Trinity College Dublin
Coláiste na Tríonóide, Baile Átha Cliath
The University of Dublin

BIG DATA ANALYTICS: RISK ESTIMATION USING THE HADOOP ARCHITECTURE

ANUVA SEHGAL

B.A.I. COMPUTER ENGINEERING

FINAL YEAR PROJECT APRIL 2020

SUPERVISOR: PROF. KHURSHID AHMAD

SCHOOL OF COMPUTER SCIENCE AND STATISTICS

O'Reilly Institute, Trinity College, Dublin 2, Ireland

DECLARATION

"I have read and I understand the plagiarism provisions in the General Regulations of the University Calendar for the current year, found at http://www.tcd.ie/calendar .

I have also completed the Online Tutorial on avoiding plagiarism 'Ready Steady Write', located at http://tcd-ie.libguides.com/plagiarism/ready-steady-write."

STUDENT NUMBER: 18338958

SIGNED: Anuva Sehgal

DATE: 26 April 2020

ACKNOWLEDGEMNT

I would like to thank my supervisor, Professor Khurshid Ahmad, for his support and guidance. Without his continuous efforts and motivation, this project wouldn't have been possible. I want to appreciate everything he did for me, especially in these exceptional circumstances in which we all see ourselves today. His understanding and compassion made me feel very confident about my work and I am really grateful to him, for helping me at every step of the way. I have learnt a lot, both academically and personally, from him. Because of him, I am now so well informed about the financial industry and the technologies that are constantly being used by it. I can't express my gratitude enough, in words, but I hope my work speaks for me and makes him proud.

I would also like to express appreciation and love for my parents, Mrs. Neena Sehgal and Mr. Sameer Sehgal, for always being there for me and giving me a constant moral support during my time away from home. Without their encouragement, this journey wouldn't have been easy. A special thanks to my friend, Kunal Garg, for helping me and giving me the right advice whenever I needed it the most. I thank my friends and family to make this journey educative and memorable.

ABSTRACT

Big Data has been the “buzz” word in the IT world in recent years. It has witnessed a rapid rise with the advancement in science and technology. Big data analytics is being used by almost every industry, to gain an edge over the others. Big data technologies are becoming faster, more scalable and more secure by every passing year. The need to manage the massive amounts of volumes both in terms of storage and processing, calls for more compatible and available platforms. As the size and rate of generation of data increases, organizations seek low latency architectures to process data. The need for high throughput and low-response time is prevalent in the financial sector for risk estimations especially in trading activities. This project focusses on using high-frequency stock market data to apply to latency critical architectures for risk estimation using statistical metrics. It also proposes a statistical approach to data generation in order to make market behavior more apparent.

TABLE OF CONTENTS

LIST OF FIGURES & TABLES

1. INTRODUCTION

Advancement and digitization of information systems has been causing a data explosion in recent years. Data is being produced at massive rates, not only in terms of its volume but also the speed at which it is generated. This data can be present in any format: structured, unstructured, text, audio/video, numerical etc. Data is called "Bid Data" only when it satisfies some characteristic properties. The field of big data analytics comes across many challenges that can't be dealt using conventional methods. These challenges arise due to anything ranging from its heterogeneity and complexity to its velocity and size. To explore the full potential of big data, the use of appropriate platforms plays a vital role. The most well-established platform, the Hadoop Framework, provides suitable tools to manage and process this kind of big data. It offers cost-effective cluster computing to maintain affordability when the volume becomes too large. Big data has been an eminent topic of research and development as organizations compete to exploit the insights generated from it, and make profits. It can drive decision making for industries across all fields and disciplines. The revolution of big data has reached all industrial sectors and they thrive on it.

The financial sector has been one of the earliest industries to step into the world of big data and they have been evolving ever since. Big data gives organizations the power, to take advantage of the data all around and convert it into profits. One way of generating profits, is to mitigate risks in the business. Big data analytics is often used to perform risk estimations that provide companies a competitive edge over others. Massive amounts of data are produced in trading activities which may be homogenous numerical data, but the velocity at which they are created, makes up for the complexity. Trading involves making fast, split-second decisions in order to gain something out of it. These time-critical activities call for low latency architectures which provide low-response time. Latency can be reduced in many ways, and different architectures achieve this differently. There are however, a common approach taken by several architectures (including the one used in this project): Parallelism.

This project discusses the techniques adopted by Hadoop to reduce latency as well as the various risk estimation metrics used in the financial sector. It also offers an implementation of how the two can be integrated to create value.

2. MOTIVATION AND LITERATURE REVIEW

2.1 WHAT IS BIG DATA?

Big Data is defined as "extremely large data sets that may be analyzed computationally to reveal patterns, trends, and associations, especially relating to human behavior and interactions."[0] Specifically, Big Data can refer to massive amount of heterogenous data which is generated at high velocities and grow at ever-increasing rates. The term heterogonous data has many different meanings: The data can originate from many (sub-) systems, or different types of data have their origination in the same system or different system, or the data maybe in different formats. My motivation is to look at high-value data that is sampled at high frequency that requires low latency processing. I am looking at stock market data where one has to analyze current data in conjunction with a historical data to see signs of turbulence in the stock markets. I have chosen stock market data in that has the 5 non-overlapping characteristics of any Big Data collection:

1. **Volume**: Volume refers to the size of the data. As the term "Big Data" itself suggests, volume is one of the differentiating aspects that defines Big Data. Recent data explosion has been caused by the increasing use of internet and the growth of information systems. It is highly imperative for organizations to effectively store and process this unprecedented amount of data in order to gain business insights and make faster decisions. The benefit gained from extracting insights from previously untapped data sources and combining them to create value for an organization, is what the real driver of big data analysis is all about.

2. **Velocity**: Velocity refers to the rate of flow of data. The speed at which the data is being generated and transmitted/moved around defines its velocity[3]. Data streams can be generated or processed in batches or in real-time. Velocity is closely related to the performance of big data techniques and frameworks as it's a perfect indicator for latency related issues. The speed at which data is accessed and analyzed is directly impacting the accuracy and swiftness of organizational decisions.

3. **Variety**: Variety refers to the heterogeneity of data. It refers to the multiple formats in which data is presented as well as the various sources from where the data comes from. Big data can be structured or unstructured, meaning that it can either be created within the RDBMS structure or might not have a pre-defined data model/schema. As variety increases, the complexity also increases[3]. The complexity of processing and creating insights out of data increases. Most of the data today, is unstructured and chaotic. Data can be in the form of numerical data in relational databases, text documents, emails, video, audio, financial transactions etc.

4. **Veracity**: Veracity refers to the truthfulness of data. It defines how clean and accurate the data is. The quality of data being captured from a source, varies greatly. Hence, veracity also accounts for the authenticity of the source or how reliable it is. A lot depends on how credible the data is, as business decisions need to be made on the basis of it. The accuracy of insights speaks volumes of the quality of analysis.

5. **Value**: Value is the most crucial characteristic of all, as without it, there is no context to our studies or techniques. Value refers to the usefulness and meaning of the generated and processed data. The "insights" mentioned above, is what value is. There is no point of collecting data if we don't turn it into value. It refers to the worth of data, in terms of how impactful it is, in an organization's operational and strategic decisions. The value, the end result or the generated insights lead to measurable improvements and decision-making.

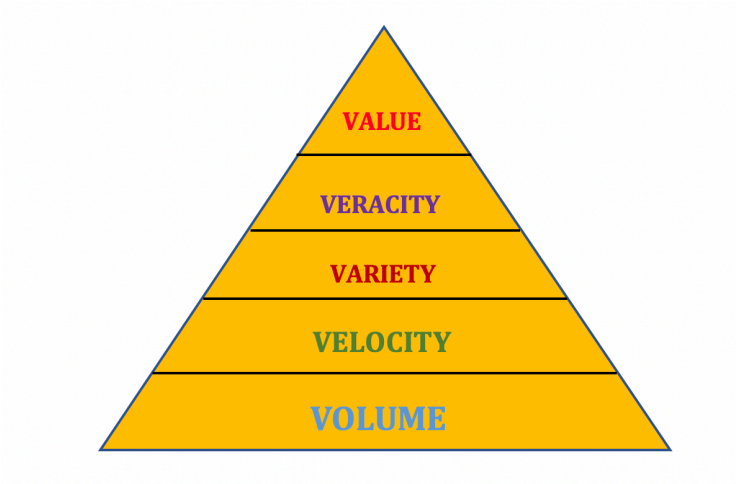


Fig 1: Big Data Pyramid representing the 5 Vs

2.2 POTENTIALITIES OF BIG DATA

Terabytes and petabytes worth of data is being generated in today's "data-centric" world, via experiments, surveys, simulations, recordings etc. This rapid growth in information calls for new paradigms to be introduced in order to address the challenges accompanied by the use of big data across different disciplines[2]. These disciplines span from the financial, commercial, defense, telecommunications sectors to healthcare, e-commerce and internet-based companies. Some sectors within this vast spectrum of applications of big data and its technologies are mentioned below:

1. **Financial Sector**: The use of big data and its technologies in the financial sector can be proven very useful as not only does this industry generate huge amounts of data by the minute but can

also benefit greatly from its analysis. Data can be used to anticipate market sentiments and customer relations. Financial services provided by banks and similar institutions, can assess customer activities and develop products that meets customer needs. Big data analytics can also help banking and financial markets to provide accurate customer analytics, risk analysis and fraud detection[6]. Collected and processed data can also cause tangible and intangible benefits in business operations. For example, modern day technologies used with big data, allows companies to effectively reduce costs or discover business opportunities for expansion. The use of big data is not just limited by business operations and decisions but extends to intelligent trading activities and into the market in general. The voluminous amount of data collected, can inform traders about market trends and predict rational or irrational behaviors of traders in the market. The market is majorly governed by phycological and external factors which are very uncertain. The rapid growth of data makes the political, social and economic events impact the financial markets faster than they did before. Big data can help predict future variability and share price volatility and bring control back in the hands of the traders. There are numerous use cases of big data in this sector and the above mentioned are just some of them.

2. **Healthcare Sector**: Digitization in the field of healthcare along with the speed of technological advancements birthed digital health. Digital health is very important for the society in general, as it helps improve and strengthen precision medicine, medical practices, population health management, clinical decision support and quality benchmarking. With the help of big data and statistics, analysts can predict the possibility of an outbreak of an epidemic, and give institutions and societies enough time to brace for the sudden impact[1]. It can further help improve individual patients' health by monitoring their medical histories and prevent/prepare for future illnesses. For example, a heart patient's cardiological records could be stored and monitored so as to predict the next disorder and weaken it. Clinical information can also be used in telemedicine solutions and creating bio-technologies that can benefit every human the same. Since this field involves the disclosure of personal records, medical histories etc. it is always under scrutiny for adhering to data privacy policies. Apart from the persistent security and data protection aspect of digital health, medical information can not only advance the scientific and biomedical community and their researches but also directly and positively impact the lives of patients.

3. **Retail Sector**: Commercialization of commodities and utility products have seen groundbreaking changes since the past couple of years. With the introduction of big data and new technologies, sales have shot up due to targeted marketing. Information on consumers and consumer needs have drastically improved sales not only in the retail sector but e-commerce as well. Understanding consumer behavior has now become an essential part of marketing and this wouldn't have been possible without the data that is being collected day in and day out. Information or data here doesn't just mean the nature of transactions and purchases but also surveys led big retail companies to come up with better sales strategies. For example, in a grocery store small bags of chips, gum or some "grab-and-go" type stuff are kept around the checkout area.

The reason for that is marketing. People tend to buy things they might not need, in a hurry but not when they're given enough time to think about it. These little tricks are applied by retailers after studying customer consumption habits and preferences. Big data plays a major role in providing this market awareness[1]. Big data analytics gives retailers a competitive advantage which can be used to increase sales and promote better shopping experience for customers.

4. **Transport Sector**: Data in this sector can be applied to all modes of transportations. It can be used to improve road safety as well as traffic resolution strategies. It can also help optimize railway routes, facilitate planning, maintenance and event response. It can also be used to control air traffic, improve air safety and similarly with the waterways. Data from multiple sources can be combined to warn authorities of possible dangers. For example, weather data, data from social media, cameras and sensors etc. can help in faster and more informed actions.

5. **Media and Entertainment Sector**: Data in this sector is growing rapidly and has seen some sharp spikes in recent years. Social media is one of the most data-driven platform today. Data is being generated and consumed rapidly and in massive amounts. As the number of platforms is increasing so is the variety in formats of data. Combining these different types of data and creating value off of it is a complex task. But greater complexity means greater reward. Data is also consumed and produced via advertising, broadcasting, publishing and other digital platforms. Data recorded through these channels is then leveraged by e-commerce websites to market their products. Media associated businesses combine content, tailored to target an audience, with their platforms and create big data. This big data can be used for any purpose, be it educational, informative, recreational, preventive etc.

6. **Manufacturing Sector**: Although it's a more recent realization, but the manufacturing sector can benefit from big data in a number of ways as well. Companies can reduce their waste or save their costs with the help of better management practices using big data analytics. Production processes can benefit from the improvement in quality and safety practices enabled by the collected data. Big data can also predict machine failure and reduce further downtime. Smart manufacturing can also be applied for autonomous production which is more data-driven and less manual. All these practices are taking advantage of the new growth in technologies and information systems. These systems are governed by data and can highly digitize the manufacturing process.

7. **Public Sector**: This is one of the sectors that can help the community as a whole, to make use of big data, directly or indirectly. There are many decisions made in this sector every day, that can be improved by adopting more data-driven strategies. Budget choices made by managers in a company or government officials, program prioritization, natural disaster and epidemic management, public infrastructure, political strategies, public policies etc. are some of the many strategic choices that are made on a daily basis by the organizations or the government[1]. These choices can't be made subjectively. In order to ensure the most efficient of approaches to these

operational and strategic decisions, generated big data needs to be used. This will not only lead to better decisions but effective implementation as well. It can further help administrations to fight off corruption, improve safety in cities, improve air quality and many other social causes that are in desperate need of government attention.

8. **Leisure Sector**: This industry pertains to businesses focused on recreation, entertainment, sports and tourism related products and services. Big data analytics can boost income of this industry by improving marketing strategies, infrastructural capacities, management moves etc. For example, organizations or the government can study the seasonal tourism and make use of the collected big data and form ways to increase it. The amount of data generated by these industries can be massive and contain trend patterns that can be studied and made use of. The travel industries are taking advantage of big data for customizing user experience, improving pricing strategies and to gain a competitive edge over others. It is similarly impacting hotels and the hospitality sector as they cater to millions of travelers each day. Meeting customer expectations is essential for their loyalty, and organizations are turning to advanced analytics to keep that going. Since this sector is generally more customer-driven, analytics help focus on factors like seasonality, customer identification, customer sentiment etc. There are multiple features that can be assessed and improved, each leading to the growth of this sector in recent years.

2.3 APPLICATION REVIEW

Application Sector	Technology Considerations for "Big Data"					General Comments	References
	Volume	Velocity	Variety	Veracity	Value		
Leisure	H	L	L	L	L	Not Real-time information based	https://www.socialmediatoday.com/content/big-data-tourism-hospitality-4-key-components
Media and Entertainment	H	H	H	L	L	Variety needs be incorporated to create meaning	Lippell, H., 2016. Big Data in the Media and Entertainment Sectors. In New Horizons for a Data-Driven Economy (pp. 245-259). Springer, Cham.
Public	H	L	L	H	H	Security of data is the most esssential aspect	Klievink, B., Romijn, B.J., Cunningham, S. and de Bruijn, H., 2017. Big data in the public sector: Uncertainties and readiness. Information systems frontiers, 19(2), pp.267-283.
Transport	H	L	H	L	L	Security in terms of travel information needs to be protected	Lv, Y., Duan, Y., Kang, W., Li, Z. and Wang, F.Y., 2014. Traffic flow prediction with big data: a deep learning approach. IEEE Transactions on Intelligent Transportation Systems, 16(2), pp.865-873.
Retail	H	L	H	H	H	Real-time analysis is not as important as the variety here	Fosso Wamba, S., Akter, S. and Bonicoli, M.P., 2013, December. Realizing 'Big Data'-enabled business value: Insights from top retailers'. Oxford Retail Futures Conference.
Manufacturing	H	L	H	H	H	Sensory information, requires very fast actions hence other technologies shouls be used	Zhong, R.Y., Newman, S.T., Huang, G.Q. and Lan, S., 2016. Big Data for supply chain management in the service and manufacturing sectors: Challenges, opportunities, and future perspectives. Computers & Industrial Engineering, 101, pp.572-591.
Healthcare	H	L	H	H	H	Most valuable insights provided and hence no scope for error	Groves, P., Kayyali, B., Knott, D. and Kuiken, S.V., 2016. The'big data'revolution in healthcare: Accelerating value and innovation.
Financial	H	H	H	H	H	Both Historical data and real-time data can be used to carry out risk management and in order to make decisions	Hussain, K. and Prieto, E., 2016. Big data in the finance and insurance sectors. In New Horizons for a Data-Driven Economy (pp. 209-223). Springer, Cham.

Table 1: Application review of different sectors

The above table summarises the application review for big data and its technologies. It has been sorted in an ascending order according to the characteristics of big data. Here, H means "High" and L means "Low"

and they signify the magnitude of the big data considerations. "L" in terms of "Value", gives the relative importance of the industrial output generated from big data for the given sector. For instance, using big data analytics in the financial sector, helps companies make better trading decisions as well as estimate their risks and put in place appropriate technologies to drive profits. Whereas, if we look at the sector immediately above it i.e. the media and entertainment sector, the value of the insights generated by big data technologies will be lesser. Similarly, if we compare the value that comes out of the generated big data in the healthcare sector and say, the leisure sector, then it becomes obvious which one is given higher importance. As can be seen from the table above, the financial sector generates data with all 5 considerations for big data having a large magnitude. This means that the financial sector produces the largest volume of data, with the highest of velocities, having a lot of variety and that it needs to be highly accurate in order to generate useful insights. As an additional benefit, financial data doesn't face much ethical dilemmas as companies publish metrics themselves. Due to these reasons, we now focus the project towards big data in the financial sector and look at the latency considerations for generated data streams.

2.4 FINANCIAL MARKET

The financial market is dynamic and is governed by the decisions of crowds of traders who are constantly trying to win the global 'game'. Variability in the market leads to financial risks that the traders should be wary of. One of the major indicators of market-wide risk is stock market index volatility[10]. Volatility is defined as "the extent to which the value of a security, bond, commodity, market index, etc. varies over time. If an asset shows high volatility, it exhibits large, frequent, and sudden fluctuations in price; it is therefore associated with a higher degree of risk"[11]. Billions of dollars can be lost because of poor management of these risks and hence cause financial disasters. It is thus imperative for companies to understand the price dynamics and how they change. There are a couple of different methods that investors use to analyse capital markets in order to estimate risks and benefits. The most common methods are: Fundamental Analysis and Technical Analysis. This project focusses more towards the latter, but it's important to introduce both and understand the difference.

2.4.1 Fundamental Analysis

This approach involves analysing various macro and microeconomic factors and determining the true value of a security. A security can be overvalued, undervalued and fairly valued[14]. The variation in the prices of these securities depends primarily on "country" and "industry" factors[12]. The difference between the two can be compared to a top-down and bottom-up approach respectively. The "industrial" approach is followed when international returns are believed to be predominantly driven by industry factors. This approach involves portfolio allocation to several industries and then selection from the most attractive stocks among those sectors. Whereas for those, who believe that domestic market factors are more important, go with the "country" approach. Here, they do country allocation first which is followed by selection of the most favourable stocks from each country[13]. The globalization of economy has increased

the chances of exposure to market risks and is the reason why companies should consider it while constructing their portfolios.

2.4.2 Technical Analysis

Today, Technical Analysis is a standard tool to forecast future price path for financial assets. It is a statistic approach which involves predicting probable future price movements for a security, based on market data- historical returns, stock prices, volume of trades etc[15]. It provides insights into the future activities of securities or general market movement. Technical analysts believe that past and present performance of a security can very well determine its future movement. It can also be considered a good indicator to pinpoint low-risk price levels. Different timeframes are considered by different traders that range from one-minute to monthly, or even yearly data. This project focusses on high frequency data and the time frames that we will examine here are:

- 1-minute (2^0) data
- 2-minute (2^1) data
- 4-minute (2^2) data
- 8-minute (2^3) data
- 16-minute (2^4) data
- 31-minute (2^5) data
- 64-minute (2^6) data
- 127-minute (26) data
- 256-minute (27) data

Fluctuations during daily trading data exposes many price patterns which an intraday trader analyses in order to increase benefits and decrease risks. There is a skew towards reliance on technical, rather than fundamental, analysis when it comes to short-term forecasting as it is the most accurate. Very little guess work goes into these predictions, as they are made from known facts and figures that are immediately behind in time. This is however, the opposite for fundamental analysis which performs better for long-range future predictions. Hence, it is used for determining the quality of long-term investments in the market. Studies from the past show that technical analysis is a broadly used tool in practice and some even suggest that technical analysis may be self-fulfilling[16]. It exploits the market information through gradual highs and lows of the asset prices and volumes of trades. It also detects changes in investors' sentiment as they cause certain recurrent patterns that can be further benefitted from.

Technical Analysis: Technical Indicators

There are many technical indicators that assist traders to make trading decisions. These technical indicators are just strategies or rules that can potentially exploit any positive dependence among returns. These trading rules are able to generate profitable investment decisions in some cases. The concept behind

these rules is to identify the current trend patterns in prices and exploit the predictability of returns when these patterns are fulfilled for a sufficiently long period of time. In this project, I use the most-widely used and frequently mentioned trading rule: **Moving Average Rule**. There are others namely: "The Filter Rule", "The Channel Rule", "Statistical Rule" and many more. These rules are just methods to convert historical prices into investment decisions[18]. Moving Average is defined as: " For a sequence of observations $x_1, x_2, x_3,\ldots$, the moving average of order n is the sequence of arithmetic means $\frac{x_1+\cdots+x_n}{n}$, $\frac{x_2+\cdots+x_{n+1}}{n}$, $\frac{x_3+\cdots+x_{n+2}}{n}$, …."[17]. Its main purpose is to offer a view of the underlying behaviour of volatile series, especially those which have periodic variation.

The moving average rule offers a comparison between two moving averages: short-term and long-term. We take two averages of lengths S (for short period of time) and L (for longer period) are calculate them at time t from the most recent price observations, including p_t :

$$a_{t,S} = \frac{1}{S}\sum_{j=1}^{S} p_{t-S+j}, \qquad a_{t,L} = \frac{1}{L}\sum_{j=1}^{L} p_{t-L+j}.$$

It's mentioned in some texts that the most popular parameter combinations for S and L have $S \leq 5$ and $L \geq 50$. After this, we calculate the relative difference between the short-term and long-term averages:

$$R_t = \frac{a_{t,S} - a_{t,L}}{a_{t,L}}.$$

The theory behind the rule suggests that if the short-term average is greater than the long-term average that means the recent prices are higher than the older prices and it can be argued that these prices are following an upward trend. Similarly, if the short-term average is smaller than the long-term average that means the recent prices are lower than the older prices and it can be said that the prices are following a downward trend. If these averages turn out to be similar, it may be that the information is not precise enough to form a view about the trend. The classification on whether to remain "Neutral" ,"Buy" or "Sell" the stock in the next time step i.e. *t+1,* depends on three parameters: S, L and B. B refers to the bandwidth and when its zero, (almost) all days are either Buys or Sells[18]:

$$\text{Buy if } R_t > B, \quad \text{Neutral if } -B \leqslant R_t \leqslant B, \quad \text{Sell if } R_t < -B.$$

These technical rules help traders make the most of equity return predictability and reduce risks. These rules are informative about the moments of return which are the statistical properties of the distribution and provide a better understanding of its shape. Although the degree of predictability has decreased over recent years[18], these rules can provide clarity when used with larger volumes. The moving-average rule used in this project, is equally applicable to high-frequency data as it is to low-frequency data. The

performance and need for these trading rules are highly debated but each rule performs differently across different traders and markets. For some, the market could be efficient but for others it might not be. Trading rules can have minimal systematic risk, even when the underlying asset is a stock index , and this is reason why it has been applied in this project.

2.5 RISK ESTIMATION OF RETURNS

Traders and investors are particularly interested in estimating market risk in investment in order to increase profitability. These risks are measured on returns of asset prices or stock prices. Using Returns are preferred over prices for many reasons, primary being that returns are not as correlated through time as prices and hence provide more promising statistical properties for analysis. Since this project uses high frequency data, further literature review will be done in terms of high-frequency observations. There is no difference in the way intraday returns and daily returns are defined. Returns are simply the change in the logarithm of the price during an interval of time[19] i.e.:

$$Return = r_t = ln\left(\frac{p_t}{p_{t-1}}\right)$$

Suppose we consider an *N* - minute interval instead of minutely observations, we will require one price for each interval. We use the simple average of all transaction prices within the *N* – minute interval as the stock price for the interval[20]. Volatility is a feature of asset prices, that contain the element of uncertainty for the whole distribution. For stock returns, it is referred as the rate at which the prices change[21] or a measure of price variability over some period of time. Investors and traders are particularly interested in estimating volatility, as higher levels imply higher chances of extreme fluctuations which further suggest higher associated risks. There are various definitions of asset volatility and for a stock return series, it might even not be directly observable. As a result, statistical methods prove to be useful while analyzing financial time series[20].

To understand asset returns and their behavior over time better, it is good to begin with their distributional properties and moments. The statistical properties of the distribution can explain the data and then can be used to make predictions. The general assumption, that asset return distributions are normally distributed is not true. We use moments of the distribution to find similarities between the normal distribution and the distribution of our collected asset returns. The Four moments of distribution that are of significance here are:

i) *Mean or expectation, central/first moment:*

$$\bar{r} = \frac{1}{n}\sum_{t=1}^{n} r_t$$

ii) *Standard deviation(s), second moment:*

$$s^2 = \frac{1}{n-1}\sum_{t=1}^{n}(r_t - \bar{r})^2,$$

iii) *Skewness(b), third moment:*

$$b = \frac{1}{n-1}\sum_{t=1}^{n}\frac{(r_t - \bar{r})^3}{s^3}$$

iv) *Kurtosis, fourth moment:*

$$k = \frac{1}{n-1}\sum_{t=1}^{n}\frac{(r_t - \bar{r})^4}{s^4}$$

The above definitions assume *r* is a discrete random variable. The summation can be changed to integral ($\int$) when working with continuous random variables. The third and fourth moments, skewness and kurtosis, are of particular importance while addressing normal (Gaussian) distributions. These moments will be discussed a little later along with the distribution effects on them. But, as a quick introduction to skewness and kurtosis, they are the measure of symmetry and similarity to the normal distribution respectively. We need this information to describe the normal distribution in terms of its statistical properties. The probability density of the normal distribution is symmetric about its mean which ensures that the skewness of this distribution is zero[28]. This implies that any skew (positive or negative) in our distribution automatically indicates that it is non-normal. The second and fourth moments are σ and $3\sigma^4$ respectively and hence, for a normal distribution the kurtosis has to be equal to three (since σ = 1 for normal distribution). There are certain properties that can be found in most of the set of daily returns as is the case for our collected data: i) the distribution is not normal, ii) there is no correlation between returns for different days iii) there is positive correlation between magnitudes of returns of nearby days that are statistically significant[29].

The first moment of a distribution, the mean, informs us about the central tendency of the distribution. For distribution of returns, it very common to have zero population mean. This is not surprising as stock series have next to equal amounts of positive returns and negative returns, if the market is stable. Overall stability of the market is characterized by positive returns (or negative returns) being eventually followed by negative returns (or positive returns). This implies that the mean of the distribution will be zero or very close to zero. However, if the market is overall profitable (or unprofitable) that will result in a positive (or negative) mean. This can be a very simple risk indicator. The second moment of distribution, the standard

deviation is a measure of dispersion of the distribution and is also known as volatility. It directly reflects the amount of fluctuations in the distribution. As the order of moment increase, the sensitivity to outliers increases. Studies have found that volatility shows a log-normal distribution[30]. These moments of distribution vary significantly depending on the day of the week, day of the month, month of the year and proximity of holidays. These are known as the calendar effects but their significance on risk estimation is weaker and won't be evaluated. Skewness and Kurtosis are the third and fourth moments respectively. As the power of observations has risen to third and fourth respectively, the impact of extreme values has increased. Symmetry of the distribution can indicate overall profitability (losses) if the skew is positive(negative). This can be attributed to the presence of extreme outliers. Kurtosis on the other hand, can be very useful in identifying values outside the standard deviation range. It is relatively easier to estimate risks in normal distributions as compared to non-normal distributions. When the kurtosis is less than or equal to three, 99.99% of the observations lie within $\pm 3\sigma$ of the mean[31]. This offers more accurate risk calculations. When Kurtosis value exceeds three, those sets are called *leptokurtic*. This means that the distribution is fat-tailed rather than thin-tailed as is the case in a normal distribution. Hence, when the kurtosis exceeds three, it implies that the distribution consists of significant outliers and is not Gaussian. Hence. risk estimation becomes complex.

These statistical properties shed a lot of light on how closely risks are related to distributions of stock return series. These predictions however, depend on historical prices of assets and address a lot of questions whether it concerns the direction of price movement, market volatility or future asset price. The main concern of this project is to estimate risks by calculating market volatility and solve computational latency issues that rise with it. The latter will be introduced next.

2.6 LATENCY CONSIDERATIONS

Across all sectors, one thing remains common - the need for low latency systems. Latency is one of the most essential measures for system performance. It is defined as the "measure of how long it takes for a given job or piece of work to be completed, or for a message to make its way from source to destination[9]". It refers to the delay in response to requests, queries or actions. Low latency systems respond faster to actions, giving companies the opportunity to earn huge profits. Latency is a challenge posed to all current system architectures. All the above-mentioned sectors along with others, highlight the fact that to truly take advantage of Big data, we need latency critical analysis. This can help organizations gain a competitive advantage by having the power of making faster decisions. Big data analytics would only be beneficial if the latency could be controlled, especially for businesses that risk management. For example, in high frequency trading, where large volumes of securities are traded within very short periods of time via intelligent systems and computers, a latency drop of a millisecond can boost revenue by 100 million per year[4]. Low latency can be achieved by using high performing computers whose processing capabilities are massive. High performing computers are like the normal desktop computers but just with more processors, memory, disks and a better operating system. Another approach to high performance computing could be

cluster computing. It's basically a scaling out process for when you can't afford a supercomputer or to add more CPUs to the existing computer, you simply connect multiple computing nodes in clusters and that gives you the benefit of multiple processors. This again wouldn't increase the latency as much as a high-performance computer because of the transmission time between the multiple computers itself. However, because of the large-scale data that is needed for analytics today, concerns have shifted from requirements of storage systems to the need for speed
for transmission of larger volumes of data. Big data is becoming too large and complex for existing architectures and hence it calls for an improvement. IDC predicts that the global datasphere will grow from 32 Zettabytes (a trillion gigabytes) in 2018 through 50 Zettabytes in 2020 to 175 Zettabytes in 2025[5]. This is the rate at which data explosion is occurring today and this is a never-ending expansion.

Latency critical big data analytics in finance, requires systems to not only provide absolute end-to-end latency in network connections but also, decreased processing speed of complex analytics on huge amounts of data[6]. It requires both hardware and software reliability for high performance and to capture trading opportunities. As volume increases, organizations demand near real-time latency so that value can be extracted before next observations are streamed in. For example, in this project the illusion has been created that the data is streaming in, which is actually just reading in historical prices every 1 minute. This means that the risk estimations and related calculations, which can be referred as the "value", need to be done before the next datapoint is read. If the system fails to do so, the delayed insight will be of no use to the user. Latency can be achieved in many ways, primary being parallelism, which can further be implemented on different levels in computation. This project takes a rather simplified approach and employs one of the more known Big Data architecture called Hadoop

2.7 HADOOP ARCHTECTURE

Hadoop is an open source implementation first introduced by Google and primarily based on distributed computing and parallel processing. This Big Data framework was developed to process and analyze large datasets. Since the past several years, Hadoop's computing model has been recognized as capable of processing petabytes of data, enabling large scale internet applications. This infrastructure has been created to provide for more storage capacity, parallel batch processing in a heterogenous environment[22]. Hadoop can also be used to achieve real-time analysis but it is more common used for the former. Hadoop offers greater flexibility, scalability, performance and lower latency, all in a cost-effective manner. It allows developers to write highly parallel codes without dealing with the intricacies of data distribution and fault tolerance. It has been designed to operate on commodity clusters which has replaced High-performance Computing (HPC)[23]. Commodity cluster computing refers to using large number of low-cost, relatively low-performance commodity computers working in parallel instead of using fewer high-performance and high-cost computers. This topic has sparked debates over whether to Scale-up or Scale-out for Hadoop. Scaling out by deploying large amounts of cheap commodity hardware has been proven to work better for the

workloads in a cost-effective manner whereas, scaling up by adding resources to increase performance of single server can lead to faster processing of "jobs" and be potentially cost and power effective[24]. Since this project uses Hadoop which has been developed using the standard scale-out thinking, which is to utilize the cheap commodity clusters via distributed file systems, we won't dig deep into this topic. Hadoop consists of three main components: HDFS (Hadoop Distributed File System) for storage, MapReduce (Computing Model) for parallel processing and YARN (Yet Another Resource Negotiator) for resource management in a cluster.

2.7.1 Hadoop Distributed File System (HDFS)

When data becomes too large to be stored on one machine, distributed file systems are needed to reduce workloads and improve storage capacities. HDFS is highly fault tolerant and is designed to be run on commodity low-cost hardware. It provides high throughput access to application data which reduces overall architecture latency[25]. An HDFS instance consists of hundreds or thousands of server machines connected together, communicating over a network and each containing different chunks of a file system. This means that the chances of hardware failure increase with the increase in connected servers. Hence, it becomes very important to make the architecture fault tolerant.

The size of a typical HDFS file is in gigabytes to terabytes. Therefore, in order to support such large files, it needs to scale to hundreds of nodes in a single cluster. HDFS follows a write-once-read-many access model which simplifies data coherency and allows high throughput data access which is compatible with applications like MapReduce. Another interesting feature about HDFS is that it enables capabilities for applications to migrate their computation closer to where the data is located rather than moving the data to where the application is running. This minimizes network delays and decreases overall latency of the system especially when the data is too large.

Architecture:

HDFS has a master-slave architecture where a "NameNode" acts as a master server and a "DataNode" which can be considered as the slave server (Fig 2). There are two types of DataNodes: primary and secondary. The former is responsible for managing the file system namespaces, metadata and regulating file access[25]. The latter checks namenode information and helps restart the system in times of failure[26]. NameNodes also keep track of which DataNodes are alive using periodically transmitted signals called Heartbeats[22]. If a heartbeat is missing, the NameNode removes the failed node and re-distributes the workload equally in order to maintain high availability and provide fault tolerance. DataNodes on the other hand, manage storage attached to the nodes they run on. There is usually one DataNode per node in the cluster and a single NameNode for the whole cluster. The files that are inputted into the HDFS are first split up into a sequence of blocks/chunks with fixed sizes and are then stored in a set of DataNodes. The block sizes are typically 127 MB except for the last one. HDFS makes sure that each of these chunks reside in different DataNodes, if possible.

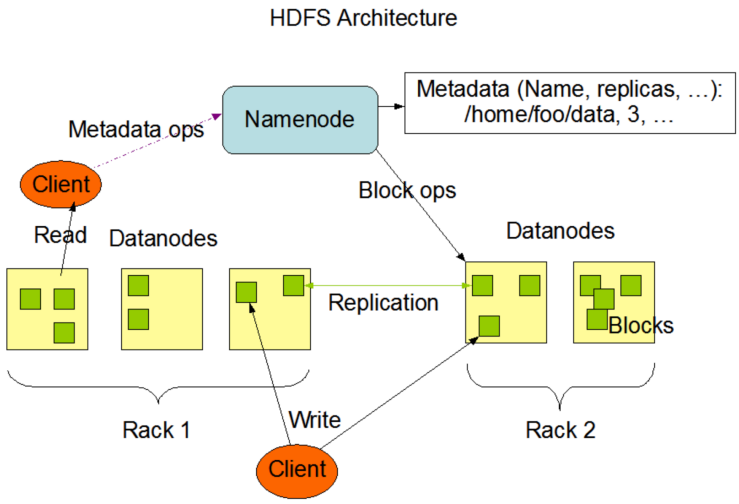


Fig 2: HDFS Architecture (Image taken from: https://hadoop.apache.org/docs/stable/hadoop-project-dist/hadoop-hdfs/HdfsDesign.html)

The NameNodes determine the mapping between blocks and DataNodes whereas DataNodes serve read/write requests from clients. As discussed above, in order to make the infrastructure fault tolerant, DataNodes perform block replication upon instruction from NameNode. All files have multiple replicas in order to help with the parallel processing, as can be seen in Fig 3. An application can specify the replication factor, the number of replicas of a file that should be maintained by the HDFS. That information is stored in the Namenode and by default the replication factor is 3. The block size and replication factors are configurable per file. The replication factor is specified during file creation and can be changed later (appends or truncates).

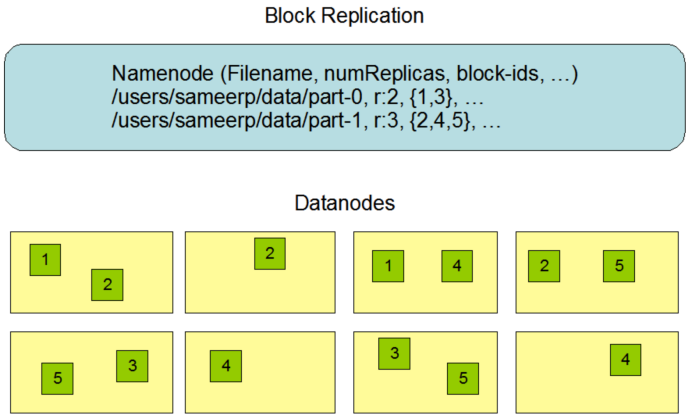


Fig 3: Replication Mechanism (Image taken from: https://hadoop.apache.org/docs/stable/hadoop-project-dist/hadoop-hdfs/HdfsDesign.html)

The placement of replicated blocks is critical to the performance of the architecture. HDFS increases data reliability/availability, bandwidth utilization and minimizes latency by satisfying read requests from the closest replica to the reader. If a cluster spans across multiple data centers, then HDFS prefers local replica access over remote access. Hadoop's computing model, MapReduce, relies on HDFS as the underlying basis for providing data distribution and fault tolerance. The objective of HDFS is to store data reliably and provide fast data access even in presence of failures.

2.7.2 Hadoop MapReduce (Hadoop's Computing Model)

Hadoop MapReduce is a programming paradigm that simplifies writing applications that process large volumes of data by distributing load over several machines and processing in-parallel on commodity hardware in a fault-tolerant manner. All tasks are specified in terms of mapping and reduction functions. The input data-set is read from HDFS, split into independent chunks and then processed by mapper function in a parallel manner[26]. The output from the mapper is then sorted and inputted into the reducer function. This distribution and computation is done parallelly. MapReduce takes advantage of this parallelism by splitting the data load and applying the same function to all the machines, hence keeping the processing steps intact. The architecture takes care of scheduling tasks and re-executing the failed tasks, further ensuring minimum latency. Although the Hadoop framework is written in Java, MapReduce application don't necessarily have to be written in Java. The Hadoop distribution supports a utility called "Hadoop Streaming" that allows users to create and run Map/Reduce jobs with any executable or script as Mapper and/or Reducer. This project uses Hadoop streaming as the Mapper and reducer functions are written in Python.

Architecture:

The MapReduce framework primarily deals in <key, value> pairs within the process flow. It reads the input file and creates intermediate <key, value> pairs which are then combined to create output <key, value> pairs which are all conceivably of different types (Fig 4). This approach brings all the parallelization in the application down to the way the mapper and Reducer functions work and communicate. The Mapper function maps the input records to intermediate key/value pairs. Maps are individual tasks and the numbers of map tasks initialized by the framework depends on the size of input splits for the job i.e. the total number of blocks of input files. In other words, there is one map task per input split generated by the "InputFormat", which describes the input specifications for the MapReduce job. The input and output key/value pairs do not need to be of the same type. The output from the mapper is then put into the Reducer which has three main phases: shuffle, sort and reduce. The shuffle and sort phase occur simultaneously. In these phases, the framework groups the output of the mapper by keys (since different mappers may have output the same key) and fetches the relevant partition of this output. These merged and sorted key/value pairs are then reduced. The reduce method is called for each <key, (list of values)> pair and the output of this task is written into the HDFS.

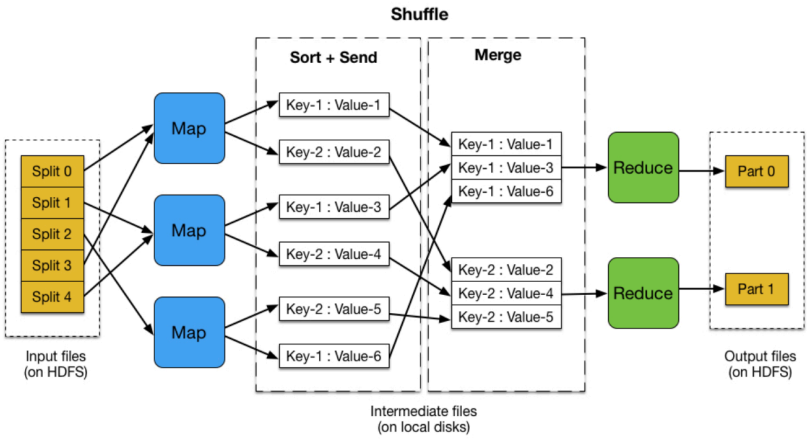


Fig 4: MapReduce Architecture (Image taken from:
http://www.sunlab.org/teaching/cse8803/fall2016/lab/mapreduce-basic/)

The number of reduce tasks launched are either 0.95 or 1.75 multiplied by (<no. of nodes> * <no. of maximum containers per node>). With 0.95 all of the reduces can launch immediately and start transferring map outputs as the maps finish. With 1.75 the faster nodes will finish their first round of reduces and launch a second wave of reduces doing a much better job of load balancing. Increasing the number of reduces increases the framework overhead, but increases load balancing and lowers the cost of failures. The scaling factors above are slightly less than whole numbers to reserve a few reduce slots in the framework for speculative-tasks and failed tasks[26]. The parameters for a MapReduce job are configurable according to the need of the client.

2.7.3 YARN (Yet Another Resource Negotiator)

The basic principle of YARN is to manage resources between applications and schedule/monitor jobs[27]. It schedules resources between all system applications for example, CPU time, memory, disk storage, network etc. It has a global ResourceManager and an ApplicationMaster per application. The ResourceManager is the ultimate authority and is responsible for monitoring/delegation of resources. The ResourceManager and NodeManager form the data-computation framework. The ApplicationMaster negotiates resources for their specific applications, from the ResourceManager and works with the NodeManager to execute and monitor tasks.

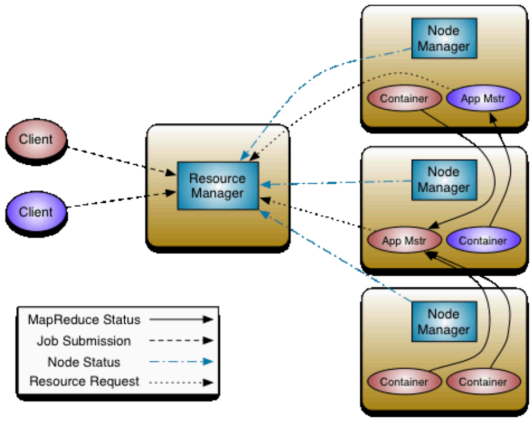


Fig 5: YARN Architecture (Image taken from:
https://hadoop.apache.org/docs/current/hadoop-yarn/hadoop-yarn-site/YARN.html)

The ResourceManager has two main components: Scheduler and ApplicationsManager. The scheduler is responsible for allocating resources to applications but doesn't guarantee restarting any failed tasks. The ApplicationsManager on the other hand accepts job-submissions and negotiations from ApplicationMasters requesting resources for their specific applications. It also provides service to restart nodes on failure.

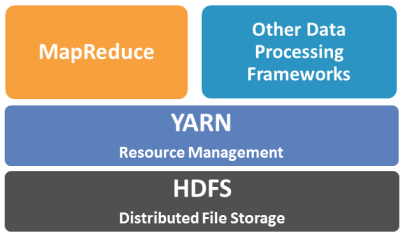


Fig 6: Hadoop's Components

These three components of Hadoop Framework are responsible for the parallelization and scalability in applications. It reduces the latency and delays involved in computation and data access. They provide maximum bandwidth utilization by load balancing and fast node recovery from failures. As can be seen, HDFS is situated at the base of the framework which means that it supports all the applications that are on top of it. YARN works together with HDFS to allocate resources to the computing model, MapReduce. YARN arbitrates resources between the applications that are located on top of this layer. MapReduce, as mentioned before, directly with HDFS and YARN communicates while processing. All these layers combine to maximize throughput and that is the reason why industries have been adopting the Hadoop's distributed systems for various operations.

3. DESIGN SOLUTION AND VALIDATION

The overall project architecture has been shown below and the implementation strategies for each section will be discussed individually.

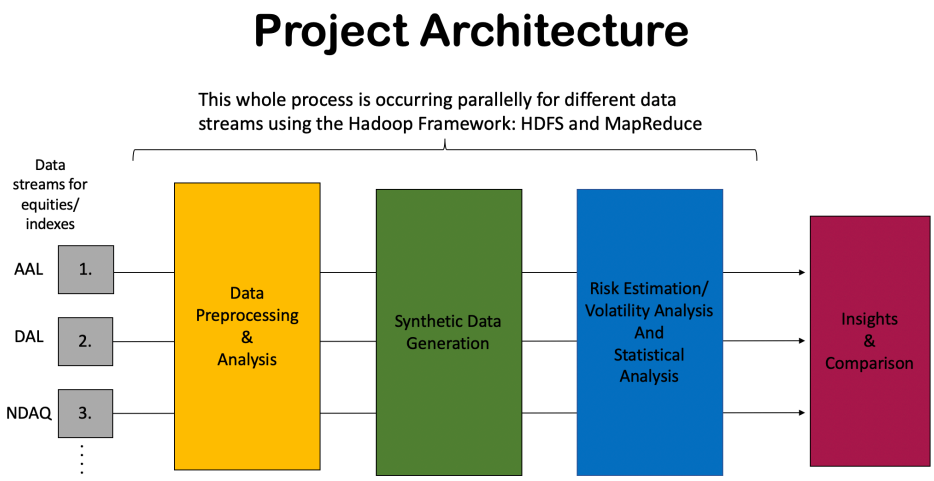


Fig 7: Project Architecture

3.1 DATA

In finance, market data refers to the recorded trade-related information- such as closing price of stock, opening price of stock, maximum price recorded and many more attributes like these, reported by a trading venue such as a stock exchange. Traders and investors are kept up to date with the latest prices as well as the historical trends, with the help of market data for equities, derivatives etc. The main focus of this project is on historic stock data which will read as stock return series. A stock returns series is a time series of price observations (Mostly closing price is used, that is why out of all the columns of data, we choose that to take computations further). A stock, alternatively called "Shares" or "Equity", is a type of security that signifies fractional ownership of the issuing corporation in proportion to the total number of shares[7]. This entitles the stockholder to that fraction of the corporation's assets and earnings[7]. For this project I have taken two American equities: American Airlines (AAL) and Delta Airlines (DAL) and one American index: Nasdaq. A quick side note, while talking about the US Market, the data imported for these equities follow the United States date format: MM/DD/YYYY. A stock market index is the average of all stocks listed in the stock market. Using this data will help in estimating or predicting whether the US market is investible or not. The collected data further aids study of whether the airline industry within the US Market is doing well or not, and estimation of relative risk in investing in one equity as opposed to the other. It can be seen that risks associated with investing in an index are comparatively lower than the risk in investment in an individual

equity. This is because the averaging of an index removes a lot of the fluctuations and extreme value effects, hence making it a safer stake.

There are two types of prices for a security: futures price and spot price. Our collected stock data is spot priced which means that it's the price at which the buyers/sellers value the asset right now. Futures prices on the other hand, refer to the delayed payment and delivery at predetermined future dates and are not for immediate buying/selling[8]. Now that the nature of the data is clearer, let's focus on its frequency. Frequency of market data refers to the intervals at which trading data is recorded. The shorter the interval, the higher the frequency and the more accurate will be the predictions. Daily stock data would be considered low frequency data and is readily available over the internet. Majority of the sources on the internet provide free daily data. As the interval gets shorter, i.e. hourly, half-hourly, 15-min, 5-min and 1 min, the frequency keeps getting higher and so does the complication of finding it. The aforementioned low frequency data is available for the latest trading days going back to at least 20 years. On the other hand, as the frequency keeps getting higher, the size of the data keeps increasing in terms of storage as well as the number of data points as a result of which, high frequency data is very hard to find over the internet. Even if there are financial data vendors online, that provide high frequency data, they either charge exorbitant amounts of money for it and enroll you in a subscription-based model or they provide the high frequency data for shorter lengths in time for example, the last 3 days or a week (maximum). The frequency of the data used in this project is intraday 1-minute data. This is the highest frequency that is possibly available online via official market data vendors or private data providers. After two days' worth of research went into finding a website that would provide intraday minutely data for free. The data requirement was for enough trading days to at least see a pattern or to simulate more data from it. This was another reason why it was more difficult to find such a vendor. I finally came across iexcloud.io which provided the last 3-months' worth of intraday minutely data for various equities. This website supplied almost complete data except for a few null values. The reason for these missing values is that, there are times when no minutely transactions are recorded which reflects as blanks for that timestamp in the data. It is also observed that the number of said missing values or blanks were much less in the index data than in separate equities. Again, as the index is an average, the chances for such anomalies is small.

My research indicates that there are three important considerations: authentic source of data; availability of high frequency data on the source, data extraction method, and cost. Preferably the cost has to be zero (see Table 2 below) of finding the right dataset for this project is concluded in the table below:

Website/Resource	High Frequency data available	Method Used	Free	Comments
finance.yahoo.com	No	Direct Download	Yes	Low frequency data available
kaggle.com	No	Direct Download	Yes	Only Low frequency daily data available
quandl.com	Yes	API Call	No	Subscription required for high frequency datasets & with limited options
algoseek.com	Yes	On order	No	Tick Data available for US Markets only & there is no data sample preview for what is being ordered so if you're new to this, you might pay for the undesired dataset
intrinio.com	Yes	API Call/Direct Download	No	Subscription based model for US Markets only & no specific sample preview unless subscribed
tickdata.com	Yes	On Order	No	Not free & requires a lot of personal details
firstratedata.com	Yes	Direct Download	No	Sample data available. 1 Minute data available for 1 month & tick data available for last 3 months. Its available at a cheaper price than the rest.
alphavantage.co	Yes	API Call	Yes	Intraday 1-minute data available but for the most recent 3 trading days
iexcloud.io	**Yes**	**API Call**	**Yes**	**Intraday 1-minute data available for the latest 3 months**

Table 2: List of websites for high-frequency data

3.2 DATA PREPROCESSING AND ANALYSIS

All data preprocessing techniques follow some general steps that are imperative for successful analysis. The steps followed here are: Studying the Data, Data Cleaning and Exploratory Analysis. As mentioned before, this project works with three stock market datasets but for these steps, I will show processing of one equity: AAL and the rest will be alike.

3.2.1 **Studying the Data:** In this step, we try to get to know the data better. We need a clear understanding of what we're working with and spot any errors or miscellaneous behavior that might be present in the distribution. As mentioned above, the data collected for this project, has some missing values. This is easily discernible by plotting the data and identifying inconsistencies

within it. Even curated datasets provided on official government sites contain errors in them, so it won't be wrong to assume that our dataset would too. However, these errors need to be fixed before further investment of time and effort into analysis has been put. This will be seen in the next steps but for now, following results of the "dirty" data shows the discontinuity because of missing values:

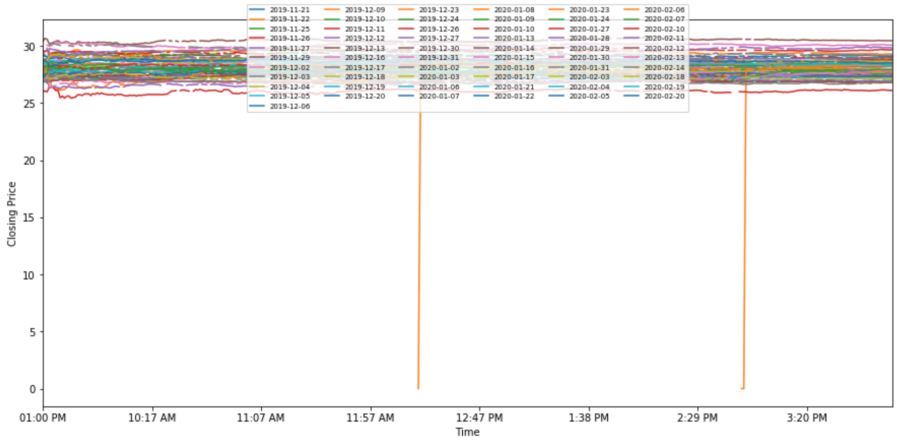


Fig 8: Time vs. Closing Price for all days

This visualization in Fig 8 looks rather chaotic and unintelligible because of various colors, legend and the scale. Despite of the grubby look, the gaps don't go unnoticed. These gaps suggest that our data contains null values that need to be dealt with. The reader must also observe the near constant value of closing price for different days and within that, the movement of prices as opposed each other, especially at the beginning. Except the two substantial drops that can be treated as outliers, the prices seem to be highly correlated. These prices move in a similar direction at every time of the day, which can be attributed as correlation between them. Consecutive changes in prices have very little/weak correlation as opposed to the prices itself[32]. Since returns are the preferred way of measuring price changes, it is more convenient to investigate returns. We'll see the behavior of returns later. A clearer image with proper scaling is shown in Fig 9. The above explanation can be seen in Fig 9 as well. The gaps might not be clearly seen here through time, but the relative ease to recognize pattern has increased.

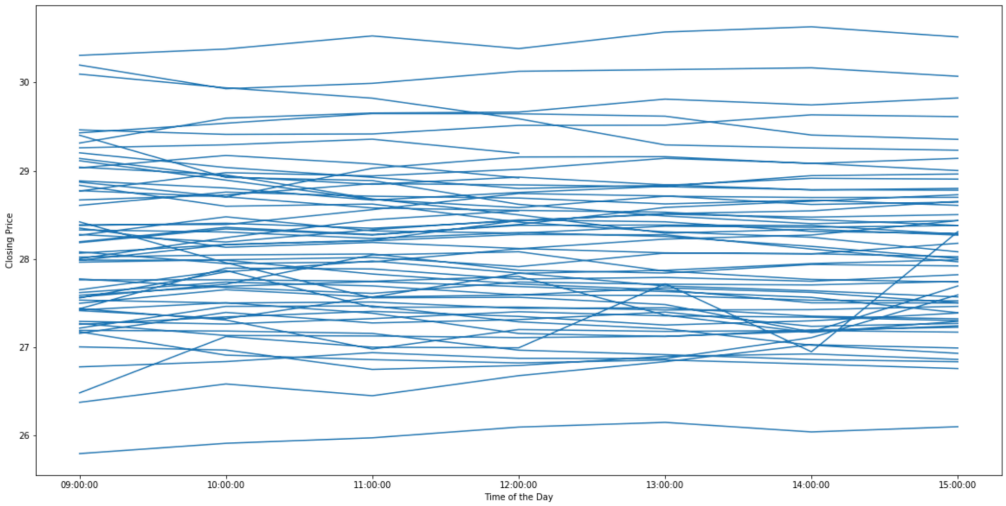


Fig 9: Time vs. Closing Price for all days (without legend)

Another representation that offers a different perspective is when the closing prices are plotted against different dates (Fig 10). Any Calendar Effects in the distribution, will be evident in this plot. For our data, there is no visible calendar effect as the plot seems rather random. If seen closely enough, a fortnightly pattern can be noticed, where there is a negative and positive trend every consecutive fortnight respectively. Although the calculations and objective of this project are not related to the analysis of these effects, it needs to be considered while studying the data. This plot however, shows the gaps distinctly.

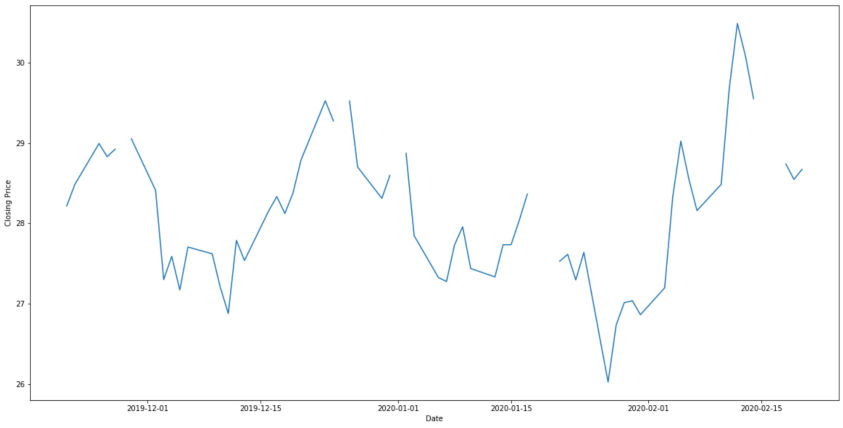


Fig 10: Date vs. Closing Price

Talking about calendar effects, Fig 11 shows the variation in closing prices for each day. The data collected for this project is for 3 months (21.11.2019 to 21.2.2020). Fig 11 shows just some of the days to give an idea of the effects and missing values. However, as a side note I want to mention that there are some missing dates present in data which are attributed to weekends and holidays on which, trading is closed. As can be seen in each case, evolution of prices during the trading day is not uniform for any weekday. Generally, day-

of-the-week effects and intraday effects can be evaluated through daily close-to-close returns. However, the negative Monday effect can be seen in intraday prices to a small extent[33]. On Monday mornings, prices tend to drop while on other weekday mornings, they rise. This can be attributed to the abnormally high Friday closing prices. This can be observed in the first row of Fig 11 in which 11/22/19 (22nd November 2019 in US date format) is a Friday and it shows a price rise at the end of the day. The next trading day, 11/25/19 (Monday), just beside it, exhibits a price drop early in the day. The reason for this behavior is highly debated where some say it might just be the result of systematic errors in data or deliberate price manipulation[33]. Another striking characteristic that is commonly found in datasets is the rise in prices on the last trade of the day. This can be observed in almost all the days in Fig 11. Other calendar effects can be observed for this dataset as well, however they are comparatively more subtle that intraday effects. Pre-Holiday effects can be seen on 12/30/19, when the price drops towards the end of the day. This is because the traders start selling the stock before the holiday and record a lower market close. Other than the above-mentioned features, prices patterns are usually similar for all weekdays. Nonetheless, calendar anomalies are puzzling and have become less pronounced over recent years. They also have little impact on correlations of returns and hence the presented plots are just a way of studying the prices before moving on to returns.

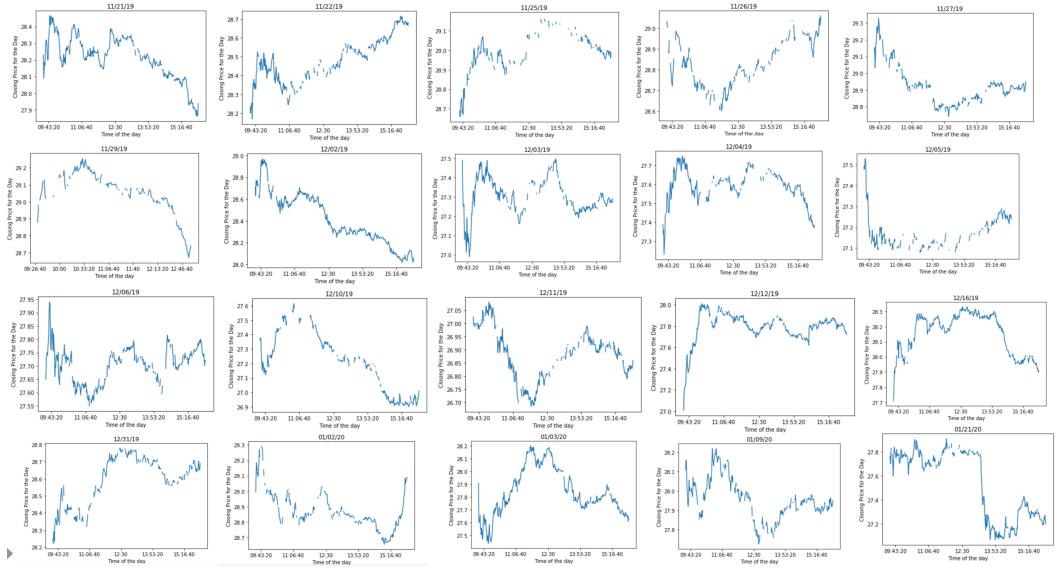


Fig 11: Time vs. Closing Price for all days shown separately

3.2.2 Data Cleaning: As has been stressed in the previous step, our collected data contains missing values which need to be fixed before moving further with analysis. In this step, we look at different techniques that can be applied and the approach we take in this project. There are two broad approaches while dealing with missing data; which is to either drop those values/rows or to replace them. In our case, as in most cases, it is not suggested to drop the rows corresponding to the missing values of closing price because that would mean dropping or deleting the minute from

our "Time Series". Hence, it seems obvious to have a continuous and complete timestamped record. Now that we have eliminated the option of removing values, we're left with replacing it. We can't replace null values with zero. This is because we are analyzing a time series of stock returns and as defined earlier, returns are the logarithmic change in closing prices which means that replacement with zero will result in "Division by zero" error while calculating returns. Hence the only practical option is to use logical or statistical reasoning for replacement of these values. I have used deductive imputation to deal with missing data. Deductive Imputation is a rather logical approach which is based upon an understanding about the relationship between variables and units to fill in missing values. Examples include deriving a value as a function of other values, adopting a value from a related unit, and adopting a value from an earlier time point. In a stock series, if there are no transactions recorded for a the minute that means the volume of trades recorded are zero. For this reason, it's safe to say that the closing price for this minute hasn't changed from its previous value. This means that the missing values of closing prices can be replaced with the prices of the previous minute. This is the approach taken for cleaning the data. Other approaches consist of Model based Imputation which can either use Mean/Median to estimate missing values using assumptions about the distribution of data or using assumptions about the relationships of variables and predict missing values. The latter wouldn't be appropriate for stock series as the pattern/trends are unknown and regressive methods would not be accurate enough. However, the Mean/Median imputation could work if the mean of the value immediately above and immediately below are taken and put in place of the missing value. Nevertheless, when there are two consecutive missing values, using this method might increase the bias in our calculations, hence taking us away from accuracy. Due to the above mentioned explanations, deductive imputation seemed most suitable.

3.2.3 **Exploratory analysis:** For the final step in terms of data preprocessing, let's explore what we're working with. This is different from the first step not only in terms the data but also in terms of its completeness. The data used in this analysis has been cleaned and the missing values have been fixed. The returns have been calculated for the "complete/clean" set of closing prices and will be used for every ensuing computation/process. We hereby, close the discussion with closing prices and talk about how the returns look like. The results in Fig 12 show the intraday returns for all the trading days in the 3-month period (21.11.2019 to 21.2.2020). The reader must observe the legend, which contains some missing dates, which can be attributed to weekends and holidays as mentioned in the first step. Again, this plot looks extremely chaotic but the randomness in the data hardly goes unnoticed. This arbitrary distribution of each date as opposed to the other, can be considered a feature for the weak correlation seen in returns for different days.

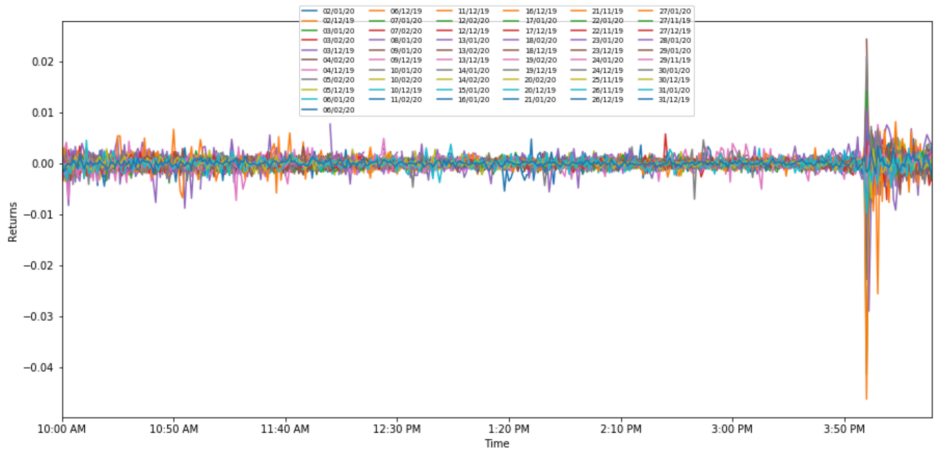


Fig 12: Time vs. Closing Price for all days (fixed data)

Notice how there are no gaps in the distribution. Fig 13 gives a scaled output for the above image. General behavior of market returns can be deduced from Fig 12 and Fig 13. It can be seen from the variation in returns that the mean of the distribution is zero or nearing zero. Except for a few outliers in the beginning that show large positive and negative deviations, the returns fluctuate between positive and negative such that each positive(negative) value is eventually followed by a negative(positive) value which when averaged out, give a zero value shows stable market behavior. The extreme changes in the beginning can be attributed to trader sentiments when market opens. This plot also shows weak to no correlation between each other as at time they show completely opposite movements and at others, somewhat similar variations. This altogether gives an uncanny randomness to their directions of movement. As mentioned before, there are certain properties that are followed by most set of stock returns. We have seen one of them: there is no correlation between returns for different days. As we progress further, other properties will be seen in our results as well.

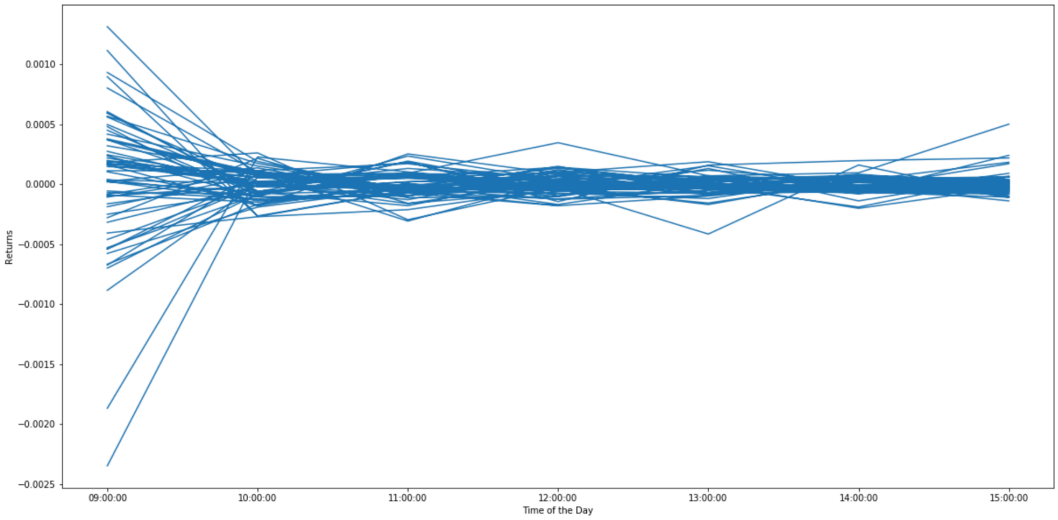


Fig 13: Time vs. Closing Price for all days (fixed data, without legend)

For additional clarity, the following images show the Time Vs. Returns for separate days. Again, not all the days/dates will be shown for the sake of conciseness. It is impractical to show the plot for 90 days, although the calendar effects will be evident and addressed.

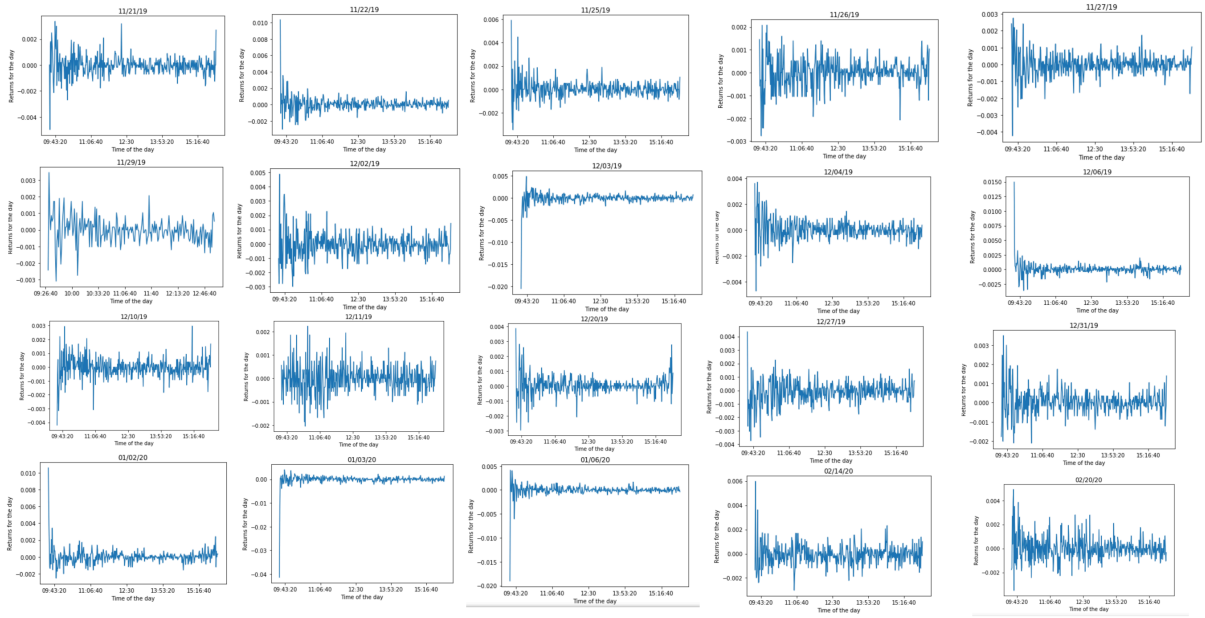


Fig 14: Time vs. Closing Price for all days shown separately (fixed data)

In Fig 14, the reader must notice that there are no gaps in the plots, consequently implying the absence of missing values in the dataset. Since we replaced the missing closing price data in the previous step with appropriate values, these changes have been reflected on returns as well. The approach adopted in the previous step, where the missing value at time *t* is replaced with the value of the closing price at *t-1*, will give us a zero return for time *t-1*. This has been calculated as $ln(1)$ is zero. Therefore, if our technique or calculations would have been erroneous, Fig 14 would have seen some anomalous behavior, which isn't the case here. Notice how zero values are present but, due to the frequency (1-minute) of data, plots look continuous and consistent. Let's discuss another property that can be seen in most stock series: positive correlation between magnitudes of returns of nearby days. As can be noticed in Fig 14 first row, the returns for the nearby days 26.11.19 and 26.11.19 seem to be moving in similar directions during the day. This hints some positive correlation between the two days, whereas for the days 25.11.19 and 26.11.19, there seems to be a striking difference in the movement of returns during the first half of the day. These dates have been chosen in way that they lie in the same week in order to avoid any week-of-the-month effect during comparison. Most of the pairs of nearby days follow the same principle except the ones which are separated by weekends or holidays. Returns can better describe the intraday effects that were discussed previously with closing prices. Some systematic return patterns can be identified but studies haven't yet been able to determine infallible causes for these recurring patterns. Negative Monday effect is observed in close-to-close returns, which often results in low or negative average return from Friday to Monday which then depresses the stock prices on Monday. This results in the returns on Mondays being lesser than

other weekdays. This can partly be due to the high closing prices on Friday[33]. This effect can be seen clearly in Fig 14 last row, 01/06/20(Monday). There are negative returns early in the day and positive returns later. These effects however are different for different market value groups. For larger firms they are accrued between Friday close and Monday open and for small firms they happen during the Monday trading day. There are significant day-of-the-week differences within the first 45 minutes of trading that can be identified in these plots. On Mondays, returns are negative, while on the other weekdays, returns in this interval are positive[33]. It needs to be stressed upon though, that these effects are weak and to consider these effects while developing trading strategies is still debatable. With that said, general strategies for individual traders can be to avoid transacting early on Monday if purchasing, and avoid early transactions on Tuesday through Friday if selling.

3.3 SYNTHETIC DATA GENERATION

The high frequency data collected for the 3-month duration, contains up to 23,000 data points per equity/index which equals 69,000 points in total which is not enough data to be called "Big Data". In order to generate more data, a statistical approach has been taken in this project. This approach is based on the numerical practice of generating pseudo-random numbers that are distributed according to a given probability distribution. Historically, methods for pseudo-random number sampling were developed for Monte-Carlo simulations that are used in forecasting models to predict uncertainty. The intention of this project is not to build computational models for prediction; thus, we consider the sampling method. The distribution of stock returns is continuous, as is for any time series. Pseudo-random number sampling for a continuous distribution encompasses many general methods for generating independent samples. We apply "Inverse Transform Sampling" as it generates sample numbers at random from any probability distribution, given its cumulative distribution function (CDF). We know that a stock return series has a probability distribution which can be used to generate a CDF. In statistics, probability distribution is a function that provides the probability of occurrences of different possible outcomes and cumulative distribution of a random variable X evaluated at x, gives the probability that X will take a value less than or equal to x. To implement Inverse Transform Sampling for stock returns, we first need to calculate the probability of occurrence for each return, in the distribution. I achieved this by creating a histogram and extracting the Bin-Frequency values for the distribution. Since the Bin-Frequency values represent the occurrence probability of intervals within bin range, we can use this information to plot the probability distribution function (Fig 15).

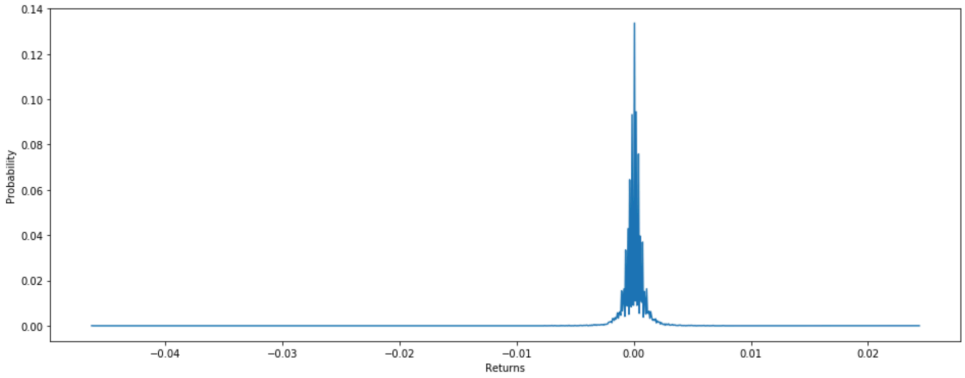


Fig 15: Probability Distribution of the collected dataset

The probability distribution given in Fig 15, is evidently not normal which can not only be categorized by the density congregation towards the mean but will also be differentiated by the values of kurtosis and skewness later in the text. This however, satisfies the last of the mentioned general properties of stock return series (non-normal distribution). The cumulative distribution function can now be calculated as the area under the probability distribution function (PDF), for our continuous series. Mathematically, it will just be an integral of the PDF, but to simplify execution, I chose to calculate the cumulative probability for each bin interval from the minimum range to maximum range, and plot these cumulative probabilities against the bins. The simplicity in this approach is that the need to compute the function for the distribution has been reduced. If the 'area under the curve' method would have been adopted, the PDF would be calculated first and then the integral be would be evaluated. Hence, the current approach saves us the time of calculating the PDF which isn't going to be used in further calculations anyway. The plotted cumulative distribution function is given in Fig 16.

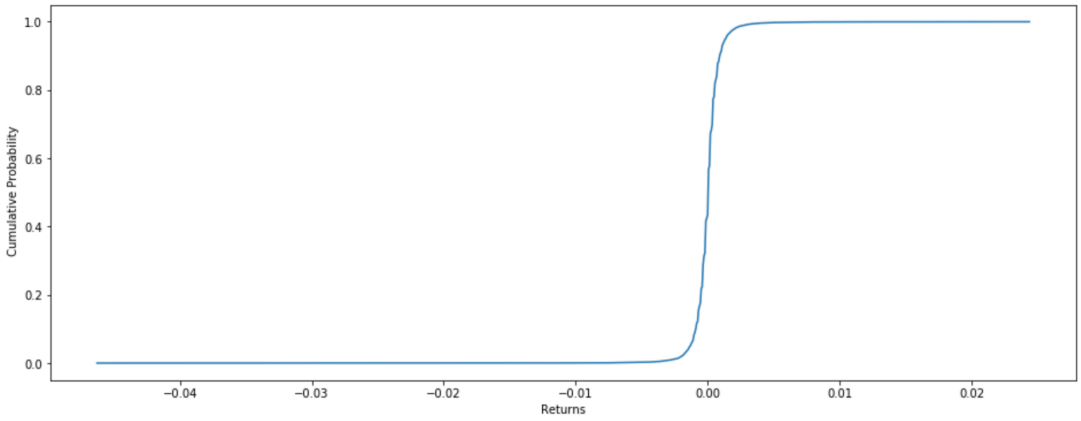


Fig 16: Cumulative distribution of collected dataset

As can be confirmed by Fig 16, the cumulative probability ranges from 0 to 1 which verifies the correctness of our calculations so far. The sampling process takes uniform random samples u, between 0 and 1, interprets them as probability, and then returns the largest number x from the domain of the distribution

P(X) such that P($-\infty < X < x$) $\leq u$. In other words, after generating the random number u, we find the inverse of the CDF $F_x(x) = F_x^{-1}(x)$, and then compute $X = F_x^{-1}(u)$. We now have a generated synthetic value of X, the random sample, whose distribution is described by the CDF $F_x(x)$.

The way this process has been approached here (per equity) is:

1) Generate 100,000 random samples between 0 and 1 (Probability).
2) Using the curve-fitting optimization functionality of SciPy, the scientific computing module of Python, estimate the coefficients to provide the best fit curve.
3) Note: To use the curve_fit function, a model or reference function should be known, in order the fit the data according to it and use the covariance between them, as an argument. 'Sigmoid' has been the use case here.
4) Using the calculated parameters and arguments, create an inverse of the CDF.
5) Put the generated random samples in the inverse CDF and synthesize new data points that follow the original distribution (Reference in Fig 17).

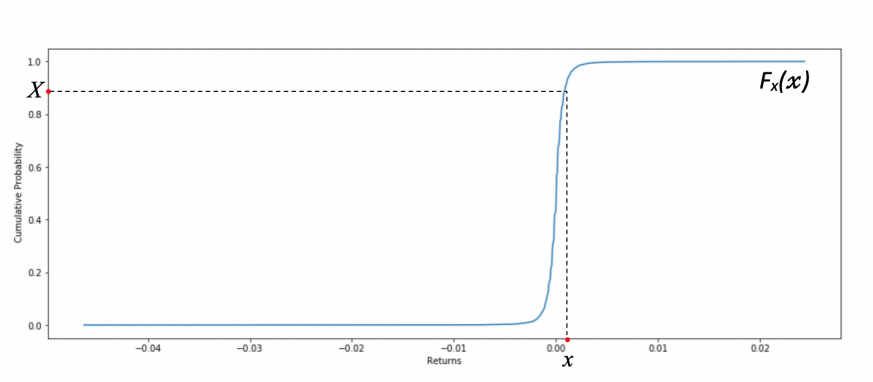


Fig 17: Inverse Transform Sampling Process

The newly synthesized data overlaps onto the cumulative distribution function as seen in Fig 18 below. This validates "pseudo"-random sampling, as the process starts with random numbers but ends up generating data within the distribution itself. We now have 123,000 datapoints per equity which sums up to a total of 359,000 datapoints which is worth 11 MB of data in storage. We now have some "Big Data" to implement and apply with latency critical frameworks.

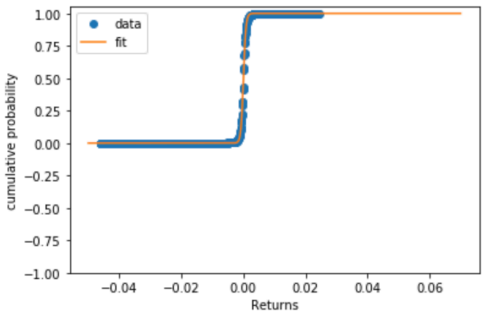


Fig 18: Fitted Plot of synthesized data with original data

3.4 RISK ESTIMATION AND VOLATILITY ANALYSIS

This section concludes the design implementation for the project and analyses the execution process. It describes the application and performance of the Hadoop Architecture for the collected data. It executes parallel processing and distributed storage to provide low latency solutions. We first discuss the implementation of the Mapper( ) and Reducer( ) functions and then tie the results together with the output of these processes. This project uses python for programming, therefore, the Hadoop streaming utility is needed, to offer the same functionalities it would to a Java developer while working with Hadoop MapReduce. System's standard input and standard output are used for input/output(i/o) operations. This i/o streams are also what bind the Mapper( ) and Reducer( ) functions together. This information will be required to understand the output from the two functions. To begin processing, we first need to put our files in HDFS, which will then be picked as input for the MapReduce jobs. After synthetic data generation, the original return files are appended with the newly sampled return data. In order to load the files onto the distributed file system (dfs), we begin with initializing and starting the Hadoop cluster, which is followed by running the YARN manager instances and configuring directories on the dfs. Fig 19 shows the loaded files in the cluster.

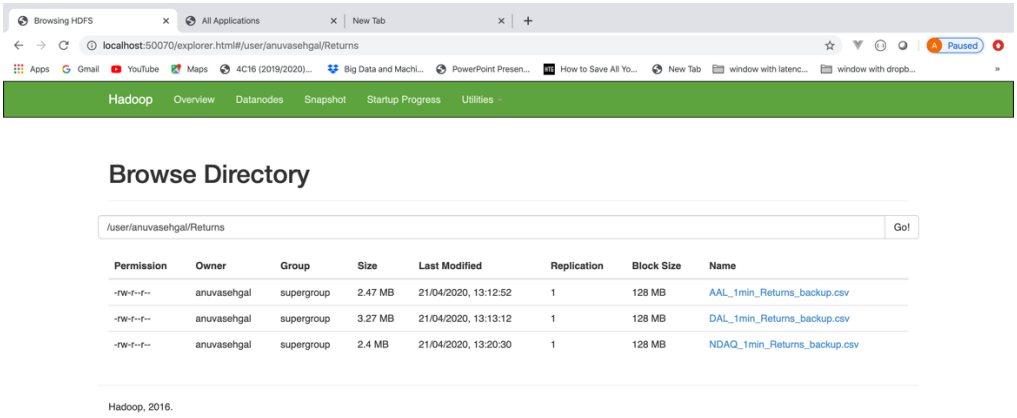


Fig 19: HDFS Directory containing uploaded files

Fig 19 displays the various features of HDFS, discussed in the previous sections. As can be seen in Fig 19, the block size is 128 MB and since the files are much smaller than that, we wouldn't require more than 1 block each, to store the file. The reader must have noticed that the replication factor has been set to 1 in this cluster. To address this detail, we need to know that the Hadoop cluster can be started in three modes:

i) Local (Standalone) Mode
ii) Pseudo-Distributed Mode
iii) Fully-Distributed Mode

It is fairly obvious that the Fully-Distributed Mode runs on multiple systems where the NameNode and DataNodes are running on different machines. This mode has not been used in this project so we now discuss the first two. The Local (Standalone) Mode runs on a single node but is mainly used for debugging where the HDFS is not really used. This mode is the default mode to run Hadoop and it uses the local file system for input/output. It is can be considered the fastest Hadoop mode as it interacts with the file system locally, but since we are trying to reduce latency for realistic industrial challenges faced in daily operations, we need an implementation that is closer to reality and which applies the HDFS structure to make use of available advantages. Due to this reason, this project adopts the Pseudo-Distributed Mode as it's mode of implementation. The pseudo-distributed mode uses HDFS and creates a cluster simulation in which multiple DataNodes and TaskTrackers can run on a single machine. This simulation will also allow developers to test how the system will behave in a fully-distributed mode. Due to use of a single machine, the replication factor in Fig 19 is 1. Otherwise, in a fully-distributed mode it would have either been 3 (default) or some user defined value.

Fig 20 and Fig 21 present summary metrics and Node information for the running Hadoop cluster. Statistics regarding cluster storage, capacity and activity confirm our mode selection and other storage related considerations.

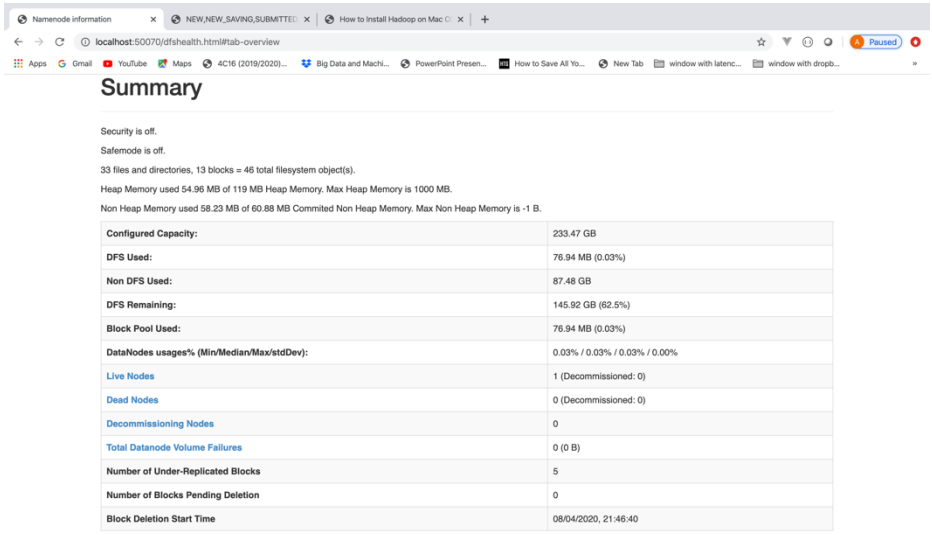


Fig 20: Summary metrics for HDFS Nodes

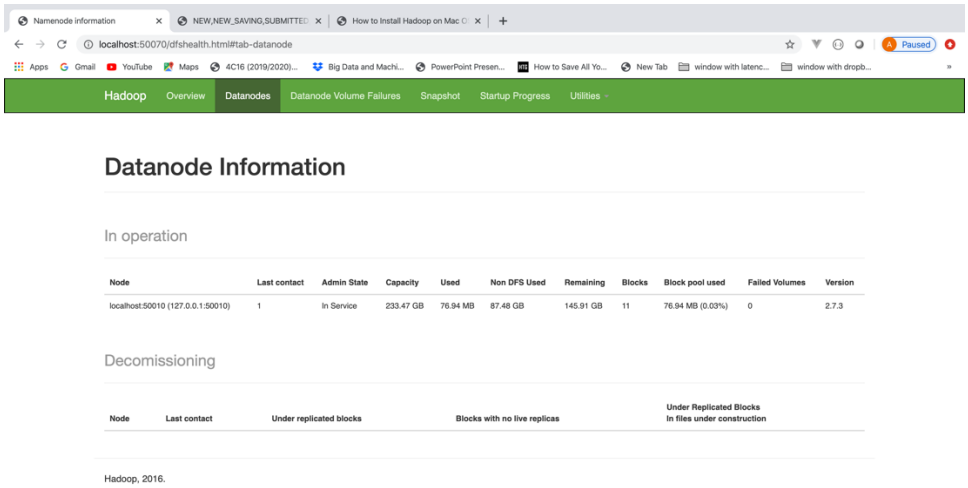


Fig 21: Summary metrics for DataNode

After the files are stored on the HDFS, which is done in almost 'real-time', they are ready to be loaded in the Hadoop MapReduce Application. They are first fed into the Mapper( ) function as inputs. This is done by the system's standard input operation which reads the data stream from the file and sends it as an input stream to the Mapper( ). The Mapper( ) function is programmed to read the standard input (sys.stdin) line by line as the file is in a csv format. The first row of each csv file contains the name of the respective equity which is followed by a row of header fields (shown in Fig 22). We store the equity name in a variable and thereafter read the returns by stripping and splitting the comma-separated lines.

	A	B	C
1	AAL		
2	date	Time	Returns
3	21/11/19	9:30 AM	0
4	21/11/19	9:31 AM	-0.0049716
5	21/11/19	9:32 AM	0.0016007
6	21/11/19	9:33 AM	0.001811
7	21/11/19	9:34 AM	-0.0005678
8	21/11/19	9:35 AM	0.0024818
9	21/11/19	9:36 AM	0.001769
10	21/11/19	9:37 AM	0.0010599
11	21/11/19	9:38 AM	0
12	21/11/19	9:39 AM	-0.0014134
13	21/11/19	9:40 AM	-0.0021239
14	21/11/19	9:41 AM	-0.000709

Fig 22: CSV files containing stock series data

Each Mapper task launches an executable on initialization which converts each line into intermediate key-value pairs of the form: <Equity Name, Return>, for each minute. The read operation, conversion and collection of key/value pairs, all happens as parallelly. The choice of key is driven by the fact that there are going to be 3 equities (AAL, DAL and NDAQ) being processed in parallel, whose returns need to be differentiated from each other. After creating a list of tuples with the key/value pairs, which is collected as the output (sys.stdout) of the mapper, we forward them into the Reducer( ) function as input.

The Mapper function doesn't have much programming complexity and it can be visualized as a pipeline to format the input for the next phase. The majority of processing is done in the Reducer phase where the shuffled and sorted intermediate key/value pairs are inputted. The reducer function uses python packages that need to be installed and integrated with Hadoop streaming. Remember that Hadoop is originally written in java and it only supports java classes/commands without external package sourcing/import. Virtualenv is a tool that creates an isolated python environment which has its own installation directories. This environment doesn't share its libraries with any another virtual environment and doesn't access globally installed libraries either (optional). Thus, a virtual environment with all the python libraries used in the Reducer phase, is created and stored in the current Hadoop directory. Before running the MapReduce command on the terminal, which launches the Map and reduce tasks serially, this virtual environment is activated and Hadoop streaming is executed within this custom python environment.

Similar to the Mapper, reducer tasks launch separate executables on initialization and convert the inputted key/value pairs from the Mapper, into lines for standard input (sys.stdin). The input is then read line by line as strings, out of which returns (from <Equity Name, Return> pair) are converted to floating point numbers. The returns for each equity are stored in a list in order to perform further calculations for results. It is important to mention here, that the reduce tasks running in parallel have the same variable names but the scope for those variables is restricted to individual processes. Returns for each minute, are then used to calculate moments of distribution and predict whether the stock should be bought or sold in the next minute. The program for these outputs is written using the formulae, properties and definitions described in the literature review. In the meantime, the reducer collects the line oriented outputs of the process and displays it using standard output (sys.stdout).
The reducer function may contain all the functional complexities but the mapper function plays a significant role in achieving this parallelism as well. Simultaneous production of <key, value> pairs that are further sent off to be sorted, shuffled, reduced and then merged, helps in achieving greater flexibility and lower latency.

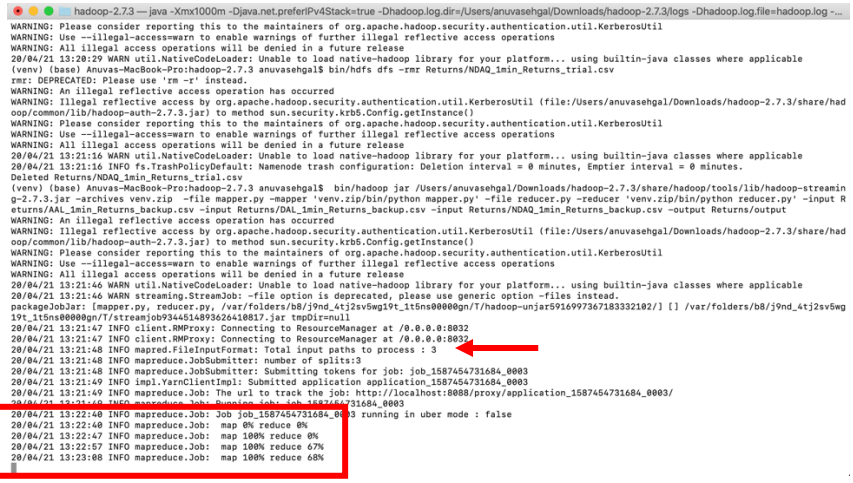


Fig 23: MapReduce running output

Fig 23 shows how MapReduce jobs split the inputs to achieve parallelism. It also describes the execution and progress of map and reduce jobs sequentially. The difference in complexity of map and reduce tasks are reflected in the processing times. Resources for MapReduce jobs can be reviewed via YARN. The resource manager and node manager instances can be reviewed in Fig 24 and Fig 25. Metrics and information regarding the submitted applications are displayed in this pane. As can be seen in the figure, the submitted application is currently running on one active node and uses 3GB of memory. These metrics are consistent with the foregoing discussions.

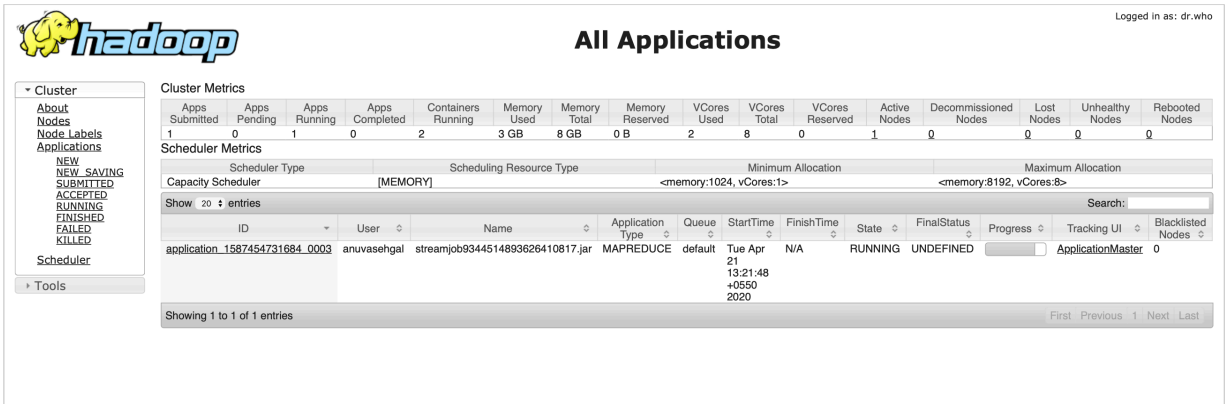


Fig 24: YARN Application status

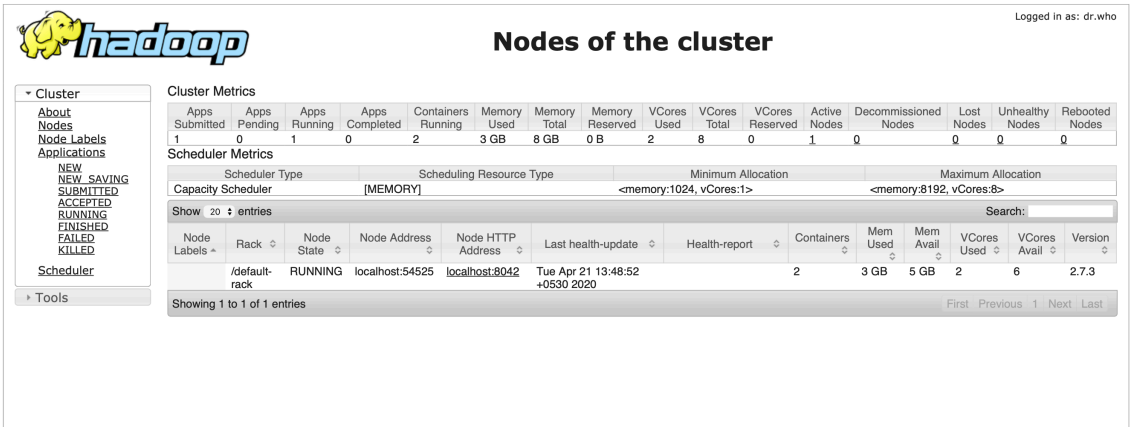


Fig 25: YARN cluster metrics

The storage and status of submitted applications can be evaluated through these daemons as well. Fig 26 shows storage metrics managed by YARN. Since the submitted MapReduce job is the only application

running, there is no queue for resource allocation. Usually, YARN handles an influx of applications, which create a queue. These queues are used to manage resource allotment according priority, processing time, execution time etc.

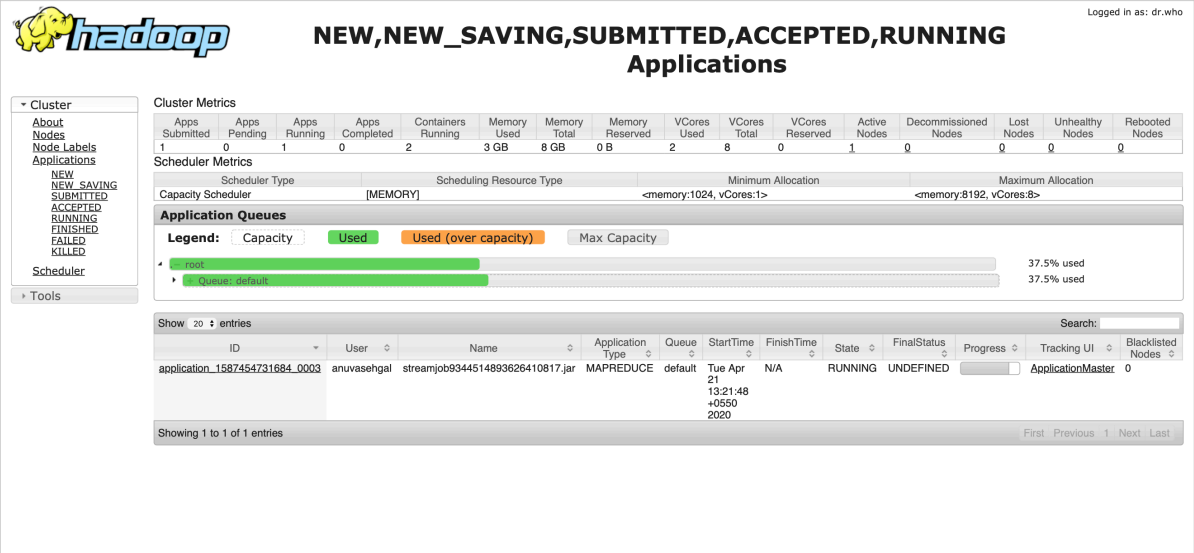


Fig 26: YARN storage metrics

This concludes implementation of the Hadoop Framework in this project. This simulation provides a good look into how the Hadoop architecture achieves low latency data processing using MapReduce, and decreases input/output overhead. It also interprets the challenge of massive data storage and organization by replicating the implementation of HDFS on a smaller scale. It improves data load performance and increases I/O bandwidth, effectively extracting valuable information which is the main purpose of financial big data analytics.

4. RESULTS AND CASE STUDIES

This project aims to see the effect of data variation on the stock return distribution and corresponding risk estimates. Furthermore, it attempts to interpret latency critical issues by multi-tasking, with the help of the common the Big Data technology: Hadoop Framework. Therefore, the results will be divided into 2 sections:

i) Risk estimation and data manipulation

ii) Latency Reduction using parallel processing

4.1 RISK ESTIMATION AND DATA MANIPULATION

Risks in financial stock series can be evaluated using multiple approaches. One might use machine learning models for forecasting or adopt analytical approaches to study market variations. This project integrates the methodical techniques with Hadoop, the big data technology, in an attempt to create a forecasting algorithm. Risks govern decisions of whether a trader should Buy or Sell a stock. Here, we estimate the Buy/Sell indicators along with the statistical properties of the distribution. The moments of distribution go through significant variations in values, when data is manipulated. Moments of distribution have a direct impact on market risks and changes in them reflect so, accordingly. The chronological order for our evaluation process is shown in Fig 27 for a better understanding.

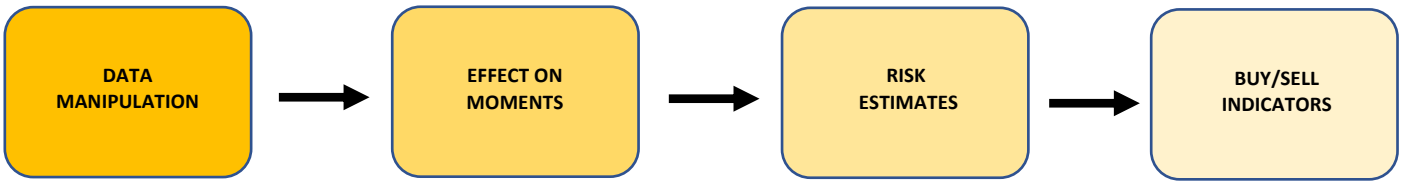


Fig 27: Step-by-Step Evaluation process

Data manipulation will be done in two ways: Rate of sampling data and Data smoothing. The rate at which we sample data doesn't denote the velocity of data. This is because the frequency at which data is collected remains 1-minute, keeping the velocity unchanged, it's just the interval at which data is being read and processed. In other words, if the data is read in (or sampled) every minute, the sampling rate is equal to the velocity but otherwise, if the sampling rate is N-minutes, it means that data is going to be read in every N minutes, making the data read in at the $\alpha * N^{th}$ minute (where α = 1,2,3....), the average of the last N unread minutes. For example, if the data is sampled every 4 minutes, the first data point will be recorded at 9:33 AM (assuming market starts at 9:29 AM, α = 1) and it will be the average of returns from 9:29 AM

to 9:33 AM. Similarly, the second datapoint will be recorded at 9:37 AM (α = 2) and it will be the average of returns from 9:34 AM to 9:37 AM (4 minutes). This goes on until the last trade of the day at 3:59 PM. Sampling rates used for assessment are:

- 1-minute (2^0) data
- 2-minute (2^1) data
- 4-minute (2^2) data
- 8-minute (2^3) data
- 16-minute (2^4) data
- 31-minute (2^5) data
- 64-minute (2^6) data
- 127-minute (26) data
- 256-minute (27) data

Data Smoothing is a technique that removes outliers from the distribution in order to make patterns more visible. It averages the data over an interval to reduce the effect of extreme values and decrease fluctuations. It eliminates noise from the dataset and can help in predicting trends in security prices. The data smoothing method used in this project is the moving average method. It places equal weight to both recent and historical prices. The moving average method uses an *M*-point window, which calculates the arithmetic mean of returns in the *M*-datapoint interval. It then slides down the fixed *M*-point window by one datapoint and aggregates the returns. For example, a 10-point moving average will have a fixed window of 10 datapoints, starting at time *t*. It calculates the average of returns in that window, from *t* to *t+10* and outputs the mean return as the newly calculates return at time *t*. The window now slides down, starting at *t+1,* calculates the average returns from *t+1* to *t+11* and records the calculation as the new averaged return at time *t+1*. This goes on for the whole dataset. For this project we use a 100-point moving average, as it's a common selected value for *M* (Fig 28).

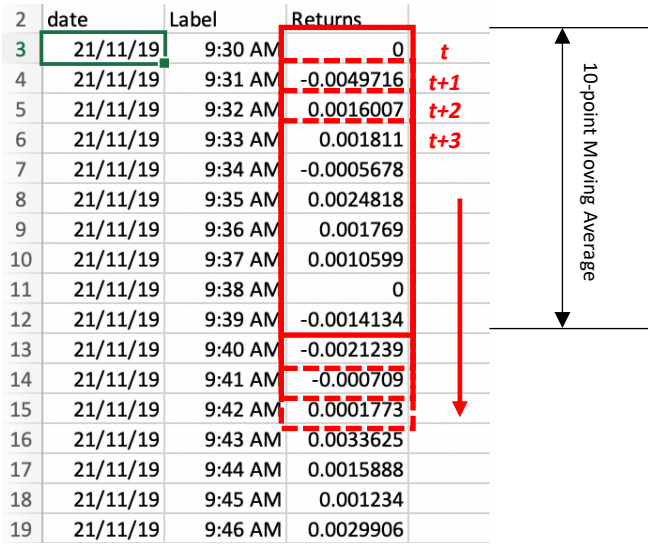
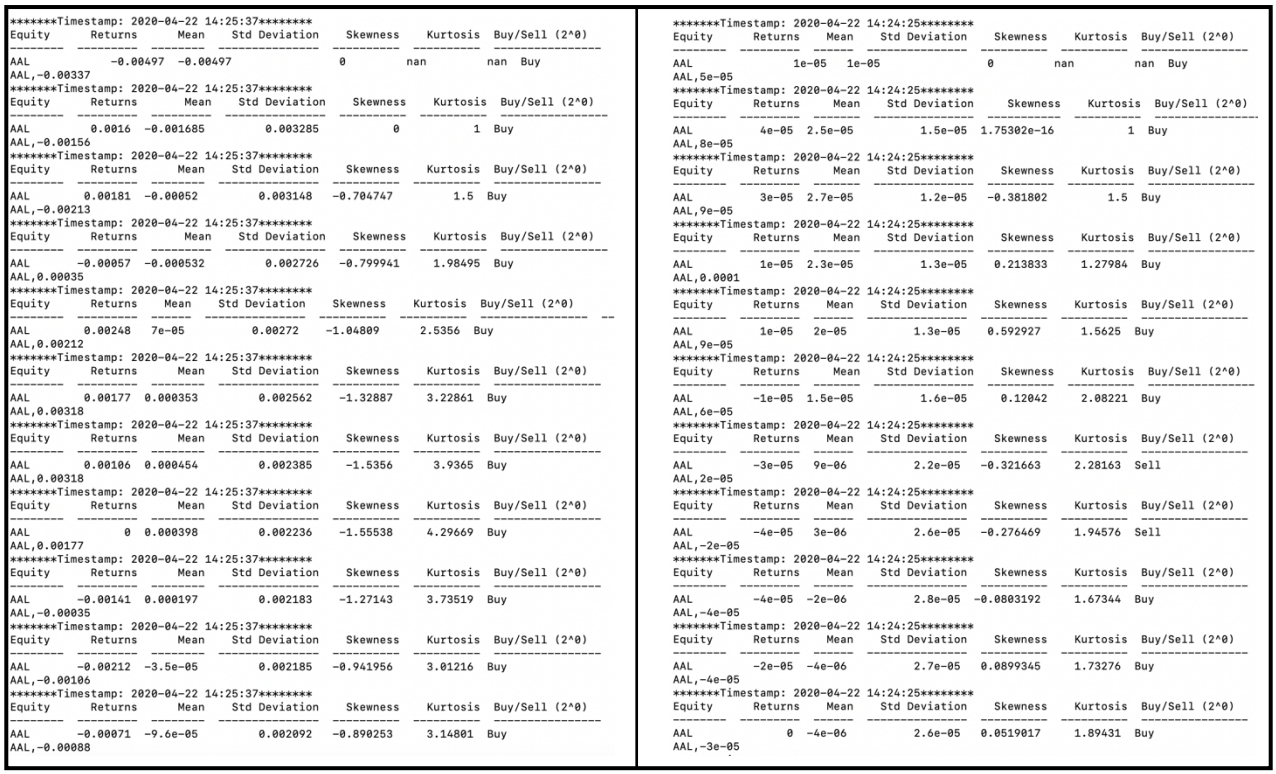


Fig 28: Moving average data smoothing

Fig 29 (a) and (b) show a snapshot of the moments of distribution for the original dataset on the left and for the 100-point Moving Average data on the right. Note that these snapshots display just a few calculations from the very beginning of processing. The standard deviation for the first datapoint is going to be zero, as only one point has been read up till now. Since, standard deviation is present as the denominator in the formulae for skewness and kurtosis, this explains the 'nan' values in the first row of Fig 29 (a) and (b). For the first row in Fig 29 (a) and (b), the returns are equal to the mean, which is consistent with the previous explanation. The mean values for both the distributions remain close to zero all throughout. This confirms that our distribution follows general market behavior. The difference in the two sets of moments lies in the value of standard deviation, skewness, kurtosis and Buy/Sell Indicator.



```
*******Timestamp: 2020-04-22 14:25:37********
Equity      Returns      Mean    Std Deviation    Skewness    Kurtosis  Buy/Sell (2^0)
---------  ---------  --------  ---------------  ----------  ----------  ----------------
AAL           -0.00497  -0.00497                0         nan         nan  Buy
AAL,-0.00337
*******Timestamp: 2020-04-22 14:25:37********
Equity      Returns       Mean    Std Deviation    Skewness    Kurtosis  Buy/Sell (2^0)
---------  ---------  ---------  ---------------  ----------  ----------  ----------------
AAL          0.0016  -0.001685          0.003285           0           1  Buy
AAL,-0.00156
*******Timestamp: 2020-04-22 14:25:37********
Equity      Returns       Mean    Std Deviation    Skewness    Kurtosis  Buy/Sell (2^0)
---------  ---------  ---------  ---------------  ----------  ----------  ----------------
AAL          0.00181  -0.00052          0.003148   -0.704747         1.5  Buy
AAL,-0.00213
*******Timestamp: 2020-04-22 14:25:37********
Equity      Returns       Mean    Std Deviation    Skewness    Kurtosis  Buy/Sell (2^0)
---------  ---------  ---------  ---------------  ----------  ----------  ----------------
AAL          -0.00057  -0.000532          0.002726   -0.799941     1.98495  Buy
AAL,0.00035
*******Timestamp: 2020-04-22 14:25:37********
Equity      Returns     Mean    Std Deviation    Skewness    Kurtosis  Buy/Sell (2^0)
---------  ---------  ------  ---------------  ----------  ----------  ----------------  --
AAL          0.00248   7e-05           0.00272    -1.04809      2.5356  Buy
AAL,0.00212
*******Timestamp: 2020-04-22 14:25:37********
Equity      Returns       Mean    Std Deviation    Skewness    Kurtosis  Buy/Sell (2^0)
---------  ---------  ---------  ---------------  ----------  ----------  ----------------
AAL          0.00177  0.000353          0.002562    -1.32887     3.22861  Buy
AAL,0.00318
*******Timestamp: 2020-04-22 14:25:37********
Equity      Returns       Mean    Std Deviation    Skewness    Kurtosis  Buy/Sell (2^0)
---------  ---------  ---------  ---------------  ----------  ----------  ----------------
AAL          0.00106  0.000454          0.002385    -1.5356      3.9365  Buy
AAL,0.00318
*******Timestamp: 2020-04-22 14:25:37********
Equity      Returns       Mean    Std Deviation    Skewness    Kurtosis  Buy/Sell (2^0)
---------  ---------  ---------  ---------------  ----------  ----------  ----------------
AAL                0  0.000398          0.002236    -1.55538     4.29669  Buy
AAL,0.00177
*******Timestamp: 2020-04-22 14:25:37********
Equity      Returns       Mean    Std Deviation    Skewness    Kurtosis  Buy/Sell (2^0)
---------  ---------  ---------  ---------------  ----------  ----------  ----------------
AAL          -0.00141  0.000197          0.002183    -1.27143     3.73519  Buy
AAL,-0.00035
*******Timestamp: 2020-04-22 14:25:37********
Equity      Returns       Mean    Std Deviation    Skewness    Kurtosis  Buy/Sell (2^0)
---------  ---------  ---------  ---------------  ----------  ----------  ----------------
AAL          -0.00212  -3.5e-05          0.002185   -0.941956     3.01216  Buy
AAL,-0.00106
*******Timestamp: 2020-04-22 14:25:37********
Equity      Returns       Mean    Std Deviation    Skewness    Kurtosis  Buy/Sell (2^0)
---------  ---------  ---------  ---------------  ----------  ----------  ----------------
AAL          -0.00071  -9.6e-05          0.002092   -0.890253     3.14801  Buy
AAL,-0.00088
```

```
*******Timestamp: 2020-04-22 14:24:25********
Equity      Returns      Mean    Std Deviation    Skewness    Kurtosis  Buy/Sell (2^0)
---------  ---------  --------  ---------------  ----------  ----------  ----------------
AAL             1e-05   1e-05                0         nan         nan  Buy
AAL,5e-05
*******Timestamp: 2020-04-22 14:24:25********
Equity      Returns      Mean    Std Deviation    Skewness    Kurtosis  Buy/Sell (2^0)
---------  ---------  --------  ---------------  ----------  ----------  ----------------
AAL             4e-05  2.5e-05          1.5e-05  1.75302e-16           1  Buy
AAL,8e-05
*******Timestamp: 2020-04-22 14:24:25********
Equity      Returns      Mean    Std Deviation    Skewness    Kurtosis  Buy/Sell (2^0)
---------  ---------  --------  ---------------  ----------  ----------  ----------------
AAL             3e-05  2.7e-05          1.2e-05   -0.381802         1.5  Buy
AAL,9e-05
*******Timestamp: 2020-04-22 14:24:25********
Equity      Returns      Mean    Std Deviation    Skewness    Kurtosis  Buy/Sell (2^0)
---------  ---------  --------  ---------------  ----------  ----------  ----------------
AAL             1e-05  2.3e-05          1.3e-05    0.213833     1.27984  Buy
AAL,0.0001
*******Timestamp: 2020-04-22 14:24:25********
Equity      Returns      Mean    Std Deviation    Skewness    Kurtosis  Buy/Sell (2^0)
---------  ---------  --------  ---------------  ----------  ----------  ----------------
AAL             1e-05    2e-05          1.3e-05    0.592927      1.5625  Buy
AAL,9e-05
*******Timestamp: 2020-04-22 14:24:25********
Equity      Returns      Mean    Std Deviation    Skewness    Kurtosis  Buy/Sell (2^0)
---------  ---------  --------  ---------------  ----------  ----------  ----------------
AAL            -1e-05  1.5e-05          1.6e-05     0.12042     2.08221  Buy
AAL,6e-05
*******Timestamp: 2020-04-22 14:24:25********
Equity      Returns      Mean    Std Deviation    Skewness    Kurtosis  Buy/Sell (2^0)
---------  ---------  --------  ---------------  ----------  ----------  ----------------
AAL            -3e-05    9e-06          2.2e-05   -0.321663     2.28163  Sell
AAL,2e-05
*******Timestamp: 2020-04-22 14:24:25********
Equity      Returns      Mean    Std Deviation    Skewness    Kurtosis  Buy/Sell (2^0)
---------  ---------  --------  ---------------  ----------  ----------  ----------------
AAL            -4e-05    3e-06          2.6e-05   -0.276469     1.94576  Sell
AAL,-2e-05
*******Timestamp: 2020-04-22 14:24:25********
Equity      Returns      Mean    Std Deviation    Skewness    Kurtosis  Buy/Sell (2^0)
---------  ---------  --------  ---------------  ----------  ----------  ----------------
AAL            -4e-05  -2e-06          2.8e-05  -0.0803192     1.67344  Buy
AAL,-4e-05
*******Timestamp: 2020-04-22 14:24:25********
Equity      Returns      Mean    Std Deviation    Skewness    Kurtosis  Buy/Sell (2^0)
---------  ---------  --------  ---------------  ----------  ----------  ----------------
AAL            -2e-05  -4e-06          2.7e-05   0.0899345     1.73276  Buy
AAL,-4e-05
*******Timestamp: 2020-04-22 14:24:25********
Equity      Returns      Mean    Std Deviation    Skewness    Kurtosis  Buy/Sell (2^0)
---------  ---------  --------  ---------------  ----------  ----------  ----------------
AAL                0  -4e-06          2.6e-05   0.0519017     1.89431  Buy
AAL,-3e-05
```

(a) **(b)**

Fig 29: Moments of distribution for the original dataset(a) and the averaged dataset(b)

Starting from the second datapoint, value of standard deviation for the averaged/smoothed data is lesser than the original dataset, indicating that the effect of outliers has been reduced by data smoothing. The fluctuations have been decreased and as a result, volatility has decreased too. We can confidently conclude that risks are lowered when data is aggregated. Due to this reason, it can be said that investing in an index is much safer than investing in individual equities. The values for skewness and kurtosis too, are smaller for the aggregated data. Skewness for Fig 29 (a) is largely negative whereas for Fig 29 (b) there are more positive values than negative. Since these figures represent a snapshot of calculations, not much can be said about overall profitability or losses by this alone. The last calculated value for skewness will however, justify profit/loss deductions. Kurtosis for both datasets is increasing over time, but the rate of change in

the aggregated dataset is much smaller than that of the original one. This means that the original dataset diverges from normal behavior at the early stages of processing. Whereas, in the aggregated dataset, the distribution becomes fat-tailed much later. We know that as time progresses, both distributions will be non-gaussian. The difference lies in how fast they get there. All these results point to the fact that risks are reduced while working with smoothed data and this can be attributed to the direct decrease in volatility.

A general pattern difference that can be observed between Fig 29 (a) and (b) is that the respective values of the 2nd, 3rd and 4th moments change slowly over time for the averaged data, whereas these values encounter drastic variations with time in the original dataset. The values see small changes for Fig 29 (b) and relatively larger for Fig 29 (a). The Buy/Sell indicator value is a consequence of risk estimations. It seems fairly obvious, that minimizing risks will result in more 'Buy' values than 'Sell'.

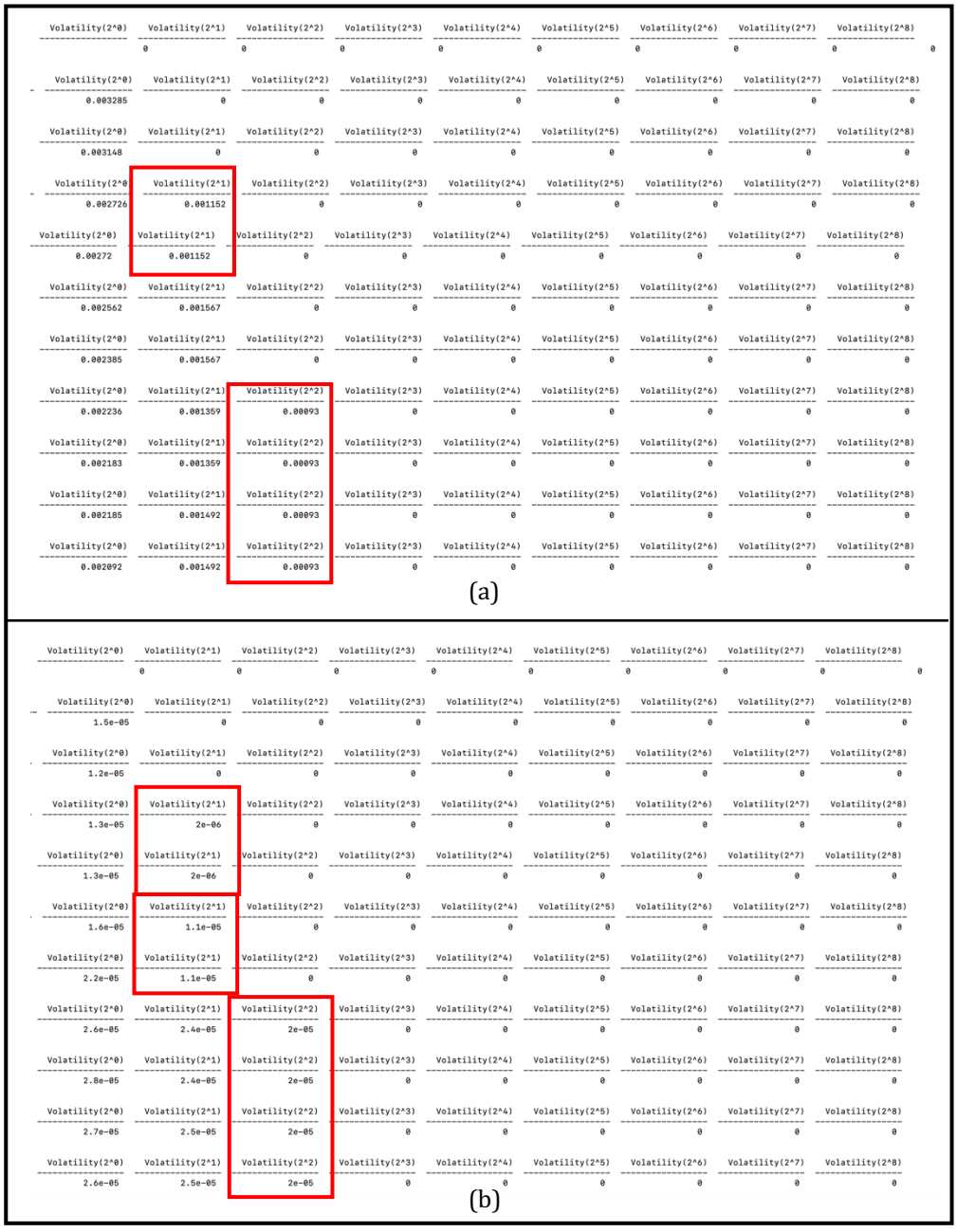


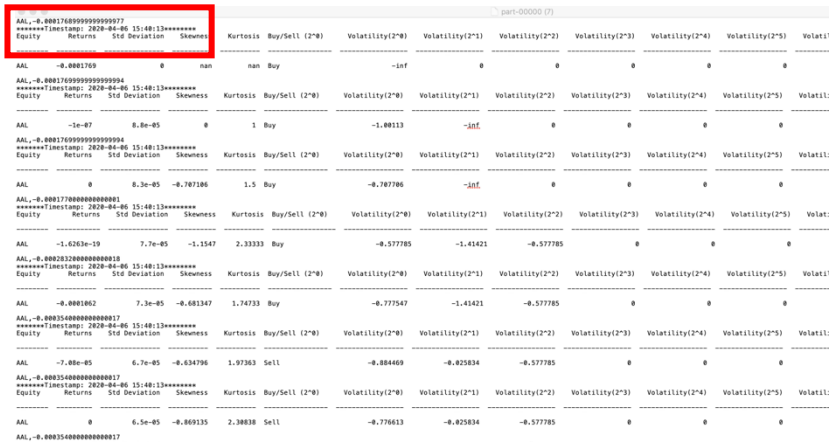
Fig 30: Volatility at different sampling rates for original dataset (a) and averaged dataset (b)

Fig 30 (a) and (b) shows a snapshot of volatility values at different sampling rates for the original dataset and aggregated dataset respectively. These values correspond to the same timestamps, as shown in Fig 30 (a) and (b). A 1-minute (2^0) sampling rate corresponds to the original frequency of the data, and hence, will be considered as reference. Before discussing obvious differences in the two results, it needs to be mentioned that volatility values for timestamps between the sampling frequencies, will remain the same as the previous calculated values. For example, with a sampling rate of 2-minutes, the first volatility value (say *Y)* will be calculated at 9:30AM (assuming market starts at 9:29AM). Now at 9:31 AM, we start a new 2-minute interval and since 9:31AM will be counted as the first minute of the next 2-minute interval, there is no new volatility calculation for this minute yet. This is why the previous calculated value, *Y,* will be carried forward. This becomes a fairly obvious deduction when referred to the boxes if the boxes in the Figure. Similar logic is applied to 4-minute, 8-minute and so on till 256-minute sampling rates.

As can be seen, volatility decreases as the sampling rate increases. This goes for both the original and aggregated dataset. However, the difference lies in the magnitude of values for the two datasets. Aggregation reduces the value of volatility for each sampling rate. Irrespective of data smoothing, sampling too has an overall flattening effect on the distribution. This is why intraday data is more complex to analyze than daily data.

4.2 LATENCY REDUCTION USING PARALLEL PROCESSING

This section shows the Hadoop output which achieves parallel processing for the 3 equities/index. Up till now, we only discussed execution in terms of one equity because the main objective was to demonstrate the approach and calculations. However, to annotate parallelism, all three streams need to be presented. Fig 31 (a), (b) and (c) show snapshots of the Hadoop output. To visualize parallelism, data has been timestamped. It can be seen in Fig 31 (a), (b) and (c) that the execution of all three streams has started simultaneously. However, the output gives no clear indication about the processing.



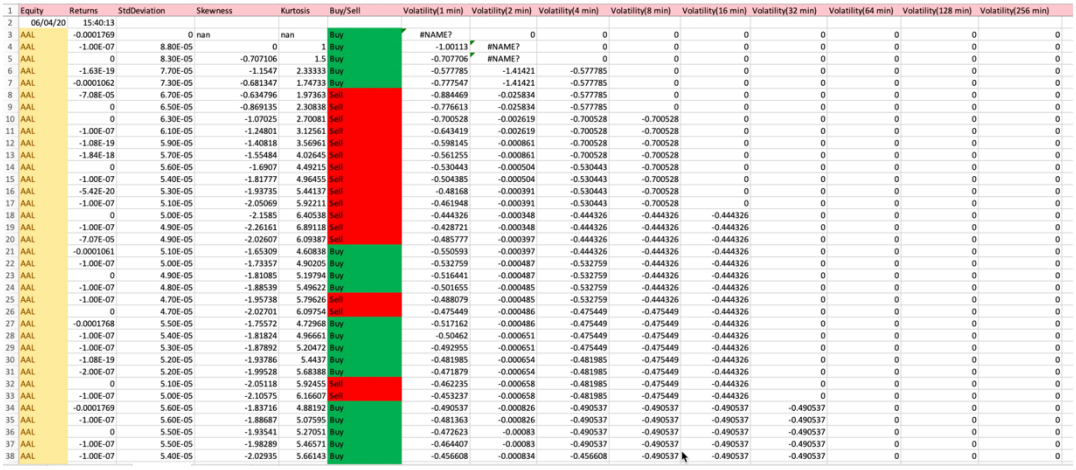
(a)

(b)

(c)

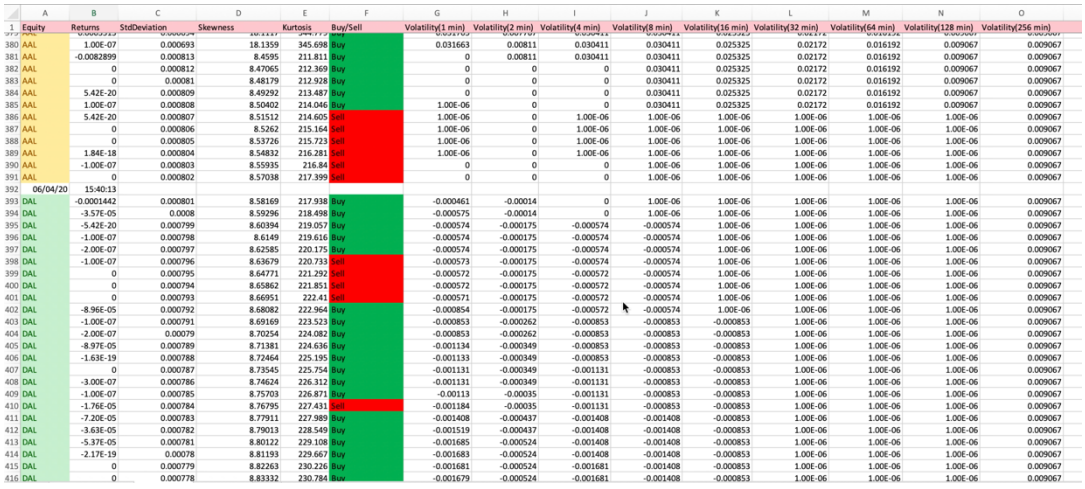
Fig 31: Hadoop Output showing parallelism in the 3 streams of AAL (a), DAL (b) and NDAQ(c)

Fig 32 (a), (b) and (c) show the same output as above, only better represented. These results took 10 hours to process in MapReduce, given limited resources. Latency can be further improved by using multiple machines with more processors and faster CPU/RAM time. The pseudo-distributed mode runs on a single machine and such processing times are impractical for industry use. This is why fully-distributed modes outperform other modes and are recommended for usage with proper resources.



	Equity	Returns	StdDeviation	Skewness	Kurtosis	Buy/Sell	Volatility(1 min)	Volatility(2 min)	Volatility(4 min)	Volatility(8 min)	Volatility(16 min)	Volatility(32 min)	Volatility(64 min)	Volatility(128 min)	Volatility(256 min)
2	06/04/20	15:40:13													
3	AAL	-0.0001769	0	nan	nan	Buy	#NAME?	0	0	0	0	0	0	0	0
4	AAL	-1.00E-07	8.80E-05	0	1	Buy	-1.00113	#NAME?	0	0	0	0	0	0	0
5	AAL	0	8.30E-05	-0.707106	1.5	Buy	-0.707706	#NAME?	0	0	0	0	0	0	0
6	AAL	-1.63E-19	7.70E-05	-1.1547	2.33333	Buy	-0.577785	-1.41421	-0.577785	0	0	0	0	0	0
7	AAL	-0.0001062	7.30E-05	-0.681347	1.74733	Buy	-0.777547	-1.41421	-0.577785	0	0	0	0	0	0
8	AAL	-7.08E-05	6.70E-05	-0.634796	1.97363	Sell	-0.884469	-0.025834	-0.577785	0	0	0	0	0	0
9	AAL	0	6.50E-05	-0.869135	2.30838	Sell	-0.776613	-0.025834	-0.577785	0	0	0	0	0	0
10	AAL	0	6.30E-05	-1.07025	2.70081	Sell	-0.700528	-0.002619	-0.700528	-0.700528	0	0	0	0	0
11	AAL	-1.00E-07	6.10E-05	-1.24801	3.12561	Sell	-0.643419	-0.002619	-0.700528	-0.700528	0	0	0	0	0
12	AAL	-1.08E-19	5.90E-05	-1.40818	3.56961	Sell	-0.598145	-0.000861	-0.700528	-0.700528	0	0	0	0	0
13	AAL	-1.84E-18	5.70E-05	-1.55484	4.02645	Sell	-0.561255	-0.000861	-0.700528	-0.700528	0	0	0	0	0
14	AAL	0	5.60E-05	-1.6907	4.49215	Sell	-0.530443	-0.000504	-0.530443	-0.700528	0	0	0	0	0
15	AAL	-1.00E-07	5.40E-05	-1.81777	4.96455	Sell	-0.504385	-0.000504	-0.530443	-0.700528	0	0	0	0	0
16	AAL	-5.42E-20	5.30E-05	-1.93735	5.44137	Sell	-0.48168	-0.000391	-0.530443	-0.700528	0	0	0	0	0
17	AAL	-1.00E-07	5.10E-05	-2.05069	5.92211	Sell	-0.461948	-0.000391	-0.530443	-0.700528	0	0	0	0	0
18	AAL	0	5.00E-05	-2.1585	6.40538	Sell	-0.444326	-0.000348	-0.444326	-0.444326	-0.444326	0	0	0	0
19	AAL	-1.00E-07	4.90E-05	-2.26161	6.89118	Sell	-0.428721	-0.000348	-0.444326	-0.444326	-0.444326	0	0	0	0
20	AAL	-7.07E-05	4.90E-05	-2.02607	6.09387	Sell	-0.485777	-0.000397	-0.444326	-0.444326	-0.444326	0	0	0	0
21	AAL	-0.0001061	5.10E-05	-1.65309	4.60838	Buy	-0.550593	-0.000397	-0.444326	-0.444326	-0.444326	0	0	0	0
22	AAL	-1.00E-07	5.00E-05	-1.73357	4.90205	Buy	-0.532759	-0.000487	-0.532759	-0.444326	-0.444326	0	0	0	0
23	AAL	0	4.90E-05	-1.81085	5.19794	Buy	-0.516441	-0.000487	-0.532759	-0.444326	-0.444326	0	0	0	0
24	AAL	-1.00E-07	4.80E-05	-1.88539	5.49622	Buy	-0.501655	-0.000485	-0.532759	-0.444326	-0.444326	0	0	0	0
25	AAL	-1.00E-07	4.70E-05	-1.95738	5.79626	Sell	-0.488079	-0.000485	-0.532759	-0.444326	-0.444326	0	0	0	0
26	AAL	0	4.70E-05	-2.02701	6.09754	Sell	-0.475449	-0.000486	-0.475449	-0.475449	-0.444326	0	0	0	0
27	AAL	-0.0001768	5.50E-05	-1.75572	4.72968	Buy	-0.517162	-0.000486	-0.475449	-0.475449	-0.444326	0	0	0	0
28	AAL	-1.00E-07	5.40E-05	-1.81824	4.96661	Buy	-0.50462	-0.000651	-0.475449	-0.475449	-0.444326	0	0	0	0
29	AAL	-1.00E-07	5.30E-05	-1.87892	5.20472	Buy	-0.492955	-0.000651	-0.475449	-0.475449	-0.444326	0	0	0	0
30	AAL	-1.08E-19	5.20E-05	-1.93786	5.4437	Buy	-0.481985	-0.000654	-0.481985	-0.475449	-0.444326	0	0	0	0
31	AAL	-2.00E-07	5.20E-05	-1.99528	5.68388	Buy	-0.471879	-0.000654	-0.481985	-0.475449	-0.444326	0	0	0	0
32	AAL	0	5.10E-05	-2.05118	5.92455	Sell	-0.462235	-0.000658	-0.481985	-0.475449	-0.444326	0	0	0	0
33	AAL	-1.00E-07	5.00E-05	-2.10575	6.16607	Sell	-0.453237	-0.000658	-0.481985	-0.475449	-0.444326	0	0	0	0
34	AAL	-0.0001769	5.60E-05	-1.83716	4.88192	Buy	-0.490537	-0.000826	-0.490537	-0.490537	-0.490537	-0.490537	0	0	0
35	AAL	-1.00E-07	5.60E-05	-1.88687	5.07595	Buy	-0.481363	-0.000826	-0.490537	-0.490537	-0.490537	-0.490537	0	0	0
36	AAL	0	5.50E-05	-1.93541	5.27051	Buy	-0.472623	-0.00083	-0.490537	-0.490537	-0.490537	-0.490537	0	0	0
37	AAL	-1.00E-07	5.50E-05	-1.98289	5.46571	Buy	-0.464407	-0.00083	-0.490537	-0.490537	-0.490537	-0.490537	0	0	0
38	AAL	-1.00E-07	5.40E-05	-2.02935	5.66143	Buy	-0.456608	-0.000834	-0.456608	-0.490537	-0.490537	-0.490537	0	0	0

(a)



	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O
1	Equity	Returns	StdDeviation	Skewness	Kurtosis	Buy/Sell	Volatility(1 min)	Volatility(2 min)	Volatility(4 min)	Volatility(8 min)	Volatility(16 min)	Volatility(32 min)	Volatility(64 min)	Volatility(128 min)	Volatility(256 min)
380	AAL	1.00E-07	0.000693	18.1359	345.698	Buy	0.031663	0.00811	0.030411	0.030411	0.025325	0.02172	0.016192	0.009067	0.009067
381	AAL	-0.0082899	0.000813	8.4595	211.811	Buy	0	0.00811	0.030411	0.030411	0.025325	0.02172	0.016192	0.009067	0.009067
382	AAL	0	0.000812	8.47065	212.369	Buy	0	0	0	0.030411	0.025325	0.02172	0.016192	0.009067	0.009067
383	AAL	0	0.00081	8.48179	212.928	Buy	0	0	0	0.030411	0.025325	0.02172	0.016192	0.009067	0.009067
384	AAL	5.42E-20	0.000809	8.49292	213.487	Buy	0	0	0	0.030411	0.025325	0.02172	0.016192	0.009067	0.009067
385	AAL	1.00E-07	0.000808	8.50402	214.046	Buy	1.00E-06	0	0	0.030411	0.025325	0.02172	0.016192	0.009067	0.009067
386	AAL	5.42E-20	0.000807	8.51512	214.605	Sell	1.00E-06	0	1.00E-06	1.00E-06	1.00E-06	1.00E-06	1.00E-06	1.00E-06	0.009067
387	AAL	0	0.000806	8.5262	215.164	Sell	1.00E-06	0	1.00E-06	1.00E-06	1.00E-06	1.00E-06	1.00E-06	1.00E-06	0.009067
388	AAL	0	0.000805	8.53726	215.723	Sell	1.00E-06	0	1.00E-06	1.00E-06	1.00E-06	1.00E-06	1.00E-06	1.00E-06	0.009067
389	AAL	1.84E-18	0.000804	8.54832	216.281	Sell	1.00E-06	0	1.00E-06	1.00E-06	1.00E-06	1.00E-06	1.00E-06	1.00E-06	0.009067
390	AAL	-1.00E-07	0.000803	8.55935	216.84	Sell	0	0	0	1.00E-06	1.00E-06	1.00E-06	1.00E-06	1.00E-06	0.009067
391	AAL	0	0.000802	8.57038	217.399	Sell	0	0	0	1.00E-06	1.00E-06	1.00E-06	1.00E-06	1.00E-06	0.009067
392	06/04/20	15:40:13													
393	DAL	-0.0001442	0.000801	8.58169	217.938	Buy	-0.000461	-0.00014	0	1.00E-06	1.00E-06	1.00E-06	1.00E-06	1.00E-06	0.009067
394	DAL	-3.57E-05	0.0008	8.59296	218.498	Buy	-0.000575	-0.00014	0	1.00E-06	1.00E-06	1.00E-06	1.00E-06	1.00E-06	0.009067
395	DAL	-5.42E-20	0.000799	8.60394	219.057	Buy	-0.000574	-0.000175	-0.000574	-0.000574	1.00E-06	1.00E-06	1.00E-06	1.00E-06	0.009067
396	DAL	-1.00E-07	0.000798	8.6149	219.616	Buy	-0.000574	-0.000175	-0.000574	-0.000574	1.00E-06	1.00E-06	1.00E-06	1.00E-06	0.009067
397	DAL	-2.00E-07	0.000797	8.62585	220.175	Buy	-0.000574	-0.000175	-0.000574	-0.000574	1.00E-06	1.00E-06	1.00E-06	1.00E-06	0.009067
398	DAL	-1.00E-07	0.000796	8.63679	220.733	Sell	-0.000573	-0.000175	-0.000574	-0.000574	1.00E-06	1.00E-06	1.00E-06	1.00E-06	0.009067
399	DAL	0	0.000795	8.64771	221.292	Sell	-0.000572	-0.000175	-0.000572	-0.000574	1.00E-06	1.00E-06	1.00E-06	1.00E-06	0.009067
400	DAL	0	0.000794	8.65862	221.851	Sell	-0.000572	-0.000175	-0.000572	-0.000574	1.00E-06	1.00E-06	1.00E-06	1.00E-06	0.009067
401	DAL	0	0.000793	8.66951	222.41	Sell	-0.000571	-0.000175	-0.000572	-0.000574	1.00E-06	1.00E-06	1.00E-06	1.00E-06	0.009067
402	DAL	-8.96E-05	0.000792	8.68082	222.964	Buy	-0.000854	-0.000175	-0.000572	-0.000574	1.00E-06	1.00E-06	1.00E-06	1.00E-06	0.009067
403	DAL	-1.00E-07	0.000791	8.69169	223.523	Buy	-0.000853	-0.000262	-0.000853	-0.000853	-0.000853	1.00E-06	1.00E-06	1.00E-06	0.009067
404	DAL	-2.00E-07	0.00079	8.70254	224.082	Buy	-0.000853	-0.000262	-0.000853	-0.000853	-0.000853	1.00E-06	1.00E-06	1.00E-06	0.009067
405	DAL	-8.97E-05	0.000789	8.71381	224.636	Buy	-0.001134	-0.000349	-0.000853	-0.000853	-0.000853	1.00E-06	1.00E-06	1.00E-06	0.009067
406	DAL	-1.63E-19	0.000788	8.72464	225.195	Buy	-0.001133	-0.000349	-0.000853	-0.000853	-0.000853	1.00E-06	1.00E-06	1.00E-06	0.009067
407	DAL	0	0.000787	8.73545	225.754	Buy	-0.001131	-0.000349	-0.001131	-0.000853	-0.000853	1.00E-06	1.00E-06	1.00E-06	0.009067
408	DAL	-3.00E-07	0.000786	8.74624	226.312	Buy	-0.001131	-0.000349	-0.001131	-0.000853	-0.000853	1.00E-06	1.00E-06	1.00E-06	0.009067
409	DAL	-1.00E-07	0.000785	8.75703	226.871	Buy	-0.00113	-0.00035	-0.001131	-0.000853	-0.000853	1.00E-06	1.00E-06	1.00E-06	0.009067
410	DAL	-1.76E-05	0.000784	8.76795	227.431	Sell	-0.001184	-0.00035	-0.001131	-0.000853	-0.000853	1.00E-06	1.00E-06	1.00E-06	0.009067
411	DAL	-7.20E-05	0.000783	8.77911	227.989	Buy	-0.001408	-0.000437	-0.001408	-0.001408	-0.000853	1.00E-06	1.00E-06	1.00E-06	0.009067
412	DAL	-3.63E-05	0.000782	8.79013	228.549	Buy	-0.001519	-0.000437	-0.001408	-0.001408	-0.000853	1.00E-06	1.00E-06	1.00E-06	0.009067
413	DAL	-5.37E-05	0.000781	8.80122	229.108	Buy	-0.001685	-0.000524	-0.001408	-0.001408	-0.000853	1.00E-06	1.00E-06	1.00E-06	0.009067
414	DAL	-2.17E-19	0.00078	8.81193	229.667	Buy	-0.001683	-0.000524	-0.001408	-0.001408	-0.000853	1.00E-06	1.00E-06	1.00E-06	0.009067
415	DAL	0	0.000779	8.82263	230.226	Buy	-0.001681	-0.000524	-0.001681	-0.001408	-0.000853	1.00E-06	1.00E-06	1.00E-06	0.009067
416	DAL	0	0.000778	8.83332	230.784	Buy	-0.001679	-0.000524	-0.001681	-0.001408	-0.000853	1.00E-06	1.00E-06	1.00E-06	0.009067

(b)

	Equity	Returns	StdDeviation	Skewness	Kurtosis	Buy/Sell	Volatility(1 min)	Volatility(2 min)	Volatility(4 min)	Volatility(8 min)	Volatility(16 min)	Volatility(32 min)	Volatility(64 min)	Volatility(128 min)	Volatility(256 min)
778	DAL	2.00E-07	0.000642	8.0928	283.259	Sell	0.000181	8.50E-05	0	0.0131	0.0131	0.0131	0.0131	0.0131	0.0131
779	DAL	0	0.000642	8.09802	283.625	Sell	0.000181	8.50E-05	0.000181	0.000181	0.0131	0.0131	0.0131	0.0131	0.0131
780	DAL	1.00E-07	0.000641	8.10324	283.99	Sell	0.000181	8.50E-05	0.000181	0.000181	0.0131	0.0131	0.0131	0.0131	0.0131
781	DAL	0	0.000641	8.10845	284.356	Sell	0.000181	8.50E-05	0.000181	0.000181	0.0131	0.0131	0.0131	0.0131	0.0131
782	06/04/20	15:40:14													
783	NDAQ	-0.002729	0.000648	7.75853	272.483	Buy	-0.00523	8.50E-05	0.000181	0.000181	0.0131	0.0131	0.0131	0.0131	0.0131
784	NDAQ	-0.000614	0.000648	7.75281	272.251	Buy	-0.006439	-0.003064	-0.006439	0.000181	0.0131	0.0131	0.0131	0.0131	0.0131
785	NDAQ	-0.001371	0.000649	7.76804	269.626	Buy	-0.00912	-0.003064	-0.006439	0.000181	0.0131	0.0131	0.0131	0.0131	0.0131
786	NDAQ	-0.000662	0.000649	7.76047	269.308	Buy	-0.010413	-0.004974	-0.006439	0.000181	0.0131	0.0131	0.0131	0.0131	0.0131
787	NDAQ	-0.001326	0.00065	7.62209	266.931	Buy	-0.012982	-0.004974	-0.006439	0.000181	0.0131	0.0131	0.0131	0.0131	0.0131
788	NDAQ	-0.000284	0.00065	7.62586	267.166	Buy	-0.01353	-0.006484	-0.01353	-0.01353	-0.01353	0.0131	0.0131	0.0131	0.0131
789	NDAQ	-0.000379	0.00065	7.62794	267.308	Buy	-0.014261	-0.006484	-0.01353	-0.01353	-0.01353	0.0131	0.0131	0.0131	0.0131
790	NDAQ	-0.000854	0.00065	7.61042	266.556	Buy	-0.015908	-0.007639	-0.01353	-0.01353	-0.01353	0.0131	0.0131	0.0131	0.0131
791	NDAQ	-0.00019	0.00065	7.61518	266.854	Buy	-0.016269	-0.007639	-0.01353	-0.01353	-0.01353	0.0131	0.0131	0.0131	0.0131
792	NDAQ	-0.00019	0.00649	7.61993	267.152	Buy	-0.016629	-0.007994	-0.016629	-0.01353	-0.01353	0.0131	0.0131	0.0131	0.0131
793	NDAQ	0	0.000649	7.6247	267.49	Buy	-0.016618	-0.007994	-0.016629	-0.01353	-0.01353	0.0131	0.0131	0.0131	0.0131
794	NDAQ	-0.000616	0.000649	7.61946	267.277	Buy	-0.0178	-0.00857	-0.016629	-0.01353	-0.01353	0.0131	0.0131	0.0131	0.0131
795	NDAQ	-0.000238	0.000649	7.62379	267.547	Buy	-0.018252	-0.00857	-0.016629	-0.01353	-0.01353	0.0131	0.0131	0.0131	0.0131
796	NDAQ	-0.000283	0.000648	7.62758	267.783	Buy	-0.018789	-0.009058	-0.018789	-0.018789	-0.01353	0.0131	0.0131	0.0131	0.0131
797	NDAQ	-0.00019	0.000648	7.63232	268.081	Buy	-0.019146	-0.009058	-0.018789	-0.018789	-0.01353	0.0131	0.0131	0.0131	0.0131
798	NDAQ	-9.50E-05	0.000647	7.63737	268.413	Buy	-0.019319	-0.009324	-0.018789	-0.018789	-0.01353	0.0131	0.0131	0.0131	0.0131
799	NDAQ	0	0.000647	7.6421	268.75	Buy	-0.019307	-0.009324	-0.018789	-0.018789	-0.01353	0.0131	0.0131	0.0131	0.0131
800	NDAQ	-9.50E-05	0.000647	7.64715	269.082	Buy	-0.019479	-0.009411	-0.019479	-0.018789	-0.01353	0.0131	0.0131	0.0131	0.0131
801	NDAQ	-9.50E-05	0.000646	7.6522	269.414	Buy	-0.019651	-0.009411	-0.019479	-0.018789	-0.01353	0.0131	0.0131	0.0131	0.0131
802	NDAQ	0	0.000646	7.65692	269.751	Buy	-0.019639	-0.009499	-0.019479	-0.018789	-0.01353	0.0131	0.0131	0.0131	0.0131
803	NDAQ	-0.00019	0.000645	7.66164	270.048	Buy	-0.019994	-0.009499	-0.019479	-0.018789	-0.01353	0.0131	0.0131	0.0131	0.0131
804	NDAQ	-0.000381	0.000645	7.66366	270.188	Buy	-0.020716	-0.010033	-0.020716	-0.020716	-0.020716	-0.020716	0.0131	0.0131	0.0131
805	NDAQ	0	0.000645	7.66837	270.525	Buy	-0.020703	-0.010033	-0.020716	-0.020716	-0.020716	-0.020716	0.0131	0.0131	0.0131
806	NDAQ	0	0.000644	7.67307	270.862	Buy	-0.02069	-0.010032	-0.020716	-0.020716	-0.020716	-0.020716	0.0131	0.0131	0.0131
807	NDAQ	-0.000192	0.000644	7.67776	271.158	Buy	-0.021047	-0.010032	-0.020716	-0.020716	-0.020716	-0.020716	0.0131	0.0131	0.0131
808	NDAQ	-1.00E-06	0.000644	7.68246	271.495	Buy	-0.021036	-0.010212	-0.021036	-0.020716	-0.020716	-0.020716	0.0131	0.0131	0.0131
809	NDAQ	-4.50E-05	0.000643	7.68738	271.833	Buy	-0.02111	-0.010212	-0.021036	-0.020716	-0.020716	-0.020716	0.0131	0.0131	0.0131
810	NDAQ	-9.70E-05	0.000643	7.6924	272.164	Buy	-0.021284	-0.010343	-0.021036	-0.020716	-0.020716	-0.020716	0.0131	0.0131	0.0131
811	NDAQ	-0.000141	0.000642	7.69733	272.483	Buy	-0.021542	-0.010343	-0.021036	-0.020716	-0.020716	-0.020716	0.0131	0.0131	0.0131
812	NDAQ	-9.50E-05	0.000642	7.70234	272.815	Buy	-0.021711	-0.010563	-0.021711	-0.021711	-0.020716	-0.020716	0.0131	0.0131	0.0131
813	NDAQ	-2.00E-06	0.000642	7.70703	273.151	Buy	-0.021702	-0.010563	-0.021711	-0.021711	-0.020716	-0.020716	0.0131	0.0131	0.0131
814	NDAQ	0	0.000641	7.71171	273.488	Buy	-0.021688	-0.010564	-0.021711	-0.021711	-0.020716	-0.020716	0.0131	0.0131	0.0131

(c)

Fig 32: Formatted representation for AAL (a), DAL (b) and NDAQ (c) respectively

4.3 CASE STUDIES

Big Data Analytics is used in the financial sector for prediction and analysis of large volumes of datasets. Elements of Apache Hadoop Framework, are used to achieve distributed processing and storage across many fields. In this section, we review related work in the field of financial data analysis as well as study associated literature concerning the implementation of the architecture used in this project. Following narrative attempts to illustrate cases that solve a problem with similar solutions proposed in this project as well as the ones that use a different approach. The novelty of this project lies in the approach to data generation and integration of Hadoop MapReduce with financial analytical approach. Numerous papers in this field, use the Hadoop interface to take advantage of the multifaceted distributed file system: HDFS. Large data storage, high availability, fault tolerance and cost effectiveness are some on the many aspects of HDFS that make it highly versatile. [34] and [35] implement HDFS along with MapReduce paradigms that provide scalable and distributed machine learning algorithms. The interface can be applied with recommendation systems as done in [34], or machine learning models to carry out efficient stock market analysis [35]. Emphasis on real-time processing has been driving big data analytics towards new paradigms within the Hadoop Framework, like Apache Hadoop Spark or Apache Mahout. This will be discussed in future work. Not many literatures explore the practicality of the Hadoop Architecture with financial big data, but there are a multitude of them when it comes to discussing the architecture theoretically. [36] employs a MapReduce based data preprocessing algorithm along with a recommendation system. After carefully reviewing many papers, it seems like MapReduce is still very eminent in the e-commerce field whereas the financial industry is seeing a transition from MapReduce to other parallel programming models. On the contrary, there are certain approaches that aim to achieve the similar objective, but don't use "big data" technologies at all. [37] uses data analytics on various platforms, to predict stock prices. [38] analyses financial statistics of high frequency data for risk management. In terms of the objective, this paper is closest to what we are trying to achieve in this project. But the difference lies in the implementation, such that [38] uses parallel R. Nonetheless, in essence, it can be said that to tackle big data challenges we need parallelism and scalability more than anything. The choice platforms, interfaces or frameworks used, is left upto the developer's perspective and understanding. Most of the papers studied in this section have the idea of distributed parallel processing and scalable storage in common and the most popular platform out there, which provides the right tools for this is the Apache Hadoop Framework.

5. FUTUTE WORK

The financial sector is a continuously growing field for applied data analytics. Since the data is growing at an exponential rate across all industries it calls for a faster processing application. There is no doubt about the fact that parallelizing tasks decreases response time for any application but there are features beyond this functionality that can result in differences in response times between applications. Apache Hadoop uses batch processing in which a machine processes batches of jobs, often simultaneously, in a non-stop sequential order[39]. When the volume of data gets too large, delays from batch processing will become significant when compared with other response time metrics. Although Hadoop outperforms other technologies when it comes to huge volumes of data, the financial industry is highly time-critical and prioritizes processing speed above all. Keeping this in mind, in future, real-time analytics can be approached by using the strength of Hadoop HDFS to store Big Data and mount a general-purpose computing framework, like Apache Spark, on top of it.

Apache spark uses real-time processing by programming entire clusters with implicit data parallelism and fault tolerance. The processing speed in Spark is in fact, so high, that it can perform streaming analytics as well. This will reduce latency considerably. The choice of batch analysis in this project is driven by the fact that 1-minute frequency cannot be considered as real-time data in terms of financial datasets. When it comes to stock market series, data is generated at micro-second frequencies, which are generated by trading algorithms on machines. While talking about such ultra-high frequencies, there might be potential for the need of real-time processing platforms like Apache Spark. Since we are using historical data instead of live data streams, there is all the more plausibility in using Hadoop. Both Hadoop and Spark have a different set of advantages that can be used to produce unique results. Nonetheless, replacing Hadoop MapReduce with Apache Spark will provide a different perspective to the problem and take it towards the direction of real-time data analytics.

6. CONCLUSION

Growth in data has led to the evolution of new technologies. Data is abundantly being generated in every industry. Each sector has a different use for information systems, to deal with the type of data they generate. The financial sector is leading all others in terms of the volume and velocity of data that it produces and records. High-frequency financial data is difficult to find on the internet and hence this project implements synthetic data generation techniques to create big data out of the given resources. Concurrent processing of high frequency data gives rise to challenges that need compatible solutions. Value and insights need to be generated promptly, for which low latency architectures are called for. This project applies the most well-known industry accepted big data technology called the Hadoop Framework.

Different components of the Hadoop Framework realize different purposes and carry distinct responsibilities, all of which work together to achieve a common goal of distributed parallel processing. This project combines the scalability, fault tolerance and high data availability provided by HDFS with parallel processing carried out by MapReduce, to yield financial statistics. These financial statistics or moments of distribution for financial stock series, are used to analyze the market and conduct risk estimations. Such performance metrics can decipher general trends and reveal market behavior. Additionally, this project also explores the working of YARN which is very "resourceful" while working with multiple applications, as it works as a scheduler. Various practical and theoretical techniques have been discussed in an effort to attain low latency risk estimation with financial big data.

7. BIBLIOGRAPHY

[0]: From https://www.lexico.com/definition/big_data (part of Oxford Dictionaries Online, 29 March 2020)

[1]: Almeida, F., 2018. Big data: Concept, potentialities and vulnerabilities. *Emerging Science Journal*, *2*(1), pp.1-10.

[2]: National Research Council, 2013. *Frontiers in massive data analysis*. National Academies Press.

[3]: Anuradha, J., 2015. A brief introduction on Big Data 5Vs characteristics and Hadoop technology. *Procedia computer science*, *48*, pp.309-314.

[4]: SAP, Big Data and Smart Trading: How a Real-time Data Platform Maximizes Trading Opportunities (2012)

[5]: Reinsel, D., Gantz, J. and Rydning, J., 2018. The digitization of the world from edge to core. *IDC White Paper*.

[6]: Tian, X., Han, R., Wang, L., Lu, G. and Zhan, J., 2015. Latency critical big data computing in finance. *The Journal of Finance and Data Science*, *1*(1), pp.32-41.

[7]: https://www.investopedia.com/terms/s/stock.asp

[8]: https://corporatefinanceinstitute.com/resources/knowledge/trading-investing/spot-price/

[9]:From https://www-oxfordreference-com.elib.tcd.ie/view/10.1093/acref/9780199688975.001.0001/acref-9780199688975-e-2745?rskey=qar5YL&result=3061

[10]: Chang, B.Y., Christoffersen, P. and Jacobs, K., 2013. Market skewness risk and the cross section of stock returns. *Journal of Financial Economics*, *107*(1), pp.46-68.

[11]: From https://www-oxfordreference-com.elib.tcd.ie/view/10.1093/acref/9780198789741.001.0001/acref-9780198789741-e-6068?rskey=J45djS&result=5371

[12]: Bai, Y. and Green, C.J., 2011. Determinants of cross-sectional stock return variations in emerging markets. *Empirical Economics*, *41*(1), pp.81-102.

[13]: Heston, S.L. and Rouwenhorst, K.G., 1995. Industry and country effects in international stock returns. *Journal of Portfolio Management*, *21*, pp.53-53.

[14]: https://corporatefinanceinstitute.com/resources/knowledge/trading-investing/fundamental-analysis/

[15]: https://www.investopedia.com/ask/answers/050515/it-better-use-fundamental-analysis-technical-analysis-or-quantitative-analysis-evaluate-longterm.asp

[16]: Griffioen, G.A., 2003. Technical analysis in financial markets.

[17]: From https://www-oxfordreference-com.elib.tcd.ie/view/10.1093/acref/9780199679591.001.0001/acref-9780199679591-e-1873?rskey=vbUk8h&result=2041

[18]: Taylor, S.J., 2011. *Asset price dynamics, volatility, and prediction*. Princeton university press, Chapter 7.

[19]: Taylor, S.J., 2011. *Asset price dynamics, volatility, and prediction*. Princeton university press, Chapter 12

[20]: Analysis of Financial Time Series by Ruey S. Tsay

[21]: Taylor, S.J., 2011. *Asset price dynamics, volatility, and prediction*. Princeton university press, Chapter 1

[22]: Bhathal, G.S. and Singh, A., 2019. Big data: Hadoop framework vulnerabilities, security issues and attacks. *Array*, *1*, p.100002.

[23]: Sur, S., Wang, H., Huang, J., Ouyang, X. and Panda, D.K., 2010, December. Can high-performance interconnects benefit hadoop distributed file system. In *Workshop on Micro Architectural Support for Virtualization, Data Center Computing, and Clouds (MASVDC). Held in Conjunction with MICRO* (p. 10).

[24]: Appuswamy, R., Gkantsidis, C., Narayanan, D., Hodson, O. and Rowstron, A., 2013, October. Scale-up vs scale-out for hadoop: Time to rethink?. In *Proceedings of the 4th annual Symposium on Cloud Computing* (pp. 1-13).

[25]: https://hadoop.apache.org/docs/stable/hadoop-project-dist/hadoop-hdfs/HdfsDesign.html

[26]: https://hadoop.apache.org/docs/stable/hadoop-mapreduce-client/hadoop-mapreduce-client-core/MapReduceTutorial.html

[27]: https://hadoop.apache.org/docs/stable/hadoop-yarn/hadoop-yarn-site/YARN.html

[28]: Taylor, S.J., 2011. *Asset price dynamics, volatility, and prediction*. Princeton university press, Chapter 3

[29]: Taylor, S.J., 2011. *Asset price dynamics, volatility, and prediction*. Princeton university press, Chapter 4

[30]: Sewell, M., 2011. Characterization of financial time series. *Rn*, *11*(01), p.01.

[31]: http://sphweb.bumc.bu.edu/otlt/MPH-Modules/BS/BS704_Probability/BS704_Probability9.html

[32]: Taylor, S.J., 2011. *Asset price dynamics, volatility, and prediction*. Princeton university press, Chapter 2

[33]: Harris, L., 1986. A transaction data study of weekly and intradaily patterns in stock returns. *Journal of financial economics*, *16*(1), pp.99-117.

[34]: P. K. M. Kanaujia, M. Pandey and S. S. Rautaray, "Real time financial analysis using big data technologies," *2017 International Conference on I-SMAC (IoT in Social, Mobile, Analytics and Cloud) (I-SMAC)*, Palladam, 2017, pp. 130-135.

[35]: Z. Peng, "Stocks Analysis and Prediction Using Big Data Analytics," *2019 International Conference on Intelligent Transportation, Big Data & Smart City (ICITBS)*, Changsha, China, 2019, pp. 299-302.

[36]: S. Saravanan, "Design of large-scale Content-based recommender system using hadoop MapReduce framework," *2015 Eighth International Conference on Contemporary Computing (IC3)*, Noida, 2015, pp. 292-297.

[37]: S. Tiwari, A. Bharadwaj and S. Gupta, "Stock price prediction using data analytics," *2017 International Conference on Advances in Computing, Communication and Control (ICAC3)*, Mumbai, 2017, pp. 1-5.

[38]: J. Zou and H. Zhang, "High-frequency financial statistics with parallel R and Intel Xeon Phi coprocessor," *2014 IEEE International Conference on Big Data (Big Data)*, Washington, DC, 2014, pp. 61-69.

[39]: https://www.bmc.com/blogs/what-is-batch-processing-batch-processing-explained/