are addressed to improve the accuracy of stock market prediction. In the dissertation, the proposed analytical system can provide:

(A) A prediction system that could be used to detect potential predictors from the data sources of stock market, technical indicators, economic, Internet, and social media

(B) Predict the stock movement trend using disparate data sources

(C) Understand the correlations among U.S. major and sector indices in the stock market and predict their price.

(D) Forecast the short-term price through deploying and comparing different machine learning methods.

(E) A web-based stock market prediction tool, which integrates all the findings in the dissertation, will be developed for the investors to provide them more valuable information for decision making.

1.1 Significance

Stock market is one of the most popular ways to invest in the United State. About 48% Americans invested in the stock market according to the CNBC news in 2015. But many people feel that investing the stock market is risky due to its difficulty to predict. When you invest in a stock, in some case, you might lose more than you invested. To understand the risks and find a better way to predict its trend or price will let you be comfortable to make decisions. Stock market prediction can be divided into two main focusing categories: data and model.

First, the question is: "Which sources of data have the most correlation with the stock market time series?". The significance of the dissertation will be integrating disparate data sources that related to stock market for the prediction. Most online brokers will provide the stock market prediction tool for the investors (e.g. Scottrade, TD Ameritrade, Fidelity

Investments and ETRADE), which has been most popularly used by investors. The data they used to predict is based on the market data. After reviewed related literatures, the significant stock market movements (i.e. spikes) over short horizons are often driven by investor perceptions of a certain stock based on information collected from disparate data sources.Various outside influences could have a big effect on the stock movement or price. For example, the economic factors could impact the stock market's performance, such as inflation and deflation, interest rates, and unemployment rate. To understand the influence of these various economic factors will be beneficial to the investors. Online data is also an important data source for stock market prediction. As an illustration, on April 23, 2013, at 1:07 p.m., Eastern Time, a tweet from the Associated Press (AP) account stated Breaking: Two Explosions in the White House and Barack Obama is injured (Megahed & Jones-Farmer, 2015). The fraudulent tweet, originated from the hacked AP Twitter account led to an immediate drop in the Dow Jones Industrial Average (DJIA). Although the DJIA quickly recovered following an AP retraction and a White House press release, this example illustrates the immediate and dramatic effects of perception/news on stock prices. In addition, Moat et al. (2013) observed that the frequency of views of Wikipedia's financially-related pages can be an early indicator of stock market moves. Changes in macroeconomic factors can have a profound impact on the stock market.

Second, the question is: "Which technological model is best at predicting the stock movement?" The significance of the dissertation will be applying machine learning techniques to develop the stock market prediction models in order to improve the prediction performance. The previous predictions did not exceed an accuracy of 83% (see Table 5 in Nassirtoussi, Aghabozorgi, Wah, & Ngo, 2014, which summarizes the outcomes of 24 related works). A higher prediction model is needed for helping investors make decisions and get significant profit from stock market. As a review, ongoing research is focused on utilizing machine learning to predict the spikes and returns of the stock market (Hassan, Nath, & Kirley, 2007; Dase & Pawar, 2010; Bollen et al., 2011; Guresen, Kayakutlu, & Daim, 2011;

Qian & Rasheed, 2007; Lai, Fan, Huang, & Chang, 2009; Atsalakis & Valavanis, 2009; Yang, Chan, & King, 2002; K.-j. Kim, 2003; Schumaker & Chen, 2009; Nassirtoussi et al., 2014). Also, according to the news stated by Bloomberg.com on 2015, Ray Dalio Ray Dalios $165B Bridgewater Associates will start a new artificial intelligence unit to usepredictive analysis-for trades. As the goal of developing a higher accuracy stock prediction model, different technological models will be compared using disparate data sources described above.

1.2 Research Objectives

The main objective of the dissertation is to examine the important and potential factors/predictors that could impact the stock market from disparate data sources and develop a system for the investors and researchers in stock prediction. The dissertation is divided into three proposed research. The first step is to predict the stock market movement using disparate data sources from online news, Internet reference (Wikipedia), technical indicators and traditional market data. The second step is to examine the impact of economic factors on the stock market indices and develop a stock price prediction model using the selected macroeconomic factors. Finally, we are going to develop a novel ensemble model for short-term stock price prediction. In summarize, the objectives of the dissertation are shown below:

(A) Detect the potential factors (e.g. economic, online news, social media) that could impact the stock market and acquire the data from disparate data sources, such as Yahoo Finance, Wikipedia, Google Trends.

(B) Study the importance of factors detected above. Even though various external factors could affect the stock market, it is important to find the most influenced ones in order to improve the accuracy and reduce overfitting for the prediction model.

(C) Compare the performance of different machine learning models and develop a higher accuracy prediction model using the selected important factors from disparate data sources.

(D) Evaluate the prediction models for different time period forecasting and provide decision making suggestions for the investors.

(E) Develop a stock prediction system to assist investors on making decisions through predicting stocks/indices/commodities of their choice.

1.3 Dissertation Layout

This dissertation is organized as follows: Chapter 2 developed a financial expert system to predict movements in the one-day ahead stock price/volume. It should be noted that Chapter 2 is a paper published in 2017 in the Journal of Expert Systems With Applications. Chapter 3 examines the utility of using macroeconomic factors and ensemble models in predicting the one-month ahead price of major U.S. stock and sector indices. Chapter 3 is a paper that under reviewed by the Journal of Decision Support Systems. Chapter 4 proposes a data-driven approach to predict the short-term stock price for different periods using disparate data sources. This chapter is a paper that under reviewed by the Journal of Decision Support Systems. Chapter 5 summarizes the contribution, implementation, and limitation of this dissertation and make some recommendations for the direction of future work.

Chapter 2

Stock Market One-Day Ahead Movement Prediction

Using Disparate Data Sources

2.1 Abstract

Expert systems have traditionally relied on a knowledge database and an artificial intelligence engine to make decisions and/or solve problems that typically required human experts. Traditionally, the inputs/rules for the knowledge database are captured through structured interviews and prolonged observations. With advances in machine learning, some of these rules are now computer-driven. There are several commercial financial expert systems that can be used for trading on the stock exchange. However, their predictions are somewhat limited since they primarily rely on time-series analysis of the market. With the rise of the Internet, new forms of collective intelligence (e.g. Google and Wikipedia) have emerged, representing a new generation of "crowd-sourced" knowledge bases. They collate information on publicly traded companies, while capturing web traffic statistics that reflect the public's collective interest. Google and Wikipedia have become important "knowledge bases" for investors. In this research, we hypothesize that combining disparate online data sources with traditional time-series and technical indicators for a stock can provide a more effective and intelligent daily trading expert system. Our proposed system will allow investors to predict the next-day movement for a particular stock or index, which is important for daily traders. Three machine learning models, decision trees, neural networks and support vector machines, serve as the basis for our "inference engine". To build these models, a list of potential predictors from these data-sources have been generated. The list includes both variables (i.e. data in raw form) and features (e.g. summary statistics and predictors combining multiple variables) selected from these variables. To evaluate the performance of our

expert system, we present a case study based on the AAPL (Apple NASDAQ) stock. Our expert system had an 85% accuracy in predicting the next-day AAPL stock movement, which outperforms the reported rates in the literature. Our results suggest that: (a) the knowledge base of financial expert systems can benefit from data captured from nontraditional "experts" like Google and Wikipedia; (b) diversifying the knowledge base by combining data from disparate sources can help improve the performance of financial expert systems; and (c) the use of simple machine learning models for inference and rule generation is appropriate with our rich knowledge database. Finally, an intelligent decision making tool is provided to assist investors in making trading decisions on any stock, commodity or index.

2.2 Introduction

Stock market prediction has attracted much attention from both academia and business. The question remains: "To what extent can the past history of a common stock's price be used to make meaningful predictions concerning the future price of the stock?" (Fama, 1965). Early research on stock market prediction was based on the Efficient Market Hypothesis (EMH) (Fama, 1965) and the random walk theory (Cootner, 1964; Fama et al., 1969; Fama, 1991, 1995). These early models suggested that stock prices cannot be predicted since they are driven by new information (news) rather than present/past prices. Thus, stock market prices will follow a random walk and their prediction accuracy cannot exceed 50% (Bollen et al., 2011).

There has been an increasing number of studies (see e.g.,Malkiel, 2003; Smith, 2003; Nofsinger, 2005; Prechter Jr & Parker, 2007; Bollen et al., 2011) that provide evidence contrary to what is suggested by the EMH and random walk hypotheses. These studies show that the stock market can be predicted to some degree and therefore, questioning the EMH's underlying assumptions. Many within the business community also view Warren Buffet's ability to consistently beat the S&P index (Kiersz, 2015; Loomis, 2012) as a practical indicator that the market can be predicted.

The significant stock market movements (i.e. spikes) over short horizons cannot also be explained by the EMH. These spikes are often driven by investor perceptions of a certain stock based on information (news) collected from disparate data sources. As an illustration, on April 23, 2013, at 1:07 p.m., Eastern Time, a tweet from the Associated Press (AP) account stated "Breaking: Two Explosions in the White House and Barack Obama is injured" (Megahed & Jones-Farmer, 2015). The fraudulent tweet, which originated from the hacked AP Twitter account led to an immediate drop in the Dow Jones Industrial Average (DJIA). Although the DJIA quickly recovered following an AP retraction and a White House press release, this example illustrates the immediate and dramatic effects of perception/news on stock prices.

While the news may be unpredictable, some recent literature suggests that early indicators can be extracted from online sources (e.g., Google Trends and blogs) to predict changes in various economic indicators. For example, Google search queries have been shown to provide early indicators of disease infection rates and consumer spending (Choi & Varian, 2012). Schumaker and Chen (2009) showed that breaking financial news can be used to predict stock market movements. Bollen et al. (2011) used measurements of collective mood states derived from large-scale Twitter feeds to predict the daily up and down changes in the DJIA. In addition, Moat et al. (2013) observed that the frequency of views of Wikipedia's financially-related pages can be an early indicator of stock market moves. The authors hypothesized that investors may be using such pages as a part of their decision making process. This work was extended in Preis, Moat, and Stanley (2013) to include data from the number of relevant searches from Google Trends, and model the effect of search volume on trading behavior. Note that Mao, Counts, and Bollen (2011) indicated that search and usage are more predictive than survey sentiment indicators.

From an expert systems perspective, the stock market prediction problem can be divided into two components: (1) what information and predictors need to be tracked as a part of our "knowledge base"; and (2) what artificial intelligence (AI) algorithms can be used for

effective rule generation and predictions. The literature discussed in the previous paragraph indicate that online sources that capture the "collective intelligence" of investors should be an integral component of a financial expert system's knowledge base. It is important to note that these online sources are not typically used in financial expert systems. Instead, the knowledge base of such systems typically rely on the historical prices of a stock and/or technical indicators extracted from a time-series analysis of stock prices (Kimoto, Asakawa, Yoda, & Takeoka, 1990; Lee & Jo, 1999; K.-j. Kim & Han, 2000; K.-j. Kim, 2003; Hassan et al., 2007; Qian & Rasheed, 2007; Lin, Yang, & Song, 2011; Chourmouziadis & Chatzoglou, 2016). We hypothesize that combining the expert's knowledge from online sources with features extracted from the price and technical indicators will offer a more accurate representation of the dynamics that affect a stock's price and its movement. Since these data sources were never combined in the context of financial expert systems, it is important to examine which AI algorithms are the most effective in translating the knowledge base into accurate predictions. Table 2.1 categorizes financial expert systems used for stock movement prediction based on their "knowledge base" and the AI approach used. From Table 2.1, it is clear that the all those papers relied on a single source for the knowledge base. The reader should note that there is a limited number of expert systems (e.g., Bollen et al., 2011) that combined traditional sources with crowd-sourced experts' data; however, they are not included in our table since they predicted price (i.e. a continuous outcome instead of our binary outcome). The integration of diverse data sources can improve the knowledge base (see Alavi & Leidner, 2001; Hendler, 2014 for a detailed discussion) and thus, improving the performance of the expert system.

Based on the insights from Table 2.1 and the discussion above, we outline a novel methodology to predict the future movements in the value of securities after tapping data from disparate sources, including: (a) the number of page visits to pertinent Wikipedia pages; (b) the amount of online content produced on a particular day about a company, the stock of which is publicly traded; and (c) commonly used technical indicators and company

Table 2.1: A review of financial expert systems that are used in stock movement prediction. ANN, GA, SVM and DT correspond to artificial neural network, genetic algorithm, decision tree and support vector machine, respectively.

Paper	Sources for Knowledge Base			AI Approach
	Traditional	Crowd-sourcing	News	
Kimoto et al. (1990)	✓			ANN
Lee and Jo (1999)	✓			Time Series
K.-j. Kim and Han (2000)	✓			ANN, GA
K.-j. Kim (2003)	✓			SVM
Qian and Rasheed (2007)	✓			ANN, DT
S.-T. Li and Kuo (2008)	✓			ANN
Schumaker and Chen (2009)			✓	SVM
Vu et al. (2012)		✓		DT
M.-Y. Chen, Chen, Fan, and Huang (2013)	✓			ANN
Adebiyi, Adewumi, and Ayo (2014)	✓			ANN, ARIMA
Nguyen, Shirai, and Velcin (2015)		✓		SVM
Shynkevich, McGinnity, Coleman, and Belatreche (2015)			✓	ANN, SVM
Chourmouziadis and Chatzoglou (2016)	✓			Fuzzy System
Our Financial Expert System	✓	✓	✓	ANN, SVM, DT

value indicators in stock value prediction. In the AI component of our expert system, we compare the performance of ANN, SVM and DT for stock movement prediction. We have chosen these three specific approaches since: (i) neural networks have been widely deployed in intelligent trading systems (Kimoto et al., 1990; S.-T. Li & Kuo, 2008; Guresen et al., 2011; Bollen et al., 2011); (ii) SVM was successfully used by K.-j. Kim and Han (2000) and Schumaker and Chen (2009); and (iii) decision trees have been effectively used in crowd-sourced expert systems (Vu et al., 2012). In those papers, the authors reported that these AI models outperformed the more traditional approaches. However, it is unclear whether: (1) such results will hold for our predictions since our knowledge base is more diverse, and (2) the results will hold when predicting different stocks and indices. Thus, our expert system will evaluate the performance of these models and select the best approach for a given prediction problem.

To demonstrate the utility of our system, we predict the one-day ahead movements in AAPL stocks over a three year period. Based on our case study, we show that the combination of online data sources with traditional technical indicators provide a higher predictive power than any of these sources alone. The remainder of the paper is organized as follows. In Section 2.3, we present a detailed description of the methodology we used

to extract the data from the online sources, the variable selection techniques employed, and the corresponding predictive models. In Section 2.4, we highlight the main results and offer our perspective on their importance/interpretation. Our concluding remarks and recommendations for future work are provided in Section 2.5. In Appendices I-III, we explain how Google News data was captured, present the formulas for our generated features, and define the predictors identified from our variable selection steps. We also present a copy of our full dataset, code and prediction tool at `https://github.com/binweng/ShinyStock`.

2.3 Methods

To predict stock movements, we propose a data-driven approach that consists of three main phases, as shown in Figure 2.1. In Phase I, we scrape four sets of data from online resources. These datasets include: (a) publicly available market information on stocks, including opening/closing prices, trade volume, NASDAQ and the DJIA indices, etc.; (b) commonly used technical indicators that reflect price variation over time; (c) daily counts of Google News on the stocks of interest; and (d) the number of unique visitors for pertinent Wikipedia pages per day. We also populated additional features (i.e. summary statistics) in an attempt to uncover more significant predictors for stock movement. In Phase II, we use variable selection methods to select a subset of predictors that provide the most predictive power/accuracy. Then, in Phase III, we utilize three AI techniques to predict stock movement. These models are compared and evaluated based on a 10-fold cross validation sample using the area under the operating characteristics curve (AUC) and seven other metrics. Based on the evaluation, we select an appropriate model for real-time stock market prediction. We present the details for each of the phases in the subsections below.

2.3.1 Data Acquisition and Feature Generation for Our "Knowledge Base"

In this paper, we focus on predicting the AAPL, Apple NASDAQ, stock movement based on a 37 month-period from May 1, 2012 to June 1, 2015. There are four datasets that were

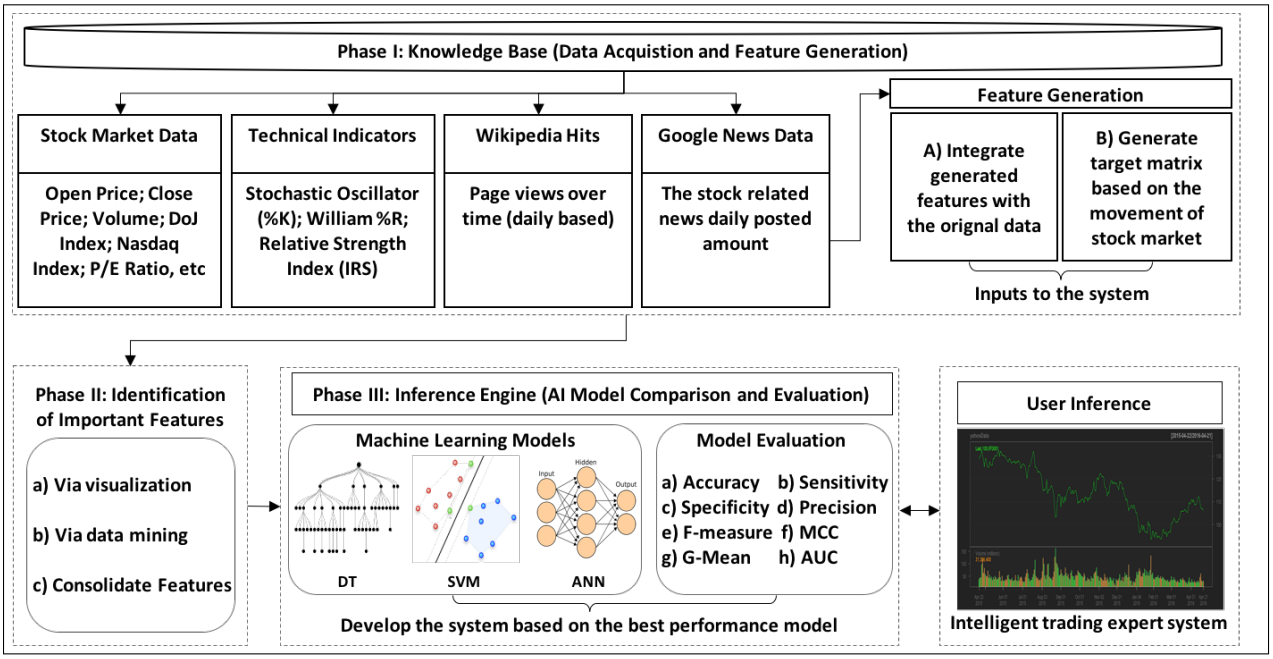


Figure 2.1: An overview of the proposed method

obtained, preprocessed and merged in Phase I. First, we obtain publicly available market data on AAPL using the Yahoo Finance website. We considered the following common predictors of stock prices (see e.g., Y.-F. Wang, 2002; Lee & Jo, 1999; S.-T. Li & Kuo, 2008; Jasemi, Kimiagari, & Memariani, 2011): the daily opening and closing prices, daily high/low, and volume of trades of the AAPL stock. In addition, we included the day-to-day movements in the DJIA and NASDAQ composite indices as indirect measures of risk that the AAPL stock is subject to due to the general market movements. We also used the price to earnings ratio (P/E) as an estimate for the fundamental health of the company (Gabrielsson & Johansson, 2015).

The second set of predictors is comprised of three indicators that are used in technical analysis. Technical analysis is used to forecast future stock prices by studying historical prices and volumes (Chourmouziadis & Chatzoglou, 2016). Since all information is reflected in stock prices, it is sufficient to study specific technical indicators (created by mathematical formula) to predict price fluctuations and evaluate the strength of the prevailing trend (Bao & Yang, 2008). In this paper, we consider three technical indicators:

(A) Stochastic Oscillator (%K), developed by George C. Lane as a momentum indicator that can warn of the strength or weakness of the market. When the market is trending upwards, it tries to measure when the closing price would get close to the lowest price in a given period. On the other hand, when the market is trending downwards, it estimates when the closing price would get close to the highest price in the given period. For additional details on the %K and its calculation, the reader is referred to: Bao and Yang (2008) and Lin et al. (2011).

(B) The Larry William (LW) % R Indicator - It is a momentum indicator that facilitates the spotting of overbought and oversold levels. For its calculation, refer to K.-j. Kim and Han (2000).

(C) The Relative Strength Index (RSI)- Similar to the LW %R, it compares the magnitude of recent gains to recent losses in an attempt to determine overbought and oversold conditions of an asset. RSI ranges from 0 to 100. In practice, investors sell if its value is ≥ 80 and buy if it is ≤ 20. For more details, see Bao and Yang (2008) and Lin et al. (2011).

The reader should note that the values for these three technical indicators were calculated based on the market price data obtained from Yahoo Finance.

In the third data source, we scrape the amount of daily online content produced about a company, and its products/services. In this paper, we obtain a count for aggregated news and blogs based on the daily count of content on Google News. We detail this step in Appendix I. The fourth and final data source is based on the Wikipedia page view counts of terms related to Apple stock (AAPL, Apple Inc., iPhone, iPad, Macbook, and Mac OS). We queried the daily visits for these pages from `www.wikipediatrends.com`. A graphical summary of the second and third set of predictors is provided in Figure 2.2.

To enhance the performance of the predictive models, we generate some additional features from the four predictor sets. We incorporate some of the underlying principles behind technical analysis (see e.g., Bao & Yang, 2008) to generate our feature set. Therefore,

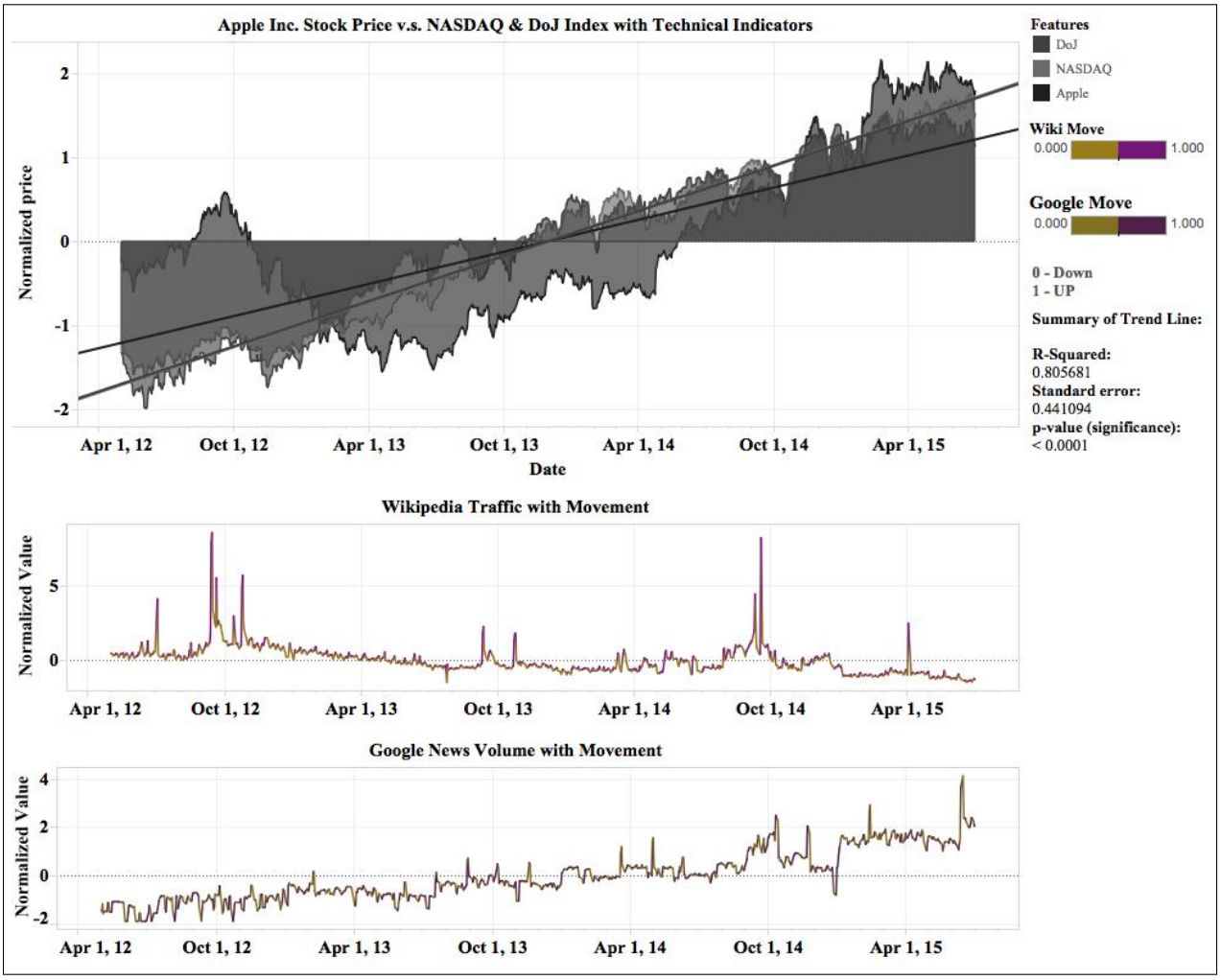


Figure 2.2: A visual summary of the main predictors from the four data sources. An interactive version of this plot is available at: https://goo.gl/fZSQEy . Note that we rescaled the variables (by subtracting the mean and dividing by the standard deviation) to facilitate the visualization of the data.

our generated features include: Wikipedia Momentum, Wikipedia Rate of Change, Google Momentum, Google Relative Strength Index, and three moving averages of stock prices (where $n = 3, 5$, and 10, respectively). For the sake of completion, we explain how each of these features are calculated in Appendix II.

2.3.2 Variable/Feature Selection

The end goal of this phase is to have the data processed for the artificial intelligence models. This phase is comprised of two steps. First, we define different one-day-ahead

outcomes (hereafter targets). Then, we use recursive feature elimination (RFE) to select the features/variables that offer the highest predictive power.

There are several one-day-ahead outcomes that can be of interest to investors. We examine five different targets. These targets are defined in Table 2.2. Target 1 compares the opening stock price of day $i+1$ with the closing price of the previous trading day. In Target 2, we compare the opening stock price of day $i+1$ with the opening price of the previous trading day. Targets 3 and 4 follow a similar logic with the closing price used for day $i+1$ instead of the opening price. In Target 5, we examine the differences in trade volume between day $i+1$ and day i. It is important to note that we only calculate these targets for the AAPL stock as a case study. In addition, we have transformed all targets to a binary variable where $0 \rightarrow$ no increase in target, and $1 \rightarrow$ an increase in the target value from the previous day.

Table 2.2: One-day-ahead targets used in this paper

Target	**Formula**
Target 1	$Open(i+1) - Close(i)$
Target 2	$Open(i+1) - Open(i)$
Target 3	$Close(i+1) - Close(i)$
Target 4	$Close(i+1) - Open(i)$
Target 5	$Trade\ Volume(i+1) - Trade\ Volume(i)$

In Step 2, we selected the significant features using the SVM recursive feature elimination (RFE) algorithm. RFE is implemented through backwards selection of predictors based on predictor importance ranking. The predictors are ranked and the less important ones are sequentially eliminated prior to deploying our three predictive models. The goal of this step is to find a subset of predictors that can result in accurate predictions without overfitting. It should be noted that we used the SVM-RFE algorithm for each target. Thus, we have obtained different predictor sets for each target. These predictors are presented in the Table 2.3 in Section 2.4. For more details on how RFE can be deployed using open-source

programming languages, the readers are referred to the **R** Package Caret (Kuhn, 2008) and to the sklearn.feature_selection module in Python. (Scikit-Learn-Developers, 2014).

2.3.3 The Inference Engine: AI Model Comparison and Evaluation

In this phase, we compare the effectiveness of artificial neural networks (ANN), decision trees (DT), and support vector machines (SVM) for predicting movements in the AAPL stock based on the predictors identified in Subsection 2.3.2. In the paragraphs below, we first introduce how we used a 10-fold cross validation approach to minimize the sampling bias. Then, we provide a very short overview of the three classification approaches, and introduce the performance evaluation metrics used to identify the most suitable approach. The reader should note that, in this paper, we deploy the described methodology for each of the five targets. Hereafter, we use the term dataset to reflect each set of features/variables with its associated target for the AAPL stock over the 37 months of the study.

The k-fold cross-validation approach is used to minimize the bias associated with the random sampling of the training and test data samples (Kohavi, 1995). The entire dataset is randomly split into k mutually exclusive subsets of approximately equal size. The prediction model is tested k times by using the test sets. The estimation of the k-fold cross validation for the overall performance criteria is calculated as the average of the k individual performances as shown in Dag, Topuz, Oztekin, Bulur, and Megahed (2016a). In our analysis, we use the stratified 10-fold cross validation approach to estimate the performance of the different classification models. Our choice for $k = 10$ is based on literature results (see e.g., Kohavi, 1995; Dag et al., 2016a) that show that 10-folds provide an ideal balance between performance and the time required to run the folds.

ANNs are widely employed in a wide variety of computational data analytics problems that include classification, regression and pattern recognition. In the context of stock market prediction, ANNs have been extensively applied in predicting stocks and indices at different markets (see K.-j. Kim & Han, 2000; Hassan et al., 2007; Atsalakis & Valavanis, 2009; Zhang

& Wu, 2009; Dase & Pawar, 2010; Bollen et al., 2011; Guresen et al., 2011, and the references within). We assume that the reader is familiar with ANNs and their construction (otherwise, refer to Hastie, Tibshirani, and Friedman (2011)). In this paper, we use the sigmoid function as the activation function for our ANN. We have also used the Multi-layer Perceptron (MLP) learning model with a back-propagation algorithm due to its superior performance to the radial basis function (RBF) in our preliminary analysis.

Decision trees are widely used in several data mining and stock market prediction problems (Qian & Rasheed, 2007; Lai et al., 2009; Atsalakis & Valavanis, 2009) since they are very easy to interpret. The modeling procedure starts with splitting the dataset into several subsets each of which consists of more or fewer homogeneous states of the target variable (Breiman, Friedman, Stone, & Olshen, 1984). Then the impacts of each independent variable on the target variable are measured. This procedure takes place successively until the decision tree reaches a stable state. Popular decision tree algorithms include Quinlan's ID3, C4.5, C5 (Quinlan, 1986, 2014) and C&RT (Breiman et al., 1984). In our data analysis, the C5 algorithm was used since it: a) is computationally efficient; and b) has outperformed the other methods examined in our preliminary analysis.

Similar to the previous two other classification approaches, SVM is a popular approach for stock market prediction (Yang et al., 2002; K.-j. Kim, 2003; Schumaker & Chen, 2009; Nassirtoussi et al., 2014). More interestingly, SVMs are favored in applications where text mining is used for market prediction (Nassirtoussi et al., 2014). SVMs can be used for both linearly and non-linearly separable datasets. When the data is linearly separable, SVMs construct a hyperplane on the feature space to distinguish the training tuples in the data such that the margin between the support vectors is maximized. For nonlinear cases, the data is typically mapped into a higher-dimensional space so that the new dataset in higher-dimension becomes linearly separable. This problem can be handled efficiently by using a Kernel function (see Han, Kamber, & Pei, 2011 for more details). Based on our

preliminary analysis, we have used the Radial Basis Function (RBF) Kernel function in our SVM classification algorithm since it has resulted in the best performance.

To evaluate the performance of the three classification methods, we present eight commonly used metrics in the literature: a) accuracy, b) area under the receiver operating characteristic curve (AUC), c) F-measure, d) G-mean, e) MCC, f) precision, g) sensitivity, and h) specificity. In addition, we provide our code and the confusion matrix for the sake of completion. Our selected measures are all suitable for our binary classification problem. For details on how any of the above metrics can be calculated, we refer the reader to Han et al. (2011), and Hastie et al. (2011). We use the AUC as our primary evaluation metric for the reasons explained in Dag et al. (2016a).

2.4 Results and Discussion

In this section, we first highlight the results from the variable/feature selection phase of our methodology. Then, we present the results from the prediction accuracy of our expert system with respect to the five potential targets. This is followed by some preliminary analysis to evaluate the impact of the information attained from the five different data sources on our prediction power. For the sake of completion and to allow for the replication of our results, we present our code and a detailed tabular view of our results as supplementary documents to this manuscript.

2.4.1 Variable/Feature Selection

As mentioned in Section 2.3.2, the end goal of this phase is to prepare the data for the three machine learning models. Here, we employed the SVM RFE model five times (once for each target). This resulted in five different sets of twenty variables/features that offer the most predictive power for each of the five respective targets. We list these sets in Table 2.3. There are several additional observations to be made from Table 2.3:

(A) For any of the five targets, the selected variables/features span all predictor sets. This implies that there are non-redundant, useful information that can be captured from each data source.

(B) The previous day's closing, opening and high prices were significant predictors for all five targets. The previous day's low price is a strong predictor for four of the five targets (with the exception of Target 3).

(C) The Price to Earnings (P/E) Ratio is predictive for the four price targets, but not for Target 5 (i.e., trade volume target). In our opinion, this makes sense since the P/E Ratio measures the current share price relative to the per-share earnings. Thus, it may not be suited for predicting trade volume since it does not capture any movements.

(D) Target 5 had the highest number of Google features of 10. This was a somewhat expected result since Google Trends should reflect interest more than price fluctuations. The number of Google features selected for any of the other targets varied between 4 and 6.

(E) Perhaps the most important observation has to do with the order of the variables/features selected. We have arranged the items in a descending order (left to right and then to the next row). For all targets, variables selected from the first set of predictors were the most significant predictors. They were followed by one or more technical indicators. Then, the list would include several Wikipedia features, which were followed by some Google News features. The final grouping included a mixture of technical indicators and Google/Wikipedia features.

Note that we provide the definition for each of the features listed in Table 2.3 in Appendix III.

Table 2.3: The twenty most predictive variables/features for each target

Target	**Variables/Features Selected**				
Target 1	Close	Open	High	Low	P/E Ratio
	Wiki_3_day_disparity	Wiki_5_day_disparity	Wiki_10_day_disparity	Wiki_Momentum_1	Wiki_ROC
	Google_MA_5	Google_EMA_3	Google_3_Day_disparity	Google_5_day_disparity	RSI
	Stochastic Oscillator (%K)	Wiki_RSI	Google_MA_4	William %R	Google_MA_3
Target 2	Close	Open	High	Low	P/E Ratio
	Wiki_5_day_disparity	Wiki_Move	Wiki_MA3_Move	Wiki_EMA5_Move	Wiki_5day_disparity_Move
	Google_EMA5_Move	Google_3day_disparity_Move	Google_ROC_Move	Google_RSI_Move	Wiki_3_day_disparity
	Stochastic Oscillator (%K)	RSI_Move	Wiki_RSI_Move	Google_MA_6	Google_Move
Target 3	Close	Open	High	P/E Ratio	Stochastic_Move
	Wiki_Monentum_1	Wiki_Move	Wiki_MA3_Move	Wiki_EMA5_Move	Wiki_ROC_Move
	Google_EMA5_Move	Google_3day_disparity_Move	Google_ROC_Move	Google_RSI_Move	Wiki_10_day_disparity
	RSI_Move	Wiki_RSI_Move	Wiki_3_day_disparity	Google_Move	Google_MA5_Move
Target 4	Close	Open	High	Low	P/E Ratio
	RSI_Move	Wiki_10_day_Disparity	Wiki_Move	Wiki_MA3_Move	Wiki_EMA5_Move
	Google_Move	Google_3day_disparity_Move	Google_ROC_Move	Google_RSI_Move	William %R
	Stochastic Oscillator (%K)	Stochastic_Move	Wiki_3day_disparity_Move	Wiki_ROC_Move	Wiki_RSI_Move
Target 5	Close	Open	High	Low	William %R
	Wiki_Monentum_1	Wiki_RSI	Google_MA_2	Google_MA_3	Google_MA_4
	Google_MA_9	Google_3_day_disparity	Google_5_day_disparity	Google_10_day_disparity	Wiki_10_day_disparity
	Wiki_3_day_disparity	Wiki_5_day_disparity	Google_MA_6	Google_MA_7	Google_MA_8

2.4.2 Predictive Modeling Outcomes

As explained in Section 2.3.3, we use the AUC as the primary evaluation criterion to evaluate the performance of the ANN, DT, and SVM models in predicting the five different day-ahead outcomes (while presenting the 7 other metrics for completion). In Figure 2.3, we present the best-case, worst-case and the mean performance of the three machine learning models for each of the five targets. Note that the best-case, worst-case, and mean performances are determined based on the 10-fold cross validation step of our approach. The reader is encouraged to visit the interactive version of this plot at `https://goo.gl/L06FSA`. Based on Figure 2.3, there are several interesting observations that can be made:

(A) Based on the AUC metric, SVM outperforms the ANN model for all five targets, DT outperforms the ANN model for Targets 2-4, DT outpeforms the SVM model in predicting Targets 2-3 (while having a similar performance in Target 4), and the DT model failed to predict one of the classes for both Targets 1 and 5.

(B) For Targets 2-4, the recommended models have an AUC value greater than 0.89. The AUC is the probability that the model will rank a randomly chosen positive instance (i.e. increase in price) higher than a randomly chosen negative one (i.e., decrease in stock price).

(C) The acquired data may not be capturing the underlying factor's for changes in trade volume (i.e., Target 5). The DT model could not predict decreases in trade volume (i.e., all its predictions were "1"s), and the ANN has a similar prediction to that of a random predictor (i.e., flipping a coin). The SVM had a somewhat reasonable mean AUC value of 0.632.

(D) Based on the eight evaluation metrics' values, our disparate data sources and machine learning models can best predict Target 2. Recall that Target 2 compares next day's opening price with today's opening price. This is a somewhat surprising result since we expected Target 1 to have the best results.

(E) Perhaps more importantly, our results (especially for Target 2) are more accurate than those typically reported in the literature. Our model resulted in $\approx 85\%$ accuracy/hit ratio with an average AUC of > 0.874 for SVM and DT for Target 2. In the literature, the previous predictions did not exceed an accuracy of 83% (see Table 5 in Nassirtoussi et al. (2014), which summarizes the outcomes of 24 text-mining-based financial expert systems).

(F) Building on the previous result, it is also clear that the addition of data from disparate data sources have resulted in improved accuracy. For example, K.-j. Kim (2003) used the SVM model with only technical indicators as inputs, and obtained an accuracy rate of 65% accuracy for their best performance model. Our > 20 % accuracy improvement (when SVM or DT are used) is significant and justifies the effort needed to include new data sources.

From the above discussion, we have established that we can predict Targets 1-4 reasonably well through the deployment of an adequate machine learning model with inputs identified in Table 2.3. To formally understand the usefulness of the four disparate data sources and our generated features, we consider several scenarios that are summarized in Table 2.4. Note that Scenarios 1-2 involve the data sources most commonly used in traditional stock market prediction. Scenarios 3-4 build on Scenario 2 with the additional of one online data source. In Scenarios 5-6, we add the generated features to Scenarios 3 and 4, respectively. Scenario 7 include all five data sources.

As an example, consider the SVM model for Target 2. Let us examine how the inclusion of data sources, according to the seven scenarios presented in Table 2.4, impact the eight evaluation metrics. We present the results in Table 2.5. From the results, it is clear that the best performance is obtained when all data sources are included. In addition, by comparing S5 and S6 (or alternatively S3 and S4), one can see that the Wikipedia data is more informative than the Google Data for Target 2. Note that we consider the results presented in Table 2.5 as a formal way to evaluate our observation in Point (F) above.

Table 2.4: Examining the impact of the non-traditional data sources

Scenario #	Description
1	Market data
2	Market data, technical indicators
3	Market data, technical indicators, Wikipedia Traffic
4	Market data, technical indicators, Google news counts
5	Market data, technical indicators, Wikipedia Traffic, generated features
6	Market data, technical indicators, Google news counts, generated features
7	Market data, technical indicators, Wikipedia traffic, Google news counts, and generated features

Table 2.5: Comparison of seven scenarios using eight evaluation metrics

Scenario	Accuracy	Sensitivity	Specificity	Precision	F-measure	MCC	G-Mean	AUC
S1	0.565	0.577	0.551	0.601	0.589	0.127	0.564	0.634
S2	0.616	0.618	0.614	0.648	0.633	0.232	0.616	0.711
S3	0.618	0.634	0.601	0.629	0.632	0.235	0.617	0.703
S4	0.618	0.639	0.595	0.639	0.639	0.233	0.616	0.708
S5	0.822	0.835	0.807	0.824	0.830	0.642	0.821	0.800
S6	0.813	0.821	0.804	0.805	0.813	0.625	0.813	0.856
S7	0.858	0.838	0.879	0.873	0.854	0.719	0.858	0.874

2.5 Conclusion and Future Work

2.5.1 An Overview of the Impacts and Contributions of our Proposed Expert System

In this paper, we developed a financial expert system to predict movements in the one-day ahead stock price/volume. To construct our "knowledge base", we scrapped four different data sets: (i) historical stock market data, (ii) commonly used technical indicators, (iii) Wikipedia traffic statistics pertaining to the company's pages (i.e. general company profile, stock page, and pages pertaining to the company's main products), and (d) Google News. Since these data sources were never used in combination in an expert system, we generated features from the two online sources to further improve our knowledge base. Our AI framework consisted of two major phases: (1) variable/ feature selection, which helps

improve the performance of our AI algorithms by reducing the dimensions of the data without the loss of information; and (2)the incorporation of ANN, SVM and DT for prediction, which allows us to select the "best" model for a given target and stock. We provide a web-based user interface (see `https://github.com/binweng/ShinyStock`) to promote the adoption of our expert system by investors and financial planners.

From an Expert and Intelligent Systems research perspective, our system is innovative and novel. Specifically, the related literature on stock movement prediction (shown in Table 2.1) primarily considered the use of traditional data sources (i.e. market data and technical indicators) and none, to our knowledge, combined multiple data sources. Our system utilizes disparate data sources in an attempt to have a more holistic representation of the factors and conditions that precede stock movement. The proposed expert system is tested using a large and feature-rich Apple Inc. dataset collected for a period of 37 months (May 1, 2012 to June 1, 2015), providing a hit ratio of 85% (which exceeds the reported results in the literature). Perhaps more importantly, we have addressed the following theoretical questions that relate to the design of expert and intelligent systems:

(A) What is the value of using online sources (specifically Wikipedia and Google News) when predicting the one-day ahead stock movement? In contrast with the majority of literature, we analyze this question through combining variables/features from these online sources with more traditional predictors. This allows us to quantify the value added rather than just obtaining a predictive model.

(B) Does the added value of these online sources differ with different targets? We chose five different one-day-ahead targets to examine if the value obtained from these sources changes according to different prediction questions.

(C) Which targets are most suitable for prediction based on the aforementioned five data sources?

(D) Which AI models provide the best predictive performance for each of the five targets?

From our case study, we have learned that the addition of these online sources are useful (especially for Targets 1-4). In addition, based on the Apple stock, it seems that Wikipedia has more predictive power than Google News. That being said, the addition of Google News indicators improve the predictive accuracy the AI models utilized by our expert system (see Figure 2.3 and Table 2.5). From our seven scenarios of data aggregation, it is clear that the addition of online data sources and our generated features can significantly improve the prediction accuracy. This can imply that there are news hidden in these sources according to the followers of the Efficient Market Hypothesis. Alternatively, one can say that changes in these data sources precede changes in the stock market. Our analysis also indicates that all five targets can be predicted (using the best model) better than a coin-flip. Our intelligent system's prediction performance is better than the results reported in the literature (see Section 2.4.2).

2.5.2 Implementing our Expert System in Practice

From an Expert and Intelligent Systems practical implementation perspective, our proposed system can be used in a number of different ways. First, on a basic level and through our interface, an investor who does not have a strong programming background can use our "knowledge base" to capture the total number of "Google News" articles and visitors of relevant Wikipedia pages. Through our plotting tools, that investor can visualize the crowd's perception of a given stock or index. We have shown that these perceptions can be predictive of stock movement. It is important to note that this information is not available by current commercial products. Second, on a more advisory level, our expert system can be used to provide a data-driven recommendation for investors; an informed short term buy, or sell strategy of stocks can be made relative to whether the investors portfolio carries the stock. From that viewpoint as well, investors can use our system to construct an ensemble of predictions (with at least two models - their current approach and our expert system's recommendation). In the case of a two-model scenario, our expert system can indirectly

help with quantifying risk/uncertainty (i.e., if both models agree, the likelihood of a correct outcome increases). If the investor already had access to multiple forecasting systems, then our expert system will present a new perspective on a stock since our model combines both traditional and nontraditional sources. In such a case, the investor can make his/her decision through a simple voting procedure. Third, our code, which provide through a link in this article, can be deployed in an existing fully automated short term trading system to make its decision-making process more comprehensive.

2.5.3 Limitations and Future Research

There are several limitations and opportunities for future work that arise from this study. First, we have only examined Apple stock over a certain time-period. Thus, it is not clear if our results and/or conclusions can extend prior or past this period. More generally, it would be interesting to examine if our conclusion would differ if a different type of commodity stock is chosen and/or if a stock index is desired. Second, we did not attempt to include other online data sources. It is not clear if the relevance of our sources would change if, for example, Facebook data is used. Therefore, there are several opportunities to extend our work by the inclusion of additional data sources. We expect a diminishing return with the inclusion of new data sources, since we expect some redundancy in the information captured from online data sources. That being said, it would be interesting to rank the value obtained from the different online data sources (for different stocks and indices). A third direction can be to consider the stochastic nature of the prediction. Our "inference engine" presented a deterministic prediction; however in practice, it might be interesting to have a level of certainty that is associated with the prediction. This can be accomplished through the incorporation of fuzzy systems, Bayesian Belief Networks (BBN), and ensemble approaches. The fourth, and perhaps the largest improvement on this financial expert system, is to attempt to predict the actual price rather than the movement. From an investor's point of

view, a 20% increase in stock price is very different than a 1% increase. In our analysis, these two scenarios are identical since they are both coded as an increase in stock price.

In conclusion, this paper presented a financial expert system to predict the movement of a stock on a daily basis. We have shown that taking into consideration predictive factors from multiple sources can improve its predictive performance. We have also shown that the performance of the AI models can change significantly depending on the target used. To encourage future research, we provide our code and data in `https://github.com/binweng/ShinyStock`.

2.6 Appendices

2.6.1 Appendix I - Process to Acquire Data from Google News

Google News data is acquired from Alphabet Inc.'s Google Search Engine. The use of Google News allows us to gather all sources of news produced over a particular time period based on some search keywords. From a stock market perspective, this allows an end user to search for a publicly traded company's stock, and obtain a number for the amount of news produced for that stock. The steps to obtain the amount of news produced are shown below:

1. Go to `www.google.com`.
2. Input the search keyword, such as "AAPL, Apple Stock".
3. Click "News" and then "Search tools".
4. Custom the date range to the date you want to search.
5. Click "Search tools" again to show the result.

2.6.2 Appendix II - Formulas for the Generated Features

In this study, we generated seven different types of features for Wikipedia traffic data and Google news data. The formulas are shown below. In the formula, n means the time periods, V_t means the data point at period t.

1. Moving Average:

$$MA(n)_t = \frac{V_t}{n} + \frac{V_{t-1}}{n} + \cdots + \frac{V_{t-n+1}}{n} \tag{2.1}$$

2. Exponential Moving Average:

$$EMA(n)_t = (V_t - MA(n)_{t-1}) \times (\frac{2}{n+1}) + MA(n)_{t-1} \tag{2.2}$$

3. Disparity:

$$Disparity(n)_t = \frac{V_t}{MA(n)_t} \times 100 \tag{2.3}$$

4. Momentum1:

$$Momentum1_t = \frac{V_t}{V_{t-5}} \times 100 \tag{2.4}$$

5. Momentum2:

$$Momentum2_t = (V_t - V_{t-5}) \times 100 \tag{2.5}$$

6. Rate Of Change:

$$ROC_t = \frac{V_t}{Momentum2_t} \times 100 \tag{2.6}$$

7. Relative Strength Index:

$$RSI(n) = 100 - \frac{100}{1 + \frac{AverageGain(n)}{AverageLoss(n)}} \tag{2.7}$$

2.6.3 Appendix III - Definition of Variables/Features in Table 2.3

We define the variables/features used in our model in the table below.

Table 2.6: Definition of the most predictive variables/features

Variable	**Definition**
Close	Closing price of the day
Google_x_day_disparity	Ratio of Google news volume to its x day moving average
Google_x_day_disparity_Move	Movement of Google_x_day_disparity as previous day
Google_EMA_x	x day exponential moving average of Google news volume
Google_EMA_x_Move	Movement of Google_EMA_x as previous day
Google_MA_x	x day moving average of Google news volume
Google_MA_x_Move	Movement of Google_MA_x
Google_Move	Movement of Google news volumes as previous day
Google_ROC_Move	Movement of the rate of change for Google news volume as previous day
Google_RSI_Move	Movement of relative strength index for Google news volume as previous day
High	Highest price of the day
Low	Lowest price of the day
Open	Opening price of the day
P/E Ratio	Price-Earning ratio
RSI	Relative strength index of the stock price
RSI_Move	Movement of RSI
Stochastic Oscillator	Compares a security's closing price to its price range over a given time period
Stochastic_Move	Movement of Stochastic Oscillator
Wiki_x_day_disparity	Ratio of Wikipedia traffic to its x day moving average
Wiki_x_day_disparity_Move	Movement of Wiki_x_day_disparity
Wiki_EMA_x_Move	Movement of x day exponential moving average for Wikipedia traffic
Wiki_MA_x_Move	Movement of x day moving average for Wikipedia traffic
Wiki_Momentum_1	Ratio of current close price to the price three day's ago
Wiki_Move	Movement of Wikipedia as previous day
Wiki_ROC	Rate of change (ROC) of Wikipedia traffic
Wiki_ROC_Move	Movement of Wiki_ROC
Wiki_RSI	Relative strength index of Wikipedia traffic
Wiki_RSI_Move	Movement of Wiki_RSI
William %R	The level of the close price relative to the highest high

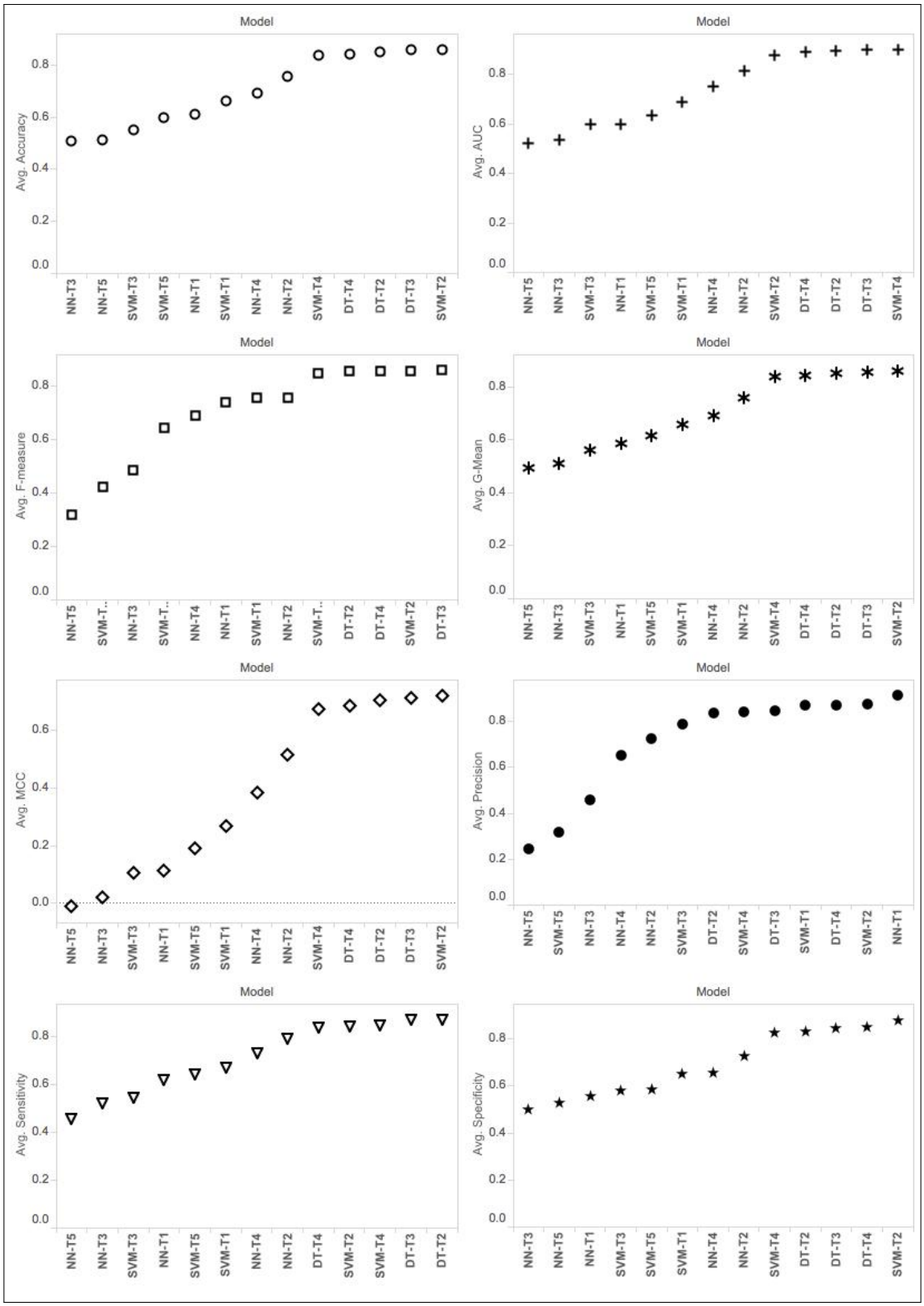


Figure 2.3: A visual summary of the performance of the 3 data mining models for each of the five targets.

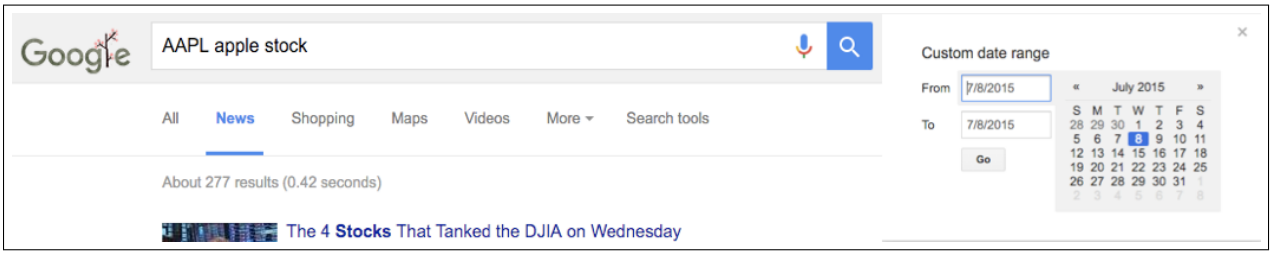


Figure 2.4: Screen-shot of an illustration of acquiring data using google's search engine

Chapter 3

Macroeconomic Ensemble Based Approaches

Accurately Predict the Monthly Closing Price of

Major U.S. Stock and Sector Indices

3.1 Abstract

Changes in macroeconomic factors can have a profound impact on the stock market. A large body of economics research attempts to quantify the relationship between macroeconomic variables and stock prices, often using time-series approaches, e.g., autoregressive integrated moving average (ARIMA) models. These approaches are somewhat limiting since they can have: (a) limited predictive power and (b) restrictive assumptions. To address these limitations, we propose an intuitive three-phase procedure for stock market prediction. In Phase I, an automated approach for acquiring the monthly values for 23 potentially relevant macro-economic factors is presented. Phase II utilizes several data mining methods with a modified leave-one-out cross validation strategy to select the most relevant macroeconomic predictors for a given stock index. In Phase III, we use four ensemble methods for predicting the one-month ahead stock and sector index prices. To evaluate the effectiveness of our approach, we collected the monthly values for 23 potentially relevant macroeconomic factors from 01/1992 to 10/2016. We trained our models to predict four major U.S. indices ($DJI, $NYA, $IXIC, $GSPC) and nine major sector indices. To test the performance of our models, we examined the one-month ahead prediction for these 13 indices for 2015-2016, and we used the mean absolute percent error (MAPE) as the evaluation metric. Across the four ensemble models, the average MAPE values for the four major indices and nine sector indices were 1.46% and 3.01%, respectively. Perhaps more impressively, our best model (out

of the four ensembles) for a given index had a MAPE of $< 1.87\%$ (with 9 out of the 13 indices having a MAPE under 1%). These results outperform the reported performance values in the literature. To encourage the implementation of our method, we provide a decision support system for investors and our code for researchers.

3.2 Preface

The prediction of stock prices has continued to fascinate both academia and business. The question remains: "To what extent can the past history of a common stock's price be used to make meaningful predictions concerning the future price of the stock?" (Fama, 1965). The Efficient Market Hypothesis (EMH) (Fama, 1965) and the Random Walk Theory (Cootner, 1964; Fama et al., 1969; Fama, 1991, 1995) laid the foundation for early stock market research . These models posited that stock prices cannot be forecasted since they are driven by new information and not present/past prices. Accordingly, prices will follow a random walk and they cannot be predicted with >50% accuracy.

There has been an increasing number of studies (Malkiel, 2003; Smith, 2003; Nofsinger, 2005; Prechter Jr & Parker, 2007; Bollen et al., 2011) that provide evidence contrary to what is suggested by the EMH and random walk hypotheses (RWH). These studies show that the stock market can be predicted to some degree and therefore, questioning the EMH's underlying assumptions. Many within the business community also highlight two main "storylines" that demonstrate that the stock market can be accurately predicted:

(A) Warren Buffet's ability to consistently beat the S&P index (Kiersz, 2015; Loomis, 2012); and

(B) The successful prediction of the 2008 Stock Market crash based on a "housing bubble", which was popularized by the New York Times Bestseller book (turned movie): "The Big Short: Inside the Doomsday Machine" (Lewis, 2015).

In this paper, we examine the utility of macroeconomic variables (including those highlighted in Lewis, 2015) in predicting the one-month ahead price for major U.S stock and

sector indices. We hypothesize that the price for different indices is driven by different economic indicators, and we attempt to quantify such effects using data-driven, nonparametric approaches.

3.3 Introduction

Investors use prediction models that can minimize the exposure of their investment risks; policy makers also benefit from having a full understanding of market volatility to revise future policies and regulations. However, stock market prediction still remains as one of the most challenging tasks, due to the non-linearity and non-stationary aspects of the time series data (Abu-Mostafa & Atiya, 1996). Numerous models have been developed to predict the stock market using patterns in historical data, but their prediction accuracy is not always guaranteed.

There are two common predictive model approaches for financial data: statistical time-series models and machine learning techniques (J.-J. Wang, Wang, Zhang, & Guo, 2012). Regression, exponential smoothing, autoregressive integrated moving average (ARIMA), and generalized autoregressive conditional heteroscedasticity (GARCH) are the statistical methods most commonly applied in stock market prediction (see e.g., Keim & Stambaugh, 1986; French, Schwert, & Stambaugh, 1987; Poon & Granger, 2003; Q. Li, Chen, Jiang, Li, & Chen, 2016). Despite their widespread applications, a major limitation in using parametric statistical methods in stock market prediction is that they require the model to be prespecified (Diebold & Nason, 1990; Enke & Thawornwong, 2005). From a practical perspective, it is not reasonable to assume that different stocks, indices and sector indices will have the same structure. In addition, as these parametric models become more complex the effect of estimation error increases (Diebold & Nason, 1990; Enke & Thawornwong, 2005). Thus, the predictive ability of traditional statistical models is often sub-par (Vaisla & Bhatt, 2010). For these reasons, machine learning methods, which are data-driven and assumption free, have

become more popular in stock market prediction (see the survey in Atsalakis & Valavanis, 2009 for details).

The application of machine learning (ML) techniques to stock market prediction can be categorized into two main groups: (1) non-voting approaches, which include artificial neural networks (ANNs) (Schöneburg, 1990; Grudnitski & Osburn, 1993; Kryzanowski, Galler, & Wright, 1993; Adya & Collopy, 1998; Zekic, 1998; Hamid & Iqbal, 2004; Khansa & Liginlal, 2011; Weng, Ahmed, & Megahed, 2017), support vector machines (SVM) (Weng et al., 2017; K.-j. Kim, 2003; Huang, Nakamori, & Wang, 2005; Schumaker & Chen, 2009; Ou & Wang, 2009), and classification & regression trees (CART) (Ou & Wang, 2009; Tsai & Hsiao, 2010); (2) voting/ensemble (Da Silva, Hruschka, & Hruschka, 2014) and hybrid methods (Pai & Lin, 2005; Hassan et al., 2007; H.-j. Kim & Shin, 2007; Lu, Lee, & Chiu, 2009; Kao, Chiu, Lu, & Chang, 2013), which combine multiple models to improve the predictions made. The existing ML-based stock market prediction papers can be characterized as follows:

(A) Only a small subset of those papers considered using macroeconomic predictors (Grudnitski & Osburn, 1993; Kryzanowski et al., 1993; Hamid & Iqbal, 2004; Enke & Thawornwong, 2005; Huang et al., 2005). We summarize the contributions of these methods in Table 3.1.

(B) The papers discussed in Table 3.1 generally had a single index and thus, it is not clear how generalizable are the results from these methods.

(C) In general, the use of ensemble-based approaches is limited and thus, there is an opportunity to improve the prediction results through voting/averaging.

Thus, there exists an opportunity to examine the effectiveness of ensemble-based machine learning methods in predicting the prices of different indices. In this paper, we focus on different U.S. stock indices and sector indices.

Table 3.1: A summary of ML papers using macroeconomic predictors

Ref.	**Period**	**Macro Economic Factors**	**Index**	**Target**	**Models Used**	Prediction Outcome
Enke and Thawornwong (2005)	1976 - 1999	M1, Production Price Index (PPI), CPI, T-Bill, Deposit Rate	S&P Index	Sign of excess stock returns	ANN	CORR: 0.0714; RMSE: 1.21
Grudnitski and Osburn (1993)	1983 - 1990	M1, Gold Price (GP)	S&P Index	Stock movement	ANN	Accuracy: 75%
Kryzanowski et al. (1993)	1981 - 1991	Industrial Production Index (IPI), GDP, Bond, Consumer Price Index (CPI), Bill rate, Montreal Exchange Index	Company	Return movement	Boltzmann Machine	Accuracy: 66.7%
Hamid and Iqbal (2004)	1984 - 1994	GP, Oil Price (OP), Bond, Foreign Currency	S&P Index	Volatility	ANN	RMSE 35-days pred. 0.003432 (log transf.)
Huang et al. (2005)	1990 - 2002	Exchange Rate of: USD to JPY	NIKKEI 225 Index	Stock Movement	RWH, Linear & Quadratic Discriminant Analyses, SVM	Hit Ratio: 73%
(Ahanger el al., 2010)	2000 - 2008	Industrial Production Rate, Inflation Rate Exchange Rate, Unemployment Rate, Oil Price, GDP, M1, M2	Company	Stock Price	Linear Regression ANN	MAPE: ANN = 1.42 & LR= 3.93
Tsai and Hsiao (2010)	2000 - 2007	A total of 14 and 17 features in their two best models (these included features from fundamental & technical analyses)	Taiwan Stock Index	Stock Movement	Multilayer Perceptron ANN, Principal Component Analysis Genetic Algorithms, & CART	Accuracy: $\approx$ 79%

In this paper, our overarching research questions are: (i) to examine whether macroeconomic factors can predict the 1-month ahead price of major U.S stock indices and sector indices; and (ii) if the answer to question (i) is "yes", then which factors are predictive of which indices. To examine these research questions, we propose a framework that is comprised of three main phases. First, an automated data acquisition procedure is developed to capture the monthly values for the different macroeconomic factors (i.e. our independent variables) and different stock indices (i.e. our different response variables). Second, variable selection approaches are used to identify the important predictors and eliminate noise/redundancies. Third, four ensemble methods are used to predict the 1-month ahead price of the different indices using the relevant macroeconomic indicators.

The remainder of this paper is organized as follows. In Section 3.4, the proposed method used for developing our prediction model is presented, which includes data preparation and acquisition, variable/feature selection, and an overview of applied methodologies. We discuss the experimental findings in Section 3.5. Our concluding remarks are presented in Section 3.6.

3.4 Methods

We propose a data-driven approach with three main phases to analyze the relationship between macroeconomic factors and the stock market. In Figure 3.1, we exhibit the process to build up our model. In Phase I, we collect data from several different online resources. The data acquisition phase is then subdivided into four steps, which include (a) a review of previous literature to learn the most significant macroeconomic factors in stock market prediction, (b) exploring other potential macroeconomic factors from public online data resources and discovering macroeconomic factors not previously used in the prediction of stock and sector indices, (c) acquiring the macroeconomic factors from disparate data sources for the four major U.S. stock indices and for nine U.S. sector indices from Yahoo Finance, and (d) manipulating raw data to derive variables for our model. Due to the changing

nature of economic data, we remove some economic factors used in previous literature and add some new factors in the process. In Phase II, we use several data mining methods with a modified leave-one-out cross validation (LOOCV) strategy for variable selection. We employ data mining methods and data visualization to gauge the level of importance for each potential factor, and select the most significant predictors. This variable selection strategy is performed for every stock and sector index. In Phase III, we then make our stock prediction using four ensemble machine learning models, which will be discussed in detail in Section 3.4.3. We use three evaluation criteria to find the most suitable prediction model for each of the major indices and each sector index.

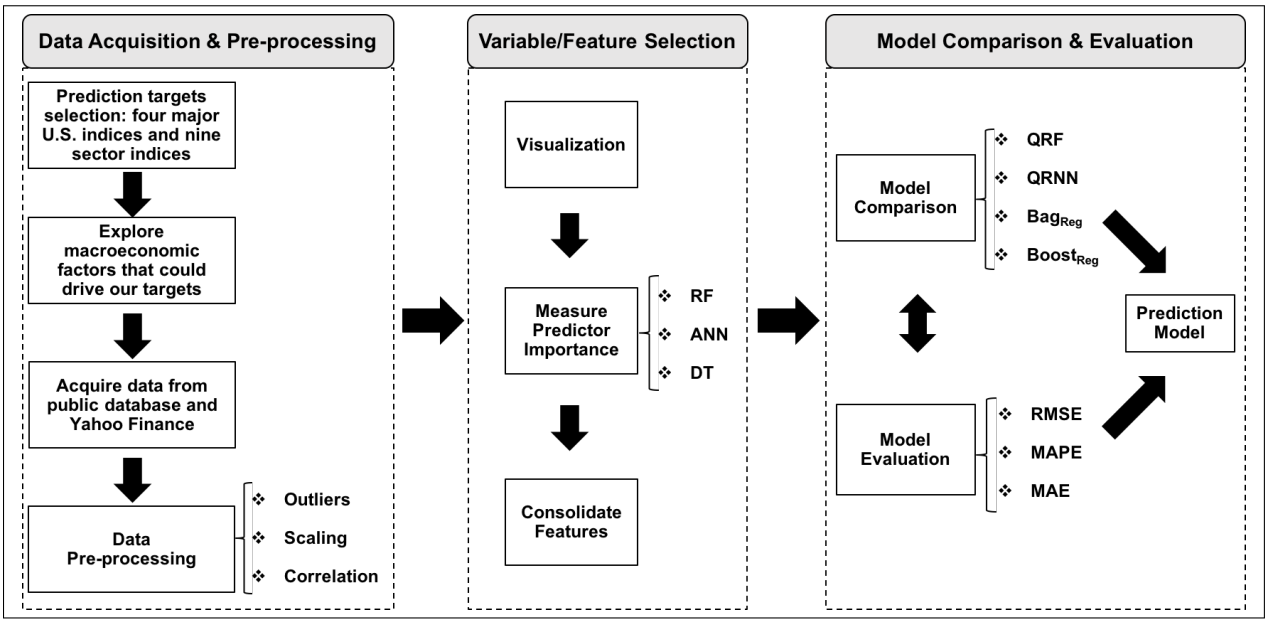


Figure 3.1: An overview of the proposed method

3.4.1 Data Preparation And Acquisition

Macroeconomic factors are elements believed to influence stock market movements (Hondroyiannis & Papapetrou, 2001). Researchers list several different macroeconomic factors that could potentially have an impact on stock market movements including oil prices (Sadorsky, 1999; Park & Ratti, 2008; Kilian & Park, 2009), housing prices (Case, Quigley, & Shiller, 2005), interest rates (Rahman, Sidek, & Tafri, 2009), foreign markets (Hamao,

Masulis, & Ng, 1990), and inflation (Tsai & Hsiao, 2010). N.-F. Chen, Roll, and Ross (1986) explored the effects of important macroeconomic variables on stock market returns. From the results, they concluded that industrial production, risk premium change, yield curve twist, and inflation all have significant effects on the variability of stock returns, but macroeconomic factors do not have significant influence on stock price changes. Some researchers have interest not only in stock returns and prices, but also in the relationship between macroeconomic factors and trading volume. For example, Flannery and Protopapadakis (2002) paid specific attention to trading volume from 1980 to 1996 and utilized 17 macroeconomic factors to analyze their relationship with high trading volume during the same time frame. Flannery and Protopapadakis (2002) observed that the Consumer Price Index (CPI), the Producer Price Index (PPI), the Monetary Aggregate (M1 and M2), the Employment Report, the Balance of Trade, and the Housing Starts are strong risk indicators for the stock market. Enke and Thawornwong (2005), for instance, selected 31 financial and economic factors to forecast stock market returns with neural network models. In addition, Ahangar, Yahyazadehfar, and Pournaghshband (2010) have found that the Inflation Rate, Money Supply (M1), and Growth Rates of Industrial Production to be predictive of stock price of individual stocks. The reader should note that the discussed references have only focused on major indices. It is not clear what macroeconomic factors impact and/or can help predict the different sector indices.

Based on the discussion above, we populated a list of 23 macroeconomic factors that are potentially predictive of four U.S. stock major indices (Dow Jones Industrial Average Index ($DJI), the NYSE Composite Index ($NYA), the NASDAQ Composite Index ($IXIC), S&P 500 Index ($GSPC)) and the nine U.S. stock major indices. Table 3.2 highlights these predictors, the source from which we extracted them and some of the papers (if any) that utilized them in their analysis of stock prices. Note that we added some potential factors that were not used in the prior literature since we are also interested in predicting the nine sector prices. These factors were hypothesized by the research team to be potentially relevant.

Table 3.2: A list of potentially predictive macroeconomic factors

Macroeconomic indices (monthly) used for prediction			
Oil Price[1] (Ahangar et al., 2010)	US unemployment rate[2] (Tsai & Hsiao, 2010)	US trade balance[3] (Tsai & Hsiao, 2010)	US consumer price index CPI[2] (Tsai & Hsiao, 2010)
US auto Sales[3] (Mahajan, Dey, & Haque, 2008)	Gold Price[5] (Hamid & Iqbal, 2004)	US monetary amount (M1)[5] (Tsai & Hsiao, 2010)	US monetary supply (M2)[5] (Tsai & Hsiao, 2010)
US industrial production index (IPI)[5] (Tsai & Hsiao, 2010)	Effective federal fund rate[5] (Mahajan et al., 2008)	US inflation_rate[5] (Ahangar et al., 2010)	
Macroeconomic indices (monthly) used for association			
Oil production[1] (Kilian & Park, 2009)	Oil supply[1] (Kilian & Park, 2009)	US house price index[5] (Case et al., 2005)	US housing starts[8] (Flannery & Protopapadakis, 2002)
US manufacturing PMI[4] (Johnson, 2011)	US house sold[6] (Flannery & Protopapadakis, 2002)	US employment change rate[2] (Ahangar et al., 2010)	
Other potential macroeconomic indices (monthly)			
US housing market index (HMI)[6]	US mortgage rate 15 years[7]	US mortgage rate 30 years[7]	US auto production[3]
US consumer sentiment[5]			
Public databases used (1): US Energy Information Administration (EIA) (3): US Bureau of Economic Analysis (BEA) (5): Federal Reserve Economic Data (FRED) (7): Federal Home Loan Mortgage Corporation (Freddi Mac)		(2): US Bureau of Labor Statistics (BLS) (4): Institute of Supply Management (ISM) (6): National Association of Home Builders (NAHB) (8): US Census Bureau (CB)	

3.4.2 Variable/Feature Selection

Current research literature shows that there is limited amount of information focusing on examining/predicting the relationship between macroeconomic factors and stock market trends by using data mining methods(Atsalakis & Valavanis, 2009). Since we selected 23 macroeconomic factors and considered these factors as potentially influential elements for the stock market, it was necessary to select the most relevant ones. The goal of this phase is to identify irrelevant and redundant features for different categories we are predicting. To achieve this goal, we use three data mining techniques, which include decision trees, random forests and artificial neural networks, to select the factors necessary to develop our prediction models. The three data mining methods we used have been shown to be suitable for feature selection in previous machine learning literature (Kohavi & John, 1997; Leray & Gallinari, 1999; Enke & Thawornwong, 2005; Tsai & Hsiao, 2010; Quinlan, 2014; Dag, Topuz, Oztekin, Bulur, & Megahed, 2016b). The modified LOOCV is also applied to the three methods, since the time slicing cross validation approach is can minimize the bias associated with the random sampling of the training and testing data samples (Arlot, Celisse, et al., 2010).

In this study, we predict two categories of targets: U.S. Major Indices and U.S. Sector Indices. We assume that the macroeconomic factors used to predict these indices should be multiple and should depend on the specific index. For example, although the price index for sectors would be affected by some common macroeconomic factors, the factors that could influence the energy sector index could differ significantly from the factors influencing the technology sector index. Also from previous literature, several researchers pay attention to the stock index prediction (Guresen et al., 2011; de Oliveira, Nobre, & Zarate, 2013), but most neglect the variety of factors that impact different sector indices. In our study, we apply a feature selection approach to different categories of targets, both major indices and sector indices, then the impact of each independent variable on each target is measured by the level of importance in the selection. The random forest method has been used as the primary tool to select the most important factors. Decision trees and artificial neural networks are

also applied and compared to determine the final subsets of factors to be used as predictors in our models. We normalize the data for the artificial neural network approach to improve the prediction performance (Ticknor, 2013). To increase the accuracy and feasibility, while eliminating the overfitting of our prediction model, we apply and obtain different predictors for each target. These predictors are presented in Table 3.3 in Section 3.5. The approach is performed using the Caret Package in R. Readers are referred to Kuhn (2008) for more details on the selection process and on how the level of importance of each variable is calculated.

3.4.3 Predictive Modeling

In this phase, we compared the effectiveness of four ensemble models, Quantile Regression Forest (QRF), Quantile Regression Neural Network Ensemble (QRNN), Bagging Regression Ensemble (BAG_{Reg}) and Boosting Regression Ensemble (BOOST_{Reg}) for predicting the stock price of two categories of targets which include four U.S. major indices and nine U.S. major sector indices. In the following paragraphs, we will first introduce the advantages of ensemble models and give a quick overview of the proposed ensemble prediction models, then explain the performance evaluation metrics used in this study. As we discussed in Subsection 3.4.1 and 3.4.2, the described methodology will be applied for each prediction target using the associated features obtained from Subsection 3.4.2.

Ensemble methods are effective fusion methods to improve the prediction accuracy of classifiers. Instead of selecting a single model, the idea of an ensemble model is to use a vote or an average of various models for a specific prediction. Ensemble models can solve statistical, computational, and representational problems for research, and there is sufficient empirical evidence pointing to ensemble performance being generally superior to that of individual classifiers (Drucker, Cortes, Jackel, LeCun, & Vapnik, 1994; Breiman, 1996a, 1996b; Quinlan, 1996; Schapire, Freund, Bartlett, & Lee, 1998; Opitz & Maclin, 1999; Dietterich, 2000a; Breiman, 2001; Maclin & Opitz, 2011). Ensemble methods are commonly used in machine learning to decrease the bias (boosting method) and variance (bagging

method) of predictions. The bagging ensemble, which stands for Bootstrap Aggregation, was introduced by (Breiman, 1996a). The main purpose of bagging is to decrease the variance of the predictions by producing bootstrap replicates of the training dataset. Each replicate is based on a different random sample with replacement from the entire training dataset, which means some observations may be drawn multiple times or be left out entirely. The boosting ensemble method, which is used to reduce mainly the bias, was introduced by (Freund & Schapire, 1995). Boosting refers to the idea of converting a weak learning algorithm into a strong learner, that is, taking a classifier that performs slightly better than random chance and boosting it into a classifier with arbitrarily high accuracy. Boosting creates weighted samples of the data based on whether each sample was predicted correctly or incorrectly in the previous iteration, while the final prediction is based on either weighted voting for classification or weighted averaging for regression problems.

To further understand the methods used, we introduce some notation and lay out some of the main assumptions. We assume a set of T weak learners, $h_t(\mathbf{x}), t = 1, 2, ..., T$, is created from the space (finite) of classifiers $\mathscr{H}$, each of which takes a $p \times 1$ input vector $\mathbf{x}$ and produces a prediction $h_t(\mathbf{x}) \in \mathbb{R}$ for a real-valued variable Y. Quantile Regression Forest (QRF) (Meinshausen, 2006), has become a powerful machine learning tool in predictive modeling. The QRF method is a generalization of the random forest regression (ensemble model of decision trees) (Breiman, 2001), which gives an approximation of the conditional mean of a response variable. (Breiman, 2001) defines a random forest (RF) as a "classifier consisting of a collection of tree structured classifiers $h(\mathbf{x}, \theta_t), t = 1, ..., T$, where θ_t are independently and identically distributed random vectors." The prediction of random forests for a new data point $X = x$ is the averaged response of all T trees, and for each tree, the prediction of a particular value x_i is based on the average values of Y of the rectangular space R_l, for the leaf l that x_i belongs to under tree h_t. Random forests are widely used in data mining and stock market prediction studies (Atsalakis & Valavanis, 2009; Lai et al., 2009; Wiesmeier, Barthold, Blank, & Kögel-Knabner, 2011). Random Forests overcome the overfitting problems of a

single decision tree, becoming a powerful method to analyze high-dimensional regression problems. The QRF method adds quantile estimation to the random forests method by recording the number of observations in each leaf l and assigning a weight to each observation based on the weight of its prediction on each tree. The Quantile Regression Forest (QRF) method estimates the quantiles of Y for a given value of $\mathbf{x}$. For the purposes of this research, we will estimate the mean. The QRF method is implemented in the quantregForest R package, and the reader is referred to Meinshausen (2006) for more details. The QRF model is used to develop the prediction model due to its superior performance compared with other decision tree based algorithms.

Similar to QRF, the Artificial Neural Networks (ANNs) are widely employed in a variety of computational data analytics including classification, regression, and pattern recognition. As we discussed in Section 3.3, ANNs can provide a more reliable prediction result for high-dimensional nonlinear data and has been a popular approach for stock market prediction (see e.g., Atsalakis & Valavanis, 2009; Ahangar et al., 2010; Vaisla & Bhatt, 2010; Guresen et al., 2011; de Oliveira et al., 2013; Schumaker, 2013). In our study, the Quantile Regression Neural Network (QRNN)(Taylor, 2000) is used for developing the prediction model. We utilize the ensemble method of bootstrap aggregation (bagging) to train the prediction model and use the sigmoid function as the activation function for our ANNs. The model is also trained and optimized by adjusting the number of neurons for each target to optimize the prediction performance. The QRNN method is implemented in the qrnn package in R, and the user is referred to Taylor (2000) for more details on the method.

Bagging, short for bootstrap aggregation, is another strong ensemble method for combining several base learners to produce a more accurate prediction. In fact, Random Forests is partially derived by the concepts proposed in bagging. Given a training set S of size n, bagging uses bootstrapping to generate a new training set S_t of size $n_t = n$ and fits a weak learner to the data. This process is repeated t times, and the final classification aggregation can be a majority vote for the classification problem, or an average of the predicted values

for regression problems. Bagging improves the performance of base classifiers, especially for unstable learners that vary significantly with small perturbations of the data set, e.g., decision trees. Breiman (1996a) suggested that the variance, which was defined as the scatter in the predictions obtained by using different training sets drawn from the same distribution, was reduced in the combination created by bagging, classifying it as a variance-reducing ensemble algorithm. For this analysis, we use a regression tree of depth = 1 (regression stump) as weak learner, and we call the algorithm BAG_{Reg}. The BAG_{Reg} method is implemented using the "fitensemble" package in Matlab.

We also use a boosting regression method (BOOST_{Reg}). Boosting refers to the idea of converting a weak learning algorithm into a strong learner, that is, taking a classifier that performs slightly better than random chance and boosting it into a classifier with arbitrarily high accuracy. Boosting originated from the PAC (probably approximately correct) learning theory (Valiant, 1984) and the question that (M. Kearns & Valiant, 1994) originally posed if a "weak" learning algorithm can be boosted into an arbitrarily accurate "strong" learning algorithm. AdaBoost, the most well-know boosting algorithm, has been shown to be a PAC (strong) learner. For regression cases, at every step t, the BOOST_{Reg} ensemble: (a) fits a new learner, i.e., a regression tree of depth 1, and then (b) computes the difference between the observed response and the aggregated prediction of all learners grown previously while minimizing the mean-squared error criterion. The "fitensemble" package in Matlab is used for implementing BOOST_{Reg}.

To evaluate the performance of the data mining procedures, three different evaluation metrics are used in this study: root mean square error (RMSE), mean absolute percentage error (MAPE) and R-square (RSQ). These three metrics are suitable for regression analysis based on previous literature. The reader is referred to Atsalakis and Valavanis (2009) for more details. The MAPE is used as the primary evaluation metric in this paper since it can be easily interpreted. For this analysis, we use ensembles of size $T = 500$.

3.5 Experimental Results and Discussion

In this section, we first highlight the results from the variable/feature selection phase of our methodology. We identify irrelevant and redundant features that do not contribute to, or have a minimal contribution to, the predictive model. We start by discussing the feature selection results for the U.S major stock and sector indices. We then present the results of the four ensemble data mining models (QRF, QRNN, BAG, and BOOST). Last, we evaluate the performance of the data mining approaches using three metrics as mentioned in Section 3.4.3. For the purpose of replication of our results, we present our code and a detailed tabular view of our results at `https://github.com/bzw0018/Stock-Index-Prediciton`.

3.5.1 Variable/Feature Selection

To predict stock price fluctuations and trends of U.S. major indices and sector indices, we started by selecting the most important macroeconomic factors as input predictors. Firstly, we assumed that the factors that could affect stock market should stand in distinctive levels among different categories of indices or sectors. As mentioned in Section 3.4.2, four data mining methods with a modified LOOCV approach were applied and evaluated for the selection. We then applied our feature selection approach for each category of target. To select the variables, we use the importance score metric for regression ensembles. The importance score metric is based on the total decrease in node impurities (using decision trees as weak learners), as measured by the sum of squares error (SSE) from splitting on the particular variable, averaged over all trees. We selected the factors with importance scores greater than 0.6 as the predictors. We discuss the results of major indices and sector indices separately.

Important Factors for U.S. Major Indices

We discuss here the macroeconomic factor influences on four U.S. major indices, and differentiate the factor importance levels based on our results. Table 3.3 shows the influential factors with importance scores greater than 0.6 for each of the indices. We list the factors

in a descending order based on their influence to the model. There are several additional discussions to be made from Table 3.3:

Table 3.3: Important factors for U.S. major Indices & Sectors

Index/Sector	Important Factors			
DJI	IPI	M2	CPI	House Price Index
	M1	Gold Price	15 Year Mortgage Rate	30 Year Mortgage Rate
GSPC	IPI	M2	CPI	House Price Index
	M1	Gold Price		
IXIC	CPI	M2	IPI	M1
	15 Year Mortgage Rate	30 Year Mortgage Rate	House Price Index	
NYA	IPI	M2	CPI	House Price Index
	M1	Gold Price	Oil Price	
Materials	CPI	M2	M1	Housing Starts
	Gold Price	Oil Production	15 Year Mortgage Rate	
Energy	CPI	M2	Housing Starts	M1
	Oil Price	Gold Price	15 Year Mortgage Rate	30 Year Mortgage Rate
	House Sold			
Financial	M2	Housing Starts	House Price	Unemployment Rate
	M1	CPI	Consumer Sentiment	
Industrials	CPI	M2	M1	Oil Production
	IPI	Housing Starts	15 Year Mortgage Rate	Gold Price
	30 Year Mortgage Rate			
Technology	M2	15 Year Mortgage Rate	CPI	30 Year Mortgage Rate
	M1			
Utilities	CPI	M2	M1	Housing Starts
	IPI	Oil Production	15 Year Mortgage Rate	Gold Price
	30 Year Mortgage Rate	House Sold	Federal Fund Rate	
Consumer Staples	M2	CPI	M1	Oil Production
	15 Year Mortgage Rate	30 Year Mortgage Rate	Housing Starts	Gold Price
	IPI	Federal Fund Rate		
Healthcare	M1	M2	Oil Production	CPI
	IPI	15 Year Mortgage Rate	30 Year Mortgage Rate	Gold Price
	House Sold			
Consumer Discretionary	M1	M2	CPI	Oil Production
	15 Year Mortgage Rate	IPI	30 Year Mortgage Rate	Gold Price
	Housing Starts			

(A) Four major indices are affected by different sets of socioeconomic factors, and this verifies our assumption.

(B) The three main indicators with the highest importance scores are IPI, Money Stock M2, and CPI. The results show that IPI, Money Stock M2 and CPI could impact the stock prices of all four indices to a great extent; however, some differences still exist among these three indicators. The change of IPI could have more influence on the stock price to the Dow Jones Industrial Average Index ($DJI), NYSE Composite Index ($NYA), and S&P 500 Index ($GSPC) independently compared to the Money

Stock M2 and CPI. The fluctuation of CPI could affect the stock price of the NASDAQ Composite Index ($IXIC) most, compared to the influence of the other two factors.

(C) Based on our variable selection rules, the factors that affect the stock price of the NASDAQ Composite Index ($IXIC) are very different from the other three indices. For example, except for the NASDAQ Composite Index ($IXIC), the stock price of the other three major indices are strongly influenced by IPI, Money Stock M2 and CPI as their importance scores are greater than 0.8 and the score differentiation is small, but less or equal to 0.8 for the NASDAQ Composite Index ($IXIC). Gold Price is also considered as an input predictor for the Dow Jones Industrial Average Index ($DJI), NYSE Composite Index ($NYA), and S&P 500 Index ($GSPC), but not for the NASDAQ Composite Index ($IXIC).

(D) The change of some macroeconomic factors could not have an obvious effect on the stock price of four indices, because of their low importance scores. For example, Manufacturing PMI, Employment Change, and Consumer Sentiment are the least powerful factors for all of the four major indices, since their importance scores are close to 0.

(E) The other factors, except for those we discussed above, could still impact the stock price of the four indices in different levels. For example, house market and oil market are the two major markets that are associated with the the stock prices of the four indices significantly. Another interesting finding is that Inflation Rate, which is an important macroeconomic indicator related with CPI and currency, has low level impact on all the four indices. In addition, the change of employment condition does not result in much fluctuation based on the feature importance score.

Important Factors for U.S. Stock Major Sectors

Similar to Subsection 3.5.1, we capture the results in Table 3.3, and discuss the relationship between macroeconomic factors and nine U.S. major sector indices. We followed the S&P Dow Jones Indices stock sector classification rule to categorize the stock sectors

including Materials, Energy, Financial, Industrials, Technology, Utilities, Consumer Staples, Consumer Discreet, and Healthcare. Unlike the influence of macroeconomic factors on the four major U.S. indices, the relationship between the factors and sector indices are primarily different for each sector.

For the sector index of Materials, Energy, and Utilities, the stock price is sensitive to the changes of CPI, Money Stock M2, Money Stock M1 and Housing Starts. This means that the stock price of the three sector indices maybe responding to the changes of these three macroeconomic factors quickly. This finding is very different when compared to the four major indices whose most important factors are IPI, Money Stock M2 and CPI. One explanation can be that the companies in these three sectors are related to infrastructure. The change of consumption and financial conditions may result in the fluctuation of these sector indices directly.

The stock prices of the Industrials, Consumer Staples, Healthcare, and Consumer Discreet sector indices are mainly impacted by four macroeconomic factors, which are CPI, Money Stock M2, Money Stock M1, and Oil Production. By comparing the importance scores among these four factors, CPI, Money Stock M2, and Money Stock M1 seem to be driving the stock price of three sector indices (Industrials, Consumer Staples and Consumer Discreet) to a high level. However, the macroeconomic factors affecting the stock price of the Health Care sector index is somewhat different when compared to the other three sectors. Specifically, the Money Stock M1, Money Stock M2, and Oil Production are the three most influential factors. Based on these results, it seems reasonable to posit that these four sectors are closely related to daily-life consumer behavior.

The results of the financial sector and technology sector indices seem to indicate that they are affected by a smaller subset of macroeconomic factors. The trends in the prices of these sections imply that they are correlated. In our estimation, this "correlation" makes sense since technology companies are disrupting the Technology sector (e.g., Apple Pay,

Google Wallet and PayPal). The results show that no macroeconomic factors had an importance score greater than 0.8. We hypothesize that this may be explained by the fact that our analysis was limited to the past 14 years. This somewhat small sample size may be insufficient to capture an emerging pattern, especially since the impact of technology on the financial sector can be seen as a recent phenomenon. Some other observations that pertain to the selected features include: (a) the financial sector is affected by consumer sentiment, and (b) changes in the housing market result in some fluctuations for both sector indices.

3.5.2 Prediction Model Outcomes

As explained in Section 3.5.1 and 3.5.1, we use root mean square error (RMSE), mean absolute percentage error (MAPE) and R-squared (RSQ) as the criteria to evaluate the performance of the quantile regression forest (QRF), quantile regression neural network (QRNN), Bagging Regression (BAG_{Reg}) and Boosting Regression (BOOST_{Reg}) models. The MAPE is used as the primary function to compare the performance of the models. In this section, the results of the prediction models are explained in two parts. The first part discusses the parameter settings for the two ensemble machine learning models. Then, the prediction results are discussed.

Parameters Settings

To maximize the performance of outputs, the first step is to set up the optimal parameters for the two models. The quantile regression forest is a generalization of random forests that can measure conditional quantiles to improve the information learned as discussed in Section 3.4. The size of terminal nodes needs to be set when applying a QRF model. The larger the size, the smaller the trees to be grown (and thus less time is need for training and execution). Based on a comparison of various tree sizes, our prediction model uses a node size = 10 for the 13 indices. The quantile regression neural network uses bootstrap aggregation to create an ensemble of models. We adjust the number of neurons for each

target to optimize the output. We use MAPE as the primary evaluation criterion for the comparison. Figures 3.2 and 3.3 show how the number of neurons affect the MAPE values for the four major indices and nine sector indices, respectively.

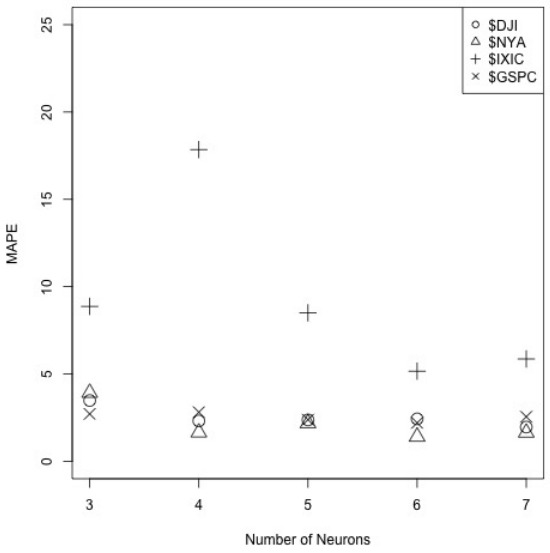


Figure 3.2: Performance of QRNN for different number of neurons: Indices

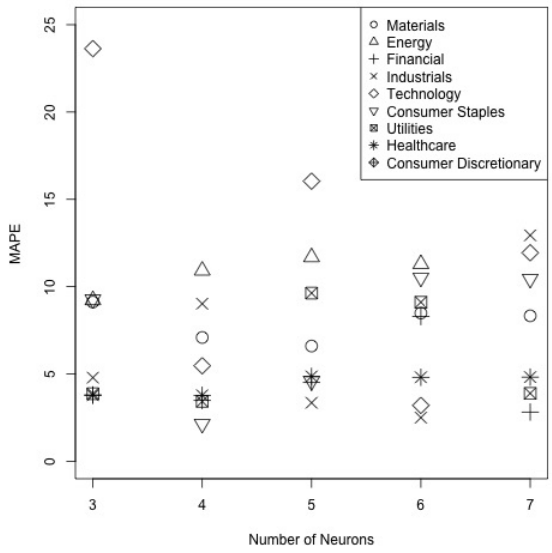


Figure 3.3: Performance of QRNN for different number of neurons: Sectors

From these two figures, there three noteworthy observations. First, There exist significant differences in the MAPE values among the major indices and sector indices with different number of neurons. Second, when we compare the four major indices, the S&P 500 Index ($GSPC) consistently has a higher MAPE, and the number of neurons has a more profound impact on its performance. Third, the optimal number of neurons used for the QRNN is four or six for our 13 indices since the improvement when more than six neurons are used is negligible (and sometimes negative). For the four major indices, six neurons results in the best combined computational efficiency and MAPE. This is also true for the Materials, Industrials and Technology sector indices. For the remaining indices, we use four neurons.

Experiment Results

To demonstrate the feasibility and effectiveness of the proposed methods, we perform experiments on predicting four major indices and nine major sectors. As we discussed above, the data is collected from January, 1992 to October, 2016 on a monthly basis. We use data from 1992 to 2014 to train the model and data from 2015 and 2016 to validate and evaluate our results. Our experimental results are presented in Table 3.4. We depict our predictions for the four major indices in Figure 3.4. Note that the straight line is the actual stock market price and the dotted line are the prediction value using the QRF, QRNN, BAG_{Reg}, BOOST_{Reg} models. Similar approaches and figures are also generated for the nine different sectors using the QRF, QRNN, BAG_{Reg}, BOOST_{Reg} models. For the sake of conciseness, we refer the reader to our Github website (`https://github.com/bzw0018/Stock-Index-Prediciton`).

Based on Figure 3.4 and Table 3.4, there are several interesting results that should be noted. First, the ensemble models have excellent predictive performances. The average MAPE across all models and indices is 2.53%. If we divide this average MAPE across the four major stock indices and the sector indices, the corresponding average MAPEs are 1.46% and 3.01%. Perhaps, what is even more impressive is that the best model for a given index performs no worse than 1.87% (which is the QRF for the NASDAQ Composite Index). This means that our best model predicts, on average, within 2% of the actual price for the next month. This result is significantly better than the reported values in the literature (see e.g., Grudnitski & Osburn, 1993; Enke & Thawornwong, 2005; de Oliveira et al., 2013; Kazem, Sharifi, Hussain, Saberi, & Hussain, 2013). Second, the BOOST_{Reg} model has the best overall performance. Third, a closer examination of Figure 3.4 shows that the prediction performance varies among different time periods. We hypothesize that this might be an indication that some macroeconomic factors might actually lag the stock market movement. While this is a reasonable justification, this is an area that need to be further studied in

Table 3.4: Performance of ensemble methods for Major/Sector Indices

Target	Prediction Model	Measurements			Target	Prediction Model	Measurements		
		RMSE	MAPE (%)	MAE			RMSE	MAPE (%)	MAE
Dow Jones Industrial Average	QRF	287.67	1.14	201.60	**NASDAQ Composite**	QRF	112.80	1.87	94.12
	QRNN	318.67	1.40	247.44		QRNN	186.98	2.58	128.59
	BAG_{Reg}	168.03	0.61	110.19		BAG_{Reg}	145.75	2.28	113.27
	$BOOST_{Reg}$	210	0.64	113.10		$BOOST_{Reg}$	146.05	2.17	109.54
NYSE Composite	QRF	184.07	1.43	148.61	**S&P 500**	QRF	45.41	1.84	38.75
	QRNN	273.36	1.96	205.78		QRNN	24.79	0.81	17.15
	BAG_{Reg}	133.72	0.91	95.46		BAG_{Reg}	43.73	1.37	29.30
	$BOOST_{Reg}$	178.28	1.32	138.67		$BOOST_{Reg}$	27.94	1.08	22.75
Industrials	QRF	1.21	1.67	0.90	**Consumer Staples**	QRF	1.00	1.59	0.79
	QRNN	3.10	4.55	2.47		QRNN	3.15	5.76	2.86
	BAG_{Reg}	1.87	1.69	0.90		BAG_{Reg}	1.02	1.54	0.77
	$BOOST_{Reg}$	0.50	0.51	0.28		$BOOST_{Reg}$	0.80	0.80	0.40
Technology	QRF	1.33	2.56	1.12	**Utilities**	QRF	1.25	2.13	0.97
	QRNN	2.25	4.16	1.76		QRNN	4.13	7.56	3.51
	BAG_{Reg}	1.37	2.62	1.13		BAG_{Reg}	1.23	2.01	0.92
	$BOOST_{Reg}$	0.99	1.67	0.70		$BOOST_{Reg}$	0.76	0.95	0.42
Materials	QRF	1.78	3.06	1.34	**Healthcare**	QRF	2.53	2.70	1.91
	QRNN	3.85	7.27	3.15		QRNN	6.59	8.00	5.51
	BAG_{Reg}	1.89	3.24	1.41		BAG_{Reg}	2.75	2.58	1.83
	$BOOST_{Reg}$	1.24	1.35	0.56		$BOOST_{Reg}$	1.13	0.93	0.65
Energy	QRF	2.33	2.80	1.80	**Discretionary**	QRF	1.54	1.67	1.29
	QRNN	6.21	7.92	5.03		QRNN	5.66	6.70	5.20
	BAG_{Reg}	2.68	3.13	1.97		BAG_{Reg}	1.51	1.58	1.22
	$BOOST_{Reg}$	0.90	1.03	0.66		$BOOST_{Reg}$	0.72	0.67	0.51
Financial	QRF	0.65	2.62	0.48					
	QRNN	1.15	4.73	0.86					
	BAG_{Reg}	0.72	2.84	0.52					
	$BOOST_{Reg}$	0.47	1.60	0.29					

future studies. Fourth, and a not obvious result, the QRNN's performance is dependent on the number of input features/predictors; this result can be seen by combining the results from Tables 3.3 and 3.4.

From a more holistic perspective, we chose 23 macroeconomic factors to evaluate their impact on 13 stock indices. Using a structured variable selection approach for each index, we obtained the subset of the most important predictors. The inclusion criterion was having factors with importance scores greater than 0.6. Based on our variable selection approach, we have determined that different sectors and indices are affected by somewhat different subsets of macroeconomic factors. While this is not a surprising result, it is not obvious from the analysis of the literature since most approaches typically predicted one target (i.e. a stock or an index, see Table 3.1 in Section 3.3). From a prediction perspective, our average MAPE

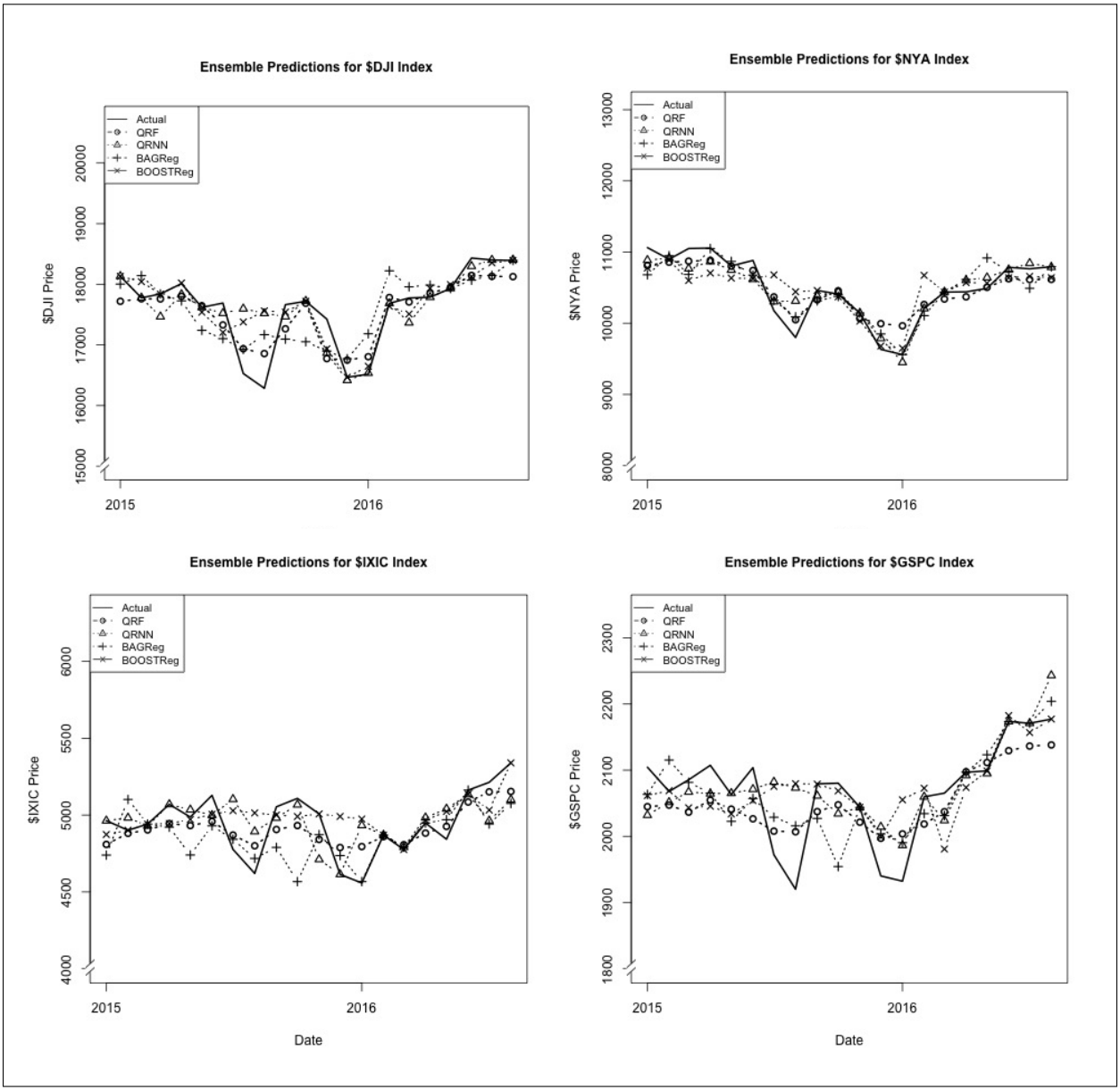


Figure 3.4: Experiment results of 4 major Indices using the QRF, QRNN, BAG_{Reg} and BOOST_{Reg} models

results and our best case performances clearly demonstrate the accuracy of our method for predicting the one-month ahead index prices.

3.6 Conclusions And Future Work

3.6.1 An Overview of the Impacts and Contributions of this Paper

The main objectives of this paper were to examine the utility of using macroeconomic factors and ensemble modeling in predicting the one-month ahead price of major U.S. stock and sector indices. To examine these objectives, we proposed an analytical framework that consisted of three main phases. In phase I, we acquired the data (01/1992-10/2016) pertaining to the price of 13 major indices and 23 potentially relevant macroeconomic factors. Phase II involved the use of variable selection methodology to reduce the subset of potential predictors without the loss of information. The results from the variable selection suggested that the important subset of important macroeconomic predictors can change according to the target index. In Phase III, we evaluated the performance of four ensemble approaches (QRF, QRNN, Bag_{Reg}, and Boost_{Reg}) in predicting the price for each of the 13 indices. The evaluation was primarily performed using MAPE. Our main contributions and results can be summarized as follows:

(A) In this work, we have examined the impact of macroeconomic factors on several stock and sector indices. As shown in Section 3.3, the literature typically considers a single index. Thus, our evaluation of how the importance of each macroeconomic factor changes when considering different indices has not been reported in the literature. This is especially true since the published methods in the literature do not combine the same data and models when different indices are evaluated by different papers.

(B) This paper is the first which examines the use of ensemble approaches for predicting the price of stock market indices.

(C) The average MAPE values for the main stock and sector indices were 1.46% and 3.01%, respectively. More importantly, our best model (out of the four ensembles) for a given index had a MAPE of $< 1.87\%$ (with 9 out of the 13 indices having a MAPE under 1%). These results present an improvement over those reported in the literature.

In addition, we provide our code to allow researchers to replicate and extend our work. From a general data mining perspective, our proposed framework can be extended to applications that involve longitudinal/panel data and continuous response variables.

3.6.2 Practical Implications from our Work

The ability to accurately predict the stock price, and consequently compute the estimated return, is the "dream" of every investor. In this paper, we presented an ensemble-based approach for predicting the one-month ahead price of 13 U.S. indices. Based on our reported results, where the MAPE of the best model for a given index was $< 1.87\%$, we believe that our approach has the potential to be informative for investors. As such, we have "packaged" our approach in an interactive decision support system (DSS) that can be used by investors. The DSS requires no coding by an investor, and is hosted on: `http://shiny.eng.auburn.edu/eco-stock/`. In our estimation, our DSS has several features that do not exist in current systems (see e.g., Gottschlich & Hinz, 2014a). First, it allows the investor to "pull" all the data needed, with a few clicks. This is only possible since our macroeconomic data is scraped from several public repositories. Second, we present some visualizations that are typically used in stock market analysis. For example, we provide the investor with an "interactive technical analysis chart". While we do not use technical analysis in our model, we believe that this analysis is useful from an exploratory data analysis viewpoint. Third, our predictions of actual price instead of movement (i.e. up or down) is insightful, especially since our prediction error is small. In our estimation, this DSS increases the appeal behind our method.

3.6.3 Limitations and Future Work

Despite the excellent predictive performance of our proposed methodology, there are a number of limitations that need to be highlighted. First, we have only examined the utility of our model for predicting the one-month ahead price. It is not clear how the selected

macroeconomic factors and/or the ensemble models will perform when predicting other time periods (e.g., quarterly or annually). Second, our analytical framework cannot be applied for intervals that are smaller than one month (e.g., daily forecasting) since macroeconomic factors are released monthly. Third, the scope of this work was limited to major U.S. stock and sector indices. Based on our scope, it is not clear if: (a) the performance of our method can be extended to individual stocks, and (b) our proposed framework can be adapted in predicting the price of major indices in non-US stock exchanges. The reader should note that the utilization of our approach in an emerging market may (and potentially should) result in a different initial set of macroeconomic factors. For example, in the case of an emerging market, it may be necessary to include other macroeconomic factors such as exchange rates. Fourth, we did not consider any additional data sources (e.g., predictors derived from technical analysis and fundamental analysis, Twitter sentiment, Wikipedia traffic volume, etc.). In our analysis, these omissions were justified since the addition of these sources would have, at best, led to a minor (practically insignificant) improvement. However, if any of the three assumptions above were changed, it is unclear whether ignoring these potential predictors can be justified. Note that we have only discussed limitations that pertain to price prediction (i.e., not stock movement) and to the utilization of our ensembles. Thus, our observations do not reflect on the literature that had fundamentally different objectives, utilized models, and/or assumptions.

In our estimation, there are two major opportunities for future research. First, researchers can examine the impact of predicting the price at different time-points. This is an important direction since it can provide insights pertaining to answering some of the gaps/limitations in our work. For example, the choice of a different time interval may lead to determining that multiple data sources are needed. Second, it seems logical to extend our work into a prescriptive trading engine, which uses our predictions to minimize investment risk and maximize the returns. For this second opportunity, researchers should examine multiple measures of risk (Szegö, 2002). In addition, we believe that the variation in price

predictions from our four ensembles can be used to quantify the uncertainty in a single index's price forecast.

In summary, this paper proposed a novel framework for predicting the one-month ahead price of major stock and sector indices. From our analysis, we have shown that our models have excellent predictive properties and we have identified the subset of macroeconomic factors that relate to the price of each index. To the best of our knowledge, the error rates achieved by our proposed method are much lower than those reported in the literature. We have also identified two major research streams that can build on our work. Our code and data are made available at `https://github.com/bzw0018/Stock-Index-Prediciton` to encourage the reproducibility of our work and future research. Finally, we present a decision support system that can be used by investors when predicting the one-month ahead price of the 13 indices. The system is hosted online at: `http://shiny.eng.auburn.edu/eco-stock/`.

Chapter 4

Predicting stock market short-term price based on machine learning

4.1 Abstract

As the stock markets grow, more investors attempt to develop a systematic approach to predict them. Since the stock market is very sensitively to external information, the performance of prediction systems is limited to merely considering traditional stock data. New forms of collective intelligence have emerged with the rise of the internet (e.g. Google Trends, Wikipedia, etc.). The changes on these platforms will significantly affect the stock market. In addition, both the financial news sentiment and volumes are believed to have an impact on stock prices. The goal of this paper is to develop and evaluate a decision making system that could be used to predict short term stock prices. In this research, we took advantage of the open source API which allows us to explore the hidden information among these platforms. A list of potential predictors are acquired and generated from these APIs. The list includes the traditional stock market information and external features. The prediction models are compared and evaluated using three machine learning models, neural networks, support vector regression and boosted trees. To evaluate the performance of our prediction system, we present a case study based on the Citi Group stock ($C). The data is collected from January 1, 2013 to December 31, 2016. From this case study, several results were obtained: (1) the use of financial news sentiment/counts, Google Trends, Wikipedia hits along with traditional metrics leads to improve the prediction performance; (2) the prediction models benefit from the dimensional reduction technique ; (3) the impact of external features on stock market gradually reduce overtime. Finally, a decision support system is provided to assist investors in making trading decision on any stocks.

4.2 Introduction

To accurately predict stock market is an attractive topic for both academia and industry. It is also a challenging task due to the complex and volatile nature of stock markets (Zhai, Hsu, & Halgamuge, 2007). Important financial theories such as the Random Walk model of stock prices and the Efficient Market Hypothesis (EMH) suggest that excessive stock prices or risk adjusted trading profits cannot be predicted based on currently available public information (Geva & Zahavi, 2014). However, many studies have rejected the random walk explanation of stock price behavior. Researchers in economics area raised the premise that stock price may correlate with economic events or seasonal variations (Kao, Chiu, Lu, & Yang, 2013). For instance, the observed daily stock returns reflected the stock market reaction to factors such as the release of economic indicators, government intervention or political issues etc. (Mok, Lam, & Ng, 2004). Financial time series do not exhibit random behavior and the future trend of stock price can be predicted (Abdullah & Ganapathy, 2000).

There are usually two types of prediction targets in stock market: i) actual stock price, index or returns on investment; ii) the movement of theses stock values (usually a binary classification, stock price rise and drop (Schumaker & Chen, 2009)). Prediction on the movement of stock market is meaningful to develop effective trading strategies for emerging markets. While as financial markets developed and more historical data becomes available, prediction of the specific stock price, index or return could provide decision makers more accurate risk-adjusted trading profits (Kara, Boyacioglu, & Baykan, 2011).

Traditional stock prediction typically uses stock market data, economic data and technical indicators. In terms of stock market data, buys and sells of stocks and shares, each stock can also be characterized by other market variables such as closing price, which is shown to be the most important variable for next day stock price prediction (Alkhatib, Najadat, Hmeidi, & Shatnawi, 2013). The statistics of economic indicators was shown to have a significant influence on the individual stock returns and the general stock index as they could show the trends in the economy performance, which possesses significant impact on the prospects of

growth and earnings of the companies (Tsai & Hsiao, 2010). Coincident indicators, leading indicators and lagging indicators are the three types of general economic indicators, which could be obtained at the same time, before or after the related economic activity occurs (Tsai, Lin, Yen, & Chen, 2011). Technical analysis considers historical financial market data such as past prices and volume of a stock and use charts as primary tool to predict price trends and make investment decisions (Murphy, 1999). Representative indicators include moving average, moving average convergence and divergence, relative strength index and commodity channel index (Tsai et al., 2011).

As crowd-sourcing and web technologies evolves, various sources of on-line data and analysis become available to the public. These include text mining of information such as financial news and media content. Zhai et al. (2007) combine the information from both news and technical indicators to enhance the predictability of the daily stock price trends and achieve higher accuracy. Tetlock (2007) found that the level of media pessimism could predict the stock market prices or the trading volume. In addition, Internet usage data obtained through APIs could give traces of information to investors making trading decisions. Moat et al. (2013) present the frequency of Wikipedia hits may contain early signs of stock market moves and suggest on-line data allow investors and researchers to gain broader insight for early information gathering. Preis et al. (2013) found that patterns in Google query volumes for financial-related search terms may be regarded as early warning signs of stock market moves. Their results indicate that combining extensive crowd-sourcing and financial news data may facilitate a better understanding of collective human behavior on the market, which could help the effective decision making for investors.

In addition to the growing multi-source data that may relate to stock markets, machine learning models have also been constructed to create an effective decision support system for investors. Artificial Neural Networks (ANNs), a data-driven and non-parametric approach, is one of the most popular tools in financial forecasting as it does not require any probability assumptions. Zhang and Wu (2009) integrate optimization with back propagation ANN and

develop an efficient forecasting model to predict stock indices. Guresen et al. (2011) compare different types of ANN in predicting the market values, and find out that the classical ANN model multi-layer perceptron (MLP) outperforms dynamic artificial neural network (DAN2) and the hybrid neural networks generalized autoregressive conditional heteroscedasiticity (GARCH-MLP) in terms of Mean Squared Error and Mean Absolute Deviate, when using real exchange daily rate values of the NASDAQ Stock Exchange Index. Though ANNs show strong ability in predicting stock market, they also suffer from shortcomings, such as difficulty in obtaining a stable solution and the risk of model over-fitting (Kao, Chiu, Lu, & Yang, 2013). Attempts to improve ANNs or find alternatives were explored in J.-Z. Wang, Wang, Zhang, and Guo (2011).

Support vector machines (SVM) have also been shown to successfully predict stock price. Trafalis and Ince (2000) show that using support vector machines for regression (SVR) could result in better financial forecasting than neural networks. Ince and Trafalis (2008) show that SVR outperforms or is as good as the MLP for a short term prediction in terms of mean squared error and risk premium, respectively. Meesad and Rasel (2013) proposed that SVR is a useful and powerful machine learning technique to recognize patterns in time series datasets, and applied SVR to predict stock market prices as well as trends.

Decision Trees, K-nearest neighbors and Genetic Algorithms etc. were also popular machine learning models that were frequently constructed for financial forecasting (Chang, 2011; Alkhatib et al., 2013; Atsalakis & Valavanis, 2009). Recent literature has proven that combining multiple classifiers could be superior to single classifiers in terms of accuracy. Qian and Rasheed (2007) show that consistent voting ensemble methods improve prediction accuracy of Dow Jones Industrial Average time series by 5%. Y. Chen, Yang, and Abraham (2007) formulated a flexible neural tree ensemble model by the local weighted polynomial regression, and show that the model could represent the stock index behavior quite accurately through experimental results. Tsai et al. (2011) used the hybrid methods of majority

voting and bagging to construct classifier ensembles, and indicate that multiple classifiers outperform single classifiers in predicting accuracy of returns on investment.

In developing a stock forecasting model, the first step is usually feature extraction (Kao, Chiu, Lu, & Yang, 2013). Performing feature extraction could help reduce the redundant features, which can reduce the measurements, storage requirements and the running time of classifiers, avoid the curse of dimensionality and improve prediction performance, as well as facilitate data visualization and understanding (Tsai & Hsiao, 2010; Y. Kim, 2006; Mladenić & Grobelnik, 2003). Hagenau, Liebmann, and Neumann (2013) examine whether stock price prediction based on textual information in financial news can be improved by enhancing existing text mining methods. They use more expressive features and employ feedback from markets as part of their feature extraction process. Kao, Chiu, Lu, and Yang (2013) use nonlinear independent component analysis as preprocessing to extract features from forecasting variables to provide more valuable information, and present the improved prediction accuracy through their empirical results. Tsai and Hsiao (2010) combine Principle Component Analysis, Genetic Algorithms and decision trees to filter out irrelevant variables based on union, intersection and multi-intersection strategies, and determine the important factors for stock prediction.

Table 4.1: A review of stock price prediction. ANN, GA, SVM, DT, VAR, SLR correspond to artificial neural networks, genetic algorithm, support vector machines, decision trees, vector autoregression, stepwise logistic regression respectively.

Paper	**Traditional**	**Crowd-sourcing**	**News**	**Tool**	**Prediction period**	**ML Approach**
Schumaker and Chen (2009)		✓	✓	✓	Twenty-minutes	GA, NB, SVM
Zhang and Wu (2009)	✓				One/ fifteen days ahead	ANN, Optimization
Boyacioglu and Avci (2010)	✓		✓	✓	Monthly	ANN, Fuzzy system, GA
Tsai and Hsiao (2010)	✓				Quaterly	ANN, DT, GA
Guresen et al. (2011)	✓				four days ahead	ANN
Khansa and Liginlal (2011)	✓	✓			Monthly, Quarterly	VAR, GNN
Tsai et al. (2011)	✓				Quaterly	ANN, DT, LR, Ensemble
J.-Z. Wang et al. (2011)	✓				Monthly	ANN
J.-J. Wang et al. (2012)	✓				Monthly	ANN, ESM, ARIMA, Ensemble
Hagenau et al. (2013)			✓		Daily	SVM
Alkhatib et al. (2013)	✓				Daily	KNN
Kao, Chiu, Lu, and Yang (2013)	✓				Intra-day	SVR
Kao, Chiu, Lu, and Chang (2013)	✓				Daily	SVR, ARIMA, ANFIS
Geva and Zahavi (2014)	✓		✓	✓	Intra-day	ANN, DT(GA), SLR
Gottschlich and Hinz (2014b)		✓		✓	Daily	Technical computation
Meesad and Rasel (2013)	✓				One/five/twenty-two days ahead	SVR
This Paper	✓	✓	✓	✓	Daily/weekly/Monthly	ANN, SVR, Ensemble

The multi-source data and models could be transformed into actionable investment opportunities through an efficient decision support system. Gottschlich and Hinz (2014b) proposed a decision support system design that enables investors to include the crowd's recommendations in their investment decisions and use it to manage portfolio. Schumaker and Chen (2009) developed the AZFinText system to estimate a discrete stock price twenty minutes after a news article was released. Boyacioglu and Avci (2010) designed the Adaptive Network-Based Fuzzy Inference System to model and predict the return on stock price index.

This paper aims at creating an adaptive decision making system to predict stock returns in short term. Compared with previous studies, in addition to designing the system, we also provides a publicly available stock return prediction tool with three targeting prediction period (daily, monthly, weekly) to provide more comprehensive information for their decision making. This system utilizes multi-sources data including stock market, Wikipedia hits, financial news, Google trends and technical indicators rather than single-sources of data. Adaptive feature selection was applied considering the uniqueness of each individual company based on their historical market data. Besides, to achieve the best possible forecasting accuracy, this system constructs a good performing ensemble machine learning models based on simulated empirical results.

4.3 Methods

To predict the stock price for different periods, we propose a data-driven approach that consists of four main phases, as shown in Figure 4.1. In Phase 1, the data is collected through four web database APIs, which are Yahoo YQL API, Wikimedia RESTful API, Quandl Database API, and Google Trend API. Then four sets of data are generated that include: (a) publicly available market information on stocks, including opening/closing prices, trade volume, NASDAQ and the DJIA indexes, etc.; (b) the number of unique visitors for pertinent Wikipedia pages per day; (c) daily counts of financial news on the stocks of interest and sentiment scores that are a measure of bullishness and bearishness of equity prices calculated

as statistical index of positivity and negativity of news corpus. (d) daily trend of stock related topics searched on Google. In addition, the commonly used technical indicators that reflect price variation over time (Stochastic Oscillator, MACD, Chande Momentum Oscillator, etc.) are obtained using R package TTR (Ulrich, 2016) as the fifth set of data. Furthermore, by using the underlying concepts of technical indicators in an attempt to uncover more significant predictors, we mold our primary data obtained from the databases into additional features. The second phase of data preprocessing consists of two sequential steps: (a) data cleaning which deals with missing and erroneous values. (b) data transformation for the strict requirements of some machine learning models, such as neural network. In Phase 3, the dimensional reduction technique is applied to reduce the dimension of the data by keeping the most important information of the data and improve the performance of the prediction models. Then in Phase 4, we make the stock price prediction with different periods (lags) using three machine learning models, which will be discussed in detail in the following section. In addition, a modified leave-one-out cross validation (LOOCV) are employed to minimize the bias associated with the sampling. These models are compared and evaluated based on the modified LOOCV using three evaluation criteria. The details for each of the phases are presented in the subsections below.

4.3.1 Data Acquisition

In this paper, we focus on developing a decision support system to assist investors investing in the stock market. Five sets of data, which are claimed as the significant features that could drive the stock market, were obtained from three open source APIs and generated using R package TTR (Ulrich, 2016). They are traditional time series stock market data, Wikipedia hits, financial news, Google trends and technical indicators. The five sets of data are preprocessed and merged in Phase I. First, we obtain publicly available market data on the stock of investors' choice through Yahoo YQL Finance API. The following five variables

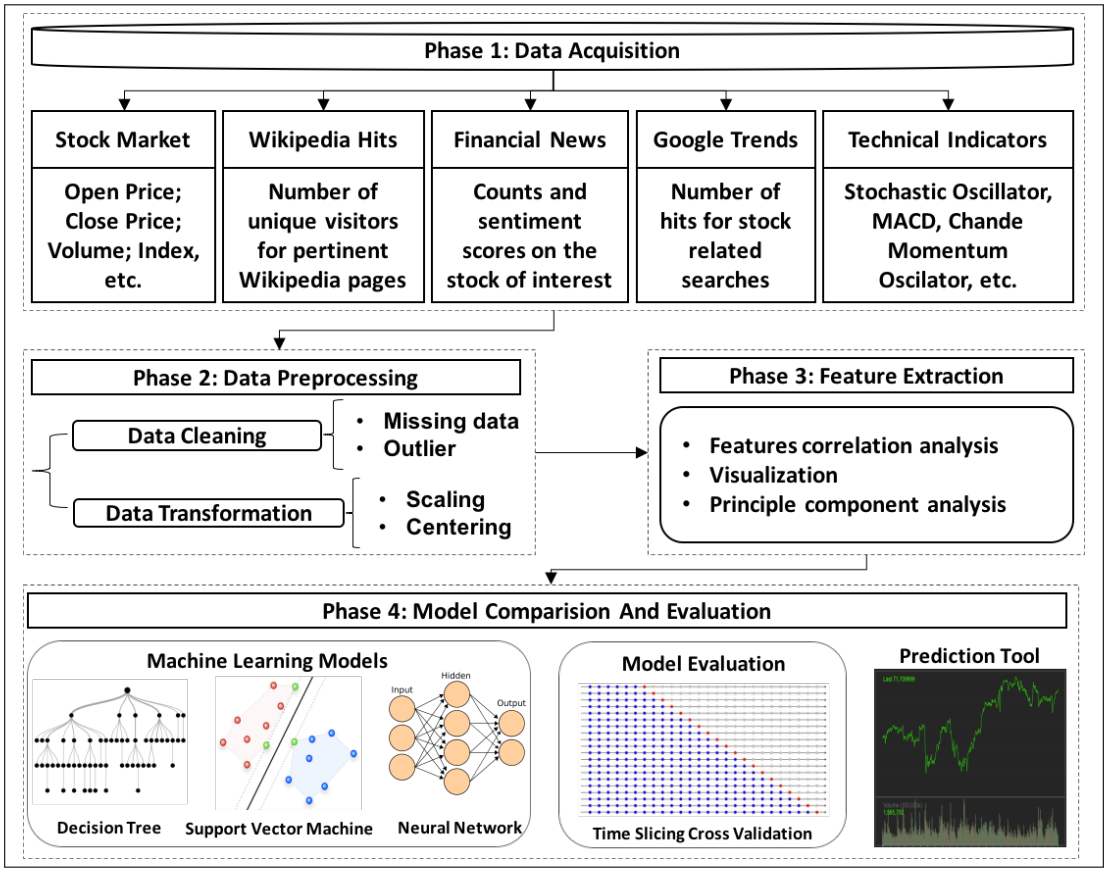


Figure 4.1: An overview of the proposed method

are obtained as part of inputs: the daily opening and closing price, daily highest and lowest price, volume of trades, and the stock related indexes (e.g. NASDAQ, DJIA).

The second set of data is queried through the Wikimedia RESTful API for pageview data, which allows us to retrieve the daily visits for the selected stock related pages with filtering around the visitor's class and platform. Please refer to `https://en.wikipedia.org/api/rest_v1/` for more details. The names of stock/company Wikipedia pages need to be input by users to process the queries. The third set of data is acquired using the Quandl Database API, which is the largest public API integrated millions of financial and economic datasets. The database "FinSentS Web News Sentiment" is used in this study. The **R** package Quandl (Raymond McTaggart, Gergely Daroczi, & Clement Leung, 2016) is used to access to the database through its API. The queried dataset includes daily news counts and daily average sentiment scores since 2013, derived from the publicly available

Internet sources. The fourth set data is the daily trends (number of hits) for stock related topics on Google Search. Our study uses the recent released Google Trends API (2017) to capture the trends information. The default setting of our prediction system is to search the trends on the stock tickers and company names. The users are highly recommended to use more accurate stock or company related terms to improve the performance of the prediction model.

In addition, researchers list several technical indicators that could potentially have an impact on the stock price/return prediction including stochastic oscillator, moving average and its convergence divergence (MACD), relative strength index (RSI), etc.(see e.g. K.-j. Kim & Han, 2000; Tsai & Hsiao, 2010; Göçken, Özçalıcı, Boru, & Dosdoğru, 2016). In our study, eight commonly used technical indicators are selected which shown in Table 4.2. Furthermore, the concepts of technical indicators are also deployed to datasets of Wikipedia, Financial News, and Google Trends in order to capture the hidden information and enhance the performance of the prediction models. Six of the selected indicators are applied to generate additional features for these three datasets. We make these six indicators in **bold** in Table 4.2. Please refer to `http://stockcharts.com/` for a detailed calculation for the eight indicators. Hereafter, ten periods of targets (based on prediction lags) are calculated using the "Close Price" which is acquired from Yahoo QYL API. Five sets of data and three types of targets are integrated to form as the original input database to our prediction model.

Table 4.2: The description of technical indicators used in this study

Technical Indicators	Description
Stochastic Oscillator	Indicator shows the location of the close relative to the high-low range.
Relative Strength Index (RSI)	Indicator that measures the speed and change of price movements
Chande Momentum Oscillator (CMO)	Capture the recent gains and losses to the price movement over the period
Commodity Channel Index (CCI)	Indicator used to identify a new trend or warn of extreme conditions
MACD	Moving average convergence or divergence oscillator for trend following
Moving Average	Smooth the time series to form a trend following indicator
Rate Of Change (ROC)	Measure the percent change from one period to the next
Percentage Price Oscillator	Measure the difference between two moving average as a percentage

4.3.2 Data Preprocessing

In this study, the data is automatically collected through four APIs. Thus, this will cause some features to have no values or no meaning for a given sample. In this subsection, our proposed methods include two parts; dealing with the missing data and removing outliers. First and foremost, we scan through all features queried from the APIs and determine if the pattern of missing data exists. If exists, the statistical average will applied to replace the missing points. Otherwise, the corresponding date that has missing values will be removed from the datasets. Next step is to understand outliers. The spatial sign (Serneels, De Nolf, & Van Espen, 2006) process is used to check outliers and remove the corresponding data points if necessary.

In order to make all predictors have a common scale, feature scaling is performed for each predictor. This is required by the models used in this study, especially support vector regression and neural networks, in order to avoid attributes in greater numeric ranges dominating those in smaller numeric ranges. This study deploys a straightforward and common data transformation approach to center and scale the predictor variables, which uses the average of each predictor value subtracted from all the values, then each value of the predictor variable is divided by its standard deviation. The new generated dataset is used in order to improve the stability of our prediction models.

4.3.3 Feature Extraction

For each of the five sets of data used here, around ten features are collected for the given period which leads to generate more than fifty variables in our original dataset. Due to the high dimensionality, the accuracy and speed of many of the common predictive techniques degrade. Therefore, the process of dimension reduction is necessary to improve the performance of the prediction model. In order to capture most of the information in the original variables, a principle component analysis (PCA) is applied to our training set for the prediction model. Researchers show that introducing PCA to the stock prediction can

improve the accuracy and stability of the model (Lin, Yang, & Song, 2009; Tsai & Hsiao, 2010).

Principal component analysis (PCA), in most cases compact the essence of the multivariate data and is probably the most commonly used multivariate technique. Its origin can be traced back to Peason (1901), who described the geometric view of this analysis as looking for lines and planes of closest fit to systems of points in space. Hotelling (1933) further developed this technique and came up with the term "principal component". The goal of PCA is to extract and only keep the important information of the data. To achieve this, PCA projects the original data into principal components (PCs), which are derived as linear combinations of the original variables so that the (second-order) reconstruction error is minimized. As we know, for normal variables (with mean zero), the (second-order) covariance matrix contains all the information about the data. Thus the PCs provide the best linear approximation to the original data, the first PC is computed as the linear combination to capture the largest possible variance, then the second PC is constrained to be orthogonal to the first PC while capture the largest possible variance left, and so on. This process can be obtained through the singular value decomposition (SVD). Since the variance depends on the scale of the variables, standardization (i.e., centering and scaling) is needed beforehand so that each variable has zero mean and unit standard deviation. Let X be the standardized data matrix, the covariance matrix can be obtained as $\Sigma = \frac{1}{n}XX^T$, which is symmetric and positive definite. By spectral theorem, we can write $\Sigma = Q\Lambda Q^T$, where Λ is a diagonal matrix consisting of ordered eigenvalues of Σ, and the column vectors of Q are the correspondent eigenvectors, which are orthonormal. The PCs then can be obtained as the columns of $Q\Lambda$. It can be shown [(Fodor, 2002)] that the total variation is equal to the sum of the eigenvalues of the covariance matrix $\sum_{i=1}^{p} \mathrm{Var}(\mathrm{PC}_i) = \sum_{i=1}^{p} \lambda_i = \sum_{i=1}^{p} \mathrm{trace}(\Sigma)$, and the fraction $\sum_{i=1}^{k} \lambda_i / \mathrm{trace}(\Sigma)$ gives the cumulative proportion of the variance explained by the first k PCs. In many cases, the first a few PCs have captured most variation, so the remaining components can be disregarded only with minor information loss.

PCA derives orthogonal components which are uncorrelated with each other, and since our stock market data seem to contain many highly correlated variables, we would apply PCA to help us alleviate the effect of strong correlations between the features, in the meanwhile reduce the dimension of feature space thus make the training more efficient. However, as an unsupervised learning algorithm, PCA does not consider the target while summarizing the data variation, in that case, the connection between the target and the derived components might be more complex, or it might also be the case that those surrogate predictors provide no suitable relationship with the target. Moreover, since PCA utilizes the first and second moments, it relies heavily on the assumption that the original data have approximate Gaussian distribution.

The method is used to capture the most possible variance to reduce space used and speed up algorithms. We set the threshold = 0.95 to retain majority of the variance. The result of the PCA will be presented and discussed in Section 4.4.But the limitation of the PCA is to seek linear combinations predictors that maximize variability. It is not clear if this assumption stands for the input features considered in this study. Thus, the prediction performance of models with or without dimension reduction approach are both compared.

4.3.4 Predictive Modeling

In this phase, we evaluate the effectiveness of three machine learning models; neural networks, support vector regression and boosted trees, are compared in predicting short term stock price. From a machine learning perspective, the consideration of stock price prediction models can be divided into two components: (a) capture the dimensionality of the input space; (b) detect the trade-off between bias and variance. A detailed discussion on our feature extraction approach using dimension reduction technique has presented in Section 4.3.3. Therefore, this section focus on selecting models for our study based on the bias/variance trade-off. The reader should note that, the cross validation approach has been applied to training the three prediction models. In the following subsections, we first

provide a short overview of the approaches of our proposed regression and time series cross validation. Then we introduce the performance evaluation metrics used in this study to identify the most suitable approach.

Neural networks

Inspired by complex biological neuron system in our brain, the artificial neurons were proposed by McCulloch and Pitts (1943) using the threshold logic. Werbos (1974) and Rumelhart, Hinton, and Williams (1985) independently discovered the backpropagation algorithm which could train complex multi-layer perceptrons effectively by computing the gradient of the objective function with respect to the weights, the complicated neural networks have been widely used since then, especially since the reviving of the deep learning field in 2006 as parallel computing emerged. Neural networks have been shown as the most successful among machine learning models in stock market prediction, due to their ability to handle complex nonlinear systems over the complex stock market data.

In neural networks, the features are based on input x and the weighted sum ($z = w^T x$). The information is then transformed by the functions in each neuron and propagated through layers, finally to the output we desire. If there are hidden layers between the input and output layer, the network is called "deep", and the hidden layers could distort the linearity of the weighted sum of inputs, so that the outputs become linearly separable. Theoretically, we can approximate any function that maps the input to the output, if the number of neurons are not limited. And that gives the neural networks the ability to obtain higher accuracy in stock market prediction, where the model is extremely complicated. The functions in each neuron are called "activations", and could have many different types. The most commonly used activation is the sigmoid function, which is smooth and has easy-to-express first order derivative (in terms of the sigmoid function itself), thus is appropriate to train by using backpropagation. Furthermore, its S-shaped curve is good for classification, but as for regression, this property might be a disadvantage. It is worth to note that the rectified linear unit

(ReLu), which takes the simple form $f(z) = \max(z, 0)$, has the advantage of a less likely to vanish gradient, but rather constant (when $z > 0$), thus results in faster learning in networks with many layers. Also, the sparsity of its weights arise as $z < 0$, thus could reduce the complexity of the representation on large architecture. Both properties allow the ReLu to become one of the dominant non-linear activation functions in the last few years, especially in the field of deep learning (LeCun, Bengio, & Hinton, 2015).

Support vector regression (SVR)

To explain the learning process from statistical point of view, Vapnik and Chervonenkis (1974) proposed VC learning theory, and one of its major components characterizes the construction of learning machines that enable them to generalize well. Based on that, Vapnik et al. developed the support vector machine (SVM) (Boser, Guyon, & Vapnik, 1992; Cortes & Vapnik, 1995), which has been shown to be as one of the most influential supervised learning algorithms. The key insight of SVM is that those points closest to the linear separating hyperplane, called the support vectors, are more important than others. Assigning non-zero weights only to those support vectors while constructing the learning machine can lead to better generalization, and the hyperplane is called the maximum margin separator. Vapnik and Drucker et al. (1997) then expanded the idea to regression problems, by omitting the training points which deviate the actual targets less than a threshold ε while calculating the cost. These points with small errors are also called support vectors, and the corresponding learning machine for regression is called support vector regression (SVR). The goal of training SVM/SVR is to find a hyperplane that maximize the margin, which is equivalent to minimize the norm of the weight vector for every support vectors, subject to the constrains that make

each training sample valid, i.e., for SVR, the optimization problem can be written as

$$\begin{aligned} \min \quad & \tfrac{1}{2}||w||^2 \\ \text{s.t.} \quad & y_i - w^T x_i - b \leq \varepsilon \\ & w^T x_i + b - y_i \leq \varepsilon \end{aligned}$$

where x_i is a training sample with target y_i. We will not show the details here, but maximizing its Lagrangian dual is a much simpler quadratic programming problem. This optimization problem is convex, thus would not be stuck in local optima, and it has well-studied techniques to solve, such as the squential minimal optimization (SMO) algorithm.

Theoretically, SVR could be deployed in our regression model to capture the important factors that significantly affect stock price and avoid the problem of overfitting. The reason is not limited to picking the support vectors but also the introduction of the idea of soft margins (Cortes & Vapnik, 1995). The allowance of softness in margins dramatically reduces the computational work while training, more importantly, it captures the noisiness of real world data (such as the stock market data) and could obtain more generalizable model. Another key technique that makes SVM/SVR so successful is the use of so-called kernel trick, which maps the non-linearly-separable original input into higher dimensional space so that the data become linearly-separable, thus greatly expand the hypothesis space (Russell, Norvig, & Intelligence, 1995).

However, SVM/SVR has its own disadvantages. The performance of SVM/SVR is extremely sensitive to the selection of the kernel function as well as the parameters. In that case, we picked Radial Basis Function (RBF) as the kernel in our SVR since the stock market data are with high noise. Another major draw back to kernel machines is that the computational cost of training is high when the dataset is large (Goodfellow, Bengio, & Courville, 2016), and also suffers the curse of dimensionality and struggles to generalize well.

Boosted tree

Rooted in probably approximately correct (PAC) learning theory (Valiant, 1984), posed the question that whether a set of "weak" learners (i.e., learners that perform slightly better than random guessing) can be combined to produce a learner with arbitrarily high accuracy. Schapire (1990) and Freund (1990) then answered this question with the boosting algorithm, and the most popular boosting algorithm Adaboost was also developed by Freund and Schapire (1995). Adaboost addresses two fundamental questions in the idea of boosting: how to choose the distribution in each round, and how to combine the weak rules into a single strong learner (Schapire, 2003). It uses the "importance weights" to force the learner pay more attention on those examples having larger errors, that is, iteratively fits a learner using the weighted data and updates the weights using the error from the fitted learner, and lastly combines these weak learners together through a weighted majority vote. Boosting is generally computationally efficient and has no difficult parameters to set, it (theoretically) guarantees to provide desired accuracy given sufficient data and a reliable base learner. However, practically, the performance of boosting significantly depends on the sufficiency of data as well as the choice of base learner. Applying base learners that are too weak would definitely fail to work, overly complex base learners could result in overfitting on the other hand. It also seems susceptible to uniform noise (Dietterich, 2000b), since it may over-emphasize on the highly noisy examples in later training and result in overfitting.

As an "off-the-shelf" supervised learning method, the decision tree method is used most common in the choice of base learners for boosting. It is one of the simplest to train yet powerful and easy to represent. It partitions the space of all joint predictor variable values into disjoint regions using greedy search, either based on the error or the information gain. However, due to its greedy strategy, the results obtained by the decision tree might be unstable and have high variance, thus often achieve lower generalization accuracy. One common way to improve its performance is boosting, which primarily reduces the bias as

well as the variances (Friedman, Hastie, & Tibshirani, 2001). We used the regression tree as the base learner for our boosting.

Time series cross validation

In this study, the modified LOOCV is applied through the prediction models comparison and evaluation approaches. The objective is to minimize the bias associated with the random sampling of the training and test data sample (Arlot et al., 2010). The traditional random cross validation (e.g. k-fold) is not suitable for this study, since the time series characteristic of the stock price prediction. Thus, the modified LOOCV approach is used, which performs a time window slicing cross validation. The method moves the training and test sets in time by creating numbers of time slice windows. There are three parameters to be set in the training process: (a) Initial Windows, which dictates the initial number of consecutive values in each training set samples; (b)Horizon, which determines the size of test set samples; and (c) Fixed Window, which is a logical parameter to determine whether the size of training set will be varied. A detailed discussion on this approach applied in this study will be shown in Section 4.4. The **R** package Caret (R Core Team, 2016) is used to perform this approach.

Performance measure

To evaluate the performance of the three modeling procedures, three commonly used evaluation criteria are applied in this study: (a) root mean square error (RMSE), (b) mean absolute error (MAE), and (c) mean absolute percentage error (MAPE). The three metrics

are obtained by the following formulas:

$$\begin{aligned}
\text{RMSE} &= \sqrt{\frac{1}{n}\sum_{t=1}^{n}(A_t - F_t)^2} \\
\text{MAE} &= \frac{1}{n}\sum_{t=1}^{n}|A_t - F_t| \\
\text{MAPE} &= \frac{1}{n}\sum_{t=1}^{n}\left|\frac{A_t - F_t}{A_t}\right| \times 100
\end{aligned}$$

where A_t is the actual target value for the t-th observation, F_t is the predicted value for the corresponding target, and n is the sample size.

The RMSE is the most popular measure for the error rate of regression models, as $n \to \infty$, it should converge to the standard deviation of the theoretical prediction error. However, the quadratic error may not an appropriate evaluation criterion for the prediction problems in which the true loss function would be unknown in most cases. Also, RMSE depends on scales, and is sensitive to outliers. On contrast, the MAE considers the absolute deviation as the loss and is a more "robust" measure for prediction, since the absolute error is more sensitive to small deviations and much less sensitive to large ones than the squared error. However, since the training process for many learning models are based on squared loss function, it is (logically) inconsistent (Woschnagg & Cipan, 2004). And it is still scale-dependent thus not suitable to compare prediction accuracy across different variables or time ranges. In order to achieve scale independence, MAPE measures the error proportional to the target value, thus consider the error as the percentage. However, MAPE is extremely unstable when the actual value is small (consider the case when the denominator $A_t = 0$ or close to 0). We will consider all three measures mentioned here.

4.4 Experiment Results and Discussions

4.4.1 Exploratory analysis

In this subsection, the exploratory analysis is applied to the original dataset in order to capture the characteristics of the features in order to improve the performance of the prediction models. Our approach uses the traditional market time-series data as well as the external online data sources. As we discussed in Section 4.3.2, the features collected through the API has high variability and contains missing/meaningless samples. Through exploring each features, the approach of data cleaning, feature centering, feature scaling are deployed. Furthermore, the correlation analysis among the features are applied.

A case study based on Citi Group stock ($C) is presented in this paper. The data is collected from January 2013 to December 2016 on a daily basis. Figure 4.2 shows a visualization of the correlation matrix of the five sets of input features, in which the features were grouped using the hierarchical clustering algorithm (so that the features with high correlations are close to each other), and the colors indicate the magnitude of the pairwise correlations between the features. The dark blue implies strong positive correlations, while the dark red stands for strong negative correlations, and white tells us the two features are uncorrelated. The dark blue blocks along the diagonal indicate that the features are fell into several large clusters, and within each cluster the features show strong collinearity, for example, the different prices (open, closed, high, or low) in the same day clearly are close to each other in most of the cases and thus probably fall into the same cluster. There are also features negatively correlated to each other, for instance, the volume and the index have opposite trends, which might due to the low volatility of the City stock, so that investors tend to buy other stocks when the corresponding market index is getting high.

Highly correlated features actually provide redundant information and add unnecessary complexity into the model. Although unlike linear regression in which orthogonal features assumption is required, highly correlated features could still significantly affect the stability

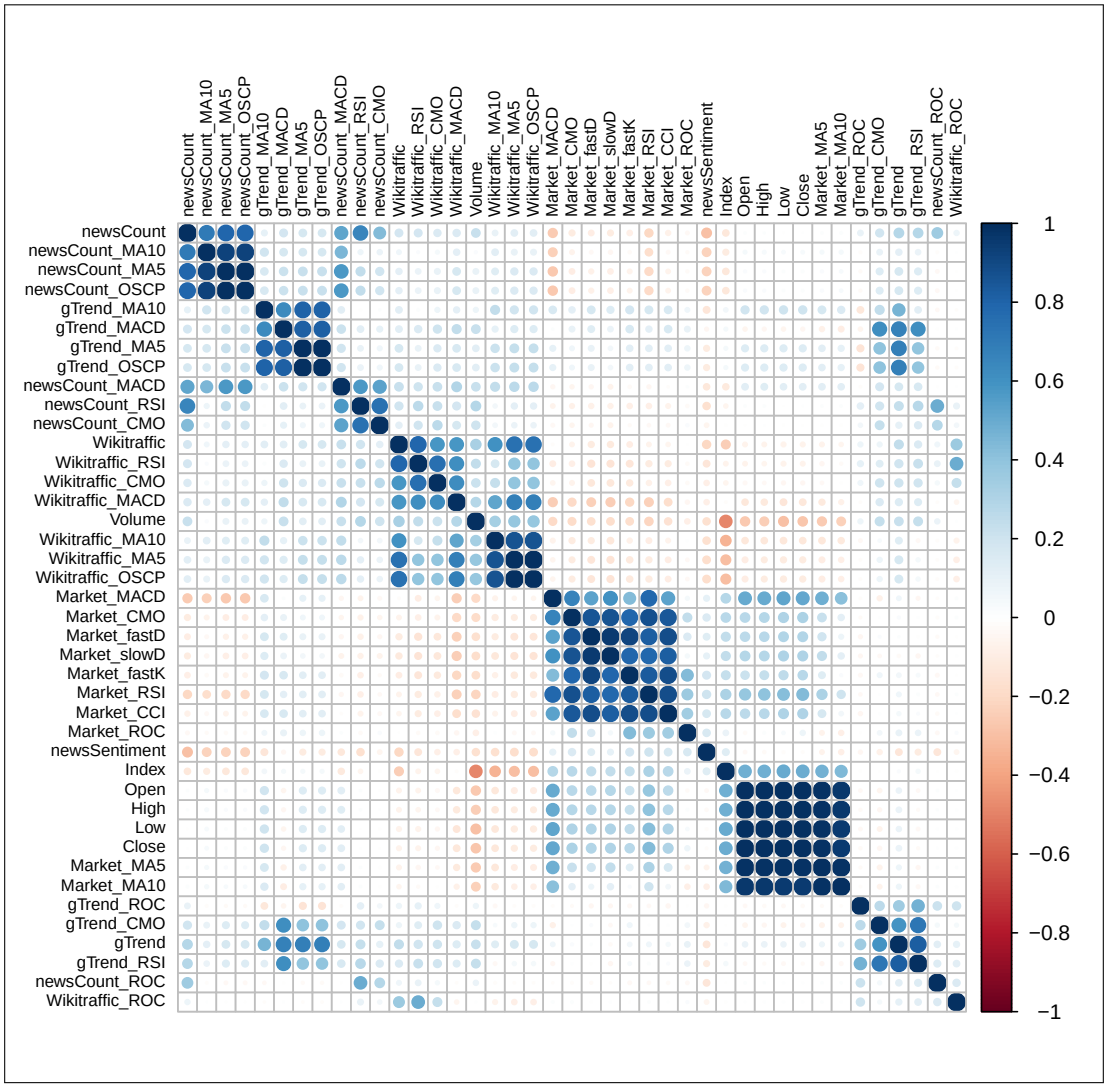


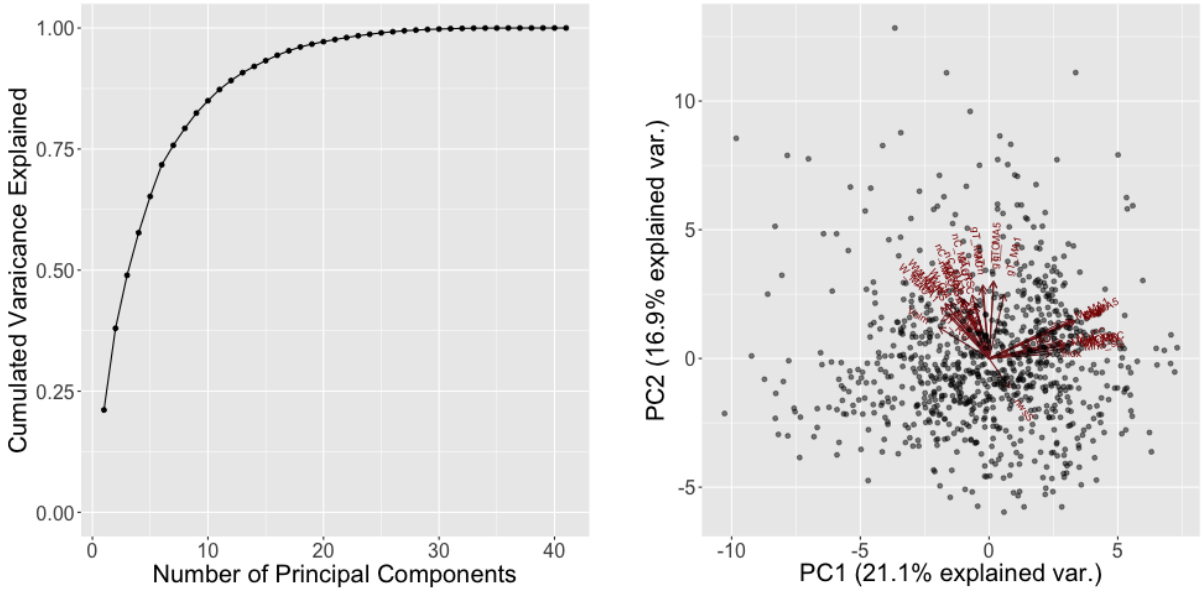
Figure 4.2: Correlation matrix for features

of machine learning models. This suggests us to deploy some feature extraction techniques, from which the strong correlations between features can be mitigated, so that the predictive performance of our models would be improved.

4.4.2 Feature extraction

The first three principal components accounted for 21.13%, 16.86%, and 10.95% of the total variance for the data, respectively. Figure (4.3a) shows the cumulative percentages of the total variation in the data which was explained by each component, from which we can

observe that the first 13 principal components described 90.78% of the information from the features, and the first 17 components captured 95.29%. After 26 components, more than 99.26% of the total variance has been explained, while the remaining 15 components only describe less than 0.74%. Deploying a threshold of 95%, we used 17 components for training. Note that we could also determine the optimal number of principal components for a specific model by using cross-validation.



(a) Percent of variance explained (b) Rotation

Figure 4.3: Results of principle component analysis

Figure (4.3b) characterized the loadings (i.e., the coefficients in the linear combination of features that derive a component) for each feature associated with the first two principal components. It is quite clear that the loadings of the prices as well as the technical indicators have the largest effect for the first component, e.g., the coefficient of close price is 0.2668, and that of RSI is 0.2645. As for the second component, **external Internet features** contribute the most in the positive direction, for instance, the coefficients for Google Trend, Wiki Traffic and News Count are 0.2547, 0.1957 and 0.2137, respectively. Also note that the News Sentiment plays a role that negatively associated with the second component with coefficient -0.1018. Figure (4.3b) also showed a scatter plot for the first two principal

components, from which we can notice that the derived components seems to be uncorrelated to each other, and this is also a key property of PCA.

PCA helped us to alleviate the effect of strong correlations between the features, also largely reduced the dimension of feature space thus would make the training more efficient. However, as an unsupervised learning algorithm, PCA did not consider the target while summarizing the data variation, in that case, the connection between the target and the derived components might be more complex, or it might also be the case that those surrogate predictors provide no suitable relationship with the target. Therefore, the prediction performance for models from features with and without PCA transformation would be compared.

4.4.3 Model comparison and evaluation

Three commonly used machine learning models has been deployed in our study: the neural networks (NN), support vector regression (SVR), and the boosting with regression tree as the base learner. We use three evaluation criterion (MAE, MAPE, RMSE) to evaluate the performance of the three models in this study. The data is split into two sets, training and testing, based on a 80/20 sampling scheme. Since the natural element of stock market is time series, the last 20% data is used as the testing set and the left 80% data as training set. As explained in Section 4.3.4 and 4.3.4, the approach of modified LOOCV using time slicing windows is applied through the model development approach. Specifically, the size of each training samples is 80% of the data and that of each validation samples is 5% of the data. The size is not varied through the time slicing. Therefore, a series of training and validation sets is generated and used for training and evaluating the models. During the time slicing approach, the corresponding training set only contains the data points that occurred prior to the data points in the validation set. Thus, no future samples are used to predict the past. Afterwards, the prediction performance is computed by averaging the validation sets. The performance of the three models using the features with and without PCA transformation are shown in Table 4.3 and 4.4. An visualization of the prediction is presented in Figure 4.4.

First, according to Table 4.3 and 4.4, the three performance measures are quite consistent in general. The RMSEs are slightly larger than the MAEs, as MAE is less sensitive to large deviations than RMSE. This indicates that our data contains quite a few outliers, which seems common due to the frequent turbulence in the stock market data.

Table 4.3: Results of comparing three machine learning models without PCA

	Train			**Test**		
	MAE	**MAPE (%)**	**RMSE**	**MAE**	**MAPE (%)**	**RMSE**
Neural Network	0.519	1.040	0.690	0.820	1.890	1.130
Support Vector Regression	0.730	1.460	0.930	10.300	23.000	11.700
Boosted Tree	0.462	0.925	0.608	1.290	3.070	1.890

Table 4.4: Results of comparing three machine learning models with PCA

	Train			**Test**		
	MAE	**MAPE (%)**	**RMSE**	**MAE**	**MAPE (%)**	**RMSE**
Neural Network	0.545	1.120	0.745	0.595	1.350	0.875
Support Vector Regression	1.080	2.220	1.420	1.010	2.280	1.380
Boosted Tree	0.409	0.837	0.530	0.409	0.919	0.521

We first compare the performances of the three models with and without PCA. NN and SVR have slightly smaller training errors using the original data than after deploying PCA, while the boosting model has smaller training error if PCA is used. As for test errors, all three models have better performance on accuracy if trained with PCA transformed predictors. The biggest differences appear on SVR, where the test errors without using PCA are about ten times as large as those obtained using PCA. Without PCA transformation, boosting also has test errors more than three times as large as those with PCA. And the ratios are around 1.3 for NN.

We then focus on the comparison of the training accuracy and test accuracy for each model. With PCA, the performance differences are quite tiny, while in most cases (except the MAE for SVR), the training errors are slight smaller than the test errors. However, without using PCA, the differences between training errors and test errors are much more obvious, again especially in the case of test error for the SVR, where the test errors are more than ten times as large as the training errors.

The explosion on test errors suggests severe overfitting. By applying all 41 features in the original data, the models are over-complicated with more parameters, thus hard to generalize well. SVR met this curse of dimensionality problem most seriously, it tries to select only those important examples to train to avoid overfitting, however, in such a high dimensional space (and it will use kernel trick to get a even higher dimensional space), the samples are quite sparse and it is very hard to obtain the discriminative examples, which depends heavily on the choice of kernel as well as hyperparameters (such as the coefficient of the regularization term which helps to control overfitting) we set, which is extremely difficult in practice. With PCA, the dimension of the feature space were reduced to 17 by discarding less significant dimensions while keeping discriminative information of the features, thus greatly helped to tame the curse of dimensionality and mitigate overfitting, not only for SVR, but for all three methods in our case.

It is also worth to note that the test MAE obtained by SVR on PCA transformed data are slightly smaller than its training MAE, which is not very common, but is still possible to happen, since the size of our training data is much larger than that of our test set, and the errors are calculated over the entire training set or test set. It might be the case that the training data contains some examples that are difficult to learn, and many examples of the test data are relatively easy to predict. Moreover, this result reveals the characteristic of SVRs that it deploys only those important examples to train and less likely to overfit. It might seem to contradict the result that SVR has the most severe overfitting problem on the original data, but here we are considering on the PCA transformed data, in which the discriminative predictors might be given.

Next, we compare the performances among the three methods. SVR obviously obtained the worst results, and we have already mentioned the mighty reasons for the poor performance of SVR in our case. Note that we also tried different kernels for SVR. The polynomial kernel helps to reduce the large gap between the training and test errors, but it generally performs worse than the radial basis function (RBF) kernel whose results are reported here,

due to the high noise in the stock market data. The prediction accuracy of NN and boosting are quite similar, while boosting performs slightly better. Note that as we increase the size of NN by adding more layers and more neurons, the results does not show obvious improvement on test error, since it might become overfit if the size of NN is too large and the nonlinear mapping from input to output is over-complicated.

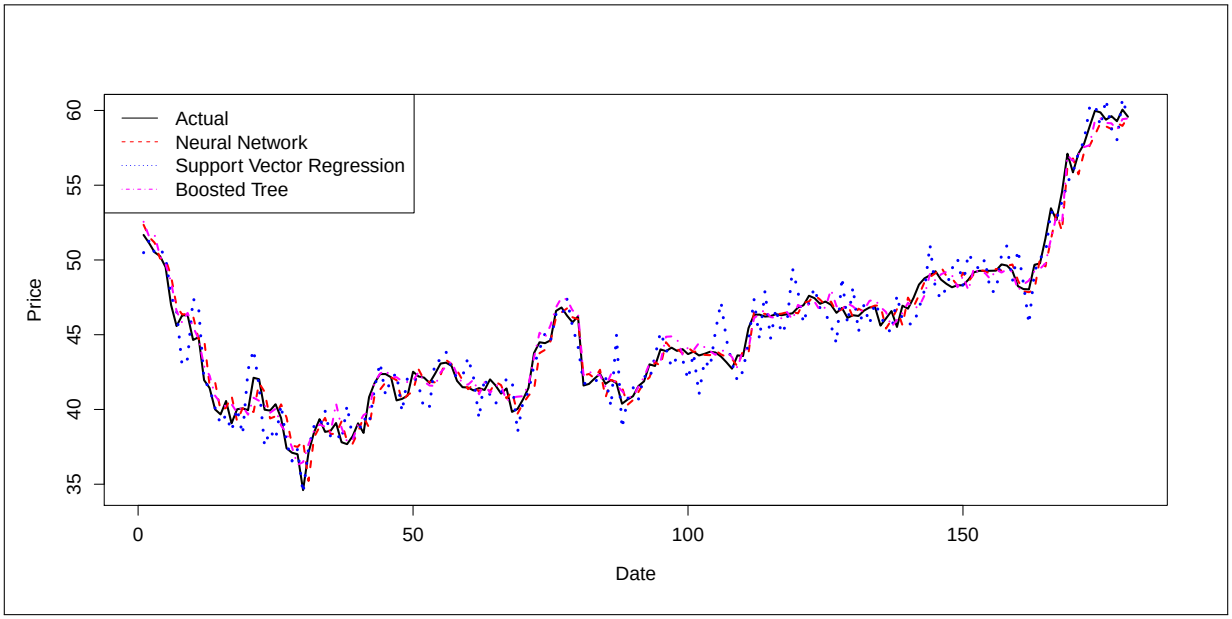


Figure 4.4: An visualization on the experiment results of three models

The visualization of the prediction performance based on the test set can be shown in Figure 4.4. It is interesting to point out that some hidden patterns could be observed from the point of view of the errors. There is an obvious delay of the prediction compared with the actual price. When the stock price is stable (i.e., only changes in a small range), SVR overestimate the volatility by exaggerating the amplitude as well as the frequency of oscillates quite a lot, and that is how SVR got the highest error. But SVR has the strongest ability to capture the trend, e.g., it fits the actual values best when there is a sudden jumps in the price. We could probably utilize this merit to predict the possible boom or slump in the stock market. In addition, the boosted tree and NN has slightly delay effect. Overall, if we need to choose a single model for prediction, we would suggest to use boosted tree or NN based on the prediction performance. It is worth mentioning that there is a lot of

possibilities to improve the model. For example, we can consider more layers for the NN which could be a direction of future studies.

From the above discussion, we have established that the performance of boosting approach is better than the other two for one-day ahead stock price prediction. To formally understand the usefulness of our approach, we considered another nine prediction periods to check the forecasting performance (Table 4.5). We use the notation **Lag X** as the target that predicts **X** day ahead price of the market. As an example, consider the approach of boosting with regression tree as the base learner using feature generated from PCA for each targets. Let us examine how our model development approach on different prediction periods based on the three evaluation metrics. The results is presented in Table 4.5. From the results, it is clear that the performance has a decreasing trend as the prediction period increases. Based on the natural of the market, the price's rate of change is commonly larger for long term than short term. Moreover, the results validate that the Internet features, such as Google Trends, significantly impact on the stock market for a very short periods (one or two days) and the impact will gradually reduce. Nonetheless, the prediction performance of the ten targets are all reasonable for assisting in the stock trading decision, since the MAPEs are far less than the actual rate of changes.

Table 4.5: The performance of the best model (boosted tree) on different targets

Target	**MAE**	**MAPE**	**RMSE**	**Target**	**MAE**	**MAPE**	**RMSE**
Lag 1	0.409	0.919	0.521	**Lag 6**	0.748	1.690	1.020
Lag 2	0.678	1.520	0.886	**Lag 7**	0.778	1.760	1.080
Lag 3	0.568	1.280	0.735	**Lag 8**	0.788	1.790	1.110
Lag4	0.662	1.490	0.868	**Lag 9**	0.837	1.890	1.150
Lag5	0.759	1.720	1.010	**Lag 10**	0.800	1.790	1.120

4.5 Conclusion and Future Work

In this paper, we developed a novel approach to predict short-term stock price. We focus on integrating multiple external data sources along with the traditional metrics to improve the performance of the short-term stock price prediction. Five sets of data sources are used

in our proposed approach, which are stock market data, technical indicators, financial news, Wikipedia and Google Trends. Three machine learning models are compared and evaluated. Base on the results in Table 4.4 and 4.5, our approach is more efficient and adaptable for the short-term stock price prediction. In addition, our prediction system is not limited to predict the future stock price for a specific period. We tested ten different periods to our model. The results is shown in Table 4.5. It shows that there is a increasing trend of prediction errors. The reason is that the rate of change of the stock is getting larger if we consider to predict the stock price for more days ahead. In general, we develop a more accurate prediction system for the short-term stock price prediction.

From a systematic prediction system practical implementation perspective, our proposed system can be used in a number of different ways. First, it allows the investors to get the prediction required data automatically pulled through the system. Second, investors can construct different prediction results for various targets, which could assist them to make decisions through a simple voting procedure. Third, our prediction system can be used to predict different periods ahead price. This could help investors to maximize their profit in the trading. The study is not without limitations. Our data were gathered over time, the Gaussian assumption may not be quite appropriate. Some higher-order techniques (i.e., using information which is not contained in the covariance matrix), or methods using non-linear transformations might be more appropriate in that case. For the future work, an effective way to improve the neural network might be using different architectures that more suitable for the stock market data, such as the recurrent neural networks which consider the time effect while connecting neuron layers. In addition, researchers could use our prediction system to minimize the investment risk and maximize returns. In summary, this paper proposed a novel framework for predicting the short-term stock price and significantly improved the prediction performance. We provide our code to allow researchers to replicate and extend our work at `https://github.com/bzw0018/SRP_code`.

Chapter 5

Conclusion and Summary of Dissertation Contributions

The main objective of my dissertation is to develop a more adaptive and effective stock prediction system by applying machine learning techniques. The three sequential papers prove the successfully of my proposed approach. A systematic prediction tool is developed could be used to assist invested make more accurate decision in their stock market investment. Our prediction system integrate the stock movement forecasting and stock price forecasting. In addition, numbers of visualization are provided to enhance our system. Various external online data sources are used include: Google Trends, Wikipedia, Google Search, Financial news, Technical indicators, Macroeconomic indicators. The models using the features from these external sources along with the traditional stock market data improve the performance for the stock market prediction. Specifically, we got more than 85% accuracy for the movement prediction and less than 1% MAPE for the price prediction.

References

Abdullah, M., & Ganapathy, V. (2000). Neural network ensemble for financial trend prediction. In *Tencon 2000. proceedings* (Vol. 3, pp. 157–161).

Abu-Mostafa, Y. S., & Atiya, A. F. (1996). Introduction to financial forecasting. *Applied Intelligence*, *6*(3), 205–213.

Adebiyi, A. A., Adewumi, A. O., & Ayo, C. K. (2014). Comparison of arima and artificial neural networks models for stock price prediction. *Journal of Applied Mathematics*, *2014*.

Adya, M., & Collopy, F. (1998). How effective are neural networks at forecasting and prediction? a review and evaluation. *Journal of Forecasting*, *17*, 481–495.

Ahangar, R. G., Yahyazadehfar, M., & Pournaghshband, H. (2010). The comparison of methods artificial neural network with linear regression using specific variables for prediction stock price in tehran stock exchange. *International Journal of Computer Science and Information Security*, *7*(2), 38–46.

Alavi, M., & Leidner, D. E. (2001). Review: Knowledge management and knowledge management systems: Conceptual foundations and research issues. *MIS quarterly*, 107–136.

Alkhatib, K., Najadat, H., Hmeidi, I., & Shatnawi, M. K. A. (2013). Stock price prediction using k-nearest neighbor (knn) algorithm. *International Journal of Business, Humanities and Technology*, *3*(3), 32–44.

Arlot, S., Celisse, A., et al. (2010). A survey of cross-validation procedures for model selection. *Statistics Surveys*, *4*, 40–79.

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.

Bao, D., & Yang, Z. (2008). Intelligent stock trading system by turning point confirming and probabilistic reasoning. *Expert Systems with Applications*, *34*(1), 620–627.

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. In *Proceedings of the fifth annual workshop on computational learning theory* (pp. 144–152).

Boyacioglu, M. A., & Avci, D. (2010). An adaptive network-based fuzzy inference system (anfis) for the prediction of stock market return: the case of the istanbul stock exchange. *Expert Systems with Applications*, *37*(12), 7908–7912.

Breiman, L. (1996a). Bagging predictors. *Machine Learning*, *24*(2), 123–140.

Breiman, L. (1996b). Bias, variance, and arcing classifiers. *Technical Report 460*.

Breiman, L. (2001). Random forests. *Machine Learning*, *45*(1), 5–32.

Breiman, L., Friedman, J., Stone, C. J., & Olshen, R. A. (1984). *Classification and regression trees*. CRC press.

Case, K. E., Quigley, J. M., & Shiller, R. J. (2005). Comparing wealth effects: the stock market versus the housing market. *Advances in Macroeconomics*, *5*(1), 1-23.

Chang, T.-S. (2011). A comparative study of artificial neural networks, and decision trees for digital game content stocks price prediction. *Expert Systems with Applications*, *38*(12), 14846–14851.

Chen, M.-Y., Chen, D.-R., Fan, M.-H., & Huang, T.-Y. (2013). International transmission of stock market movements: an adaptive neuro-fuzzy inference system for analysis of taiex forecasting. *Neural Computing and Applications*, *23*(1), 369–378.

Chen, N.-F., Roll, R., & Ross, S. A. (1986). Economic forces and the stock market. *Journal of Business*, 383–403.

Chen, Y., Yang, B., & Abraham, A. (2007). Flexible neural trees ensemble for stock index modeling. *Neurocomputing*, *70*(4), 697–703.

Choi, H., & Varian, H. (2012). Predicting the present with google trends. *Economic Record*, *88*, 2 - 9.

Chourmouziadis, K., & Chatzoglou, P. D. (2016). An intelligent short term stock trading fuzzy system for assisting investors in portfolio management. *Expert Systems with Applications*, *43*, 298–311.

Cootner, P. H. (1964). The random character of stock market prices.

Cortes, C., & Vapnik, V. (1995). Support vector machine [j]. *Machine learning*, *20*(3), 273–297.

Dag, A., Topuz, K., Oztekin, A., Bulur, S., & Megahed, F. M. (2016a). A preoperative recipient-donor heart transplant survival score. *Decision Support Systems*, -. Retrieved from `http://dx.doi.org/10.1016/j.dss.2016.02.007`

Dag, A., Topuz, K., Oztekin, A., Bulur, S., & Megahed, F. M. (2016b). A probabilistic data-driven framework for scoring the preoperative recipient-donor heart transplant survival. *Decision Support Systems*, *86*, 1–12.

Dase, R., & Pawar, D. (2010). Application of artificial neural network for stock market predictions: A review of literature. *International Journal of Machine Intelligence*, *2*(2), 14–17.

Da Silva, N. F., Hruschka, E. R., & Hruschka, E. R. (2014). Tweet sentiment analysis with classifier ensembles. *Decision Support Systems*, *66*, 170–179.

de Oliveira, F. A., Nobre, C. N., & Zarate, L. E. (2013). Applying artificial neural networks to prediction of stock price and improvement of the directional prediction index–case study of petr4, petrobras, brazil. *Expert Systems with Applications*, *40*(18), 7596–7606.

Diebold, F. X., & Nason, J. A. (1990). Nonparametric exchange rate prediction? *Journal of international Economics*, *28*(3-4), 315–332.

Dietterich, T. G. (2000a). Ensemble methods in machine learning. In *International workshop on multiple classifier systems* (pp. 1–15).

Dietterich, T. G. (2000b). An experimental comparison of three methods for constructing ensembles of decision trees: Bagging, boosting, and randomization. *Machine learning*, *40*(2), 139–157.

Drucker, H., Burges, C. J., Kaufman, L., Smola, A., Vapnik, V., et al. (1997). Support vector regression machines. *Advances in neural information processing systems*, *9*, 155–161.

Drucker, H., Cortes, C., Jackel, L. D., LeCun, Y., & Vapnik, V. (1994). Boosting and other ensemble methods. *Neural Computation*, *6*(6), 1289–1301.

Enke, D., & Thawornwong, S. (2005). The use of data mining and neural networks for forecasting stock market returns. *Expert Systems with Applications*, *29*(4), 927–940.

Fama, E. F. (1965). The behavior of stock-market prices. *The Journal of Business*, *38*(1), 34-105. Retrieved from `http://www.jstor.org/stable/2350752`

Fama, E. F. (1991). Efficient capital markets: Ii. *The journal of finance*, *46*(5), 1575–1617.

Fama, E. F. (1995). Random walks in stock market prices. *Financial analysts journal*, *51*(1), 75–80.

Fama, E. F., Fisher, L., Jensen, M. C., & Roll, R. (1969). The adjustment of stock prices to new information. *International economic review*, *10*(1), 1–21.

Flannery, M. J., & Protopapadakis, A. A. (2002). Macroeconomic factors do influence aggregate stock returns. *Review of Financial Studies*, *15*(3), 751–782.

Fodor, I. K. (2002). A survey of dimension reduction techniques. *Center for Applied Scientific Computing, Lawrence Livermore National Laboratory*, *9*, 1–18.

French, K. R., Schwert, G. W., & Stambaugh, R. F. (1987). Expected stock returns and volatility. *Journal of financial Economics*, *19*(1), 3–29.

Freund, Y. (1990). Boosting a weak learning algorithm by majority. In *Colt* (Vol. 90, pp. 202–216).

Freund, Y., & Schapire, R. E. (1995). A desicion-theoretic generalization of on-line learning and an application to boosting. In *European conference on computational learning theory* (pp. 23–37).

Friedman, J., Hastie, T., & Tibshirani, R. (2001). *The elements of statistical learning* (Vol. 1). Springer series in statistics Springer, Berlin.

Gabrielsson, P., & Johansson, U. (2015). High-frequency equity index futures trading using recurrent reinforcement learning with candlesticks. In *Computational intelligence, 2015 ieee symposium series on* (pp. 734–741).

Geva, T., & Zahavi, J. (2014). Empirical evaluation of an automated intraday stock recommendation system incorporating both market data and textual news. *Decision support systems*, *57*, 212–223.

Göçken, M., Özçalıcı, M., Boru, A., & Dosdoğru, A. T. (2016). Integrating metaheuristics and artificial neural networks for improved stock price prediction. *Expert Systems with Applications*, *44*, 320–331.

Goodfellow, I., Bengio, Y., & Courville, A. (2016). *Deep learning*. MIT Press. (`http://www.deeplearningbook.org`)

Gottschlich, J., & Hinz, O. (2014a). A decision support system for stock investment recommendations using collective wisdom. *Decision support systems*, *59*, 52–62.

Gottschlich, J., & Hinz, O. (2014b). A decision support system for stock investment recommendations using collective wisdom. *Decision support systems*, *59*, 52–62.

Grudnitski, G., & Osburn, L. (1993). Forecasting s&p and gold futures prices: An application of neural networks. *Journal of Futures Markets*, *13*(6), 631–643.

Guresen, E., Kayakutlu, G., & Daim, T. U. (2011). Using artificial neural network models in stock market index prediction. *Expert Systems with Applications*, *38*(8), 10389–10397.

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.

Hamao, Y., Masulis, R. W., & Ng, V. (1990). Correlations in price changes and volatility across international stock markets. *Review of Financial Studies*, *3*(2), 281–307.

Hamid, S. A., & Iqbal, Z. (2004). Using neural networks for forecasting volatility of s&p

500 index futures prices. *Journal of Business Research*, *57*(10), 1116–1125.

Han, J., Kamber, M., & Pei, J. (2011). *Data mining: concepts and techniques*. Elsevier.

Hassan, M. R., Nath, B., & Kirley, M. (2007). A fusion model of hmm, ann and ga for stock market forecasting. *Expert Systems with Applications*, *33*(1), 171–180.

Hastie, T. J., Tibshirani, R. J., & Friedman, J. H. (2011). *The elements of statistical learning: data mining, inference, and prediction*. Springer.

Hendler, J. (2014). Data integration for heterogenous datasets. *Big data*, *2*(4), 205–215.

Hondroyiannis, G., & Papapetrou, E. (2001). Macroeconomic influences on the stock market. *Journal of Economics and Finance*, *25*(1), 33–49.

Hotelling, H. (1933). Analysis of a complex of statistical variables into principal components. *Journal of educational psychology*, *24*(6), 417.

Huang, W., Nakamori, Y., & Wang, S.-Y. (2005). Forecasting stock market movement direction with support vector machine. *Computers & Operations Research*, *32*(10), 2513–2522.

Ince, H., & Trafalis, T. B. (2008). Short term forecasting with support vector machines and application to stock price prediction. *International Journal of General Systems*, *37*(6), 677–687.

Jasemi, M., Kimiagari, A. M., & Memariani, A. (2011). A modern neural network model to do stock market timing on the basis of the ancient investment technique of japanese candlestick. *Expert Systems with Applications*, *38*(4), 3884–3890.

Johnson, K. J., Mark A.and Watson. (2011). Can changes in the purchasing managers index foretell stock returns? an additional forward-looking sentiment indicator. *The Journal of Investing*, *20*(4), 89-98.

Kao, L.-J., Chiu, C.-C., Lu, C.-J., & Chang, C.-H. (2013). A hybrid approach by integrating wavelet-based feature extraction with mars and svr for stock index forecasting. *Decision Support Systems*, *54*(3), 1228–1244.

Kao, L.-J., Chiu, C.-C., Lu, C.-J., & Yang, J.-L. (2013). Integration of nonlinear independent component analysis and support vector regression for stock price forecasting. *Neurocomputing*, *99*, 534–542.

Kara, Y., Boyacioglu, M. A., & Baykan, Ö. K. (2011). Predicting direction of stock price index movement using artificial neural networks and support vector machines: The sample of the istanbul stock exchange. *Expert systems with Applications*, *38*(5), 5311–5319.

Kazem, A., Sharifi, E., Hussain, F. K., Saberi, M., & Hussain, O. K. (2013). Support vector regression with chaos-based firefly algorithm for stock market price forecasting. *Applied Soft Computing*, *13*(2), 947–958.

Kearns, M., & Valiant, L. (1994). Cryptographic limitations on learning boolean formulae and finite automata. *Journal of the ACM (JACM)*, *41*(1), 67–95.

Kearns, M. J., & Valiant, L. G. (1988). *Learning boolean formulae or finite automata is as hard as factoring*. Harvard University, Center for Research in Computing Technology, Aiken Computation Laboratory.

Keim, D. B., & Stambaugh, R. F. (1986). Predicting returns in the stock and bond markets. *Journal of financial Economics*, *17*(2), 357–390.

Khansa, L., & Liginlal, D. (2011). Predicting stock market returns from malicious attacks: A comparative analysis of vector autoregression and time-delayed neural networks. *Decision Support Systems*, *51*(4), 745–759.

Kiersz, A. (2015, 03). *Here's how badly warren buffett beat the market.* http://www.businessinsider.com/warren-buffett-berkshire-hathaway-vs-sp-500-2015-3.

Kilian, L., & Park, C. (2009). The impact of oil price shocks on the us stock market. *International Economic Review*, *50*(4), 1267–1287.

Kim, H.-j., & Shin, K.-s. (2007). A hybrid approach based on neural networks and genetic algorithms for detecting temporal patterns in stock markets. *Applied Soft Computing*,

7(2), 569–576.

Kim, K.-j. (2003). Financial time series forecasting using support vector machines. *Neurocomputing*, *55*(1), 307–319.

Kim, K.-j., & Han, I. (2000). Genetic algorithms approach to feature discretization in artificial neural networks for the prediction of stock price index. *Expert systems with Applications*, *19*(2), 125–132.

Kim, Y. (2006). Toward a successful crm: variable selection, sampling, and ensemble. *Decision Support Systems*, *41*(2), 542–553.

Kimoto, T., Asakawa, K., Yoda, M., & Takeoka, M. (1990). Stock market prediction system with modular neural networks. In *Neural networks, 1990., 1990 ijcnn international joint conference on* (pp. 1–6).

Kohavi, R. (1995). A study of cross-validation and bootstrap for accuracy estimation and model selection. In *International joint conference on artificial intelligence, 1995* (p. 1137-1143).

Kohavi, R., & John, G. H. (1997). Wrappers for feature subset selection. *Artificial Intelligence*, *97*(1), 273–324.

Kryzanowski, L., Galler, M., & Wright, D. W. (1993). Using artificial neural networks to pick stocks. *Financial Analysts Journal*, *49*(4), 21–27.

Kuhn, M. (2008). Caret package. *Journal of Statistical Software*, *28*(5).

Lai, R. K., Fan, C.-Y., Huang, W.-H., & Chang, P.-C. (2009). Evolving and clustering fuzzy decision tree for financial time series data forecasting. *Expert Systems with Applications*, *36*(2), 3761–3773.

Lawrence, R. (1997). Using neural networks to forecast stock market prices. *University of Manitoba*.

LeCun, Y., Bengio, Y., & Hinton, G. (2015). Deep learning. *Nature*, *521*(7553), 436–444.

Lee, K., & Jo, G. (1999). Expert system for predicting stock market timing using a candlestick chart. *Expert Systems with Applications*, *16*(4), 357-364.

Leray, P., & Gallinari, P. (1999). Feature selection with neural networks. *Behaviormetrika*, *26*(1), 145–166.

Lewis, M. (2015). *The big short: Inside the doomsday machine (movie tie-in)*. WW Norton & Company.

Li, Q., Chen, Y., Jiang, L. L., Li, P., & Chen, H. (2016). A tensor-based information framework for predicting the stock market. *ACM Transactions on Information Systems (TOIS)*, *34*(2), 11.

Li, S.-T., & Kuo, S.-C. (2008). Knowledge discovery in financial investment for forecasting and trading strategy through wavelet-based {SOM} networks. *Expert Systems with Applications*, *34*(2), 935 - 951. doi: http://dx.doi.org/10.1016/j.eswa.2006.10.039

Lin, X., Yang, Z., & Song, Y. (2009). Short-term stock price prediction based on echo state networks. *Expert systems with applications*, *36*(3), 7313–7317.

Lin, X., Yang, Z., & Song, Y. (2011). Intelligent stock trading system based on improved technical analysis and echo state network. *Expert systems with Applications*, *38*(9), 11347–11354.

Loomis, C. J. (2012, 02). *Buffett beats the sp for the 39th year.* http://fortune.com/2012/02/25/buffett-beats-the-sp-for-the-39th-year/.

Lu, C.-J., Lee, T.-S., & Chiu, C.-C. (2009). Financial time series forecasting using independent component analysis and support vector regression. *Decision Support Systems*, *47*(2), 115–125.

Maclin, R., & Opitz, D. (2011). Popular ensemble methods: An empirical study. *Journal of Artificial Intelligence Research*, *11*, 169–198.

Mahajan, A., Dey, L., & Haque, S. M. (2008). Mining financial news for major events and their impacts on the market. *Proceedings of the 2008 IEEE/WIC/ACM International Conference on Web Intelligence and Intelligent Agent Technology*, *1*, 423–426.

Malkiel, B. G. (2003). The efficient market hypothesis and its critics. *The Journal of Economic Perspectives*, *17*(1), 59–82.

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

McCulloch, W. S., & Pitts, W. (1943). A logical calculus of the ideas immanent in nervous activity. *The bulletin of mathematical biophysics*, *5*(4), 115–133.

Meesad, P., & Rasel, R. I. (2013). Predicting stock market price using support vector regression. In *Informatics, electronics & vision (iciev), 2013 international conference on* (pp. 1–6).

Megahed, F. M., & Jones-Farmer, L. A. (2015). Frontiers in statistical quality control 11. In S. Knoth & W. Schmid (Eds.), (pp. 29–47). Cham: Springer International Publishing. doi: 10.1007/978-3-319-12355-4_3

Meinshausen, N. (2006). Quantile regression forests. *Journal of Machine Learning Research*, *7*(Jun), 983–999.

Mladenić, D., & Grobelnik, M. (2003). Feature selection on hierarchy of web documents. *Decision Support Systems*, *35*(1), 45–87.

Moat, H. S., Curme, C., Avakian, A., Kenett, D. Y., Stanley, H. E., & Preis, T. (2013). Quantifying wikipedia usage patterns before stock market moves. *Scientific reports*, *3*.

Mok, P., Lam, K., & Ng, H. (2004). An ica design of intraday stock prediction models with automatic variable selection. In *Neural networks, 2004. proceedings. 2004 ieee international joint conference on* (Vol. 3, pp. 2135–2140).

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

Nassirtoussi, A. K., Aghabozorgi, S., Wah, T. Y., & Ngo, D. C. L. (2014). Text mining for market prediction: A systematic review. *Expert Systems with Applications*, *41*(16), 7653–7670.

Nguyen, T. H., Shirai, K., & Velcin, J. (2015). Sentiment analysis on social media for stock movement prediction. *Expert Systems with Applications*, *42*(24), 9603–9611.

Nofsinger, J. R. (2005). Social mood and financial economics. *The Journal of Behavioral Finance*, *6*(3), 144–160.

Opitz, D., & Maclin, R. (1999). Popular ensemble methods: An empirical study. *Journal of Artificial Intelligence Research*, 169–198.

Ou, P., & Wang, H. (2009). Prediction of stock market index movement by ten data mining techniques. *Modern Applied Science*, *3*(12), 28.

Pai, P.-F., & Lin, C.-S. (2005). A hybrid arima and support vector machines model in stock price forecasting. *Omega*, *33*(6), 497–505.

Park, J., & Ratti, R. A. (2008). Oil price shocks and stock markets in the us and 13 european countries. *Energy Economics*, *30*(5), 2587–2608.

Peason, K. (1901). On lines and planes of closest fit to systems of point in space. *Philosophical Magazine*, *2*(11), 559–572.

Poon, S.-H., & Granger, C. W. (2003). Forecasting volatility in financial markets: A review. *Journal of economic literature*, *41*(2), 478–539.

Prechter Jr, R. R., & Parker, W. D. (2007). The financial/economic dichotomy in social behavioral dynamics: the socionomic perspective. *The Journal of Behavioral Finance*, *8*(2), 84–108.

Preis, T., Moat, H. S., & Stanley, H. E. (2013). Quantifying trading behavior in financial markets using google trends. *Scientific Reports*, *3*, 1684.

Qian, B., & Rasheed, K. (2007). Stock market prediction with multiple classifiers. *Applied Intelligence*, *26*(1), 25–33.

Quinlan, J. R. (1986). Induction of decision trees. *Machine learning*, *1*(1), 81–106.

Quinlan, J. R. (1996). Bagging, boosting, and c4. 5. *AAAI/IAAI, Vol. 1*, 725–730.

Quinlan, J. R. (2014). *C4. 5: programs for machine learning*. Elsevier.

R Core Team. (2016). R: A language and environment for statistical computing [Computer software manual]. Vienna, Austria. Retrieved from `https://www.R-project.org/`

Rahman, A. A., Sidek, N. Z. M., & Tafri, F. H. (2009). Macroeconomic determinants of

malaysian stock market. *African Journal of Business Management*, *3*(3), 95.

Raymond McTaggart, Gergely Daroczi, & Clement Leung. (2016). Quandl: Api wrapper for quandl.com [Computer software manual]. Retrieved from `https://CRAN.R-project.org/package=Quandl` (R package version 2.8.0)

Rumelhart, D. E., Hinton, G. E., & Williams, R. J. (1985). *Learning internal representations by error propagation* (Tech. Rep.). DTIC Document.

Russell, S., Norvig, P., & Intelligence, A. (1995). A modern approach. *Artificial Intelligence. Prentice-Hall, Egnlewood Cliffs*, *25*, 27.

Sadorsky, P. (1999). Oil price shocks and stock market activity. *Energy Economics*, *21*(5), 449–469.

Schapire, R. E. (1990). The strength of weak learnability. *Machine learning*, *5*(2), 197–227.

Schapire, R. E. (2003). The boosting approach to machine learning: An overview. In *Nonlinear estimation and classification* (pp. 149–171). Springer.

Schapire, R. E., Freund, Y., Bartlett, P., & Lee, W. S. (1998). Boosting the margin: A new explanation for the effectiveness of voting methods. *Annals of Statistics*, *26*, 1651–1686.

Schöneburg, E. (1990). Stock price prediction using neural networks: A project report. *Neurocomputing*, *2*(1), 17–27.

Schumaker, R. P. (2013). Machine learning the harness track: Crowdsourcing and varying race history. *Decision Support Systems*, *54*(3), 1370–1379.

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

Scikit-Learn-Developers. (2014). *1.13 feature selection - scikit-learn documentation.* http://goo.gl/GDedwn.

Serneels, S., De Nolf, E., & Van Espen, P. J. (2006). Spatial sign preprocessing: a simple way to impart moderate robustness to multivariate estimators. *Journal of Chemical*

Information and Modeling, *46*(3), 1402–1409.

Shynkevich, Y., McGinnity, T. M., Coleman, S., & Belatreche, A. (2015). Stock price prediction based on stock-specific and sub-industry-specific news articles. In *Neural networks (ijcnn), 2015 international joint conference on* (pp. 1–8).

Smith, V. L. (2003). Constructivist and ecological rationality in economics. *The American Economic Review*, *93*(3), 465–508.

Szegö, G. (2002). Measures of risk. *Journal of Banking & Finance*, *26*(7), 1253–1272.

Taylor, J. W. (2000). A quantile regression neural network approach to estimating the conditional density of multiperiod returns. *Journal of Forecasting*, *19*(4), 299–311.

Tetlock, P. C. (2007). Giving content to investor sentiment: The role of media in the stock market. *The Journal of Finance*, *62*(3), 1139–1168.

Ticknor, J. L. (2013). A bayesian regularized artificial neural network for stock market forecasting. *Expert Systems with Applications*, *40*(14), 5501–5506.

Trafalis, T. B., & Ince, H. (2000). Support vector machine for regression and applications to financial forecasting. In *Neural networks, 2000. ijcnn 2000, proceedings of the ieee-inns-enns international joint conference on* (Vol. 6, pp. 348–353).

Tsai, C.-F., & Hsiao, Y.-C. (2010). Combining multiple feature selection methods for stock prediction: Union, intersection, and multi-intersection approaches. *Decision Support Systems*, *50*(1), 258–269.

Tsai, C.-F., Lin, Y.-C., Yen, D. C., & Chen, Y.-M. (2011). Predicting stock returns by classifier ensembles. *Applied Soft Computing*, *11*(2), 2452–2459.

Ulrich, J. (2016). Ttr: Technical trading rules [Computer software manual]. Retrieved from `https://CRAN.R-project.org/package=TTR` (R package version 0.23-1)

Vaisla, K. S., & Bhatt, A. K. (2010). An analysis of the performance of artificial neural network technique for stock market forecasting. *International Journal on Computer Science and Engineering*, *2*(6), 2104–2109.

Valiant, L. G. (1984). A theory of the learnable. *Communications of the ACM*, *27*(11),

1134–1142.

Vapnik, V. N., & Chervonenkis, A. J. (1974). *Theory of pattern recognition*. Nauka.

Vu, T.-T., Chang, S., Ha, Q. T., & Collier, N. (2012). An experiment in integrating sentiment features for tech stock prediction in twitter.

Wang, J.-J., Wang, J.-Z., Zhang, Z.-G., & Guo, S.-P. (2012). Stock index forecasting based on a hybrid model. *Omega*, *40*(6), 758–766.

Wang, J.-Z., Wang, J.-J., Zhang, Z.-G., & Guo, S.-P. (2011). Forecasting stock indices with back propagation neural network. *Expert Systems with Applications*, *38*(11), 14346–14355.

Wang, Y.-F. (2002). Predicting stock price using fuzzy grey prediction system. *Expert Systems with Applications*, *22*(1), 33 - 38. doi: http://dx.doi.org/10.1016/S0957-4174(01)00047-1

Weng, B., Ahmed, M. A., & Megahed, F. M. (2017). Stock market one-day ahead movement prediction using disparate data sources. *Expert Systems with Applications*, -. Retrieved from `http://www.sciencedirect.com/science/article/pii/S0957417417301331` doi: http://dx.doi.org/10.1016/j.eswa.2017.02.041

Werbos, P. (1974). *Beyond regression: New tools for prediction and analysis in the behavioral sciences* (Unpublished doctoral dissertation). Harvard University.

Wiesmeier, M., Barthold, F., Blank, B., & Kögel-Knabner, I. (2011). Digital mapping of soil organic matter stocks using random forest modeling in a semi-arid steppe ecosystem. *Plant and Soil*, *340*(1-2), 7–24.

Woschnagg, E., & Cipan, J. (2004). Evaluating forecast accuracy. *University of Vienna, Department of Economics*.

Yang, H., Chan, L., & King, I. (2002). Support vector machine regression for volatile stock market prediction. In *Intelligent data engineering and automated learningideal 2002* (pp. 391–396). Springer.

Zekic, M. (1998). Neural network applications in stock market predictions-a methodology analysis. In *proceedings of the 9th international conference on information and intelligent systems* (Vol. 98, pp. 255–263).

Zhai, Y., Hsu, A., & Halgamuge, S. K. (2007). Combining news and technical indicators in daily stock price trends prediction. In *International symposium on neural networks* (pp. 1087–1096).

Zhang, Y., & Wu, L. (2009). Stock market prediction of s&p 500 via combination of improved bco approach and bp neural network. *Expert systems with applications*, *36*(5), 8849–8854.