Aalto University
School of Science
Master's Programme in Computer, Communication and Information Sciences

Ville Vainio

Engineering analytics of big data pipelines for stock market data

Master's Thesis
Espoo, May 1, 2019

DRAFT! — June 7, 2019 — DRAFT!

Supervisor:	Professor Linh Truong
Advisor:	Professor Linh Truong

Aalto University
School of Science
Master's Programme in Computer, Communication and Information Sciences

ABSTRACT OF MASTER'S THESIS

Author:	Ville Vainio		
Title:			
Engineering analytics of big data pipelines for stock market data			
Date:	May 1, 2019	**Pages:**	31
Major:	Computer Science	**Code:**	SCI3042
Supervisor:	Professor Linh Truong		
Advisor:	Professor Linh Truong		

The abstract provides goal, motivation, background, and conclusions of the work. It has to fit to one page together with the bibliographical information. !Fixme **Add abstract.** Fixme!

Keywords:	stock market, big data, cloud computing, stream processing
Language:	English

Aalto-yliopisto
Perustieteiden korkeakoulu
Tieto-, tietoliikenne- ja informaatiotekniikan maisteriohjelma

DIPLOMITYÖN TIIVISTELMÄ

Tekijä:	Ville Vainio		
Työn nimi: Osakemarkkina dataa käsittelevien big data järjestelmien tekninen analyysi			
Päiväys:	1. toukokuuta 2019	**Sivumäärä:**	31
Pääaine:	Tietotekniikka	**Koodi:**	SCI3042
Valvoja:	Professori Linh Truong		
Ohjaaja:	Professori Linh Truong		
!FIXME **Lisää tiivistelmä** FIXME!			
Asiasanat:	osakkeet, big data, pilvilaskenta		
Kieli:	Englanti		

Espoo, May 1, 2019

Ville Vainio

Abbreviations and Acronyms

EMH	Efficient Market Hypothesis
RSI	Relative Strength Index
MACD	Moving Average Convergence Divergence
TF-IDF	Term Frequency–Inverse Document Frequency
HDFS	Hadoop Distributed File System
QoS	Quality of Service
SLA	Service Level Agreement

Contents

Chapter 1

Introduction

The modern economy revolves around stock market. Stock market is a way for companies to obtain capital which they can invest into their own business. In exchange, the person who invests into the companies stocks technically owns a piece of the company which can return profit to the investor two different ways. The stock can grow in value, which allows the investor to sell the stock in higher price or the company itself can pay dividends to investors based on the number of stocks the investor owns from the company.

The price of the stock is simply determined by the law of supply and demand. If somebody is willing to pay a higher price for the stock then the price of the stock can grow. Because of this the stock market is in continuous fluctuation where people are selling and buying the stocks with the price they think the stock is worth using stockbrokers as the middleman. [13] All of this has lead to the question, how can we invest most optimally into stocks? This is where the following computational methods come in.

There are many strategies on how to invest into these stocks which depend on multiple factors such as; how much do you expect to profit with your investment, how much are you willing to take risk, do you want to make money by selling the stocks or by receiving dividends and so on. The underlying principle with every strategy is to minimize the risk you need to take in order to gain as much as profit as possible. Some of the strategies are based on subjective evaluation of the companies, but more technical strategies use metrics that are calculated from the financial statistics or the real-time market values. Strategies using the former data are called fundamental analysis and the strategies using latter data technical analysis. Neither of these approaches can predict the future of the market, but can statistically decrease the probability of larger losses in the market for the investor altough the probability of large losses is still not zero with these methods. [10]

Fundamental analysis is based on the idea that each stock has a intrinsic value that can be larger than the actual price of the stock in the market and buying these will eventually lead to profits.[2] The fundamental analysis focuses on the financial metrics that consist of companys overall statistics. These are for example how much the company has made profit, how much the company has paid dividends and what is companys cash flow. These tell a lot about the growth of the company and how the future of the company looks like. These metrics are usually published quarterly four times a year and present more long-term statistics about the company. Because of this, the amount of data these values present is quite small in terms of space.

The technical analysis that focuses on the real-time market values, on the other hand, needs new data almost daily. Stock exchanges are usually open from morning, opening around 8 to 10am, until evening, closing around 5 to 7pm on weekdays. Before and after this there are more limited pre- and after-hours trading which lasts usually around 1 to 2 hours depending on the exchange in which more limited stock trades can be made. During these hours multiple values are recorded on the prices of the stock from which the most important ones being: the highest price the stock was sold, the highest price the stock was sold and the number of stocks traded during the time interval. The technical analysis focuses on finding recognizable patterns through this data. [22] Where the data used by the fundamental analysis was relatively small, these values can generate gigabytes of raw data in a week.

Developing a system that can do both of these analyses automatically would mean that the system should be planned from the start to be able to handle large amounts of these data as time progresses. As such task is not trivial, the goal of this thesis is to provide developers, who want to analyse this data efficiently, basic knowledge on what are the best current solutions on handling this data. With this knowledge these developers can save considerable amount of time without the need of trial and error when developing this kind of system from the ground up.

1.1 Application scenario

To have a better picture of how a normal system that handles stock data operates, we are next going to introduce an application scenario which is based on real-life system. From this real-life example we are going to apply the used data sources, the requirements of the possible extensions and the requirements of the results to create a more realistic scenario. This scenario will be then used throughout the thesis to give reader more practical and meaningful results.

This application scenario is presented in figure 1. In the figure, the components marked with solid lines represent the core system functions, and the dashed lines represent add-on functionalities that should be possible to extend into the system in the future. The requirements of the system should be as follows; The system should produce long (r_1) and short (r_2) term predictions of stock prices. The computation of long term predictions can take time because of the nature of these predictions but the short term predictions should be between two minutes to one hour available. Long term predictions are the result of fundamental analysis (f_2) and historical technical analysis (f_4) whereas the short term prediction should come mostly from quick technical analysis (f_3). The cost of the system should grow logarithmically/linear with time meaning that the cost of processing and storing data should not exponentially increase over time. Finally, the core system should be able to fulfill these requirements for at least the +5000 companies in the major U.S stock markets.

The data to the application is ingested from two main types of data sources quote and fundamental. These sources consist of values that were briefly described previously in technical and fundamental analysis respectively. The quote data is usually updated with minute intervals depending on the provider whereas the fundamental data does not change so often. Theoretically, the fundamental data can change anytime, because of dividends which can be payed whenever the companies want but this does not happen often and these values are mostly used in the long term fundamental analysis so longer update intervals are acceptable. In the figure, these are separated into the U.S market ones (d_1 and d_2) and the other sources (d_3 and d_4) that provide the same data on other global markets. The extendable global data sources are grouped into one box but in reality this data would be ingested from numberable different providers as there is no single entity at the time of writing this that provides all of this data. Theoretically the maximum size of this extendable data would be 5Gb per day which is extrapolated in from the U.S market data based on statistics that in January 2019 there were globally 51 599 companies listed in the stock markets [32]. This amount can and will fluctuate as companies enter and exit the markets but it gives us the scale of data we are working with.

Today, stock market analysis has also a large focus on predicting stock prices using secondary data sources that can have reflect and affect the prices of stocks. These secondary data sources can be anything but at the moment one of the most researched sources are traditional media and social media data. Examples of using this kind of data to predict predict stocks can be found in [34], [28] and [21]. This is why the system should have the ability to extend to ingest data from arbitrary secondary sources (f_6 and d_5 in the figure) to provide more versatile predictions about the stocks. The amount of this data can be unlimited but is restricted to relevant sources.

Data is ingested from these data sources mainly using HTTP-protocol as this is the main method that these services (d_1 - d_4) provide. Other possible methods that are usually available are Excel sheets and sometimes websockets, of which the websockets can be actually useful in cloud system, but as the HTTP-methods are currently the most used technology, this thesis is also going to focus on these. Here we have separated the main ingestion functions into two main types of functions f_1 and f_2. f_1 is constantly polling and processing data whereas f_2 handles batch processing. Both of these function store their raw output into the storage, but f_1 passes this also to the immediate technical analysis.

The system has two technical analysis functions. For methdos that allow streaming updates there is f_3, which can for example be cumulative/reinforced ML models and for methods that need historical data in order to calculate the prediction, there is f_4. For fundamental analysis, there is no a specific function as the introduced data sources usually provide these values pre-calculated and these values are usually easy to calculate dynamically with little to none amount of processing.

Finally, at the center of the system is the storage which is used to store the calculated predictions as well as the raw data from the data sources for later analysis. For historical technical analysis, f_4, the storage should provide reasonable range query times when quering historical data and for the results the storage should provide efficient point queries for the results ($<$ 1s).

1.2 Research Questions

The focus of this thesis is going to be the ingestion of stock data (f_1 and f_2) using big data technologies. As the field of possible technologies is large, the main focus of this thesis would be the comparison of current relevant technologies in the context of stock data to help developers to decide what technologies to use in their own projects. The main result of this thesis would be information and possible tools that developers could use to develop this kind of system more quickly. Developers here can be people from companies that either do stock analysis as their main business or just want to analyze stock data efficiently. These results could also be used in research to implement analyzing pipelines more efficiently so that the research group can focus on the analyzing of the stock data instead of worrying with getting the data to these parts.

Analyzing the stock data is also what most of the research today is focused on as this is the part that can actually produce profit. This means that most papers ignore the steps of ingesting and storing this data. Examples of this kind of papers are [33], [15] and [34]. So one of the goals of this thesis would be to bring also some practical knowledge on subject that has not been researched that much before.

This thesis will approach this subject from three different perpectives that cumulate on one another. These perspectives are represented by the following three research questions:

What are the needs and requirements of this kind of system data-wise? Which is important knowledge when starting to design or implement any kind of big data system. This thesis plans to provide the information on these so that when a reader is developing their own system they have a point of reference which they can use to evaluate their data sources.

What are technological options to implement this in practice? As there are enormous amount of different technological frameworks, this thesis plans to provide the reader information on the most relevant ones currently. This way the reader does not need to go through and learn variety of technologies in order to build their system.

How do the most viable big data ingestion options compare to one another in the context of stock data perfomance- and development-wise? Finally the thesis plans to provide comparison and prototype implementations of stock data ingestion using the technologies that seem most prominent. The metrics used to compare these systems would be the response time, possible scalability and the time it takes to develop these systems. This is to save possible time of a person developing these when the trial and error is done beforehand and if the prototype system fits the developed architecture it can also be used as a basis for further development by the reader.

1.3 Expected Outcome

For the first research question "What are the needs and requirements of this kind of system data-wise?" this thesis plans to provide an analysis of the necessary stock market data and its usages. From this analysis, the thesis would derive the main requirements for the system to fulfill in order to satisfy the needs of the possible subsequent analysis stage. This result could then be used in the future if one would want to build their own ingestion system from the ground up as a base.

For the second question "What are technological options to implement this in practice?" the thesis would perform an analysis on the current trends in data ingestion solutions. The thesis would provide information on the latest open-source technologies that could be used to implement this kind of system, how these would fulfill the requirements introduced by the first research question with application scenario and conclude this with a comparison of these technologies on what are the advantages of using one over another. The result of this part could be used to decide what seems to be the most suitable technology to use to implement the application scenario technically.

For the last question "How do the most viable big data ingestion options compare to one another in the context of stock data perfomance-wise?" the thesis would implement open-source prototype solutions based on the results of the second research question. This prototype could be used by anybody (company or individual) as it is or as a base to build a more complex system on top of it. The system would be targeted to practically implement the described application scenario, which could be used by companies with similar products allowing them to write more complex analysis algorithms based on the larger amount of data.

1.4 Structure of the Thesis

In the first chapter of the thesis we will be focusing on solving the first research question "What are the needs and requirements of this kind of system data-wise?". The thesis would start by going through scientific papers about stock markets and stock analysis. There would be first text generally about the characterics of the stock markets, what are they based on, what affects them, can they actually be predicted (random walk hypothesis) and so on. Then the thesis would go through the main directions of analysing the stock markets (fundamental analysis, technical analysis) and explain briefly some of the methods (about three methods per direction) that characterize these directions (Gordon model, Magic formula, LSTM etc.) focusing on the data that these methods need in order to calculate their predictions. This part would be concluded by deriving the requirements for the system based partly on these analysis methods and their data needs, and partly on the definitions that make up a scalable, secure and stable cloud system. The next step would be solving the second research questions "What are technological options to implement this in practice?". This step would perform literary research on what is currently used to perform big-data ingestion and storing, selecting from the list of technologies mostly those that seem to fulfill the requirements derived in the step 1 and would fit in the application scenario introduced in figure 1, specifically in f_1 and f_2. For data ingestion, these technologies would probably be Apache NIFI, Apache Flume, Fluentd etc. For data storage, these could be HDFS, HDFS, Apache HBase, Apache Cassandra etc.

This step would consist of first introducing all of the selected technologies and going through how do they work, what are they supposed to solve and what are the advantages and disadvantages of using one. After this, the section would do a comparison of these technologies in the context of stock data and conclude with analysis on which of the technologies would be the most prominent ones to solve this problem. After this would start the experimental part of the thesis and the rest of thesis would be focusing on solving the final research question "How do the most viable big data ingestion options compare to one another in the context of stock data perfomance-wise?". Based on the results of the step 2, I would implement couple of the most prominent solutions as a prototype systems. This step would include the actual implementations and reporting of these implementations. The report would consist of technical details; what parts does each of the system consists of, what versions were used, where the system was run etc. And subjective remarks; was it easy to implement, was there parts that did not fit together etc. The reporting part would also explain the metrics that would be measured for the subsequent analysis. For these metrics, the dataset used to test the system would be the open-source REST API from IEX [1], which is open for consumers until first of June, 2019. These metrics could be, for example, the time it takes to batch process, processor usage, memory usage, database fetching times etc. As these systems would be run inside Docker containers, the tools used to measure these metrics would most probably be programming language specific functions and Docker specific statistics tools. In the final step, the thesis would first inspect the results from the step 3 and based on these make remarks on what could be the best potentially the best implementation in this context. After this there would be an wrap-up on each of the previous sections concluding in retrospective what could've been possibly done better and what could be done in the future, concluding in recommendation what could be based on this thesis the best technical solution to implement system described in the application scenario.

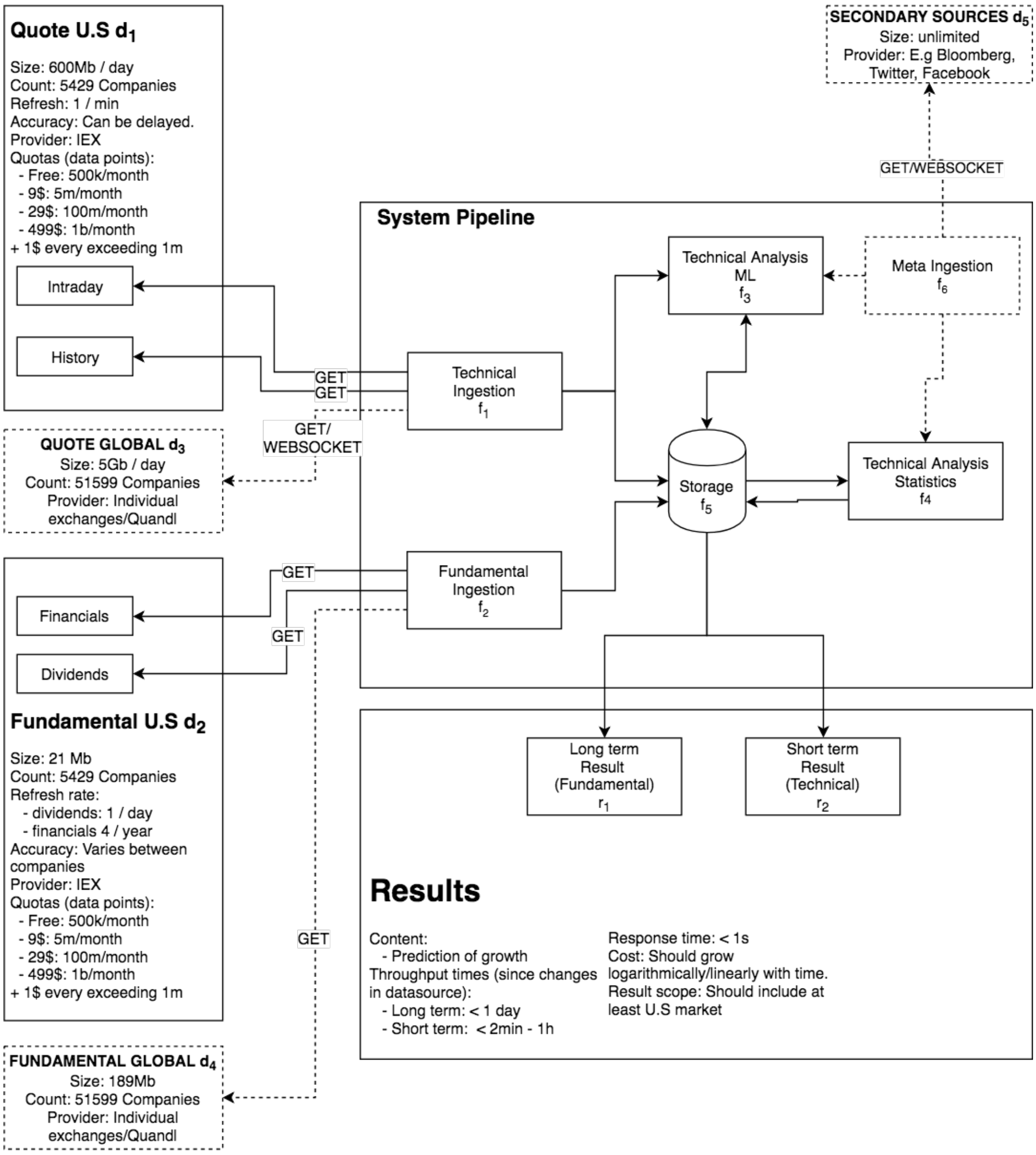

Figure 1.1: Application scenario for the thesis

Chapter 2

Stock market analysis

In order to build practical stock analysis systems, we start by looking at the methods of stock data analysis to understand the underlying domain. Stock market analysis is an enormous domain by itself and there are multiple ways to approach this data. For example possible problems to solve are finding anomalies in the data, predicting the future price and analysing possible causalities. In this thesis we will be approaching this from the price prediction perspective which can also be used in other problems such as anomaly detection [12].

In this chapter we examine what are the state of art methods that researchers use to analyse stock market data and we are especially focusing on methods that in some way use big data for this analysis. We will be also looking at the implementation side of things and examine some of the existing big data pipelines for stock market analysis and what are the technologies used to build these. As we will see, big data can be part of the pipeline in the form of complete stock data in which case the main method of analysing this is technical analysis. Or the big data can aspect can come from other sources of information such as news and social media in which case the main method of analysis is fundamental analysis.

2.1 Stock price prediction

The current methods on stock price prediction can be divided roughly into two different categories; statistical methods and machine learning methods. Both of these categories can be further divided into fundamental and technical analysis categories depending on what data they use as a source of analysis, but these categories overlap a little because of new ensembled models. With statistical methods we mean here the traditional mathematical models that need quite a lot of understanding of the domain in order to derive them. With machine learning methods, we mean algorithms and statistical models that derive underlying principles from data using patterns and inference.

Both of these methods are trying to beat the efficient market hypothesis (EMH). EMH states that the all the current public information available should already be seen in the price of the stock. In other words, the only thing that can affect the price of the stock would be unknown new information and the randomness of the system, which leads to a claim that stock prices can not be predicted using historical data. However, there have been multiple studies shown that this hypothesis could possibly be beaten using big data. [23] This hypothesis has also been challenged with overreaction hypothesis that states that the market overreacts to new information making it possible to somewhat predict the market before the prices change [8].

2.1.1 Statistical methods

The driving force of stock analysis in the past have been the statistical methods and there are still a lot of new papers analysing stocks using these methods. There are hunreds of ways to approach this problem from statistician point of view so here we have listed only some of those that have some relevance to the subject of this thesis.

From the technical analysis perpective, where only the stock data is analysed, we are going to be looking at momentum indicators. [30] Momentum indicators, as the implies, measure the speed of change in the stock prices and investors make decisions based on the thresholds of these values whetever a stock is being overbought or oversold. [3] When searching research on big data usage in stock analysis there are some key indicators that show up frequently. These are relative strength index (RSI), moving average convergence divergence MACD and Williams %R. These indicators can be used as they are but these have been also used as features that machine learning algorithms use to predics prices. [27]

All of these indicators measure the momentum of the price, but they all do it differently. RSI and Williams %R are both called oscillators because their values oscillate between maximum and minimum values they can get. Where as MACD compares long-term and short-term trends to predict if there is currently a notable trend. Because of the differences in the formulas and the factors that these values are measuring, the results from these formulas can be conflicting. [3]

Autoregressive integrated moving average (ARIMA) models have also been successful way of analysing time series stock data and they have been also used with big data. [31] However, with the new advances in machine learning, there seems to better performance to be achieved with machine learning methods. [14] Due to this and the complexity of ARIMA models, we are going to only briefly make a note of them here and focus more on the machine learning methods in the next section.

In the fundamental analysis side, most of the recent developments have happened with textual data from news and social media using machine learning methods meaning that traditional statistical methods have not gained that much interest recently. However, as there is a lot of relevant financial big data that can be used with traditional methods, here are couple of recent examples of the usage of this kind data: Day et al. [8] tried to link oil prices to stock prices using global financial data streams with stochastic oscillator techniques. Kyo [17] combined technical and fundamental analysis by using regressive model that takes into account business cycles in Japan's stock market.

2.1.2 Machine learning methods

As stated before, machine learning is the current trending stock analysis methology that has gained a lot of interest. Both supervised and unsupervised learning techniques has been tried to predict the prices but recently neural networks especially have gained a lot of interest due to their adaptablity to any non-linear data. Neural networks have the advantage that they usually do not need any prior knowledge about the problem and this is the case when we are looking at seemingly random stock market data.

Let us start by looking at neural networks used to predict the market using only the market data (technical analysis data). There are multiple different neural network architectures to choose from and the most popular one currently seems to be a basic multilayered feed forward network (MFF) when analysing only the stock market data. There is some results that have shown that increasing the size of MFF, by adding layers and neurons, can produce better results. This however, increases the amount of needed computation and will probably have some upper bound before it starts to overfit the training data. [26]

Although MFF is seemingly the most popular, better results have been able to achieve with convolutional neural networks (CNN) and long short-term memory (LSTM). With convolutional neural networks, the upper hand to regular MFF seems to be the ability to express same amount of complexity with smaller amount neurons, although no direct comparisons have been made with these two. With LSTM however, it has been directly shown that it performs better than other memory-free machine learning methods including MFF. As stock data is literally a time series, the LSTM's ability to remember previous states seems to separate it with other methods. There currently seems to be no papers on how the LSTM performs compared to CNN so we can only assume that the performance of these two is pretty close when it comes to the complexity of the network. After these standalone architectures the next step to seem to be hybrid architectures combining more than one model into one and this has already achieved seemingly better results than these standalone models. [26]

In order to train a neural network to predict stock markets we need features and classes to label these features correctly. Because there is such a huge amount of data in stock transactions, manual labeling is not an option. The simplest way automatically generate features is to take calculate change in price in each time step. This way the network can for example output simple binary classification telling whetever the price is changing more or less than median price. [9] When we take this to a bit more complex level, similar features can be made from RSI, MACD and Williams %R values which we introduced in the previous section. [27]

When we move on to the fundamental analysis, similar kind of networks are used to with financial news and social media data. Again LSTM and ensemble models are used to connect this data from outside with the price trend of stock market. There are research using just general social media data that concerns the whole market but there are also usages of stock specific posts. Fundamental data that is in textual form, term frequency–inverse document frequency (TF-IDF) is a common choice to present features [19]. This data is then classified based on the general sentiment that they seem to represent. Positive sentiment toward company would lead to the rise of stock price and negative sentiment vice versa.

2.2 Existing pipelines

Now that we have some kind of understanding what the current stock analysis is, let's move on to examine the actual implemetations that do this analysis. For this section, we examined 19 different stock data analysis pipeline that handle or are able to handle big data in some sense. What we mean here by "in some sense" is that the big data might not have been the actual stock data but for example social media data that was used to analyse the actual stock data. Information about these pipelines were all publicly available, although most of the pipelines were not open-sourced. Fourteen of these pipelines were reported in academic publications and the rest five are, or at least have been, in industrial use. There were also multiple open-source pipelines that have been developed for big data stock analysis, but information about the actual usage of these pipelines were not found. This is why we excluded these from this study in order to get results that might have more relevance to pipelines that are actually in use.

We have divided this section into subsections based on different parts from the pipeline starting from the furthest away of the possible clients and moving our way up towards the analysis phase. The biggest problem here is that the companies that work in the forefront of stock analysis, seldom share their software achitectures for business reasons. Nevertheless, with the public information available we can get an excellent view of the state of art pipelines in academic world and a general idea how the pipelines are build in the industry side.

It is also not that easy to compare technologies used in academia and industry. Both have different needs and goals that they wish to achieve. In academia, the pipeline is usually made in ad-hoc manner trying to minimize the time used for development and maximise the time needed for testing. This is possible, because researchers do not have same client side restrictions that industry might have. In industry side, it is usually important that the pipeline stays operational during the whole live cycle of the system. Where researchers only need these pipelines in order to conduct a couple of experiments, these industry pipelines have to survive possibly multiple of years of usage. These industry pipelines also serve the results to multiple clients that all expect small latencies from the system in order to use the system efficiently whereas in academia, this is not a requirement but helps the research to be made efficiently. So bearing these domain-specific requirements in mind, different technologies can be used to achieve optimal solution in different situations.

2.2.1 Data sources

Stock market data is not cheap. This is mostly because the exchanges that run the stock markets, usually make most of their profit with trade data and thus do not want to give this data freely away. There are however services who do provide part of this data with a much lower cost available to companies and researches which cannot possibly afford the complete real-time data. This partial data usually consist of historical end of day data, which is the aggregated statisctics of the stocks after the market has closed for the day. Although this data is complete in the sense that it tells all the necessary information about stocks price evolution on day level, we will be referring this data as the aggregated data during this thesis and reserve the term complete stock market data for the data that contains all of the transactions. What this means for the systems is that the stock data can exponentially smaller at the beginning of a financial companys life when they possibly have the ability to access the complete stock market data, but system must be able to scale to this complete data set size.

In academic papers, the Yahoo Finance API has been the de facto service used as the source of this partial stock data. [12] [7] [18] [27] Unfortunately, although this service have been used in papers published in 2019, this API was shutdown by Yahoo in 2017. Today, there are no service that provides the same amount of data that this API did, but some substituting services do exist.[20]

2.2.2 Used technologies

Next let's look at the actual technologies and frameworks that are used to implement this pipeline by starting from the furthest away from the client; ingestion layer. What we mean by ingestion here is the fetching and preprocessing of the data. This step with stock market data varies a lot depending who is fetching the data and for what purpose. Most common method for stock market data ingestion is just using custom scripts. This is usually enough with the aggregated stock market data but it does not scale.

Common big data technologies usually come in to play when the system has to ingest for example huge amounts of textual data such as news and social media feeds. For these purposes, the most common technology in scientific papers is Apache Flume which for example used in [25] and [6]. Another framework that is commonly reported in the ingestion step is the Apache Kafka streaming platform. [19] [5] There are also indications of usage of Google Dataflow in industry side. [24]

As we have now this ingested data, we need to store it somewhere. Stock market data is quite structured but the data usually used to enrich this data is mostly unstructured. This puts a restriction on what are the possible storage options available. Academic papers usually do not have to worry about this as the systems just have support ad hoc calculations but in the company context this becomes a crucial part of the system as it is the component that enables low latency responses without having to fetch data all the way from the original data source. With big data systems there is usually two options which are either cloud platform specific products or open-source HDFS-based systems and this is also the case with stock analysis systems. From the cloud platform products, there are information on usage of Amazons S3 and Googles BigTable with BigQuery. [29] [24] In the open-source side HBase [11] and Cassandra are the two used database solutions.

Then as we finally have the data in control, we can apply some analysis to it. In the analysis step, there is not that much variety in used technologies. Apache Spark with its MLlib library is dominating this field with its capabilities to process data efficiently. [12] [4] [19] [7] Where Apache Hive and Apache Pig were previously most used technologies, now Spark seems to be taking their place [29].

In the industry side, there is indications of Apache Storm being used with real-time classification [5]. The strong point of Spark is that same models can be used with Spark Streaming framework, but better latencies can be achieved with Storm making it possibly better choice for companies which have resources to implement two separate systems [16].

2.2.3 Problem of existing pipelines

In most of the pipelines, the system is produced so that in can scale with the input allowing no real perfomance issues to rise. None of the cases however, report how they monitor the system while it is running. In cases where for example Spark is used, we can assume that the process is monitored using Sparks own monitoring tools which measure load and resource usages. These tools are valuable to measure the perfomance of the application, but they do not always tell the whole truth about the applications state.

Let's look at an example case, where we have machine learning model training pipeline implemented in Spark which reads stock data from the database and trains a model based on this. What would happen if this data corrupted on the way to the training algorithm. In some cases the corruption would make the training algorithm possibly crash which would be noticed with performance monitoring tools. However, in some cases the corruption could only make the value of the data change ever so slightly that it would pass as a input to the algorithm. In this case, this could affect the final trained model making the model performance decrease tremendously. Which could then lead to even false conclusions on the model.

So in order to make right decisions concerning models and save time while training, it is crucial to identify these kinds of problems early as possible. This is where good monitoring comes into play. In the next chapters, we are first going to take a look at different monitoring solutions and then implement these on the most common stock data machine learning pipeline in order to solve this problem.

Chapter 3

Environment

Chapter 4

Methods

Chapter 5

Implementation

Chapter 6

Evaluation

You have done your work, but that's[1] not enough.

You also need to evaluate how well your implementation works. The nature of the evaluation depends on your problem, your method, and your implementation that are all described in the thesis before this chapter. If you have created a program for exact-text matching, then you measure how long it takes for your implementation to search for different patterns, and compare it against the implementation that was used before. If you have designed a process for managing software projects, you perhaps interview people working with a waterfall-style management process, have them adapt your management process, and interview them again after they have worked with your process for some time. See what's changed.

The important thing is that you can evaluate your success somehow. Remember that you do not have to succeed in making something spectacular; a total implementation failure may still give grounds for a very good master's thesis—if you can analyze what went wrong and what should have been done.

[1]By the way, do *not* use shorthands like this in your text! It is not professional! Always write out all the words: “that is”.

Chapter 7

Discussion

Chapter 8

Conclusions

Bibliography

[1] https://iextrading.com/.

[2] Bartram, S. M., and Grinblatt, M. Agnostic Fundamental Analysis Works. *Journal of Financial Economics (JRE)* (June 20 2017).

[3] Chen, J. *Essentials of Technical Analysis for Financial Markets*. John Wiley & Sons Inc, 2010.

[4] Chen, L., and Yang, C.-Y. Stock price prediction via financial news sentiment analysis. https://github.com/Finance-And-ML/US-Stock-Prediction-Using-ML-And-Spark Accessed: 08.05.19.

[5] Cheng, J. Real time machine learning architecture and sentiment analysis applied to finance. Slide set available at: https://www.slideshare.net/Quantopian/real-time-machine-learning-architecture-and-sentiment-analysis-applied-to-finance Accessed: 08.05.19.

[6] Das, S., Behera, R. K., Kumar, M., and Rath, S. K. Real-time sentiment analysis of twitter streaming data for stock prediction. *International Conference on Computational Intelligence and Data Science* (2018).

[7] David Andrešić and Petr Šaloun and Ioannis Anagnostopoulos. Efficient big data analysis on a single machine using apache spark and self-organizing map libraries. *12th International Workshop on Semantic and Social Media Adaptation and Personalization* (2017).

[8] Day, M.-Y., Ni, Y., and Huang, P. Trading as sharp movements in oil prices and technical trading signals emitted with big data concerns. *Physica A: Statistical Mechanics and its Applications 525* (2019).

[9] Fischer, T., and Krauss, C. Deep learning with long short-term memory networks for financial market predictions. *European Journal of Operational Research* (2017).

[10] Fox, J. *Myth of the Rational Market.* Harper Business, 2009.

[11] Gu, R., Zhou, Y., Wang, Z., Yuan, C., and Huang, Y. Penguin: Efficient query-based framework for replaying large scale historical data. *IEEE Transactions on parallel and distributed systems 29* (2018).

[12] Islam, S. R., Ghafoor, S. K., and Eberle, W. Mining illegal insider trading of stocks: A proactive approach. *IEEE International Conference on Big Data* (2018).

[13] John L. Person. *Mastering the Stock Market: High Probability Market Timing and Stock Selection Tools.* John Wiley & Sons, Incorporated, 2013.

[14] Khashei, M., and Hajirahimi, Z. A comparative study of series arima/mlp hybrid models for stock price forecasting. *Communications in Statistics Simulation and Computation* (2018).

[15] Kiarash, A., and Abbas, K. A New Approach to Predict Stock Big Data by combination of Neural Networks and Harmony Search Algorithm. *International Journal of Computer Science and Information Security 14* (2016).

[16] Kumar, H., and Kumar, M. P. Apache storm vs spark streaming. `https://www.ericsson.com/en/blog/2015/7/apache-storm-vs-spark-streaming` Accessed: 19.05.19.

[17] Kyo, K. Big data analysis of the dynamic effects of business cycles on stock prices in japan. *15th International Symposium on Pervasive Systems, Algorithms and Networks (I-SPAN)* (2018).

[18] Le, L., and Xie, Y. Recurrent embedding kernel for predicting stock daily direction. *IEEE/ACM 5th International Conference on Big Data Computing Applications and Technologies* (2018).

[19] Lee, C., and Paik, I. Stock market analysis from twitter and news based on streaming big data infrastructure. *IEEE 8th International Conference on Awareness Science and Technology* (2017).

[20] LOTTER, J. C. Bye yahoo, and thanks for all the fish. `https://financial-hacker.com/bye-yahoo-and-thank-you-for-the-fish/` Accessed: 19.05.19.

[21] MUN, F. W. Big data, small pickings: Predicting the stock market with google trends. *The Journal of Index Investing 7* (2017).

[22] MURPHY, J. J. *Technical analysis financial markets: A comprehensive guide to trading methods and applications*. New York Institute of Finance, 1999.

[23] NAM, K., AND SEONG, N. Financial news-based stock movement prediction using causality analysis of influence in the korean stock market. *Decision Support Systems 117* (2019).

[24] PALMER, N., RICKER, T., AND PAGE, C. DATA & ANALYTICS: Analyzing 25 billion stock market events in an hour with NoOps on GCP. Available at: `https://www.youtube.com/watch?v=fqOpaCS117Q` Accessed: 08.05.19.

[25] PENG, Z. Stocks analysis and prediction using big data analytics. *International Conference on Intelligent Transportation, Big Data & Smart City* (2019).

[26] SENGUPTAA, S., BASAKA, S., SAIKIAB, P., PAULC, S., TSALAVOUTISD, V., ATIAHE, F., RAVIF, V., AND PETERS, A. A review of deep learning with special emphasis on architectures, applications and recent trends.

[27] SEZER, O. B., OZBAYOGLU, A. M., AND DOGDU, E. An artificial neural network-based stock trading system using technical analysis and big data framework.

[28] SKUZA MICHAL AND ROMANOWSKI ANDRZEJ. Sentiment analysis of twitter data within big data distributed environment for stock prediction. *2015 Federated Conference on Computer Science and Information Systems (FedCSIS)* (9 2015).

[29] SNIVELY, B. AWS re:Invent 2018: Big Data Analytics Architectural Patterns & Best Practices. Available at: `https://youtu.be/ovPheIbY7U8` Accessed: 08.05.19.

[30] UTTHAMMAJAI, K., AND LEESUTTHIPORNCHAI, P. Association mining on stock index indicators. *International Journal of Computer and Communication Engineering 4* (2015).

[31] WANG, W. A big data framework for stock price forecasting using fuzzy time series. *Multimedia Tools and Applications 77* (2018).

[32] WORLD FEDERATION OF EXCHANGES. Monthly report january 2019. https://www.world-exchanges.org/our-work/statistics Accessed 15.04.19.

[33] YANBIN, W., GUO YIQIANG, L. L., NI, H., AND LI, W. Trend analysis of variations in carbon stock using stock big data. *Cluster Computing 20* (2017).

[34] YU-CHENG, K., JONCHI, S., AND JIM-YUH, H. eWOM for Stock Market by Big Data Methods. *Journal of Accounting, Finance & Management Strategy 10* (2015).