Hedging Optimisation with K-means Clustering on Cryptocurrencies

Chengkai Lu

Submitted for the Degree of Master of Science in

Data Science and Analytics with a Year in Industry

Department of Computer Science
Royal Holloway University of London
Egham, Surrey TW20 0EX, UK

August 28, 2018

Declaration

This report has been prepared on the basis of my own work. Where other published and unpublished source materials have been used, these have been acknowledged.

Word Count:

Student Name: Chengkai Lu

Date of Submission:

Signature:

Abstract

Cryptocurrencies like Bitcoin and Ripple are becoming popular in these years. They can be obtained through mining or transactions. Every individual cryptocurrency can be transferred directly using a public key in digital wallets. Their price is mostly tied to supply/demand and hard to be interfered by governments. In addition, cryptocurrencies have some dependencies because some of them like Dogecoin can only be bought by some major cryptocurrencies such as Bitcoin and Ethereum in cryptocurrency exchanges.

Value at Risk is a popular traditional method in financial technical analysis for estimating potential total risks of a financial portfolio. A well-estimated risk can prevent an investor or a financial institution from losing more than their expectation. In reality, the financial markets sometimes are unpredictable. Investors are striving for putting all the risks under their control.

K-means clustering is a popular unsupervised learning algorithm for grouping unlabelled data. It aims on finding the natural way of separating observations into diffident clusters and is quite popular in the area of marketing for discovering potential customer behaviour. However, it might be also applicable to group stocks/cryptocurrencies having similar price movement together, which means a clustering algorithm may help diversify investors' investment. It states that a systematic way of hedging is achievable.

A modern machine learning algorithm for diversifying investment before a traditional technical analysis might improve the performance of risk estimation. A better hedging can prevent investors from suffering disastrous loss. Especially in the market of cryptocurrency, the price movement is more irrational and abnormal because of lacking regulation. This project will try to research if the optimisation of risk estimation performs well in the market of cryptocurrency.

Contents

List of Figures

List of Tables

Acknowledgement

Thanks to Dr. Yuri Kalnishkan for his supervision and guidance on this project. He shows his passion and expertise in the area of machine learning, data analysis and finance. His effort and comprehensive knowledge have guided this project and myself onto the right path of concrete analysis on financial data. In addition, thanks to Professor Alberto Paccanaro for his help in the early stages of this project and introducing me to Dr. Yuri Kalnishkan.

Some of the contents referenced the slides and notes of CS5100-Data Analysis and CS5920-Machine Learning which are compiled by Professor Vladimir Vovk and Professor Zhiyuan Luo. Any other references have been clarified in the end of this project on the Reference section. This project is used for non-profit purposes. Any other uses of this project should be informed, and the author reserves the right to refuse any inappropriate or illegal usage.

1 Introduction

Hedging is an important investing strategy to avoid risks that will result in substantial losses or gains. Investors have been developing and implementing traditional technical analysis on evaluating risks of an investment, but it is still tricky to capture some events such as a financial crash which breaks the previous risk estimation. Some of the tasks are computationally expensive and human beings are hard to achieve, therefore, computer science and machine learning has been used for improving the performance of analysis.

Machine learning has become a popular area which focuses on enabling machines or computers to learn and discover the patterns. It can be applied to different domains as long as there exists data. Especially in the financial area, people are striving for using modern computing power to deal with problems efficiently because time is a crucial factor in finance, and this is called computational finance.

1.1 Motivation

When we talk about the risk of an investment, we are normally curious about the risk for the future. Prediction of risks can be tricky because a randomly unpredictable event will impact on the market movement. An unexpected financial event might collapse the market and result in a crash. All the products in the market will be influenced by the risks. These kinds of risks are called *Systematic Risk*, which cannot be avoided through a diversified investment portfolio. However, we can spread the risks over diffirent investments by generating a well-diversified investment portfolio.

Traditional risk management methods have been developed for centuries and they can produce reasonable result of hedges, but there still exists possibilities to improve the performance. Machine learning are also used for avoidance of risk or market prediction. We want to seek for a way of improving traditional risk management methods by applying machine learning algorithms on top.

1.2 Aims And Objectives

Value at Risk is a traditional method which can evaluate the risk of a investment portfolio or a single investment. It is widely used by a large number of financial institutions, especially banks because it provides a quantitative criterion for risk estimation. However, the risk might be overestimated or underestimated since the products in a portfolio might have varied volatil-

ities and activities. There also might be some events that have happened on one product, and another product might experience it in the future. For example, we might apply VaR of the combination of two products with similar activities and cover the worst/best cases of a product. Another product will then have the ability to consider the possible worst cases.

A diversification of investment before generating a portfolio can remove the concern of wrongly evaluating the risks. Well-diversified financial portfolios will ensure that products in a investment combination have similar risks or volatilities.

K-means clustering is an appropriate method for diversifying financial investments. It can create natural grouping on the financial products by putting those with similar price volatilities together. That is, it looks at the shape of each product along the timeline and estimates the similarity between two lines to decide if the two products should be in the same cluster.

This project aims on seeing if K-means clustering can diversify financial products with their daily volatilities and improve the result of risk estimation on cryptocurrencies. In addition, it will examine if the effect on cryptocurrencies is different from that on stocks.

1.3 Project Structure

The project includes several parts, experiment, implementation of the analysis and report. The experiment is for developing and adjusting the analysis. After receiving a satisfactory result from the experiment, the analysis will be implemented on a minimal production-level basis. This report will then present the most critical detail of the analysis.

1.3.1 Technologies

In order to facilitate the process of analysis, we use several technologies to develop this project. Scalability is also important for a proper project. As a result, we will build a well-structured programme that meet the minimal requirement of scalability and can be productionised in the future.

In this project, the following technologies are used for different purposes:

- Programming Languages
  - *Python* : A popular programming language for statistics, machine learning and data analysis due to its flexibility, powerful packages, ease of learning and data visualisation. We use Python for most of the analysis because it is handy for quick experiments.

  - *Java* : Many companies are using Java as the main languages for their servers and applications because of its efficiency, performance and stability. We use Java for the ETL pipeline and a preparation of productionising the result of analysis.

- Database
  - *PostgreSQL*[11] : We use PostgreSQL as our database because it is well-developed and supports efficient quaries.

- Containerisation
  - *Docker*[5] : An open source operating-system-level virtualisation tool to build an environment for running programmes. It not only provides an diffirent environment such as Linux on a local machine, but makes the same configuration and environment portable. We use Docker for a quick construction of PostgreSQL database.

- Version Control
  - *Git*[6] : A popular tool for version control. The project uses it for managing records of development.
  - *GitHub*[7] : All the records and resources of this project are stored into a private repository on GitHub which is a website for storing local Gits onto the cloud.

- Frameworks
  - *Apache Spark*[3] : An open source framework for cluster computing. This is for preparation of productionisation of this project and changing the scripting python code to an automated programme. We will use Spark SQL to accelerate reading data from the database and the MLlib for analysis with machine learning.
  - *Spring Framework*[13] : An open source Java framework for implementation of inversion of control. We use it to facilitate programming process and for preparation of building a web server in the future.

- Project Management
  - *Apache Maven*[2] : A tool for project management in Java. It can create an automatic pipeline for installing libraries, packaging programmes, managing package dependencies, etc.

- Python Packages
  - *Numpy*[21] : Array manipulation.
  - *Pandas*[21] : Data manipulation and analysis.
  - *Scikit-Learn*[21] : Machine learning and data mining.
  - *Tslearn*[24] : Time series data processing and analysis.
  - *Matplotlib*[21] : Data visualisation.
- Java Libraries
  - *Joda-Time*[10] : Datetime and timestamps manipulation.
  - *Project Lombok*[12] : Being used for simplifying code and logging.
  - *Gson*[8] : Json format data processing.
  - *Apache HttpComponents*[1] : Asynchronous http request.

1.3.2 Programme Structure

The programme is mainly composed by the following 3 parts:

- Database
- Java Server
- Jupyter Notebook

The database is built by Docker and PostgreSQL. The ***Dockerfile*** under the ***<root>/db/*** directory contains the script for building a PostgreSQL image and implementing the ***<root>/db/init.sql*** SQL query for database initialisation. Then we use docker-compose to generate the container of the image and assign the port to the local container. The ***<root>/docker-compose.yml*** file owns the configuration. Also, there is a ***<root>/.env-dev*** file containing the environment variables of the Docker container. The database will set up the database name, database user and the database password through referencing these variable settings. This local database is used for temporary development purposes. There are two major schemas which are input and output. All the input and transformed data will be stored into the input schema. The output schema is used to store the output data once we have productionised the analysis. The website or the prgramme being built in the future can then access the data and visualise them onto the website.

We use Maven for managing our Java project. It facilitates the process of acquiring libraries and managing the dependencies between projects or services. The ***<root>/parent/pom.xml*** manages the Java programme. It controls the versions of dependencies and the project hierarchy. The main services are in ***<root>/analysis/***. The ***<root>/analysis/pom.xml*** contains the dependencies of the analysis server. We use Spring Framework for development and database access in the future. The ***application.properties*** in ***<root>/analysis/src/main/resources/*** holds the database and Spark configuration. Spring Boot is used for preparation of productionisation with websites. It can help us quickly build up a single page web application in the future.

There are mainly 2 services currently, which are ETL and analysis. ***Application.java*** is the entry point. If we keep running the server, it will run the data updates and analysis routinely every day. ***TaskRunner.java*** is responsible for arrangement of all the tasks. The tasks and classes for ETL are in the ***com.kenlu.crypto.extraction*** package. ***com.kenlu.crypto.domain*** contains the data models and enums. Analysis are implemented in the ***com.kenlu.crypto.analysis*** package. However, currently, we only implement the ETL service because the productionisation is not completed yet.

All the experiments and visualisations are writen in Python and stored under ***<root>/experiment/***. The 2 main scripts are ***var_crypto.ipynb*** and ***var_stock.ipynb*** with the format of Jupyter Notebook. The scripts import the data either from the local database or the csv files in case the database connection or the API of cryptos and stocks collapse. The result of these experiments will be transformed to a productionised version in Java once it is ready.

1.3.3 Report Structure

This report is mainly composed by several parts including an introduction, a background research, analysis and a conclusion. We use LaTeX for facilitation of the writing process. ***<root>/report/*** contains all the LaTeX files and resources for compilation of the report. ***Report.tex*** includes the main scripts. ***<root>/report/resources/*** holds the images for the report. After compilation, a ***Report.pdf*** file will be built for a complete presentation.

2 Background Research

2.1 Value at Risk

Value at Risk (VaR) is a method used to evaluate the risk of a financial portfolio. It summarises the degree of investment risk with a single number. Financial institutions have been widely using VaR as a metric to decide how much capital they should keep to bear with risks.

Financial specialists are usually interested in the statement which indicates a specific criterion that helps there decision making. As a result, VaR is normally interpreted as follow:

"I have X% of confidence that the loss of our investment will not be more than $V in the next N days."

where:

- X is the confidence level
- V is the VaR of the portfolio
- N is the time horizon

A proper VaR states that there will only be a (100 - X)% of chance that our loss of investment will be exceeded. For example, given V is $100,000, X is 95, and N is 10, there will only be 5% of scenarios that the loss of our portfolio will be more than $100,000 in the next 10 days based on our estimation.

An N-day VaR(2.1) is usually calculated on the basis of 1-day VaR:

$$\text{N-day VaR} = \text{1-day VaR} \times \sqrt{N} \tag{2.1}$$

There are several approaches for calculating VaRs. In this project, as our objective is to see if clustering algorithms can improve the VaR estimation, we will only talk about the simplest one - *historical simulation approach*[20].

2.1.1 Historical Simulation Approach

Historical simulation approach looks at historical data and simulate the past scenarios that might happen in the future. Then it chooses the worst (100 - X)% of scenarios as the worst cases of the loss of our investment where the $(100 - X)_{th}$ percentile is our VaR.

For example, we want to estimate the 1-day VaR with 95% of confidence for tomorrow, and there are 500 days of historical data. The data for each day is the return/loss from the previous day, which means there are 500 scenarios(2.2) that might happen tomorrow. We can then multiply the value

of investment we are holding today by the 5_{th} least return rate (most loss rate) and gain the loss that might happen tomorrow as our 1-day VaR[20].

$$\text{Value under } i_{th} \text{ scenario } = v_n \frac{v_i}{v_{i-1}} \tag{2.2}$$

where:

- v_n is the value of today
- v_i is the value of Day i

2.2 Correlation Analysis

In statistics, covariance(2.3) defines the variability between two individual attributes, which means the level of influence from one feature to another. The numbers correspond to similarity/dissimilarity of the two variables. Positive numbers represent a similar behaviour between them, and vice versa[15].

$$\text{cov}(X, Y) = \text{E}[(X - \mu_X)(Y - \mu_Y)] \tag{2.3}$$

where:

- E is the expectation
- X, Y are vectors of all the samples
- cov is the covariance
- μ_X is the mean of X
- μ_Y is the mean of Y

However, if we want to measure the strength of the linear relationship in between, covariance is not enough. We also need to consider the variance in each feature to tell whether the linear relationship is strong. In this case, the correlation was introduced. Correlation is a normalised form of covariance. It restricts the numbers to a certain range which shows how strong the relationship is. The most commonly used correlation coefficient is Pearson correlation coefficient(2.4). it is calculated by considering the standard deviation of both groups. This can ensure that dispersion of either attribute does not interfere our identification on the strength of mutual linear relationships[18].

$$\rho_{X,Y} = \frac{\text{cov}(X, Y)}{\sigma_X \sigma_Y} \tag{2.4}$$

where:

- ρ is the Pearson correlation coefficient
- X, Y are vectors of all the samples

- cov is the covariance
- σ_X is the standard deviation of X
- σ_Y is the standard deviation of Y

The value of Pearson correlation coefficient is always between -1 and +1. A positive number means a positive linear correlation, and a negative number means a negative linear correlation. The closer the number towards the extremes, the stronger the relationship is. If the number is 0, it means there is no linear correlation among the pair(2.5).

$$\text{relationship} = \begin{cases} \text{total positive linear correlation} & \text{if } \rho = 1 \\ \text{positive linear correlation} & \text{if } \rho > 0 \\ \text{no linear correlation} & \text{if } \rho = 0 \\ \text{negative linear correlation} & \text{if } \rho < 0 \\ \text{total negative linear correlation} & \text{if } \rho = -1 \end{cases} \tag{2.5}$$

2.2.1 Correlation Matrix

Given a set of data with multiple attributes, we may want to tell people how these attributes interact with each other. In addition, the result of analysis, especially in a simple regression, may not be reasonable when those features are highly dependent.

To achieve this, we can create a matrix which contains all the correlation coefficient calculated from the expanded equation(2.6) with a set of given samples.

$$\rho_{xy} = \frac{\sum_{i=1}^{n}(x_i - \bar{x})(y_i - \bar{y})}{\sqrt{\sum_{i=1}^{n}(x_i - \bar{x})^2}\sqrt{\sum_{i=1}^{n}(y_i - \bar{y})^2}} \tag{2.6}$$

where:

- ρ is the Pearson correlation coefficient
- n is the sample size
- x_i, y_i are the single samples indexed with i
- $\bar{x} = \frac{1}{n}\sum_{i=1}^{n} x_i$ (the sample mean)
- $\bar{y} = \frac{1}{n}\sum_{i=1}^{n} y_i$ (the sample mean)

A correlation matrix is a symmetirc matrix to its main diagonal. The values on the main diagonal always equal to 1 because the attributes are fully dependent on themselves. Table 1 gives an example of how a correlation matrix looks like.

Features	f1	f2	f3	f4	f5
f1	1	0.74	-0.38	0.12	0.43
f2	0.74	1	0.26	0.88	-0.57
f3	-0.38	0.26	1	0.61	0.59
f4	0.12	0.88	0.61	1	-0.22
f5	0.43	-0.57	0.59	-0.22	1

Table 1: Correlation Matrix

2.3 Principal Component Analysis

It is always challenging to analyse a dataset with high-dimensional data points. Due to the curse of dimensionality, which was discovered by Richard Ernest Bellman in 1957, higher-dimensional space increases the difficulties of analysing and organising data exponentially[22]. Especially in machine learning, given a certain number of samples, the accuracy of predictions on these samples will increase followed by the rising dimensions to a peak but then gradually drop. This is known as Hughes phenomenon[19]. Figure 1 shows how the dimensionality influence the accuracy of predictions.

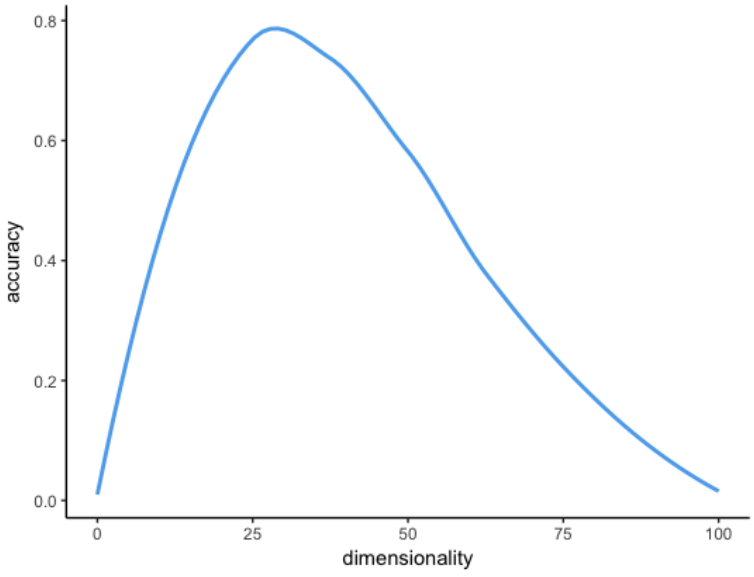


Figure 1: Hughes phenomenon

In order to reduce the dimensionality, there are two approaches can be

implemented:

- Feature Selection: To select a subset that is more informative or relevant among all the attributes[25].
- Feature Extraction: To generate new features from the initial attributes of existing data[16].

The reasons and benefits of executing dimensionality reduction can be summarised as follow:

1. Computational efficiency: Fewer features mean less computation on dissimilarity between pairs of data points and lower arithmetic complexity. It also implies less storage usage as the variables in each sample decrease.
2. Statistical generalisation: By removing noise or irrelevant information from the inputs for building models, the prediction rules can be more general among the datasets.
3. Better explanation: Visualising a lower-dimensional space is much easier. We can effortlessly illustrate the structure of data when the dimension is lower than 3. Higher-dimensional space will be more challenging to visualise, explain and comprehend.

Principal component analysis(PCA) is a method for feature extraction. It projects features onto a lower-dimensional space. A traditional PCA is a kind of single representation approach as opposed to classification on revealing underlying information in a lower-dimensional space with a linear function.

In practice, an optimal mapping function is usually non-linear. In order to fit the data in a non-linear way, we can apply a kernel method on top of the traditional PCA, and this is called kernel PCA. It performs a linear PCA mapping in a higher dimensional kernel Hilbert space to provide a better classification. The kernel can be a polynomial function, a radial function or other functions[23]. However, in this project, we will assume that the relationship between the dimensions (cryptocurrencies) are linear and will only use a standard linear PCA to perform the dimensionality reduction.

2.3.1 Principal Components

The new features derived are called principal components (PCs). They represent new orthogonal axes in an order based on the amount of information it contains.

Imagine that we have a dataset X with data points $\{x_1...x_n\}$. Each point is a D-dimensional vector. The goal is to project the data points onto a M-dimensional space where $M < D$ and maximise the variance of data after projection. The dimensions will be the top M principal components which represent the most informative new features. M is generally determined by the following factors:

- Informativity: How much information of the original features has been involved?
- Interpretability: Are we able to visualise or make useful attribution among the principal components?
- Computational efficiency: The curse of dimensionality. A large number of dimensions will decrease the speed of computation in algorithms.

To obtain the first principal component, the λ_1 in equation (2.7) should be maximised. u_1 is then known as the first principal component. $u_1^T u_1$ should be equal to 1. We can then define other principal components by choosing new directions of the projection which disregards the elements that are already considered.

$$u_1^T S u_1 = \lambda_1 \tag{2.7}$$

where:

- λ_1 is the eigenvalue
- u is the eigenvector having the largest eigenvalue λ_1
- S is the data covariance

2.4 Clustering

Clustering, cluster analysis or data segmentation is a non-parametric algorithm in the subtree of unsupervised learning. It is used to separate data into different groups using their dissimilarities (similarities) or possible distributions. Unlike supervised learning, this type of learning algorithms does not have any indicator for assessing the quality of results, and this means that it does not have any meaning or objective itself. Instead, it discovers the distribution of data and uses the definition given by people who have the specific domain knowledge. By giving the rules for partitioning data self-defined meanings, useful information can be obtained and utilised in different domains[25].

In general, clustering can be defined into two types, parametric and non-parametric. A parametric clustering groups the clusters with a a assumed density function which is usually a Gaussian, while a non-parametric one does not have any assumed distribution, it only aims on finding natural groupings within the givien dataset.

K-means clustering and hierarchical clustering are two of the most popular methods in the non-parametric cluster analysis. In K-means clustering, we specify the number of groups we want to classify into. In contrast, hierarchical clustering does not have an initial number of clusters that we want for the result. It instead shows all the possible clusters into a tree structure and allows us to choose the number of clusters we want at the end. In this project, we will only focus on the K-means algorithm in non-parametric methods.

2.4.1 K-means Clustering

K-means clustering is a intuitive approach which allows us to separate data points into distinct groups. To implement this, we first need to specify an initial number of clusters - K, and then randomly assign a number (cluster) from 1 to K to each object (data point). In this case, the clusters will have two features[17]:

1. $C_1 \cup C_2 \cup ... \cup C_K = \{O_1, O_2, ..., O_n\}$
2. $C_k \cap C_{k'} = \emptyset, \quad \text{for} \wedge k \neq k'$

where:

- C_k is the kth cluster
- O_n is the nth object

These properties mean that each single object will be in exactly one cluster and the clusters does not overlap. After this initial setting, we want to optimise(2.8) the grouping because the previous assignment is just a random initialisation. We want to make sure that the objects are concentrated which means the data point fit the best in the assigned cluster.

$$\underset{\mathbf{C_1},...,\mathbf{C_K}}{\text{minimize}} \left\{ \sum_{k=1}^{K} \frac{1}{|C_k|} \sum_{i,i' \in C_k} \sum_{j=1}^{p} (x_{ij} - x_{i'j}) \right\} \tag{2.8}$$

where:

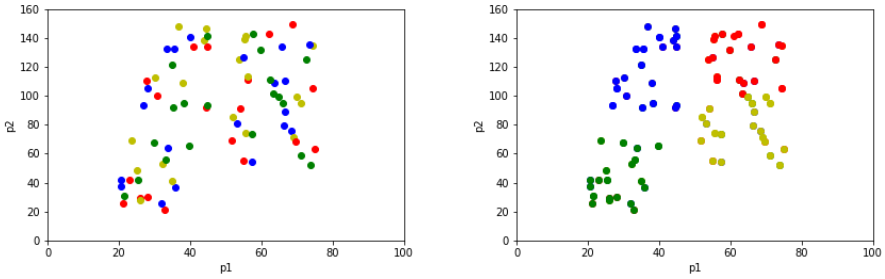
- C_k is the kth cluster
- x_{ij} is the jth attribute of the ith object

To fulfil the condition above which is (2.8), we can simplify the apporach as below to classify our data points into the multiple clusters[17]:

1. Randomly assign an initial number from 1 to K to each observation.
2. Iterate over the following steps until the assigned cluster of each observation stops changing:
   (a) for i in range(1, K):
   Compute the centroid of each cluster which is the mean of vectors with the same k (cluster) assigned.
   (b) Calculate the distance (usually Euclidean distance) between each object and each of the clusters.
   (c) Assign the nearest k (cluster) to each observation.

Figure 2 shows the difference after an optimisation of the cluster assignment. The colors indicate different clusters, and the groups are separated after the implementation of K-means clustering.

$$K = 4$$



(a) Before Iteration (b) After Iteration

Figure 2: K-means Clustering

2.4.2 Dynamic Time Warping Distance

Euclidean distance is a metric for evaluating the distance between sequences. It is useful as it produces non-negative result and has linear time complexity. However, it is restricted by the alignment of sequences, which means that the distance can only be calculated if the two sequences are in the same length. If we want to learn how similar the shapes of two objects/lines in time series are, this is not a good method to be used.

Dynamic time warping (DTW) is a popular shape-based algorithm based on dynamic programming in time series analysis. DTW is a time distortion method which adjusts the corresponding elements in two vectors and finds the minimal distance among the neighbours. The accumulative value will then become the final distance in between. It is widely used in temporal sequence data such as speech recognition[14].

Given two vectors t and r, and the lengths are M and N respectively, DTW aims on finding a path to minimise the distance between t and r.

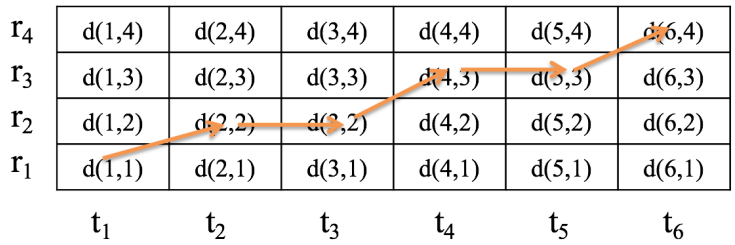


Figure 3: DTW Optimal Path

The calculation of the DTW distance needs to satisfy the following conditions[26]:

1. The first element from t must match the first element from r.
2. The last element from t must match the last element from r.
3. Given $d(i,j)$ is the distance of a point on the optimal path. The points that connect into $d(i,j)$ can only be $d(i-1,j)$, $d(i-1,j-1)$ and $d(i,j-1)$.
4. Given any element in t, there must be at least one corresponding element in r, and vice versa.

We can use a recursion(2.8) to find out the minimal accumulative distance $A(M,N)$ which is our goal of the DTW distance[26].

$$A(i,j) = d(i,j) + min \left\{ \begin{array}{c} \text{A(i-1,j)} \\ \text{A(i-1,j-1)} \\ \text{A(i,j-1)} \end{array} \right\} \tag{2.8}$$

where:

- $D(i,j)$ is the distance between t_i and r_j
- $A(i,j)$ is the accumulative distance from the starting point to (i,j)

3 Analysis of Cryptocurrencies

The data of cryptocurrencies is obtained from the RESTful API `https://min-api.cryptocompare.com/`[4] which is free for non-profit purposes. We store the data into our local Docker based PostgreSQL database for development and convenience in case the internet or the API are unstable. Our Java programme will then update the latest data on a daily basis.

This analysis will use the historical daily OHLC (Open price, High price, Low price, Close price) data of cryptocurrencies as the initial input obtained from the API and will only consider 29 cryptocurrencies that have complete data from 1st January 2016 to 31st July 2018. The reason why we choose the number 29 is because we want to compare the performance on cryptocurrencies with that on stocks and we will choose Dow Jones 30 as our counterpart. We want to make them have the same size of products and DWDP (DowDuPont) in Dow Jones does not have the data before 1st September 2017.

3.1 Data Preprocessing

The original OHLC data of cryptocurrencies contains the open price, highest price, lowest price and close price of different cryptocurrencies for each day. The unit of the price is USD. There is no specific trading hours for cryptocurrencies. The metric the API used to split the values into days is based on 00:00 GMT time. For simplicity, we will use symbols to represent cryptocurrencies. Table 2 shows the corresponding cryptocurrency of each symbol.

Symbol	Crypto	Symbol	Crypto	Symbol	Crypto
AEON	Aeon	BAY	BitBay	BLOCK	Blocknet
BTC	Bitcoin	BTS	BitShares	BURST	Burst
CRW	Crown	DASH	Dash	DGB	DigiByte
DOGE	Dogecoin	EMC2	Einsteinium	ETH	Ethereum
FCT	Factom	FTC	Feathercoin	GAME	GameCredits
GRS	Groestlcoin	LTC	Litecoin	MONA	MonaCoin
NAV	NavCoin	NLG	Gulden	POT	PotCoin
PPC	Peercoin	RDD	ReddCoin	SYS	Syscoin
VIA	Viacoin	VTC	Vertcoin	XCP	Counterparty
XMR	Monero	XRP	Ripple		

Table 2: Symbols of Cryptocurrencies

Before we start our analysis, we need to preprocess the data to ensure

that the data is clean enough and in the format which can be used as the input of the analysis. The historical daily OHLC data requested from the API is on the crypto by crypto basis. In this case, we extract all the data, combine them into the format of Table 3 and store them into the database. Overall, there is a huge growth among the price of most of the cryptocurrencies.

Date	AEON_open	AEON_high	AEON_low	...	XRP_low	XRP_close
2016-01-01	0.01454	0.0217	0.01316	...	0.005132	0.0055
2016-01-02	0.01498	0.01734	0.01388	...	0.005	0.005125
2016-01-03	0.01378	0.01536	0.01379	...	0.005	0.0052
2016-01-04	0.01391	0.0143	0.01218	...	0.0051	0.0051
...	...	...	...	...	...	...
2018-07-28	1.58	1.75	1.58	...	0.4482	0.4576
2018-07-29	1.67	1.78	1.58	...	0.4503	0.4529
2018-07-30	1.68	1.69	1.57	...	0.4346	0.4458
2018-07-31	1.55	1.55	1.42	...	0.4272	0.4351

Table 3: Historical OHLC of Cryptocurrencies

Our objective is to evaluate the risk based on price volatility which is the daily price changes (g) or daily rate of return (r). We can calculate the daily price changes through dividing the close price by the open price. The rate of return (r) is then equal to $1 - g$. Table 4 shows the daily price changes after transformation from the historical OHLC dataset.

Date	AEON	BAY	BLOCK	...	XCP	XMR	XRP
2016-01-01	0.030949	-0.296502	-0.265306	...	-0.038405	0.108141	0.056676
2016-01-02	-0.073431	0	0.180451	...	-0.033988	-0.153010	-0.001364
2016-01-03	0.002903	-0.085601	-0.036252	...	-0.025285	0.059783	0.014634
2016-01-04	-0.097052	0.124730	0.009491	...	0.048847	-0.035897	-0.019231
...	...	...	...	...	...	...	...
2018-07-28	0.056962	-0.024547	0.012285	...	-0.005362	-0.000714	0.006157
2018-07-29	0.011976	0.002037	-0.040519	...	-0.028340	-0.030859	-0.010487
2018-07-30	-0.023810	-0.016599	-0.078947	...	0	-0.025429	-0.015677
2018-07-31	-0.064516	-0.059292	-0.094542	...	-0.098820	-0.076615	-0.024002

Table 4: Historical Daily Returns of Cryptocurrencies

In order to test how good the risk estimation is, we will use sliding window mehtod to split the whole dataset into multiple parts. We use 100 as the size of the sliding window in this analysis. This kind of method will conduct a prediction or estimation on each part of dataset. For example, there are 500 datapoints, and the size of the window is 200. We can then estimate the 201_{th} VaR using the first 200 datapoints and use the 2_{th} to 201_{th} datapoints to estimate the 202_{th} value and so on. As a result, there

will be 300 VaRs that can be predicted.

3.2 Risk Estimation

We will use VaR to evaluate the risk of cryptocurrencies since it provides a simple way of estimating the degree of our loss under risks. In here, we regard each cryptocurrency as a single investment instead of generating a combined portfolio because we want to construct a metric to compare the performance improvement after a risk diversification which will be conducted later. As a result, there will be 29 VaRs corresponding to the 29 cryptocurrencies.

3.2.1 Historical Simulation of VaR

A 95% confidence level of 1-day VaR estimation will be used in this project, which means we want to know how bad our loss of the investment can become for the next day in a 95% of confidence. A historical simulation method will be applied to the sliding windows in here to simulate the scenarios that have happened and find out the worst 5% of the experience. The 5_{th} percentile of the whole scenarios will then become our VaR.

Crypto	2016-04-10	2016-04-11	2016-04-12	...	2018-07-30	2018-07-31
AEON	-0.17212	-0.17212	-0.17212	...	-0.06261	-0.06261
BAY	-0.16117	-0.15626	-0.15626	...	-0.06569	-0.06569
BLOCK	-0.22696	-0.19087	-0.19087	...	-0.08441	-0.08441
...	...	...	...	...	...	...
XCP	-0.13631	-0.13631	-0.13631	...	-0.07843	-0.07843
XMR	-0.15443	-0.15443	-0.15443	...	-0.09825	-0.09825
XRP	-0.08568	-0.08568	-0.08568	...	-0.07522	-0.07522

Table 5: VaRs of Cryptocurrencies Before Clustering

Table 5 states the 95% 1-day VaRs of each cryptocurrency. The values are the daily returns. Negative numbers represent loss in percentage. Normally, the VaRs stay in the same values for several days because the historical data used for the estimation only changes slightly in every steps. When we estimate a new VaR, we only move the window one-day forward. This will not cause the 5_{th} percentile of the returns change significantly.

3.3 Risk Diversification

There are many ways of diversifying our investments, either through observation of direct correlations between daily volatilities of individual products

or discovery of long-term similarity between volatility movements of the products. We will use both methods to help our construction of trading strategies because short-term and long-term hedging strategies are equally important for investors.

3.3.1 Product Correlations

The Pearson correlation coefficient is useful and straight forward in here for researching the direct relationships between two cryptocurrencies. It summarises the relationships of pairwise observations between two cryptocurrencies and restricts the result in the range between -1 and +1, and this allows us to observe the strength of connections on those cryptocurrencies based on the daily returns. A larger correlation coefficient means there exists a stronger correlation between their daily returns. This analysis aims on helping the construction of trading strategies. The whole dataset should be involved to generate a well-diversified hedging strategy.

We can generate a correlation matrix which states the pairwise correlations of all the cryptocurrencies. Next, a visualisation of the matrix helps us to analyse and observe the degree of correlations easier. Figure 4 illustrates the result of a visualisation. The colour bar explains the meaning of different colours. Lighter colours represent a stronger correlation, and vice versa.

There is a diagonal on the chart which represents the correlations between all the cryptocurrencies and themselves. These blocks are all yellow which means the correlations are total positive linear correlation. And the chart is symmetric against the diagonal. We can look at either part to analyse the values.

BTC (Bitcoin) has some interesting features. It has extreme values against other cryptocurrencies. It has a strong correlation at around 0.7 with LTC (Litecoin) and a negative correlation at approximately -0.1 with NLG (Gulden) and MONA (MonaCoin). ETH (Ethereum) and XMR (Monero) are also strongly correlated to BTC (Bitcoin) at arround 0.6 of correlation coefficient. When it comes to XMR (Monero), it is dependent on several cryptocurrencies such as AEON (Aeon), ETH (Ethereum), FCT (Factom) and LTC (Litecoin), and this means the daily changes or returns of XMR (Monero) is similar to those cryptocurrencies.

In contrast, some of the cryptocurrencies seem to be independent and have less influence from other products. BTCD (BitcoinDark) is one of these as the colours on the bar are all close to dark blue whose correlation is around 0, and that means there is no obvious correlation in between. BAY

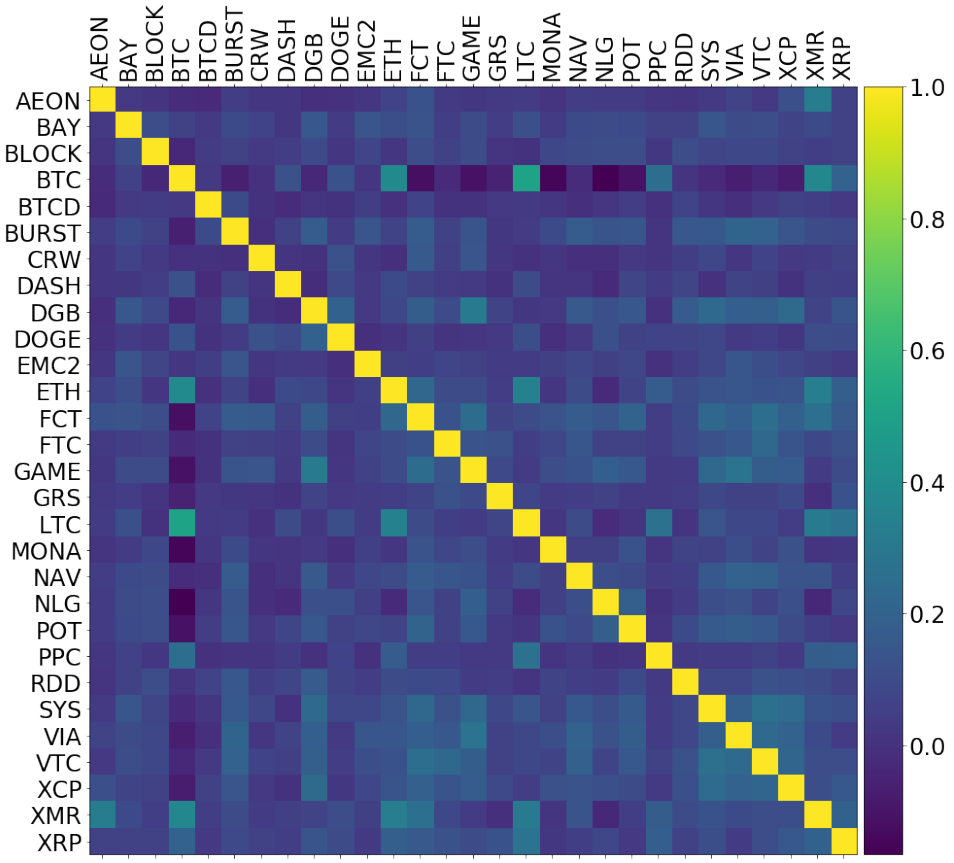


Figure 4: Correlation Matrix of Cryptocurrencies

(BitBay), BLOCK (Blocknet), DOGE (Dogecoin) and MONA (MonaCoin) have this kind of phenomenon as well.

When constructing an investment portfolio, the less dependent products should be considered because strongly correlated investments might result in great return, but it might cause enormous loss simultaneously. An unstable investment behaviour should be prevented, especially our purpose is hedging.

3.3.2 K-means Clustering on Price Movement

In the context of clustering, we use all the data without any split because we want to make the best use of every datapoints to illustrate the pattern of the attributes.

Before starting our clustering analysis, there are some prerequisites to be

implemented. First, we need to transpose our data to make the cryptocurrencies become observations/objects and see the time-series as the features. A clustering can then group the cryptocurrencies together based on the similarity/dissimilarity of their daily returns along the timeline. After the trnaspose, a dimensionality reduction should be executed as there are too many dimensions which are dates among the transposed dataset, and this will cause a computational difficulty on K-means clustering.

We first take the daily returns dataset as our input dataset. Table 6 illustrates how the data looks like after a transpose. It is now a 29 by 943 table which used to be 943 by 29. The rows become the cryptocurrencies and the columns are dates from 2016-01-01 to 2018-07-31. The columns are the attributes we will use for our clustering. We can now use the PCA (Principal Component Analysis) to reduce the number of dimensions.

	2016-01-01	2016-01-02	2016-01-03	...	2018-07-30	2018-07-31
AEON	0.030949	-0.073431	0.002903	...	-0.023810	-0.064516
BAY	-0.296502	0	-0.085601	...	-0.016599	-0.059292
BLOCK	-0.265306	0.180451	-0.078947	...	0.059783	-0.094542
...	...	...	...	...	...	...
XCP	-0.038405	-0.033988	-0.025285	...	0	-0.098820
XMR	0.108141	-0.153010	0.059783	...	-0.025429	-0.076615
XRP	0.056676	-0.001364	0.014634	...	-0.015677	-0.024002

Table 6: Transposed Daily Returns of Cryptocurrencies

The number of Principal Components can be decided by the amount of variance that has been explained. Figure 5 shows the percentage of explained variance with different number of PCs. We choose the number 26 as our number of PCs because it explains around 99% of the attributes and our objective is to make the clustering computationally efficient without losing too much information. With respect to visualisation, because we only want to know the similarity between shapes of the volatilities, we can illustrate line charts of the cryptocurrencies' PCs even on a higher dimensional space.

After the preparation, K-means clustering can be implemented. The number of clusters, K, we select is 8 because of minimisation of the cost in each cluster and properly distribute the cryptos into clusters. The distance metric we are using is DTW (Dynamic Time Warping) because we want to know how similar the activities between cryptocurrencies are instead of the direct similarity which is restricted by strict time criterion. DTW is computationally more expensive than Euclidean distance, but it provides a more flexible way of comparing the distance between two objects.

The cryptocurrencies are distributed into 8 clusters after multiple itera-

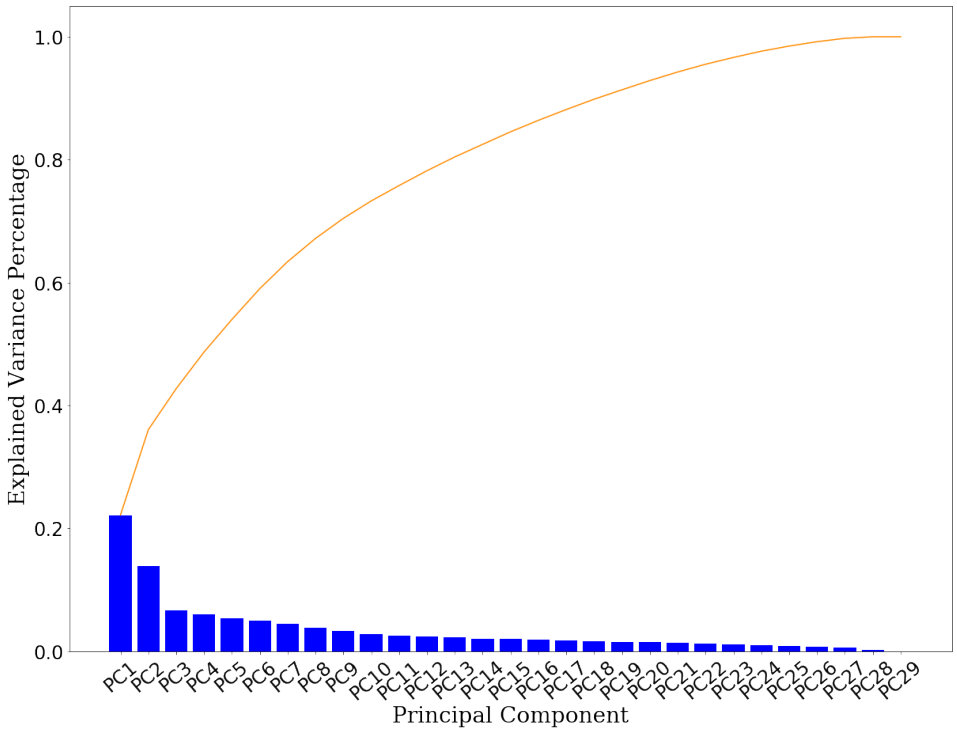


Figure 5: Cumulative Explained Variance Percentage of PCs on Cryptocurrencies

tion until the cost(2.8) is minimised. The cluster contents are described as follow:

- *Cluster 1*: DGB, MONA, POT, VIA, VTC
- *Cluster 2*: BTC, BTS, ETH, FCT, LTC, NLG, XMR, XRP
- *Cluster 3*: AEON, BLOCK, GRS
- *Cluster 4*: CRW
- *Cluster 5*: DOGE
- *Cluster 6*: BAY, BURST, FTC, GAME, PPC, SYS, XCP
- *Cluster 7*: EMC2, NAV
- *Cluster 8*: DASH, RDD

There is a skewness among the clusters. *Cluster 1*, *Cluster 2* and *Cluster 6* have more components than other clusters, and *Cluster 2* contains some of the most popular cryptocurrencies of the market. Apart from this, other cryptocurrencies seem to scatter around clusters.

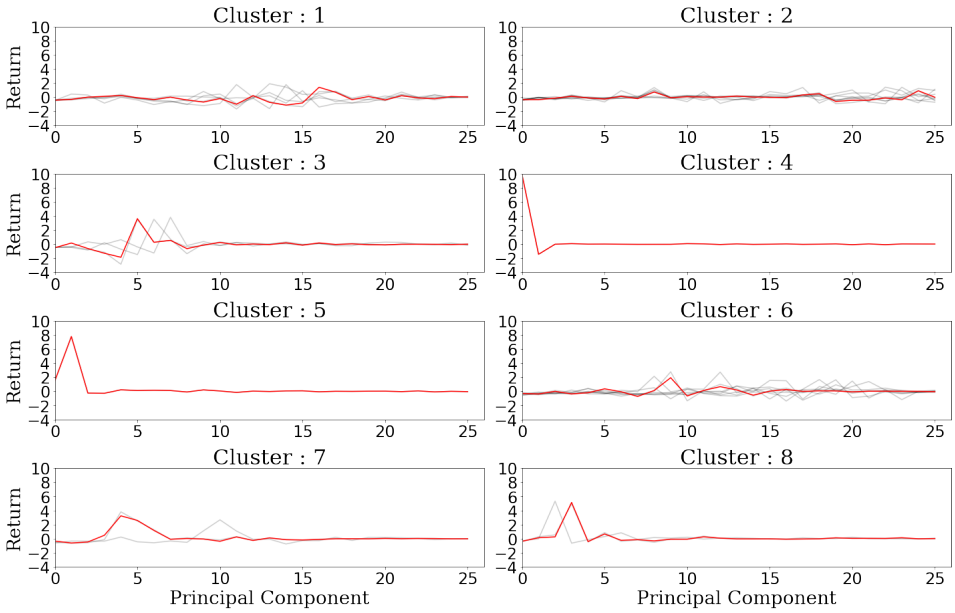


Figure 6: Clusters of Cryptocurrencies

We can then look at Figure 6 which shows the result of our K-means clustering. The red lines represent the centres or the means of elements in each cluster, and the grey lines are the cryptocurrencies. Each cluster has a distinct shape of the central line. Those elements which are in the same cluster should have similar shapes of lines. The objects in *Cluster 2* are more stable than other those in other clusters. *Cluster 1*, *Cluster 3*, *Cluster 6*, *Cluster 7* and *Cluster 8* are more fluctuated. *Cluster 4* and *Cluster 5* have strange activities and have some extreme phenomena.

3.4 Performance

3.4.1 Original Accuracy and p-value

The accuracy of a 1-day VaR can be tested by comparing the VaRs with the actual returns of each day. If the actural return is smaller than the VaR, we can say that the VaR is not accurate and mark it as an incorrect estimation

of VaR. By using the sliding window method, we can test how many times the daily returns exceed the 1-day VaRs and calculate the percentage of correct estimation. Table 7 shows the VaR accuracy of each cryptocurrency.

Symbol	Accuracy	Symbol	Accuracy	Symbol	Accuracy
AEON	95.02%	BAY	94.9%	BLOCK	94.31%
BTC	93.48%	BTCD	94.66%	BURST	94.31%
CRW	94.78%	DASH	94.07%	DGB	94.54%
DOGE	93.59%	EMC2	94.9%	ETH	93.12%
FCT	94.31%	FTC	94.54%	GAME	91.93%
GRS	94.42%	LTC	93.71%	MONA	93.59%
NAV	94.07%	NLG	94.54%	POT	94.31%
PPC	93%	RDD	94.42%	SYS	93.83%
VIA	94.78%	VTC	93.95%	XCP	94.31%
XMR	94.54%	XRP	94.42%		

Table 7: VaR Estimation Accuracies of Cryptocurrencies Before Clustering

Most of the accuracies are lower than 95% which is the confidence level of our VaR estimation. Only that of AEON (Aeon) with 95.02% is slightly higher than the 95% level of confidence. That means our VaR estimation is not accurate. It might be because the VaR estimation did not capture some of the financial events. Overall, the average of all the accuracies is around 94.15% over the 29 cryptocurrencies and 843 days of VaR estimation.

To examine if the VaR estimation is just accidentally incorrect, we can construct a null hypothesis of how many times the VaRs are exceeded. The threshold of a 95% 1-day VaR is around 42 days out of 843 days, which means there should be less than 42 days of daily returns breaking the VaR estimation. We can then assume our null hypothesis as follow:

- H_0: Number of VaRs being exceeded < 42
- H_a: Number of VaRs being exceeded ≥ 42

The null hypothesis means that we believe the number of VaRs that are exceeded should be lower than 42, and the alternative hypothesis is that the number of VaRs being exceeded is greater or equal to 42. We can now calculate the p-value of this null hypotheis to test if the evidence of our alternative hypothesis is significant enough to reject the null hypothesis. The p-value after calculation is approximately 3.45%. In here, we use the significance level of 5% which also represents a confidence level of 95%. The p-value is smaller than the significance level. As a result, we can reject

the null hypothesis and say the evidence of that the number of VaRs being exceeded is greater or equal to 42 times is significant enough to say our assumption on the null hypothesis is wrong at the 95% level of confidence. This means the estimation of VaRs in here is not accurate enough.

3.4.2 Improvement After Risk Diversification

Since we have distributed the cryptocurrencies into different clusters, we can try to use the result to improve the estimation of VaRs. Instead of calculating the VaRs of each cryptocurrency, we regard the cryptocurrencies in the same cluster as a huge investment and estimate the 95% 1-day VaRs among all the scenarios. This can be calculated by choosing the 5_{th} percentile of all the returns of scenarios. As a result, the cryptocurrencies in the same cluster will have the same VaRs.

Crypto	2016-04-10	2016-04-11	2016-04-12	...	2018-07-30	2018-07-31
AEON	-0.18918	-0.17296	-0.17296	...	-0.08250	-0.08250
BAY	-0.12574	-0.11807	-0.12574	...	-0.08953	-0.09228
BLOCK	-0.18918	-0.17296	-0.17296	...	-0.08250	-0.08250
...	...	...	...	...	...	...
XCP	-0.12574	-0.11807	-0.12574	...	-0.08953	-0.09228
XMR	-0.10639	-0.11001	-0.10913	...	-0.08023	-0.08023
XRP	-0.10639	-0.11001	-0.10913	...	-0.08023	-0.08023

Table 8: VaRs of Cryptocurrencies After Clustering

Table 8 illustrates the result of VaR estimation after an implementation of K-means clustering. Some of the values become smaller which means the predicted loss is larger. This is because the cryptocurrency has bigger risks in its cluster at that time under our risk diversification. Bigger values mean that the risk of the product has been overestimated in the previous estimation at that point. We have adjusted the risks based on the result of clustering.

Similar to metric of calculating the VaR accuracies before the clustering, we can test the performance of our VaR estimation after an implementation of clustering. Table 9 shows the accuracies after a risk diversification. Red numbers indicate the cryptocurrencies that have an improvement on the VaR estimation in comparison with the accuracies before any actions. Over 50% of the cryptocurrencies benefit from the risk diversification and many of them are the most popular products such as BTC (Bitcoin), ETH (Ethereum), LTC (Litecoin) and XRP (Ripple) although it results in some

decrease on the other cryptocurrencies. In addition, the average accuracy is about 94.91% which is higher than the average before clustering at 94.15%.

Symbol	Accuracy	Symbol	Accuracy	Symbol	Accuracy
AEON	96.09%	BAY	93.83%	BLOCK	95.14%
BTC	97.98%	BTCD	93.71%	BURST	94.78%
CRW	94.78%	DASH	98.34%	DGB	95.61%
DOGE	93.59%	EMC2	94.54%	ETH	96.44%
FCT	93.12%	FTC	94.31%	GAME	95.02%
GRS	94.66%	LTC	96.92%	MONA	96.56%
NAV	95.14%	NLG	93.24%	POT	94.9%
PPC	96.44%	RDD	91.22%	SYS	94.31%
VIA	93.95%	VTC	94.19%	XCP	94.66%
XMR	94.36%	XRP	95.49%		

Table 9: VaR Estimation Accuracies of Cryptocurrencies After Clustering

We then use the same null hypothesis as the previous hypothesis testing to test if the VaR estimation is good enough to retain our null hypothesis which assumes that the number of VaRs being exceeded by the corresponding returns is smaller than 42. After the calculation, the p-value is around 37.93% which is not significant enough to reject our null hypothesis at the 95% level of confidence. That means our VaR estimation after the clustering is possiblely a good fit to the data. This result gives us a good indication that K-means clustering might be able to improve the traditional VaR estimation.

4 Analysis of Stocks

Since the risk diversification helps our estimation of risks on cryptocurrencies, it might be also applicable to stock data. The stock data is retrieved from the RESTful API `https://iextrading.com/developer/`[9] which is also an open source. As what has been conducted on cryptocurrencies, we extract the data to our local database and use the Java programme to implement daily ETL.

The analysis of stocks will use the same criterion and standard as that of cryptocurrencies, which means the input dataset is also the historical dialy OHLC. It contains the data from 1st January 2016 to 31st July 2018 of 29 stocks in Dow Jones 30, and this excludes DWDP (DowDuPont) because there is no complete data for DWDP (DowDuPont) over the period.

4.1 Data Preprocessing

The data for stocks is also using USD as the unit of prices. However, there are trading hours for stocks. The stocks of Dow Jones are listed under NYSE (New York Stock Exchange) or the Nasdaq Stock Market, and the trading hours for both exchanges are 09:30 to 16:00 EST from Monday to Friday except public holidays. As a result, there is no data for weekends. Table 10 lists the full company name of each stcok.

Symbol	Stock	Symbol	Stock
AAPL	Apple	AXP	American Express
BA	Boeing	CAT	Caterpillar
CSCO	Cisco Systems	CVX	Chevron
DIS	Walt Disney	GS	Goldman Sachs
HD	The Home Depot	IBM	IBM
INTC	Intel	JNJ	Johnson & Johnson
JPM	JPMorgan Chase	KO	Coca-Cola
MCD	McDonald's	MMM	3M
MRK	Merck & Company	MSFT	Microsoft
NKE	Nike	PFE	Pfizer
PG	Procter & Gamble	TRV	Travelers
UNH	UnitedHealth Group	UTX	United Technologies
V	Visa	VZ	Verizon
WBA	Walgreens Boots Alliance	WMT	Walmart
XOM	ExxonMobil		

Table 10: Symbols of Stocks

Similar to our previous analysis on cryptocurrencies, we also need to preprocess the data extracted from the RESTful API. An organised OHLC dataset after extraction looks like Table 11. The data starts from 4th January 2016 because 1st January 2016 is a public holiday, and the following two days are Saturday and Sunday, which are day offs of the exchanges.

Table 12 illustrates the daily retruns transfromed from the previous ohlc data - Table 11. This dataset includes daily returns of each stock and is also stored in the database to help us quickly load our input dataset for analysis without transforming the data every time, which causes a computational issue.

4.2 Risk Estimation

The same methods will be used in the context of stcoks to estimate the risk of each product. The whole daily returns data is split into sliding windows

	AAPL_open	AAPL_high	AAPL_low	...	XOM_low	XOM_close
2016-01-04	97.6663	100.2915	97.0857	...	68.3743	69.2731
2016-01-05	100.6551	100.7502	97.476	...	68.7922	69.8633
2016-01-06	95.7151	97.4379	95.0584	...	68.495	69.282
2016-01-07	93.9257	95.3058	91.7841	...	67.878	68.1731
...	...	...	...	...	...	...
2018-07-26	193.9298	195.2751	192.9333	...	82.5189	83.38
2018-07-27	194.3085	194.5078	189.4356	...	79.9801	81.0837
2018-07-30	191.2293	191.5283	188.4092	...	79.887	80.9055
2018-07-31	189.6349	191.4685	188.6783	...	80.6086	80.6779

Table 11: Historical OHLC of Stocks

	AAPL	AXP	BA	...	WBA	WMT	XOM
2016-01-04	0.026703	-0.007343	-0.006224	...	-0.007766	0.015867	-0.000515
2016-01-05	-0.028748	-0.012172	0.000568	...	-0.023191	0.014347	0.012048
2016-01-06	0.001393	-0.012569	0	...	-0.011299	0.017126	0.010830
2016-01-07	-0.022599	0.008372	-0.024711	...	-0.014928	0.032714	0.002236
...	...	...	...	...	...	...	...
2018-07-26	-0.002055	-0.003306	0.011087	...	0.002523	-0.003050	0.004891
2018-07-27	-0.020565	0.009527	-0.001136	...	0.013436	-0.005754	0.011733
2018-07-30	-0.010370	-0.018300	-0.030194	...	0.011259	0.010000	-0.004748
2018-07-31	-0.000053	-0.015141	0.005418	...	-0.024383	0.002585	-0.001591

Table 12: Historical Daily Returns of Stocks

with 100 days in each window and the data in each window is used to estimate the 95% 1-day VaRs for the next days from 26th May 2016 to 31st July 2018 excluding the holidays.

4.2.1 Historical Estimation of VaR

The 29 by 549 estimated 95% 1-day VaRs of the stocks are presented on Table 13. In comparison with the VaRs of cryptocurrencies on Table 5, the values for stocks are proportionally much smaller than that of cryptocurrencies, and this means that the risks of investing in stocks are lower. When investors lose money on their investment of stocks, they do not suffer from the same degree of loss as their investment on cryptocurrencies.

The VaR estimation of stocks starts from 26th May 2016 which is different to the previous VaR estimation at cryptos because of the trading day restrictions of the exchanges. With respect to the variance between the VaRs, those of stocks are much smaller than those of cryptos, which means the loss can be smaller when suffering from a crash.

Crypto	2016-05-26	2016-05-27	2016-05-31	...	2018-07-30	2018-07-31
AAPL	-0.02591	-0.02591	-0.02568	...	-0.02049	-0.02049
AXP	-0.01537	-0.01537	-0.01537	...	-0.01493	-0.01514
BA	-0.02219	-0.02219	-0.02219	...	-0.02868	-0.02945
...	...	...	...	...	...	...
WBA	-0.02121	-0.02121	-0.02068	...	-0.02412	-0.02412
WMT	-0.01608	-0.01608	-0.01608	...	-0.01939	-0.01939
XOM	-0.01770	-0.01770	-0.01770	...	-0.01298	-0.01298

Table 13: VaRs of Stocks Before Clustering

4.3 Risk Diversification

A popular way of diversify investments in stock products is to find the industrial relationships between them. The stocks of the companies in the same industry normally have the similar risk factor which is called industrial risk. Industrial risks can be hedged by putting investments into different industry sectors. In this part, we will also use correlations and K-means clustering to find the dissimilarities between stocks and try to see if there is an industrial causation behind the correlations.

4.3.1 Product Correlations

The whole stock data will be used in here to calculate the correlation matrix. There is a similar correlation matrix chart, Figure 7, to the one of cryptos. They have same attributes such as yellow diagonal and symmetry against the diagonal. In comparison with the pattern of cryptos at 4, more colours are lighter in the chart of cryptos. That means there are more correlated pairs of stocks.

GS (Goldman Sachs) and JPM (JPMorgan Chase) have a light green with a correlation coefficient of around 0.85 on their intersection which means they are highly positive correlated. This relationship is caused by their business similarity because they are both investment banking companies. They also have positive correlations with some financial companies, e.g. AXP (American Express), TRV (Travelers).

With respect to technology companies, there are connections between them including software and hardware companies. AAPL (Apple) has higher correlations with CSCO (Cisco Systems), INTC (Intel) and MSFT(Microsoft). It is also positively correlated to V (Visa), and this might be because AAPL (Apple) has Apple Pay which is relevant to credit cards and debit cards which are the products of V (Visa).

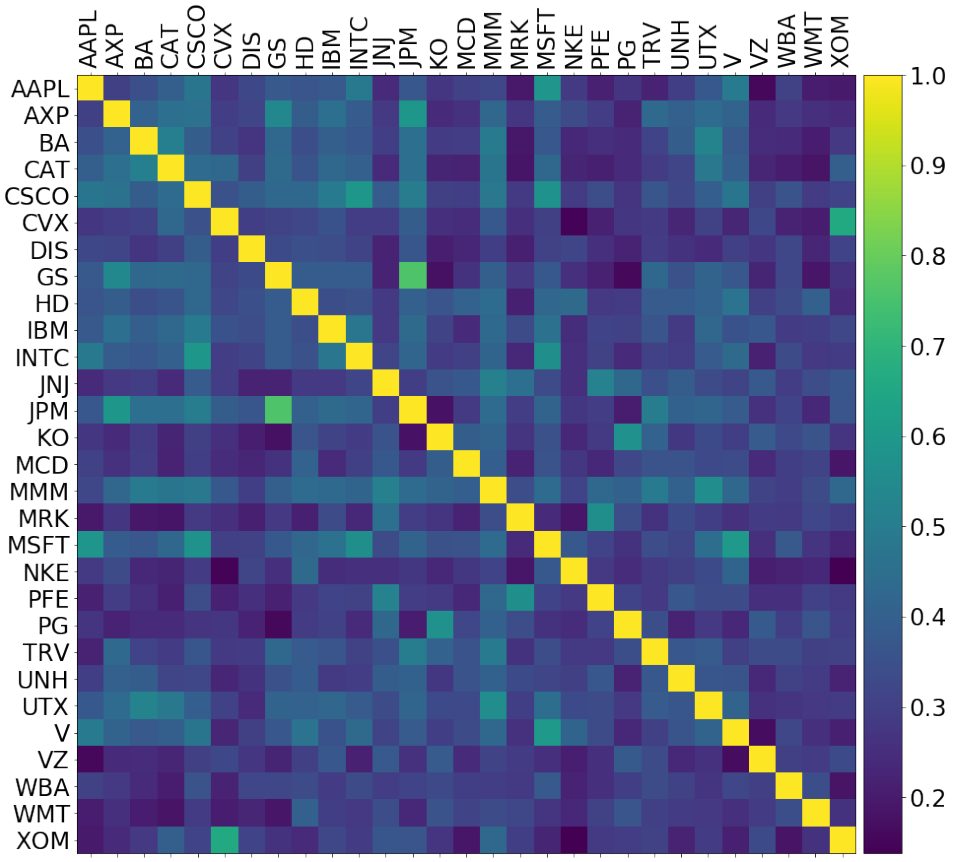


Figure 7: Correlation Matrix of Stocks

In the pharmaceutical industry, the daily returns between the stocks are highly correlated as well. For example, the colours on the blocks between PFE (Pfizer), JNJ (Johnson & Johnson) and MRK (Merck & Company) are lighter than others. Some of the companies producing the complementary goods like MMM (3M) and PG (Procter & Gamble) are also correlated to this industry.

XOM (ExxonMobil) and CVX (Chevron) are both oil and gas companies. As a result, they have a large correlation. Apart from this, they have positive correlations with MMM (3M) and CAT (Caterpillar) because these companies manufacture products using oil as materials.

4.3.2 K-means Clustering on Price Movement

Another data preprocessing needs to be completed in this section before starting K-means clustering analysis. A transpose of the daily returns dataset is crucial because this part of analysis aims on grouping the stocks together. The stocks should be the objects or observations in here. Table 14 illustrates the data of daily returns after transposing Table 12.

	2016-01-04	2016-01-05	2016-01-06	...	2018-07-30	2018-07-31
AAPL	0.026703	-0.028748	0.001393	...	-0.010370	-0.000053
AXP	-0.007343	-0.012172	-0.012569	...	-0.018300	-0.015141
BA	-0.006224	0.000568	0	...	-0.030194	0.005418
...	...	...	...	...	...	...
WBA	-0.007766	-0.023191	-0.011299	...	0.011259	-0.024383
WMT	0.015867	0.014347	0.017126	...	0.010000	0.002585
XOM	-0.000515	0.012048	0.010830	...	-0.004748	-0.001591

Table 14: Transposed Daily Returns of Stocks

Similarly, the dimensionality needs to be reduced as 649 attributes are too many for an efficient computation. We use the PCA (Principal Component Analysis) to conduct data extraction and obtain the most informative elements from all the dimensions. The PCs (Principal Components) are ordered by their informativity. The first PC which is the most informative explains approximately 11.73% of the variance across all the PCs.

The cumulative percentage of explained variance is shown on Figure 8. The blue bars represents the explaned variance percentage of each PC. The orange line means the cumulative percentage of the PCs in an order of the informativity, and it will approach 1 until PC29. We select 27 as the number of PCs because the percentage of explained variance is around 99% in here.

The data has become a 29 by 27 array which means there are 27 dimensions in each stock. We can then use this dataset to implement K-means clustering. To form consistency and compare with the result of cryptocurrencies, we select the same number of K with 8. DTW (Dynamic Time Warping) is used as the distance metric, and this metric outperforms the L2 distance (Euclidean) in the context of shape comparison.

The result after implementation of K-means clustering on the transposed daily returns dataset of stocks after PCA is presented as follow:

- *Cluster 1*: JNJ, KO, MCD, MRK, PFE, PG, VZ, WMT
- *Cluster 2*: GS, JPM
- *Cluster 3*: AXP, BA, INTC, NKE, WBA

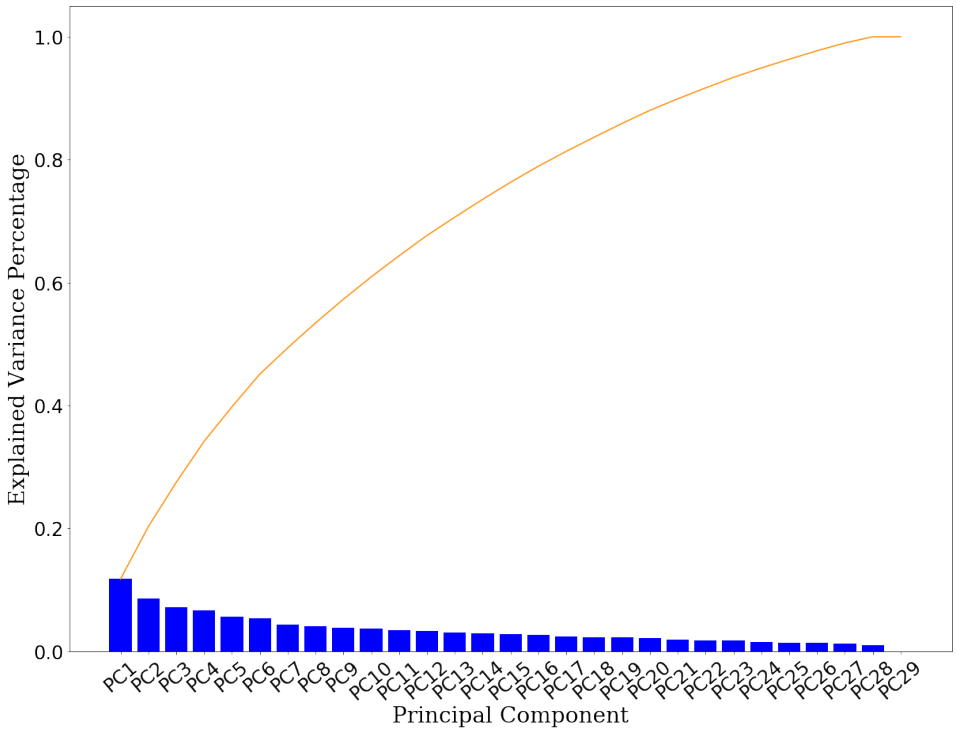


Figure 8: Cumulative Explained Variance Percentage of PCs on Stocks

- *Cluster 4*: AAPL, CSCO, MSFT, V
- *Cluster 5*: CAT
- *Cluster 6*: CVX, UNH, XOM
- *Cluster 7*: DIS, HD, IBM, TRV, UTX
- *Cluster 8*: MMM

There are some industrial correlations behind each cluster. In *Cluster 2*, GS (Goldman Sachs) and JPM (JPMorgan Chase) are both investment banks. Just as what we have anlalysed on the correlation matrix Figure 7, These 2 stocks are also correlated in the long-term analysis of activity similarity. *Cluster 4* also shows the similar result to the one we have analysed on the technology companies. AAPL (Apple), CSCO (Cisco Systems) and MSFT (Microsoft) are all technology companies except V (Visa), but Visa is also slightly relevant to technology as they are developing electronic devices.

These results may prove that our assumption on the relation in industry is correct, which states that stocks/companies in the same industry have similar price fluctuation.

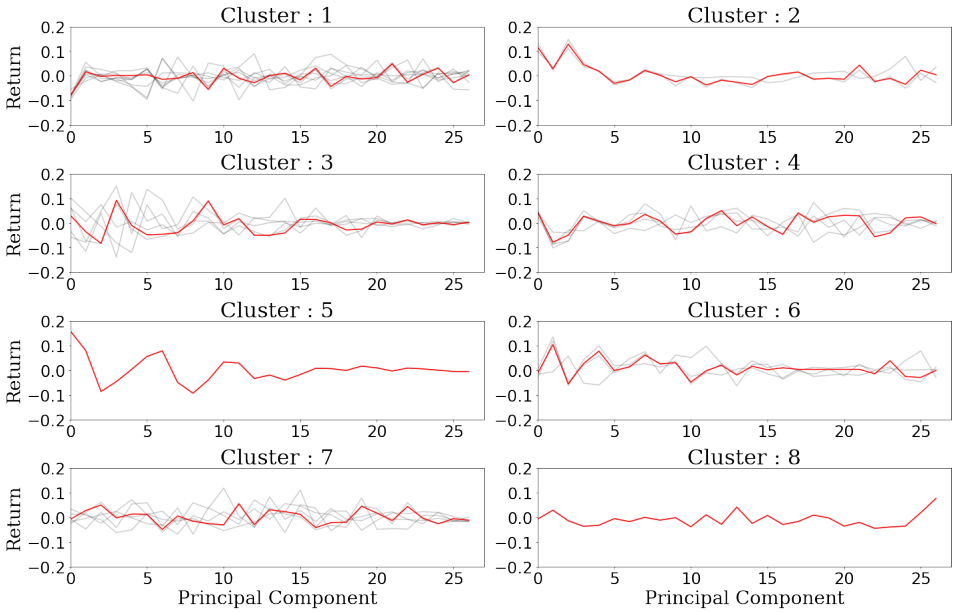


Figure 9: Clusters of Stocks

Figure 9 shows the movement of the daily returns for each cluster after the implementation of K-means clustering. The returns of the stocks generally fluctuate between a small range between -0.2 and 0.2, and this is significantly smaller than the range of cryptocurrencies. *Cluster 2*, *Cluster 3*, *Cluster 5* and *Cluster 6* have a bigger range of return movement, which means investment in these clusters might cause significant gain or loss.

4.4 Performance

4.4.1 Original Accuracy and p-value

There are 549 VaRs calculated from the sliding windows with a size of 100 in each subset. These VaRs represent the estimation of the corresponding 101_{th} day. The accuracy can then be evaluated by comparing the VaR with the actual return at that day, and this method is the same as how we evaluate the perfomance on the analysis of cryptocurrencies. If the return is lower than the VaR, the result will be indicated as a bad estimation.

Symbol	Accuracy	Symbol	Accuracy	Symbol	Accuracy
AAPL	93.26%	AXP	94.35%	BA	93.62%
CAT	93.44%	CSCO	93.44%	CVX	94.35%
DIS	93.26%	GS	93.99%	HD	93.99%
IBM	93.99%	INTC	93.44%	JNJ	93.08%
JPM	94.17%	KO	93.44%	MCD	94.72%
MMM	93.08%	MRK	93.26%	MSFT	94.17%
NKE	95.45%	PFE	94.17%	PG	94.17%
TRV	93.08%	UNH	94.35%	UTX	93.99%
V	94.35%	VZ	93.62%	WBA	93.44%
WMT	94.17%	XOM	93.81%		

Table 15: VaR Estimation Accuracies of Stocks Before Clustering

Table 15 illustrates the result of the analysis performance. Each stock has its corresponding estimation accuracy. Almost every stocks have a dissatisfactory estimation accuracy except NKE (Nike) with an accuracy of 95.45%. It can be caused by the size of sliding windows which only includes 100 data points for each portion. That is, insufficiently covering the information might result in an underestimation of the risks. The average accuracy is slightly below 94% at 93.85%.

Again, a null hypothesis can be implemented in here to test the performance of the estimation. The 5_{th} percentile of 549 days is at around 27 days. If the VaRs fail more than 27 times, the estimation breaks the 95% confidence level. The null hypothesis and its corresponding alternative hypothesis are described as follow:

- H_0: Number of VaRs being exceeded < 27
- H_a: Number of VaRs being exceeded ≥ 27

In the above hypothesis testing, we generally believe that the failures of VaR estimation will be less than 27 times. We want to prove if the evidence is strong enough to say that the number of failures can be larger or equal to 27 times. The p-value of this null hypothesis is about 6.9%. The setting of confidence level is 95%, and this means the significance level is 5%. The p-value is higher than the 5% level of significance. The evidence is not strong enough for us to reject the null hypothesis. It states that more than 27 VaRs being exceeded can be just an accident under the 95% level of confidence.

4.4.2 Improvement After Risk Diversification

Although the performance of the previous VaR estimation on stocks has satisfied the hypothesis testing at the 95% confidence level, but a p-value of 6.9% is still not good enough. A K-means clustering before estimating the VaRs might improve the performance of the VaR estimation. Table 16 contains the VaRs after the implementation of K-means clustering from 26th May 2016 to 31st July 2018 with a total of 549 values for each stock. Some of the stocks have same VaRs because every data points of a single cluster are used for estimating the VaRs, and the VaRs are regarded as the VaRs of every stocks under the cluster.

Some VaRs are different to the previous ones. Bigger VaRs mean the model reckons the risk used to be overestimated, and vice versa. A clustering helps us rebalance the risks by considering all the stocks with similar risks.

Crypto	2016-05-26	2016-05-27	2016-05-31	...	2018-07-30	2018-07-31
AAPL	-0.02346	-0.02346	-0.02326	...	-0.02061	-0.02068
AXP	-0.02099	-0.02099	-0.02076	...	-0.02263	-0.02303
BA	-0.02099	-0.02099	-0.02076	...	-0.02263	-0.02303
...	...	...	...	...	...	...
WBA	-0.02099	-0.02099	-0.02076	...	-0.02263	-0.02303
WMT	-0.01541	-0.01541	-0.01541	...	-0.01523	-0.01523
XOM	-0.01950	-0.01950	-0.01950	...	-0.01521	-0.01521

Table 16: VaRs of Stocks After Clustering

With regard to evaluating of the VaR estimation perfomance after the risk diversification, Table 17 lists the accuracies of the VaR estimation on each stock. The same metric as the previous risk estimation for performance evaluation is used in this part. We compare the VaRs with the actual returns and compute the percentage of correct VaR estimation. Red colour indicates an increase of the accuracy compared to the original performance. About 2/3 of the stocks benefit from the risk diversification on their VaR estimation. The accuracy of KO (Coca-Cola) has a remarkable improvement with around 2.73%. In *Cluster 4*, the VaR estimation of every stocks has improved, and all of them are related to technologies.

The average accuracy is about 94.37% which is higher than the estimation before clustering at around 93.85%. The same null hypothesis is used in this part to test how convincing the VaR estimation is. The p-value in here is approximately 24.14%, and this means we should retain the null hypothesis which states that the errors of VaR estimation are less than 27 times.

Symbol	Accuracy	Symbol	Accuracy	Symbol	Accuracy
AAPL	93.62%	AXP	96.9%	BA	94.35%
CAT	93.44%	CSCO	94.9%	CVX	93.81%
DIS	93.99%	GS	93.08%	HD	94.17%
IBM	93.81%	INTC	94.17%	JNJ	95.08%
JPM	95.45%	KO	96.17%	MCD	95.26%
MMM	93.08%	MRK	92.71%	MSFT	94.54%
NKE	96.17%	PFE	93.99%	PG	96.72%
TRV	94.72%	UNH	94.9%	UTX	94.54%
V	94.72%	VZ	91.07%	WBA	91.99%
WMT	93.99%	XOM	95.45%		

Table 17: VaR Estimation Accuracies of Stocks After Clustering

Also, the p-value is now much larger than the previous p-value at 6.9%. We may now confirm that a risk diversification with K-means clustering does improve the performance of traditional VaR estimation on stocks.

5 Evaluation and Conclusion

5.1 Comparison Between Cryptocurrencis and Stocks

Same analysis methods and processes have been used on both cryptocurrencies and stocks. The data formats are also similar except for the length of timeline and the product type. They all use OHLC data as their input. Both analysis follows the order of data preprocessing, risk estimation, risk diversification and risk estimation after risk diversification. However, the performance and results are slightly between each other.

The correlation matrix of cryptocurrencies is harder to correlate with the backgrounds of the products themselves. There is no obvious connections like the industrial relationship of stocks. Cryptocurrencies are less intuitive because of their virtual properties. Although some of the cryptocurrencies are highly positively correlated, we can not explain the causation behind them whereas the correlations behind stocks can be explained easier by their industrial relationships.

In the context of K-means clustering, both analysis suffers from skewness. Stocks spread into multiple clusters more evenly whereas cryptocurrencies seem to be more centralised. The reason of these phenomena might be an inappropriate choice of K. The number of cluster was not optimally chosen for each dataset because we tried to use the same metrics and compare the result of analysis on both datasets.

	Before Clustering		After Clustering	
Product Type	Average Accuracy	p-value	Average Accuracy	p-value
Cryptocurrencies	94.15%	0.03448	94.91%	0.37931
Stocks	93.85%	0.06897	94.37%	0.24138

Table 18: Performance Comparison Between Cryptocurrencies and Stocks

Table 18 compares the results of the risk estimation between cryptocurrencies and stocks. The risk estimation before risk diversification performs better on the data of cryptocurrencies in the aspect of accuracy, but the p-value of stocks is higher, and it helps retain the null hypothesis. The 94.15% accuracy in cryptocurrencies seem to be less convincing because of a lower p-value.

After a K-means clustering, both risk estimations improve significantly. The result of analysis on cryptocurrencies outperform that of stocks. The improvement accuracy on the cryptocurrencies is approximately 0.76%, and the p-value dramatically improves from 0.03448 to 0.37931 which is a 0.34483 increase. The optimisation on the analysis of stocks also performs well in the context of accuracy, as well as the p-value. The risk diversification seems to be more effective on the cryptocurrencies because it changes the risk estimation from a rejection to retaining the null hypothesis. And the increase of the accuracy is also higher then that of stocks which is around 0.52%. The p-value of cryptocurrencies surpasses that of stocks, and this is a remarkable improvement because it used to be lower than the p-value of stocks.

5.2 Self Assessment

Despite the analysis has produced a satisfactory result, there are some drawbacks and insufficient consideration among the whole process. The actions need to be completed can be organised as follow:

- **Systematic process for crypto selection**: The 29 cryptocurrencies we used were selected by inspecting if the data is complete in the specific period. As most of the cryptocurrencies emerged in the past 2 years, the cryptocurrencies that can be analysed beginning from 1st January 2016 is limited. We need to select the products for analysis based on their backgrounds instead of filtering out the cryptocurrencies by the completeness. Some of the most popular cryptocurrencies might be launched later, but they are important for comprehension of the

market behaviour.

- **Better way of reducing the dimensionality**: Currently, we are using PCA for dimensionality reduction, but the features might lose the interpretability. The original dimensions represent the dates and are in orders. When we extract the information for the features and transform them to the PCs, they become unordered and not interpretable.

- **Enforcement of exactly same metrics on both analysis**: The size of dates and time is different between the data of cryptocurrencies and stocks because of different trading hours. The market of cryptocurrencies operate 24 hours a day without any gaps whereas that of stocks only operates in certain hours of a day and stops working on holidays. This might result in a price incontinuity on the data of stocks.

- **Consideration of other algoritms**: This project only considers historical simulation as the method of estimating VaRs. There are still other approaches for VaR estimation such as Monte Carlo methods. Also, we should consider other clustering methods, and these can be hierarchical clustering or Mixture of Gaussians.

- **Modularised code for analysis**: The analysis is now implemented through the Jupyter notebook and Python. This is an ad hoc analysis process because the code is written in a scripting way without generating reusable methods. We should create a modularised codebase for scalable and reusable units.

5.3 Further Work

This project can be develped further in the future. Some of the tasks have not been completed because of the time and project size limitation. A single page web application can be built though construction of a RESTful API on the Java programme and a React.js based front-end, and this can be simply implemented by the Spring Boot. This website will visualise the result of the analysis. Also, we should separate the services into components. For example, we can split the analysis service and ETL service.

The whole analysis can be transformed into a near real time analysis on a daily basis. We can finish the analysis part of the Java programme and organise the routine tasks. The daily analysis can then be applied to

trading strategy generations. In addition, another service for connecting to the exchanges and trading can be built. This service can use the trading strategies to implement a real time algo trading. Finally, a code refactoring should be executed to improve the maintainability of the programme.

6 Professional Issues

7 Appendix

7.1 Programme Usage

References

[1] Apache httpcomponents. `https://hc.apache.org/`. [Online; accessed 3-July-2018].

[2] Apache maven. `https://maven.apache.org/`. [Online; accessed 1-June-2018].

[3] Apache spark. `https://spark.apache.org/`. [Online; accessed 25-June-2018].

[4] Cryptocompare. `https://min-api.cryptocompare.com/`. [Online; accessed 1-July-2018].

[5] Docker. `https://www.docker.com/`. [Online; accessed 21-June-2018].

[6] Git. `https://git-scm.com/`. [Online; accessed 1-June-2018].

[7] Github. `https://github.com/`. [Online; accessed 1-June-2018].

[8] Gson. `https://github.com/google/gson`. [Online; accessed 1-July-2018].

[9] Iex api. `https://iextrading.com/developer/`. [Online; accessed 10-August-2018].

[10] Joda-time. `http://www.joda.org/joda-time/`. [Online; accessed 25-June-2018].

[11] Postgresql. `https://www.postgresql.org/`. [Online; accessed 20-June-2018].

[12] Project lombok. https://projectlombok.org/. [Online; accessed 26-June-2018].

[13] Spring framework. https://spring.io/. [Online; accessed 25-June-2018].

[14] C.C. Aggarwal and C.K. Reddy. *Data Clustering: Algorithms and Applications*. Chapman & Hall/CRC Data Mining and Knowledge Discovery Series. CRC Press, 2013.

[15] Wikipedia contributors. Covariance — Wikipedia, the free encyclopedia. https://en.wikipedia.org/wiki/Covariance, 2018. [Online; accessed 26-May-2018].

[16] Wikipedia contributors. Feature selection — Wikipedia, the free encyclopedia. https://en.wikipedia.org/wiki/Feature_selection, 2018. [Online; accessed 27-May-2018].

[17] Trevor Hastie Gareth James, Daniela Witten and Robert Tibshirani. *An Introduction to Statistical Learning, with Applications in R*. Springer, New York, sixth edition, 2015.

[18] G. Hall. Pearsons correlation coefficient. http://www.hep.ph.ic.ac.uk/~hallg/UG_2015/Pearsons.pdf, February 2015.

[19] G. Hughes. On the mean accuracy of statistical pattern recognizers. *IEEE Transactions on Information Theory*, 14(1):55–63, January 1968.

[20] John C. Hull. *Options, Futures, and Other Derivatives*. Pearson, London, ninth edition, 2017.

[21] Eric Jones, Travis Oliphant, Pearu Peterson, et al. SciPy: Open source scientific tools for Python. http://www.scipy.org/, 2001–. [Online; accessed 10-July-2018].

[22] Eamonn Keogh and Abdullah Mueen. *Curse of Dimensionality*, pages 314–315. Springer US, Boston, MA, 2017.

[23] Hessam Mirgolbabaei and Tarek Echekki. Nonlinear reduction of combustion composition space with kernel principal component analysis. *Combustion and Flame*, 161(1):118 – 126, 2014.

[24] Romain Tavenard. tslearn: A machine learning toolkit dedicated to time-series data. https://github.com/rtavenar/tslearn, 2017. [Online; accessed 15-July-2018].

[25] Robert Tibshirani Trevor Hastie and Jerome Friedman. *The Elements of Statistical Learning: Data Mining, Inference, and Prediction.* Springer, New York, second edition, 2009.

[26] Wikipedia contributors. Dynamic time warping — Wikipedia, the free encyclopedia. `https://en.wikipedia.org/w/index.php?title=Dynamic_time_warping&oldid=848932991`, 2018. [Online; accessed 11-August-2018].