THE REPUBLIC OF TURKEY
BAHCESEHIR UNIVERSITY

STOCK MARKET PREDICTION BY COMBINING STOCK PRICE INFORMATION AND SENTIMENT ANALYSIS

Master's Thesis

ADNAN GÜMÜŞ

ISTANBUL, 2019

THE REPUBLIC OF TURKEY
BAHCESEHIR UNIVERSITY

THE GRADUATE SCHOOL OF NATURAL AND
APPLIED SCIENCES
COMPUTER ENGINEERING

STOCK MARKET PREDICTION BY COMBINING STOCK PRICE INFORMATION AND SENTIMENT ANALYSIS

Master's Thesis

ADNAN GÜMÜŞ

Supervisor: ASSIST. PROF. DR. C. OKAN ŞAKAR

ISTANBUL, 2019

THE REPUBLIC OF TURKEY
BAHCESEHIR UNIVERSITY

GRADUATE SCHOOL OF NATURAL AND APPLIED SCIENCES
COMPUTER ENGINEERING

Name of the thesis : Stock Market Prediction By Combining Stock Price Information And Sentiment Analysis
Name/Last Name of the Student : Adnan GÜMÜŞ
Date of the Defense of Thesis : 19/08/2019

The thesis has been approved by the Graduate School of Natural and Applied Sciences.

Assist. Prof. Dr. Yücel Batu SALMAN
Graduate School Director

I certify that this thesis meets all the requirements as a thesis for the degree of Master of Science.

Assist. Prof. Dr. Tarkan Aydın
Program Coordinator

This is to certify that we have read this thesis and we find it fully adequate in scope, quality and content, as a thesis for the degree of Master of Science.

Examining Comittee Members ________Signature____ _____

Thesis Supervisor ----------------------------------
Assist. Prof. Dr. C. Okan ŞAKAR

Member ----------------------------------
Prof. Dr. Alper TUNGA

Member ----------------------------------
Assist. Prof. Dr. Burak PARLAK

ACKNOWLEDGEMENT

I would like to give special thanks to my thesis supervisor, Asist. Prof. C. Okan Şakar and I really appreciate for his all support, contribution and encouragement in the realization of this thesis.

I would like to thank Bahçeşehir University Faculty of Engineering and Natural Sciences members for sharing their experience and knowledge with me.

I would like to thank all my friends who support and encourage me.

Finally, I would like to express my gratitude to my beloved wife Yasemin Alkan GÜMÜŞ who gave me strength with her presence, without her love this work would never be possible.

İstanbul, 2019 Adnan GÜMÜŞ

ABSTRACT

STOCK MARKET PREDICTION BY COMBINING STOCK PRICE INFORMATION AND SENTIMENT ANALYSIS

Adnan GÜMÜŞ

Computer Engineering

Thesis Supervisor: Assist. Prof. Dr. C. Okan ŞAKAR

August 2019, 41 pages

Predicting the stock market instrument price is a quite precious, at the same time pretty difficult machine learning task. With an increase of data collection through the internet, data scientists aim to extract valid data points for the estimations. However, these data points are usually obtained from one data source and thus may not cover all the factors affecting the stock market. The stock market is sensitive and highly depends on the political and macroeconomic environments. Therefore, to improve the prediction for stock market closing price movements, in this thesis, we apply sentiment analysis to the tweets related with the market and specific stock that is being predicted. We use this information to predict the stock market movements by correlating them with the existing daily quantative dataset. We also use some technical indicators such as SMAVG, MACD, RSI and Sig. The obtained cumulative dataset is used in training a recurrent neural network model on the data consisting of the Istanbul Stock Exchange prices of 2010 to 2018. The output of the regression model is used in prediction of the closing price movement in order to decide whether the closing price is up or down next day. Accuracy value of the model obtained using stock prices and technical analysis attributes as input was measured as 56 percent. Then, we classified twitter news as positive, negative and neutral and integrated these values into the model. The accuracy of the combined model was increased to 66 percent. The results shows the accuracy of the model significantly increases when sentiment analysis output that contains information about the stock and market is integrated into the model.

Keywords: Recurrent Neural Network, Artificial Neural Network, Long Short-Term Memory, Technical Analysis, Sentiment Analysis

ÖZET

HİSSE FİYAT BİLGİSİ VE DUYGU ANALİZ KOMBİNASYONU İLE PAY PİYASASINDA FİYAT TAHMİNİ

Adnan GÜMÜŞ

Bilgisayar Mühendisliği

Tez Danışmanı: Dr. Öğr. Üyesi C. Okan ŞAKAR

Ağustos 2019, 41 sayfa

Pay piyasasındaki bir enstrümanın fiyatını tahmin etmek, oldukça değerli ve aynı zamanda oldukça zor makine öğrenme görevlerinden biridir. İnternet üzerinden veri toplamanın artmasıyla birlikte, veri bilim insanları tahminler için geçerli veri noktaları çıkarmayı hedeflemektedir. Ancak bu veri noktaları tek bir veri kaynağından elde edilir ve bu nedenle borsaya etki eden tüm noktaları kapsamaz. Borsa politik ve makroekonomik haberlere son derece duyarlıdır. Bu nedenle borsa kapanış fiyat hareketleri tahminini iyileştirmek için bu araştırmamızda piyasa ile ilgili tweet'lere ve öngörülen spesifik hisse senetlerine duygu analizi uyguladık. Bu bilgileri mevcut günlük niceliksel veri kümesiyle ilişkilendirerek borsa hareketlerini tahmin etmek için kullanıyoruz. Ayrıca SMAVG, MACD, RSI ve Sig gibi bazı teknik göstergeler kullanıyoruz. Elde edilen kümülatif veri seti, 2010 - 2018 arasındaki İstanbul Menkul Kıymetler Borsası fiyatlarından oluşan veriler üzerinde tekrarlayan bir sinir ağı modelinin eğitilmesi için kullanılmaktadır. Regresyon modelinin çıktıları kapanış fiyatının bir sonraki gün aşağı veya yukarı yönlü olup almayacağına karar vermek için kullanılmaktadır. Teknik analiz verilerinin girdi olarak kullanılması ile elde edilen modelin doğruluk değeri yüzde 56 olarak ölçülmüştür. Daha sonra twitter haberlerini pozitif, negatif ve tarafsız olarak sınıflandırdık ve bu değerleri modele entegre ettik. Kombine modelin doğruluğu yüzde 66'ya yükseltildi. Sonuçlar, hisse ve piyasa hakkında bilgi içeren duygu analizi çıktısının modele entegre edilmesinin, modelin doğruluğunu önemli ölçüde arttırdığını göstermektedir.

Anahtar Kelimeler: Tekrarlayan Sinir Ağı, Yapay Sinir Ağları, Uzun Kısa Süreli Bellek, Teknik Analiz Yöntemleri, Duygu Analizi

CONTENTS

TABLES

FIGURES

ABBREVIATIONS

ANN	:	Artificial Neural Network
ARMA	:	Autoregressive Moving Average
API	:	Application Programming Interface
CPU	:	Central Processing Unit
GPU	:	Graphics Processing Unit
LSTM	:	Long Short-Term Memory
MA	:	Moving Average
MACD	:	Moving Average Convergence Divergence
RELU	:	Rectified Linear Unit
RMSE	:	Root Mean Square Error
RMSprob	:	Root Mean Square Error Probability
RNN	:	Recurrent Neural Network
RSI	:	Relative Strength Index

SYMBOLS

Hidden Unit	:	h_t
Input	:	X_t

1. INTRODUCTION

Creating a model for the financial market and predicting price movements through an estimation model is a popular topic for researchers and academicians. In financial markets, buyers and sellers deal with products such as bonds, stocks and precious metals. Especially in stock exchanges, the machine learning algorithms and statistical analysis methods constitute the basis of most of the stock prediction systems. Financial forecasting is a difficult task due to its variability and nonlinear dynamics. A volume gap between the past stock price and the future price refers to many inputs which cannot be easily integrated to the system. Political and economic instant news may impact to the market due to the sensitive structure of stock markets. In this study, we aim to see the effect of financial news in the prediction of stock prices. It is known that these types of information are unstable and pretty complex to collect and integrate to the prediction model. We use sentiment analysis methods to determine how the related news will affect the price of the stock. We classify each news as positive, negative or neutral and feed this information to recurrent neural network (RNN) model as input together with the stock price. We present the experimental results on a stock price dataset collect from for Istanbul Stock Market.

1.1 THESIS SCOPE

This study aims to create a model that captures long term patterns, cyclical and non-linear movements instead of investigating high frequent trade patterns. Therefore, the prediction horizon is set to one day in our experiments. Specifically, using today’s and previous stock prices and financial news, we predict tommorrow’s closing price and direction. The purpose is to indicate the impact of sentiment analysis and technical indicators to the stock price with the use of recurrenct neural network for Istanbul Stock Market. Therefore, the focus of the study will be on building, modifying and evaluating the RNN model with and without sentiment labels of the financial news.

We perform the experiments on a specific stock that has a high trading volume in Istanbul Stock Market. First, the stock price and technical analysis indicators are given to the Long-Short Term Memory RNN model and the next day's closing price of the stock is predicted. Then, this prediction is used to evaluate the success of the model in predicting the movement of the stock as a binary classification problem with labels 'up' and 'down'. Afterwards, since the main goal is to investigate to what extent the news about the stock will be useful to generate a better prediction, a sentiment analysis model has been built on a news dataset collected in the context of this study. Then, the output of the sentiment analysis model is integrated to the stock price estimation model with data level fusion and LSTM-RNN is trained again on the combined data. The comparative results, with and without sentiment analysis, are presented on a 1-year test data.

1.2 ORGANIZATION OF THESIS

Initially, a collection of related works and the findings from them will be presented, followed by a Literature Review section describing in detail the theory behind the neural network architecture and briefly touches upon the technical indicators used in this thesis. This part can be found under Chapter 2, Literature Review. Regression model is created for the experiments which aims to predict the next day's closing price value given a training tensor of previous prices. The methods used to prepare and collect the news, stock prices and the preprocessing steps applied on the dataset will be discussed under Chapter 3, Dataset Description. The technologies used in this study and the approach followed to optimize the neural network built for regression are covered under Chapter 4, Methods. The experimental results comparing the prediction success of the model under different conditions will be presented under Chapter 5, Results. The conclusions and discussions that includes the results from regression model with sentiment analysis and technical indicators as well as their implications for the industry are given in Chapter 6.

2. BACKGROUND

In this section, we provide some background information about financial markets and prediction methods that we used in our research.

2.1 RELATED WORKS

In recent studies, it has been observed that artificial neural networks work effectively especially on financial problems. (Black & McMillan 2004), (Wood & Jasic 2004), (Bildirici & Ersin 2009), (Kara, Boyacioglu, & Baykan, 2011) and in many other studies, artificial neural networks (ANN) have shown successful results in topics such as index estimation, transaction volume, stock trends. The financial forecasting studies carried out in Turkey mainly trading volume, change the direction of share prices and focused on estimating the risk of bankruptcy (Avcı, 2007). We also use an ANN-based model in our experiments with the aim of predicting the next day’s stock price.

There are some studies that performs financial sentiment analysis using different techniques. In one of these studies, Sohangir et al. (2018) analyzed financial sentiment by using the content shared in a financial social media platform, StockTwits, where investors, entrepreneurs and traders share financial ideas. StockTwits compares shared stock forecasting to models that predict the emotions of authors using deep learning methods such as Long Short-Term Memory, doc2vec, Convolutional Neural Network (CNN), and the best results are obtained in the CNN model. A model that makes an estimation of the stock market with an accuracy rate of 90 percent shows that it is a financially significant and exciting field of study.

In another financial sentiment analysis study (Smailovic et al., 2014), the relationship between Twitter feeds and stock price changes. The results showed that sentiments in the related tweets carry significant information about the direction of the price of the related stock. They proposed a new stream-based active learning approach based on Support Vector Machines classifier to sentiment analysis and the experiments showed that the stream-based analysis of positive sentiment probability may indicate the changes in the closing prices of the stock of a company.

Day and Lee (2016) performed experiments to evaluate the effect of using different finance news providers in stock price movement using a deep-neural network architecture. They collected the news data from 4 news provides and the obtained results on the collected data showed that different financial news providers have significantly different effects to investors and their investments.

There are some studies that uses sentiment analysis of the Turkish sharings and news published in some social media platforms. In one of these studies, Şimşek and Özdemir (2012) made a study of classifying Twitter messages as happy and sad. They created a 113-word Turkish dictionary used in the expressions of happiness and sadness by using Amazon's Mechanical Turk (MTurk, 2017) service and classified them according to their frequency of passing in their Twitter messages. They analyzed the relationship between the level of happiness in the stock market index. They observed a 45 percent correlation.

Eroğul (2009) created a data set from a Turkish movie review site. The reviews were labeled by their authors with one of the positive, negative or neutral symbols. In the generated dataset, the text and icon of the review are associated with each other to produce a data item that is labeled with sensitivity. A polar dataset was created from another movie review site where users rated the movie they were reviewing. Using the Zemberek tool as a morphological separator, a combination of n-gram and POS information is used for a classification task. Regression and individual techniques are used to estimate the scores of the polarity-labeled data set. The accuracy of his model is 85 percent on Turkish movie review data.

Another study that conducted emotion analysis on Turkish tweets investigated by Demirci (2014). Instead of performing a sensitivity analysis, the author talks about emotion analysis. The categories are determined as fear, pleasure, sadness, anger, dislike and surprise. It also reveals that Turkish is a sticky language. This creates problems with the analysis because each derivative suffix has the possibility to change the meaning of the word, and therefore it is necessary to obtain the true meaning of a word that each derivative suffix needs to be examined.

Twitter is an important data source for sentiment analysis studies. In 2010, Davidov et al. conducted a classification of emotions on Twitter messages (Davidov et al., 2010). In their study, a method based on supervised machine learning used features such as hashtags and smiley characters that are specific to Twitter, in addition to the common features used in text processing in the feature vector. They have shown that using these Twitter-specific features increases the success rate.

Kang and Park conducted emotion analysis on users' comments on mobile services in 2014 (Kang and Park, 2014). In this study, they tried to determine the emotion orientations of the users about the different aspects of these services. Although the study was conducted on a feature-based level, automatic feature extraction was not performed, and emotion orientations were classified according to the characteristics determined manually. In their study, Kang and Park not only classified as positive / negative, but also graded from -2 to +2.

2.2 FINANCIAL MARKET

If we need to make a simple financial definition of the very general concept of Market we can say that there are various platforms where buyers and sellers exchange money and valuable documents in continuous communication. Exchanges, where current supply and demand are transformed into transactions in physical or digital environments are a good example of such platforms. Markets are divided into real and financial markets. While the supply and demand of goods and services in real markets are at stake, the values traded and sold in financial markets which are the subject of this article are more valuable documents. In the rest of the article, we will take a closer look at the financial markets to provide an introduction for those interested. In capital markets, financial assets with a maturity of more than one year are traded. The loans obtained from these markets are mostly used to finance the fixed assets of enterprises such as buildings, machinery, and equipment. As in the money market, resources in these markets are the savings of owners. In the capital market, fundholders may purchase documents from the first issue or second hand. So capital markets divided into two main groups, the primary market and the secondary market.

2.2.1 Primary Market

Securities that are first circulated in the market are traded in these markets. In the primary markets, firms issuing securities may come together with an intermediary with surplus funds, or directly.

2.2.2 Secondary Market

A secondary market is a market in which previously traded values are traded. The secondary market, which increases the liquidity of securities, thus creates demand for the primary market and contributes to its development.

2.3 DEFINITION OF EQUITY

Equities are valuable papers issued by capital companies to certify their shareholding in the capital and enable their owners to benefit from all kinds of partnership rights. Companies wishing to issue shares must first obtain approval from the CMB (Capital Markets Board). The organizations that have the authority to issue shares can be listed as follows: Anonymous companies, Commandite companies with share capital, Institutions established by special law (T.C Central Bank, insurance companies, etc.) Stocks that meet the requirements of the CMB begin to be traded on the Istanbul Stock Exchange (ISE). The securities that have started to be traded can be bought and sold by investors after this point. Since the shares purchased represent a portion of the company's capital; it provides shareholders with certain shareholding rights, such as voting in the company, participating in capital increases, and receiving dividends.

2.4 FINANCIAL PREDICTION

One of the most important elements of today's decision-making world in both public and private sectors is the estimation of macroeconomic and financial variables. Over the past few decades, econometric model-based estimation has become quite popular in the private and public decision-making process. To better understand the meaning of the Time Series Estimation, divide the term into two parts, Time series is a series of consecutive observations over time. The prediction means predicting a future event. When a prediction is made on a time series data such as events occurring at a time interval, the time series is called the estimate. Time series prediction is the process of predicting future events based on historical data. Time series estimation has been used in many industries for some time; It is widely used in all sectors to guide future decisions, for example in retail sales estimates, so that raw materials can be supplied accordingly. The most famous example is the weather forecast; past and future changes can be estimated according to the future. These estimates are very important and are often the first step to solve the other problem, because they are planning generations of power to avoid unnecessary power outages or overproduction.

In any prediction scenario, there are three questions you always want to ask yourself before creating the forecasting model:

What is the time horizon for my estimates? What is the temporal frequency required for my estimates? Can the forecasts be updated frequently over time, or should they only be generated once and remain static over time?

The answer to these three questions will help us identify the most critical components of the time series: Trend: A long-term component that defines a gradual increase / decrease in your series.

Loop: Long-term component defining series swings.

Seasonal: A normal component that observes the relatively short-term effects of the series.

Error: Random variability in observations that cannot be explained by the model.

2.5 RECURRENT NEURAL NETWORKS

In order to understand Recurrent Neural Networks (RNN), we have to first understand the structure of neural networks using feedforward. We can easily say that the information receiving from the neurons to both networks produces output by applying some mathematical operations. In feedforward structures, incoming information is processed only in forward direction without a loop structure. In this architecture, the input data is passed through the network and an output is produced in the final layer. In a feedforward network, training should continue until the error value in categorizing the inputs is sufficiently reduced. Thus, the weights to neurons are updated after each epoch and a structure is formed that can categorize the given input. RNN, on the other hand, evaluates inputs based on previous inputs in a sequential manner. In addition to the input data, the content units that represent the previous output also affect the network. The decision for the input at the moment “t-1” also impacts the decision to be made at time “t”. So, in these networks, inputs produce output by combining current and previous information. Recurrent structures are separated from feedforward structures because they use their outputs as input in the next process. We can say that recurrent networks have a memory. The reason for adding memory to a network is that in some problems the order of the received input set has a meaning for the output. For these kind of datasets, feedforward networks may not produce satisfactory results since they do not take the sequential information into account while building the model. Recurrent networks are used to understand the structure of incoming data in a certain order, such as time dependent on various sensors, stock prices or statistical data.

2.6 LSTM CELLS

LSTM is developed by Sepp Hochreiter and Juergen Schmidhuber in 1997 while solving vanishing gradient problem. Then it is improved and popularized with the contribution of many people, currently it has widely usage area.

Figure 2.1: Standard RNN contains a single layer

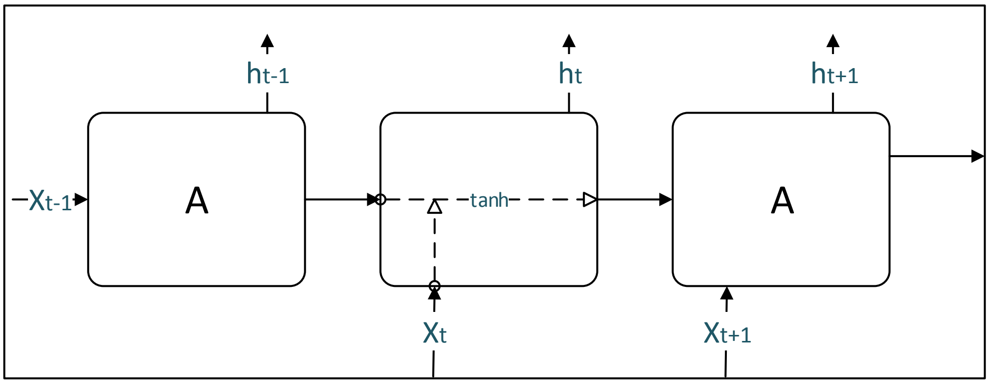


Figure 2.1 indicates standart RNN which includes a single layer. Rather than a single neural network layer, there are 4 layers connected in a particular way. These layers are also called gates. Out of the normal flow, it is a structure that receives information from outside. This information can be stored, written to the cell and readable. Using this gates, an LSTM cell decides what to store, when to allow it to read, write or delete. These Gates have a network structure and activation function. Just like in neurons, it passes or stops the incoming information according to its weight. These weights are calculated during the learning of the recurrent network. With this structure, the cell learns whether it will receive or release data. LSTM is available in different models according to the variety of needs. These are provided by receiving the inputs of the above mentioned gates from different places or sending their outputs to different places.

2.7 FINANCIAL MARKET PREDICTION USING RNN

In these days, developing information technologies, increasing information sources, data multiplicity and diversity, have created complex decision environments. The term stock analysis covers the studies performed in order to predict market movements if we make a general definition. Although there is no way of predicting what will happen in the market with a hunderent percent guarantee but investors wishing to maximize profit from stock trading will need a comprehensive stock analysis to anticipate the possible changes as accurately as possible. basic analysis and technical analysis are two best known and preferred stock analysis methods by the investors. However, it is insufficient at some points. Therefore, we will see the effect of the combination of artificial neural networks with technical analysis indicators on the prediction. Artificial neural network algorithms are turned to the most popular trends in machine learning and have applications to many areas, including driverless cars and robotics, speech and image recognition, financial forecasting.

A set of algorithms creates neural networks that are designed to recognize patterns. Recurrent Neural Networks (RNN) uses sequential information, this is the idea behind of this algorithm. All inputs are independent in a conventional neural network. But the best idea is not for many tasks. RNNs are called repeating because they perform the same task for each element of an array and the output depends on the previous calculations.

RNNs are successful in time-based problems due to the attribute of linking and understanding with the past. However, in RNNs, it is not known which activities will be remembered and how long. All information is kept within the model. While some information is important for activities, some information is unnecessary. Therefore, in the classification of certain activities, there is no need to keep the whole history.

In theory, RNNs have the capacity to remember long-term activities, as they repeat themselves in interior architecture. However, the parameters must be selected carefully to be remembered. Since such a parameter selection is practically impossible, RNNs cannot learn (remember) the long history.

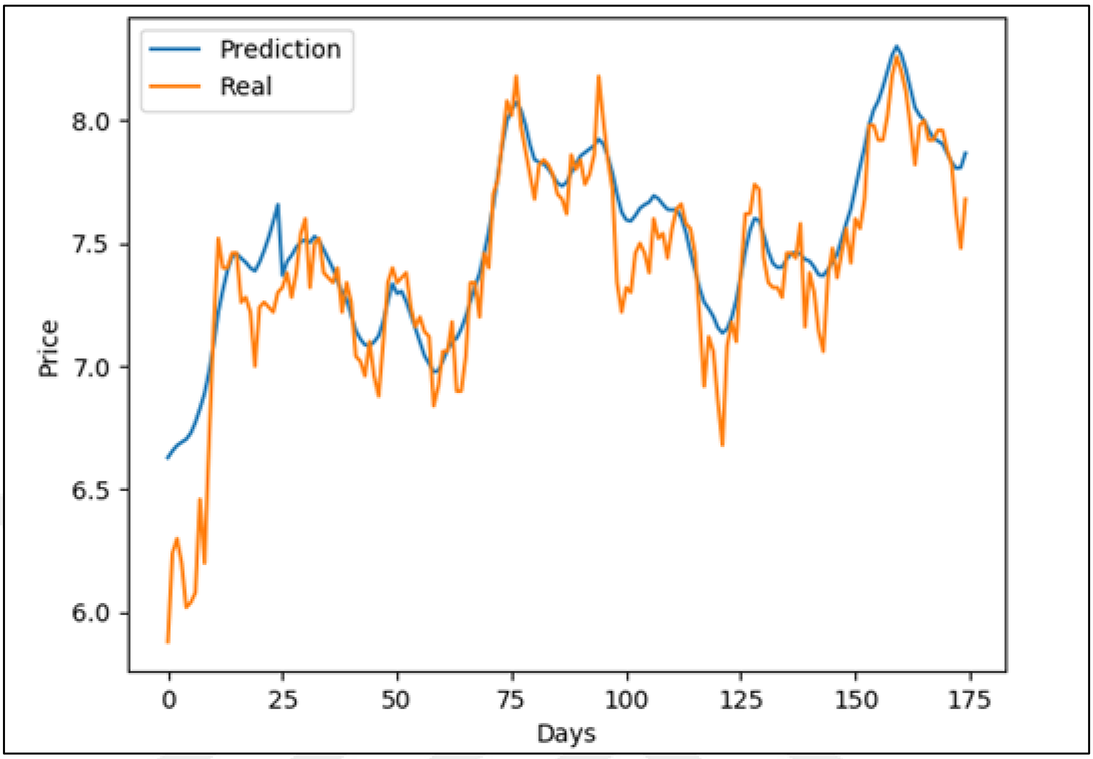
This problem is addressed by Long-Short Term Memory (LSTM) networks. Therefore, LSTMs are often preferred in many classification and regression problems based on sequential data. LSTMs have been used in the majority of recent successful studies with RNNs.

Table 2.1: Behaviour of RMSE under different epocs

Number of epochs	RMSE of learned RNN	Duration
16	0.002474	00:01:22
32	0.001028	00:02:17
64	0.0007549	00:04:44
72	0.0007589	00:05:04
100	0.0005417	00:07:18
124	0.0006015	00:08:54
170	0.0005116	00:12:42
200	0.0005211	00:15:40
234	0.0005775	00:16:41
400	0.0005213	00:28:52

Table 2.1 indicates the behaviour of RMSEs under different epochs of the recurrent neural network. The results show that a small number of epocs do not enable RNN learning enough. RMSE strongly dependent on epocs, after 200, the learning process has became stable and increasing of number of epocs no longer helps to model learning. The closing price movement assumption has been increased 66 percent after 200 epocs. The number of neurons is a very important parameter in the learning process of recurrent neural networks. A larger amount of neurons can have better predictive results. However, it takes a much longer training period. We did a few experiments with a different number of neuron settings to find the best configuration. Figure 2.2 indicates the prediction and actual closing prices of GARAN, the model not trained using optimal configurations and sentiment analysis results still not inserted to the model. It is shown there prediction and actual prices are quite different from each other.

Figure 2.2: Prediction vs Actual GARAN Closing Price after 200 epocs



2.8 BACKPROPAGATION THROUGH TIME

The purpose of recurrent networks is to classify sequential inputs correctly. In order to perform these operations, we use the backprop and the gradient descent of the error. Backprop in feedforward networks is responsible for distributing the derivative of the output error to the weights. Using this derivative, the learning coefficient is regulated by gradient descent and weights are arranged to reduce the error. The method used for RNN is called BPTT. This means that backprop is applied to all of the time-sequential sequence calculations. Artificial networks use a series of functions in the form of nested f (h (g (x)). the derivative can be resolved by the chain rule when a time dependent variable is added here.

2.9 TECHNICAL INDICATORS

Technical analysis has been used by professional investors for more than a century and is still popular. Technical analysis of securities is based on graphs and charts consisting of historical market data rather than basic indicators such as financial statements, sectoral and macroeconomic data. Using these graphs and charts, some special forms in stock prices are determined and the prices are estimated based on these analysis. Technical analysis is used to estimate the suitable time to buy or sell the stocks. Technical analysts use charts with technical data such as price, transaction volume, highest and lowest prices to predict the future movement of the stock. There are also many different indicators useful to make some predictions about the movement of stock. Technical indicators are obtained on the basis of information regarding price and transaction volume. Technical indicators are based on mathematical formulas which allows to create models from price movements. It is intended to be applied to price or transaction volume data of a security. In our research, we used RSI, MA and SMA indicators for feeding our model**.**

2.9.1 Moving Average Convergence Divergence

The Moving Average Convergence Divergence (MACD) indicator was discovered in 1979 by Gerald Appel and has since become one of the most commonly used indicators in technical analysis. It is an indicator based on the moving averages and gives an idea about the direction of the trend. The calculations resulting from the increase or decrease of the difference between the exponential moving averages (12 and 26 days) constitute the MACD line, while the 9-day exponential moving average of the difference (trigger level) is used as the signal line. It’s claimed that it can reveal subtle changes in a price’s direction because it estimates the velocity, the derivative of the price, with respect to time. Since MACD is calculated on long and short term moving averages, which by themselves can be interpreted as certain trend indicators, it carries information related to the index and inserted to the model as a new feature. The MACD chosen for this project is MACD(1, 10, 200) which means it’s based on 1- day, 10-day and 200-day moving averages. For clarification, the 1-day moving average is simply the daily return from one day and the day before, and is used as a signal line. The buy signal is given when the MACD line cuts the signal line from the bottom upwards and rises above the signal line. The sell signal is given when the MACD line cuts the signal line from top to bottom and falls below the signal line. Negative mismatch occurs when price increases are not supported by the indicator. When a MACD value is larger than the signal line at that point in time, a positive trend can be concluded from the graph. The reverse holds true, which means that the momentum of a movement is decreasing if the long term moving average overtakes the short term and a negative trend is concluded if the MACD is below the signal line.

2.9.2 Relative Strength Index

The Relative Strength Index (RSI) indicator, published by J. Welles Wilder in 1978, is an indicator used to determine whether excessive prices of the financial instrument are oversold. The most common period is 14 days [Wilder, 1978]. The RSI value is measured in a range from 0 to 100, with a “high” value located at 70 and a “low” value located at 30, with a high value indicating that the underlying security is overbought and a low value indicating that it is oversold. The RSI in turn is based on the relative strength, RS, which is calculated as:

RS = Avg(14D Gain) / Avg(14D Loss)

where the gains and losses are simply the closing price difference of one day and the day before. The RSI at each time step is then calculated as:

RSI = 100 - (100/(1 + RS))

2.9.3 Moving Average

It is the most widely used indicator among whole indicators. The trend is followed by a certain period of time in the past, taking the average of the current period, shows how far the current price on the trend. When you apply the moving average, if the current price cuts the average upward, it gives buy signal. If it cuts down, it gives sell signal. In these cases it is associated with the bear and bull market. On the other hand, if we use a different two-term average, if the short-term average cuts the long-term average upwards, buying means and buying downwards means selling.

2.10 VANISHING/EXPLODING GRADIENT

Gradient is a value that allows us to adjust all weights. However, in long networks which are connected together, the effect of the error decreases considerably and the gradient may begin to disappear. This makes it impossible to find the right result. Since all layers and time-dependent steps are connected by multiplication, their derivatives are in danger of extinction or flying. Gradient exploding will cause the network to produce very large values, which will distort the correct result. It is a simple and effective way to cut very high value gradients by setting thresholds for this. Excessive shrinkage of gradients is a much more difficult problem. It is not clear where and when it should be stopped. There are several solutions to solve this problem. Selecting the appropriate initial values for W will reduce the extinction effect. Another solution is to use ReLU instead of sigmoid and tanh activation functions. The derivative of the ReLU function is 0 or 1. Therefore, such a problem will not happens. Another method is LSTM which is designed to solve this problem. To avoid this problem, we used RELU activation function and LSTM cells while creating our model.

3. DATASET DESCRIPTION

In this section, we provide details about the financial news and stock price dataset used in our experiments. Financial data can be collected from data providers like euroline, matriks and bloomberg, governments and so on. We used bloomberg as our data provider for quantative dataset. For the sentiment analysis part, we used historical tweets from twitter. We also give the preprocessing operations applied to text data.

3.1 FINANCIAL NEWS DATA

We know that ForeksTurkey twitter user's shares include political and financial news. In this study, we received political and financial news that we need from the tweets of this user. We collected 8 years of tweets about economic, political news and stock itself. Working with 8 years of data allows much more consistency during result validation because regression model learns better using more data points for training also this gives better results on testing. Another plus is to prevent our results from being affected by seasonal movements and trends

As a result, we obtained 55,341 tweets for the years between 2010 and 2018. On average there are 62 news per day. Also we have favorite_count information which indicates the importance of a tweet and we used this information as a coefficient of related tweet so which means that not all news has the same importance. Considering this situation, we calculated weighted average values using positive, negative and neutral prediction results and PositiveAVG, NegativeAVG, and NotrAVG features are created. Also standart deviations are calculated between prediction of each day’s news and this results also added to the model as PositiveSTDEV, NegativeSTDEV, NotrSTDEV features. The dataset is now structured in rows like:

Open, High, Low, Close, Close_Previous, RSI14Day, RSI3Day, MA10Day, MA30Day, MA50Day, RSI9Day, RSI30Day, Volume, SMAVG5, PositiveAVG, NegativeAVG, NotrAVG, PositiveSTDEV, NegativeSTDEV, NotrSTDEV

3.2 STOCK DATA

First, we collected daily GARAN market data from Bloomberg for the years between 2010 and 2018. This dataset has 1892 data points which indicates that the Istanbul stock market is up and running on that days. In this thesis, we generally consider the daily closing price directions of the GARAN stock. Predicted and actual closing price values transformed to the binary variables. The value 1 means closing price is increasing and 0 means decreases. We also need some calculations using GARAN price informations namely technical indicators such as MACD, RSI, MA, SMAVG and this technical indicators also added to the model as new inputs. In addition previous closing price is also added to the model as a new input. Here is the current inputs after we received stock data from Bloomberg and calculating technical indicators using closing prices of GARAN.

Open, High, Low, Close_Previous, RSI14Day, RSI3Day, MA10Day, MA30Day, MA50Day, RSI9Day, RSI30Day, Volume, SMAVG5

3.3 PRE-PROCESSING

Naturally, some of our data will have spaces, web addresses, hashtags, emojis in it, some will only be garbage, and will be completely outside the left area, and some can be made more effective through scaling or normalization. In this step, all improperly formatted messages that may prevent learning has been cleared. Such as incomplete, incorrect or duplicated data. The raw data must be preprocessed before being sent through a machine learning model and this is because real-world data is often incomplete, noisy, and inconsistent and if this is applied to the machine learning model, the results may come unexpectedly. Data preprocessing is a proven method for solving such problems. After pre-processing step, the data is become ready to use in Fasttext for the model creation.

3.3.1 Stopwords

Stop words are the most commonly occuring words which are not relevant in the context of the data and do not contribute any deeper meaning to the phrase. We removed this words from our news dataset to reduce complexity. Some of the turkish stop words are below.

[“acaba, bazıları, birçoğu, bizden, bunlar, kendisi, kimisi, itibariyle, madem, şayet …”]

3.3.2 Casing The Characters

We see that some words contain uppercase letters. Converting character to the same case is necessary to improve the performance of our model due to this reason we converted our whole dataset to lowercase.

3.4 TEXT CLASSIFICATION

Text classification is a main problem to many applications, such as sentiment analysis, spam detection. In this section we described building a text classifier using fasttext tool. The goal of text classification is to assign news to one or multiple categories. Our categories are positive, negative and neutral. We have to set labels to the pre-processed data to build classifier. We built a classifier which automatically classifies economical and political news into one of possible tags, such as positive, negative or neutral.

FastText allows us to change many parameters during the learning process. Since we did not know which of these parameters would give the most accurate results, we wanted to try the learning process with many parameters and use the most successful one. The most successful parameters for Turkish are; minn: 2, maxn: 4, wordNgrams: 2, dim: 25, epoch: 25, bucket: 1000. Using the above parameters, we achieved a success rate of 90 percent. it seems sufficient for us to use this outputs as a new input in our regression model. Using the resulting senitment model, we started to identify 54,841 news. After the step of receiving data through twitter, some simple pre-processing steps are applied and dataset is filtered by keyword GARAN also some economy news added to the filtered dataset which are potentially impact to the daily GARAN prices such as “ertem: sanayi üretimi rakamlarıyla birlikte görüyoruz ki 2017'de büyümede yüzde 7,1-7,2 rakamlarını göreceğiz-trt haber”. we labeled 500 news into one of the possible tags. All labels start with the pre-attachment __label__; this is how fastText understands that it is neither a label nor a word. The model is then trained to guess the labels given the word in the document. Before we train our classifier, we need to divide the data into trains and verification sets. We used the verification kit to assess how good the learned classifier is on new data. Our full dataset contains 55,341 tweets, we labeled 500 tweets and shuffle it 10 times. After every shuffled dataset, we splitted data into a training set of 450 and validation set of 50 news. Table 2.1 shows the commands that we used for splitting dataset.

Table 2.2: Randomly split data into training and test sets

ls | sort –R garan.preprocessed.txt > garan.preprocessed_1.txt
head -n 450 garan.preprocessed_1.txt > garan.train
tail -n 50 garan.preprocessed_1.txt > garan.valid

Then we started training on randomly splitted dataset 10 times and obtained following results. Now we created a trained model.

Table 2.3: Creating a model using fasttext

./fasttext supervised -input garan.train -output model_garan
Average number of words: 1401
Number of labels: 3
Progress: 100.0% words/sec/thread: 12430 lr: 0.000000 loss: 0.388244 eta: 0h0m

This step we applied a validation test on our new model. Table 2.4 shows precision and recall values on validation set

Table 2.4: Testing model with validation set

./fasttext test model_garan.bin garan.valid
Number of samples: 50
Precision: 0.84
Recall: 0.84

FastText uses only five epoch values by default in each model training sample, that is quite insufficient to be a valid training. Also our training dataset only have 450 training samples. Fasttext let us change number of epoch value using –epoch option. We obtained optimal epoch value as 25 after various tests so for the next model we increased this parameter to 25 and build a new model. Following this change increased average precision and recall values to 0.90. It is much better than previous results.

Table 2.5: Sample of tagging twitter news

__label__positive	credit suisse garanti bankası için fiyat hedefini 7 81tl 8 19 tl yükseltti.
__label__negative	citigroup garanti bankası için tavsiyesini al'dan sat'a indirdi.

As a summary, we classified 500 tweets as positive, negative and notr identifiers. This dataset splitted into a training dataset (450 news) and a validation dataset (50 news). Then we calculated training accuracy together so we applied 10 fold cross validation on the classified dataset. Overall the model gives an accuracy of 98.2 percent on the training set and 90.0 percent on the validation set.

3.5 TRAINING NEURAL NETWORK

The input values that we used to train our network is explained in detail in this section. A total of 19 features from different sources (twitter and bloomberg) and indicators (MACD, MA, SMA, RSI…) were also added to the model as new features. We used recurrent neural network to train our model. The technology and libraries that we used are explained in detail in the methods section. We obtained a dropout value as 0.4, while analyzing the optimum parameters. RMSprob is also used to reduce oscillation You can also find the optimal parameters of the regression model we created under results section. These values were obtained from many experiments and the model with the most performance model was recorded and parameter values were taken. Training an RNN is similar to train traditional Neural Network. Our LSTM model is composed of a sequential input layer followed by 2 LSTM layers and dense layer with activation and then finally a dense output layer with RELU activation function.

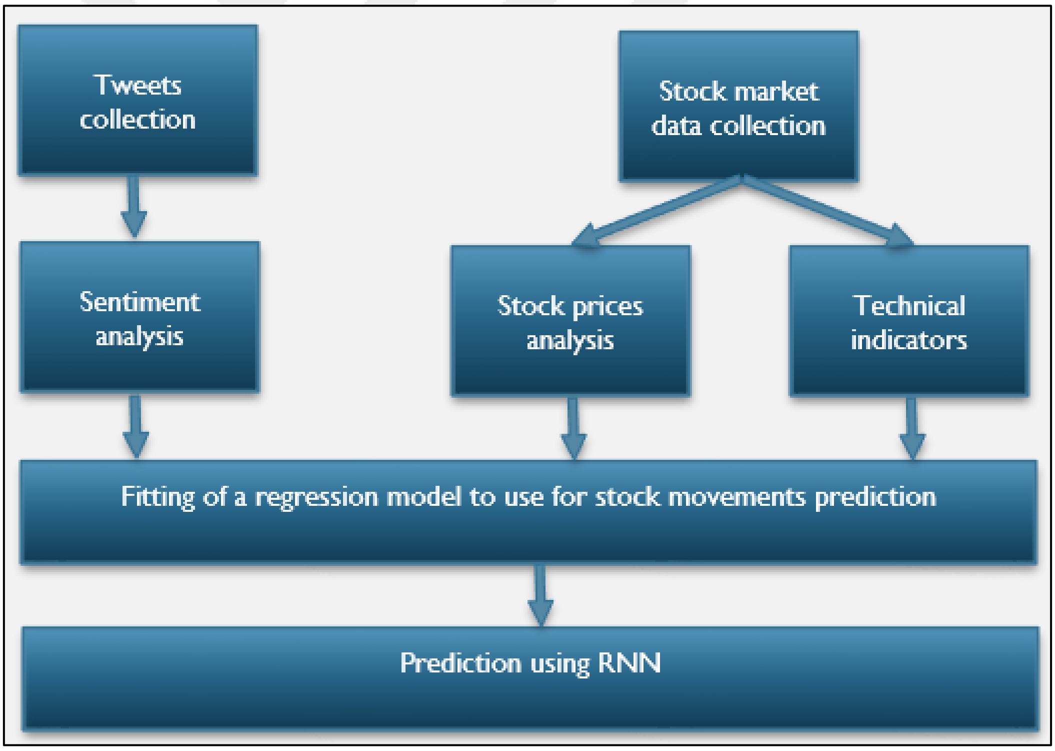
3.6 ANALYZING DATA

There is a known fact that a model learns better if we have more data. Also better learning will also bring high accuracy. A large dataset will prevent overfitting as well as increasing model accuracy. Synthetic data creation is the first method used to increase the data set when the size of the data is small. In fact, we want our data set to reflect the real-world complexity of the problem as much as possible. If we can fit the dataset into this prensible, accuracy value will be better. In this sense, the presence of different versions of the data in the dataset will increase the conformity of the dataset to the principle. For this reason, the size of the dataset is enlarged to increase the performance by including different variations of the data and their appearance in the dataset. The working logic in model improvement is based on the assumption that the model with small weight values is simpler, interpretable than the model with large weights. Therefore, choosing low value weights instead of high value producing weights will increase performance and prevent overfitting. We used the historical stock price data for GARAN in this reasearch. As you can see there are around 1892 items, each representing one day’s stock market attributes for the company this is around 8 years data.

4. METHODS

The application follows the general framework shown in Figure 4.1. The inputs of the model consist of sentiment analysis, stock price analysis and technical indicators. We analyze the time series of daily tweets in parallel to the daily stock market time series. After determining the general idea based on the content of tweets for each day, we can try to fit a model to look for correlation. Since most studies do not address this step and accept simple assumptions, we attach importance to control assumptions. However, before we check assumptions, we still regress because we use them to validate with a particular model to validate conclusive results, but it can still provide intuitive good predictions.

Figure 4.1: Stock price movement prediction architecture



4.1 TENSORFLOW

In this thesis, we used tensorflow framework on the sidelines of building and trainining our model. Tensorflow is an open source library which allow us to use machine learning techniques developed by Google. Running on both CPUs and GPUs is another advantage. Also using GPUs dramatically decrease training time. The reason why we chose Tensorflow is it has some main advantages such as easy to build, fast and train neural networks and capability to be extended. The key concepts of building a neural network in Tensorflow can be summarized in four points:

i. Creating a Tensorflow session, the core of the Tensorflow object model, which handles scopes and variable access through time and between models

ii. Building a computational graph and using its nodes: Placeholders, Variables and Tensors

iii. Training the model by iteratively calling the Run function of the current session on pre-defined Optimization objects

iv. Saving and loading existing models to generalize new data by using the saver object

v. The idea is to try two different types of machine learning models; an autoregressive model and a binary classifier.

4.2 KERAS

Keras is a wrapper that uses Theano or Tensorflow as a backend. It uses Python and helps to identify and train models very easy. Before installing Keras, it is necessary to install one of its backend engines such as Tensorflow backend. While fitting models during training, Keras expects several parameters, I would like to describe some of them. With the validation_split feature, we can examine how successful our model is by providing a certain percentage of the inputs and data in the target. With Epochs feature, we specify how many times our data should pass through our neural network. You can also see the Batch Size feature in many samples. This is especially the size of how many examples you enter to your neural network each time. If you do not specify this when fitting your model, this value will automatically be set to 32. The Verbose feature allows us to show only the results obtained after each epoch during training. If you enter zero, you cannot review them. My observations while performing the experiments indicated that displaying the errors during training by setting this value to one helps to understand how the model evolves throughout the training process.

4.3 FASTTEXT

FastText by Facebook Research is a library that efficiently performs vector representations and sentence classification of words, especially in languages such as Turkish and Finnish where the same word differs structurally in grammar. (Ruder et al. 2017). FastText supports supervised (classifications) and unsupervised (embedding) representations of words and sentences. The main disadvantage of deep neural network models is that they took a large amount of time to train and test. Here, fastText have an advantage as it takes very less amount of time to train and can be trained on our home computers at high speed. In our reseach training and prediction process takes a few seconds.

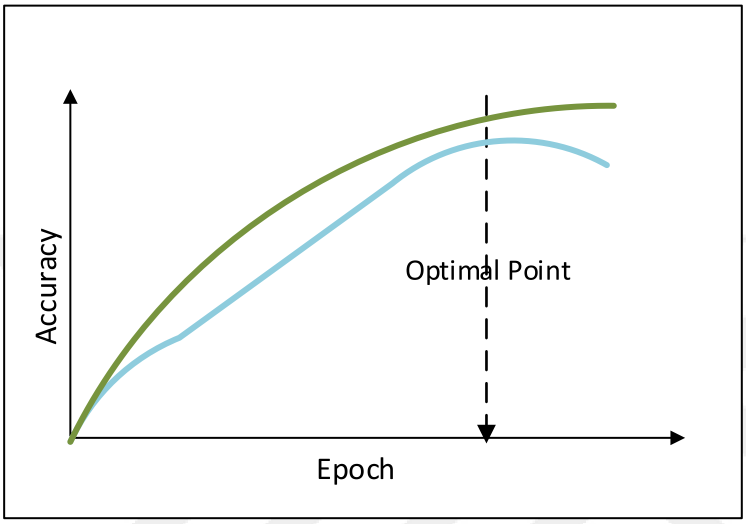
4.4 REGULARIZATION

Overfitting, which occurs when a machine learning model that is more complex than the actual function is fitted to the data, is an important problem affecting the quality of the prediction model. In this case, not only the pattern in the data but also the noise that is not a part of the actual function is learnt by the model. On the other hand, if the fitted model is less complex than the actual function, underfitting occurs.

To avoid these problems and obtain a “good” fit, a proper hyper-parameter optimization procedure and training process should be carried out in building a machine learning model. The common approach used for this purpose is called cross-validation, in which a portion of the dataset is left out for validation. During training, while the training error decreases with increasing complexity, the validation error firstly decreases and then starts to increase due to overfitting.

In Figure 4.2, the typical training/validation error scenario is shown for increasing numbef of epochs for a neural network model. While both training and test accuracies increase up to a point, after dashed line test set accuracy starts to decrease. This is where the distinction takes place, the model has started to overfit to the training data, at that point the training of the model should be stopped. If the training process is stopped somewhere before the dashed line, underfitting occurs since the training is stopped before reaching to the optimal point shown with the dashed line.

Figure 4.2: Overfitting detection



4.4.1 Dropout

It has been observed that rarefy of nodes less than a certain threshold value in fully connected layers increases the success. In other words, it is observed that forgetting weak information that is not a part of the general pattern may increase the generalization ability of the model. One of the regularization methods used to avoid overfitting and helps the model to be optimized faster is the Dropout method.

Figure 4.3: Dropout layer presumption of holding 0.5

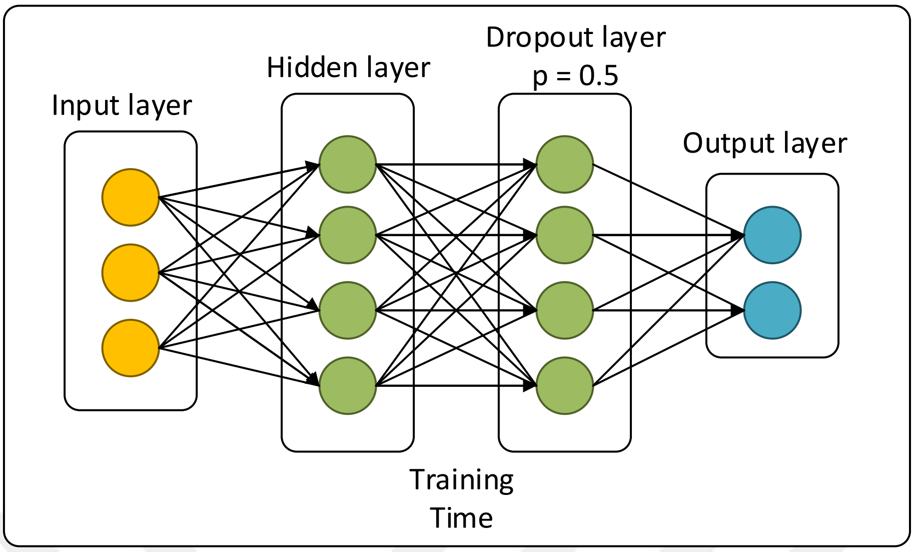
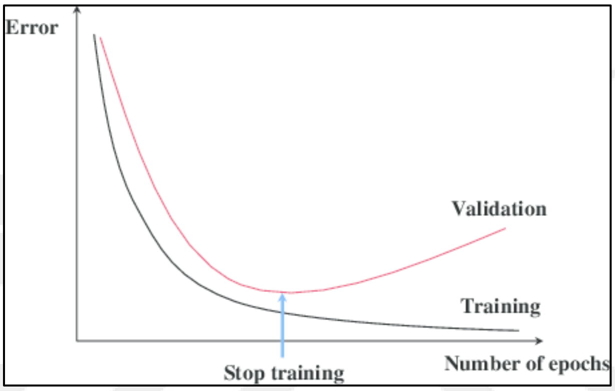


Figure 4.3 shows training time of an input with 0.5 dropout layer. The dropout value is generally used as 0.5 but this value can also be a part of learning process that is optimized during hyper-parameter optimization procedure since its value varies according to the problem and dataset. We obtained a dropout value as 0.4 while analyzing the optimum parameters. If a dropout value is used as a threshold value, it is defined as a value in the range [0,1]. It is not necessary to use the same dropout value on all layers; different dropout values can also be used. Dropout technique is usually used after fully-connected layers. Using dropout, the links on fully-connected layers are broken. Thus, the nodes have less information about each other and as a result the nodes are less affected by each other's weight changes. Therefore, more robust models can be created with dropout method. At the same time, a better learning will take place because different combinations of hidden units work on each layer. In this sense, when dropout is used, it can be thought that hidden layers work like ensemble in random forest. This will provide better performance for the model both in time and accuracy. Dropout method is one of the most common regularization methods used in deep learning methods.

4.4.2 Early Stopping

During the learning phase, if the value difference between validation loss and test loss gets higher, the models starts to overfit or learn noise. In this case, when the validation error starts to increase, the training is stopped and the returned to the previous step. In order to return to the previous step, the data of the previous step must be stored in each learning step (epoch) during training. As can be seen in the graph in Figure 4.4, at the beginning of the learning, training error and validation error both decrease together. However, we see that the test error increases after the step indicated by the dashed line in the example, at this point there has been memorization and learning should be stopped before this step. Changes in weights after overfitting will adversely affect the model.

Figure 4.4: The early stopping principle



4.5 FEATURE SELECTION

Trading decisions are not solely made with regards to daily returns, but also widely supported by at least two so called technical indicators: The moving average convergence-divergence values (MACD) and the relative strength index (RSI). The theory behind these indicators, their calculations and their impact on decision making has been covered in the theory chapter. The experiment is performed on the same dataset; a discrete time series of stock data known as the GARAN.E. The added features have been calculated based on the closing price and added to the dataset. The resulting 19 features are:

{'Open', 'High', 'Low', 'Close', 'RSI14Day', 'RSI3Day', 'MA10Day', 'MA30Day', 'MA50Day', 'RSI9Day', 'RSI30Day', 'Volume', 'SMAVG5', 'PositiveAVG', 'NegativeAVG', 'NotrAVG', 'PositiveSTDEV', 'NegativeSTDEV', 'NotrSTDEV'}

The dataset is now structured in rows as shown in Tables 4.1, 4.2, and 4.3.

Table 4.1: First part of the initial structure of the data features

Date	Open	High	Low	Close	RSI14Day	RSI3Day	MA10Day
01.06.2018	8.95	8.97	8.32	8.52	36.7173	16.4300	9.136
31.05.2018	9.56	9.56	8.91	8.91	41.9589	26.6360	9.180
30.05.2018	9.33	9.58	9.33	9.48	52.0413	67.4764	9.175
29.05.2018	9.51	9.52	9.26	9.33	49.0496	55.5083	9.126
28.05.2018	9.44	9.61	9.39	9.53	53.1549	82.4907	9.089

Table 4.2: Second part of the initial structure of the data features

MA30Day	MA50Day	RSI9Day	RSI30Day	Volume	SMAVG5
9.2117	9.7250	33.9210	39.1325	257323552	134100576
9.2673	9.7778	41.3005	41.8511	124860840	101661536
9.2883	9.8196	57.5720	46.4059	129514152	106803528
9.3040	9.8472	53.2650	44.8798	85694504	107738952
9.3197	9.8836	60.5500	46.5897	73109816	113103568

Table 4.3: Third part of the initial structure of the data features

PositiveAVG	NegativeAVG	NotrAVG	PositiveSTDEV	NegativeSTDEV	NotrSTDEV
0.9645	0.6823	0.4824	0.0467	0.2772	0.000
0.9292	0.4395	0.5063	0.1086	0.0000	0.1086
0.9451	0.5088	0.0000	0.0991	0.0176	0.0000
0.9243	0.6392	0.4448	0.1198	0.0758	0.0172
0.9234	0.6397	0.0000	0.1030	0.1409	0.0000

The first 50 rows and the last row are removed and not used for training and validation, respectively, considering that 50-days closing price technical indicator is used as a feature. This indicator depends on the 50 previous values, so training starts at the 51th row. Since the goal in this thesis is to investigate long-term closing price movements, there is no interest in using more frequent data than daily data. Table 4.4 shows inputs and output data types used in our experiments. The description of the last 6 features related to sentiment analysis has been given in Section 3.1 of this thesis.

Table 4.4: Inputs and output data types

Feature	Data Type	Dataset
Output	float64	Stock Price
Open	float64	Stock Price
High	float64	Stock Price
Low	float64	Stock Price
Close	float64	Stock Price
Volume	int64	Stock Price
RSI14Day	float64	Technical Indicator
RSI3Day	float64	Technical Indicator
MA10Day	float64	Technical Indicator
MA30Day	float64	Technical Indicator
MA50Day	float64	Technical Indicator
RSI9Day	float64	Technical Indicator
RSI30Day	float64	Technical Indicator
SMAVG5	int64	Technical Indicator
PositiveAVG	float64	Sentiment Analysis
NegativeAVG	float64	Sentiment Analysis
NotrAVG	float64	Sentiment Analysis
PositiveSTDEV	float64	Sentiment Analysis
NegativeSTDEV	float64	Sentiment Analysis
NotrSTDEV	float64	Sentiment Analysis

4.6 SLIDING WINDOWS

The first step consists of partitioning the data into equally-sized windows, which are small sets of individual data points that are adjacent in the time series. Consider the following example of a slice of the closing price of GARAN shown in Table 4.5, which only contains the date and closing price features.

Table 4.5: GARAN.E Closing Prices

Date	Close
01.06.2018	8.52
31.05.2018	8.91
30.05.2018	9.48
29.05.2018	9.33
28.05.2018	9.53

This data segment can equivalently be re-written as a sequential learning problem by reshaping it as in Table 4.6.

Table 4.6: Shifted Closing Prices

X	y
-	9.33
9.33	9.48
9.48	8.91
8.91	8.52
8.52	-

The new formulation has some interesting properties:

i. Each row represents a time step, and data at each previous time step is used as the input X, and the following time step is the output y.

ii. The first row is useless, because it contains no information about X.

iii. The inherent order of the data points, although shifted, is preserved.

This method of partitioning is called the sliding windows method, or in statistician's terms it's known as the lagging method. In a formal way, this partitioning can be expressed as using the windows W_0 up to , each window containing closing prices p_i with size W. All features in our dataset will go through this partitioning process. Regression model results are converted to a binary variable showing the direction of the stock price change. It is a simple and effective way to increase accuracy by adding threshold values to the next closing price. This prevents to generate faulty signals for small changes. In our research we determined threshold values as 0.02. This value is equal to two price step of GARAN. The output of the binary classification problem is obtained with the following rule:

```
if last_close < next close + threshold:
        y_i = [1, 0]
else:
        y_i = [0, 1]
```

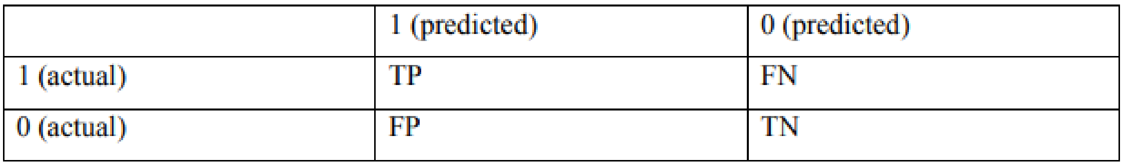
5. RESULTS

In this section, the evaluation metrics, the number of samples used for training and testing, the values of the hyper-parameters of the RNN model and the obtained results are given.

5.1 EXPERIMENTAL SETUP

We have 1892 rows of dataset covering the Istanbul stock exchange data between 06.10.2010 and 01.06.2018. Each row represents one day’s stock market attributes. We used 1659 days of data to train our model and take the latest 233 days of data as a test dataset which is used for testing predictions and assessing the results of the experiments. In the literature, not only the size of the dataset is sufficient for a good model, but also its diversity is also important. As diversity increases, the performance of the model will also increase. One of the biggest misconceptions in the studies using the classification algorithms is to look at the accuracy rate as a criterion of success. Especially in imbalanced data sets, using only the overall accuracy rate may result in incorrect conclusions. The imbalanced dataset is used to define data sets where the distribution between classes is very different from each other. There are various evaluation metrics used to assess the success of machine learning models on imbalanced datasets. Table 5.1 shows confusion matrix which is a table with 4 different combinations including predicted and actual values.

Table 5.1: Confusion matrix.



	1 (predicted)	0 (predicted)
1 (actual)	TP	FN
0 (actual)	FP	TN

The first value calculated is the incorrect classification rate obtained by the following equation:

$$Misclassification\ Rate = \frac{FP + FN}{P + N} \tag{5.1}$$

where sum of P + N shows the total number of samples in the dataset. Accuracy of computation based on the misclassification rate is straightforward:

$$Accuracy = 1 - Misclassification\ Rate \tag{5.2}$$

Comparing the two models with low sensitivity or high recall (or vice versa) is difficult. So, a metric that combines these two values called F1-score is used. The F1-score helps to measure sensitivity and recall at the same time and it punishes extreme values more and uses the harmonic mean instead of an arithmetic mean. It is one of the most commonly used evaluation metrics for binary classification problems on imbalanced datasets.

$$Precision = \frac{TP}{TP + FP}$$

$$Recall = \frac{TP}{TP + FN}$$

$$F1 - measure = \frac{2 * Precision * Recall}{Precision + Recall} \tag{5.3}$$

5.2 REGRESSION MODEL

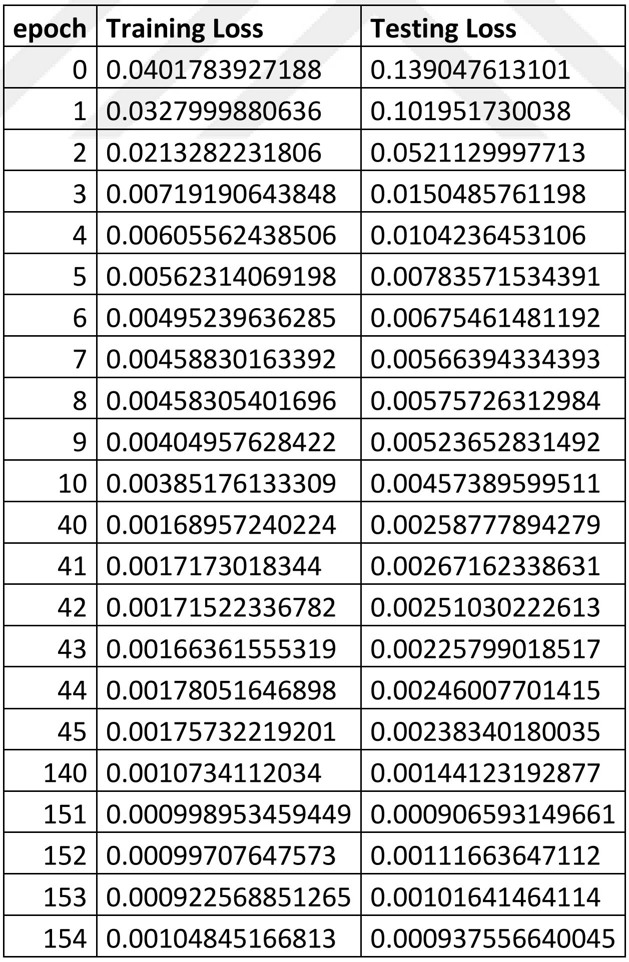
We can see the regression model parameters belonging to the model that has given the best test results in Table 5.2.

The test characteristics of the best model is shown in Table 5.3. Changing number of epochs indicates that, 192 epochs later, the learning process is getting stable and further increasing epoch value does not benefit any more. The best testing loss value is obtained at around 192 epochs.

Table 5.2: Values of the parameters of the model yielding the best testing results

Window size	**Learning rate**	**#Hidden layers**
1	0.001	1
Time Steps	**Batch size**	**Activation**
7	25	ReLU
#LSTM cells	**Training epochs**	**Dropout probability**
60	200	0.4
Optimizer	**Loss function**	**#Features**
RMSprop	RMSE	19

Table 5.3: Regression test characteristics



epoch	Training Loss	Testing Loss
0	0.0401783927188	0.139047613101
1	0.0327999880636	0.101951730038
2	0.0213282231806	0.0521129997713
3	0.00719190643848	0.0150485761198
4	0.00605562438506	0.0104236453106
5	0.00562314069198	0.00783571534391
6	0.00495239636285	0.00675461481192
7	0.00458830163392	0.00566394334393
8	0.00458305401696	0.00575726312984
9	0.00404957628422	0.00523652831492
10	0.00385176133309	0.00457389599511
40	0.00168957240224	0.00258777894279
41	0.0017173018344	0.00267162338631
42	0.00171522336782	0.00251030222613
43	0.00166361555319	0.00225799018517
44	0.00178051646898	0.00246007701415
45	0.00175732219201	0.00238340180035
140	0.0010734112034	0.00144123192877
151	0.000998953459449	0.000906593149661
152	0.00099707647573	0.00111663647112
153	0.000922568851265	0.00101641464114
154	0.00104845166813	0.000937556640045

155	0.0010266315633	0.00107496178576
156	0.00100294734185	0.000948094808596
157	0.00100254377876	0.00101483302257
158	0.000971370241486	0.000911975986258
159	0.00100465285418	0.000914903357625
190	0.000944616066112	0.000947026972426
191	0.00091603798698	0.000922161343624
192	0.000967960669294	0.00081242898055
193	0.00093014194123	0.000854386850343
194	0.000861888701566	0.00125051290628
195	0.000861883517878	0.000861927792097
196	0.000894154106529	0.00107638282186
197	0.000823225052554	0.00137817396899
198	0.000858533953837	0.00112225216747
199	0.000890343086705	0.000989580543579
200	0.000869520301329	0.00161047840291

Figure 5.1: Train and test errors at each epoch

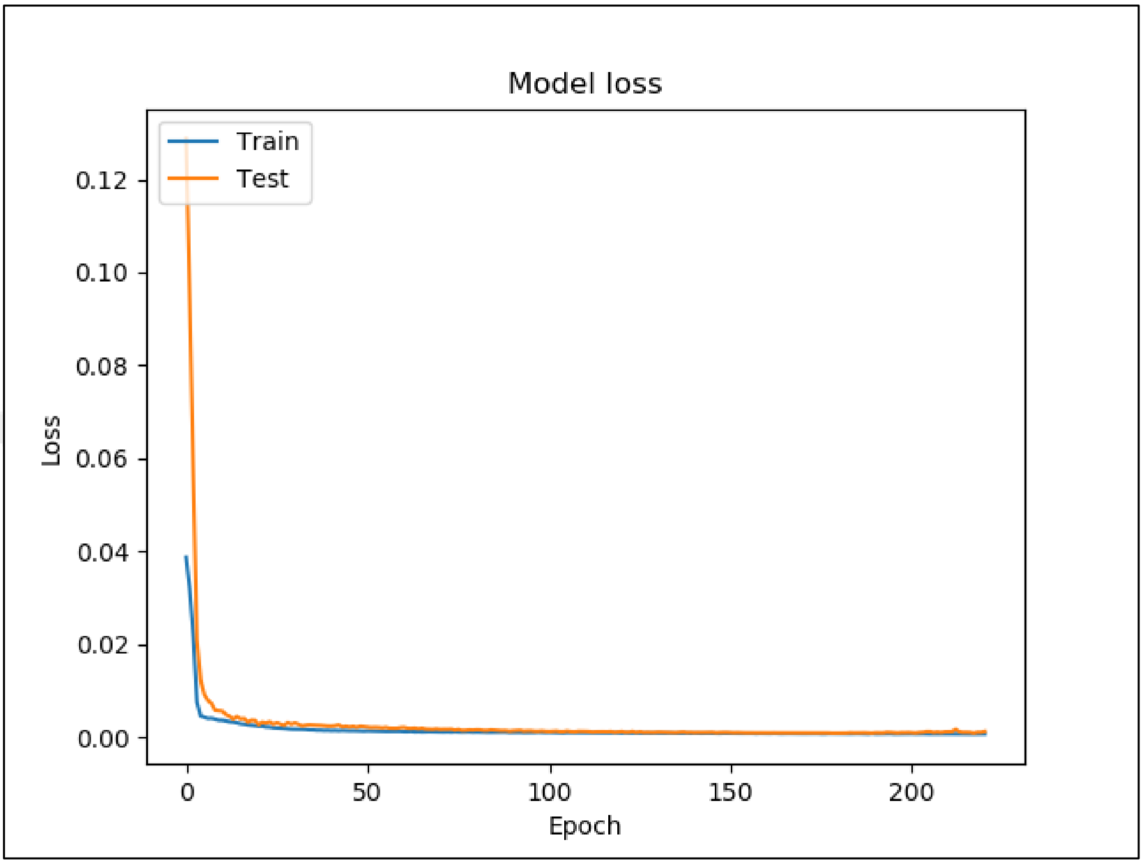


Figure 5.1 shows the performance of the model created on validation and training datasets. There are small fluctations in the error value in some parts of the error curve. The parameters selected for the data in some iterations may be appropriate, while others may not. However, these fluctuations are expected to decrease with the increasing number of iterations and the model will better fit to the data. But using more than enough iterations waste operational resources and also may cause overfitting. Here we need to find a point where the validation error does not decrease or change significantly anymore. A good fitted model will correspond to a number of iterations where the validation loss value does not reduce significantly anymore.

Figure 5.2: Number of daily news

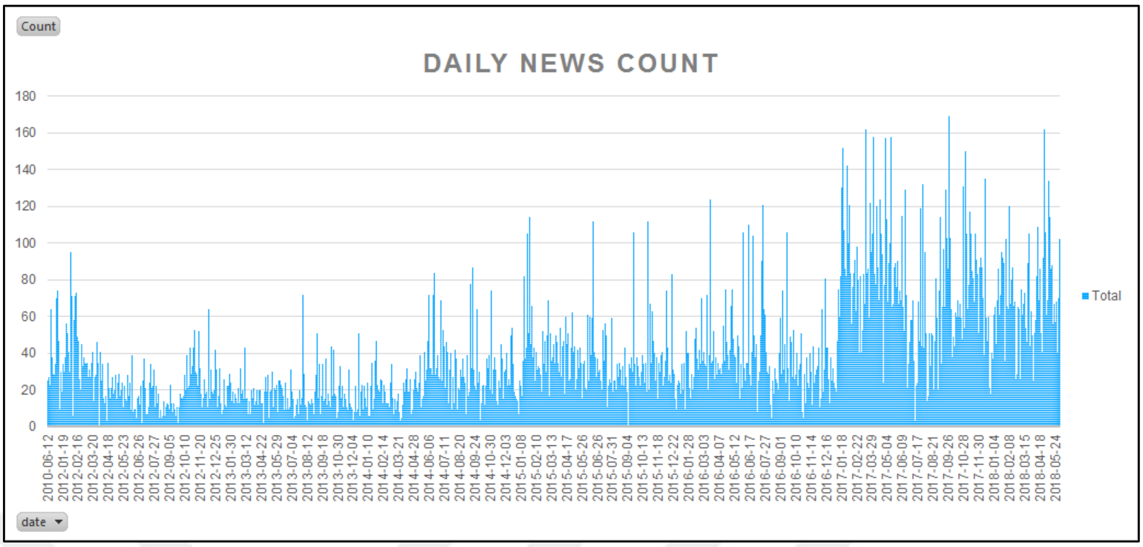
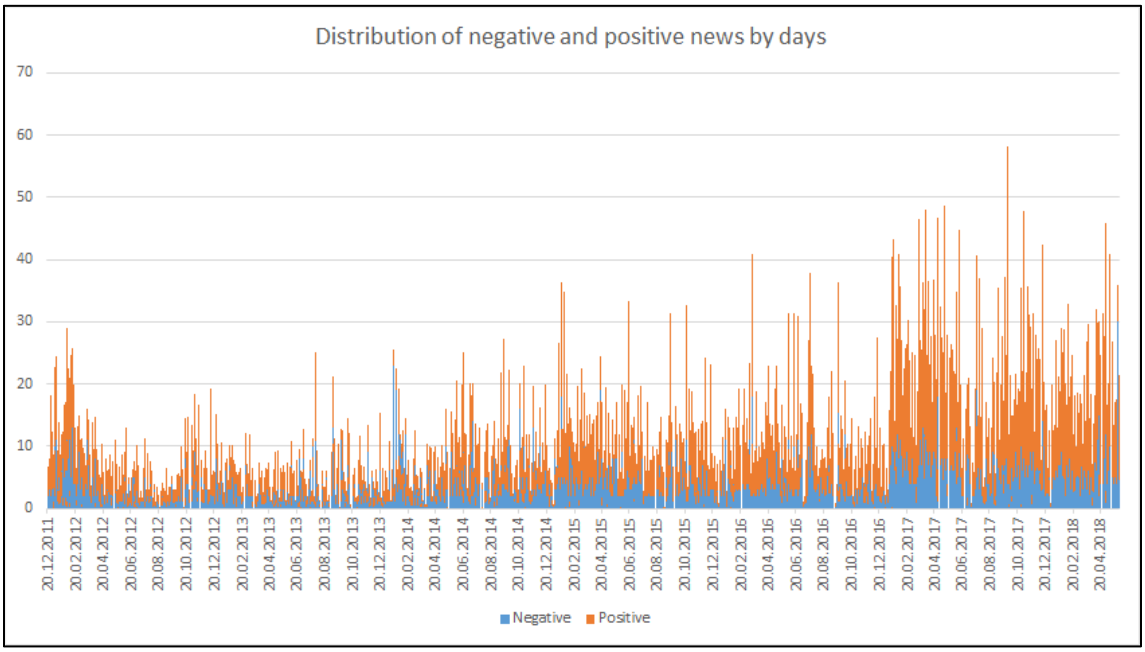


Figure 5.2 shows the distribution of economic and political news informations which is received from Twitter on a daily basis. It is seen that the number of comparably recent news is more than those of the previous years.

Figure 5.3: Distribution of negative and positive news by days



Another important point that can be noted is the distribution of the classified news obtained after the sentiment analysis. Figure 5.3 indicates distribution of negative and positive news. As a result of sentiment analysis, 7118 news were marked as negative, 9548 news positive and 38675 news labeled as neutral.

Figure 5.4: Comparision of prediction and actual GARAN prices using technical analysis and stock prices

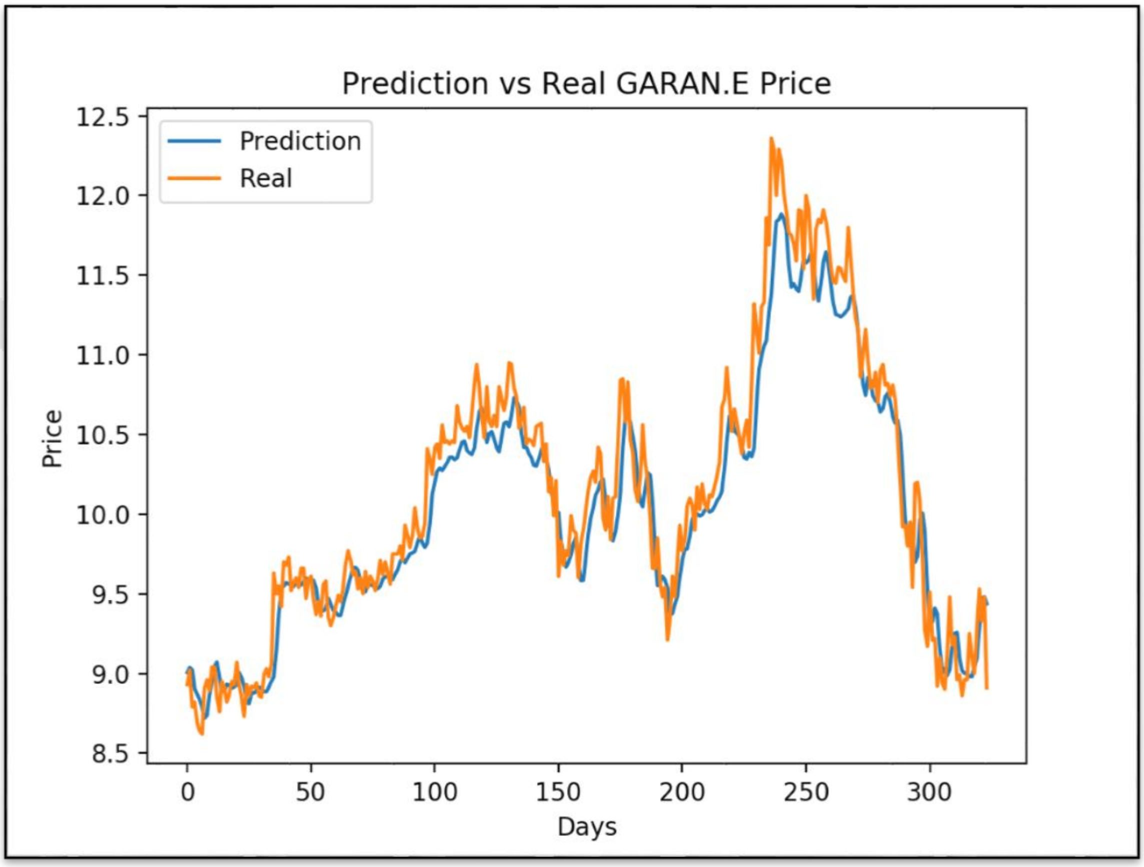


Figure 5.4. shows the predicted and actual prices for the test data. We can see that predicted price and actual stock price do not fit in some points so this will effect movement directions in a negative way. In table 5.4 model-1 (stock price + technical indicators) indicates the performans and accuracy of the model.

Figure 5.5: Comparision of prediction and actual GARAN prices using technical analysis, stock prices and sentiment analysis

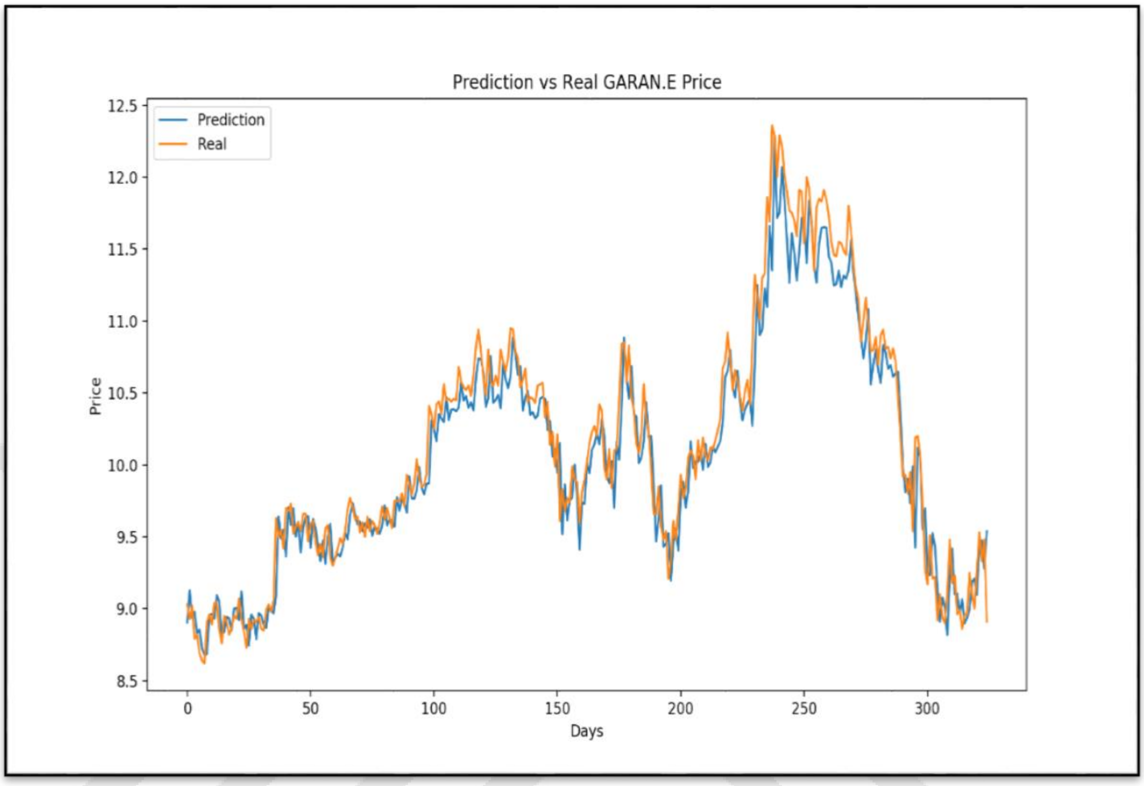


Figure 5.5. shows the predicted and actual prices for the test data. Predicted price is in orange color and actual stock prices are in blue color. We can see that predicted price and actual stock fits better in most points better than model-1 and movement directions seems better after stock prediction using LSTM with sentiment analysis features.

Table 5.4: Results obtained using technical indicators and sentiment analysis as input for LSTM-RNN model

Model		Precision	Recall	F1-score	Count
Model 1 Stock Price + Technical Indicators	1 (Up)	0.59	0.57	0.57	120
	0 (Down)	0.52	0.55	0.53	113
	Average/Total	0.56	0.56	0.55	233
Model 2 Stock Price + Technical Indicators + Sentiment Analysis	1 (Up)	0.69	0.66	0.67	120
	0 (Down)	0.62	0.65	0.63	113
	Average/Total	0.66	0.65	0.65	233

Table 5.4 shows the results obtained using technical indicators and sentiment analysis as input for LSTM-RNN model. We should note that the number of class 1 (up) and class 0 (down) is 120 and 113, respectively, on the test set. As seen in Table 5.4, while F1-score of Model 1, the model built using only stock price and technical indicators, was 0.55, it has increased to 0.65 in Model 2 in which sentiment analysis label information is integrated to the input. Model 1 gives 71 true positive movements and 59 true negative movements on the test data whereas Model 2 achived 83 true positive movements and 70 true negative movements. The results indicate that the financial news includes important information about the direction of the stock and integrating this information with sentiment analysis into the model with data level fusion increases the success of the model on both of the classes.

6. DISCUSSION AND CONCLUSION

The number of investors wishing to utilize their savings in the stock market is rapidly growing day by day. This increases the interest in the Stock Exchange and the popularity of the Stock Exchange trading and encourages researchers to investigate new techniques to estimate stock prices. New methods not only help researchers, but also help investors and any person interested in the stock market. Many different approaches are proposed and applied to increase the success of the stock prediction task.

The main goal of this thesis was to assess the effect of financial news on the success of the stock price-based estimation models. For this purpose, we used a machine learning method, namely Long Short-Term Memory Recurrent Neural Network (LSTM-RNN), to estimate the next day's closing stock price. We collected a news dataset and assigned a sentiment label to each of the news using sentiment analysis tools. Then, the output of the sentiment analysis was given to the regression model together with the stock price-based features. The results of this regression model showed that, even with sentiment analysis inputs, estimating the exact future price of a stock is a difficult task due to the inherent chaotic nature of market or lack of some other important hidden factors that are not included in our dataset. there are not enough features. It is known that there are many factors that influence the movements of the stock prices. Therefore, we modelled the estimation problem as a binary classification problem and aimed to predict the movement direction of the stock price, i.e. whether the closing price of the stock next day will be higher than today's price. The results showed that classifying economical and political news as positive, negative, neutral and integrating these values to the model as new inputs increases the percentage of prediction success.

As a summary, accuracy value of the model obtained by using stock prices with technical indicators was measured as 56 percent. In addition, when we train our model with the sentiment analysis outputs of the news, then accuracy value of the model increased to 66 percent. The results also showed that more consistent predictions are obtained for price movements after sentiment analysis score is integrated to the system.

As a future direction, in addition to news, the reviews and comments of the users in social media platforms can be analysed and integrated to the model. In order to build a model that can make short-term predictions, the characteristics of an order book may be combined with the inputs used in this study. The analysis of the ITCH market data feed and the information such as the quantity, size, count, side and frequency of orders in every order book level on a specific stock may give us more accurate results for short-term estimation of stock price movement.

REFERENCES

Books

Alpaydın, E., 2004. Introduction to Machine Learning, Cambridge MIT Press.

Mitchell, T. M., 1997 Machine Learning, McGraw-Hill and MIT Press.

Witten, I. and Frank, E., 2005. Data mining: practical machine learning tool techniques.

Ke-Lin, Du. and Swamy, S.. 2013. Neural Networks and Statistical Learning.

Periodicals

Application of Information and Communication Technologies, Tiflis, Gürcistan ,ss. 1-4, 2012.

Bildirici, M, ve Ersin, O.O. (2009). Improving forecasts of GARCH family models bwith the artificial neural networks: An Application to the daily returns in Istanbul Stock Exchange. Expert Systems with Applications 36, 7355-7362.

Black, A.J., ve McMillan, D.G. (2004). Long run trends and volatility spillovers in daily exchange rates. Applied Financial Economics, 14(12), 895-907.

C. Özsert and A. Özgür. Word polarity detection using a multilingual approach. Computational Linguistics and Intelligent Text Processing, 7817:75–82, 2013.

Cort J Willmott and Kenji Matsuura. Advantages of the mean absolute error (mae) over the root mean square error (rmse) in assessing average model performance. Climate research, **30** (1):63–72, 2005.

Davidov, D., Tsur, O., Rappoport, A., “Enhanced Sentiment Learning Using Twitter Hashtags and Smileys”, In Proceedings of COLING'10, 23rd International Conference on Computational Linguistics, Pekin, China, 241-249, 2010.

Day, M.Y. and Lee, C.C., 2016, August. Deep learning for financial sentiment analysis on finance news providers. In 2016 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining (ASONAM) (pp. 1127-1134). IEEE.

Graves, A., Schmidhuber, J. Framewise, 2005. Phoneme classification with bidirectional LSTM and other neural network architectures. Neural Networks, **18**(5), pp. 592-605.

Hunter, J. D. Matplotlib, 2007. A 2D graphics environment. Computing In Science & Engineering, **9**(3), pp. 45-53.

Iebeling Kaastra and Milton Boyd. Designing a neural network for forecasting financial and economic time series. Neurocomputing, **10**(3):183–202, 1996. ISSN 0925-2312.

Jasic, T., ve Wood, D. (2004). The profitability of daily stock market indices trades based on neural network predictions: Case study for the S&P 500, the DAX, the TOPIX and the FTSE in the period 1965-1999. Applied Financial Economics, 14, 285-297.

Kang, D., Park, Y., 2014, “Review-based Measurement of Customer Satisfaction in Mobile Service: Sentiment Analysis and VIKOR Approach”, Expert Systems with Applications, Cilt 41, ss.1041-1050.

Kara, Y., Boyacioglu, M.A., ve Baykan, O.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, 5311-5319.

Mikolov, T., Grave, E., Bojanowski, P. Puhrsch, C., Joulin, A., 2017. Advances in Pre-Training Distributed Word Representations. arXiv:1712.09405v1

Nils Reimers and Iryna Gurevych. Optimal hyperparameters for deep lstm-networks for sequence labeling tasks. arXiv preprint arXiv:1707.06799, 2017.

Pedregosa, F., Varoquaux, G., 2011. Scikit-learn: Machine learning in Python. Journal of Machine Learning Research, **12**(1), pp. 1712-1740.

Qiu, M. and Song, Y., 2016. Predicting the direction of stock market index movement using an optimized artificial neural network model. PloS one, **11**(5):e0155133.

Ruder, S., Vulic, I., Søgaard, A., 2017. A survey of cross-lingual word embedding models. arXiv preprint arXiv:1706.04902.

S. Demirci, Emotion analysis on turkish tweets. Diss. M. Sc. Thesis, Middle East Technical University, Ankara, 2014.

Smailović, J., Grčar, M., Lavrač, N. and Žnidaršič, M., 2014. Stream-based active learning for sentiment analysis in the financial domain. Information sciences, vol. 285, pp.181-203.

Sohangir, S., Wang, D., Pomeranets, A. and Khoshgoftaar, T. M., 2018. Big data: Deep learning for financial sentiment analysis. Journal of Big Data, vol. 5, no. 1, p. 3.

Şimşek, M.U., Ozdemir, S., “Analysis of the Relation between Turkish Twitter Messages and Stock Market Index”, In Procedings of AICT ‘12, 6th Conference on

U. Eroğul. Sentiment analysis in turkish. Master’s thesis, Middle East Technical University, 2009

Van der Walt, S., Colbert, S. C., 2011. The NumPy Array: A Structure for Efficient Numerical Computation. Computing in Science & Engineering, **13**(2), pp. 12-18.

Other Publications

Github: Long Short Term Memory, 2015. URL http://colah.github.io/posts/2015-08-Understanding-LSTMs/[Accessed:2019-04-13].

Keras. The python deep learning library, 2018a. URL https://keras.io[Accessed: 2019-05-19].

Medium. Multivariate Time Series 2019 https://medium.com/datadriveninvestor/multivariate-time-series-using-rnn-with-keras-7f78f4488679[Accessed:2019-06-22].

MTurk, Amazon Mechanical Turk, https://www.mturk.com/[Accessed:2018-10-12].

Pythonmachinelearning. Recurrent Neural Networks, 2018. URL https://pythonmachinelearning.pro/advanced-recurrent-neural-networks/[Accessed:2019-04-12].

Tensorflow. An open-source machine learning framework for everyone, 2018. URL https: //www.tensorflow.org[Accessed:2019-05-19].

FourYears. The Mystery of Early Stopping, 2017. URL http://fouryears.eu/2017/12/06/the-mystery-of-early-stopping/[Accessed:2019-06-23].

APPENDICES

Appendix A.1 Technical Specifications

The experiments were written in Python 3.6.4, using Jet Brain’s PyCharm development tool with the following configuration:

```
Keras=2.1.5
numpy=1.12.1
pandas=0.22.0
matplotlib=2.2.2
scikit-learn=0.19.1
scipy=1.0.1
tensorflow=1.1.0
tqdm=4.31.1
xlrd=1.1.0
```

The hardware used for implementation is:

Processor = 2,2 GHZ Intel Core i7
RAM = 16GB 1600 MHz DDR3
Graphics = Intel Iris Pro 1536 MB
Operating System = macOS Mojave