國立臺灣大學電機資訊學院電信工程學研究所

碩士論文

Graduate Institute of Communication Engineering

College of Electrical Engineering and Computer Science

National Taiwan University

Master’s Thesis

以異質化捉放法估算內容傳遞網路大小的資料分析

Data analysis for CDN server population estimation based on CMR model with heterogeneity

許誠

Cheng Hsu

指導教授: 黃寶儀 博士

Advisor: Polly Huang Ph.D.

中華民國 113 年 1 月

January, 2024

doi:10.6342/NTU202400040

doi:10.6342/NTU202400040

Acknowledgements

一路走來，除了自身的努力，更是受到不少人的幫忙。到現在回顧就讀碩士的期間，雖然很忙碌，但因為有你們，讓我不管在學習上還有各種生活的點點滴滴都收穫豐富。

首先，我要特別感謝我的指導教授 Polly Huang。Polly 是我在大三修的電腦網路導論認識的教授。從我認他的第一天到現在，他都是為了學生非常盡心盡力的好老師。我從 Polly 身上學的絕對不僅僅是做研究的方法，更還有面對生活的態度和提高人生的視野。很感謝大三的我去修了電網導這門課，才讓我有機會遇到這麼好的老師。

再來，我要感謝口試委員陳伶志博士和林竫茹教授。你們專業且客觀的建議點出許多我自身的盲點，最終也使我的論文能夠更加完整豐富。

接著，我要感謝實驗室的好夥伴翁瑋襄和黃珮欣，和你們一起在實驗室的時光實在太幸福了。在研究上，瑋襄和我的討論給我相當多的靈感，珮欣從第三者的角度出發也給出不少實用的建議。越是深入認識你們，越覺得自己能認識你們好幸運，因為有你們，讓我覺得多數的碩士時光都是非常愉快的！

最後，我要感謝家人。家人一直是我最重要的支柱。不管是經濟上的、心靈上的、以及填飽肚子的，有了家人的支持，我才有辦法走到今天。真的是非常謝謝你們。希望未來，我也能成為你們堅強的支柱。

doi:10.6342/NTU202400040

doi:10.6342/NTU202400040

摘要

近年來，串流媒體越來越熱門。其中，Twitch 主宰了遊戲直播的市場。在 2021 年，Twitch 擁有平均兩百七十多萬的線上即時觀眾與超過十萬名在線直播主。如同其他的串流媒體，Twitch 使用內容傳遞網路（Content Delivery Network）來提供服務給來自世界各地的廣大觀眾。內容傳遞網路可以降低內容傳播的延遲時間，是影響觀眾收看品質的關鍵之一。

在 2017 發表的一篇論文中，有團隊仔細分析過 Twitch 內容傳遞網路的架構。然而，這樣的實驗結果是一次性且高成本的。由於 Twitch 近年來的高速發展，加上缺乏長期監控的方法，大眾對於 Twitch 的內容傳遞網路的資訊所知相當有限。

在我們實驗室先前的成果中，我們成功使用捉放法中的 CJS 模型來預估 Twitch 內容傳遞網路的伺服器數量。然而在這個模型當中，所有伺服器的存活率與被抓取率皆為相同—此假設明顯不符合實際情況。如果假設每個伺服器都有各自的存活率與被抓取率，那麼 CJS 模型將可能會花費許多時間來計算。此外，對於太小的分群，CJS 模型中的最大概似估計將可能有巨大誤差。

因此，在我的研究中，我總共對五個地區的資料進行分群。我試著以在不同時段的 transaction count 作為分群用的屬性，將擁有相似存活率與被抓取率的伺服器分在同一群，藉此實現異質化的 CJS 模型。一開始，我使用 S_Dbw 來分析分群的結果。然而，我發現 S_Dbw 無法幫助預測 CJS 模型預測錯誤率。因此，我提出

doi:10.6342/NTU202400040

了 Avg/Std 來分析 CJS 的結果，越大的 Avg/Std 傾向會有越大的錯誤率。

關鍵字： Twitch、內容傳遞網路、捉放法、分群、Cormack-Jolly-Seber 模型

doi:10.6342/NTU202400040

Abstract

Streaming media become more and more popular and important in recent years. Among streaming platforms, Twitch is dominating the game streaming market. In 2021, Twitch had 2,778,000 average concurrent viewers and 105,000 average concurrent streamers.

Similar to other streaming media, Twitch uses Content Delivery Network to provide the service to massive viewers from all around the world. Content Delivery Network (CDN), which is the key part of the streaming system, is crucial for the quality of service.

In the early work, a one-time experiment has been done to survey Twitch's CDN. However, due to the rapid growth of Twitch and the high cost of a detailed scan on CDN, Twitch's CDN remains largely unknown to the public.

In our previous work, we used the CJS model, which assumes every individual shares the same time-dependent survival rate and capture probability, to estimate the CDN size. However, different servers may have different survival rates and capture probability. If

doi:10.6342/NTU202400040

we assume every server has its own survival rate and capture probability, the computation overhead of the CJS model may be too high since there are many parameters needed to estimate. Besides, maximum likelihood estimation would have a large bias if the sample size is too small [13].

In this research, I use the transaction count in hour periods to do clustering on the data from 5 countries and use the CMR model with heterogeneity with these clustering results. Next, I use S_Dbw score [7] to evaluate the clustering results. However, I find a better S_Dbw score does not lead to have a lower error rate in the MLE-CJS model. Instead, if Avg/Std in the number of sample servers larger than 0.3 of a cluster, it will tend to have a larger the estimation error rate. As a result, the clustering results with number of clusters less than 5 tend to have a lower estimation error rate since these clustering results contain less clusters with Avg/Std larger than 0.3.

Keywords: Twitch, Content Delivery Network, Capture-Recapture Models, Clustering, Cormack-Jolly-Seber Models

doi:10.6342/NTU202400040

Contents

doi:10.6342/NTU202400040

doi:10.6342/NTU202400040

doi:10.6342/NTU202400040

doi:10.6342/NTU202400040

List of Figures

doi:10.6342/NTU202400040

doi:10.6342/NTU202400040

doi:10.6342/NTU202400040

doi:10.6342/NTU202400040

List of Tables

doi:10.6342/NTU202400040

doi:10.6342/NTU202400040

Chapter 1 Introduction

Videos make up over 80% of global Internet traffic today. There are three types of online video services - stored video, conversational video, and live video. In the past, stored videos such as YouTube and Netflix accounts for the major network traffic. However, due to the Covid-19 and the technical progress, the importance of live streaming services has rapidly increased in recent years.

My research target is one of the most popular live-streaming platforms, Twitch. Twitch is a platform that hosts game streams and eSport events, and it is the one dominating the live video traffic now. In 2021, the platform had 2,778,000 average concurrent viewers and 105,000 average concurrent streamers [19]. Due to the massive numbers of viewers and streamers, Twitch accounts for more than 70% of the game streaming market with 51460 billion minutes total watched in 2021.

Twitch operates on a content distribution network (CDN) to service viewers all around the world. The mechanism of streaming on Twitch can be divided into the streamer part and the viewer part. In the streamer part, one could upload the video to the Internet, and then the copies of content distributed to servers of the CDN. The viewers can draw the videos by sending requests to one of the CDN servers. One of the advantages to deploy CDN is reducing latency for viewers. It makes CDN a key to the quality of service in live

doi:10.6342/NTU202400040

streams.

To better understand how Twitch maintains its CDN with the rapid growth of demand, continuous monitoring of the CDN is crucial. The early work [6] on the discovery of Twitch' s CDN showed that there were 876 servers in total from 12 countries in 2016. However, this work is an one-time experiment. The average number of concurrent viewers on Twitch has grown from 611 thousand in 2016 to 2.78 million in 2021. The rapid growth in scale makes it difficult to track the latest accurate information on Twitch's CDN.

It is a challenge to collect data from CDN continuously. A detailed scan to CDN is costly, and high probing overhead may disturb the service itself. On the contrary, if the probing overhead is controlled too low, one may be unable to collect enough data for a convincing result. In our vision, we think CDN detection can be divided into two-phase. Phase-1 is to sample and estimate the CDN size with lightweight probing traffic. If the CDN size changes significantly, one could conduct a detailed scan in phase-2.

1.1 Motivation

In our early work [21], we focus on phase-1. We aim to find a long-term and lightweight method to monitor the CDN population. Therefore, one could deploy a detailed scan on the CDN after detecting CDN population change significantly.We borrowed the method in biology, Capture-Mark-Recapture (CMR) model [9], to estimate the population of CDN servers on Twitch. There are two kinds of CMR models, the Lincoln-Petersen (LP) model [10, 15] and the Cormack-Jolly-Seber (CJS) model [4, 8, 18]. The LP model is the simplest CMR model that assumes the population is closed, which means the population won't change over time. The CJS model is more general than the LP model in that it allows a

doi:10.6342/NTU202400040

dynamic population. In our previous work, we have proven that the CJS model can have a better estimation accuracy with 50% less probing traffic than the LP model.

However, the CJS model assumes that every individual shares the same time-dependent capture probability and survival rate. This assumption may be too strong that different CDN servers could have different capture probabilities and survival rates. On the contrary, for the most general model, which assumes each server has its own time-dependent capture probability and survival rate, the computation time may be too high. Besides, maximum likelihood estimation would have a large bias if the sample size is too small [13]. That is the reason why I try to do the clustering on the servers to allow individual heterogeneity and avoid high computation overhead in the CJS model.

1.2 Research Goal and Challenges

Based on the theory of Open Capture-Recapture Models with Heterogeneity [16], my research goal is to improve the estimation error rate by extending the CJS model to a more general form and avoiding high computation overhead by using a clustering algorithm to divide servers into groups. Therefore, we may improve the estimation accuracy of long-term and cost-effective monitoring of Twitch's CDN and avoid high computation overhead in the estimation model in the meantime. Because the CJS model needs to estimate two parameters and capture the probability and survival rate in every sample, the data should be divided into several clusters with different values in these two parameters.

The server with a higher capture probability and survival rate tends to have a higher count in ”transactionList”. Thus, I use counts in ”transactionList” to be the attributes. Our CJS model samples at specific hours every day. For example, the model may sample

doi:10.6342/NTU202400040

from 12 pm to 1 pm every day. Therefore, transaction counts at different times of the day would lead to different estimation results. According to the above two reasons, I use the transaction counts in hour periods as the attributes to do clustering.

However, I encounter two challenges, discontinuous data, and unstable servers. The former challenge, discontinuous data, is a problem in the dataset. Take the United States data as an example, the dataset contains the data from April 13 to May 17. On April 28 and May 6, however, the crawler did not keep collecting data every hour. Because the crawler did not work for 24 hours on some days, it may cause severe estimation bias when the CJS model samples in the missing hours. The second challenge, unstable servers, is the servers with very low transaction counts. It would lead to lower estimation accuracy in the CJS model because these servers did not show up steadily.

The structure of the paper is as follows: Chapter 2 reviews the background research, including the theories of the Capture-Mark-Recapture (CMR) model, the clustering algorithm, and the clustering evaluation method. Chapter 3 introduces how Twitch's dataset was collected and what shows in this dataset. Chapter 4 describes the clustering method and the analysis of the clustering result. Chapter 5 shows the estimation error rate of the CJS model with the US data. Chapter 6 compares the results in the US-0 and the US-1, and shows the CJS results in data from other regions. Chapter 7 shows the computation time in the CJS model. Chapter 8 discusses another clustering method, online clustering. Chapter 9 is the conclusion of my thesis.

doi:10.6342/NTU202400040

Chapter 2 Related Works

2.1 Capture-Recapture-Mark Model

Our goal is to monitor the server population continuously with as little probing traffic as possible. We find the problem very similar to that of surveying the animal populations in the wild. In fact, frequent and exhaustive probes are costly and disturbing to the ecosystem we aim to preserve, not to mention that one will never be sure of the true population. Drawing from the observation, we explore the use of Capture, Mark, and Recapture (CMR) [9] for server population estimation. Elaborated below are the two models we have experimented with in the study.

2.1.1 Lincoln-Petersen Model

Lincoln-Petersen (LP) [10, 15] is the simplest model of the CMR methods. To begin with, one would capture a few animals, mark them, and release them back into the wild. The proportion of the marked animals at this point will be the number of animals captured (C) over the entire population (N). To close the deal, one would capture again. The proportion of the marked animals should equal the number of marked animals (M) over the number of animals captured in the 2nd round (R). Knowing C, M, and R, one derives the animal population N as Eq. (2.1).

$$N = \frac{RC}{M} \tag{2.1}$$

The LP method works for a close population and assumes: (1) the animal population in between 2 captures does not change and (2) the probability of the animals being captured is independent and identical (i.i.d) over time. These can be too strong for CDN discovery, where the server population is likely changing between crawling events and the chance of discovering a server is not i.i.d, knowing the server allocation is biased to its proximity to the client. Next, we introduce an open population CMR method that relaxes the assumptions.

2.1.2 Cormack-Jolly-Seber Model

Cormack-Jolly-Seber (CJS) [4, 8, 18] is designed to estimate an open population. In that, an animal might stay alive with a varying survival rate over time, i.e., the population can be dynamic. An animal might be captured with a varying probability over time as well, i.e., the chance of an animal being captured does not need to be uniform, or identical. The way it works is to capture and release the animals continuously. With the capturing history, it co-estimates the population, survival rate, and capturing probability at every capture, by maximum likelihood estimation (MLE) [2, 5, 12, 17]. The population at capture t (N_t) is calculated as Eq. (2.2), where M_t is the number of marked animals and PM_t is the proportion of marked animals.

doi:10.6342/NTU202400040

$$N_t = \frac{M_t}{PM_t} \tag{2.2}$$

In the interest of space, we present the most intuitive derivation [4] of the key term M_t and divert the readers to [18] for details. Consider the probability of marked animals being caught again in the future are identical for both marked animals released after the t th capture (every animal released is marked) and marked animals not captured in the t th capture. We can derive Eq. (2.3).

$$\frac{R_t}{CN_t} = \frac{Z_t}{M_t - CM_t} \tag{2.3}$$

CN_t is the number of animals captured in round t, and CM_t is the number of marked animals captured in round t. R_t is the number of animals, captured in round t, and being recaptured in the future. Z_t is the number of marked animals not captured in round t, but recaptured in the future. By manipulating the terms in Eq. (2.3), we come to M_t and PM_t as Eq. (2.4) and Eq. (2.5).

$$M_t = \frac{(CN_t) * Z_t}{R_t} + CM_t \tag{2.4}$$

$$PM_t = \frac{CM_t}{CN_t} \tag{2.5}$$

Note the two terms, R_t and Z_t. They account for the chance of the animals being recaptured in the future, which depends on whether they will survive in a future time and the chance of them being recaptured at the time. The two terms are essential compounds

doi:10.6342/NTU202400040

of survival rates and capturing probabilities into the future. That is why, in CJS, the population estimations in the past are often adjusted as they are regressed to fit the new data. As the new data are added, the values of R_t and Z_t are affected. In the meantime, the population estimations in the past are adjusted. CJS is unique in that it takes into account data in the past and the future.

Eq. (2.4) and Eq. (2.5) produce a biased estimation of the population. The tendency is to overestimate and the bias can be large for small samples (e.g., animals that are hard to capture or close to extinction), and the following extension Eq. (2.6) and Eq. (2.7) is often applied to mitigate the bias.

$$M_t = \frac{(CN_t + 1) * Z_t}{R_t + 1} + CM_t \tag{2.6}$$

$$PM_t = \frac{CM_t + 1}{CN_t + 1} \tag{2.7}$$

2.2 CMR model with Heterogeneity

In the CJS model, it is assumed that survival rate and capturing probability are homogeneous among each individual. However, the assumptions may be too strong for CDN servers because each server may have a different survival rate and capturing probability.

In the CMR model with heterogeneity [16], it provides a flexible framework of likelihood-based models that allow individuals from different classes would have different survival rates and capture probability. Thus, we try to use the CMR model with heterogeneity to improve the estimation accuracy.

doi:10.6342/NTU202400040

To calculate the overall likelihood, we first consider the individual likelihood in the homogeneous case. For the animal i with known capture history CH_i, first capture time f_i, last capture time l_i, and unknown departure time d_i, we can derive the probability for the observed capture history on the condition of f_i, d_i and the probability of departure time given f_i.

$$Prob(CH_i|\{f_i, d_i\}) = \prod_{j=f_i+1}^{d_i} p_j^{x_{ij}}(1-p_j)^{1-x_{ij}} \tag{2.8}$$

The probability of this departure time d_i, given f_i, is Eq. 2.9

$$(\prod_{j=f_i}^{d_i-1} \phi_j)(1-\phi_{d_i}) \tag{2.9}$$

Thus, based on the above two equations, we can calculate the probability of capture history CH_i by summing up all possible departure times.

$$Prob(CH_i|f_i) = \sum_{d=l_i}^{K}\{(\prod_{j=f_i}^{d-1} \phi_j)(1-\phi_d) * (\prod_{j=f_i+1}^{d} p_j^{x_{ij}}(1-p_j)^{1-x_{ij}})\} \tag{2.10}$$

To calculate the likelihood for animal i in heterogeneity case, we assume there are C classes of animals. Each class has its own time-dependent capture probability and survival rate. Each animal has a probability π_c of coming from class c (sum(πc) = 1), which has the capture probability p_{jc} and the survival rate $\phi_j c$ at j th sample. Sum up the values of $Prob(Ch_i|f_i) * Prob(class = c)$ for c = 1,2, ⋯C, we can derive the likelihood function for animal i in heterogeneity case.

doi:10.6342/NTU202400040

$$L_i = \sum_{c=1}^{C} \pi_g [\sum_{d=l_i}^{K} \{(\prod_{j=f_i}^{d-1} \phi_{jc})(1-\phi_{dc}) * (\prod_{j=f_i+1}^{d} p_{jc}^{x_{ij}}(1-p_{jc})^{1-x_{ij}})\}] \tag{2.11}$$

For n observed animals, we can derive the overall likelihood function in the following equation.

$$\begin{aligned} L_i &= \prod_{i=1}^{n} L_i \\ &= \prod_{i=1}^{n} \sum_{c=1}^{C} \sum_{d=l_i}^{K} \{\pi_c(\prod_{j=f_i}^{d-1} \phi_{jc})(1-\phi_{dc}) * (\prod_{j=f_i+1}^{d} p_{jc}^{x_{ij}}(1-p_{jc})^{1-x_{ij}})\} \end{aligned} \tag{2.12}$$

The above equation is the full likelihood function of the CMR model with C classes of animals.

2.3 Clustering Algorithm

2.3.1 K-means

K-means is one of the simplest clustering algorithms [14]. The target of k-means is dividing n points into k clusters in which each point belongs to the cluster with the nearest cluster center. Firstly, we have to choose the number of clusters. Let's assume we choose k as the number of clusters. The algorithm randomly chooses k different points as cluster centers in the beginning. After choosing the initial cluster centers, k-means will assign every data point to the nearest cluster. Based on all the points in each cluster, calculate the mean of each cluster and assign the mean as the cluster center of each cluster. Next, we repeat assigning each point to the nearest cluster and recalculate each cluster center until the cluster center doesn't change between iterations.

K-means has some pros and cons. The main advantage is fast, which means k-means is an efficient algorithm. On the contrary, k-means has two main disadvantages. Firstly, the number of clusters has to be decided by the user. If the number of clusters to too large or too small number, the clustering result will be very bad. Secondly, k-means cannot always find the best cluster result. The different initial cluster centers may generate different results, which means k-means may not be able to produce consistent results.

2.3.2 Mini-Batch K-Means

The Mini-batch K-means is a variant of the k-means algorithm which reduces the time required for the k-means algorithm to find convergence. This algorithm uses small, random, and fixed-size samples to generate batches and store them in memory. In each iteration, a new random sample of the dataset is used to update the clusters - each data in the batch would be assigned to the nearest cluster, and then update locations of the centers based on the new result. The iteration will continue until convergence.

In the API of Mini-Batch K-Means, the default of the initial cluster centers is 'K-Means++' [1]. 'K-Means++' is a method to speed up the convergence by choosing the initial cluster centers with the distance between centers as large as possible. 'K-Means++' starts by choosing a point from data as the first initial cluster center. In the next step, calculate the distance between each point in data to the first initial cluster center. The larger the distance between a point to the first initial cluster center, the higher the chance of a point being chosen as the next initial cluster center. Repeat these steps until k cluster

doi:10.6342/NTU202400040

centers have been chosen.

2.3.3 Mean Shift

Mean-shift clustering is an unsupervised clustering algorithm [3]. It is a centroid-based algorithm, which shifts each data point to the average of data points in its neighborhood. Mean shift starts from initializing a sliding window for each point in the data. Next, each sliding window is shifted toward the mean of the points in the sliding window. The shift will continue until convergence, which means it has a maximum density of points.

The Mean-shift algorithm automatically decides the number of clusters. On the contrary, there is a parameter, ’bandwidth’, which determines the size of the sliding window. In the API of Mean-shift [1], the default of ’bandwidth’ is provided by ’estimate_bandwidth’, which estimates the bandwidth for the data.

2.4 Clustering Evaluation - Sdb_w

Sdb_w validation [7] has a better performance than other clustering algorithms in many kinds of situations [11]. The basic idea of Sdb_w is to consider both inter-cluster density and intra-cluster variance. In the following paragraphs, I will introduce two terms, inter-cluster density, and intra-cluster variance, respectively.

The inter-cluster density is expressed as Dens_bw, which is used to evaluate the density of the clusters and density among clusters, shown as Eq. 2.13. A good clustering result will have a low density among clusters in comparison with the density of the clusters.

doi:10.6342/NTU202400040

$$Dens\_bw(c) = \frac{1}{c(c-1)} \sum_{i=1}^{c} [\sum_{j=1, i \neq j}^{c} \frac{density(u_{ij})}{max\{density(v_i), density(v_j)\}}] \tag{2.13}$$

The intra-cluster variance will be evaluated by the average scattering for clusters. We use Scat to express intra-cluster variance as Eq. 2.14.

$$Scat(c) = \frac{1}{c} \sum_{i=1}^{c} \|\sigma(v_i)\| / \|\sigma(S)\| \tag{2.14}$$

By the definition of the above two terms, the clustering validation S_Dbw is defined as Eq. 2.15:

$$S\_Dbw(c) = Scat(c) + Dens\_bw(c) \tag{2.15}$$

The lower value of S_Dbw represents the better clustering result.

doi:10.6342/NTU202400040

doi:10.6342/NTU202400040

Chapter 3 Pilot Experiment

In this section, I will introduce Twitch's CDN discovery done by our lab member, Caleb Wang, and the data mining in this dataset which is done by me.

3.1 Data Set

This dataset is collected by Wang in 2020 and 2021 [20]. It contains Twitch's CDN servers from all around the world.

3.1.1 Twitch CDN Discovery

Twitch is an interactive live-streaming service for gaming, entertainment, and more. Take the statistics in February 2022 as an example, the average number of viewers is 2.96M, and the maximum number of viewers is 5.52M [19]. Twitch is one of the most popular live-streaming service providers around the world.

As shown in Fig. 3.1, for each viewing request, the client starts by connecting to the load balancer (Usher), which replies with the file containing the addresses of the playlist server corresponding to different videos. After that, the client sends the request to the playlist server for a video playlist (.m3u8 file) containing an order of URLs, each pointing

doi:10.6342/NTU202400040

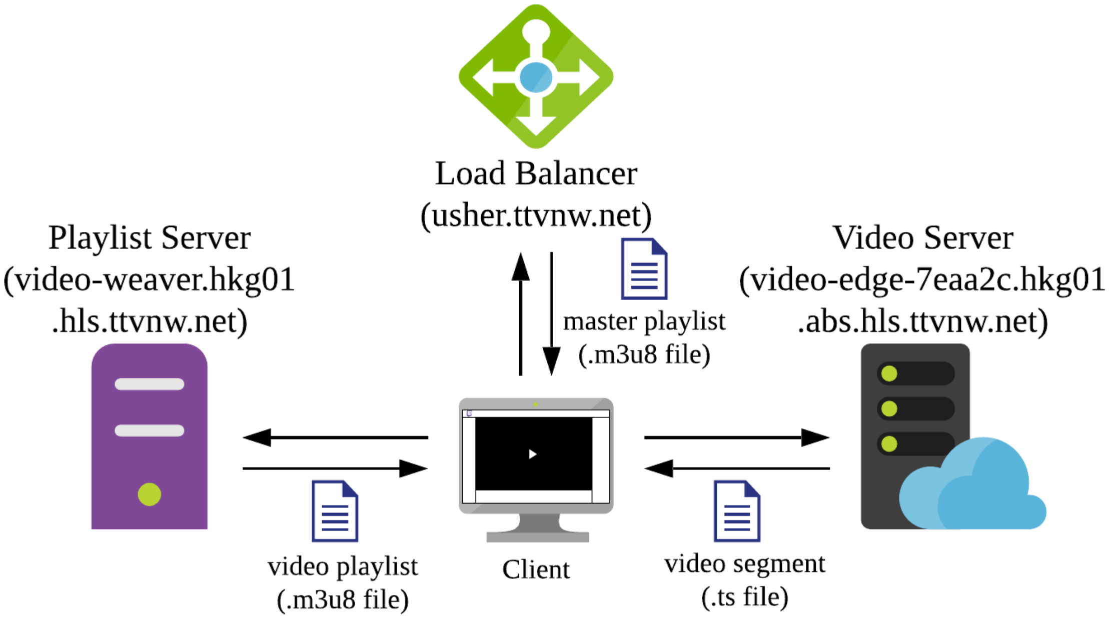


Figure 3.1: 3-Way Redirection Video Lookup

to the content server of a video segment (.ts file).

Although the early work on Twitch [6], Twitch's CDN remains a lot of unknowns to the public. The CDN network of Twitch is very large. If we try to scan the whole of Twitch's CDN continuously, it will cause a high amount of probing traffic. To monitor Twitch's CDN without high-volume traffic, we borrowed the method from biology, the CMR model, to estimate the CDN size with a low amount of probing overhead.

3.1.2 Data Collection

Our lab member, Wang, implemented the crawler by using Twitch's API and VPN server to collect the CDN data from different regions. In the beginning, Wang used public Twitch APIs to get real-time information about channel viewer count. Based on the statistics of viewer count, the algorithm will choose the top K channels that account for 80% of the total viewers. Next, Wang initiated the VPN connections in Docker containers. A container is a process that runs on the top of the operating system's kernel. Compared to

doi:10.6342/NTU202400040

a virtual machine, a container is low resource-intensive to isolate applications from their environments and work uniformly across systems in the meantime. In the end, Wang selected VPN servers in 18 different countries that contain about 75% traffic on Twitch. In each country, Wang deployed VPN servers with a wide geographic span. For example, there were 7 VPN servers on both the west coast and east coast of the United States.

3.1.3 Data Structure

The data is stored in MongoDB, and the dataset in MongoDB is divided into several collections that each represent VPN servers in one country. In each data in the collection of the database, there are nine different attributes as shown below. I will introduce seven of these attributes. [’_id’, ’vpnServerId’, ’channel’, ’language’, ’serverPool’, ’start’, ’end’, ’transactionList’, ’addrPool’]

’_id’: It is the primary key in MongoDB. It is auto-generated by MongoDB. Each data has its unique ’_id’ in the database for identification purposes.

’vpnServerId’: It represents which VPN server this connection used. In each connection, only one VPN server was used.

’channel’: It means what the streamer was in this connection. The algorithm selected the top K channels, and the crawler chose one of the K channels to connect.

’language’: The channel languages we selected are English(en), Spanish(es), Korean(ko), Chinese(zh), and French(fr), which cover nearly 70% of channels on Twitch according to the statistics in TwitchTracker [19].

’start’: This is the start time of the connection. The time is accurate to seconds. (e.g.

doi:10.6342/NTU202400040

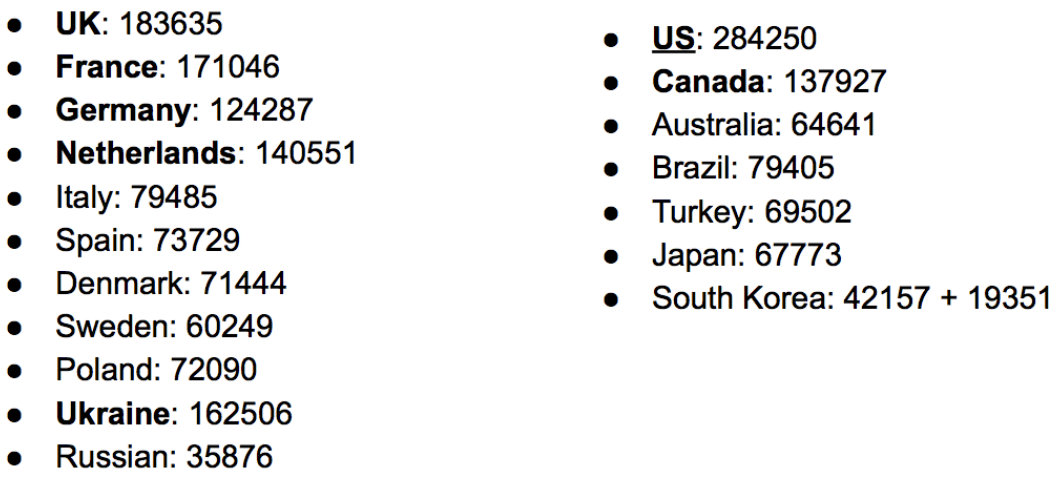
2020-10-19T20:56:04)

’end’: This is the end time of the connection. Same as the ’start’, the time is accurate to seconds.

’transactionList’: This attribute is the record of probing. It contains a list of times and the corresponding server IPs. The times in the list are neither earlier than ’start’ nor later than ’end’. The following experiments use values in this attribute to simulate ’capture’ events in the CMR model. (e.g. ’2020-10-19T14:56:08’: ’52.223.247.211’, ’2020-10-19T15:02:04’: ’45.113.128.160’)

3.2 Data from different regions

The data was collected through VPN servers from 18 countries in 2021. The number of data in different countries is shown in the below figure Fig. 3.2.



- **UK**: 183635
- **France**: 171046
- **Germany**: 124287
- **Netherlands**: 140551
- Italy: 79485
- Spain: 73729
- Denmark: 71444
- Sweden: 60249
- Poland: 72090
- **Ukraine**: 162506
- Russian: 35876

- **US**: 284250
- **Canada**: 137927
- Australia: 64641
- Brazil: 79405
- Turkey: 69502
- Japan: 67773
- South Korea: 42157 + 19351

Figure 3.2: Data Size in Each Region

The definition of a ’ unit’ in the data size is how many ’_id’ in the datasets. The bold words mean the number of data in these countries is over 100,000. Among these data,

doi:10.6342/NTU202400040

we can notice that the number of data in the US (the United States) is the largest. I dug into the data in countries with more data and plotted the relationship between dates and numbers of observed IPs, as shown in Fig. 3.3, Fig. 3.4, Fig. 3.5, Fig. 3.6, Fig. 3.7.

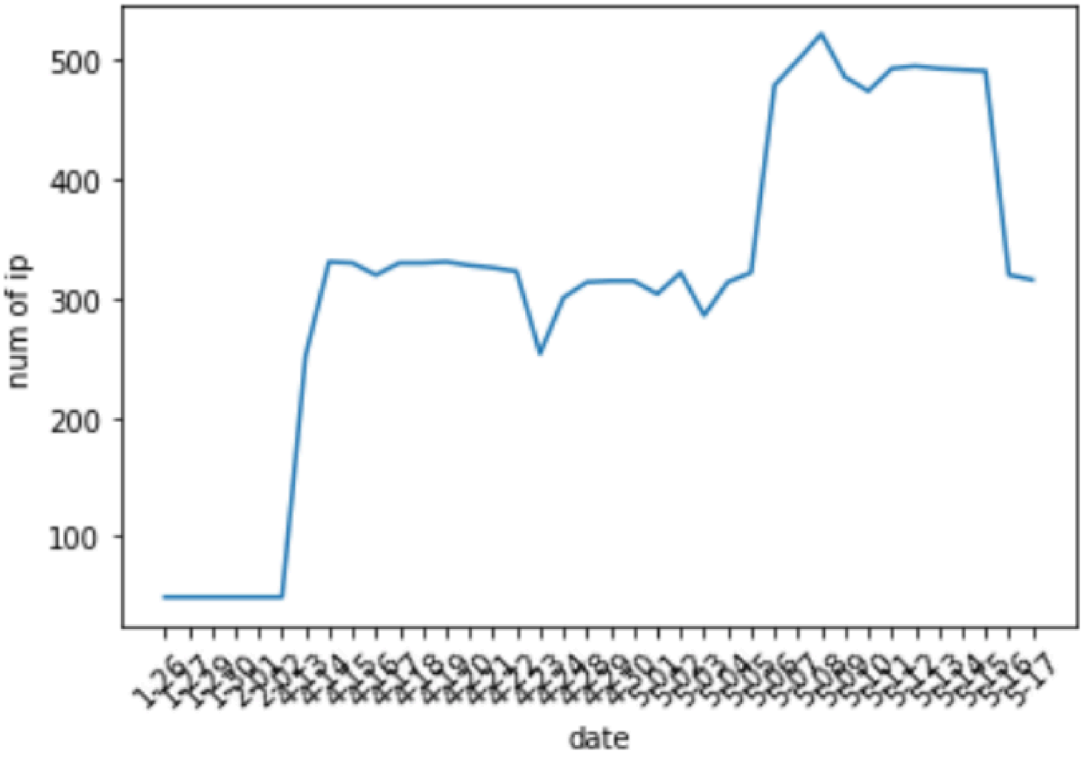


Figure 3.3: Number of servers in the US

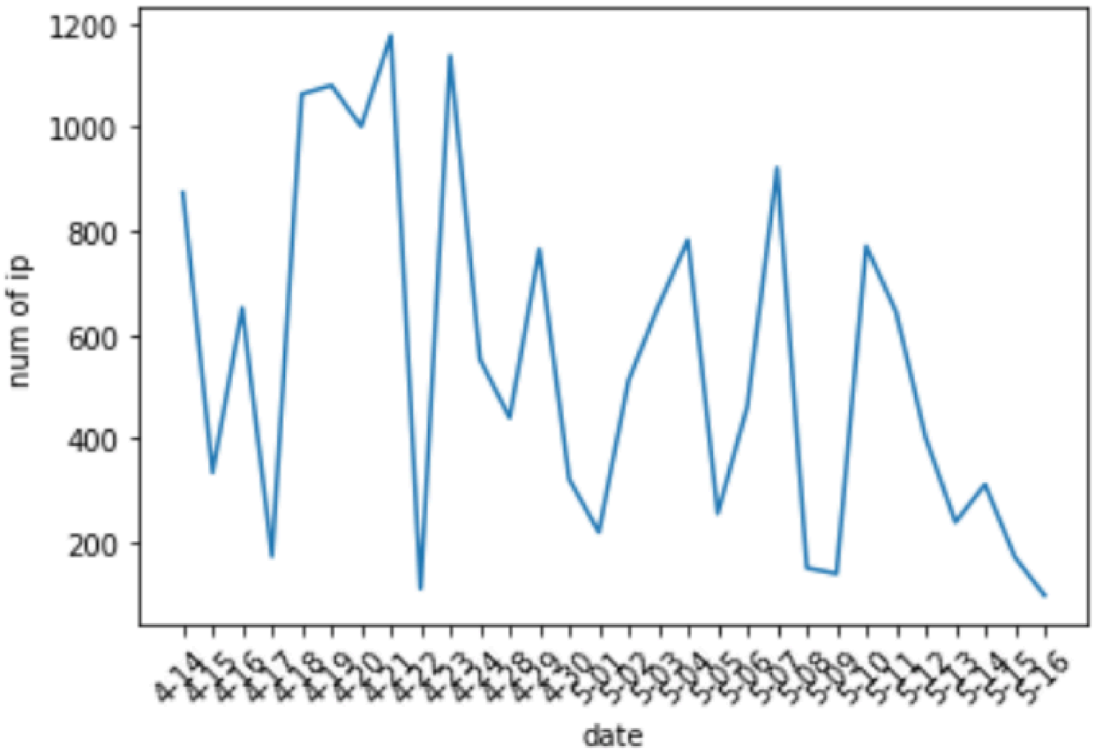


Figure 3.4: Number of servers in the UK

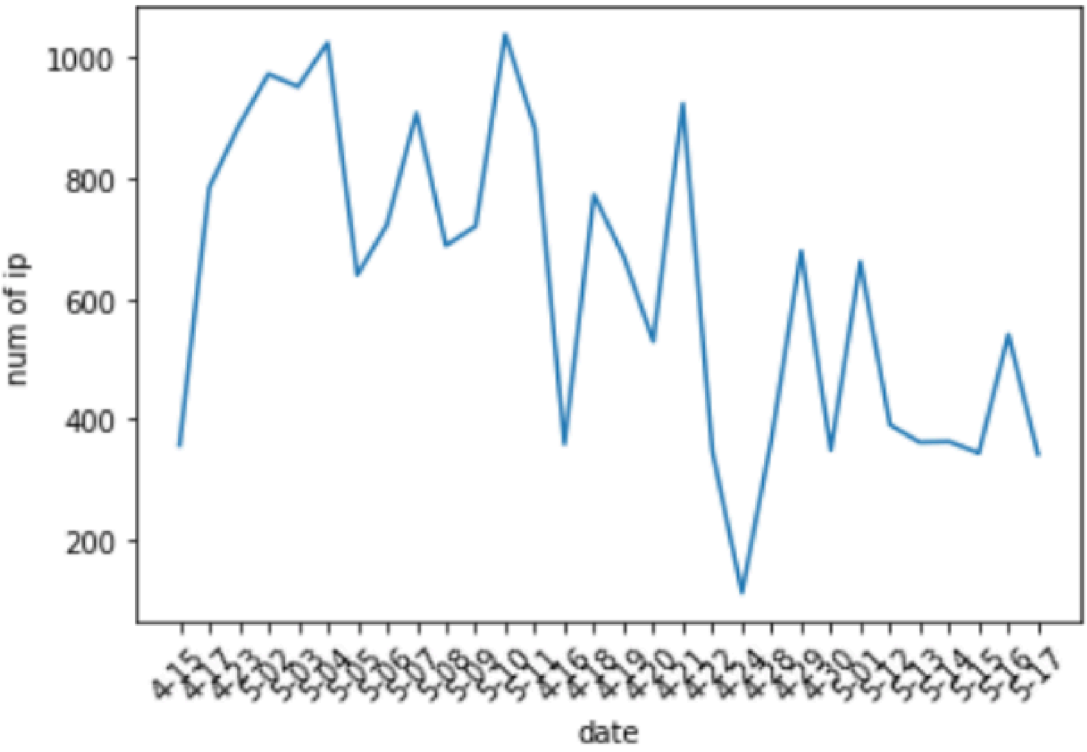


Figure 3.5: Number of servers in France

doi:10.6342/NTU202400040

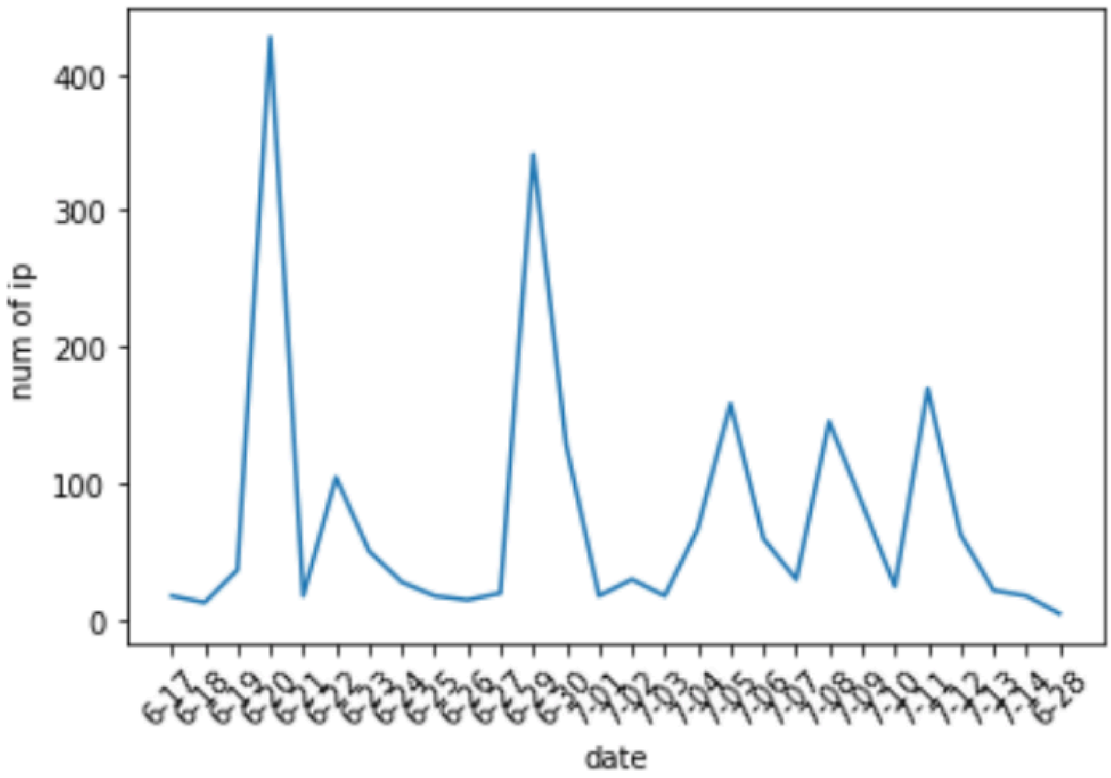


Figure 3.6: Number of servers in the Netherlands

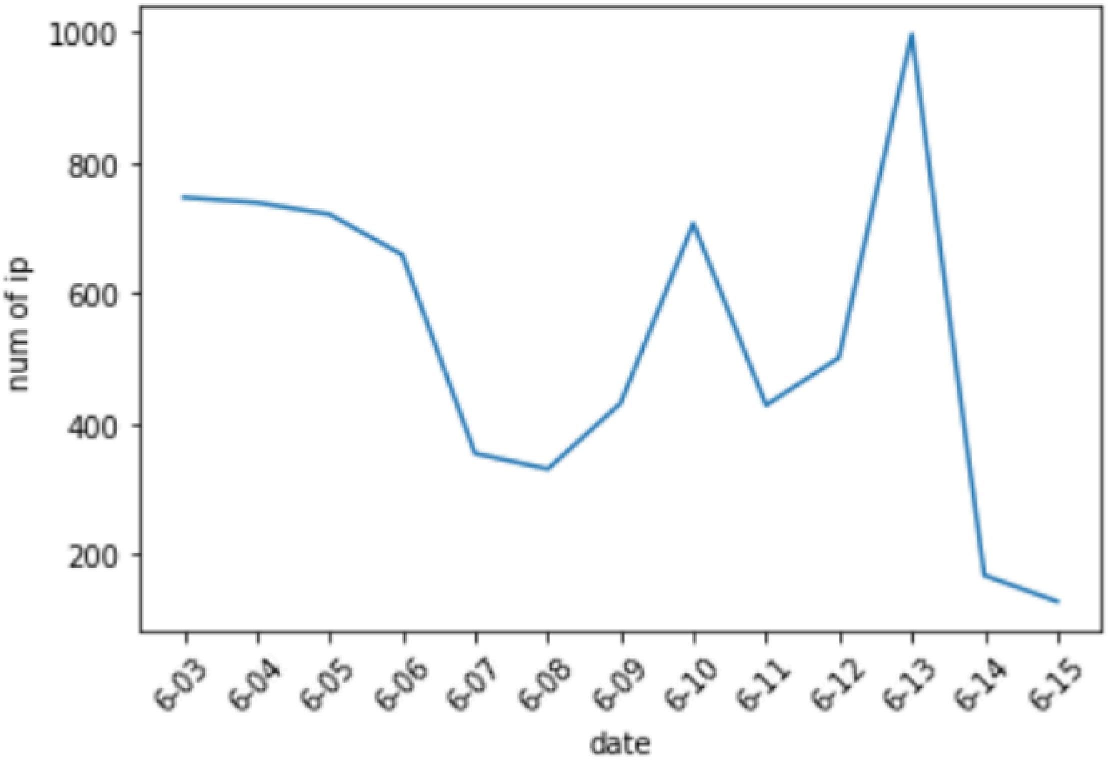


Figure 3.7: Number of servers in Germany

doi:10.6342/NTU202400040

As the figure shows above, the number of IPs in the United States is relatively stable, which ranges from about 300 to 500 in 2021. Since the US data is the most abundant, I use the data in the United States to demonstrate the data mining and clustering method. The CJS model results with the UK, France, Netherlands, and Germany data are in Chapter 6.

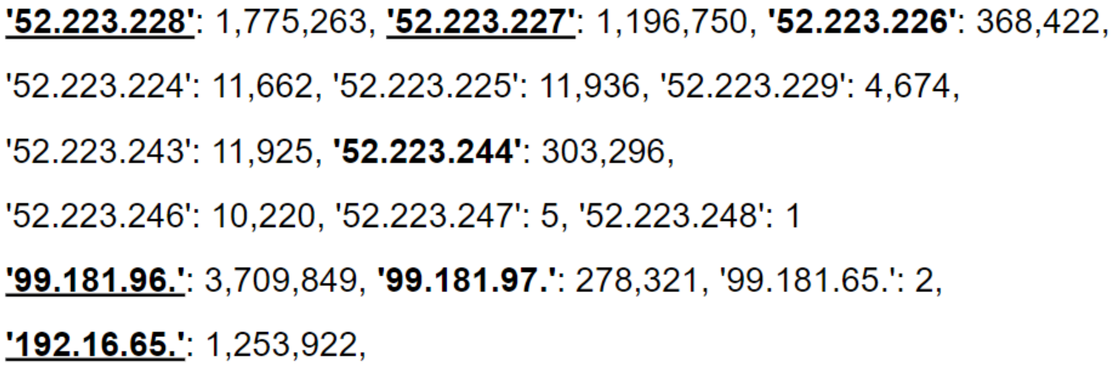
3.3 Data mining

The total number of data in the US is 284250, and the total number of servers in this dataset is 619. The data is distributed in two time periods. The earlier one is from November 26th to December 2nd 2020. The latter one is from April 13th to May 17th 2021. Eventually, I chose the data in the later period for the following two reasons. Firstly, the data in the earlier period has no data on November 28th, 2020. The missing data on November 28th may cause the CMR model to have a lower estimation accuracy. Secondly, the data size between the two periods differs a lot. The number of data in the earlier period is 23011 while the number of the latter one is 261238. The size of the later one is over 10 times larger than the earlier one. Based on the above two reasons, I chose the data from April 13th to May 17th in 2021 to do the research.

3.3.1 Subnet Overlook: 24 Subnet Mask

For the data in the 'transactionList', I count how many times did each subnet appear. The result is in Fig. 3.8. It is obvious that the transaction count is highly concentrated in several specific subnets, such as '52.223.228', '52.223.227', '99.181.96', and '192.16.65'. All four subnets mentioned above have been recorded in 'transactionList' more than one million times, which contains 88.8% of the whole data.

doi:10.6342/NTU202400040



'52.223.228': 1,775,263, **'52.223.227'**: 1,196,750, **'52.223.226'**: 368,422,
'52.223.224': 11,662, '52.223.225': 11,936, '52.223.229': 4,674,
'52.223.243': 11,925, **'52.223.244'**: 303,296,
'52.223.246': 10,220, '52.223.247': 5, '52.223.248': 1
'99.181.96.': 3,709,849, **'99.181.97.'**: 278,321, '99.181.65.': 2,
'192.16.65.': 1,253,922,

Figure 3.8: Transaction Count in Subnets

3.3.2 Hour-Count Distribution

In our CMR model, we would sample the server for a fixed period every day to estimate the whole population of servers. Thus, it is important to check the data each hour.

To find the relationship between the counts of 'transactionList' in each hour, I plot Fig. 3.9. The x-axis is the hour range from "00" to "23". The y-axis is the total times that the servers are observed in the corresponding hour of every day.

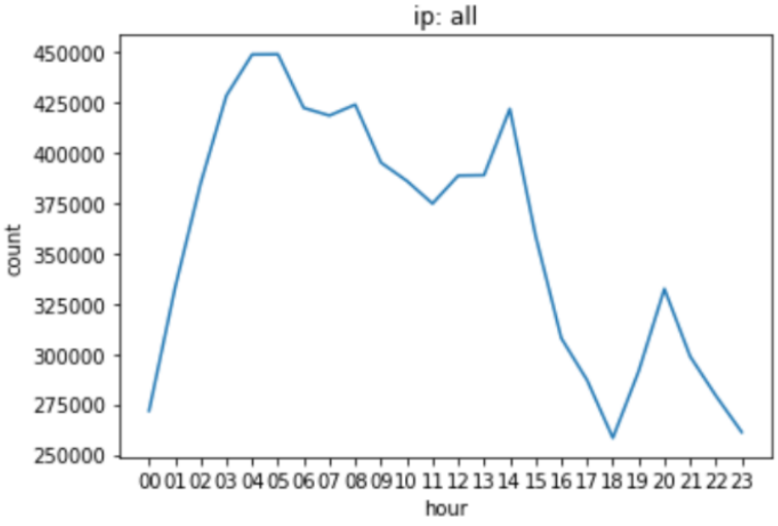


Figure 3.9: Hour-Count Distribution of All Servers in the US

I find that the servers from the same subnet have a similar hour-count relationship.

Take the '52.223.228' subnet as an example, the left chart in Fig. 3.10 is a server, '52.223.228.8', in '52.223.228' subnet, and the right chart in Fig. 3.10 is the total count of all servers in '52.223.228' subnet. Both charts have a peak in the hour of '11' to '13' and remain low from '18' to '07'. The hour-count charts of other servers in the '52.223.228' subnet share a similar shape. As a consequence, it makes sense to divide the servers by 24 subnet masks.

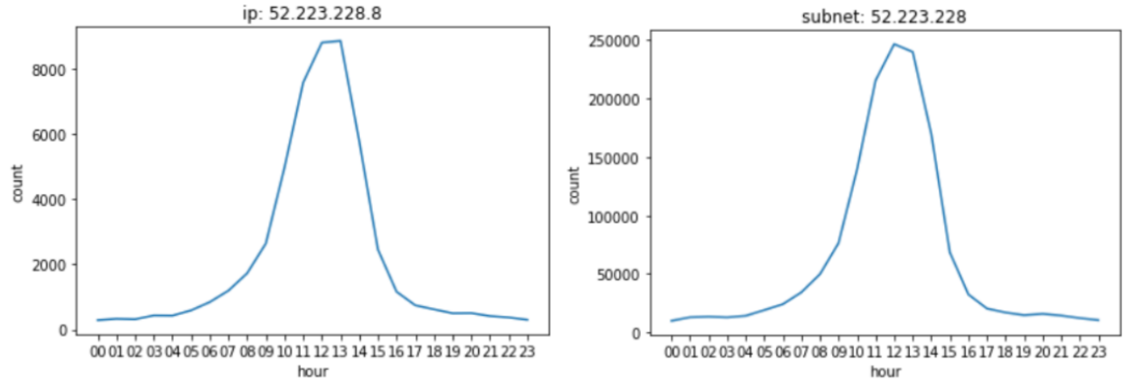


Figure 3.10: Hour-Count Distribution - 52.223.228.8 and 52.223.228

3.4 Continuous Data

The CJS model takes the historical data and future data into consideration. This leads to the estimation number in the first two days and the last two days will have a relatively large bias since the model does not converge well on these days. Thus, we only choose the data with working hours equal to 24 for more than 7 continuous days to deploy the CJS model.

3.4.1 Data in the US

In the US dataset, the number of servers in 2021 is shown as the Fig. 3.11. The number of servers is quite steady from April 13 to May 6 the number is between 250 to 350. However, there was a significant jump in the number on May 7 the number of servers is more than 450. The number stayed at more than 450 until May 16.

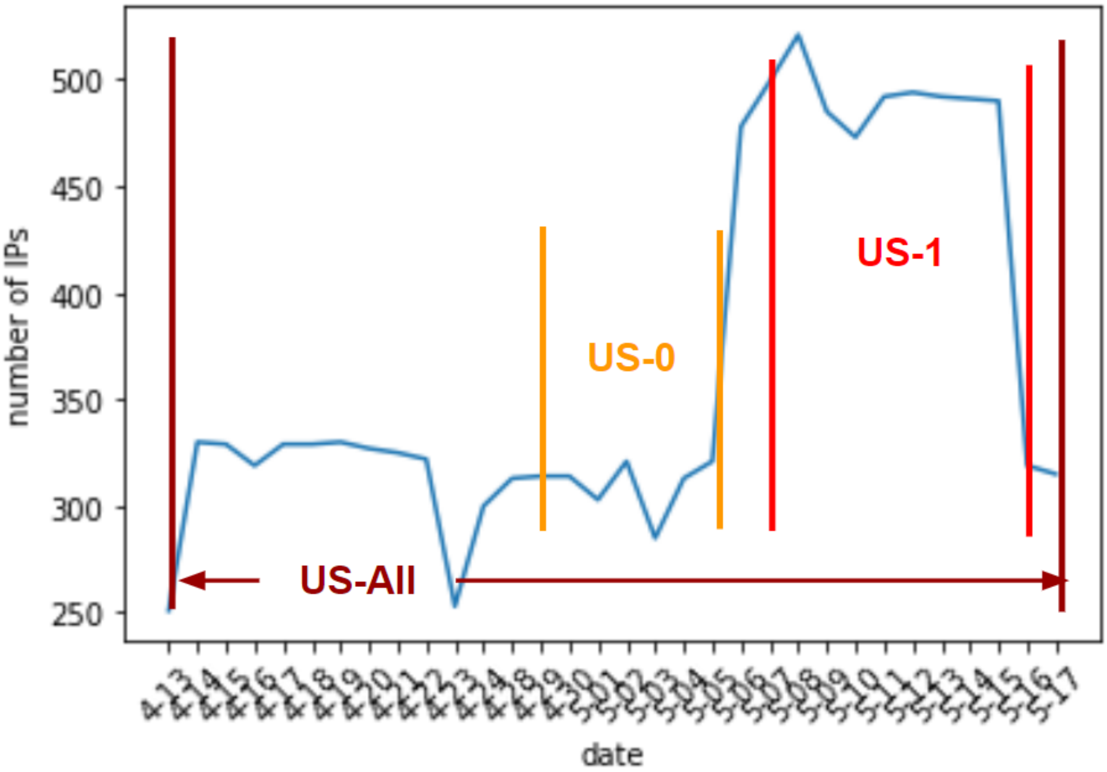


Figure 3.11: Number of IPs in the US

To understand the reason for discontinuity in server number, I checked how many hours did the crawler collect the US data every day. The Fig. 3.12 shows the working hours, which represents how many hours are recorded in the ”transactionList” every day in 2021. There are only two periods that the crawler functioned 24 hours for more than 7 continuous days, April 29 to May 5 and May 7 to May 16. I will use the US-0 to represent the data from April 29 to May 5, the US-1 for the data from May 7 to May 16, and the US-All for the data from April 13 to May 17. Nevertheless, the crawler only worked for less than 20 hours on May 6, so did some dates that were not in the US-0 and period-1.

doi:10.6342/NTU202400040

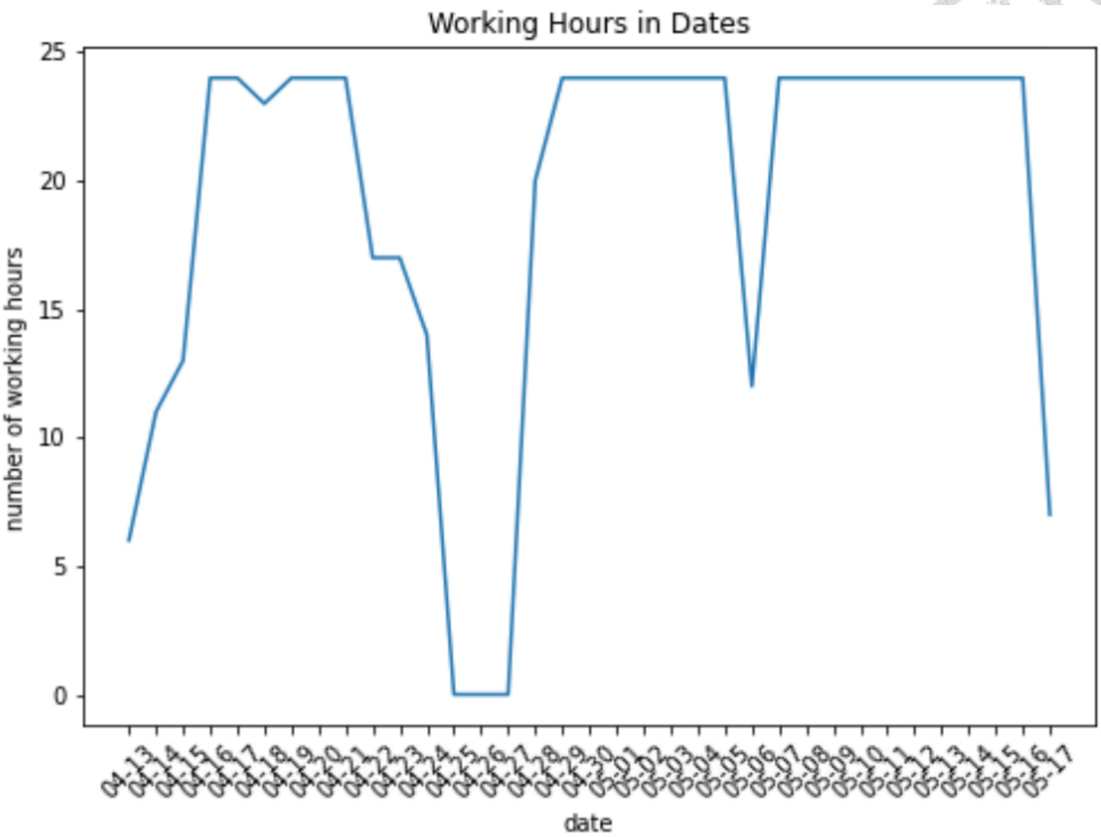


Figure 3.12: Number of Working Hours in US-All

The Fig. 3.13 shows the data on May 6, the date that the crawler did not work for 24 hours. The blue line is the number of servers every hour, and the orange line is the counts in “transactionList” every hour. Before the 18-hour clock, the numbers of IPs and transaction counts are almost zero. In fact, 12 over 24 hours on May 6 did not have any data collection record.

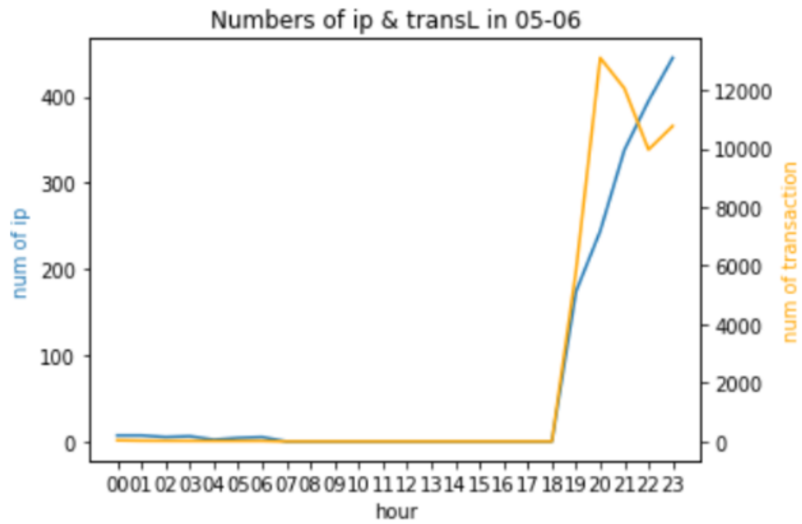


Figure 3.13: Number of IPs and Transaction Count on May 6

Because the CJS model samples in a few specific hours every day, the data missing in some hours may lower the estimation accuracy. For example, if the CJS model samples from 12 pm to 2 pm, the model would not be able to work normally on May 6 since the ”transactionList” didn’t have any record in these hours. Furthermore, we evaluate the accuracy of the model by comparing the estimation number to the ’baseline’, which is the total number of IPs collected in one day. If the crawler works less than 24 hours, the ’baseline’ may not be close to the ground truth. Thus, I try to use the data from the periods, the US-0 (April 29 to May 5) and the US-1 (May 7 to May 16) to do clustering, which are the only two periods that meet the CJS model needs in the US data.

Take the US-1 as an example, the state of the data is shown in Table 3.1. The first value in every row is the number of hours the crawler collects data, which is all equal to 24 in the US-1 (May 7 to May 16). Next, ’total count’ means the count in ”transactionList” on that date. In the end, ’mini hour count’ represents the minimum transaction count in one hour, which means it would be 0 if the working hours are less than 24. The average daily data size in the US-1 is approximately 10 times the data on May 6 and May 17, which

doi:10.6342/NTU202400040

means the data is much more robust in the US-1. That is one of the reasons why I chose data from the US-0 and period-1 to do clustering, which is shown in Chapter 4.3.

dates	working hours	transaction counts	mini hour count
May-6	12	51894	0
May-7	24	436552	13101
May-8	24	511106	13064
May-9	24	460176	9374
May-10	24	362727	10053
May-11	24	403603	11393
May-12	24	438546	7760
May-13	24	444512	8894
May-14	24	473601	13575
May-15	24	487068	10961
May-16	24	255964	5757
May-17	7	40057	0

Table 3.1: Data from 2021 May 6 to May 17

3.4.2 Data in Other Regions

To see which data meets the criterion that could be deployed on the CJS model, I dug into the data in the UK, France, Netherlands, and Germany. Fig. 3.14, Fig. 3.15, Fig. 3.16, and Fig. 3.17 show the transaction count and working hours on each date. In these figures, the blue lines are the number of working hours, and the orange lines are the number of transaction counts.

doi:10.6342/NTU202400040

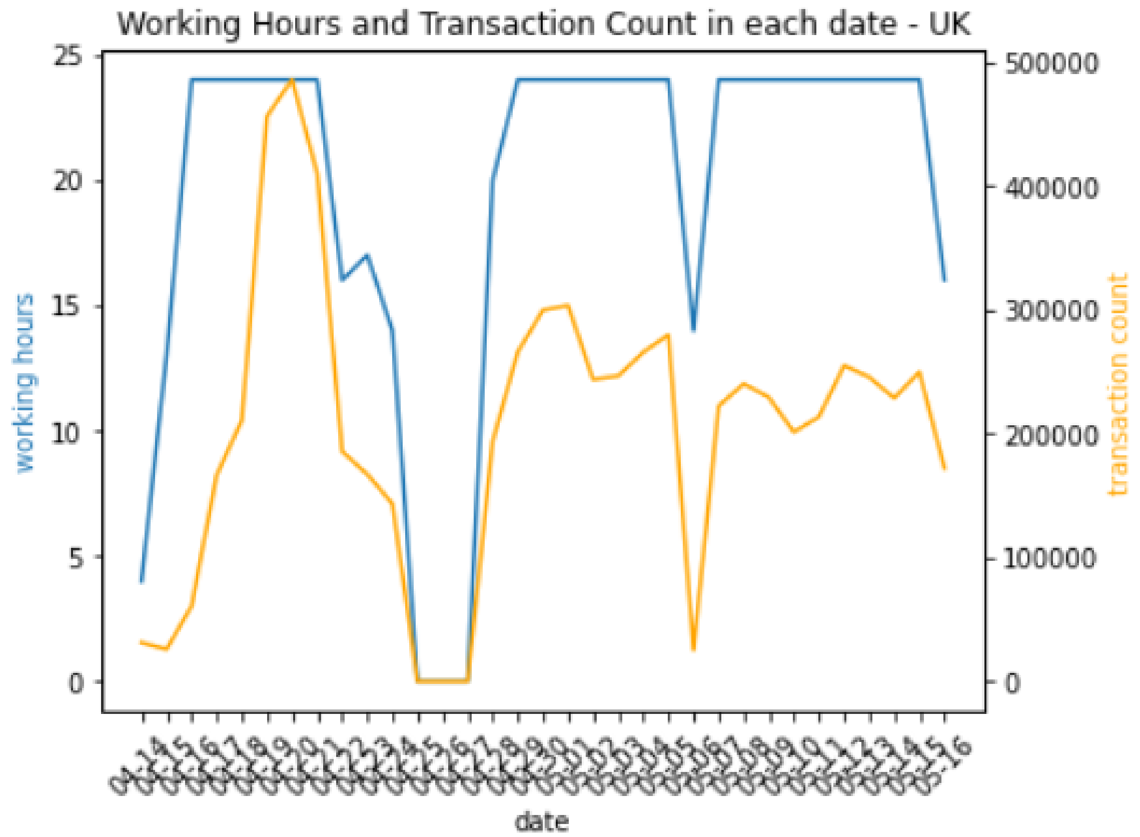


Figure 3.14: Transaction Count and Working Hours in the UK

doi:10.6342/NTU202400040

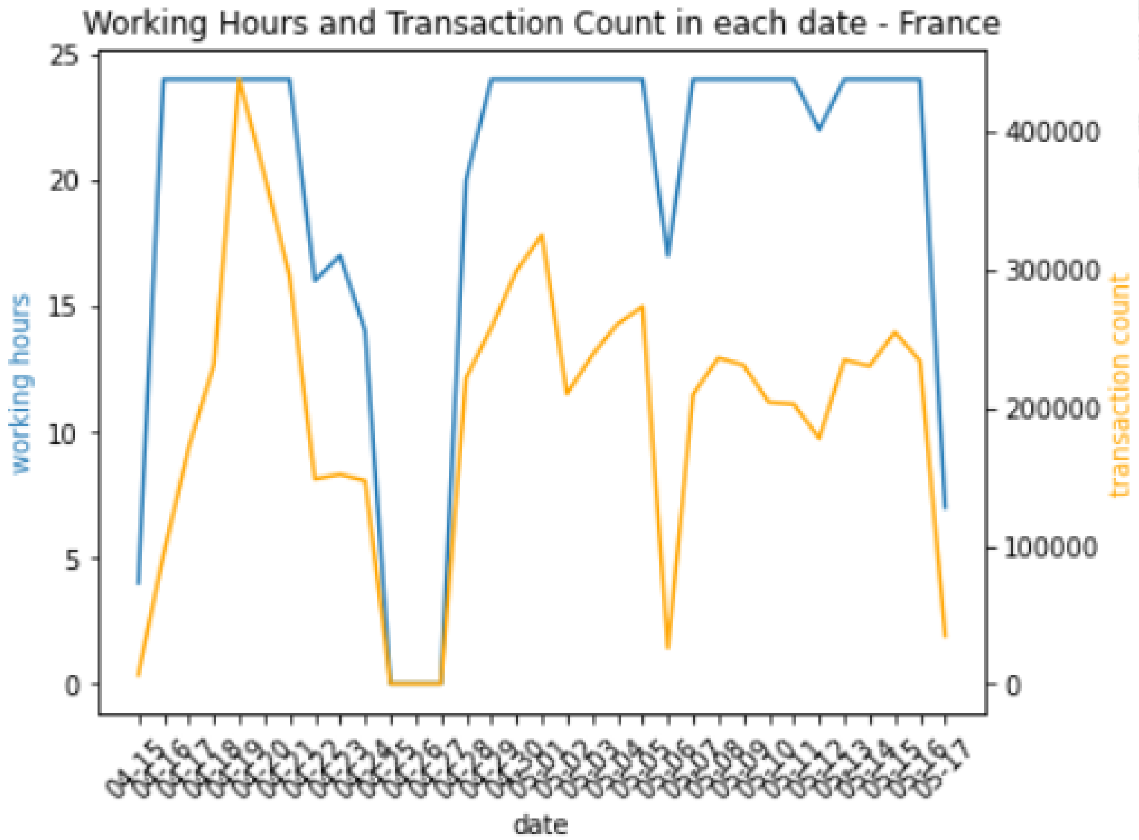


Figure 3.15: Transaction Count and Working Hours in France

doi:10.6342/NTU202400040

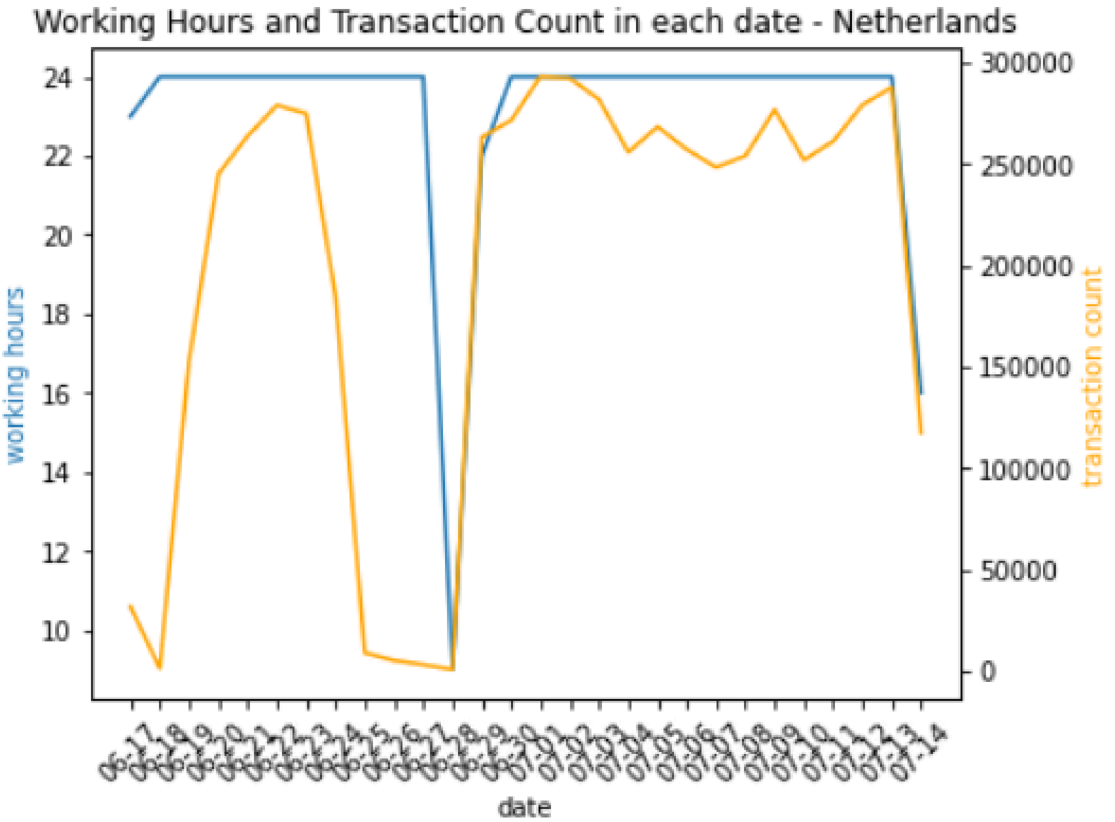


Figure 3.16: Transaction Count and Working Hours in the Netherlands

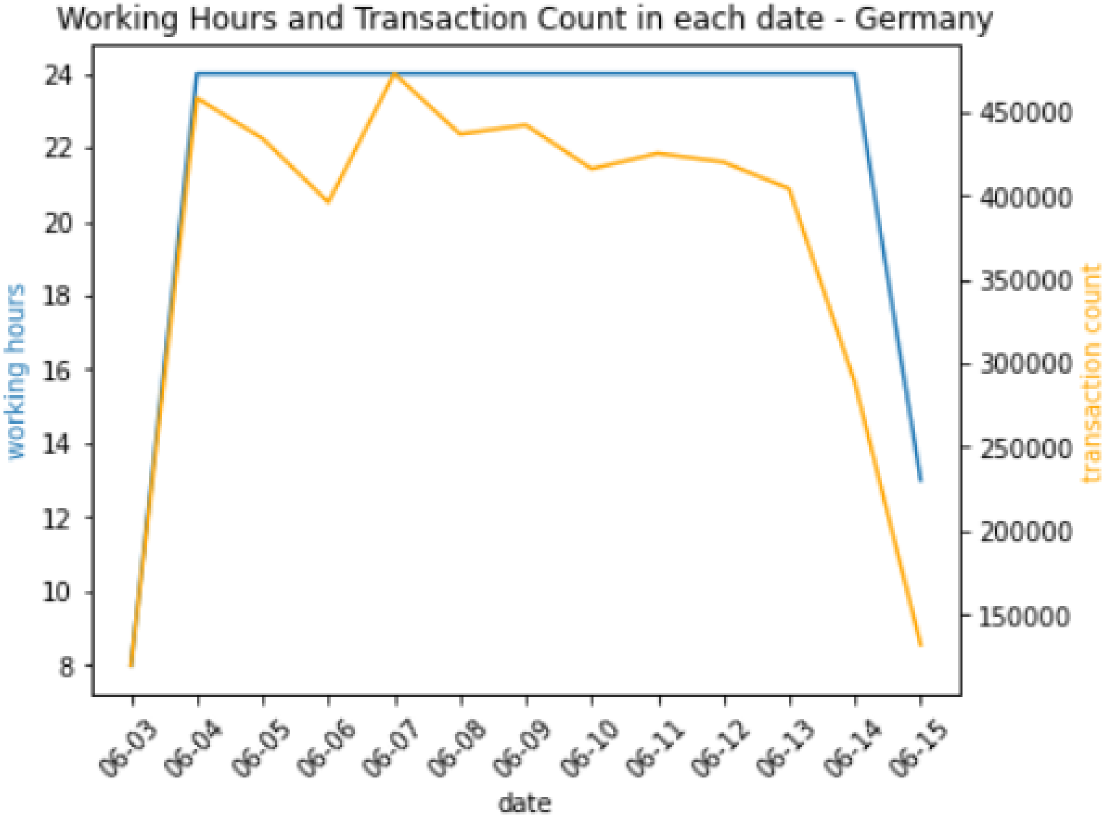


Figure 3.17: Transaction Count and Working Hours in Germany

doi:10.6342/NTU202400040

In the data of these four regions, there are six periods have the working hours equal to 24 for more than 7 continuous days. There are April 29 to May 05 (UK-0) and May 07 to May 15 (UK-1) in the UK, April 29 to May 05 in France, June 18 to June 27 (Netherlands-0), and June 30 to July 13 (Netherlands-1) in the Netherlands, and June 04 to June 14 in Germany. In Chapter 6, I will use the CJS model to estimate the number of servers in these 6 periods.

doi:10.6342/NTU202400040

Chapter 4 Clustering Method

Based on the observation in Chapter 3, this chapter proposes the methodology of clustering and the evaluation of the clustering results.

4.1 Number of Servers Every Hour in US-1

In the beginning, I checked the number of servers every hour in US-1. In the Fig. 4.1, the blue line is the number of IPs in the corresponding hour of all days, and the orange line is the transaction count in every hour of all days. To be noticed, the number of IPs indicated in the blue line is the number of distinct IPs, not the ’average’ number of IPs in these hours. The blue line has a negative correlation with the orange line. This may be caused by the mechanism of the Usher, which would block our crawler if we kept probing during the peak hours. The peak of the number of IPs is from ’00’ to ’01’, which is mostly close to the number of the ’baseline’ servers in one day. That may be the reason why the CJS model has the best estimation accuracy with sampling at 12 am, which is shown in the next chapter.

Fig. 4.2 to Fig. 4.7 is the number of IPs and ’new IP’ in every hour from May 7 to May 17. The orange lines are the numbers of IPs, and the blue lines are the numbers of the ’new IP’. ’New IP’ is the IP that shows in this hour but not in the previous hour. In the

doi:10.6342/NTU202400040

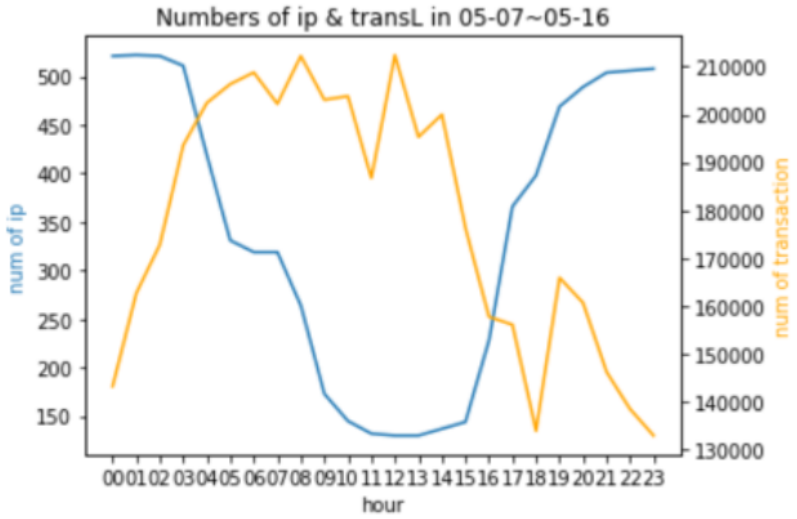


Figure 4.1: Number of IPs and Transaction Counts in US-1

Fig. 4.2, the peaks of the number of servers are in ’00’ to ’02’ and ’23’, and the bottom is in ’07’ to ’15’ in the data of May 7. Similar to Fig. 4.1, the data in US-1 has a relatively low number of IPs, approximately 100, in ’07’ to ’15’ and a relatively high number of IPs in ’23’ to ’02’.

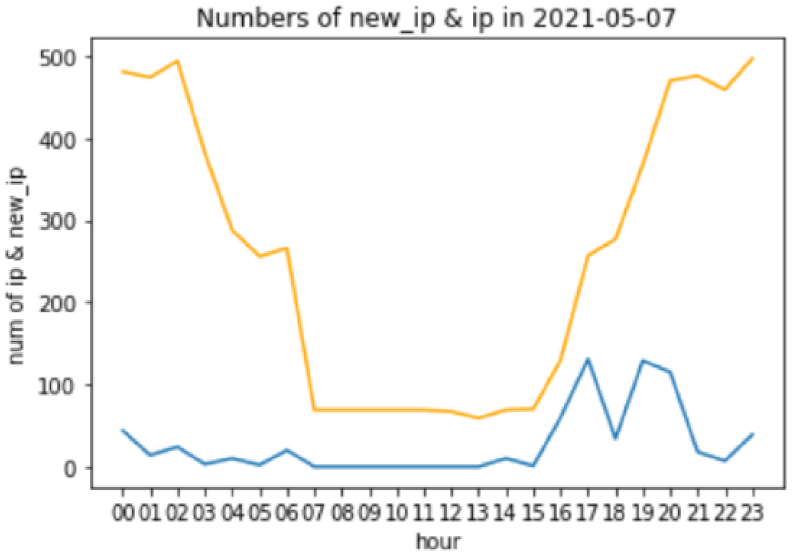


Figure 4.2: Number of ’new IPs’ and ’IPs’ on may-7

In the Fig. 4.2 to Fig. 4.7, the ’new IP’ in 9 of the 11 days has a peak in ’17’ to ’23’. Because many new IPs start to appear in ’17’ to ’23’, the peaks of the number of IPs are in ’23’ to ’02’. In the off-peak time, ’07’ to ’15’, the numbers of ’new IP’ are almost zero,

doi:10.6342/NTU202400040

which shows that Twitch's CDN seldom changes the servers during the off-peak time. Fig. 4.8 shows the number of the distinct new IPs in the corresponding hour of US-1.

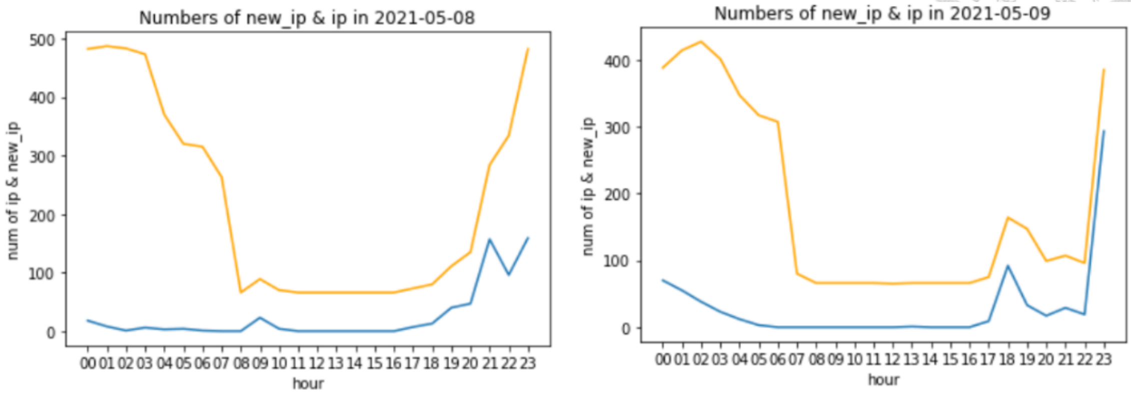


Figure 4.3: Number of 'new IPs' and 'IPs' on may-8 and may-9

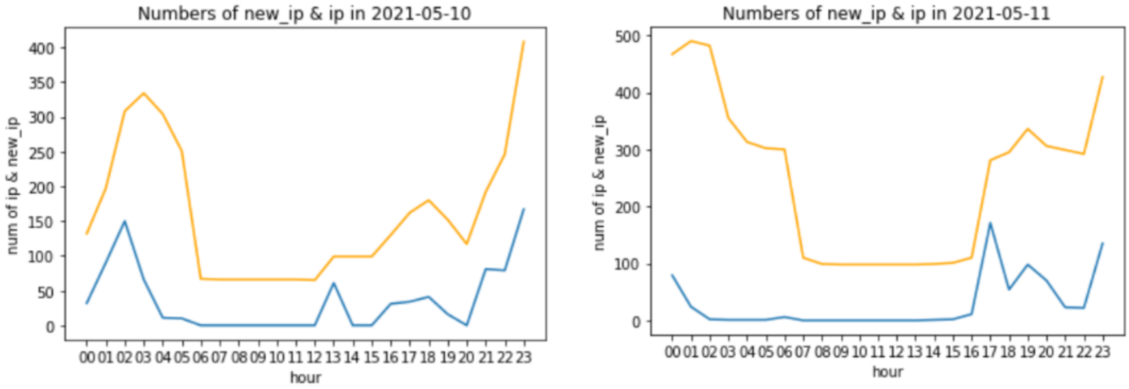


Figure 4.4: Number of 'new IPs' and 'IPs' on May 10 and May 11

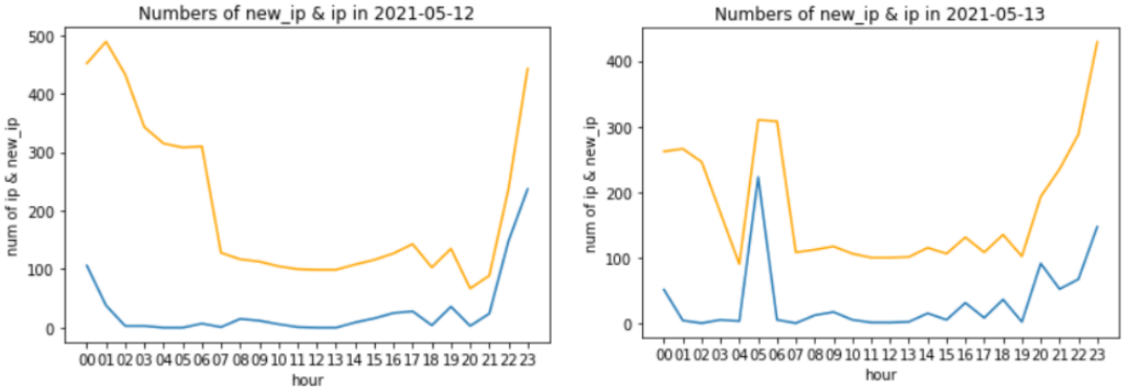


Figure 4.5: Number of 'new IPs' and 'IPs' on May 12 and May 13

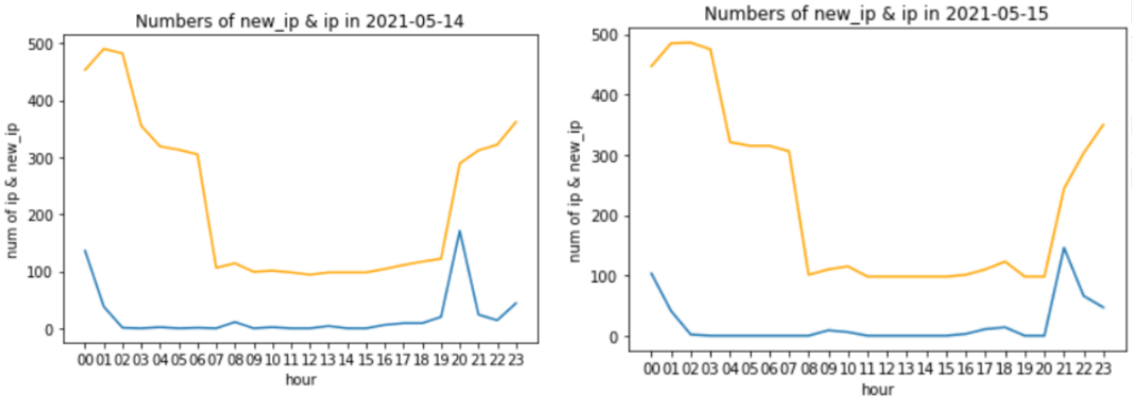


Figure 4.6: Number of ’new IPs’ and ’IPs’ on May 14 and May 15

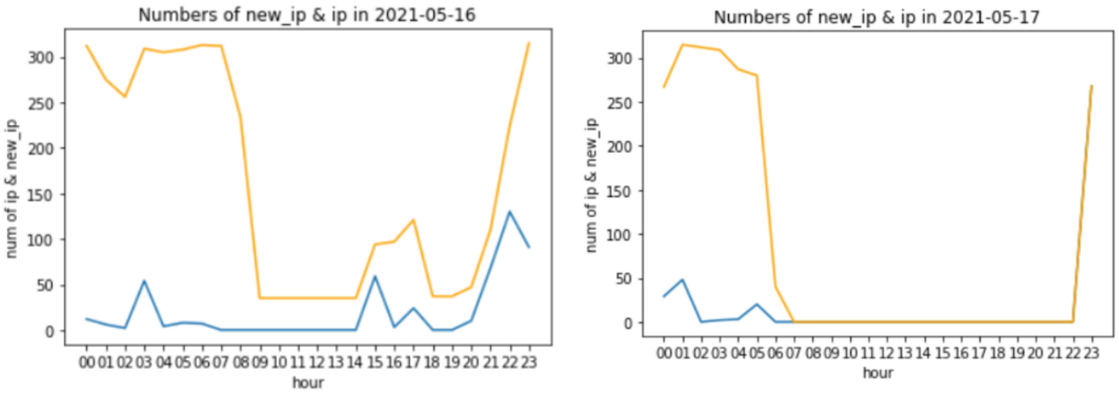


Figure 4.7: Number of ’new IPs’ and ’IPs’ on May 16 and May 17

The peak in the Fig. 4.8 is ’17’. Over 70% of new servers in Fig. 4.8 concentrate in ’16’ to ’21’. Thus, the sample time in the CJS model should choose after the peak of the new IP to fit the ’baseline’, which is the total number of servers observed in one day.

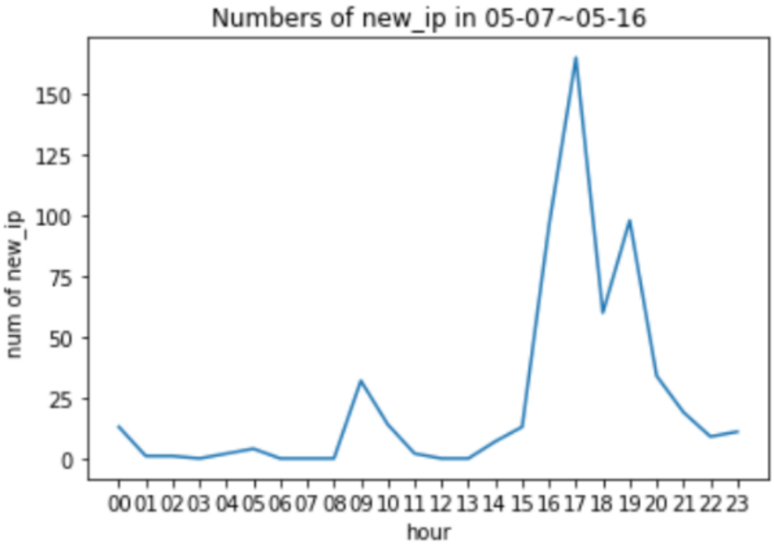


Figure 4.8: Number of 'new IPs' in US-1

4.2 K-Means Clustering of US-1 - Preliminaries

In the previous chapter, I show that each server in the same 24 subnet mask has a similar "hour-count chart". Therefore, I select the transaction counts in the specific hours of the day as attributes to do clustering. Taking the Fig. 4.9 as an instance, I divide 24 hours into three periods with US-All. The blue line is the number of IPs, and the orange line is the number of transaction counts. In this case, each server would have three attributes, the transaction count in hour period 1, the transaction count in hour period 2, and the transaction count in hour period 3. Next, the servers would use these three attributes to do clustering.

4.2.1 Clustering Algorithm

K-means is one kind of flat geometry clustering for general purposes. It will separate data into k groups with equal variance. In Fig. 4.10, the XYZ labels, 00 $\sim$ 07, 08 $\sim$ 15, and 16 $\sim$ 23 represent the transaction counts in these hour periods. I choose three as the

doi:10.6342/NTU202400040

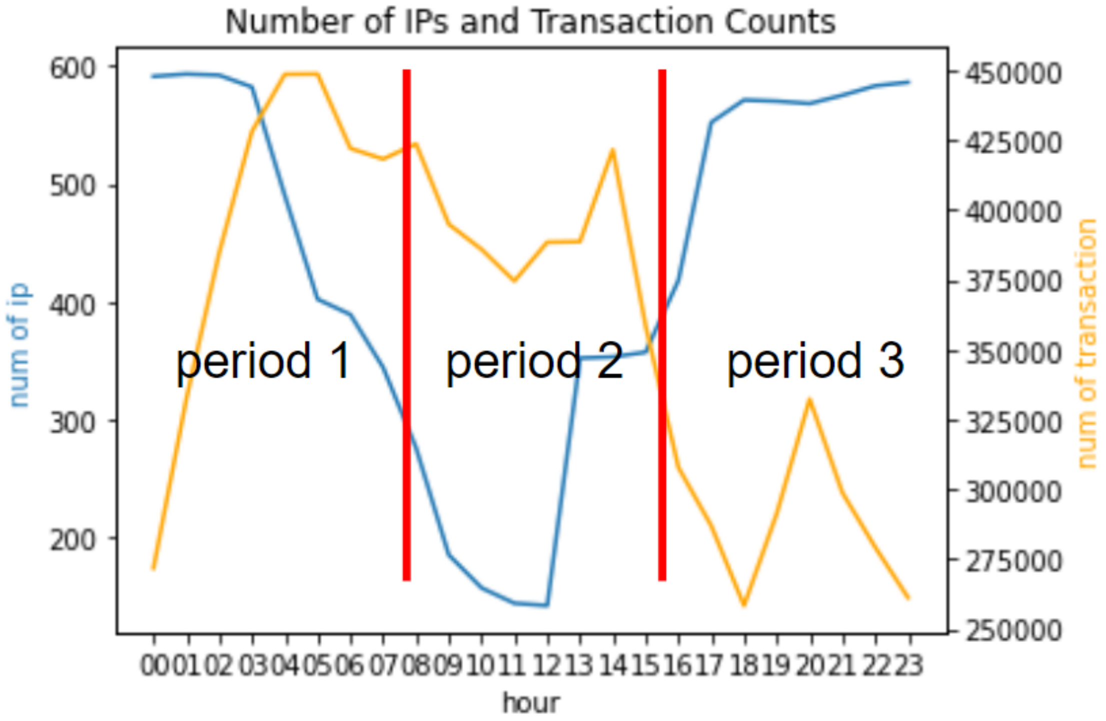


Figure 4.9: Number of IPs and Transaction Count in Every Hour - US-All

number of clusters, and then the space between the clusters is wide enough that k-means can successfully separate the data. Therefore, I choose k-means to do clustering. We can clearly observe that the orange cluster is the servers that have the largest counts in all three-hour periods. The green cluster stands for the servers mainly shown in one one-hour period. The servers in the blue cluster have the least counts in all three-hour periods.

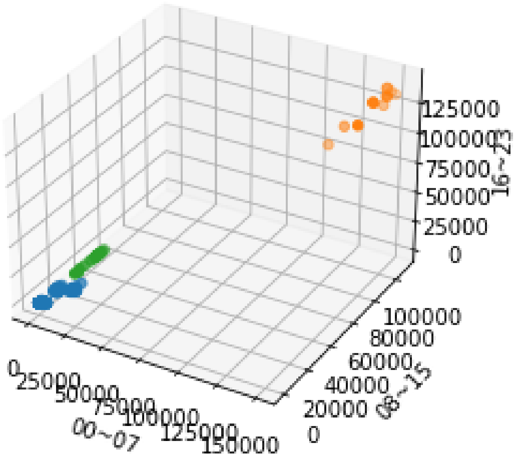


Figure 4.10: Clustering Result with K-means

doi:10.6342/NTU202400040

4.2.2 Dimension Reduction

Dimension reduction can avoid the curse of dimension and enable the data to show in two-dimensional images. I use principal component analysis (PCA), a linear transformation technique, to reduce data into two-dimensional space. Next, I deploy k-means to do clustering on the two-dimensional data, and the result is in Fig. 4.11. The overall result is similar to the clustering result without dimension reduction. In this picture, the inter-cluster variance is much larger than the intra-cluster variance so we can confirm that the clustering result is fine.

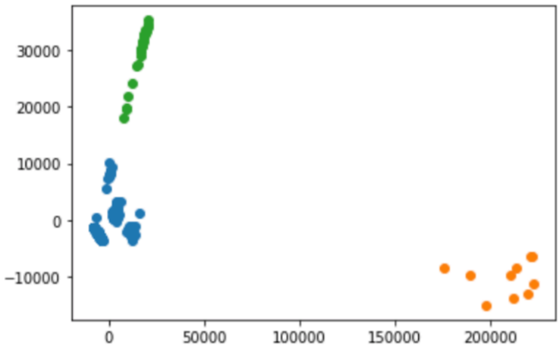


Figure 4.11: Clustering Result with PCA

4.2.3 Internal Evaluation

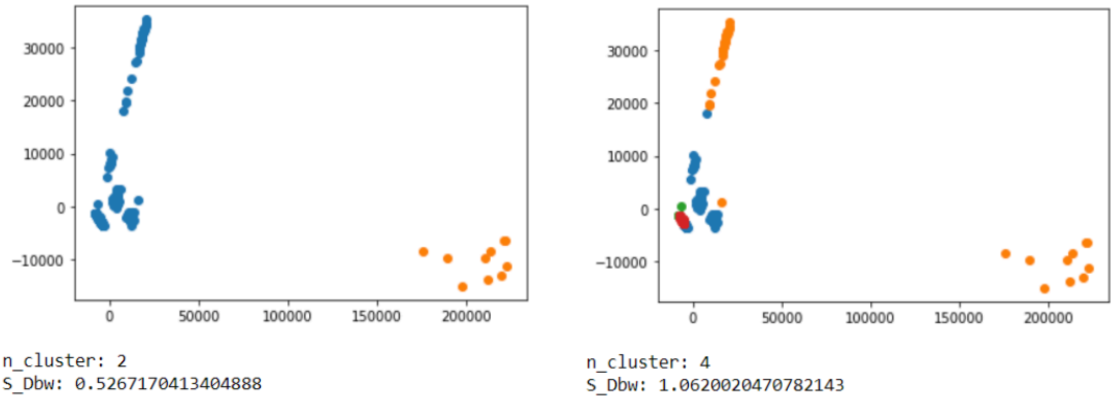
There are two kinds of clustering analysis, internal evaluation, and external evaluation. In external evaluation, the clustering is compared to the ”ground truth”, which does not exist in our experiment. Thus, I use one of the best internal evaluation methods, the S_dbw index, to evaluate the clustering results.

Firstly, I examine the clustering result without PCA (Fig. 4.10) and with PCA (Fig. 4.11). The S_Dbw scores of the clustering result with PCA (0.3601) are slightly better than the result without PCA (0.3673). However, the attributes after dimension reduction

doi:10.6342/NTU202400040

lose their original meaning, which means we could not directly get the properties of the clusters through the figure.

Next, I evaluate the different clustering results with different numbers of clusters, a parameter in k-means. As the below images Fig. 4.12, when the number of clusters is 2 or 4, the S_Dbw scores show that the results are much worse than the clustering result with a number of clusters equal to 3. If the number of clusters is set to be larger than 4, the clustering result will get even worse.



```
n_cluster: 2
S_Dbw: 0.5267170413404888
```

```
n_cluster: 4
S_Dbw: 1.0620020470782143
```

Figure 4.12: Number of Clusters equal 2 or 4

To explore more possible combinations of the features, I try to use counts in different hour periods to be the attributes. As the Table 4.1, I divide 24 hours into n hour periods with different ”slide_hour”. ”n_period” represents the number of hour periods in 24 hours, and ”slide_hour” indicates sliding windows for hour periods. For example, ”n_period:3, slide_hour:0” means 3-hour periods are ’00’ to ’07’, ’08’ to ’15’, and ’16’ to ’23’, and ” n_period:3, slide_hour:3” means 3 hour periods are ’03’ to ’10’, ’11’ to ’18’, and ’19’ to ’02’. The total number of combinations with the number of hour periods equaling 3 is 7.

After computing the transaction counts in each hour period for all IPs, I use PCA to reduce the dimensions to 2, do clustering, and compute S_Dbw scores based on the attributes after PCA, which is shown in Table 4.1. Among all the results, ”n_period:3, slide_hour:2” has the best performance in the S_Dbw score.

After I check all the results, I find that for the result with n_period=3, ”n_period:3, slide_hour:0”, ”n_period:3, slide_hour:1”, ”n_period:3, slide_hour:2”, and ”n_period:3, slide_hour:7” are all same, which means the servers’ distribution of 3 clusters are same in these clustering results. So are the ”n_period:4, slide_hour:0”, ”n_period:4, slide_hour:1”, ”n_period:4, slide_hour:2”, ”n_period:4, slide_hour:3”, and ”n_period:6, slide_hour:0”, they all have the same servers’ distribution of 3 clusters to the result of ”n_period:3, slide_hour:0”. In actuality, all the clustering results with S_Dbw score less than 0.37 have the same servers’ distribution of 3 clusters.

In the end, I deploy the clustering result of ”n_period:3, slide_hour:0” to the CJS model with heterogeneity, which is shown in the next chapter.

doi:10.6342/NTU202400040

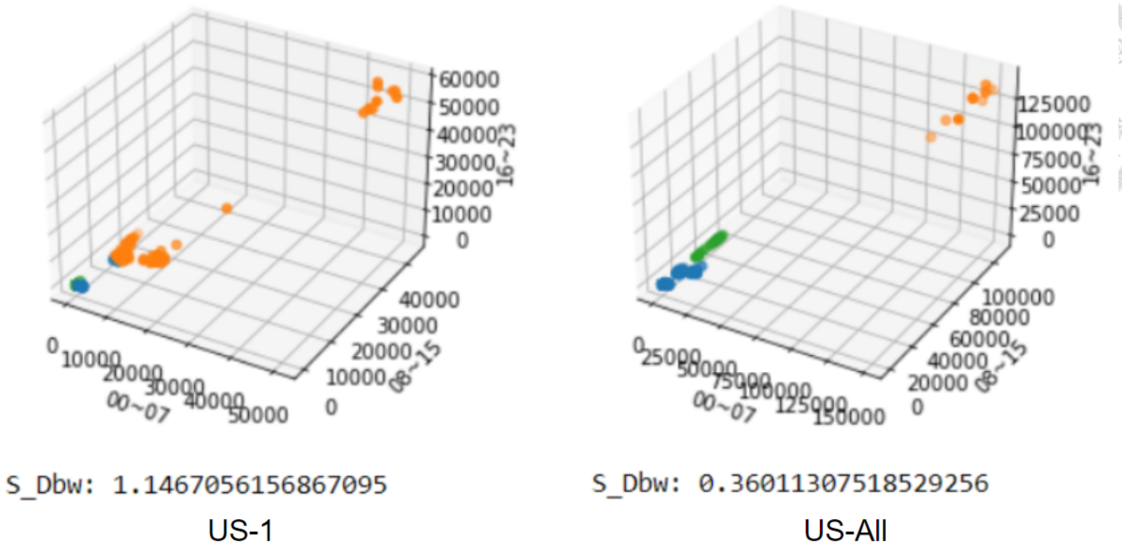
n_period	slide_hour	S_Dbw score
3	0	0.3601
3	1	0.3549
3	2	0.3504
3	3	0.4792
3	4	0.5270
3	5	0.4227
3	6	0.3818
3	7	0.3635
4	0	0.3946
4	1	0.3584
4	2	0.3591
4	3	0.3598
4	4	0.4601
4	5	0.4480
6	0	0.3586
6	1	0.4549
6	2	0.4653
6	3	0.4695

Table 4.1: S_Dbw Scores with n_period = 3, 4, 6 (n_cluster=3)

4.3 Alternative Dataset

I use k-means to do clustering on the data in US-1. In the beginning, I set the n_cluster to be 3 and the features to be ”n_period:3, slide_hour:0”. As the Fig. 4.13, compared to the clustering result in the previous section, the S_dbw score with US-1 is more than 3 times larger than the clustering result with US-All.

doi:10.6342/NTU202400040



US-1

US-All

Figure 4.13: Clustering Results in US-1 and US-All (n_clusters=3)

I try to do clustering with different numbers of clusters. I use the feature ”n_period:3, slide_hour:0”, and do k-mean without a fixed random seed 10 times for each n_clusters = 2 to 8. The S_dbw scores are shown in Table. 4.2. ”stdev of S_Dbw scores” in Table. 4.2 represents the standard deviation of the 10 S_Dbw scores of each n_clusters. For the n_clusters = 2 to 8, the S_Dbw score tends to be better as the number of clusters gets larger. When the number of clusters equals to 8, it has the best mean S_Dbw score, 0.3398, among these results.

n_clusters	mean S_Dbw scores	stdev of S_Dbw scores
2	1.1379	0.2645
3	0.8343	0.3407
4	0.5669	0.1912
5	0.4849	0.2382
6	0.4583	0.1447
7	0.3792	0.0900
8	0.3398	0.1085

Table 4.2: Clustering Results with Number of Clusters = 2 to 8

To find out the reason why the best number of clusters are different between the clustering results with US-All and with US-1, I dig into the subnets in each cluster when the n_cluster=3 first.

cluster	US-1	US-All

0	99.181.96(10), 192.16.65.(48)
	52.223.227(31), 52.223.228(25)
		99.181.96(10)

1	52.223.226(67), 52.223.244(41)
	99.181.97(79), 192.16.65(13)
		52.223.227(3), 52.223.228(28)

2	52.223.224(19), 52.223.225(34)
	52.223.226(1)
	52.223.228(14), 52.223.229(12)
	52.223.243(97)
	52.223.246(35), 52.223.247(2)
	52.223.248(1)
	99.181.65(1)
	99.181.96(6), 99.181.97(2)
		52.223.224(19), 52.223.225(34)
		52.223.226(74), 52.223.227(30)
		52.223.228(21), 52.223.229(12)
		52.223.243(97), 52.223.244(71)
		52.223.246(35), 52.223.247(3)
		52.223.248(1)
		99.181.65(1)
		99.181.96(6), 99.181.97(83)
		192.16.65(82)

Table 4.3: Clustering Results with Different Data - n_clusters=3

Table 4.3 shows the subnets in each cluster of a clustering result with US-1 and the result with US-All when the n_cluster=3, n_period=3, and slide_hour=0. The number in the brackets is the number of servers in the subnet of the cluster. In cluster-0 of the clustering result with US-All, the subnet is 99.181.96, which has the servers with the largest transaction counts in all 3 periods.

doi:10.6342/NTU202400040

subnet	US-1	US-All	ratio of US-1/US-All
99.181.96(10)	449686	1428415	0.315
	378697	989846	0.383
	513367	1291581	0.397
192.16.65(48)	704711	876326	0.804
52.223.227(31)	1117383	1801669	0.620
52.223.228(25)	561842	682912	0.823

Table 4.4: Transaction Counts in 3-period of the Servers from cluster-0

However, in the clustering result with US-1, the subnet 192.16.65, 52.223.227, and 52.223.228 are added into cluster-0. This is because the transaction counts in the 3-period of these 3 subnets are relatively closer to the transaction counts of 99.181.96 in US-1. Table. 4.4 shows the total transaction count of 3-period in US-1 and US-All for the servers from cluster-0 (US-1) in Table. 4.3. For the servers in cluster-0 subnet 99.181.96, the ratio of the 3-period transaction count in US-1 and US-All is [0.315, 0.383, 0.397]. However, for the servers in cluster-0 subnet 192.16.65, 52.223.227, and 52.223.228, this ratio is [0.804, 0.620, 0.823], which means the capture probability relatively increases compared to the servers in cluster-0 subnet 99.181.96. In the end, k-means divides the servers in these subnets into cluster-0 with servers in 99.181.96 in US-1.

Secondly, I dig into a clustering result with US-1 when the n_cluster=8 with the same features n_period=3 and slide_hour=0. The subnets in each cluster are shown in Table 4.5.

doi:10.6342/NTU202400040

cluster	subnet(number of IPs)
0	52.223.226(26), 52.223.244(41)

1	52.223.225(4), 52.223.226(1)
	52.223.243(51), 52.223.247(2) 52.223.248(1)
	99.181.65(1), 99.181.96.(6)

2	52.223.228(25), 52.223.227(3), 192.16.65(61)

3	52.223.224(4), 52.223.225(2)
	52.223.243(46), 52.223.246(28)

4	52.223.224(14), 52.223.225(28), 52.223.228(14)
	52.223.229(12), 52.223.246(7), 99.181.97.(1)

5	52.223.227(28), 99.181.96.(1)
6	52.223.224(1), 52.223.226(41), 99.181.97.(80)
7	99.181.96.(9)

Table 4.5: Subnets in Each cluster - n_clusters=8

Compared to the clustering result in US-All (n_cluster=3), 9 out of 10 servers in subnet 99.181.96 still have the largest transaction counts in all 3-hour periods and be divided into an independent cluster. The leftover server in subnet 99.181.96 has a relatively lower value of transaction count in 3-hour periods, this makes it be divided into another cluster.

In Fig. 4.14, it is the 3D plot of the clustering result with US-1 when the n_cluster=8. Between the green cluster(cluster-2) and the blue cluster(cluster-0), there is an obvious gap in the y-axis, transaction count in hour period 2 ('08' to '15').

doi:10.6342/NTU202400040

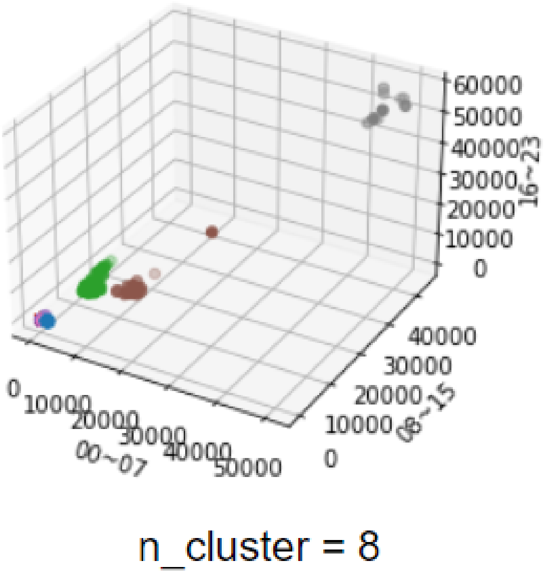


Figure 4.14: K-Means with US-1

Table 4.6 shows the average 3-period transaction counts of each cluster of Table 4.5 in US-1 and US-All. In cluster-0, cluster-4, and cluster-6, the average transaction counts in hour period 2, largely decrease, which is only 1.7%, 5.1%, and 1.5% of the average number in US-All. On the contrary, in cluster-2 and cluster-5, the average transaction counts in hour period 2 of US-1 still remain 58% and 80% of US-All. As a result, there is an obvious gap between [cluster-2, cluster-5] and [cluster-0, cluster-4, cluster-6], which leads to the servers being divided into different clusters in US-1.

doi:10.6342/NTU202400040

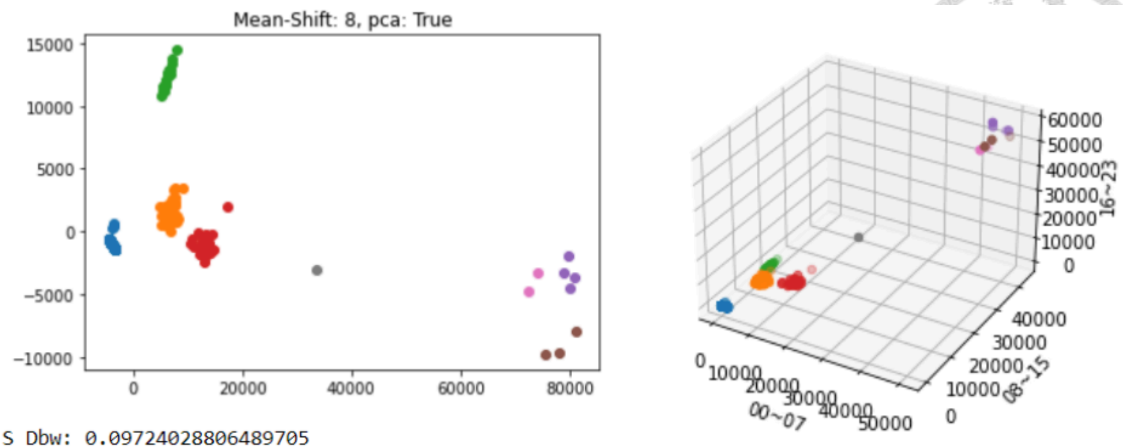
cluster	US-1	US-All
0	[1585, 2, 238]	[3615, 153, 1054]
1	[40, 0, 12]	[40, 0, 16]
2	[4378, 10598, 4707]	[5624, 18249, 5700]
3	[150, 0, 52]	[150, 0, 69]
4	[325, 52, 97]	[475, 1032, 244]
5	[14082, 9723, 8203]	[19878, 12157, 12240]
6	[1008, 1, 133]	[3020, 106, 816]
7	[47243, 40019, 54505]	[144784, 100580, 131880]

Table 4.6: The Average 3-Period Transaction Count in US-1 and US-All

4.4 Alternative Clustering Method

I also use mean shift, a centroid-based algorithm, to find the best number of clusters on US-1. Fig. 4.15 shows the clustering results by mean-shift with the feature, " n_periods:3, slide_hour:0". The number of clusters decided by the mean-shift algorithm is 8. The centers of the clusters in the 3D plot of Mean-Shift, the right figure Fig. 4.15, is shown in Table. 4.7. The S_dbw score is very low, only 0.0972, which is less than the half value of the mean S_dbw score in the clustering results with k-means when the n_cluster is also 8.

doi:10.6342/NTU202400040



```
S_Dbw: 0.09724028806489705
```

Figure 4.15: Mean-Shift with n_periods=3 and slide_hour=0

cluster	color	center location
cluster_0	blue	[654, 11, 109]
cluster_1	orange	[5778, 7831, 6078]
cluster_2	green	[1332, 16628, 1723]
cluster_3	red	[13711, 9409, 7681]
cluster_4	purple	[46878, 42820, 55720]
cluster_5	brown	[49086, 37101, 55913]
cluster_6	pink	[45211, 38794, 49966]
cluster_7	gray	[24496, 18525, 22820]

Table 4.7: Centers of the clusters in Mean Shift

4.5 Alternative Clustering Metrics

In the next chapter, I show that S_Dbw is not a good metric for the CJS model. Therefore, I develop additional metrics to evaluate clustering results, ’Std/Avg’, ’mean Std/Avg’, ’cluster size’, and ’min cluster size’.

The definition of ’Std/Avg’ is shown in Eq. 4.1. It is calculated by a series of sample numbers in one cluster. For example, for a clustering result in US-0 (April 29 to May 05), its 1st cluster has 100 servers in total. The numbers of 1st cluster's servers in the sample hour from April 29 to May 05 are [80, 60, 70, 85, 60, 70, 85], and each number in the series is the sample number on one date. I use this series of sample numbers to calculate the standard deviation (std) and average (avg). A higher value of 'Std/Avg' represents the numbers of sample servers fluctuating more widely.

To be notified, 'Std/Avg' is used for each cluster, not for an entire clustering result. For a clustering result with n_cluster = 3, there would be 3 'Std/Avg' for each cluster. I use 'mean Std/Avg' for an entire clustering result, which is computed by the mean value of Std/Avg from all clusters in that clustering result.

'Cluster size' is the number of IPs in one cluster, and 'min cluster size' is the size of the minimum cluster in one clustering result. I developed these metrics because a too-small sample size would lead to a large estimation bias in MLE [13]. To avoid such large bias in the MLE-based CJS model, I add 'cluster size' for each cluster and 'min cluster size' for each clustering result.

$$Std/Avg = Standard\ Deviation/Average\ Number \tag{4.1}$$

4.6 Random Clustering - Baseline

To be the control group for other clustering results, I randomly divide the servers into n clusters with roughly equal size to generate 20 clustering results for each n = 2 to 8.

The average S_Dbw scores of the random clustering are much larger than k-means and mean shift clustering. The S_Dbw scores of the random clustering results range from

1.6 to 2.8 in US-0. The S_Dbw scores of the random clustering results range from about 1 to 4.5 in US-1. The details of the random clustering results are shown in chapter 5.3, CJS with Random Clustering Results.

doi:10.6342/NTU202400040

doi:10.6342/NTU202400040

Chapter 5 CJS Estimation Error

The goal of my data mining and clustering works is to improve the accuracy of the CJS model by enabling heterogeneity in servers and avoiding high computation overhead in the CJS model. In this section, I will show the (1) estimation result of the CJS model done by Jill [22] and me and (2) the analysis of the CJS model results done by me.

5.1 Population Estimation of CJS Model - Preliminaries

In the chapter 5.1, we deploy the CJS model on US-1 (May 07 to May 16), which has the longest continuous working hour equal to 24 in the US data. We use the k-means clustering result,"n_periods=3, slide_hour=0", with US-All, which has the same servers' distribution in 3 clusters as all the results with S_Dbw scores less than 0.37 in Table. 4.1, to deploy on the CJS model. Also, I use the mean-shift result with US-1, which is shown in Fig. 4.15, to deploy on the CJS model. In the mean-shift result, servers are divided into 8 clusters.

5.1.1 Estimation Error Rate

Jill used the clustering result done by me in above to deploy on the MLE-CJS model with heterogeneity. Jill defines the error rate of the CJS model on one date as the following

doi:10.6342/NTU202400040

Eq. 5.1.

$$error\ rate(\%) = (baseline - estimation\ number)/baseline * 100\% \tag{5.1}$$

In the Eq. 5.1, ”baseline” is the number of servers the crawler discovered in the whole day, which is the best knowledge of the ground truth of the total number of CDN servers. Because the CJS model often does not converge well on the first and last two days, the error rate of each result is the mean value of the error rate in estimation dates without the first and last two days. With the different sample hours, the estimation error rate of the CJS model is shown in Table 5.1.

error rate	12 am	6 am	12 pm	6pm
no clustering	0.82%	35.38%	81.64%	55.81%
k-means	1.92%	35.38%	X	54.87%
mean-shift	1.87%	37.52%	X	37.29%

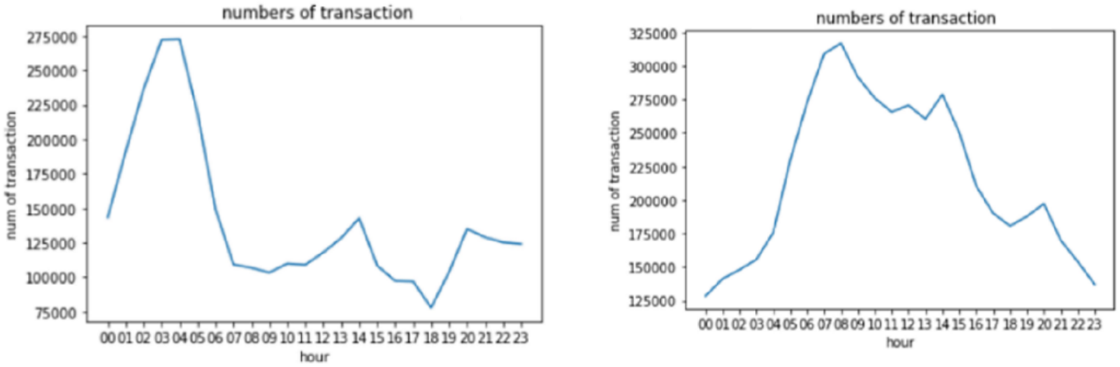
Table 5.1: Estimation model - Error Rate

All three estimation models have the lowest error rate when the sample time is 12 am. This result can be explained in Fig. 4.1. The number of IPs has the maximum value when the time is ’00’. On the contrary, the number of IPs has the minimum value when the time is ’12’. If the sample is in the ’12’ o’clock, some IPs may never show up. This could be the reason why the CJS model has the worst error rate when sampling in ’12’.

For the k-means and mean-shift results, the CJS models fail to converge when the sample hour is 12 pm. It may be caused by the servers from some clusters not showing in the first few days, this will make the CJS model unable to calculate the first few days capture probability in these clusters.

5.1.2 Dig into Cluster 2 in K-Means

To discover the reason why the estimation model with k-means (n_cluster=3) has a higher error rate than the model without clustering when the sample hour is 12 am, I dig into cluster-2 in the k-means result which contains most servers with the lowest transaction counts as shown in the right plot of Fig. 4.13. The number of servers in cluster-2 is 570, which accounts for 93% of total servers. The Fig. 5.1 shows the hour-count distribution plot in cluster-2 and cluster-0 + cluster-1.



cluster 2

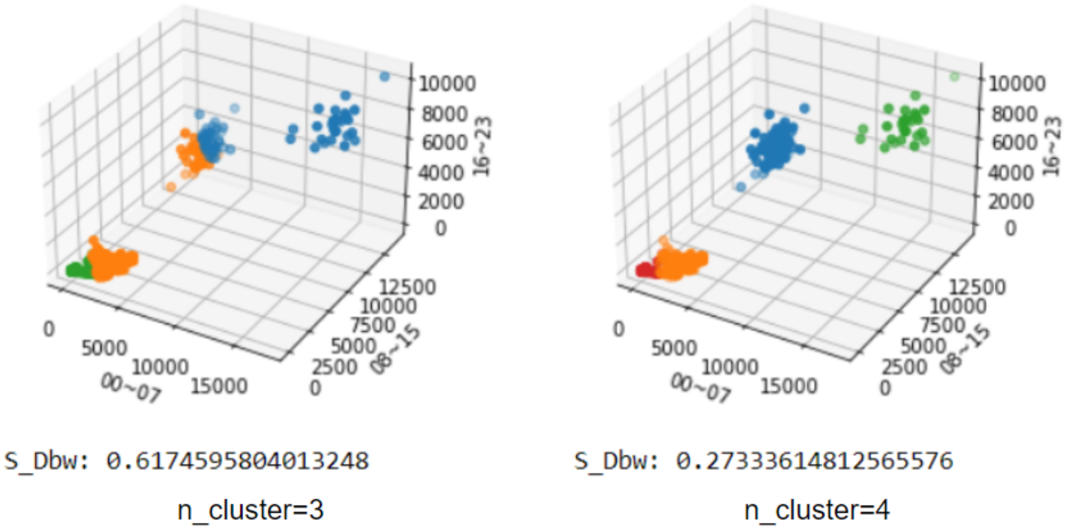
cluster 0 + cluster 1

Figure 5.1: Transaction Counts in Different Clusters - K-Means with US-All

I do further clustering to cluster-2. As the images in Fig. 5.2, I divide the servers in cluster-2 into 3 or 4 clusters. When the number of clusters is 4, the clustering gets a better result in the S_Dbw score. It indicates that in cluster-2, one can further divide it into several clusters. As for the green cluster in the right figure of Fig. 5.2 (n_cluster=4), which represents the servers with the highest transaction counts in all 3 periods from cluster-2, the subnet of the servers is ’52.223.227’ (28).

The Table. 5.2 shows the subnets in 3 clusters. The subnet of cluster-0, which represents servers with the highest transaction counts in 3 periods, is ’99.181.96’, and the subnet in cluster-1 is ’52.223.227’, ’52.223.228’. I discovered that the servers with higher

doi:10.6342/NTU202400040



S_Dbw: 0.6174595804013248

n_cluster=3

S_Dbw: 0.27333614812565576

n_cluster=4

Figure 5.2: Clustering Again in K-Means Cluster-2

transaction counts, including cluster-0, cluster-1, and the green cluster (right plot in Fig. 5.2) in cluster-2, all come from the subnets - ’52.223.227’, ’52.223.228’, and ’99.181.96’. As for the servers from other subnets, their transaction counts are all lower than the servers from cluster-0, cluster-1, and the green cluster in cluster-2.

doi:10.6342/NTU202400040

cluster	subnet	number of servers
cluster-0	99.181.96	10
cluster-1	52.223.28	28
	52.223.27	3
cluster-2	52.223.228	21
	52.223.227	30
	52.223.226	74
	52.223.225	34
	52.223.224	19
	52.223.229	12
	52.223.243	97
	52.223.244	71
	52.223.246	35
	52.223.247	3
	52.223.248	1
	99.181.97	83
	99.181.96	6
	99.181.65	1
	192.16.65	82

Table 5.2: Subnets in Each Cluster - K-Means with US-All

5.2 CJS with Multiple K-Means Clustering Results

In this section, I discover the relationship between the CJS model and clustering results by using multi-clustering results with the data in US-0 and US-1 to deploy on the CJS model.

5.2.1 Clustering Result with US-1 - May 07 to May 16

To inspect the relationship between the CJS model and clustering results, I remove the random seed in k-means and do clustering 10 times for each number of clusters equal to 2 to 8 with the feature ”n_period=3, slide_hour=0” with data in US-1. The sample hour of the CJS model is 12 am, which has the lowest error rate.

5.2.1.1 S_Dbw Score and Estimation Error Rate

There are 70 clustering results in total, the S_Dbw score and the error rate of the CJS model are shown in Fig. 5.3. All the CJS models with clustering have a higher error rate than the CJS model without clustering, 0.82%.

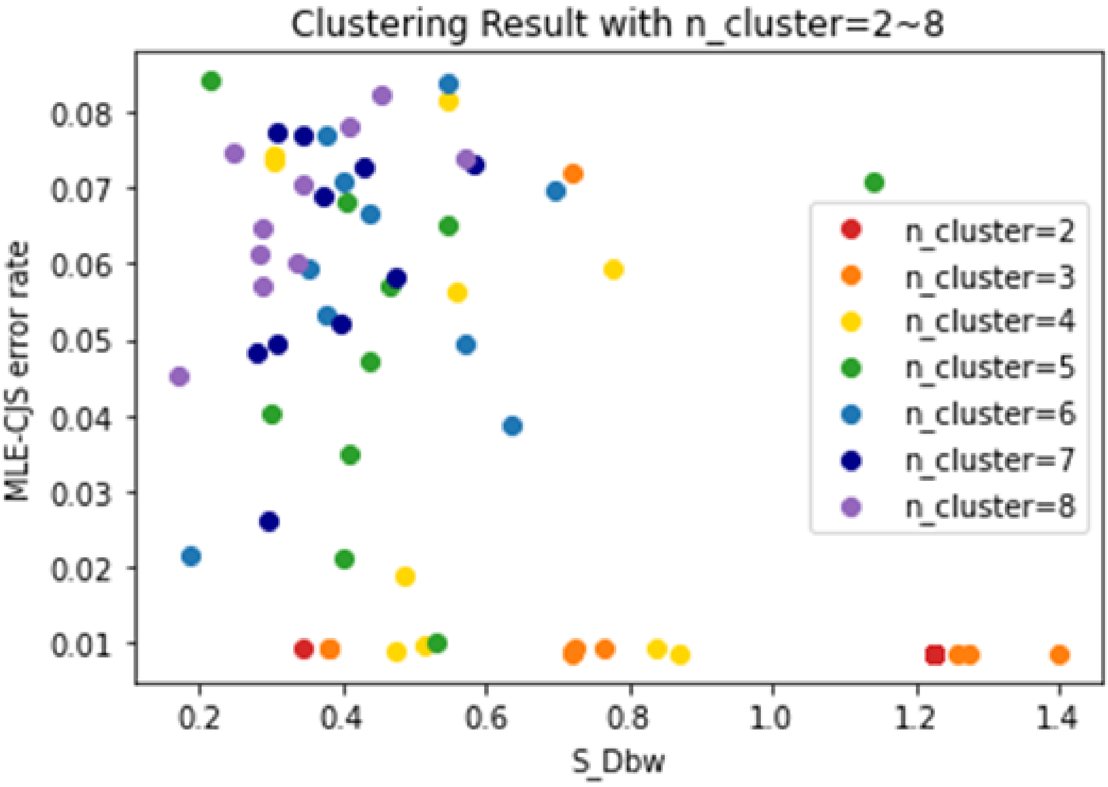


Figure 5.3: S_Dbw Score and Error Rate - US-1

In my previous expectation, a clustering result with a better S_Dbw score would tend

doi:10.6342/NTU202400040

to have a CJS model with a better error rate, which means the dots in Fig. 5.3 should close to a slant line from the button left to the top right. However, the result is far away from my previous anticipation. In fact, for those results with about 1% error rates, the number of clusters is mainly equal to 2, 3, and 4, which have S_Dbw scores ranging from 0.3 to 1.4.

Table 5.3 shows the correlation matrix of the number of clusters, S_Dbw score, error rate, and standard deviation of error rate every day (stdev). The correlation between the S_Dbw score and the error rate is -0.5717, which means a better S_Dbw score (lower value) tends to have a worse error rate (higher value). The correlation between the number of clusters and the S_Dbw score is -0.7144, and the correlation between the number of clusters and the error rate is 0.7193, which means a larger number of clusters tends to have a better S_Dbw score and a worse error rate.

	n_cluster	S_Dbw	error rate	stdev
n_cluster	1.0000	-0.7144	0.7193	0.6783
S_Dbw	-0.7144	1.0000	-0.5717	-0.5442
error rate	0.7193	-0.5717	1.0000	0.9943
stdev	0.6783	-0.5442	0.9943	1.0000

Table 5.3: Correlation Matrix - US-1

Table 5.4 shows the mean value of S_Dbw and the error rate with the number of clusters equal to 2 to 8. When the number of clusters is equal to 2, the mean S_Dbw score is 1.1379, however, the mean error rate is only 0.87%. When the number of clusters equals 8, the S_Dbw score improves to 0.3398, however, the error rate rises to 6.68%.

Fig. 5.4 shows the correlation of S_Dbw and error rate when the number of clusters equals 2 to 8. When the number of clusters equals 2 to 4, the correlation of S_Dbw and the error rate is negative. However, when the number of clusters equals 5 to 8, the correlation

n_cluster	S_Dbw mean	error rate mean
2	1.1379	0.0087
3	0.8343	0.0153
4	0.5669	0.0401
5	0.4849	0.0499
6	0.4583	0.0591
7	0.3792	0.0604
8	0.3398	0.0668

Table 5.4: Mean Value of S_Dbw and Error Rate with n_cluster=2 to 8

of S_Dbw and the error rate is positive, which means a better S_Dbw score tends to have a better error rate.

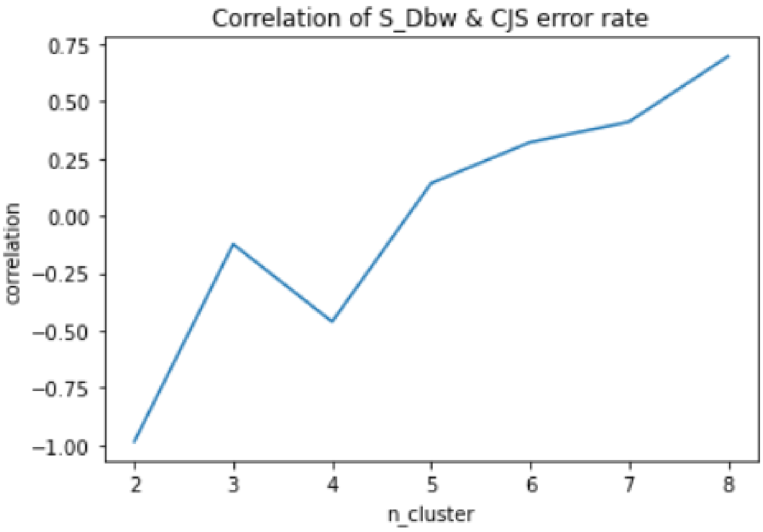


Figure 5.4: The Correlation of S_Dbw and Error Rate - US-1

5.2.1.2 Alternative Clustering Metrics

To find out the reason for a larger number of clusters tend to have a higher error rate, I use the alternative metrics, Std/Avg, mean Std/Avg, cluster size, and min cluster size, to inspect the CJS model results.

Fig. 5.5 shows the relationship between the Std/Avg and the error rate of each cluster in each clustering result. I use the term, cluster error rate, to represent the error rate of a cluster. In this figure, I remove one extreme value with cluster error rate > 2.5 in this

doi:10.6342/NTU202400040

plot to show other data more clearly. For the clusters with Std/Avg less than 0.3, all the clusters have cluster error rates less than 0.80%. However, for the clusters with Std/Avg larger than 0.3, the mean cluster error rate is 9.57%.

When the n_cluster=2,3, there are 54.00% of clusters with Std/Avg larger than 0.3. On the contrary, when the n_cluster=4,5,6,7,8, there are 80.07% of clusters with Std/Avg larger than 0.3. This phenomenon may be explained by when the number of clusters gets larger, some clusters may only contain unstable servers which do not show steadily. Fig. 5.5 indicates the CJS model cannot converge well when the cluster has high Std/Avg. As a result, the CJS model with a larger number of clusters tends to have a worse error rate.

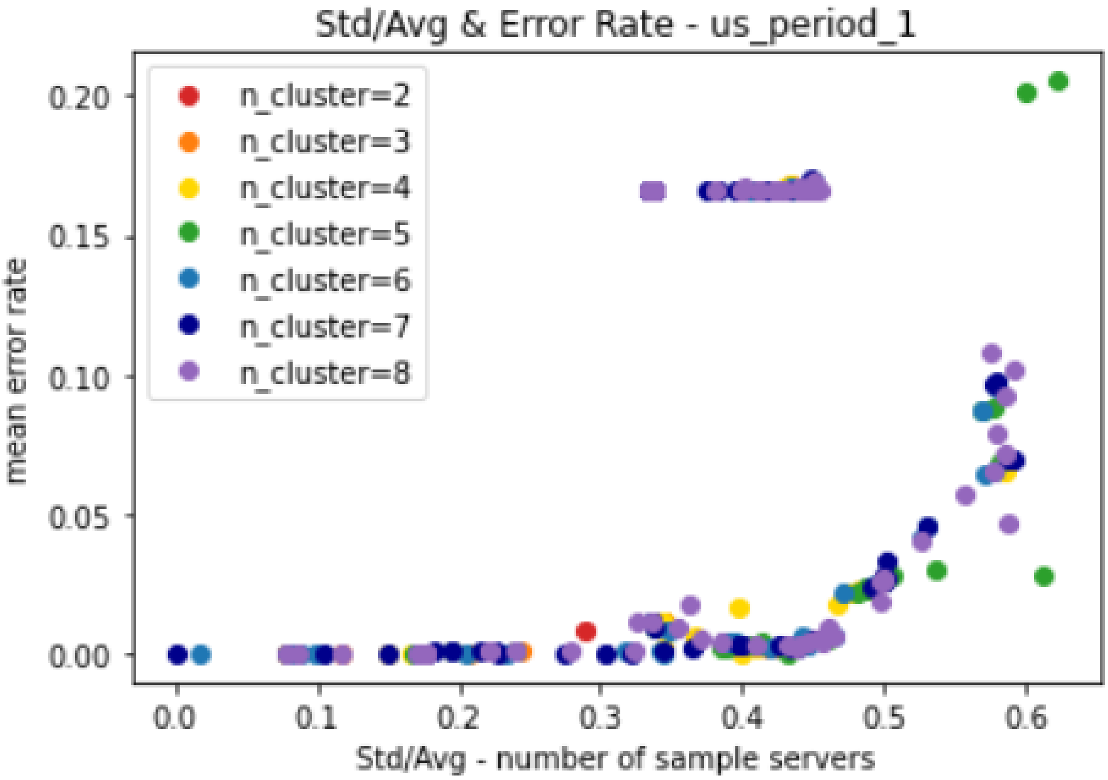


Figure 5.5: Std/Avg and Cluster Error Rate - US-1

I dug into the reason for the high cluster error rate, and then I found the estimation numbers of some clusters were 0 on May 13. However, the baseline number is as usual as the example shown in Table. 5.5. The reason why estimation numbers on May 13 are 0

doi:10.6342/NTU202400040

is that when the servers are divided into too many clusters, the sample numbers in some clusters are 0 on May 13 while the baseline is not 0, which would lead to high Std/Avg as well as a high cluster error rate.

	May 9	May 10	May 11	May 12	May 13	May 14
baseline number	67	67	67	67	67	67
estimation number	67.0	67.0012	67.0	67.0	0.0	67.0

Table 5.5: An Example of Estimation Number = 0 - US-1

To find out how the estimation number wrongly equals 0 affect error rates, I dig into how many clusters in each clustering result have an estimation number wrongly equaling 0 on one date (est_0_clusters), and how many IPs are in such clusters of each clustering result (est_0_ip). The correlation of error rate and est_0_clusters is 0.7164, and the correlation of error rate and est_0_ip is 0.9960. It indicates that a high est_0_ip would lead to a high error rate of the clustering result.

Next, I dig into the (1) Std/Avg of the clusters estimation numbers are wrongly equal to 0 on one date and (2) which date estimation numbers are wrongly equal to 0. The mean Std/Avg of such clusters is 0.3896, and the standard deviation of Std/Avg is 0.0476. The date estimation numbers are wrongly equal to 0 concentrates on May 13, only one cluster has the estimation number wrongly equal to 0 on May 10 (May 13: 110, May 10: 1).

To be notified, est_0_clusters and est_0_ip need to compare the estimation number with the baseline. est_0_clusters and est_0_ip can explain why high Std/Avg leads to a high error rate, however, these metrics could not be computed without a baseline.

I think a cluster with a small size may lead to a high probability of the estimation number of a cluster being wrongly equal to 0 as well as a high cluster error rate. Thus, I dig into the relationship of cluster size and cluster error rate, which is shown in Fig. 5.6.

doi:10.6342/NTU202400040

Almost all the clusters with cluster error rates larger than 0.15 have cluster sizes less than 100. The correlation between the cluster size and cluster error rate is -0.2070, which is closer to 0 than the correlation between the Std/Avg and cluster error rate, 0.3392.

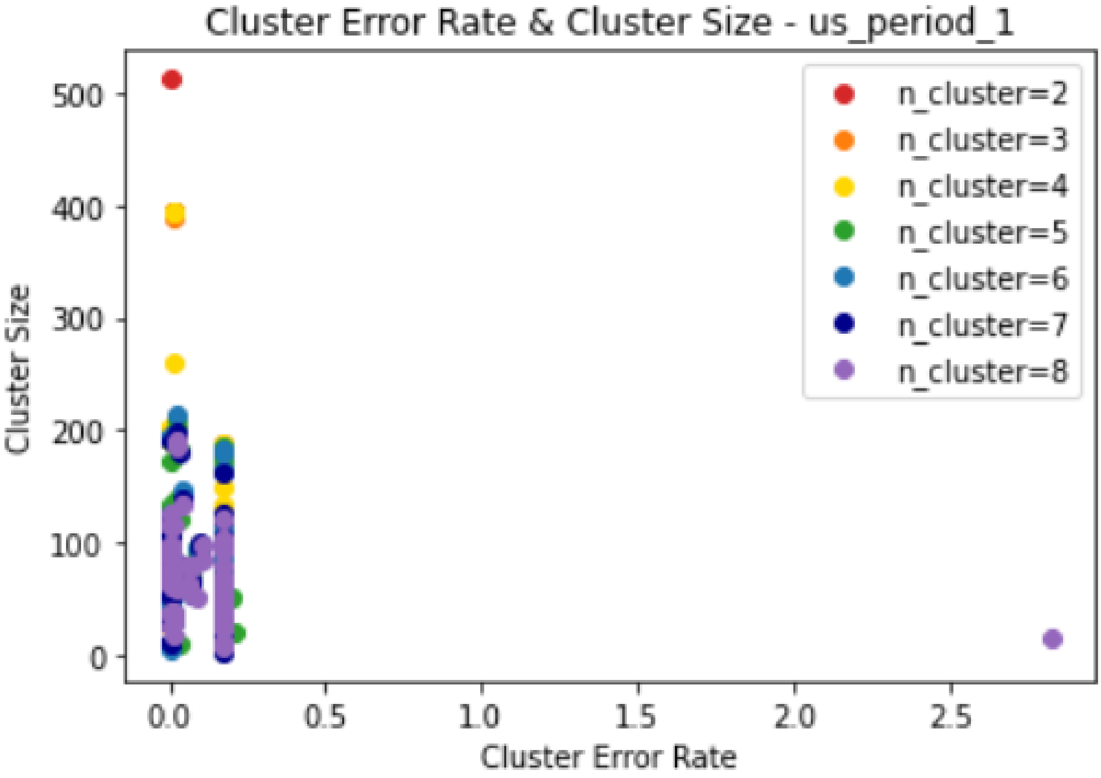


Figure 5.6: Cluster Size and Cluster Error Rate - US-1

In the above content, I show that for each cluster, a higher Std/Avg and a lower cluster size would lead to a higher cluster error rate. Next, I use ’min cluster size’ and ’mean Std/Avg’ to inspect the error rate of each clustering result. In Fig. 5.7, the clustering results with higher error rate (dark color) are concentrated in the top-left corner, which indicates lower min cluster size and higher mean Std/Avg would tend to generate a higher error rate.

doi:10.6342/NTU202400040

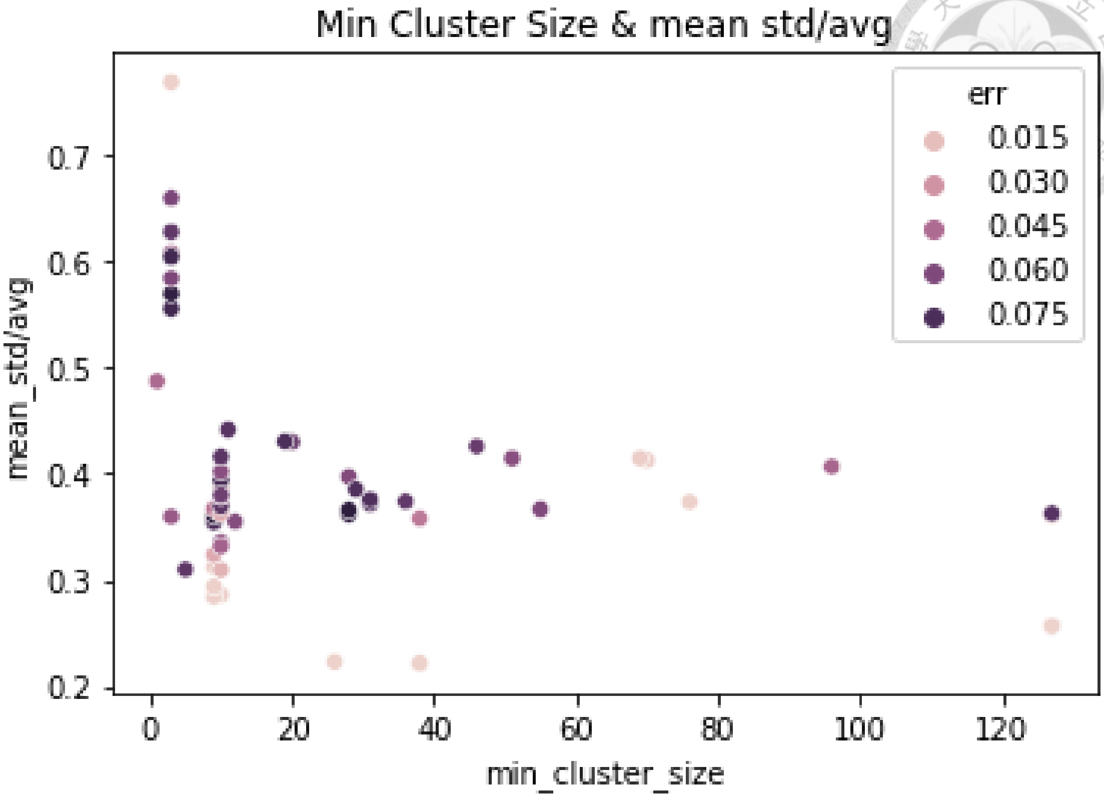


Figure 5.7: Min Cluster Size, Mean Std/Avg, and Error Rate - US-1

The correlation matrix of n_cluster, error rate, mean std/avg, and min cluster size is shown in Table. 5.6. The correlation between error rate and mean std/avg is 0.6956, and the correlation between error rate and min cluster size is -0.5246. The correlation justifies again that lower min cluster size and higher mean Std/Avg would tend to have a higher error rate.

	n_cluster	error rate	mean std/avg	min cluster size
n_cluster	1.0000	0.7193	0.6101	-0.6740
error rate	0.7193	1.0000	0.4619	-0.5246
mean std/avg	0.6101	0.4619	1.0000	-0.4393
min cluster size	-0.6740	-0.5246	-0.4393	1.0000

Table 5.6: Correlation Matrix of Min Cluster Size and Mean Std/Avg - US-1

doi:10.6342/NTU202400040

5.2.2 Clustering Result with US-0 - April 29 to May 05

5.2.2.1 S_Dbw Score and Estimation Error Rate

In US-0, the result of the CJS model without clustering is shown in Fig. 5.8 when the sample hour is 12 am. The error rate is about 5.92%, which is worse than the error rate, of 0.82%, in US-1.

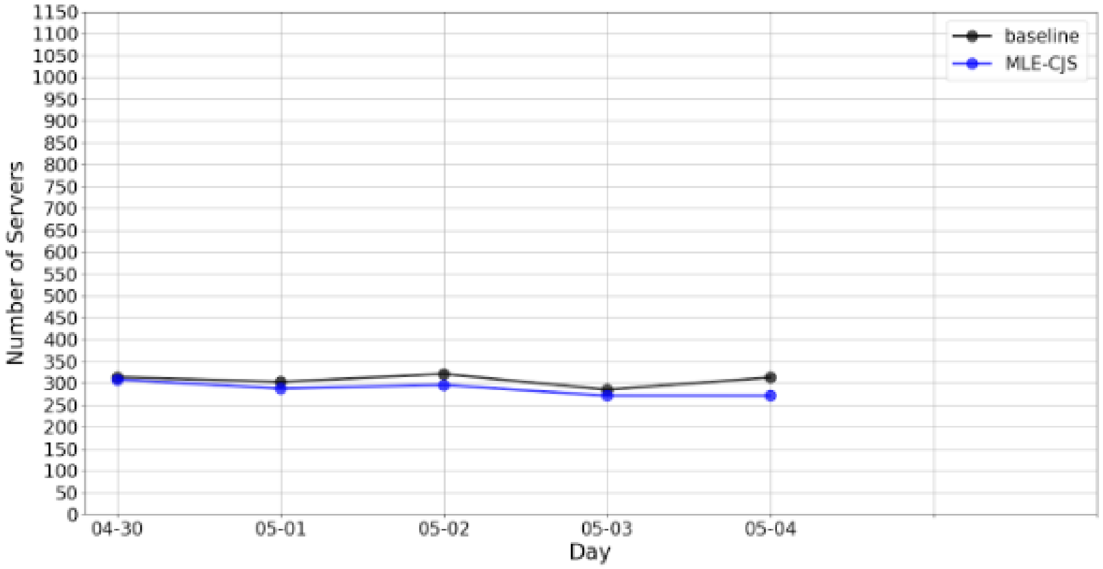


Figure 5.8: The CJS model without Clustering - US-0

Same to the clustering result with US-1, I do k-means 10 times for each number of clusters equal to 2 to 8 with feature ”n_period=3, slide_hour=0”. The S_Dbw scores and the error rates of the CJS model results are shown in Fig. 5.9. In US-0, several CJS results with clustering have a better error rate than the result without clustering, which does not happen in US-1.

doi:10.6342/NTU202400040

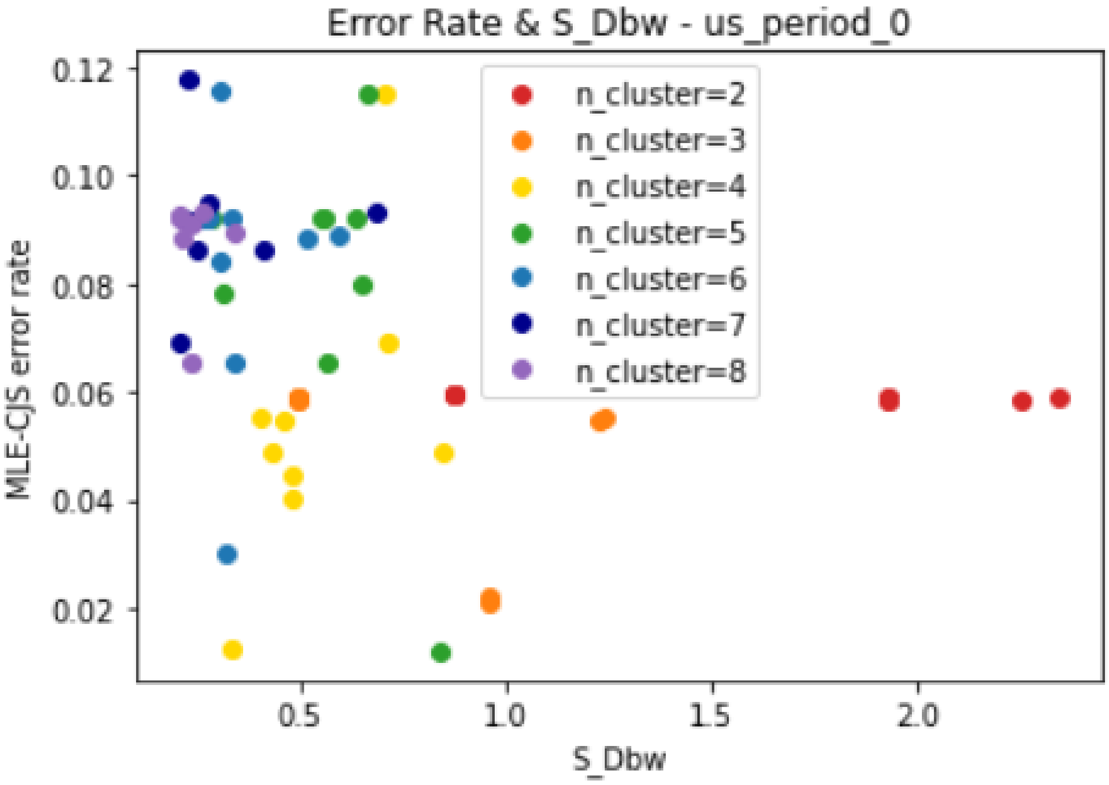


Figure 5.9: S_Dbw Score and Error Rate - US-0

The correlation matrix of the number of clusters, S_Dbw score, error rate, and standard deviation of error rate (stdev) in US-0 is shown in Table. 5.7. Similar to the CJS results in US-1, the correlation between S_Dbw score and error rate is negative, and the correlation between the number of clusters and error rate is positive.

	n_cluster	S_Dbw	error rate	stdev
n_cluster	1.0000	-0.7163	0.5712	0.7551
S_Dbw	-0.7163	1.0000	-0.3466	-0.5208
error rate	0.5712	-0.3466	1.0000	0.8127
stdev	0.7551	-0.5208	0.8127	1.0000

Table 5.7: Correlation Matrix - US-0

Fig. 5.10 shows the correlation of S_Dbw and error rate when the number of clusters equals to 2 to 8. Unlike Fig. 5.4 in US-1, in US-0, the correlation of S_Dbw and error rate doesn't have an obvious increase as the number of clusters gets larger. When the number

doi:10.6342/NTU202400040

of clusters = 6, 7, and 8, the correlations of S_Dbw and error rate are close to 0 (0.0467, 0.0266, 0.0445).

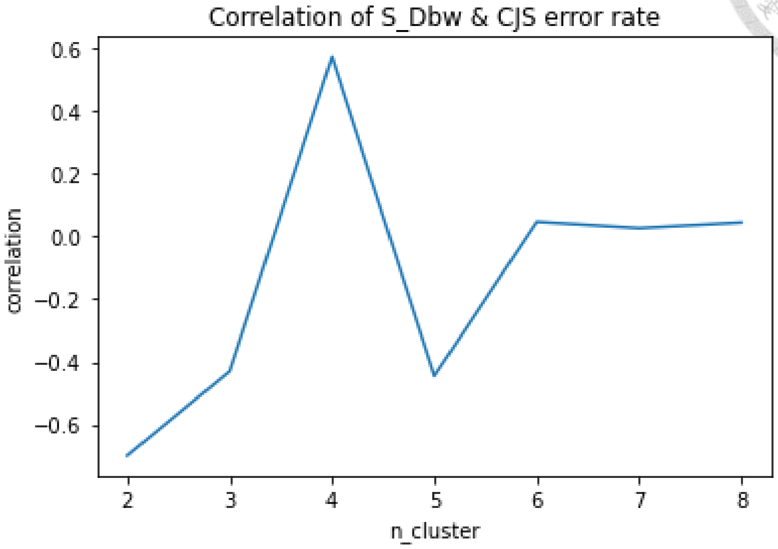


Figure 5.10: The Correlation of S_Dbw and Error Rate - US-0

5.2.2.2 Alternative Clustering Metrics

Similar to the analysis of US-1, I use the alternative metrics, Std/Avg, mean Std/Avg, cluster size, and min cluster size, to inspect the CJS model results.

Fig. 5.11 shows the relationship between the Std/Avg and the mean cluster error rate of each cluster. The correlation of Std/Avg and the cluster error rate is 0.9347. For the clusters with Std/Avg less than 0.3, the mean cluster error rate is 1.90%. However, for the clusters with Std/Avg larger than 0.3, the mean cluster error rate increases to 58.14%.

When the n_cluster=2,3, there are 4.08% of clusters with Std/Avg larger than 0.3. On the contrary, when the n_cluster=4,5,6,7,8, there are 23.45% of clusters with Std/Avg larger than 0.3. As a result, the CJS model with a larger number of clusters tends to have a worse error rate.

Similar to US-1, the estimation number of some clusters in US-0 is wrongly equal

doi:10.6342/NTU202400040

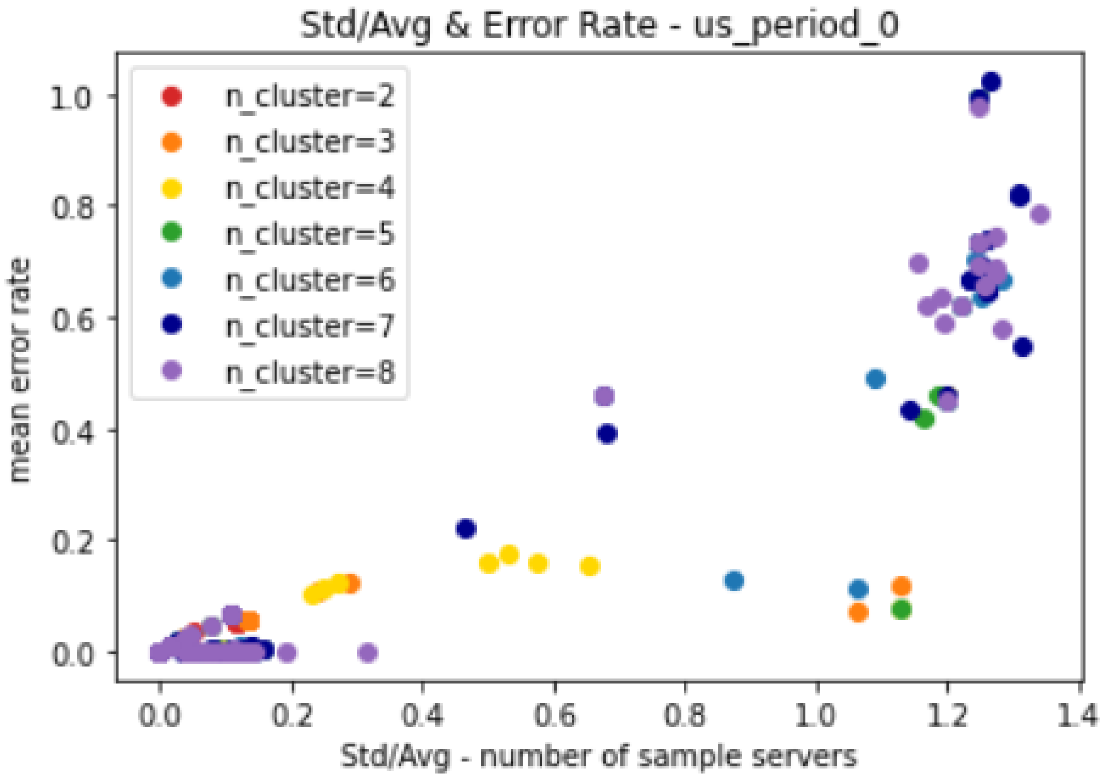


Figure 5.11: Std/Avg and Cluster Error Rate - US-0

to 0 on a date. Again, I dig into how many clusters in each clustering result have an estimation number wrongly equaling 0 on one date (est_0_clusters), and how many IPs are in such clusters of each clustering result (est_0_ip). The correlation of error rate and est_0_clusters is 0.7195, and the correlation of error rate and est_0_ip is 0.8151. Once again, it indicates that a high est_0_ip would lead to a high error rate of the clustering result.

Also, I dig into the (1) Std/Avg of the clusters estimation numbers are wrongly equal to 0 on one date and (2) which date estimation numbers are wrongly equal to 0, which is shown below. The mean Std/Avg of such clusters is 1.2204, and the standard deviation of Std/Avg is 0.1450. The date estimation numbers are wrongly equal to 0 concentrates on May 1, only one cluster has the estimation number wrongly equal to 0 on May 2 (May 01: 59, May 02: 1).

doi:10.6342/NTU202400040

The scatter plot of cluster size and the cluster error rate is shown in Fig. 5.12. All the clusters with cluster error rates > 0.2 have cluster sizes < 100. The correlation between the cluster size and cluster error rate is -0.1431, which is closer to 0 than the correlation between the Std/Avg and cluster error rate, 0.93

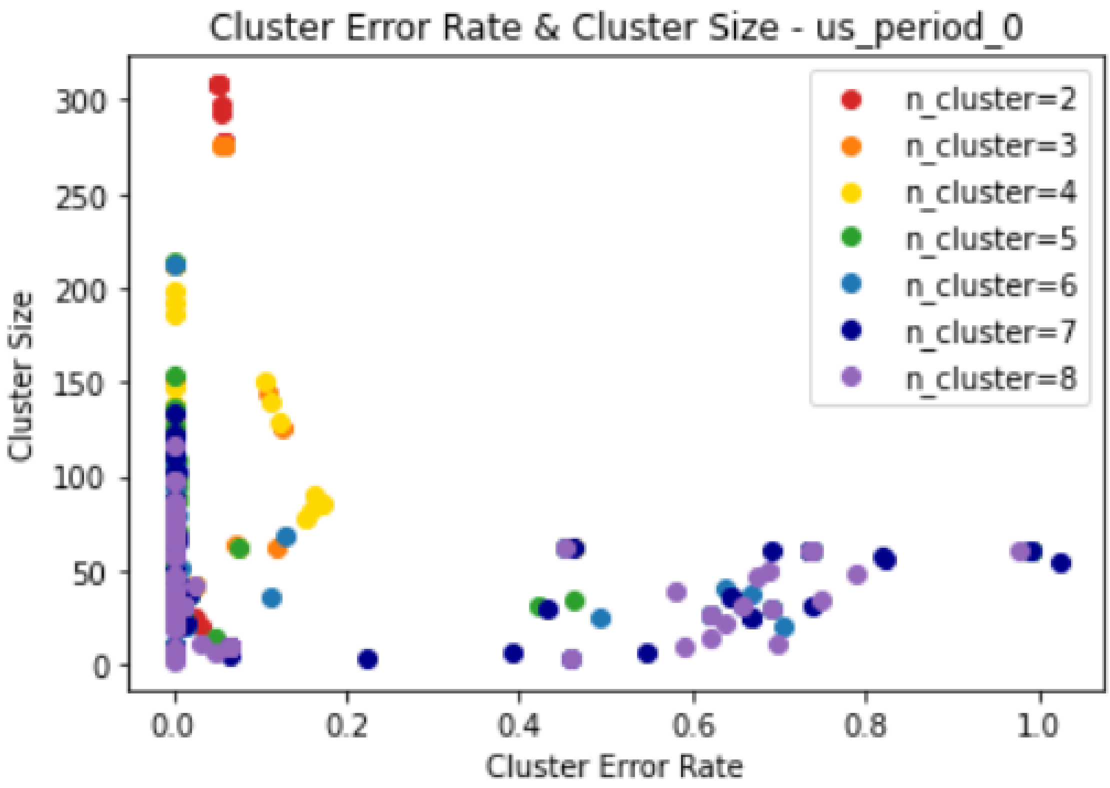


Figure 5.12: Cluster Size and Cluster Error Rate - US-0

For each clustering result, I inspect the min cluster size, Std/Avg, and error rate as shown in Fig. 5.13. In the top-left corner of Fig. 5.13, there are a lot of clustering results with an error rate larger than 0.08. Once again, it indicates that higher Std/Avg and lower min cluster size would tend to have a higher error rate.

doi:10.6342/NTU202400040

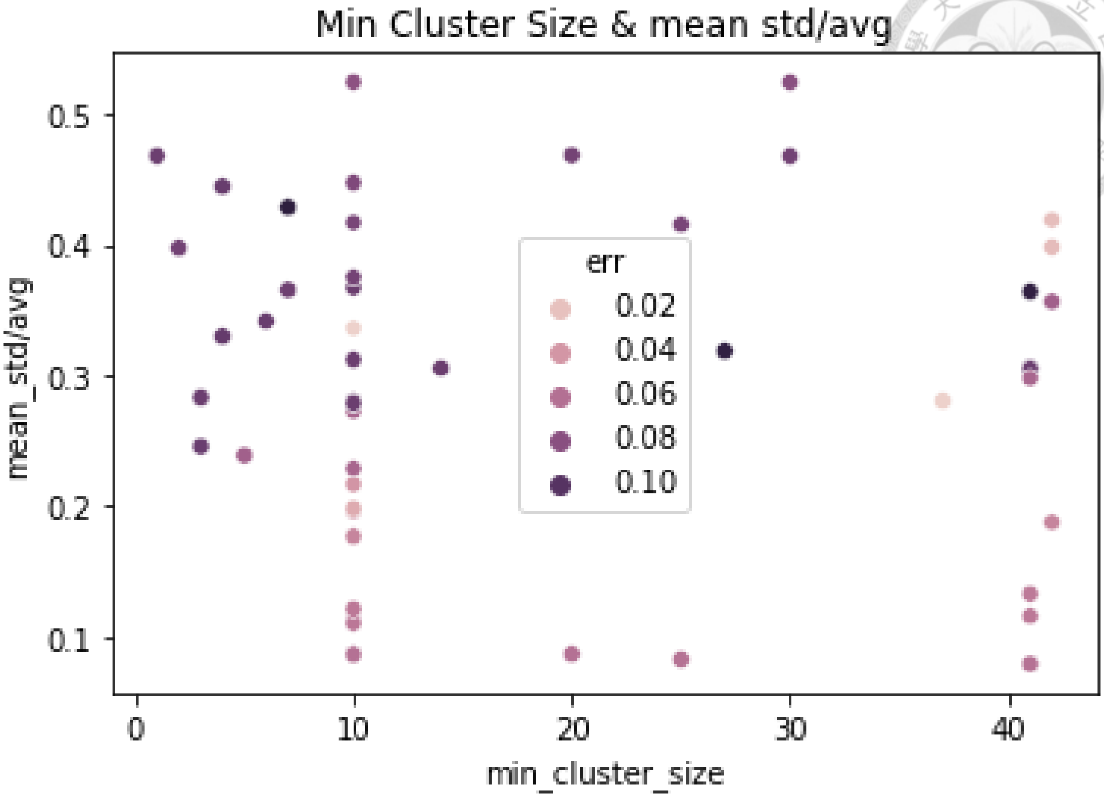


Figure 5.13: Min Cluster Size, Mean Std/Avg, and Error Rate - US-0

The correlation matrix of n_cluster, error rate, mean Std/Avg, and min cluster size is shown in Table. 5.8. The correlation between error rate and mean Std/Avg is 0.5472, and the correlation between error rate and min cluster size is -0.2532. The correlation justifies again that lower min cluster size and higher mean Std/Avg would tend to have a higher error rate.

	n_cluster	error rate	mean Std/Avg	min cluster size
n_cluster	1.0000	0.5712	0.6724	-0.4793
error rate	0.5712	1.0000	0.4430	-0.2532
mean Std/Avg	0.6724	0.4430	1.0000	-0.2060
min cluster size	-0.4793	-0.2532	-0.2060	1.0000

Table 5.8: Correlation Matrix of Min Cluster Size and Mean Std/Avg - US-0

doi:10.6342/NTU202400040

5.3 CJS with Random Clustering Results

In this section, I deploy the CJS model with random clustering results to compare with the k-means results in US-0 and US-1. For each n_cluster = 2 to 8, I do random clustering 20 times to divide servers into clusters of approximately the same size.

5.3.1 Random Clustering Results in US-0

In Fig. 5.14, it shows the error rate and S_Dbw score of k-means results (green dots) and random clustering results (blue dots). The error rate from random clustering results is about 6%, which is close to the error rate of the CJS model without clustering, 5.92%.

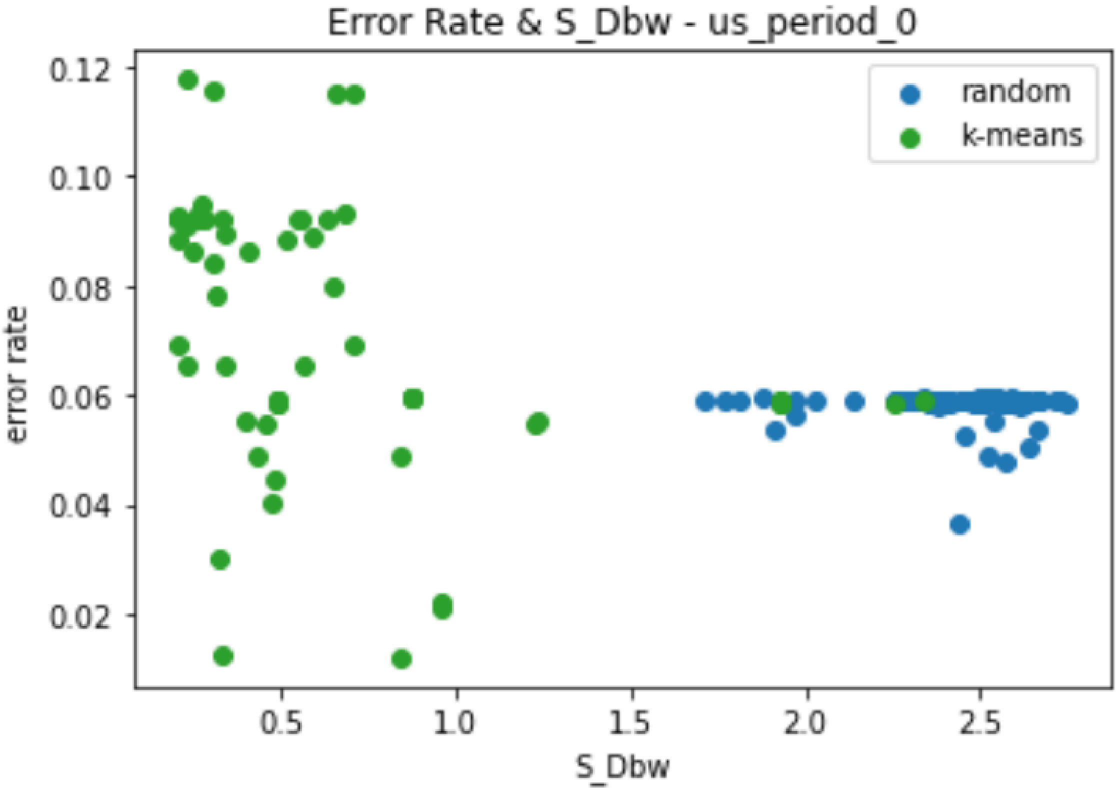


Figure 5.14: K-Means Results and Random Clustering Results - US-0

The error rate of the CJS model and the S_Dbw scores of the random clustering results with n_cluster = 2 to 8 are shown in Fig. 5.15. Most of the random clustering results have

doi:10.6342/NTU202400040

error rates between 0.055 to 0.06, while the S_Dbw scores range from 1.6 to 2.8.

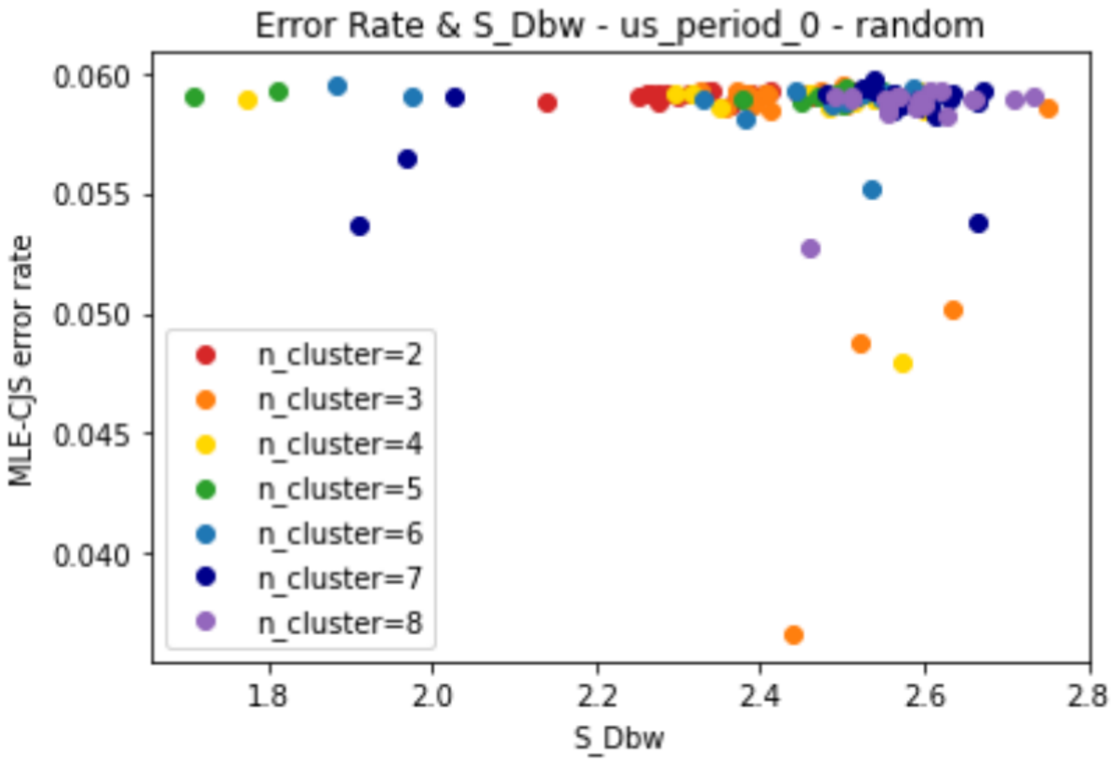


Figure 5.15: Error Rate and S_Dbw of Random Clustering Results - US-0

The correlation matrix of the random clustering results in US-0 is shown in Table. 5.9. The correlation between S_Dbw and the error rate is close to 0, which means the S_Dbw score has little matter with the error rate in the random clustering results. Besides, the correlation between n_cluster and the error rate is also close to 0, which shows the phenomenon that the error rate gets worse as n_cluster gets larger does not exist here.

	n_cluster	S_Dbw	error rate	stdev
n_cluster	1.0000	0.3275	0.0417	0.0015
S_Dbw	0.3275	1.0000	-0.0168	-0.0338
error rate	0.0417	-0.0168	1.0000	-0.7964
stdev	0.0015	-0.0338	-0.7964	1.0000

Table 5.9: Correlation Matrix - Random Clustering in US-0

For each cluster in every clustering result in US-0, the cluster error rate and the Std/Avg is shown in Fig. 5.16. All the clusters have Std/Avg less than 0.2 in random clustering

doi:10.6342/NTU202400040

results, while the maximum Std/Avg is larger than 1.2 in US-0 k-means results. The correlation of Std/Avg and cluster error rate is 0.2840, while the correlation of Std/Avg and cluster error rate in k-means results is 0.9347. The correlation of Std/Avg and cluster error rate in random clustering is closer to 0 than the correlation in k-means results.

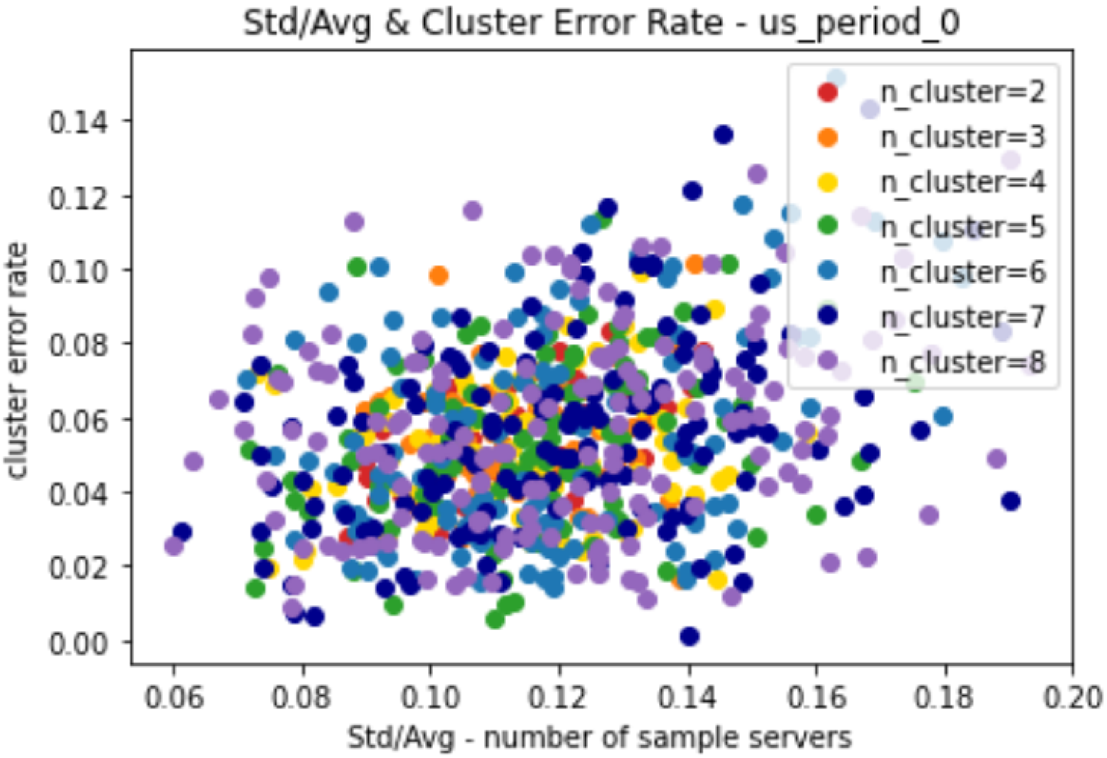


Figure 5.16: Cluster Error Rate and Std/Avg of Random Clustering Results - US-0

The reason why random clustering results get a better error rate, in general, is because the estimation numbers are not wrongly equal to 0 in random clustering results. In the random clustering results, no cluster wrongly estimates the number of servers to 0 on all dates. However, for the k-means results, this situation happens in many clusters. I think it is because K-means divides the servers too ”well”. Because k-means uses the transaction counts in 3 3-hour periods as the attributes, it has a higher probability of dividing the servers that do not show in the sample hour but show on baseline into the same cluster. As a result, k-means generates many clusters with a sample number equal to 0 but a baseline number not equal to 0, which causes the CJS model to wrongly estimate the number to 0

doi:10.6342/NTU202400040

on one date.

5.3.2 Random Clustering Results in US-1

In Fig. 5.17, it shows the error rate and S_Dbw score of k-means results and random clustering results. The error rate of random clustering results is better than the error rate of k-means results. This could be also explained by the estimation numbers not wrongly equal to 0 in random clustering results, while it happens usually in the k-means results of US-1.

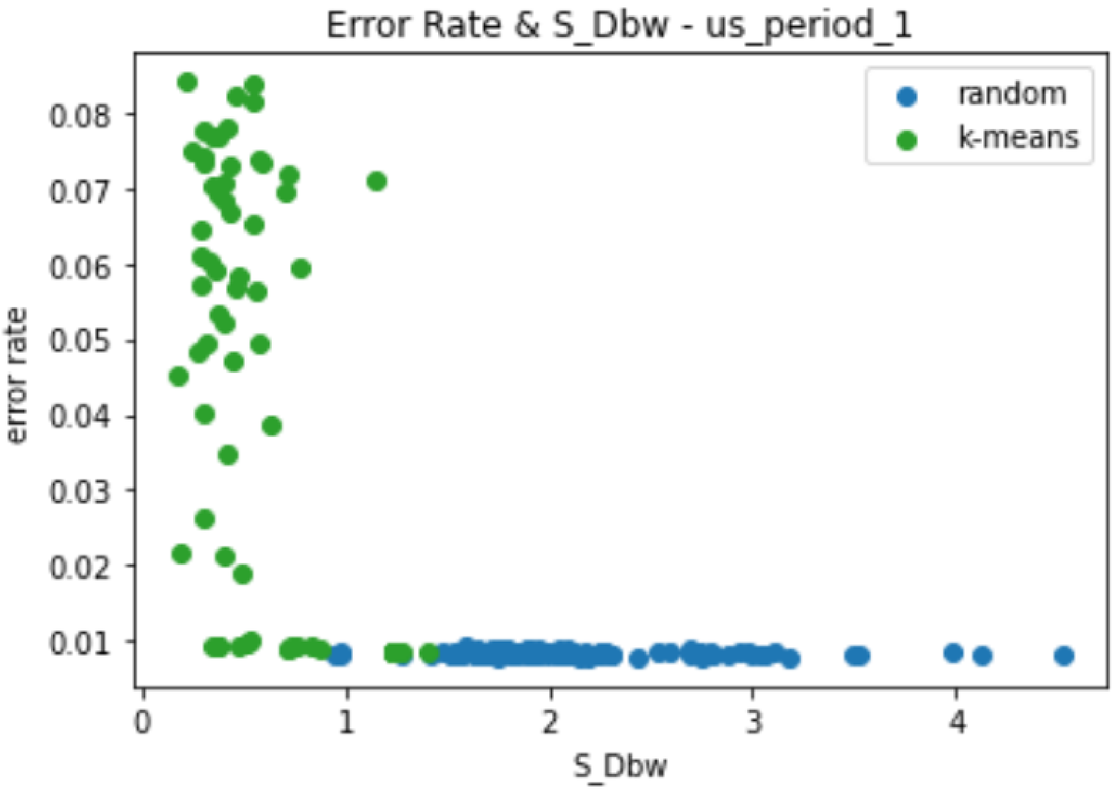


Figure 5.17: K-Means Results and Random Clustering Results - US-1

The error rate of the CJS model and the S_Dbw scores of the random clustering results with n_cluster = 2 to 8 are shown in Fig. 5.18. The error rates are concentrated from 0.0076 to 0.0094, while the S_Dbw scores range from about 1.0 to 4.5.

The correlation matrix of the random clustering results in US-1 is shown in Table.

doi:10.6342/NTU202400040

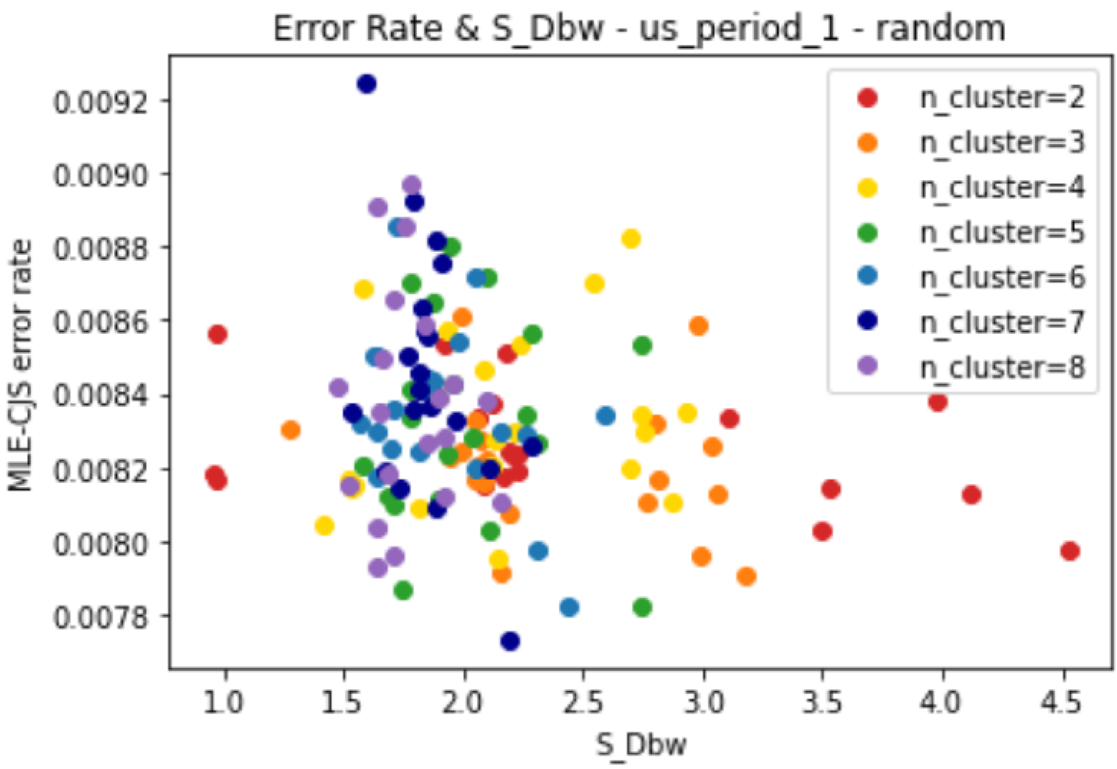


Figure 5.18: Error Rate and S_Dbw of Random Clustering Results - US-1

5.10. The correlation of n_cluster and the error rate is 0.2323, while this correlation in k-means results is 0.5712, which means n_cluster has a smaller impact on the error rate in the random clustering results. The correlation of S_Dbw and the error rate is -0.2212, this correlation is also closer to 0 in the random clustering results than in the k-means results (-0.3466).

	n_cluster	S_Dbw	error rate	stdev
n_cluster	1.0000	-0.4273	0.2323	0.2106
S_Dbw	-0.4273	1.0000	-0.2212	0.0036
error rate	0.2323	-0.2212	1.0000	-0.4869
stdev	0.2106	0.0036	-0.4869	1.0000

Table 5.10: Correlation Matrix - Random Clustering in US-1

For each cluster in every clustering result in US-1, the cluster error rate and the Std/Avg are shown in Fig. 5.19. All the clusters have Std/Avg less than 0.34 in random clustering results, while the maximum Std/Avg is larger than 2.5 in k-means results. The

doi:10.6342/NTU202400040

correlation of Std/Avg and cluster error rate is -0.0135, while the correlation of Std/Avg and cluster error rate in k-means results is 0.3392. Similar to US-0, the correlation of Std/Avg and cluster error rate in random clustering is closer to 0 than the correlation in k-means results.

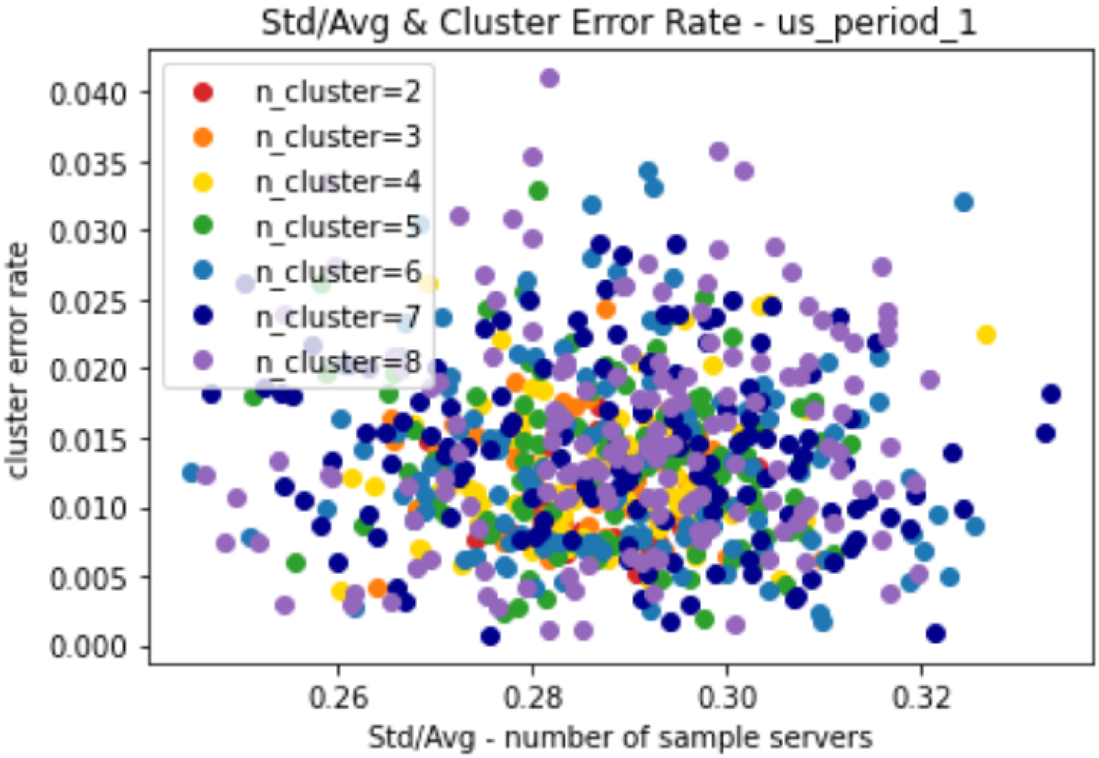


Figure 5.19: Cluster Error Rate and Std/Avg of Random Clustering Results - US-1

doi:10.6342/NTU202400040

Chapter 6 Sensitivity Analysis

In this chapter, I compare the CJS model results in the US-0 and US-1. Besides, I deploy the CJS model in 6 periods from 4 different regions to see whether the CJS model with and without heterogeneity could fit into other data.

6.1 CJS Model in US-0 V.S. US-1

In the United States data, there are 2 periods with working hours equal to 24 for more than 7 continuous days, US-0 and US-1. In the last chapter, I have shown and analyzed the US data. In this chapter, I will compare the CJS model results from the 2 data.

Table. 6.1 shows the total number of servers, the error rate of the CJS model without clustering, and Std/Avg without clustering. In this table, I calculate Std/Avg by sample number and baseline number without clustering on each date in the US-0 and US-1. The Std/Avg of the sample number in the 2 periods are all below 0.3, and the error rates of the CJS model without clustering are all below 6%.

doi:10.6342/NTU202400040

	US-0	US-1
total number of servers	322	538
error rate (no clustering)	5.92%	0.82%
Std/Avg (sample number)	0.1109	0.2858
Std/Avg (baseline number)	0.0377	0.1123

Table 6.1: Overview of US-0 and US-1

6.1.1 Estimation Error Rate in the US

In Table. 6.2, I compare the correlation in multiple k-means results and random clustering results. For the k-means results in both 2 data, a larger number of clusters tends to have a worse error rate. As for the S_Dbw score, if a result with a better S_Dbw score (lower value) would lead to a better error rate (lower value) in the CJS model, the correlation of S_Dbw and error rate should be positive. However, in both k-means and random clustering results of the US-0 and US-1, the correlations of S_Dbw and error rate are all negative, which means S_Dbw is not a good metric for the CJS model in the US data.

data	correlation	US-0	US-1
k-means	n_cluster	0.5712	0.7193
	S_Dbw	-0.3466	-0.5717
random clustering	n_cluster	0.0417	0.2323
	S_Dbw	-0.0168	-0.2212

Table 6.2: Correlation with Error Rate - the US

doi:10.6342/NTU202400040

6.1.2 Cluster Error Rate in the US

Also, I have dug into the error rate of each cluster - cluster error rate. I use 2 metrics, Std/Avg and cluster size, to evaluate their correlation with the cluster error rate, which is shown in Table. 6.3.

data	correlation	US-0	US-1
k-means	n_cluster	0.1494	0.1686
	Std/Avg	0.9347	0.3392
	cluster size	-0.1431	-0.2070
random clustering	n_cluster	0.0561	0.1903
	Std/Avg	0.2840	-0.0135
	cluster size	-0.0458	-0.1791

Table 6.3: Correlation with Cluster Error Rate - the US

From the above results, I find that Std/Avg could be a good metric for cluster error rate when Std/Avg is high (k-means). On the contrary, the correlations of cluster size and cluster error rate are closer to 0 than in k-means results. In the random clustering results, the Std/Avg of clusters are all less than 0.20 in US-0 and 0.34 in US-1, and their correlation of Std/Avg and cluster error rate are low, 0.2840 in US-0 and -0.0135 in US-1. On the contrary, in the k-means results, the maximum Std/Avg exceed 1.1 in US-0 and 0.6 in US-1, and their correlation of Std/Avg and cluster error rate are relatively high, 0.9347 in US-0 and 0.3392 in US-1.

doi:10.6342/NTU202400040

6.2 CJS Model in the United Kingdom

In the United Kingdom data, there are 2 periods with working hours equal to 24 for more than 7 continuous days (UK-0: April 29 to May 05, UK-1: May 07 to May 15). I use the data in the two periods to generate the clustering results with k-means. For each n_cluster equals 2 to 8, I run k-means 10 times with the feature, n_period=3, slide_hour=0.

When the sample time is 12 am, the CJS model fails to converge. Hence, I dig into how many servers in each hour, shown in Fig. 6.1. At 12 am, there are only 97 and 98 IPs in the UK-0 and UK-1 of the United Kingdom data. There are many IPs that seldom or never show up at 12 am. Only the main servers were shown at 12 am, which leads to the CJS model cannot converge. Therefore, I change the sample time to the hour with most IPs in the Transaction List, 20 o'clock.

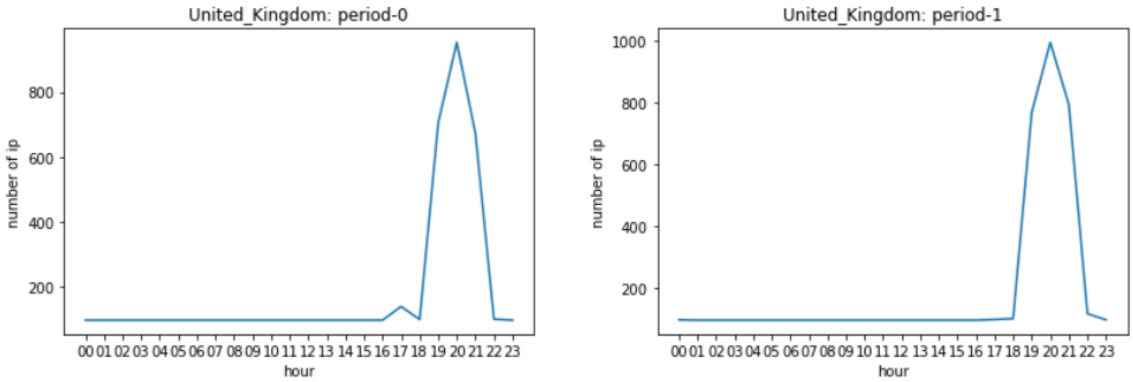


Figure 6.1: The Number of IPs in the UK of each hour

6.2.1 S_Dbw Score and Estimation Error Rate

In the UK-0, from April 29 to May 05, the result of the CJS model without clustering is shown in Fig. 6.2. The error rate is 114%, which is much more terrible than the result in the US-0 and US-1.

doi:10.6342/NTU202400040

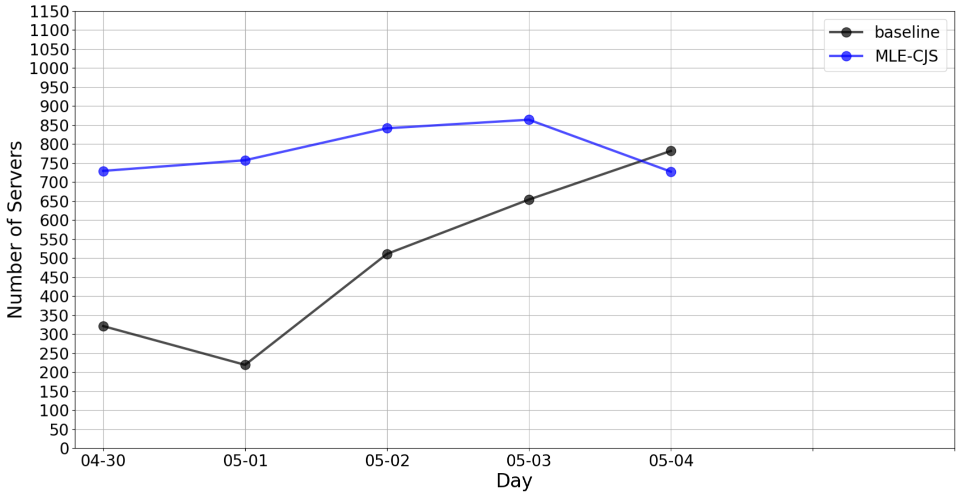


Figure 6.2: The Estimation Result of the CJS model without Clustering - the UK-0

As for the CJS model with clustering, the error rate is worse than the CJS model without clustering. In Fig. 6.3, all the estimation results have an error rate larger than 130%, which are all higher than the CJS without clustering. In UK-0, 20 out of 70 results with clustering even have an error rate > 10000%. I define the results with an error rate < 10000% as ”converge” and results with an error rate > 10000% as ”not converge”. I remove the ”not converge” results from Fig. 6.3.

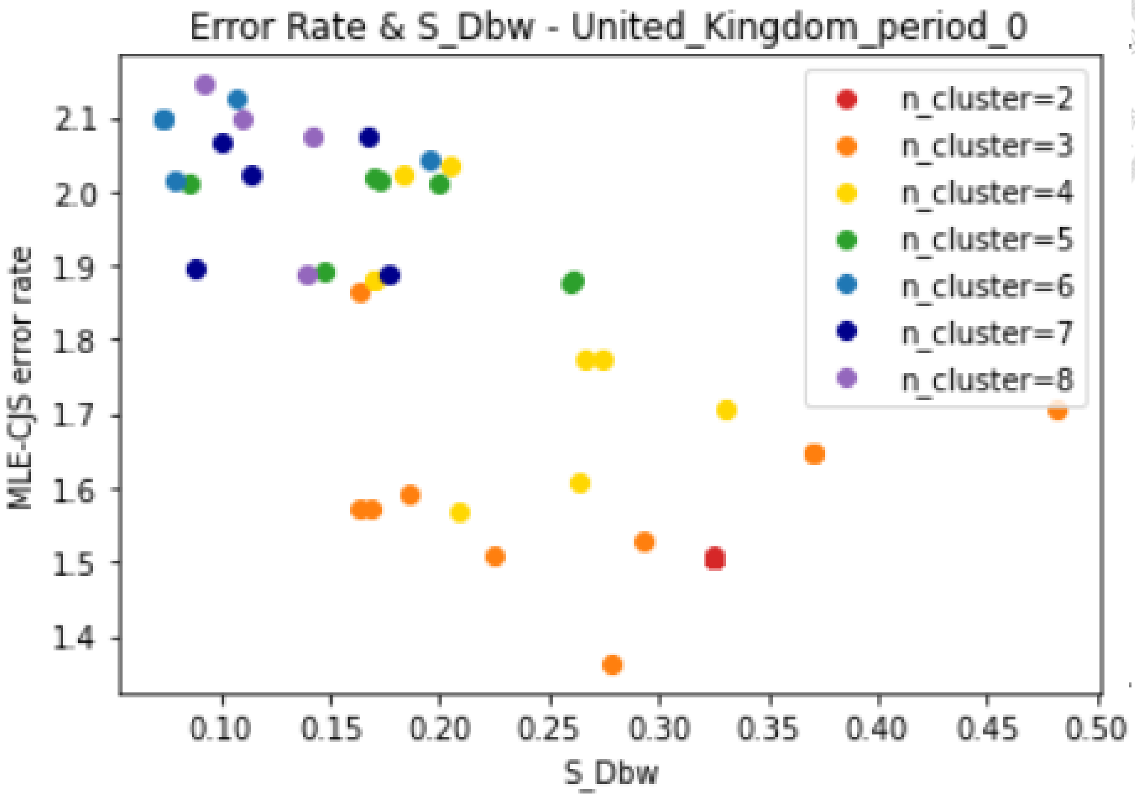


Figure 6.3: The CJS Result of the UK-0

In the UK-1, May 07 to May 15, the result of the CJS model without clustering is shown in Fig. 6.4. The error rate is 144%, which is much more terrible than the result in the US-0 and US-1.

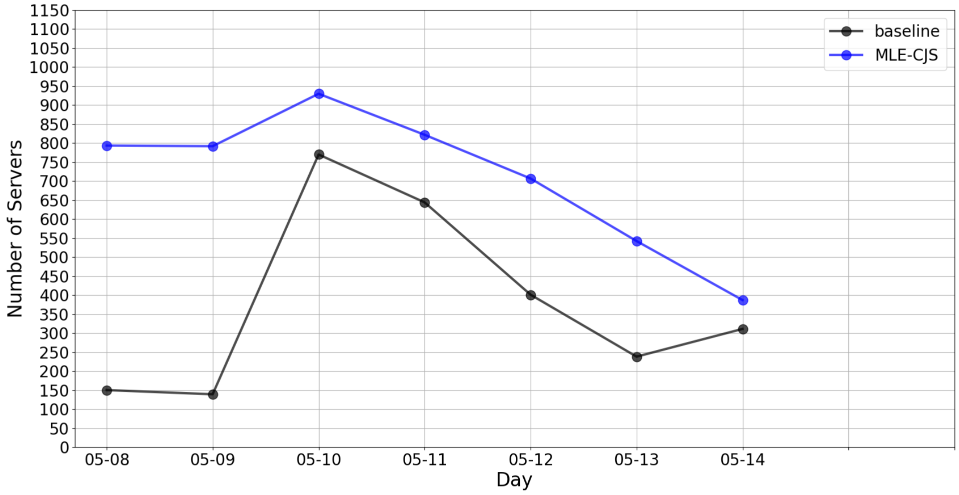


Figure 6.4: The Estimation Result of the CJS model without Clustering - the UK-1

doi:10.6342/NTU202400040

As for the CJS model with clustering, all the estimation results have an error rate of over 220%, which is higher than the CJS without clustering as shown in Fig. 6.5. In UK-1, 17 out of 70 results do not converge (error rate > 10000%). I remove the ”not converge” results in Fig. 6.5.

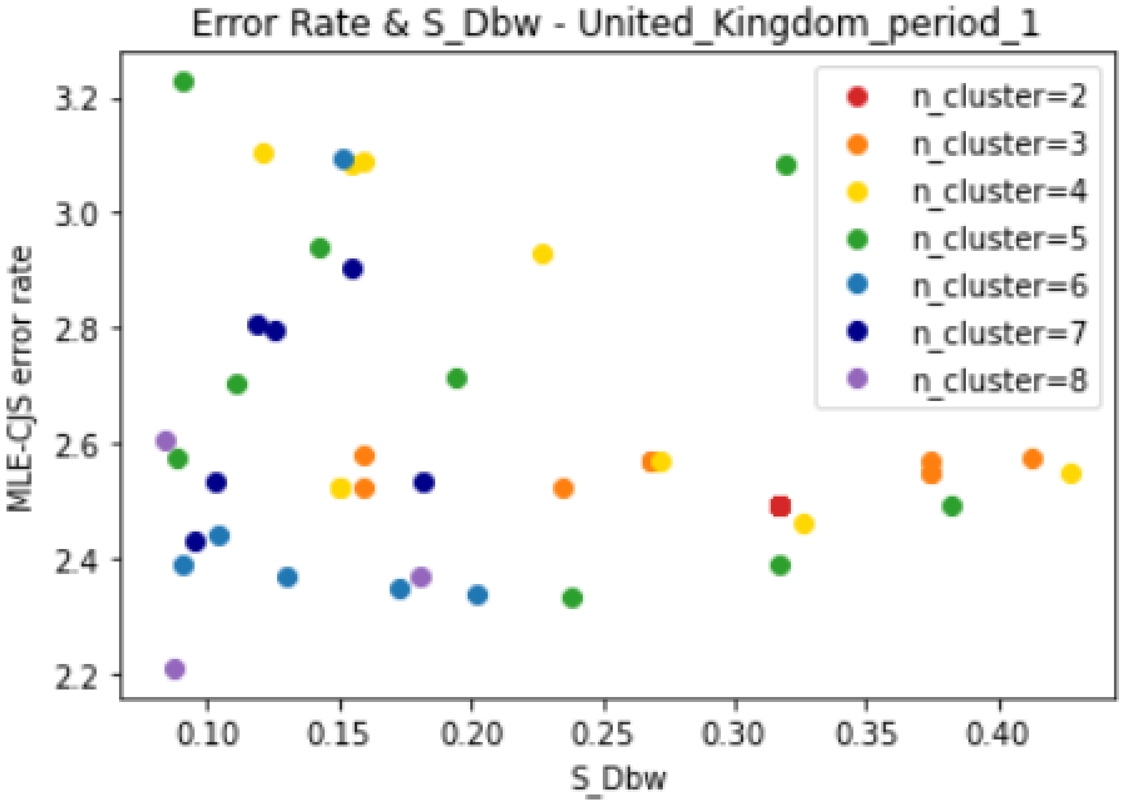


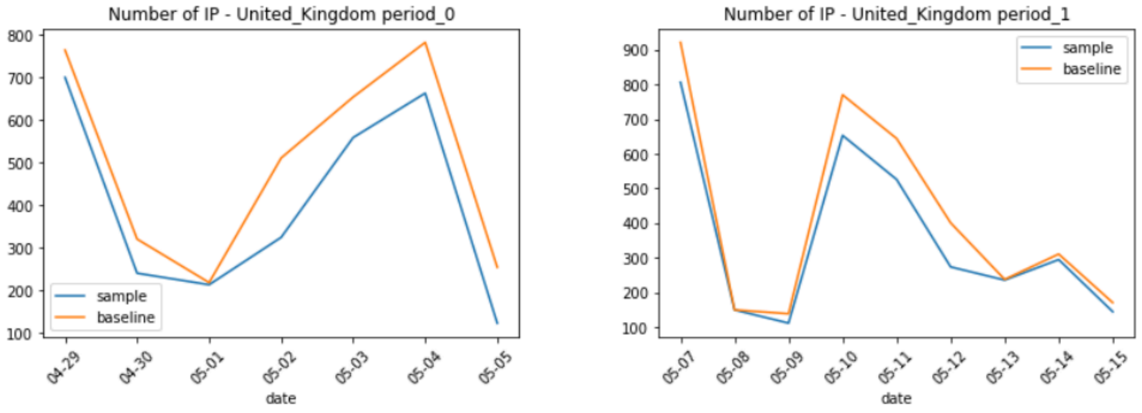
Figure 6.5: The CJS Result of the UK-1

6.2.2 Why CJS Model Cannot Fit Well in the UK

To find out the reason why the CJS model cannot fit well in the UK data, I dig into the number of servers in sample hour and the baseline, which is shown in Fig. 6.6. On May 05 in UK-0, May 08, and May 09 in UK-1, the number of IPs in the sample hour was only about 100, which is much less than the numbers on other dates. Besides, the standard deviations of the number of IPs in the sample hour in 2 periods are all above 200, and the Std/Avg of sample number without clustering are 0.5346 and 0.6575. On the contrary, the

doi:10.6342/NTU202400040

Std/Avg of sample numbers in the US are 0.1109 and 0.2858, which are much lower than the numbers in the UK. Based on the above two reasons, I think it is why the CJS model cannot fit well in the UK data.



std(sample_num): 215.8145462366866

std(sample_num): 233.55740563808945

Figure 6.6: The Number of IPs in Each Date - the UK

When n_cluster = 8, 6 out of 10 results in the UK-0 and 7 out of 10 results in the UK-1 do not converge. I find that it is caused by the capture probabilities of some clusters being extremely low. Take one cluster from a clustering result with n_cluster=8 in UK-0 as an example, the capture probability equal to $3.23 * 10^{-8}$ on May 02 and $3.33 * 10^{-6}$ on May 05, as shown in Table. 6.4. In this cluster, there are 94 out of 110 servers only shown in the sample hour on one date. Among these servers which only show on one date, the dates are distributed on 6 out of 7 dates (April 29 to May 05), which makes them hard to converge in the CJS model. As a result, the estimation number in this cluster will be extraordinarily large since the estimation number is calculated by the sample number divided by capture probability, which is extremely low on specific dates.

date	April 30	May 01	May 02	May 03	May 04	May 05
capture probability	0.0357	0.0536	$3.23 * 10^{-8}$	0.0282	0.1023	$3.33 * 10^{-6}$
sample number	2	6	12	19	29	2

Table 6.4: Cluster-2 in CJS result of n_cluster=8, label=2

doi:10.6342/NTU202400040

Also, I dig into Std/Avg and cluster error rate in ”converge” k-means results (error rate < 10000%), which are shown in Fig. 6.7 and Fig. 6.8. The maximum Std/Avg of the clusters is over 0.8 (UK-0) and 1 (UK-1). The correlations of Std/Avg and cluster error rate are 0.6824 (UK-0) and 0.8052 (UK-1).

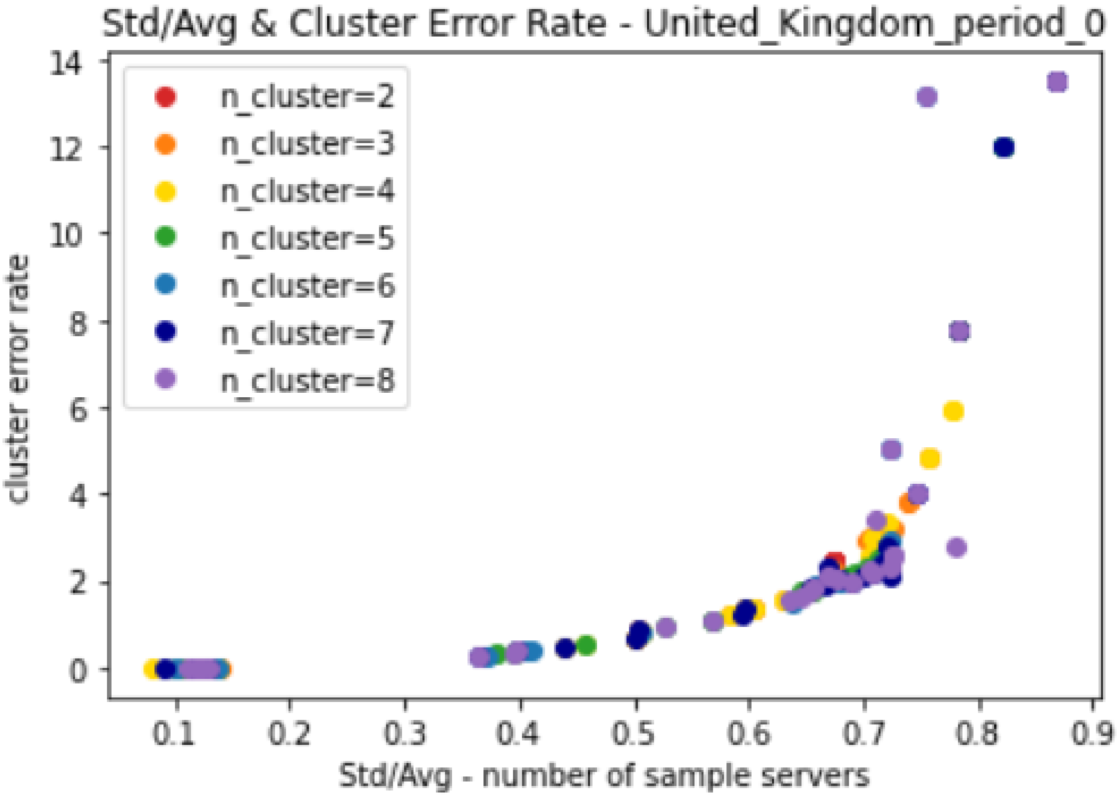


Figure 6.7: Std/Avg and Cluster Error Rate in K-Means - the UK-0

doi:10.6342/NTU202400040

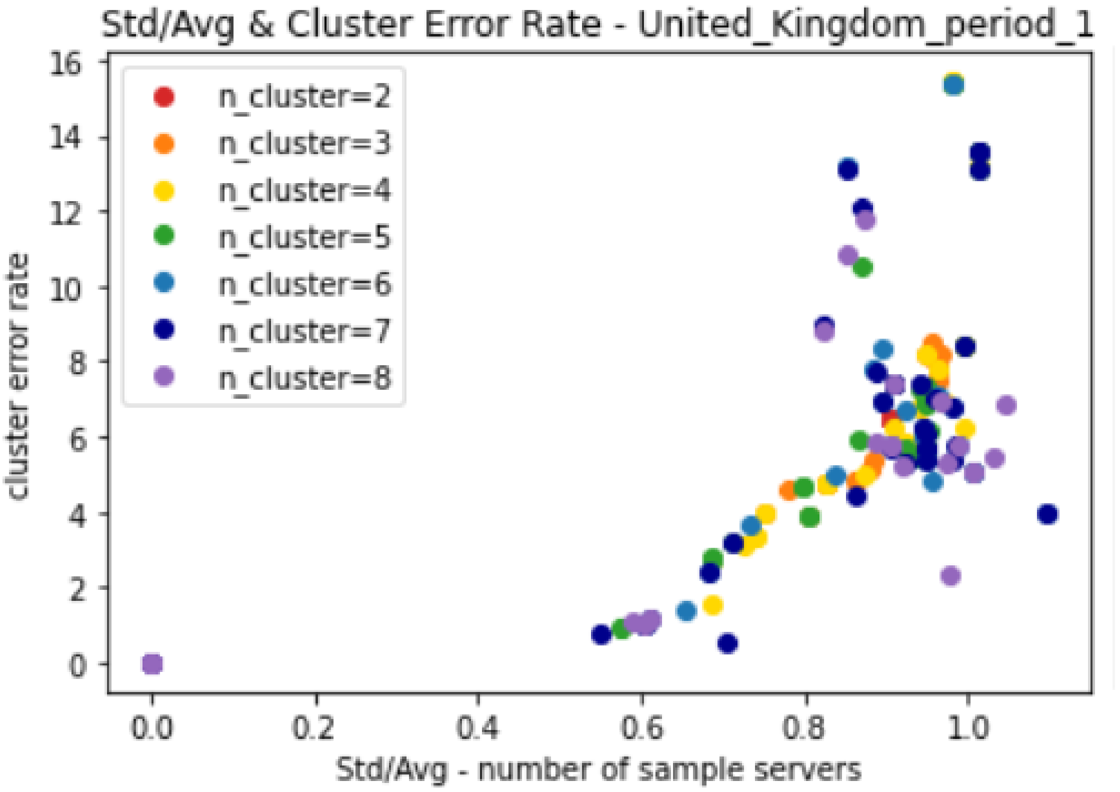


Figure 6.8: Std/Avg and Cluster Error Rate in K-Means - the UK-1

6.2.3 CJS Model with Random Clustering

Similar to what I have done to the US data, I do random clustering 20 times for each n_cluster = 2 to 8 in the UK-0 as well as in the UK-1. The CJS model with k-means and random clustering in the UK-0 and UK-1 are shown in Fig. 6.9 and Fig. 6.10. CJS models with random clustering results obviously have lower error rates than k-means in both the UK-0 and UK-1. Besides, all the CJS models with random clustering converge (error rate < 10000%) in the UK-0 and UK-1.

doi:10.6342/NTU202400040

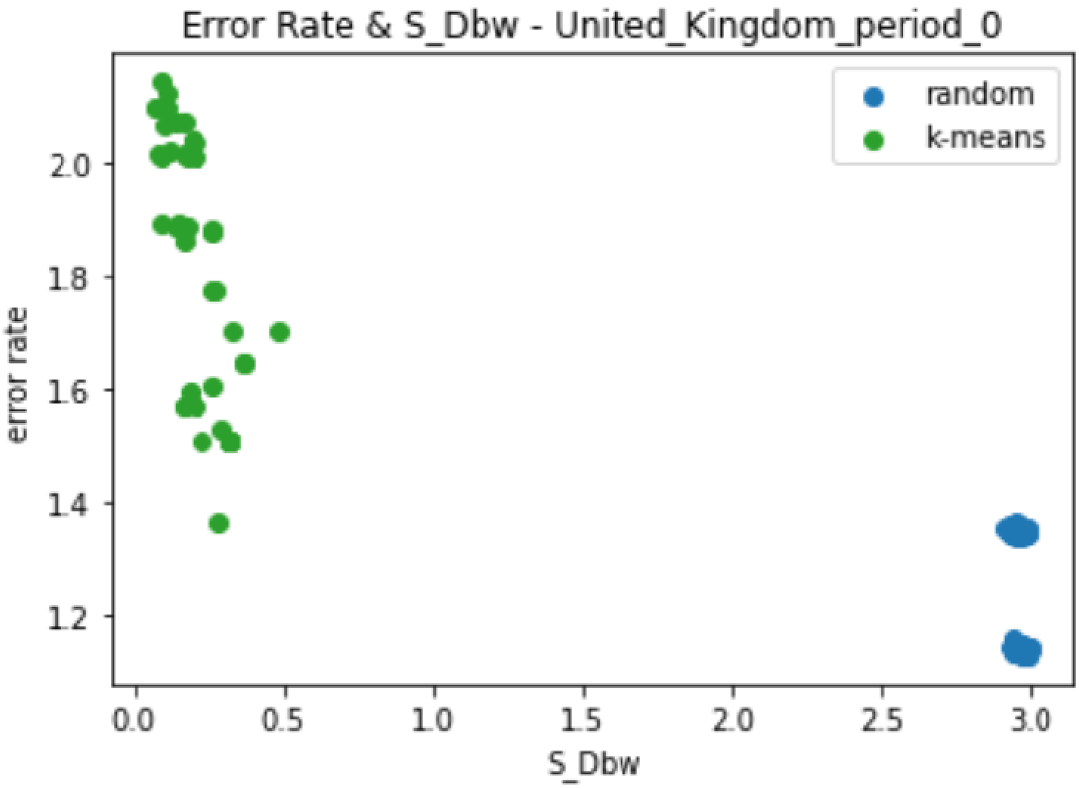


Figure 6.9: K-Means and Random Clustering - the UK-0

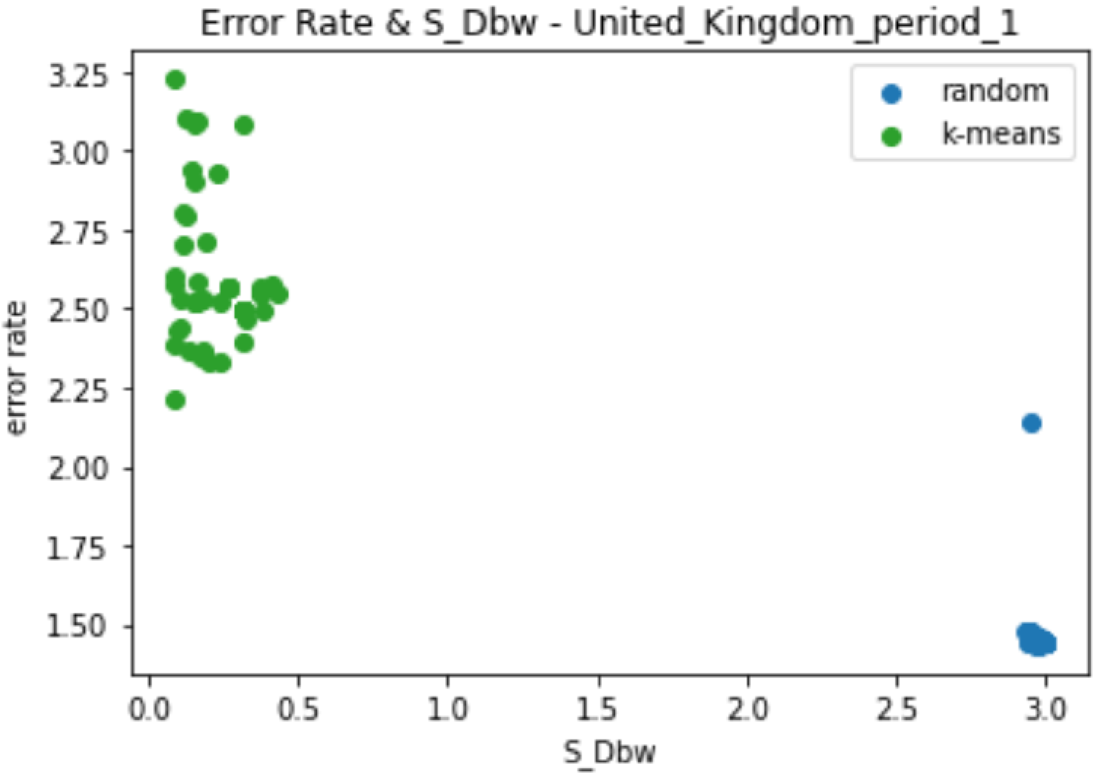


Figure 6.10: K-Means and Random Clustering - the UK-1

doi:10.6342/NTU202400040

In the UK-0, the error rate and S_Dbw of random clustering results are shown in Fig. 6.11, and the correlation matrix is shown in Table. 6.5. The correlation between S_Dbw and the error rate is -0.4812, and the correlation between n_cluster and the error rate is 0.7661.

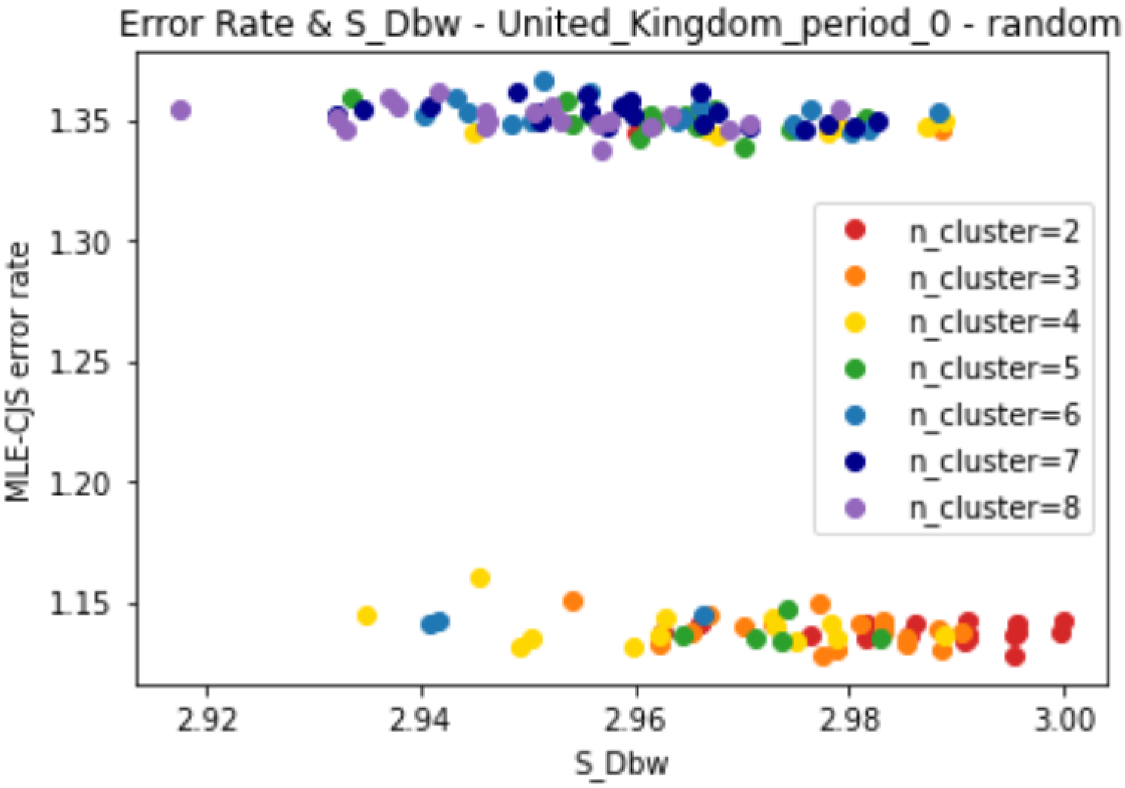


Figure 6.11: Error Rate and S_Dbw of Random Clustering - UK-0

	n_cluster	S_Dbw	error rate	stdev
n_cluster	1.0000	-0.6294	0.7661	0.7615
S_Dbw	-0.6294	1.0000	-0.4812	-0.4992
error rate	0.7661	-0.4812	1.0000	0.9924
stdev	0.7615	-0.4992	0.9924	1.0000

Table 6.5: Correlation Matrix - Random Clustering in UK-0

In the UK-1, the error rate and S_Dbw of random clustering results are shown in Fig. 6.12, and the correlation matrix is shown in Table. 6.6. The correlation between S_Dbw and the error rate is -0.2038, and the correlation between n_cluster and the error rate is 0.1589.

doi:10.6342/NTU202400040

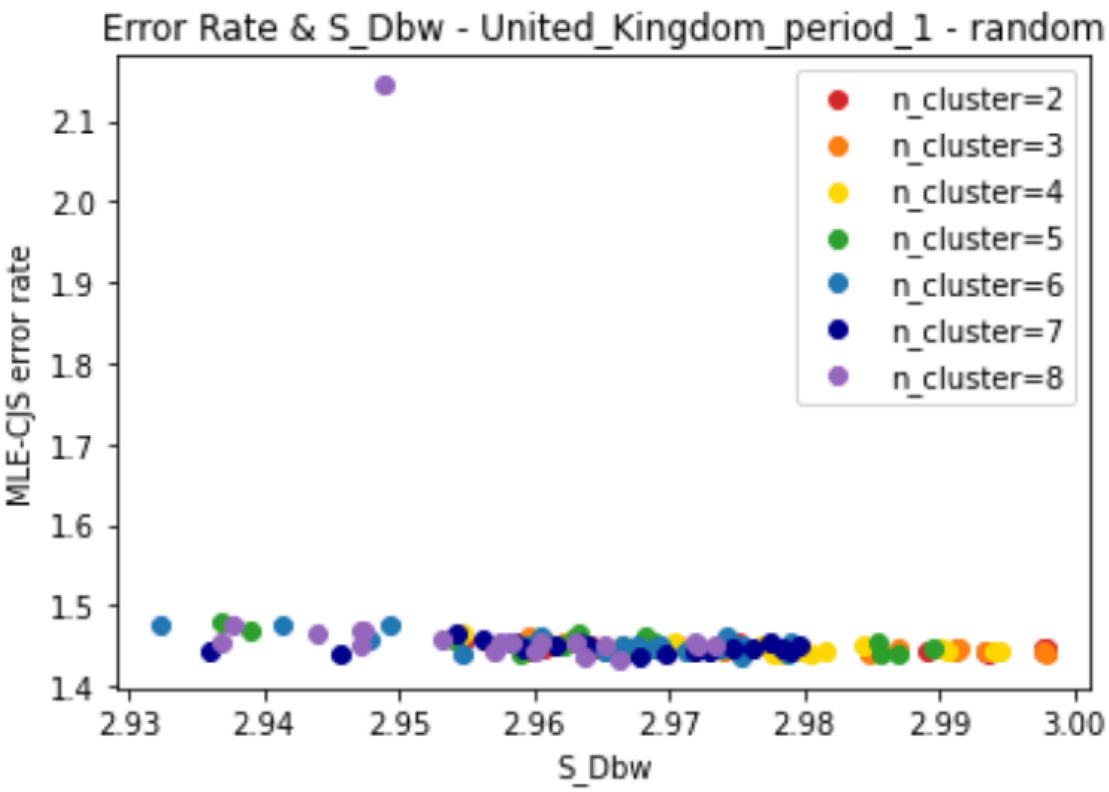


Figure 6.12: Error Rate and S_Dbw of Random Clustering - UK-1

	n_cluster	S_Dbw	error rate	stdev
n_cluster	1.0000	-0.5022	0.1589	0.0510
S_Dbw	-0.5022	1.0000	-0.2038	-0.2390
error rate	0.1589	-0.2038	1.0000	0.8422
stdev	0.0510	-0.2390	0.8422	1.0000

Table 6.6: Correlation Matrix - Random Clustering in UK-1

For each cluster in random clustering results, the cluster error rate and the Std/Avg are shown in Fig. 6.13 and Fig. 6.14. The correlations of Std/Avg and cluster error rate are 0.6836 in UK-0 and 0.7522 in UK-1, which are much larger than the correlations of random clustering in the US-0 (0.2840) and the US-1 (-0.0135). The reason why the correlation of Std/Avg and cluster error rate in the UK with random clustering results is larger than the correlation in the US with random clustering results may be explained by all the clusters of random clustering results in the US having Std/Avg < 0.34, while the Std/Avg of clusters from random clustering in the UK could up to 0.65 (UK-0) and 0.8

(UK-1). When the Std/Avg is high, Std/Avg has a more obvious relationship with cluster error rate.

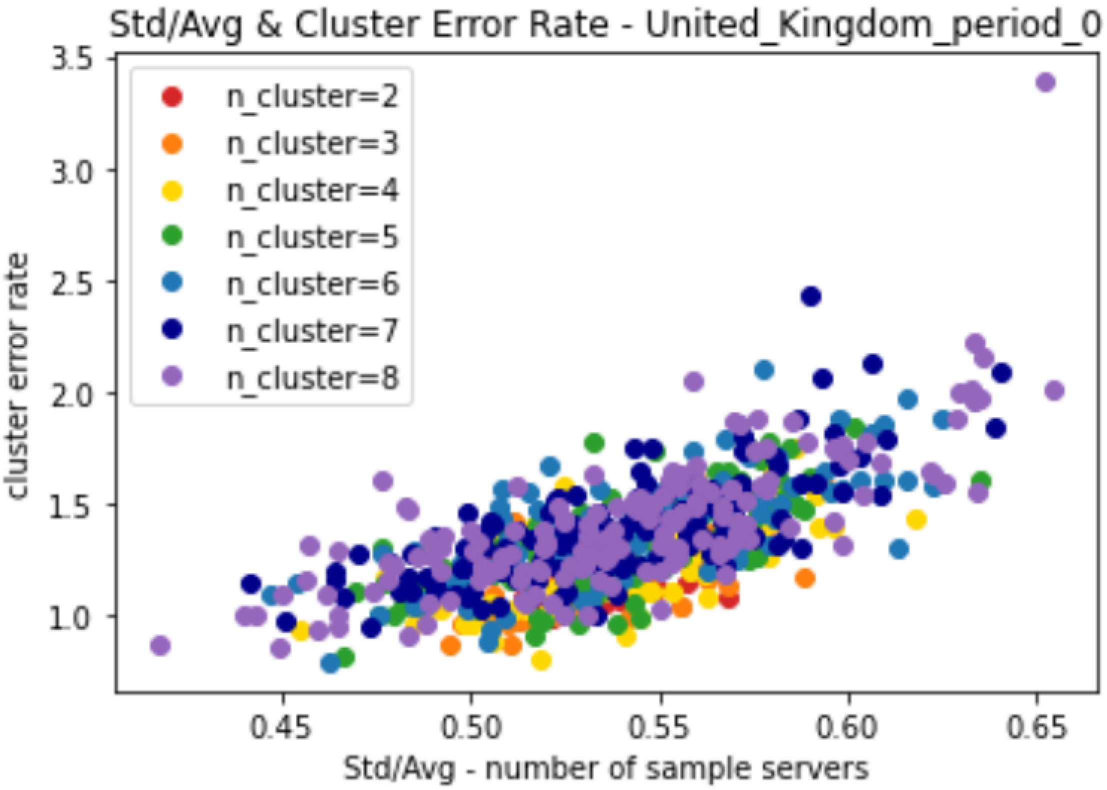


Figure 6.13: Std/Avg and Cluster Error Rate of Random Clustering - UK-0

doi:10.6342/NTU202400040

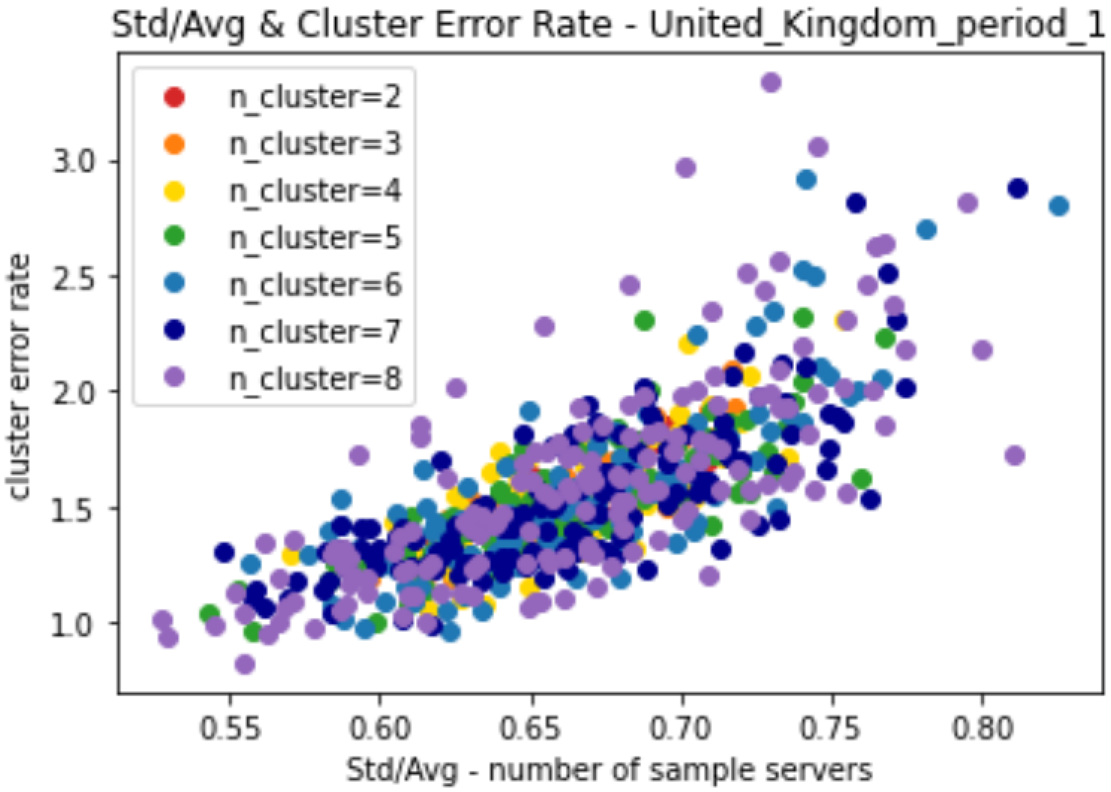


Figure 6.14: Std/Avg and Cluster Error Rate of Random Clustering - UK-1

6.3 CJS Model in France

In the France data, there is one period, April 29 to May 05, with working hours equal to 24 for more than 7 continuous days. The sample hour I choose is '19', which is the hour that contains the largest number of servers.

6.3.1 S_Dbw Score and Estimation Error Rate

The result of the CJS model without clustering is shown in Fig. 6.15. The error rate is about 6.12%, which is worse than the result in the US, but much better than the result in UK-0 and UK-1.

The results of the CJS model with clustering are shown in Fig 6.16. When the number

doi:10.6342/NTU202400040

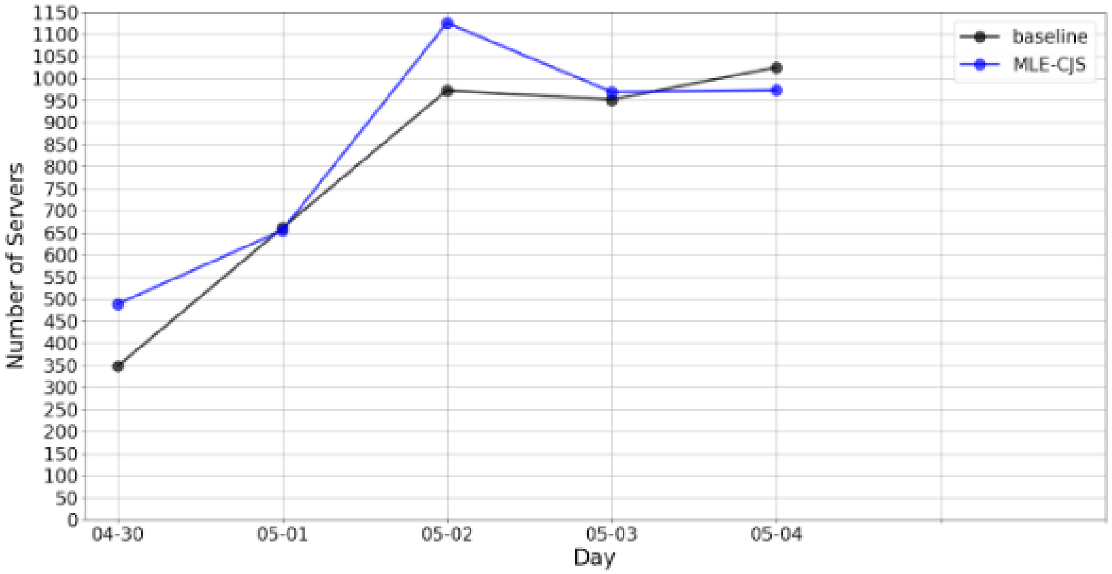


Figure 6.15: The Estimation Result of the CJS model without Clustering - France

of clusters equals 2 and 3, some CJS results achieve better error rates than the CJS model without clustering. The best error rate among all results is 1.47%, which happens when the number of clusters equals 2. On the contrary, when the number of clusters is equal to 7 and 8, all the CJS results have error rates larger than 25%.

doi:10.6342/NTU202400040

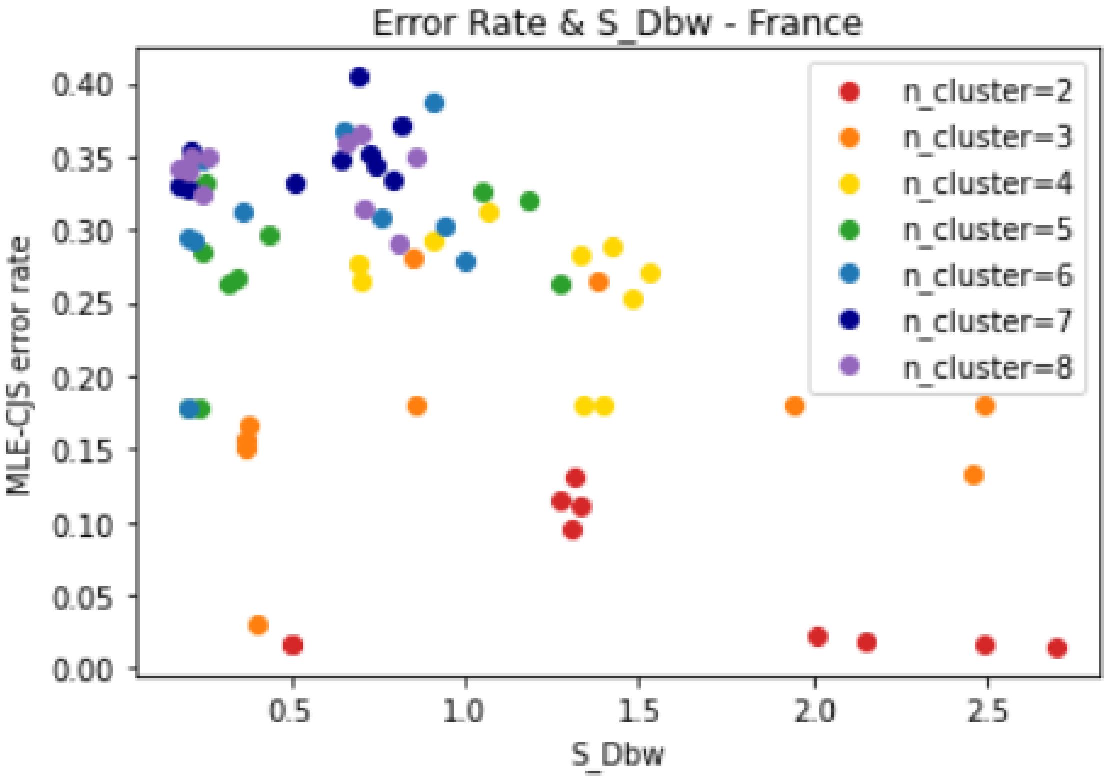


Figure 6.16: The CJS Result of France

The correlation matrix of the number of clusters, S_Dbw score, error rate, and standard deviation of error rate (stdev) in France is shown in Table. 6.7. Same as the CJS results with k-means in the US data, the correlation between S_Dbw score and error rate is negative, and the correlation between the number of clusters and error rate is positive.

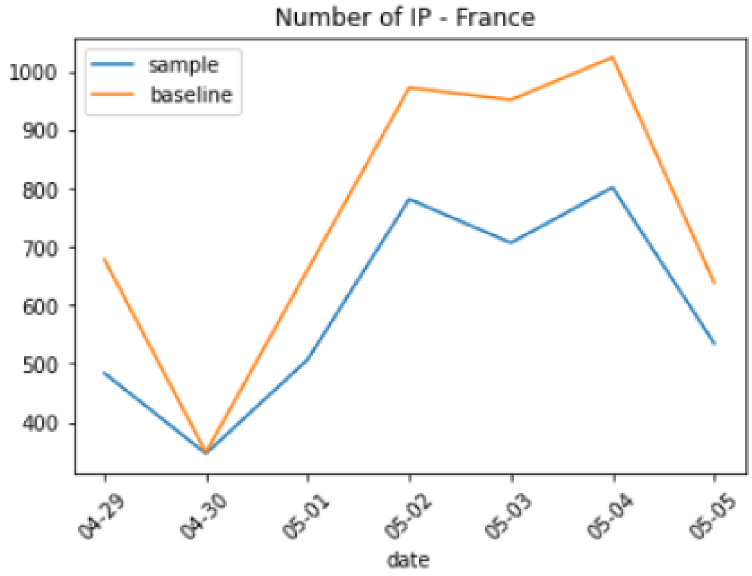
	n_cluster	S_Dbw	error rate	stdev
n_cluster	1.0000	-0.5716	0.8329	0.7510
S_Dbw	-0.5716	1.0000	-0.4871	-0.4625
error rate	0.8329	-0.4871	1.0000	0.9489
stdev	0.7510	-0.4625	0.9489	1.0000

Table 6.7: Correlation Matrix - France

doi:10.6342/NTU202400040

6.3.2 Why CJS Model Can Fit in France

To be the control group for the result in the UK, I also dig into the number of IPs on each date in France data, as shown in Fig. 6.17. The standard deviation of the number of servers in the sample hour is 158.58, and Std/Avg of the sample number without clustering is 0.27. Compared to Std/Avg in the UK, which is 0.5346 in UK-0 and 0.6575 UK-1, the number of sample servers is more stable in France. Besides, all the numbers of sample servers are larger than 300, while some numbers of sample servers are only about 100 in the UK. As a result, the error rate of the CJS model with and without clustering is much lower in France than in the UK.



std(sample_num): 158.57734866683856

Figure 6.17: The Number of IPs in Each Date - France

Also, I dig into Std/Avg and cluster error rate in k-means results, which is shown in Fig. 6.18. The maximum Std/Avg of the clusters is over 0.7, which is lower than the maximum Std/Avg of k-means results in the UK-0 and UK-1. The correlation of Std/Avg and the cluster error rate is 0.5726.

doi:10.6342/NTU202400040

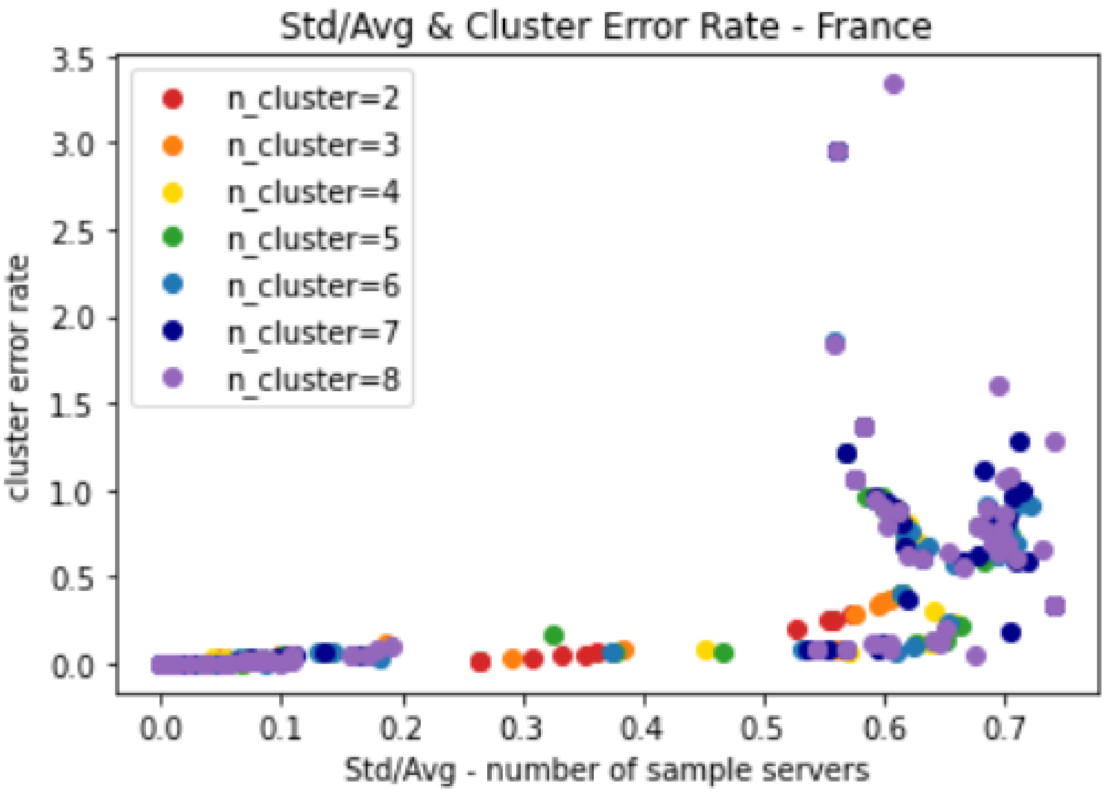


Figure 6.18: Std/Avg and Cluster Error Rate in K-Means - France

6.3.3 CJS Model with Random Clustering

In Fig. 6.19, CJS models with random clustering results obviously have lower error rates than k-means. The error rates of random clustering results are all better than the CJS model without clustering, 6.12%.

The error rate and S_Dbw of random clustering results are shown in Fig. 6.20, and the correlation matrix is shown in Table. 6.8. The correlation between S_Dbw and the error rate is 0.0655, and the correlation between n_cluster and the error rate is 0.1890. Both correlations show that error rates of random clustering results have little matter with S_Dbw and n_cluster in France.

doi:10.6342/NTU202400040

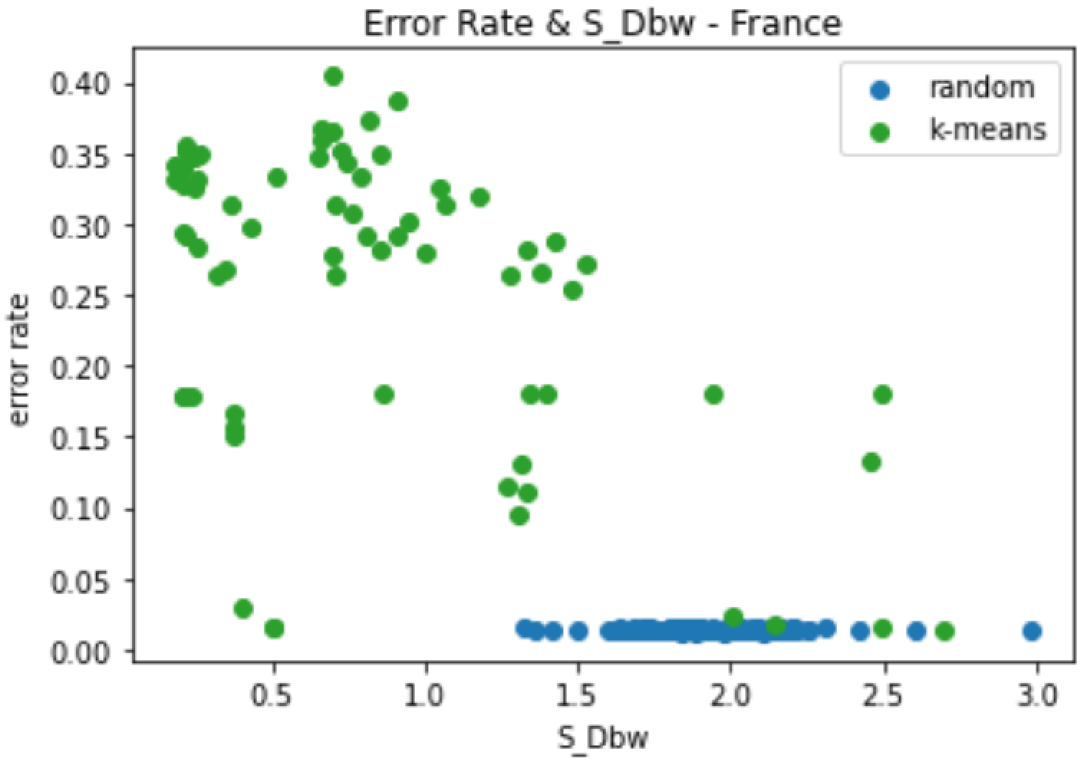


Figure 6.19: K-Means and Random Clustering - France

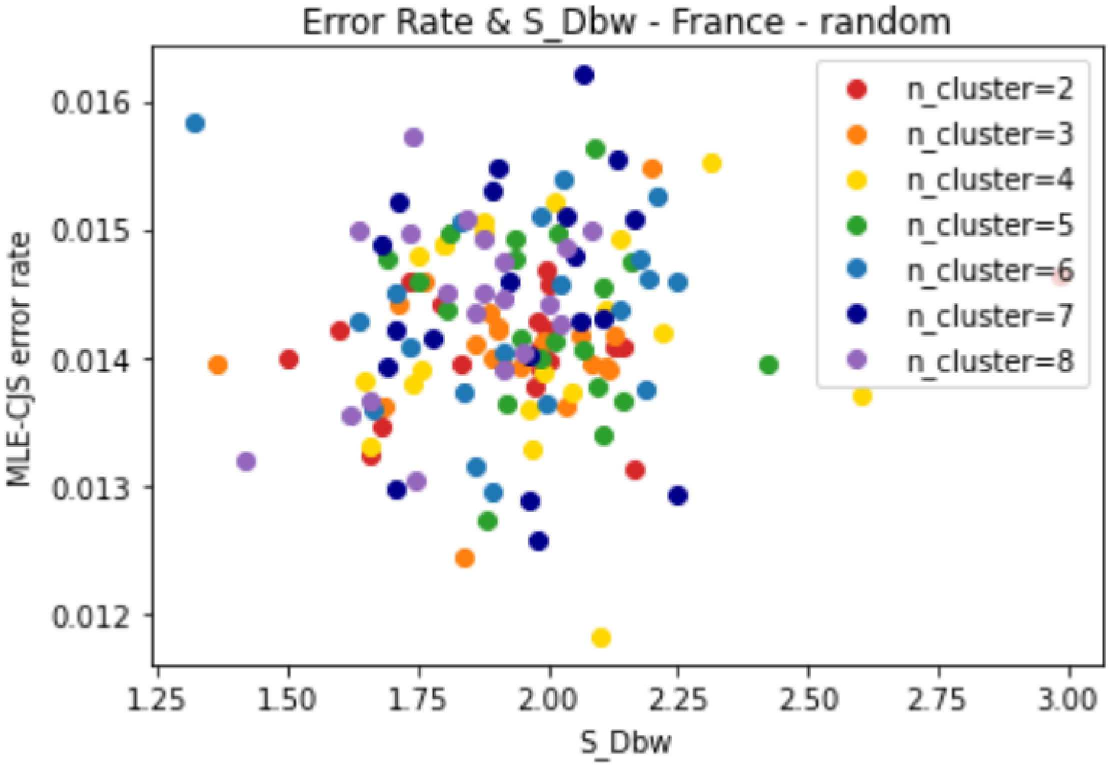


Figure 6.20: Error Rate and S_Dbw of Random Clustering - France

For each cluster in random clustering results, the cluster error rate and the Std/Avg are

	n_cluster	S_Dbw	error rate	stdev
n_cluster	1.0000	-0.1231	0.1890	-0.2066
S_Dbw	-0.1231	1.0000	0.0655	0.0354
error rate	0.1890	0.0655	1.0000	-0.0946
stdev	-0.2066	0.0354	-0.0946	1.0000

Table 6.8: Correlation Matrix - Random Clustering in France

shown in Fig. 6.21. All clusters in France data have Std/Avg less than 0.4. The correlation of Std/Avg and cluster error rate is 0.1642, which is much less than the correlations in the UK-0 (0.6836) and UK-1 (0.7522). It justifies again that for the clusters with low Std/ Avg (ex: random clustering results in the US and France), the correlations of Std/Avg and cluster error rate would be low. On the contrary, for the clusters with high Std/Avg (ex: the UK), Std/Avg matters a lot to cluster error rate.

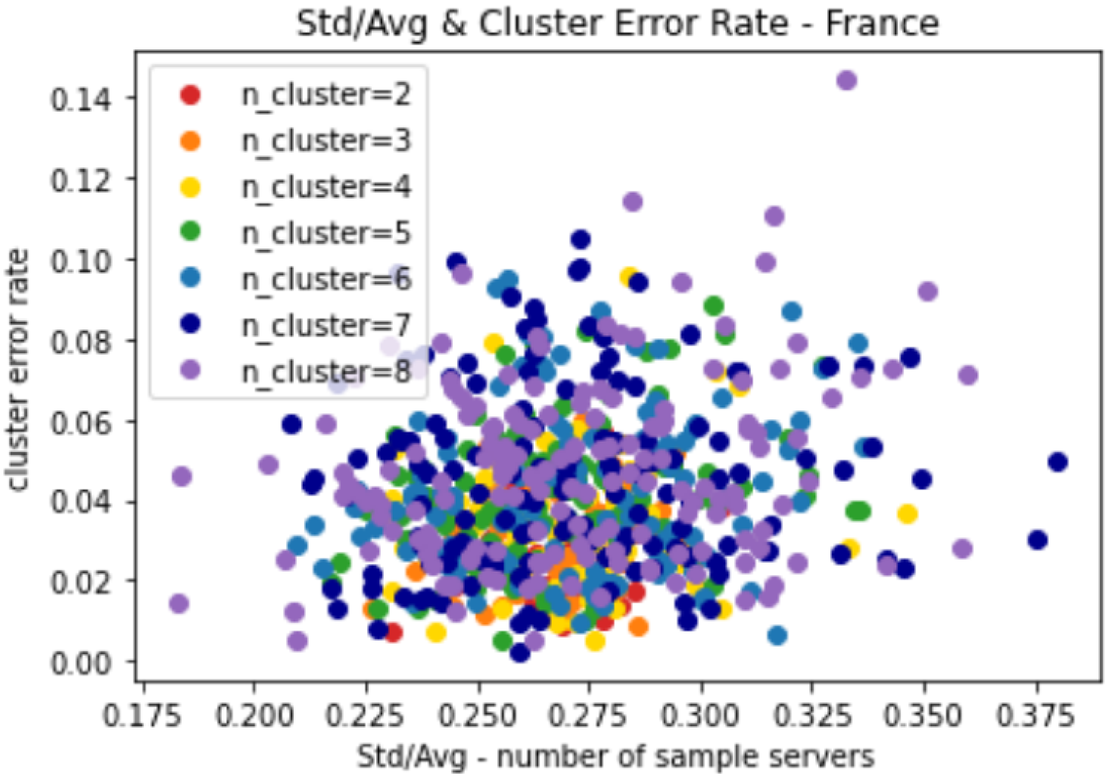


Figure 6.21: Std/Avg and Cluster Error Rate of Random Clustering - France

6.4 CJS Model in the Netherlands

In the Netherlands data, there are two periods, Netherlands-0: June 18 to June 27 and Netherlands-1: June 30 to July 13, with working hours equal to 24 for more than 7 continuous days. The sample hour I chose is '19', which is the same as the sample hour in France.

6.4.1 S_Dbw Score and Estimation Error Rate

The results of the CJS model without clustering are shown in Fig. 6.22 and Fig. 6.23. The error rates are 88.82% and 342% respectively, which are all much worse than the result in the US. On June 21 in Netherlands-0, the baseline number of servers is 17, however, the estimated number of CJS models is about 100, which contributes a lot to the mean error rate of the Netherlands-0 without clustering.

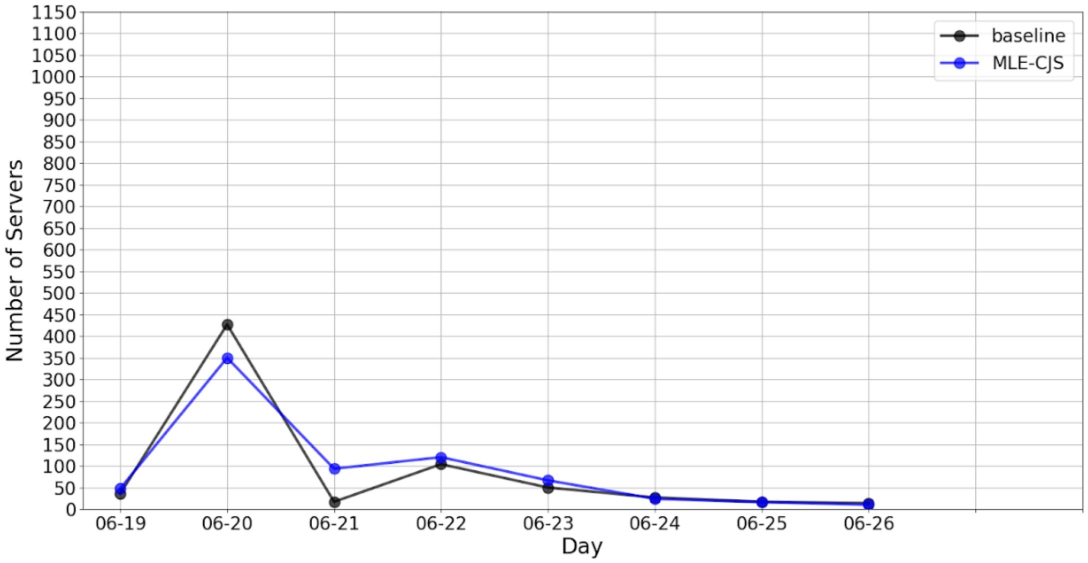


Figure 6.22: The Estimation Result of the CJS model without Clustering - Netherlands-0

doi:10.6342/NTU202400040

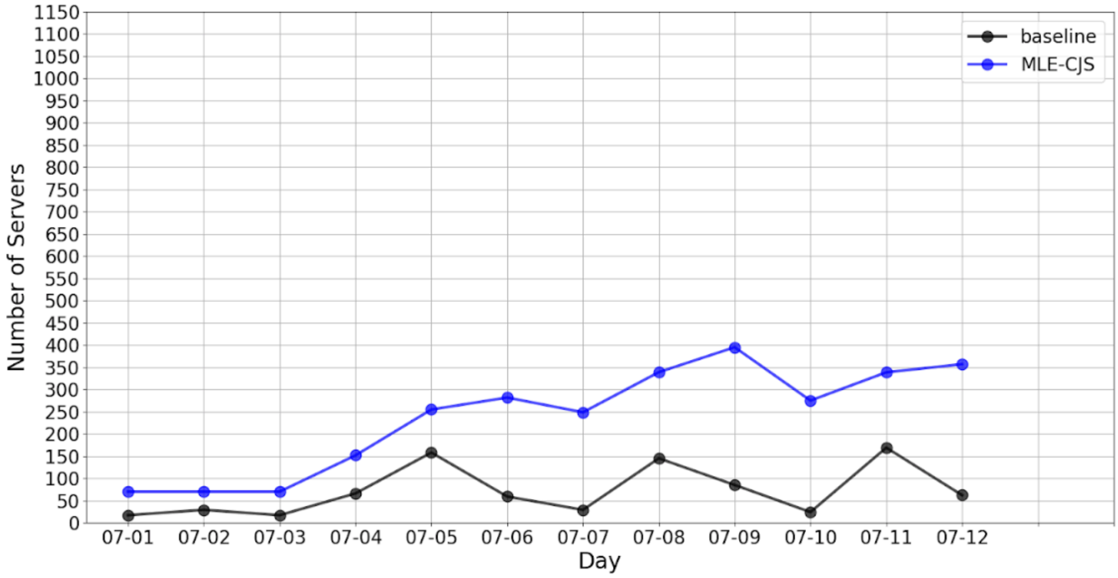


Figure 6.23: The Estimation Result of the CJS model without Clustering - Netherlands-1

The result of the CJS model with k-means is shown in Fig. 6.24 and Fig. 6.25. There 60 out of 70 results do not converge in both Netherlands-0 and Netherlands-1. All the clustering results with n_cluster > 3 do not converge.

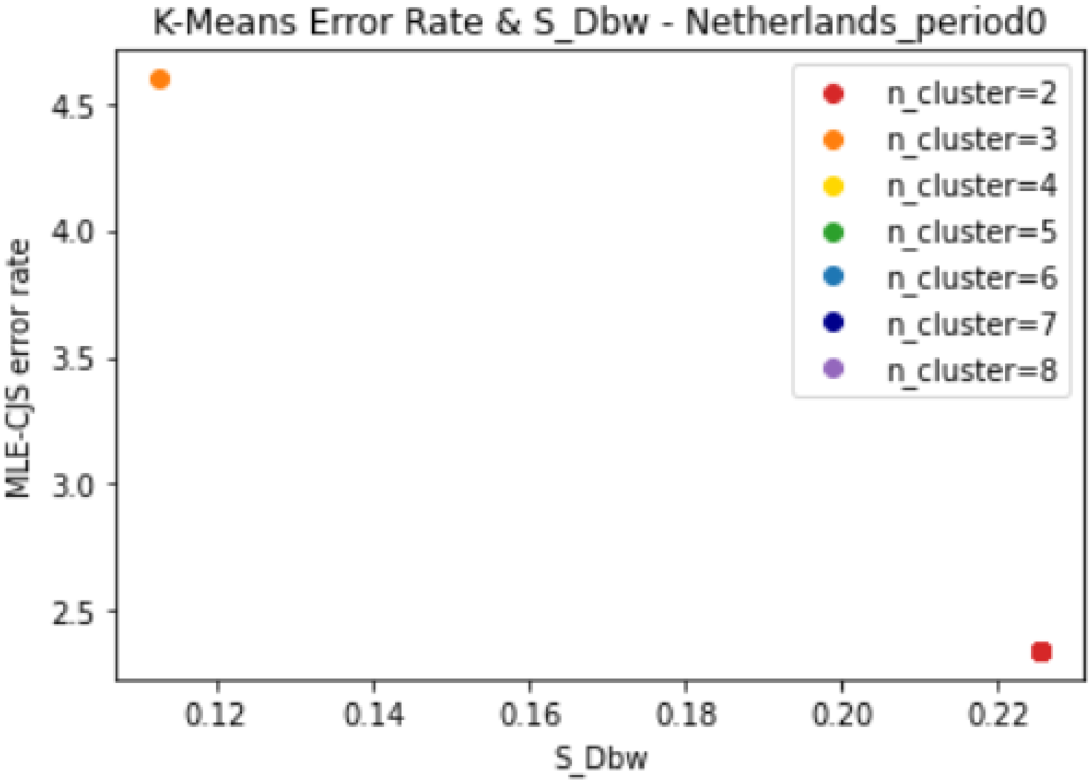


Figure 6.24: The CJS Result of the Netherlands - Netherlands-0

doi:10.6342/NTU202400040

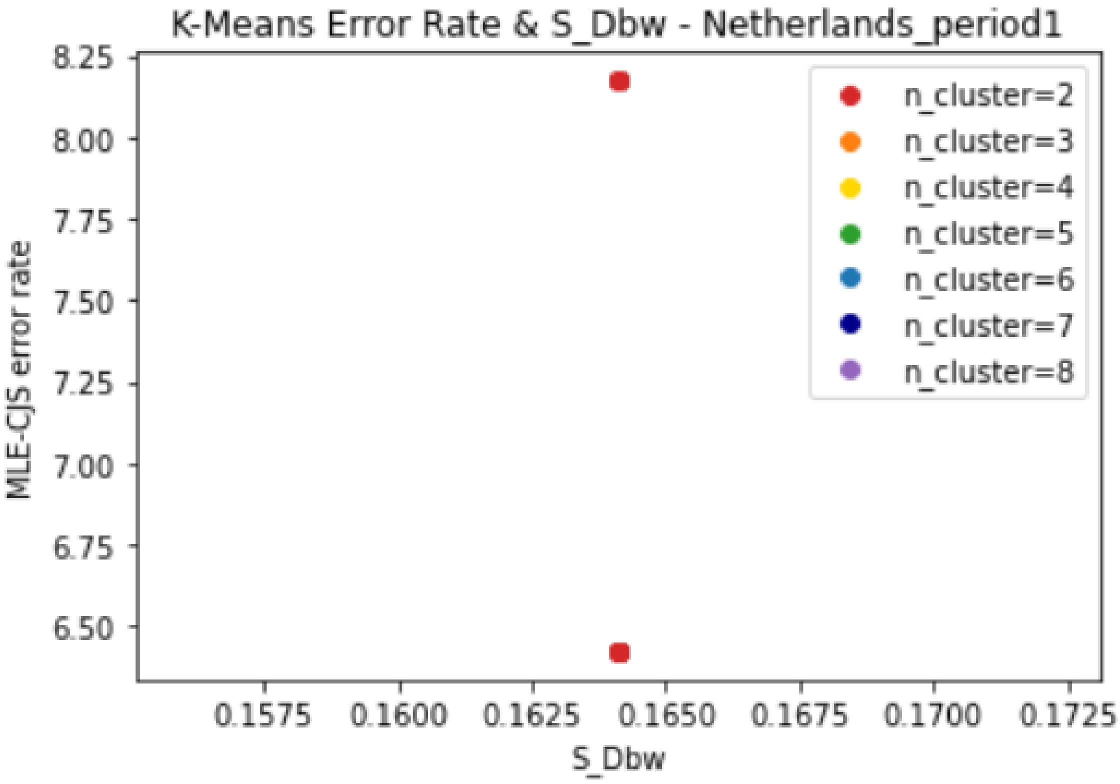
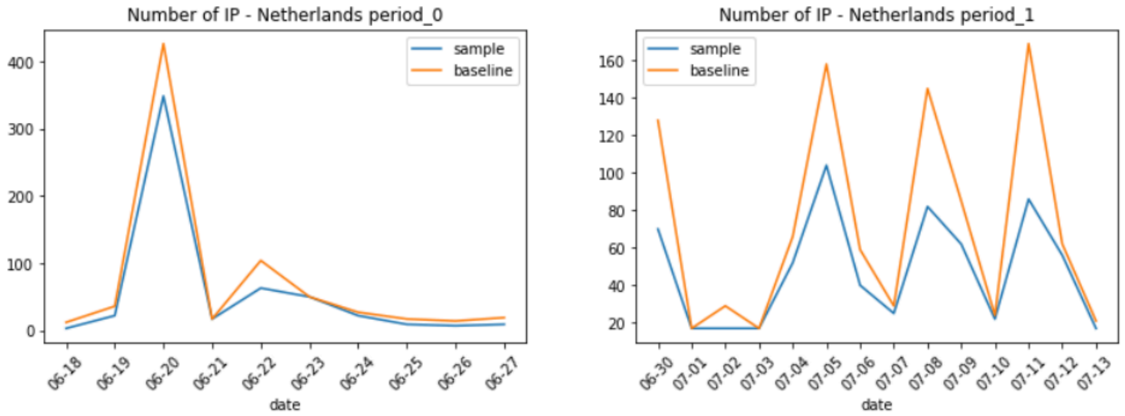


Figure 6.25: The CJS Result of the Netherlands - Netherlands-1

6.4.2 Why CJS Model Cannot Fit Well in the Netherlands

The error rates of the CJS model with clustering are very large in the Netherlands data. To find out the reason why the CJS model does not fit in the Netherlands data, I dig into the number of servers in the sample hour and the baseline of each date, which is shown in Fig. 6.26. The number of IPs in a sample hour is often less than 100, while there are only 3 days in the UK-0 and UK-1 has a sample number close to 100. The standard deviation of the number of IPs in sample hour is 99.69 and 28.79 in Netherlands-0 and Netherlands-1 respectively. Considering the average number of IPs in sample hour is 55.10 and 47.64 in Netherlands-0 and Netherlands-1, the Std/Avg of sample numbers are 1.8093 and 0.6043. As a result, many CJS results in the Netherlands data do not converge.

doi:10.6342/NTU202400040



std(sample_num): 99.69297868957473

std(sample_num): 28.79386626859751

Figure 6.26: The Number of IPs in Each Date - Netherlands

Because there are too many results that are ”not converge”, it is meaningless to discuss the correlation between Std/Avg and the cluster error rate of k-means in the Netherlands,

6.4.3 CJS Model with Random Clustering

In Fig. 6.27 and Fig. 6.28, CJS models with random clustering results have lower error rates than k-means. There are 15 out of 140 results that do not converge in Netherlands-0 and 26 out of 140 results that do not converge in Netherlands-1. I remove the not converge result from the following scatter plots. For the converging results in both Netherlands-0 and Netherlands-1, the error rates of random clustering results are all better than the CJS model without clustering.

doi:10.6342/NTU202400040

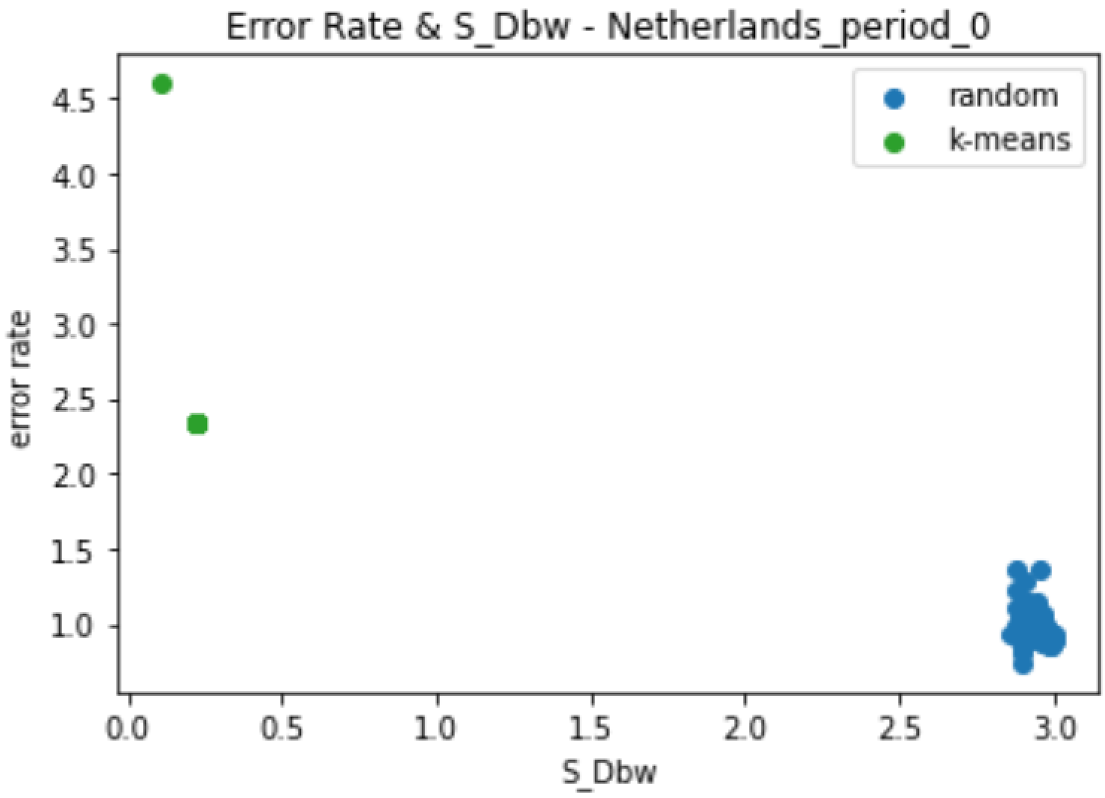


Figure 6.27: K-Means and Random Clustering - Netherlands-0

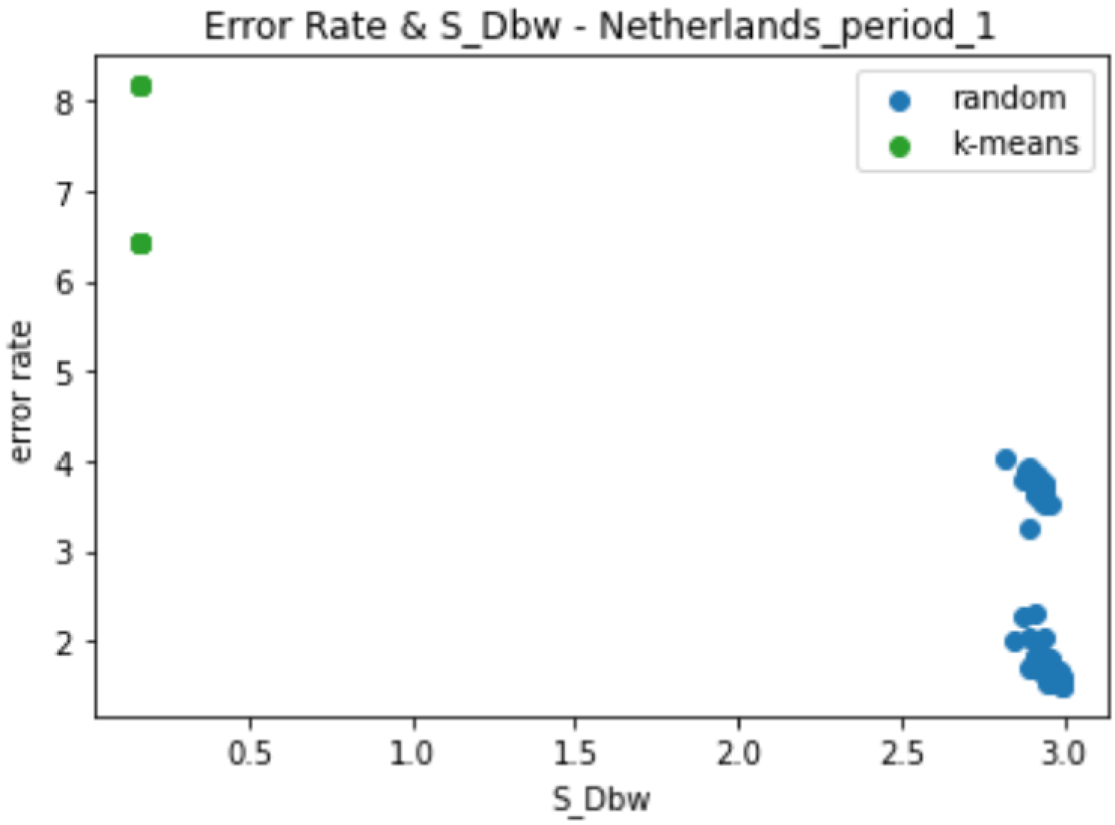


Figure 6.28: K-Means and Random Clustering - Netherlands-1

doi:10.6342/NTU202400040

In the Netherlands-0, the error rate and S_Dbw of random clustering results are shown in Fig. 6.29, and the correlation matrix is shown in Table. 6.8. The correlation between S_Dbw and the error rate is -0.4164, and the correlation between n_cluster and the error rate is 0.3714.

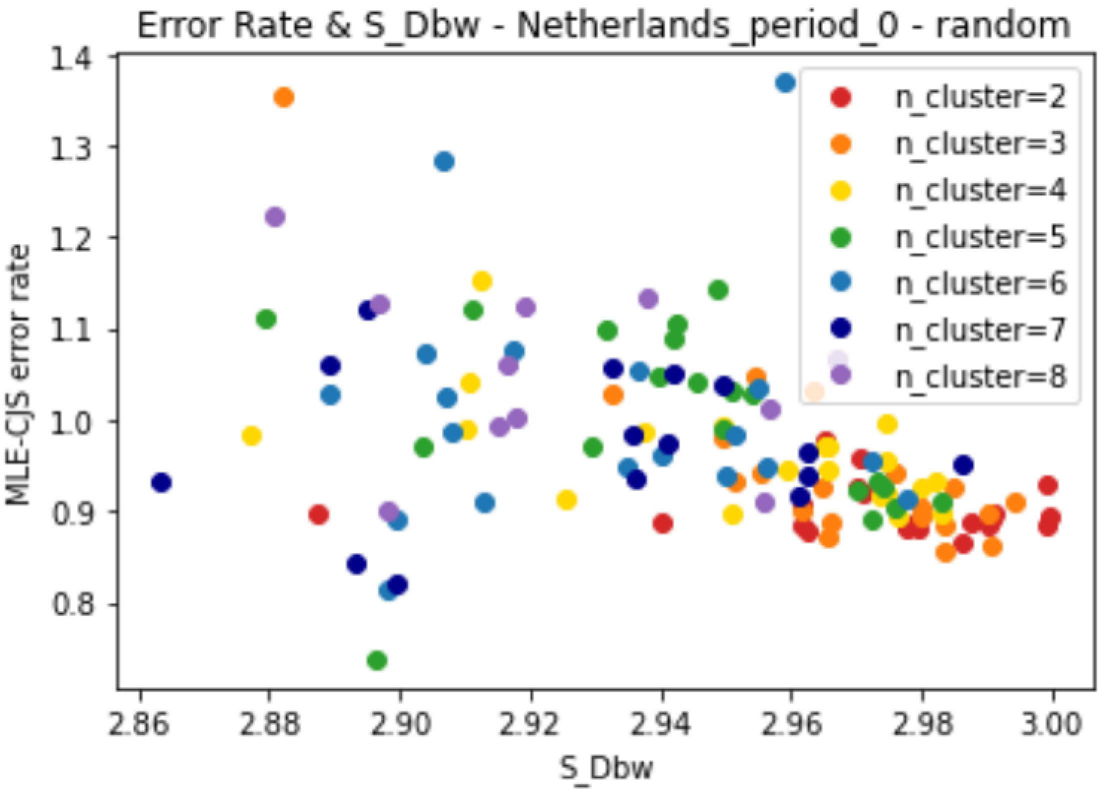


Figure 6.29: Error Rate and S_Dbw of Random Clustering - Netherlands-0

	n_cluster	S_Dbw	error rate	stdev
n_cluster	1.0000	-0.5198	0.3714	-0.0773
S_Dbw	-0.5198	1.0000	-0.4164	0.1426
error rate	0.3714	-0.4164	1.0000	0.4641
stdev	-0.0773	0.1426	0.4641	1.0000

Table 6.9: Correlation Matrix - Random Clustering in Netherlands-0

In the Netherlands-1, the error rate and S_Dbw of random clustering results are shown in Fig. 6.30, and the correlation matrix is shown in Table. 6.10. The correlation between S_Dbw and the error rate is -0.6240, and the correlation between n_cluster and the error rate is 0.3379.

doi:10.6342/NTU202400040

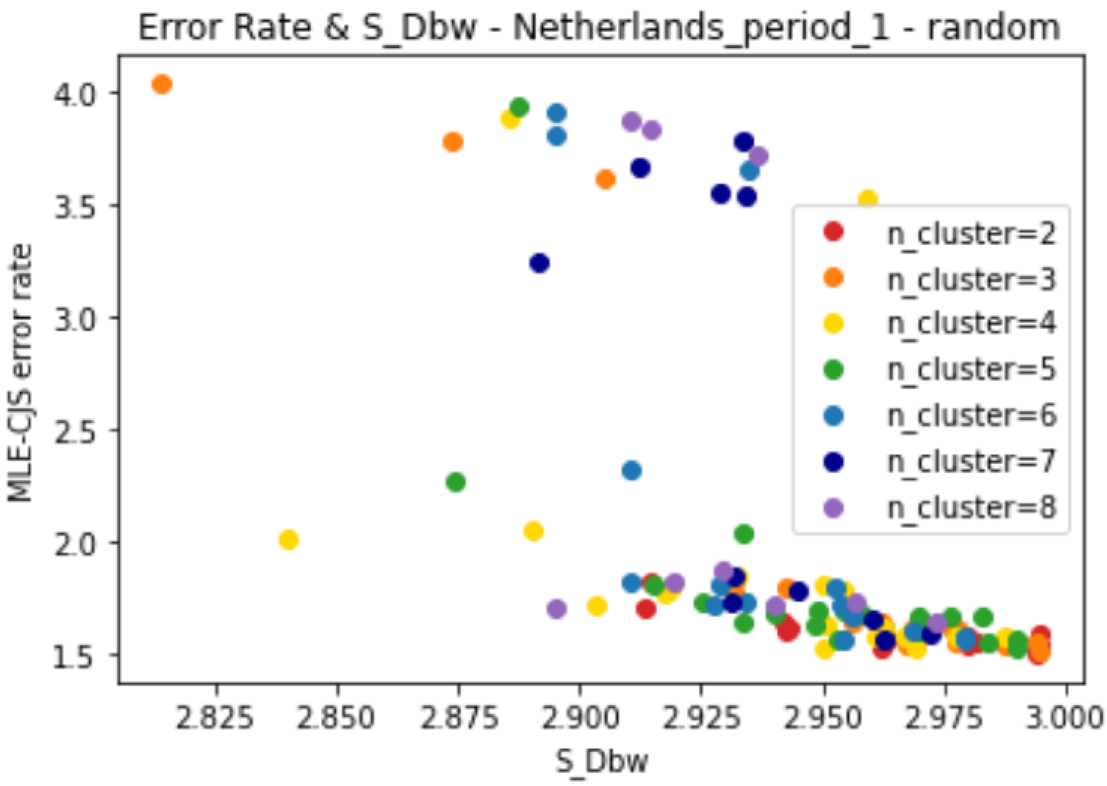


Figure 6.30: Error Rate and S_Dbw of Random Clustering - Netherlands-1

	n_cluster	S_Dbw	error rate	stdev
n_cluster	1.0000	-0.2892	0.3379	0.3390
S_Dbw	-0.2892	1.0000	-0.6240	-0.6431
error rate	0.3379	-0.6240	1.0000	0.9907
stdev	0.3390	-0.6431	0.9907	1.0000

Table 6.10: Correlation Matrix - Random Clustering in Netherlands-1

The cluster error rate and the Std/Avg in the Netherlands-0 and Netherlands-1 are shown in Fig. 6.31 and Fig. 6.32. The correlations of Std/Avg and cluster error rate are 0.5069 in Netherlands-0 and 0.6937 in Netherlands-1. It shows again that the data with high Std/Avg (ex: the UK and the Netherlands), would have a higher correlation of Std/Avg and cluster error rate than the data with low Std/Avg (ex: random clustering results in the US and France).

doi:10.6342/NTU202400040

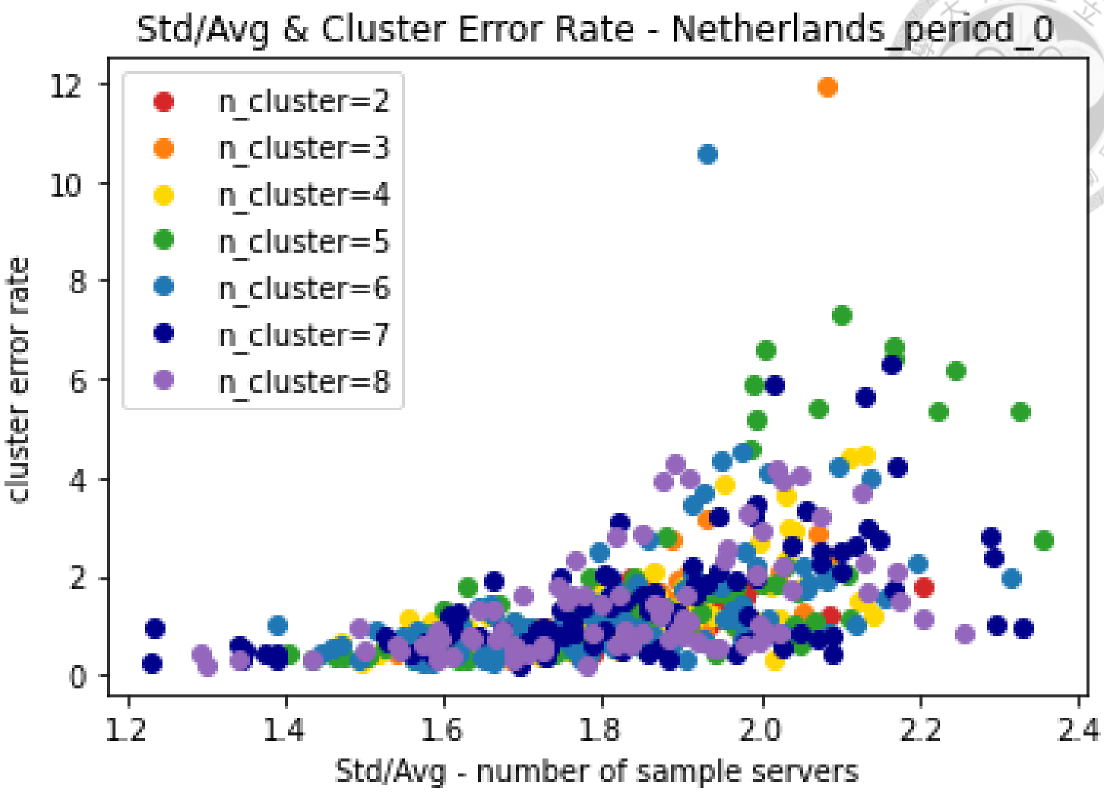


Figure 6.31: Std/Avg and Cluster Error Rate of Random Clustering - Netherlands-0

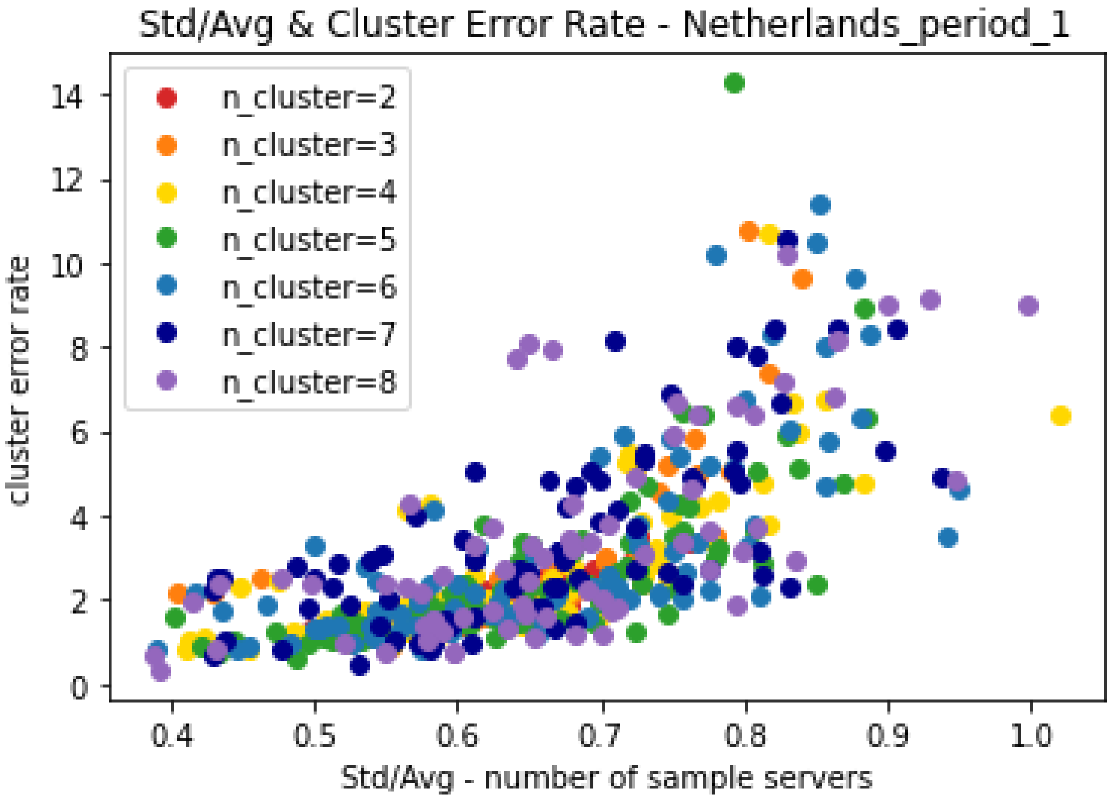


Figure 6.32: Std/Avg and Cluster Error Rate of Random Clustering - Netherlands-1

doi:10.6342/NTU202400040

6.5 CJS Model in Germany

In the German data, there is one period, June 04 to June 14, with working hours equal to 24 for more than 7 continuous days. The sample hour is ’19’, which is the same as the sample hour of France and the Netherlands.

6.5.1 S_Dbw Score and Estimation Error Rate

The result of the CJS model without clustering is shown in Fig. 6.33. The error rate is about 61.10%, which is much worse than the result in the US and France.

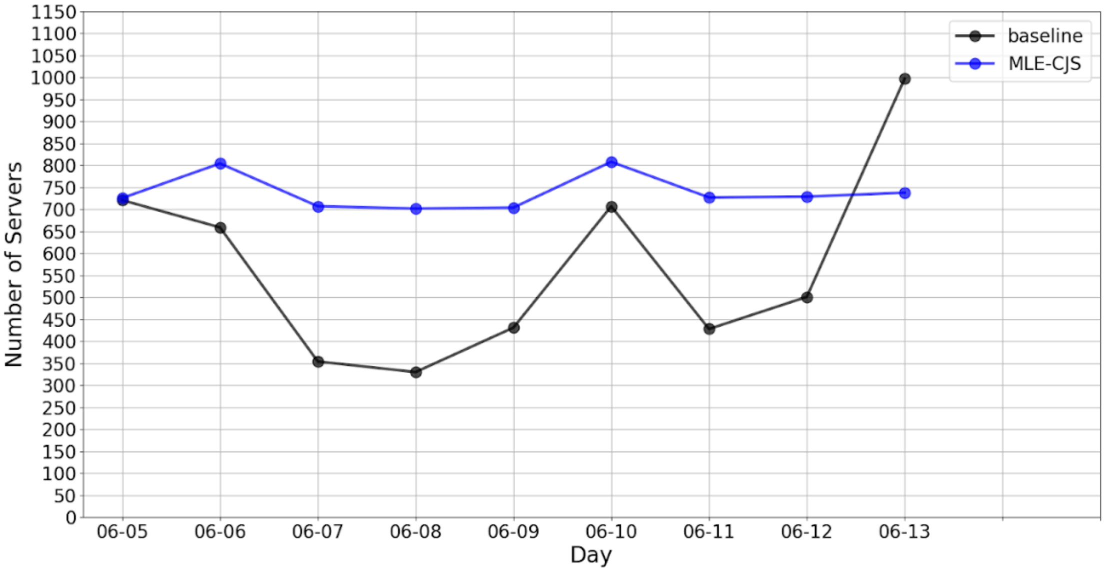


Figure 6.33: The Estimation Result of the CJS model without Clustering - Germany

Fig 6.34 shows the CJS results without error rates > 10000% (not converge). There are 6 out of 70 results that do not converge. All the CJS results have an error rate larger than 40%, which is better than the results in the UK and the Netherlands but worse than the results in the US and France.

doi:10.6342/NTU202400040

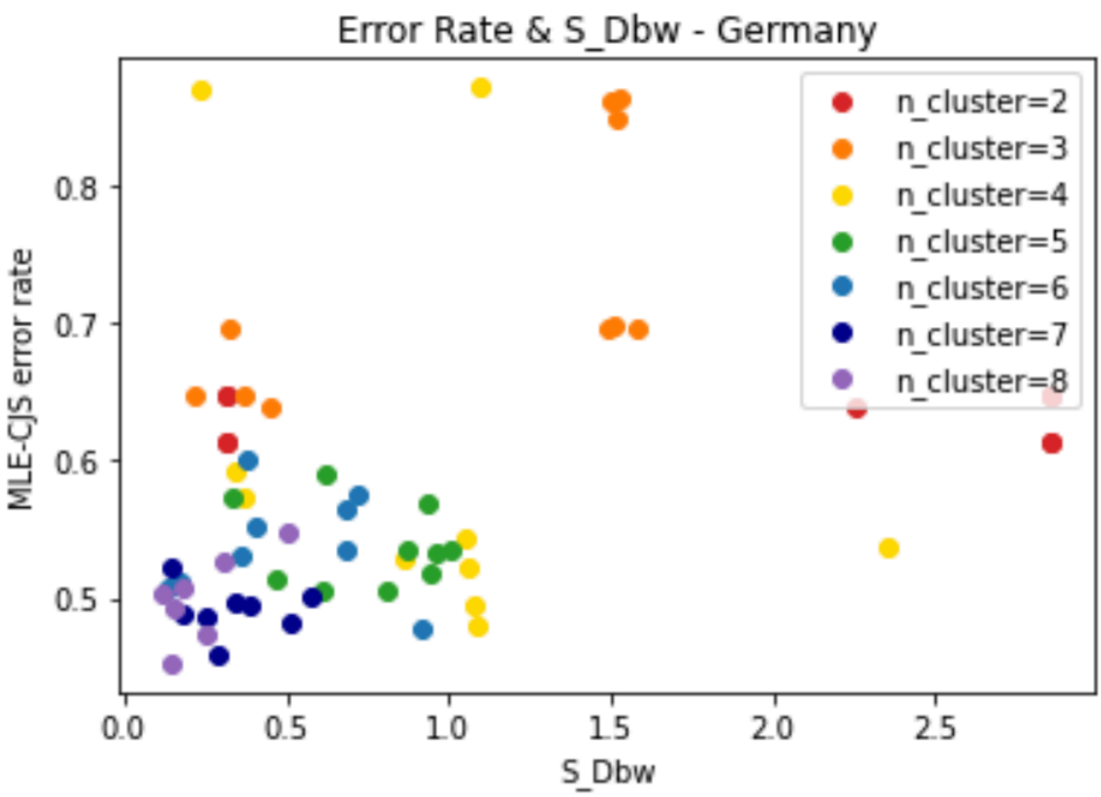
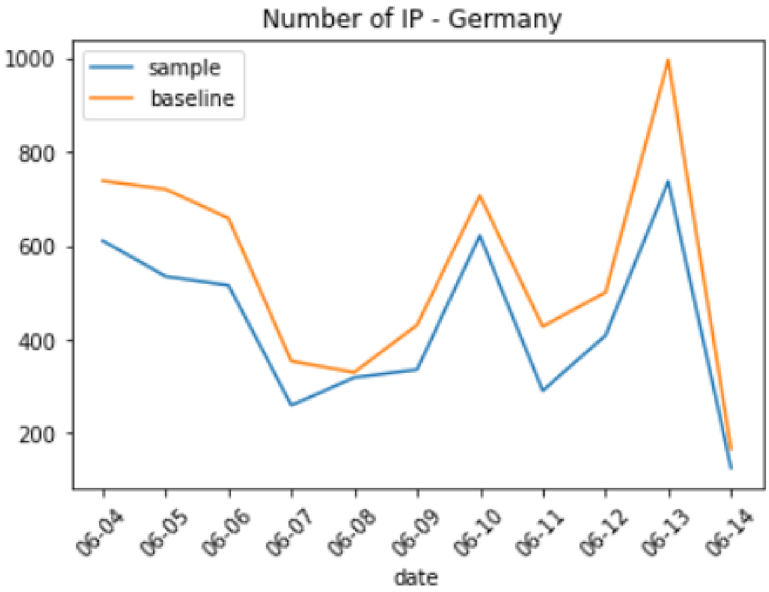


Figure 6.34: The CJS Result of Germany

6.5.2 Why CJS Model Cannot Fit Well in Germany

To find out the reason why the CJS model does not fit well in the German data, I dig into the number of servers in sample hour and the baseline of each date, which is shown in Fig. 6.35. The standard deviation of the number of servers in the sample hour is 177.11, and the Std/Avg of the sample number without clustering is 0.4090, which is larger than the Std/Avg in France data (0.27) and the US data (US-0: 0.1109, US-1: 0.2858). Besides, on June 07, June 11, and June 14, the sample number is below 300. For the CJS model without clustering and the CJS model with the number of clustering equal to 2 and 3, all the estimation numbers have error rates larger than 100% on June 07, which decreases over 40% of the number of servers in baseline compared to the last day. The sudden decrease in sample number also causes a large bias in the CJS model.

doi:10.6342/NTU202400040



std(sample_num): 177.1096219346035

Figure 6.35: The Number of IPs in Each Date - Germany

Also, I dig into Std/Avg and cluster error rate in ”converge” k-means results (error rate < 10000%), which is shown in Fig. 6.36. The maximum Std/Avg of the clusters is over 1, which is higher than the maximum Std/Avg in France. The correlation of Std/Avg and the cluster error rate is 0.8279.

doi:10.6342/NTU202400040

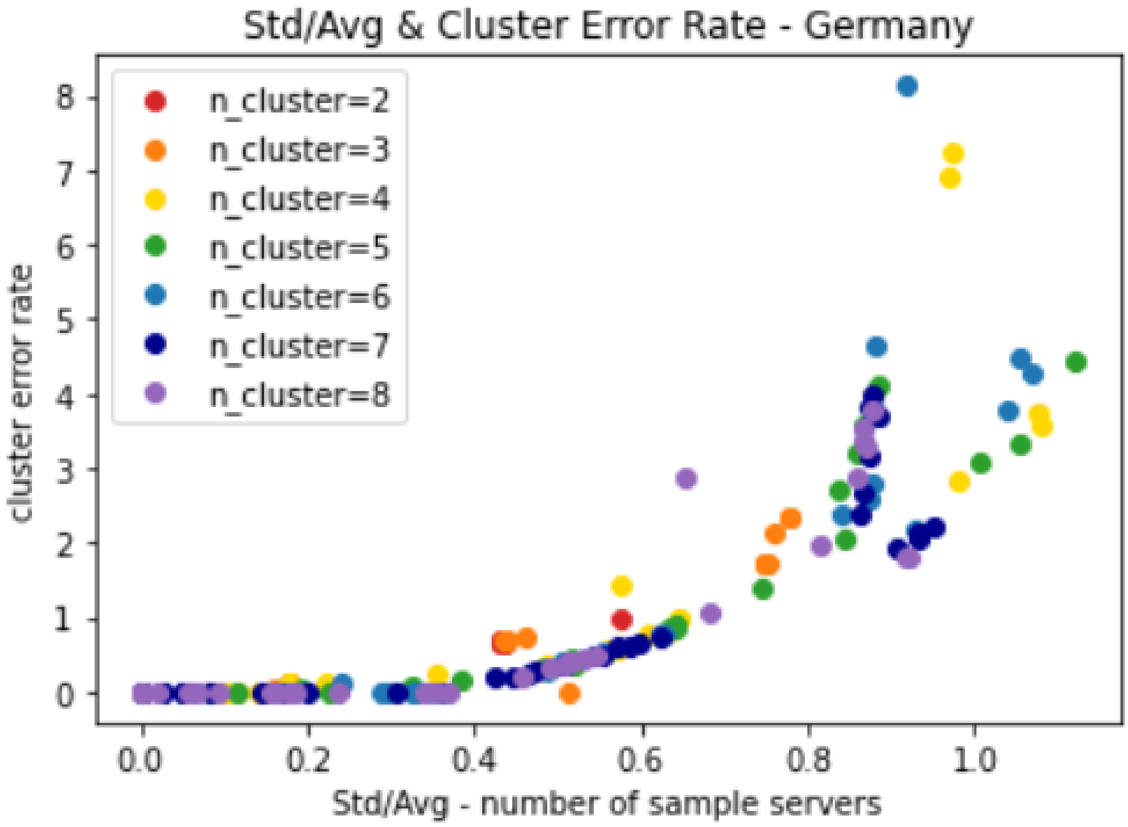


Figure 6.36: Std/Avg and Cluster Error Rate in K-Means - Germany

6.5.3 CJS Model with Random Clustering

The CJS models with random clustering results and k-means are shown in Fig. 6.37. The error rates of random clustering results are concentrated in 0.6 to 0.7, while The error rates of k-means results range from 0.4 to 0.9.

doi:10.6342/NTU202400040

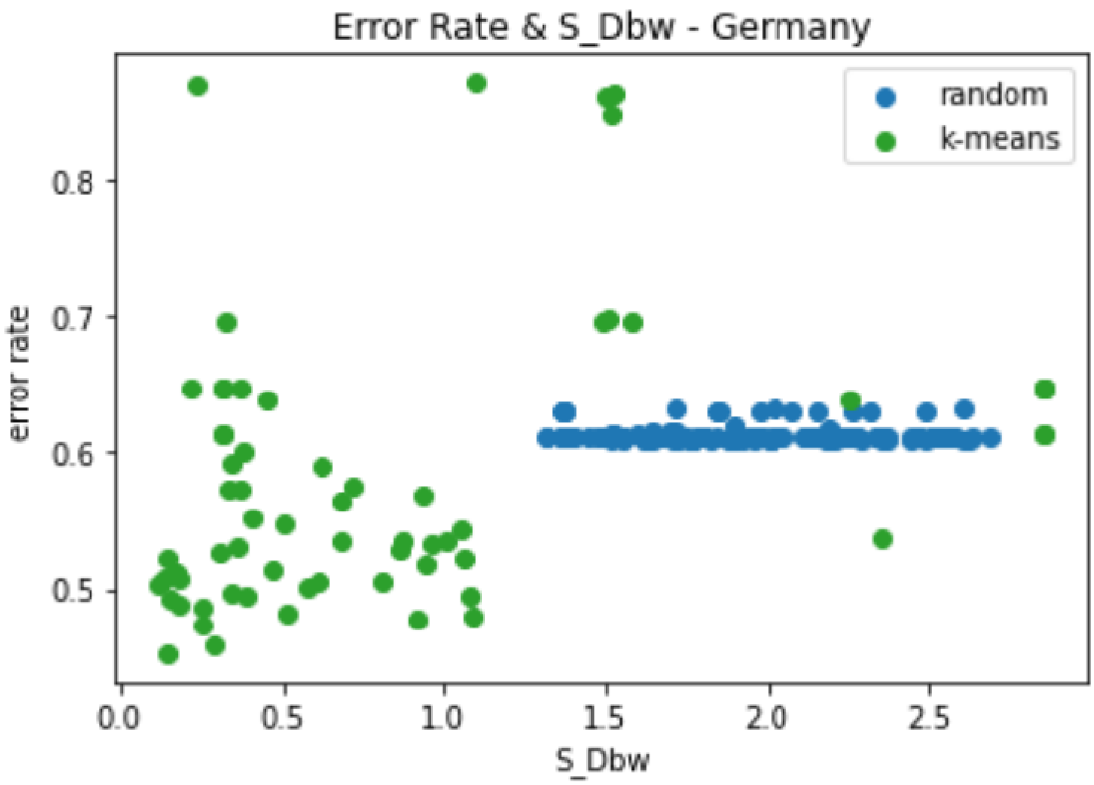


Figure 6.37: K-Means and Random Clustering - Germany

The error rate and S_Dbw of random clustering results are shown in Fig. 6.38, and the correlation matrix is shown in Table. 6.11. The correlation between S_Dbw and the error rate is -0.0324, and the correlation between n_cluster and the error rate is -0.3484.

doi:10.6342/NTU202400040

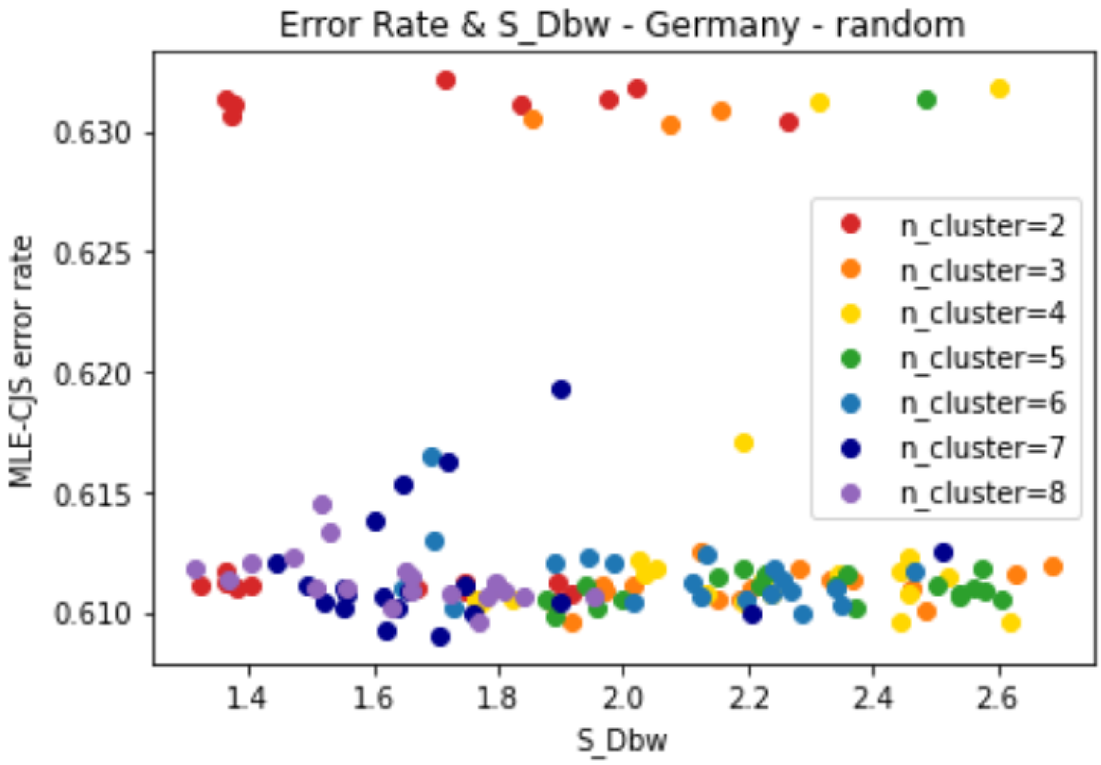


Figure 6.38: Error Rate and S_Dbw of Random Clustering - Germany

	n_cluster	S_Dbw	error rate	stdev
n_cluster	1.0000	-0.2305	-0.3484	0.0619
S_Dbw	-0.2305	1.0000	-0.0324	-0.1215
error rate	-0.3484	-0.0324	1.0000	0.7900
stdev	0.0619	-0.1215	0.7900	1.0000

Table 6.11: Correlation Matrix - Random Clustering in Germany

For each cluster in every clustering result in Germany, the cluster error rate and the Std/Avg are shown in Fig. 6.39. The correlation of Std/Avg and the cluster error rate is 0.8401.

doi:10.6342/NTU202400040

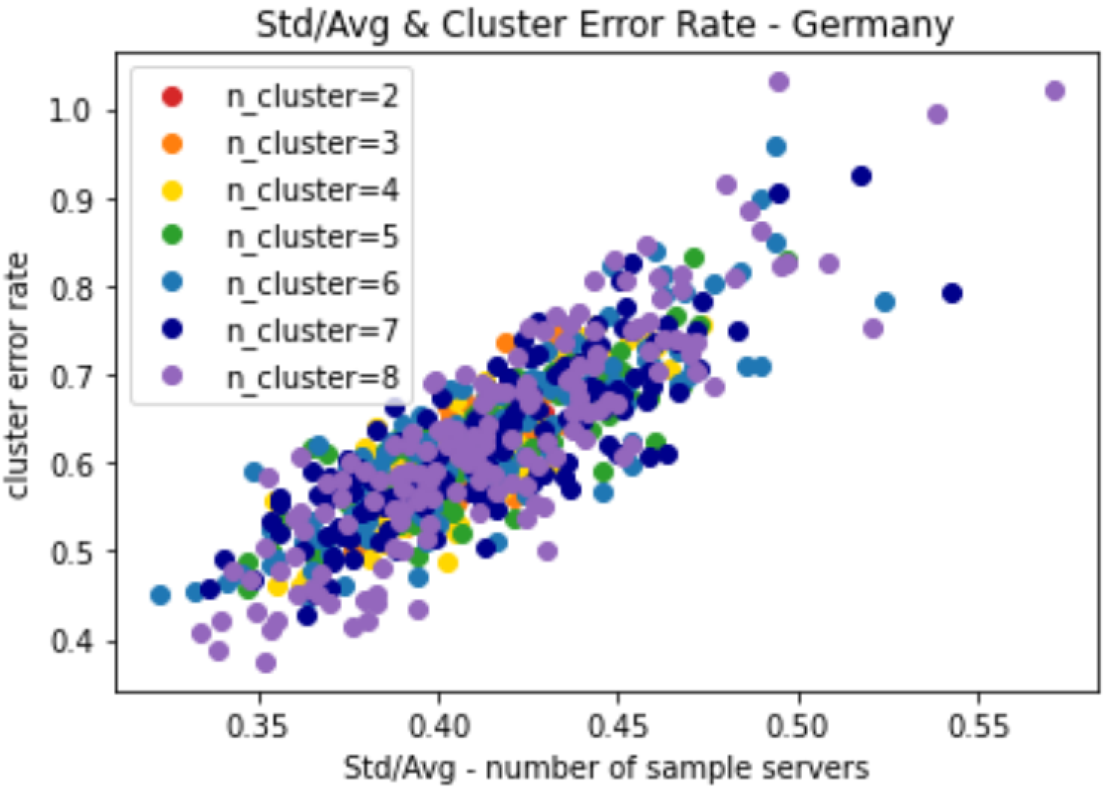


Figure 6.39: Std/Avg and Cluster Error Rate of Random Clustering - Germany

Take the random clustering results in different regions as examples, for the data with low Std/Avg such as the US and France, the correlations of Std/Avg and cluster error rate are relatively low (US-0: 0.2840, US-1: -0.0135, France: 0.1642). On the other hand, for the data with high Std/Avg such as the UK, the Netherlands, and Germany, the correlations of Std/Avg and cluster error rate are relatively high (UK-0: 0.6836, UK-1: 0.7522, Netherlands-0: 0.5069, Netherlands-1: 0.6937, Germany: 0.8401). In conclusion, when Std/Avg is high, Std/Avg could be a good metric for cluster error rate.

doi:10.6342/NTU202400040

Chapter 7 Computation Time of the CJS Model

In my research, how to avoid high computation overhead is one of the key performances in the CJS model. In this chapter, I show the computation time in the CJS model of different periods from different regions.

7.1 Computation Time in the US Data

In the US data with k-means clustering results, the computation time with the number of clusters = 2 to 8 is shown in Fig. 7.1 . All the computation time in the US-0 is less than 5 seconds, and all the computation time in the US-1 is less than 14 seconds. The period length of the US-1 is longer than the US-0, thus, the computation times in the US-1 are longer than in the US-0. Furthermore, the correlations between computation time and number of clusters are 0.9755 and 0.9708 in in the US-0 and US-1. As a result, the computation time in Fig. 7.1 is close to a linear model.

doi:10.6342/NTU202400040

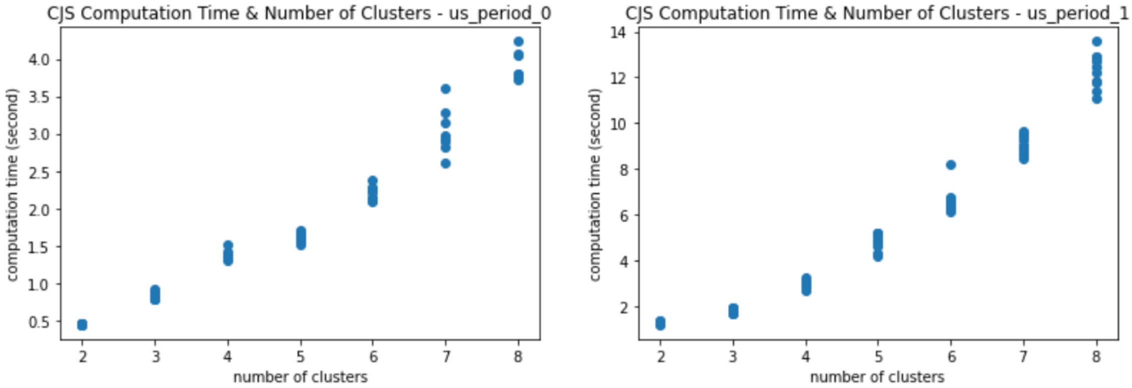


Figure 7.1: Computation Time of K-Means Clustering Results in the US

7.2 Computation Time in Different Regions

To discover how the clustering helps the CJS model avoid high computation overhead in different data, I dig into the correlation between the number of clusters and the computation time of the CJS in different regions, shown in Table. 7.1. The ’correlation’ in Table. 7.1 represents the correlation between CJS computation time and number of clusters. All the correlations between CJS computation time and the number of clusters are higher than 0.9472, which means it is close to a perfect positive correlation. As the number of clusters gets larger, the computation time of the CJS will tend to have a linear growth.

All the CJS models with k-means take less than 1 minute. The computation time and the number of clusters are close to a linear model when the number of clusters is 2 to 8. However, when the n_cluster is too high, the CJS may not converge since the capture histories in some classes are too small to have enough data to derive for all parameters. Besides, the CJS computation time would increase as the period length gets longer. One could decide on an upper limit on the number of clusters based on the time constraints of the CJS model.

data	period length (day)	correlation	max time (second)
US-0	7	0.9755	4.24
US-1	10	0.9708	13.58
UK-0	7	0.9588	6.15
UK-1	9	0.9562	15.23
France	7	0.9584	6.97
Netherlands-0	10	0.9751	8.82
Netherlands-1	14	0.9472	41.21
Germany	11	0.9585	45.15

Table 7.1: Computation Time in the CJS model with K-Means

doi:10.6342/NTU202400040

doi:10.6342/NTU202400040

Chapter 8 Discussions

8.1 Online Clustering

To explore the clustering results with different periods of data, I try to do clustering online in the US data. Clustering online means clustering with the data collected so far. For example, the result of the first date, April 13, is used in the data on April 13. The result of the 10th date, April 22, is used in the data in the first 10th date. In the beginning, I used one-time k-means for online clustering, as shown in Fig. 8.1. Since the n_cluster with the best S_Dbw scores on different dates are unstable, I try to use mean-shift clustering to let the algorithm choose n_cluster automatically, as shown in Fig. 8.2. The S_Dbw scores of mean shift float a lot in the first few days. After April 29, the S_Dbw scores of the mean shift are often less than 0.1. The number of clusters mean-shift chosen is not fixed. Mean shift tends to choose more n_cluster when the data contains more dates.

doi:10.6342/NTU202400040

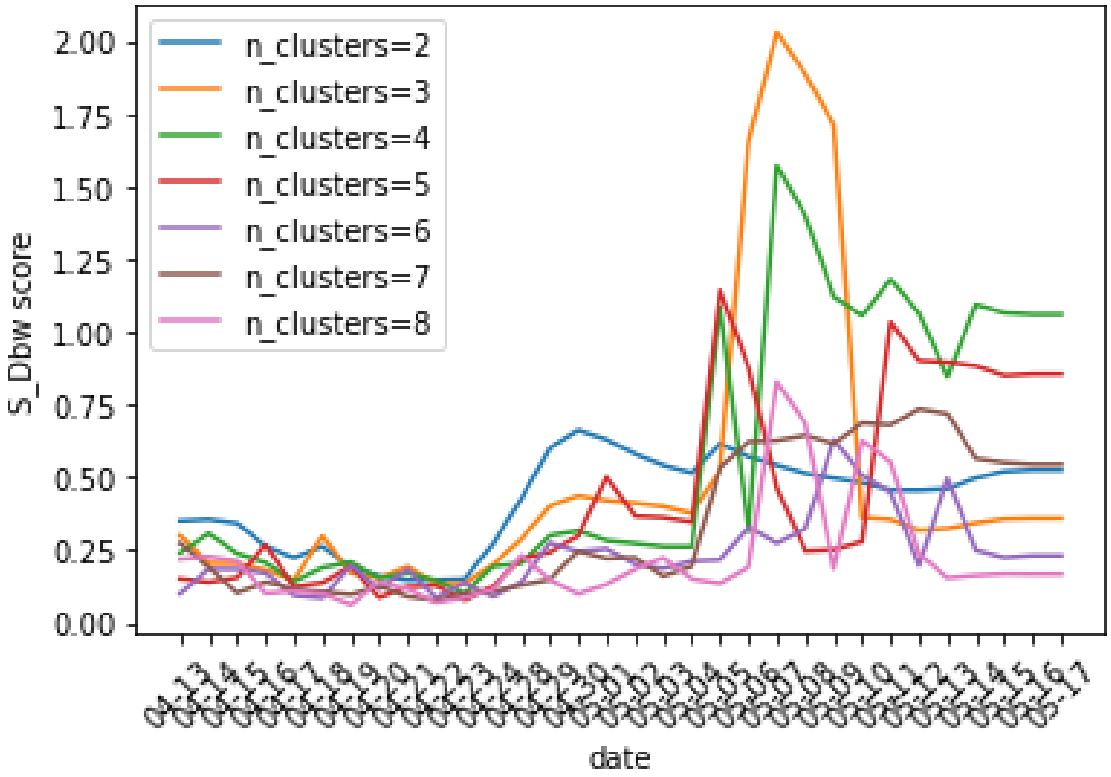


Figure 8.1: Online Clustering with K-Means - n_cluster= 2 to 8

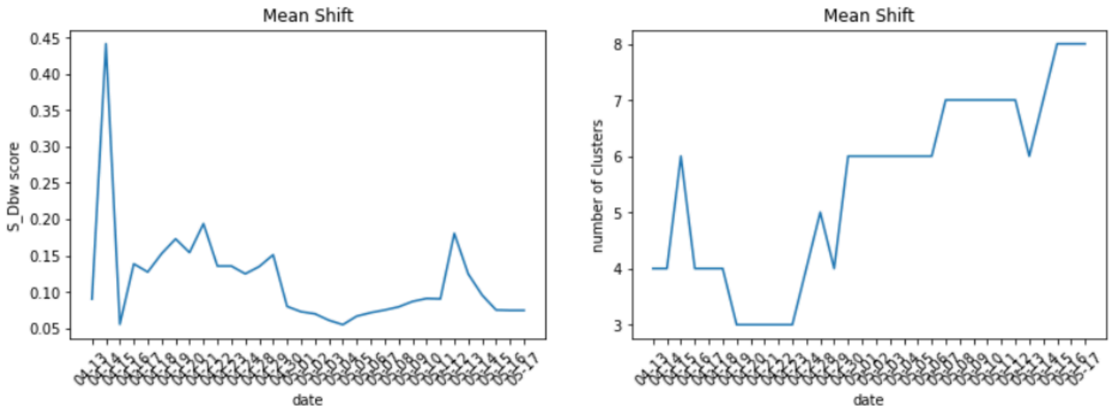


Figure 8.2: Online Clustering with Mean Shift

Chapter 9 Conclusion and Future Work

9.1 Conclusion

The goal of this research is to build the CMR model with heterogeneity for CDN server population estimation without high computation overhead in the CJS model. I discovered that the servers from the same 24-bit subnet have similar hour-count distribution. Hence, I try to use transaction counts in different hours to be the attributes for clustering.

In the beginning, I do clustering on the US-All. I divide 24 hours into 3 hour periods, ’00’ to ’07’, ’08’ to ’15’, ’16’ to ’23’. I use the transaction counts in these hour periods as the attributes to do clustering. When the number of clusters is 3, the S_dbw score is 0.3673, which is much better than the results with several clusters is 2 or 4. Next, I use principal components analysis (PCA) to reduce the data dimension from 3 to 2 and do clustering again. The S_dbw score is 0.3601, which is slightly better than the clustering result without PCA.

To explore all the possibilities of hour periods chosen, I divide 24 hours into 3, 4, and 6 periods, and slide the time window of periods. When the number of clusters is 3, I find

doi:10.6342/NTU202400040

that all the clustering results with S_Dbw score less than 0.37 are the same. Thus, I use the number of periods equal to 3 and sliding hours equal to 0 to do the clustering, which is one of the clustering results with an S_Dbw score less than 0.37.

I also do clustering to US-0 and US-1. I find that the best number of clusters is not 3 in the US-1. I use mean-shift on the US-1 data and find that the number of clusters decided by the algorithm is 8. Furthermore, I remove the fixed random seed in k-means and run k-means 10 times for each n_cluster = 2 to 8 in both US-0 and US-1.

Firstly, we deploy the k-means result with US-All and mean-shift with US-1 to the CJS model with heterogeneity. The estimation model without clustering has a lower error rate than the model with clustering when the sample time is 12 pm. In this sample hour, the error rate of the model without clustering is 0.82% while the model with k-means (n_clusters=3) is 1.92% and the model with mean-shift (n_clusters=8) is 1.87%. Although these clustering results do not improve the accuracy of the CJS model, the clustering result with k-means indicates that the servers can be divided into a few ’main’ servers and ’support’ servers which only show in a specific period. All the main servers come from 3 specific subnets, ’52.223.227’, ’52.223.228’, ’99.181.96’.

For the CJS model with multi-run of k-means in the US data, the correlation of S_Dbw score and error rate in the CJS model with heterogeneity are all negative in the US-0 and US-1. It indicates that a clustering result with a better S_Dbw score tends to have a worse error rate in the CJS model. Besides, the CJS model tends to have a worse error rate when n_cluster gets larger.

Next, I deploy the CJS model with random clustering results. In general, the CJS model with random clustering results has a lower error rate than the CJS model with k-

doi:10.6342/NTU202400040

means in both US-0 and US-1. It could be explained by k-means dividing the servers too ”well”. K-means has a higher probability of dividing the servers that do not show in the sample hour but show on baseline into the same cluster. As a result, k-means generates many clusters that wrongly estimate the number to 0 on one date.

Since S_Dbw may not be a good metric for the CJS model, I use the additional metric, Std/Avg, to evaluate the cluster error rate in each result. For the data with high Std/Avg (ex: k-means in the US data), the correlations between Std/Avg and cluster error rate are relatively high. On the other hand, for the data with low Std/Avg (ex: random clustering results in the US data), the correlations between Std/Avg and cluster error rate are relatively low.

In Chapter 6, I deploy the CJS model with heterogeneity in different periods from other regions. I discovered that the CJS model cannot fit well in the UK, the Netherlands, and German data. All the CJS models with and without clustering in these regions have error rates larger than 40%. On the other hand, the error rates of the CJS model in France are 6.12% (no clustering) and 1.47% to 40% (k-means). The reason why the CJS model cannot fit well in some regions may be explained by the high Std/Avg of the sample numbers without clustering. Std/Avg is 0.5346 in UK-0 and 0.6575 in UK-1, 1.8093 in Netherlands-0 and 0.6043 in Netherlands-1, and 0.4090 in Germany. On the contrary, Std/Avg is only 0.27 in France, 0.1109 in US-0, and 0.2858 in US-1 and period-1. If the Std/Avg is too high, the CJS model may fail to converge well.

Also, I dig into the Std/Avg of each cluster in different data. The result is similar to the US data - for the data with high Std/Avg (ex: the UK, the Netherlands, and Germany), the correlations between Std/Avg and cluster error rate are relatively high. On the contrary, for

doi:10.6342/NTU202400040

the random clustering results in France, all the Std/Avg are less than 0.4, and its correlation with the cluster error rate is only 0.1642, which is relatively low.

In Chapter 7, I discovered that the correlation between the number of clusters and the computation time of the CJS model with k-means are all above 0.9472. In this experiment, the relationship between the number of clusters and CJS computation time is close to a linear model. However, when the n_cluster is too high, the CJS may not converge since the capture histories in some clusters are too small to have enough data to derive for all parameters. In this research, all the CJS computation times are less than 1 minute. Nevertheless, the CJS computation time would increase as the period length gets longer. One could decide on an upper limit for the number of clusters based on the time constraints of the CJS model.

In chapter 8, I try to do clustering online. I find that the best number of clusters is not fixed in the US data. The best number of clusters ranges from 3 to 8 in the k-means and mean-shift results through the US data in 2021.

9.2 Future Work

In my research, I discovered that k-means cannot improve the error rate of the CJS model. Many clusters from k-means results wrongly estimate the number to 0 on one date. Std/Avg in clusters can help avoid such issues. For the data with high Std/Avg, the correlation of Std/Avg and cluster error rate is high. Besides, maybe the lower limit of cluster sizes could avoid the high Std/Avg in clusters.

For the CJS model in other regions, the CJS model cannot fit well in the UK, the Netherlands, and Germany. This may be caused by the high Std/Avg in the data. In these

doi:10.6342/NTU202400040

data, the sample number often suddenly decreases significantly into a small value. This may be caused by the measurement error or the mechanism of Twitch’s CDN. Besides, the CJS model with some k-means results do not converge. I think when the number of servers is low on some dates, it is wise to avoid clustering into too many clusters.

doi:10.6342/NTU202400040

References

[1] L. Buitinck, G. Louppe, M. Blondel, F. Pedregosa, A. Mueller, O. Grisel, V. Niculae, P. Prettenhofer, A. Gramfort, J. Grobler, R. Layton, J. VanderPlas, A. Joly, B. Holt, and G. Varoquaux. API design for machine learning software: experiences from the scikit-learn project. In ECML PKDD Workshop: Languages for Data Mining and Machine Learning, pages 108–122, 2013.

[2] K. P. Burnham. Design and analysis methods for fish survival experiments based on release-recapture / Kenneth P. Burnham ... [et al.]. American Fisheries Society Bethesda, Md, 1987.

[3] Y. Cheng. Mean shift, mode seeking, and clustering. IEEE Transactions on Pattern Analysis and Machine Intelligence, 17(8):790–799, 1995.

[4] R. M. CORMACK. Estimates of survival from the sighting of marked animals. Biometrika, 51(3-4):429–438, 12 1964.

[5] A. W. F. Edwards. Likelihood. Cambridge University Press, 1972.

[6] D. et al. Internet scale user-generated live video streaming: The twitch case. In Proceedings of PAM, pages 60–71, New York, NY, 02 2017. Springer.

[7] M. Halkidi and M. Vazirgiannis. Clustering validity assessment: finding the optimal

partitioning of a data set. In Proceedings 2001 IEEE International Conference on Data Mining, pages 187–194, 2001.

[8] G. M. Jolly. Explicit estimates from capture-recapture data with both death and immigration-stochastic model. Biometrika, 52(1/2):225–247, 1965.

[9] C. Krebs. Ecological Methodology, 3rd ed. 2014. https://www.zoology.ubc.ca/~krebs/books.html(visited 2021-05-25).

[10] F. Lincoln. Calculating Waterfowl Abundance on the Basis of Banding Returns. Circular (United States. Department of Agriculture). U.S. Department of Agriculture, 1930.

[11] Y. Liu, Z. Li, H. Xiong, X. Gao, and J. Wu. Understanding of internal clustering validation measures. In 2010 IEEE international conference on data mining, pages 911–916. IEEE, 2010.

[12] A. M. Mood, F. A. Graybill, and D. C. Boes. Introduction to the theory of statistics. McGraw-Hill New York, 3rd ed. edition, 1974.

[13] N. I. of Standards, Technology, and I. SEMATECH. NIST/SEMATECH e-Handbook of Statistical Methods.

[14] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. Scikit-learn: Machine learning in Python. Journal of Machine Learning Research, 12:2825–2830, 2011.

[15] C. G. J. PETERSEN. The yearly immigration of young plaice into the limfjord from the german sea, ect. Report of the Danish Biological Station for 1895, 6:1–48, 1896.

doi:10.6342/NTU202400040

[16] S. Pledger, K. H. Pollock, and J. L. Norris. Open capture-recapture models with heterogeneity: I. cormack-jolly-seber model. Biometrics, 59(4):786–794, 2003.

[17] B. L. S. Prakasa Rao. Maximum likelihood estimation for markov processes. Annals of the Institute of Statistical Mathematics, 24(1):333–345, Dec 1972.

[18] G. A. F. Seber. A note on the multiple-recapture census. Biometrika, 52(1-2):249–260, 06 1965.

[19] TwitchTracker. Twitch statistics and charts, 2021.

[20] C. Wang”. ”discovering twitch’s video delivery infrastructure utilizing cloudservices and vpns”.

[21] H. C. Y.-T. L. C. H. G.-T. T. P. H. Wei-Shiang Wung, Caleb Wang. Poster: Twitch’ s cdn as an open-population ecosystem, 08 2020.

[22] W.-S. Wung”. ”estimation on server population with mle-based cmr (capture-mark-recapture) model”.