Master Thesis
Computer Science
Thesis no: MCS-20YY-NN
06 2019

Clustering of Driver Data based on Driving Patterns

Amit Kabra

Dept. Computer Science & Engineering
Blekinge Institute of Technology
SE–371 79 Karlskrona, Sweden

This thesis is submitted to the Department of Computer Science & Engineering at Blekinge Institute of Technology in partial fulfillment of the requirements for the degree of Master of Science in Computer Science. The thesis is equivalent to 20 weeks of full-time studies.

Contact Information:
Author(s):
Amit Kabra
E-mail: amitkabra59@gmail.com

University advisor:
Dr. Hüseyin Kusetogullari
Dept. Computer Science & Engineering

Dept. Computer Science & Engineering
Blekinge Institute of Technology
SE–371 79 Karlskrona, Sweden

Internet : www.bth.se/didd
Phone : +46 455 38 50 00
Fax : +46 455 38 50 57

Abstract

Data analysis methods are important to analyze the ever-growing enormous quantity of the high dimensional data. Cluster analysis separates or partitions the data into disjoint groups such that data in the same group are similar while data between groups are dissimilar. The focus of this thesis study is to identify natural groups or clusters of drivers using the data which is based on driving style. In finding such a group of drivers, evaluation of the combinations of dimensionality reduction and clustering algorithms is done. The dimensionality reduction algorithms used in this thesis are Principal Component Analysis (PCA) and t-distributed stochastic neighbour embedding (t-SNE). The clustering algorithms such as K-means Clustering and Hierarchical Clustering are selected after performing Literature Review. In this thesis, the evaluation of PCA with K-means, PCA with Hierarchical Clustering, t-SNE with K-means and t-SNE with Hierarchical Clustering is done. The evaluation was done on the Volvo Cars' drivers dataset based on their driving styles. The dataset is normalized first and Markov Chain of driving styles is calculated. This Markov Chain dataset is of very high dimensions and hence dimensionality reduction algorithms are applied to reduce the dimensions. The reduced dimensions dataset is used as an input to selected clustering algorithms. The combinations of algorithms are evaluated using performance metrics like Silhouette Coefficient, Calinski-Harabasz Index and Davies-Bouldin Index. Based on experiment and analysis, the combination of t-SNE and K-means algorithms is found to be the best in comparison to other combinations of algorithms in terms of all performance metrics and is chosen to cluster the drivers based on their driving styles.

Keywords: Clustering, Driving Patterns, Markov Chain, Cars, Machine Learning

Acknowledgement

Dedicated to the bright memory of my grandfather, Dr. Sriram Kabra, who always believed in me. You are gone but your belief in me has made this journey possible. It is a genuine pleasure to express my deep sense of gratitude and sincerity to my university supervisor Dr. Hüseyin Kusetogullari and my external supervisors Peter Ålleving and Emil Staf whose expertise, patience and support made it possible for me to work in an area of great interest to me. I would also like to thanks Lars Unéus for giving this opportunity to me. Finally, I would also like thank my parents and friends who always believed in me and pushed me to my best.

List of Figures

List of Tables

Contents

Chapter 1

Introduction

In the automotive industry, vehicles are becoming more and more software-intensive complex systems where most of the innovation is based on software and electronics. Modern vehicles have more than 100 Electronic Control Units (ECUs) and these ECUs are mainly small computers which are continuously executing gigabytes of software. [1]

Volvo Cars is a major international company that manufactures and markets sport utility vehicles, station wagons, sedans and compact executive sedans[2]. Volvo always strives to improve the quality of experience of the drivers, related to both driving and servicing. The proper utilization of diagnostic data plays a keen role in identifying quality issues. Anonymised data is collected from the vehicle and used for quality improvement when needed.

Data analysis methods are important to analyze the ever-growing enormous quantity of the high dimensional data [22]. Clustering analysis [61] separates or partitions the data points in the data into disjoint group or clusters such that data points in the same group are similar to each other and data points present in other groups are dissimilar [22].

The road traffic flow plays a big role and has an impact on the driving style. The vehicle may go from idle speed to the next start speed and again to idle speed depending upon the traffic. Vehicle may perform start and stop action several times between start to the park of the vehicle. Hence driving cycle of the vehicle can be seen or viewed as a combination of micro-trips [6]. Driving condition may vary depending on the geographical location where the vehicle is driven. A lot of research has been done on driving cycle but it mainly focused on fuel and battery consumption field but not to identify the driving style of the driver[3][4][5].

1.1 Problem Statement

On-Board Diagnostics (OBD) refers to an automotive term which relates to the vehicle's reporting and self-diagnostic capability. The demand of OBD in the European Union (EU) started in 2001 [63]. The OBD in a vehicle uses different functions at ECUs involved for various groups of emission-relevant components. To avoid overburdening of ECUs, the monitoring is not done constantly, but only at specific points of time in a driving cycle. To make sure a fault in OBD is not caused by rare monitoring only, the EU demands monitoring for monitoring itself. [63]

This monitoring system is already implemented in the vehicles. It is important to understand and analyze the monitoring system's behaviour to predict the failure. The first step towards this prediction can be to make effective use of diagnostic data. An effective way of utilizing the data based on driving styles can be to group the drivers based on their driving styles. Understanding the driving behaviour of the driver could also be helpful to understand the OBD monitoring system's behaviour. Driving style is chosen because it can affect a lot of system in the car and can be used to understand quality issues, eg., repairs. The primary focus of the thesis is to group drivers based on their driving styles. Understanding the OBD monitoring system's behaviour and predicting failure is out of the scope of this thesis and can be the future work.

1.2 Aim and Objectives

The aim of this thesis study is to identify natural groups of drivers using the data based on driving style. The term 'natural group' means that groups are based on the natural driving of the drivers. In order to achieve the aim of this thesis, the following objectives are formulated:

1. Identify suitable clustering algorithms by conducting a literature review
2. Reduce the dimensions of the high dimensional dataset in the experiment
3. Perform clustering on the reduced dataset to identify natural groups of drivers
4. Evaluate the clusters obtained using clustering algorithms to find the best clustering algorithm for the given dataset

1.3 Research Questions

- **RQ1**: Which clustering algorithms should be used for grouping drivers based on driving style for the given dataset?

Motivation: The motive behind choosing the RQ1 is to gain knowledge about clustering. It is also important to know which clustering algorithms can be applied for user data analysis since we cannot apply all the algorithms to our data due to time constraint.

- **RQ2**: Which clustering algorithm gives the best results for the used dataset?

  Motivation: The motive behind choosing RQ2 is to evaluate the best clustering algorithm among the chosen from the literature review, based on evaluation of performance metrics and expert's evaluation.

The most significant application for Machine Learning is data mining [66]. People often prone to make mistakes during analyses or while trying to establishing the relationship between multiple features. Hence, finding solutions to certain problems can be difficult. On the other hand, Machine Learning can be often applied to these problems, improving the designs of a machine and efficiency of the system [66]. This is the motivation for choosing Machine Learning to solve this problem.

The thesis is structured as follows: In the first chapter, the problem statement, aim and objectives are introduced together with the research questions and motivation. The second chapter introduces the reader to the theoretical background. The third chapter introduces the reader to some related work to this thesis. In the fourth chapter, the research methods adopted to answer the research questions are introduced along with the motivation for choosing these methods. In the fifth chapter, the results obtained from the experiment are visualized. The sixth chapter deals with the evaluation of the results obtained in the fifth chapter based on performance metrics. The clusters obtained by the best algorithm is further analyzed using the metadata. The seventh chapter discusses the answers to research questions, the contribution of the author and the threats to validity for the thesis. The eighth chapter presents the conclusion and possible future work.

Chapter 2

Background

2.1 Introduction to Clustering

Machine Learning (ML) is a field of Artificial Intelligence(commonly known as AI) that enables the machine to learn on its own from data, using statistical techniques. Machine Learning can be categorized into three types: Unsupervised Machine Learning, Supervised Machine Learning and Reinforcement Learning.

2.1.1 Clustering

According to [11], clustering is defined as the grouping of the data objects in a data set in such a way that the objects in the same cluster are similar to each other but are different from the objects outside the cluster. Clustering analysis is an unsupervised ML approach where the task is to group set of objects so that object in same group or cluster is more similar to each other than the ones in other groups [7]. It is the main task of exploratory data mining [7]. Clustering can be performed for various reasons but it is mainly performed for these two reasons: Data Interpretation and Data Compression [9]. There are many applications of clustering, but people mainly use it to detect the patterns, data mining, classification, grouping the objects based on similarity or dissimilarity criteria, knowledge discovery and so on [8]. Besides of this, clustering has also been used in different fields such as image processing [67, 68, 69]. Model-Based Clustering, Partitioning Clustering, Hierarchical Clustering, Grid-Based Clustering, Density-Based Clustering, etc. are the major clustering techniques [10].

2.2 Introduction to Markov Chain Model

Markov Chain is a stochastic or random process that satisfies Markov property [12]. Markov property is characterized as Memorylessness property which states that the probability distribution of future states depends only on the present state and not the sequence of the events preceding it [13]. It is named after Andrey Markov, a Russian Mathematician. It is used to compute the probability of occurring of events, by viewing the events as states transitioning into other states,

or the same state as before. [13]

Generally, the term Markov Chain is used to refer discrete-time Markov chain and term Markov Process is used to referring continuous-time Markov chain [12]. The **PageRank** algorithm proposed by Google for their internet search engine is based on the Markov process. Markov Chains are widely used as **statistical models** of real-world processes. For example, queues of drivers at the airport, currency exchange rates, cruise control system in vehicles [64]. Markov chains are used in various other domains ranging from text generation to Modeling of finance.

Consider a random variables sequence X_1, X_2..... with Markov property. The probability of transition to the next state depends only on the current or present state and not on the states proceeding it. The Markov property states that

$$P(X_{t+1} = x_{t+1} | X_1 = x_1, X_2 = x_2, \ldots, X_t = x_t) = P(X_{t+1} = x_{t+1} | X_t = x_t)$$

The above equation says that the probability that at time $t + 1$ state is x_{t+1} given at time 1 state is x_1, at time 2 state is x_2 and so on to at time t state is x_t is equal to probability that at time $t + 1$ state is x_{t+1} given at time t state is x_t.

Let's take weather as an example: If we randomly pick probabilities, predictions about the weather can be following: If the day is rainy, the probability of the next day to be sunny is 40%, if the day is sunny, there is 30% probability that next day is rainy. If the day is sunny, there is a 70% chance that next day is sunny and if day is rainy, there is 60% chance that next day is rainy day. This weather example is summarized in a transition diagram figure 2.1 which describes all the probabilities:

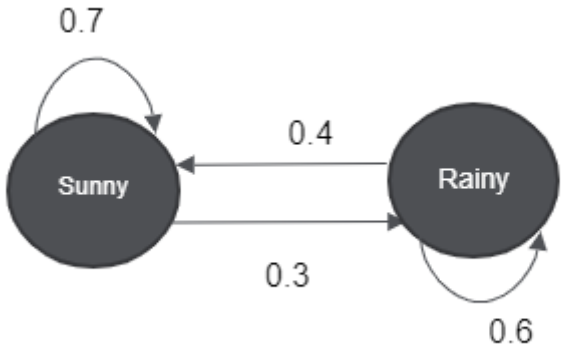

Figure 2.1: Transition diagram for weather prediction

2.3 Introduction to Dimensionality Reduction

In many domains, high dimensional data visualization has been a problem. In clustering, high dimensional data poses several challenges for the clustering algorithms which needs specialized solutions are discussed in [74]. Dimensionality reduction is an essential task for information processing problems like searching over web datasets, classifying documents set [73]. Clustering of high dimensional data is important in many applications like gene expression analysis, metabolic screening, text documents collection clustering, customer recommendation systems [75]. When data has a lot of features, it is difficult to understand the data and examine the relationship between features. This problem not only makes the process of Exploratory Data Analysis difficult but also effects the performance of Machine Learning models and may result in overfitting [72]. Overfitting can be limited by restricting the number of dimensions but it is not the only way. Regularization, training more data are some other ways that can also be used to reduce overfitting.

Dimensionality reduction is the process of obtaining a set of latent variables by reducing the number of considered random variables. Traditional dimensionality reduction and state-of-art methods can be classified generally into Feature Selection and Feature Extraction approaches [24][72][74].

Advantages of dimensionality reduction:

1. Visualization becomes much easier when high-dimensional data is reduced to low dimensions (2D or 3D)
2. Helps to avoid effects of the curse of dimensionality
3. Reduces time and required storage space [24]
4. Removes multi-collinearity which helps to improve parameters interpretation of the model [24]
5. Limits the overfitting [72]

2.3.1 Curse of Dimensionality

As the number of features or dimensions increases, the quantity of data needed to generalize accurately also grows exponentially. To overcome this curse of dimensionality, there are several techniques, one of them is to project the high dimensional data to low dimensional space. For example, when a light source is placed in front of a three-dimensional object, a two-dimensional shadow is projected in against the wall. Dimensionality reduction algorithms can help to overcome the curse of dimensionality. The main benefits of reducing the dimensionality of high

dimensional data are less dimensional redundancy, the computational workload becomes lighter. Dimensionality reduction is performed prior to clustering to avoid the curse of dimensionality effects for high-dimensional data.

2.3.2 Crowding Problem

The crowding problem arises from the curse of dimensionality. The surface of a sphere in high dimensional space grows faster with its radius when compared to a sphere present in low dimensional space. The high dimensional space can have several data points present at a medium distance from a certain point. When these data points are mapped to low dimensional space, the medium data points will try to gather at a medium distance. But the main problem is that at a medium distance the room is very less low dimensional space. Hence, the points will get squashed in the lower dimensional space, causing crowding. To summarize, when data points are mapped from high dimensional space to low dimensional space, the data points tends to get crowded in a low dimensional space because of the curse of dimensionality causing crowding problem. [25]

2.3.3 Dimensionality Reduction Methods

Principal Component Analysis

Principal Component Analysis (PCA) is a linear feature extraction technique [23]. PCA is a multivariate statistical method that transforms a number of correlated variables into a smaller set of uncorrelated variables known as principal components. PCA is used to reduce the dimensions of a dataset while performing unsupervised machine learning. The goal of this technique is to extract maximum information of original variables from the dataset and represent it as a set of new uncorrelated orthogonal variables known as principal components. [23]

PCA looks for a linear combination of variables in such a way that maximum variance is summarized from the variables. This linear combination is called the first principal component and it explains the maximum variance of the data. After finding the first principal component, PCA will look for a second linear combination that would explain the maximum proportion of leftover variance and so on. The second principal component should be orthogonal to the first. Practically, a linear combination that reflects more than 80% of the information of the original variable can be chosen. [6]

In a simple way, it can be said that PCA combines the input features in a specific way retaining the most valuable parts/information of all features while dropping the least important features in such a way that PCA components are

linearly independent.

PCA has three steps from a high-level point of view [75]:

- Covariance matrix of data is computed

  To properly measure the variance, the data is normalized first to have zero mean and unit variance so that each feature will be equally weighted. Variance calculates the variation of a single random variable. The covariance calculates how much two random variables are correlated to each other or vary together. If the covariance is positive then when one variable decreases the other will also decrease. If the covariance is negative then when one variable will decrease, others will increase. The covariance matrix is an array which will specify the covariance between the two variables(feature) based on position in the matrix. The formula is given by:

  $$\sum = \frac{1}{n-1}((X - \overline{x})^T (X - \overline{x}))$$

  where n is the number of data points, $\overline{x}$ is a mean values vector for each feature of X. Each of the features for each data points is multiplied together by multiplying the transpose matrix by the original matrix.

- Eigenvalues and vectors of this covariance matrix are computed

  Eigenvectors are the principal components representing the vector directions of the newly obtained feature space. Eigenvalues represent the magnitude of eigenvectors. Since our matrix is a covariance matrix, the eigenvalues will quantify the variance contributed by each vector.

- The eigenvalues and eigenvectors are used to select the most important feature vectors and after that data is transformed on those vectors to get reduced dimensions

  Eigenvector with corresponding high magnitude of eigenvalue means that data has a high variance value along that vector in feature space. If changing the feature vector value does not affect the data greatly, then it can be concluded that the feature is not very important and can be removed if necessary. The eigenvectors and eigenvalues of the covariance matrix are computed and eigenvectors are sorted in descending order based on their respective eigenvalues. Now that there is a sorted list of eigenvectors based on their importance, it is possible to select only the most important feature vectors and remove the remaining feature vectors. The most important feature vectors can be selected by looking at *explained variance percentage* of the feature vectors. This percentage shows how much variance (information) can be attributed to each component out of 100%. The final step is to project data onto the feature vectors we decided to keep.

It is highly recommended to normalize the data first before performing PCA. Otherwise, the variables with large value and variance will dominate the first component when they are not supposed to.

t-Distributed Stochastic Neighbor Embedding

t-Distributed Stochastic Neighbor Embedding (t-SNE) is a popular method introduced by Van der Maaten and Hinton [25] for dimensionality reduction and exploring high-dimensional data. t-SNE is a non-linear dimensionality reduction technique with the ability to create a 2D or 3D map from data with even thousands of dimensions. [25]

The main question is why t-SNE is needed when there is PCA already? One of the disadvantages or drawback of PCA is that it only captures linear projection. t-SNE creates a low dimensional mapping using the local relationship between data points. This is how it captures the non-linear structure. This makes t-SNE suitable for most sorts of datasets. Well, some might also ask there are other methods like Kernal PCA, Local Linear Embeddings, etc which also uses the local structure. t-SNE works quite well in practise [76][25], this could be due to several reasons like it handles crowding problem well, it uses the concept of "stochastic neighbors" which allows t-SNE to take both local and global structure into account. [76][25]

t-SNE has following steps from a high-level point of view [76][25]:

- A probability distribution is created which shows the relationship between several neighbouring points in the high-dimensional space. Gaussian distribution is used which is a natural choice since we are dealing with probability measure of similarity between data points. An important hyperparameter for t-SNE is *perplexity* which is the number of neighbours at any point. t-SNE generally works quite well for any value between five and fifty. Large perplexity means that global structures are taken into account, where are small means that local structures are the main focus of embedding.

- In t-SNE, t comes from t-distribution, stochastic (S) and neighbor (N) means that probability distribution across the neighbouring points is being used. When recreating lower dimension mapping there is a need to create a similar distribution. t-SNE maps high dimensional points to low dimensions in such a way that similar points in high dimensional are similar even after mapping to low dimensions and vice-versa.

The detailed t-SNE algorithm can be found in the orignal paper [25].

Disadvantages[76]

- t-SNE is non-deterministic i.e., you can run it several times and get a different result each time
- Assumes that the local structure of manifold is linear. This poses a problem when there are complex manifolds

2.4 Clustering Methods

2.4.1 K-means Clustering

K-means is a popular clustering algorithm for executing unsupervised learning tasks [22]. The goal of this algorithm is to find natural group or cluster of objects in such a way that objects in same clusters are similar to each other and clusters in different groups are different in terms of their properties.

Algorithm

The inputs to the algorithm are the number of clusters 'K' and the dataset. We need to choose the K value. K-means begins with initial estimates for K centroids, which could be randomly selected or randomly generated from the dataset. After this, the algorithm iterates between the following two steps [78]:

Data Assignment
Each cluster is defined by a centroid. In the data assignment step, based on squared Euclidean distance, each data point or sample is assigned to its nearest centroid. Assuming c_i is a collection of centroids in a set of centroids C, then each data point x can be assigned to a group or cluster based on the following:

$$\underset{c_i \in C}{argmin}\ dist(c_i, x)^2$$

where dist(.) is the standard Euclidean distance. Let S_i be the set of data point for each i^{th} cluster centroid

Centroid update
In the centroid update step, centroids are recomputed. The recomputation is done by taking the mean of all the data points present/assigned to that centroid's cluster.

$$c_i = \frac{1}{|S_i|} \sum_{x_i \in S_i} x_i$$

The K-means algorithm iterates between the above two steps until a stopping criterion is met. The criterion is that no data points change their cluster mem-

bership, the maximum number of iterations are reached, or the sum of distances is minimized.

K-means algorithm always converges to a result, i.e., the algorithm will stop either when a number of iterations are reached or when no data points will change cluster membership. Hence, the result is guaranteed. But the result produced may be a local optimum i.e., might not be the best possible result. Assessing the algorithm for more than one run by varying different parameters may give a better result.

Advantages

- Easy to implement
- K-means is computationally faster than hierarchical clustering for a large number of variables (if k value is small)
- K-means produce tighter clusters than hierarchical clustering
- K-means algorithm is known to have $O(n^2)$ time complexity, where n is the number of data points [86]

Disadvantages

- Initial seeds will have an impact on final results
- Sensitive to scaling
- Difficult to predict k value i.e., number of clusters

2.4.2 Hierarchical Clustering

Hierarchical Clustering is a tree which represents a set of nested clusters [27][21].

Measure of Dissimilarity
A cluster can be split into multiple clusters or multiple clusters can be merged into one. The decision of where a cluster is split or how clusters are merged depends upon a measure called measure of dissimilarity. The dissimilarity can be measured by distance between pair of data points or observation, like Euclidean distance.

Euclidean Distance
Euclidean distance is the straight line distance measure between two points in Euclidean space [70, 71].

Suppose, there are two point p(x_1, y_1) and q(x_2, y_2) in a plane, then euclidean distance between p and q can be calculated as:

$$\text{d(p,q)} = \text{d(q,p)} = \sqrt{(x_1 - x_2)^2 + (y_1 - y_2)^2}$$

In K dimensions, the euclidean distance between p and q is

$$\sqrt{\sum_{i=1}^{K}(q_i - p_i)^2}$$

where p_i (or q_i) is coordinate of p (or q) in dimension i.

Hierarchical Clustering is divided into two namely, Agglomerative and Divisive. For this thesis work, Agglomerative Hierarchical Clustering is chosen because it has efficient linkage methods implemented (e.g., single linkage and complete linkage) which are shown in the next section [28].

Hierarchical Agglomerative Clustering (HAC)

In this bottom-up approach, we initially consider each data point as an individual cluster. At each iteration, similar clusters are merged with other others until they merged (agglomerative) together as a single cluster. Hence, the name hierarchical agglomerative clustering [26, 27]. In this method, like K-means, there is a need to choose the initial number of clusters.

The basic agglomerative algorithm is straight forward.

- Proximity matrix is computed
- Consider each data point as a cluster
- Repeat: The two closest clusters are merged and the proximity matrix is updated
- Until: Only a single cluster is remaining

The metric used for merge strategy is determined by linkage criteria [29][79]:

1. Ward: The sum of squared differences is minimized within all clusters
2. Maximum or complete linkage: Maximum distance between data points/observations of pairs of clusters is minimized
3. Average linkage: Average of distance between all data points/observations of pairs of clusters is minimized
4. Single linkage: Distance between the nearest data points/observations of pairs of clusters is minimized

In general, HAC has space complexity of $O(n^2)$ and time complexity of $O(n^3)$, where n is the number of data points. This makes HAC work slower for medium-sized datasets. However, efficient agglomerative methods like complete linkage and single linkage clustering have time complexity if $O(n^2)$ [28].

Advantages

- Easy to implement
- Hierarchical structure returned by hierarchical clustering is more informative than unstructured clusters set returned by K means

Disadvantages

- Once, split or merge is done, cannot be undone [27]
- Not suitable for large datasets since the time complexity is $O(n^3)$ [28]
- Does not work with missing data
- Initial seeds will have an impact on final results
- Sensitive to good initialization [27]
- Difficult to compute distance matrix for many data types
- Coincident clusters might result [28]

Chapter 3

Related Work

Clustering is studied in many fields including databases[34], statistics[35], machine learning[36], visualization [36].

Cerebellum Model Articulation Controller (CMAC)[54][40] introduced by Albus is an early example to understand driving behaviour since each and every action is controlled by Cerebellum [40]. Brake and gas pedal pressure was their main focus. CMAC's performance can be analyzed by understanding the potential of feature extraction for identifying driver's driving behaviour[40]. In 2009, the Gaussian Mixture Model (GMM) [40] was used for the identification of driving features that effectively helped to efficiently profile each driver. Later, the features which were extracted from brake pedal pressure and accelerator were used as inputs to the fuzzy neural network to driver's identification [55].

Yi Lu Murphey in 2009, analyzed drivings state by using jerk variations, such as acceleration and deceleration. Yi Lu Murphey also proposed an algorithm to classify driving style by utilizing statistical information from the jerk profile and level of traffic congestion prediction and roadway type [56].

CarSafe[57] is an android application which is developed for driver's safety. It makes use of cameras and smart embedded sensors on smartphones and stores all the information [57]. A prediction system [58] called Mobile Crash was developed which classifies the driver as unfit, fit or partially fit based on the input of different attributes of driver profile [58].

According to research, people have different speed graphs in different situations and these speed graphs can be utilized to detect the driver behaviour and their other psycho-physiological states [59]. Research on facial recognition analysis for unsafe driving behaviour prediction analyzed twenty-two facial features. A dataset to build computational models was created by using the driving simulator data and collected videos. This feature was proved to be beneficial to predict an accident three to four seconds prior [60].

Zhang, Guo and Huang in [6] investigate the driving cycle for electric cars. For Data collection management and transition, a GPRS communication-based system is used. The collected data is divided into micro-trips and features are selected. Principal Component Analysis is used to get a representation of the drive cycle for Beijing roads. The k-means clustering algorithm is applied to the data obtained from the Principal Component Analysis. The result from research shows that overall traffic conditions in Beijing can be represented by fitted driving cycle.

Shi, Zhou and Qiu in [47] investigate the driving data collected from real driving tests. Twelve characteristics features that represent the driving feature are chosen to perform the experiment. At first, the dimensions of data is reduced using Principal Component Analysis to obtain three principal components. The scores of the three components in all kinetic segments are classified using a combination of SOFM neural network algorithm and K-means clustering [47]. From each category, proper segments are selected based on their duration percentage. The results from the research show that the combination of clustering technique has high accuracy when compared to K-means clustering in drive cycle fitting and reflects real urban traffic scenarios.

Kalsoom and Halim in [40] used K-means and Hierarchical clustering algorithms to classify based on driver's driving feature like the number of brakes, average gear, the ratio of indicators to turns, maximum speed and gear, average speed, etc. The aim is to group the slow, normal and fast driving styles. The result of the experiment showed that the K-means clustering algorithm outperformed the hierarchical clustering algorithm for the recorded data.

Chapter 4

Methodology

4.1 Method

The research methods chosen to answer the research questions are:

1. Literature Review
2. Experiment

4.1.1 Literature Review

The primary basis of research in nearly every field are Literature Reviews [49]. The literature review has been conducted to answer the first research question i.e., RQ1. The motive behind conducting the literature review is to gain knowledge about clustering, understand different clustering algorithms and chose suitable clustering algorithm for our analyzing our dataset. A Systematic Literature Review cannot be opt for this research since the results gathered from the review are not used the final result. A simple literature review has been conducted to gain knowledge about clustering algorithms. The results and the knowledge gained during the literature review has been used in the experimental method. Summary of the papers reviewed during the literature review has also been documented. Though the literature review was conducted only to gain knowledge about clustering methods, it has also been useful to understand about dimensionality reduction.

The steps performed in order to search the relevant sources are shown in the figure 4.1:

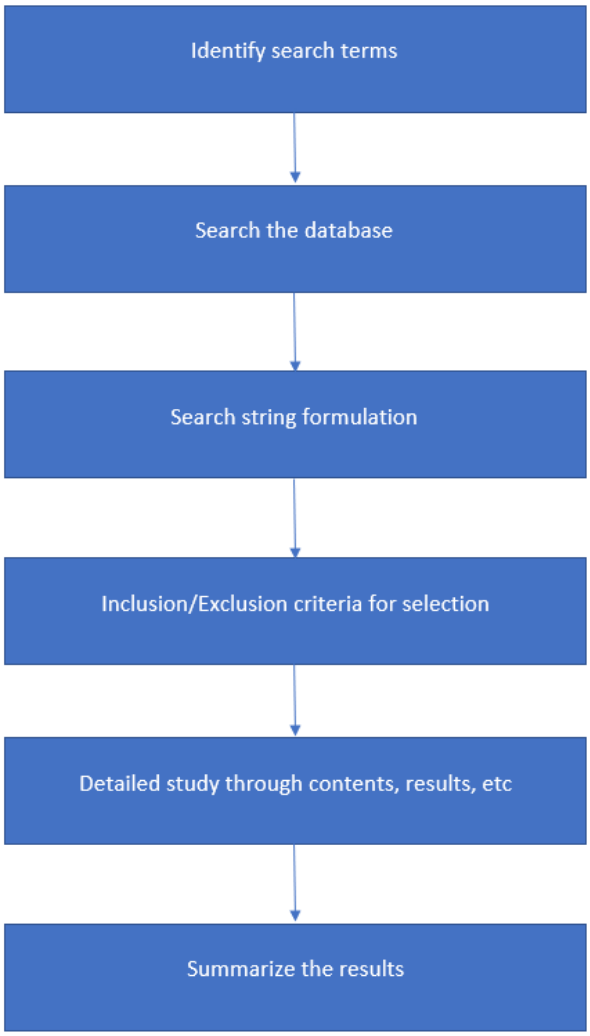

Figure 4.1: Steps performed during literature review

Following steps have been performed during literature review:

1. Before formulating the search string, the following keywords were identified. "Machine Learning", "Unsupervised Machine Learning", "Exploratory", "Clustering", "Driving Style", "Markov Chain clustering", "Clustering in Automotive", "Driver classification", "User behavior", "Driving Cycle".

2. Based on the above-listed keywords, primary keywords were selected to formulate the search string.

3. The following search strings were formulated and used to perform a search in different digital libraries:

Search string 1: "Clustering analysis"
Search string 2: "User behaviour analysis"
Search string 3: "Driving Cycle analysis"
Search string 4: "Exploratory Analysis"
Search string 5: "'Markov Chain' AND 'Clustering' "
Search string 6: "'Automotive Engineering' AND 'Clustering'"
Search string 7: "Clustering Methods"

4. Inclusion and exclusion criteria are implemented after obtaining the articles, conference papers, journals to limit the results

   Inclusion Criteria

   - Papers published over the past 25 years have been selected
   - Title and abstract of the papers should match with the problem domain
   - Article should be available in English
   - Papers related to Supervised Machine Learning that also includes Clustering
   - Combination of clustering algorithms

   Exclusion Criteria

   - Non-English language
   - Full text is unavailable
   - Papers only related to Supervised Machine Learning

5. The various clustering algorithms which can be adopted for our study are listed in the experiment section. The knowledge about clustering and the chosen clustering algorithms is documented in the chapter Background under the Clustering section and Clustering Methods section respectively.

The following is the summary of the papers reviewed:

Maia, Almeida, and Almeida in [32] used clustering to identify and group the users based on their behaviour on online social networks like YouTube and also derived a useful set of profiles based on the analysis. This identification of users of different classes can use for improvements, for example, advertisement recommendations for a targeted class of users on online social networks[32].

An unsupervised system is built by Wang et al. in [33] which used clustering to group similar users and identify user behaviour from clickstream data. The system is also able to identify new, previously unknown behaviours and their visualization tools can be used to interpret any identified behaviours.

Portnoy, Eskin, and Stolfo, in [37] elaborate on the use of Clustering to detect new intrusions with unknown signature. The paper [37] shows how clustering is applied on data to help them find the intrusion buried within the data, and these intrusion signatures are used in other applications like network security [37].

Collins and Singer in [38] used clustering to solve the problem of entity classication. Schuit et al. in [39] elaborate the use of clustering to identify common lifestyle risk factors in the general adult population and found that about 20% of adults had at least three lifestyle risk factors.

Garg and Rani in [41] used clustering to analyze user behaviour on social networking sites. The authors used k-means clustering to analyze individuals behaviour and visualize based on Twitter data.

Nijhawan, Srivastava and Shukla in [42] proposed a k-means clustering algorithm to map land cover accurately. The authors have tried with several combinations of parameters and chose the one which gave the best classification results. The algorithm gave 93.5% classification accuracy.

Lu et al. in [43] used clustering to separate raw meal composition's production conditions. The authors used the principal component analysis method to reduce dimensions and obtain principal components of clustering variables. K-means was used to conduct clustering on features which were obtained after dimensionality reduction. According to the author the results obtained showed that the clustering method was feasible for their problem.

Shi et al. in [44] used clustering to analyze Nantong University campus network user's behaviour. The authors used a k-means clustering algorithm for data mining analysis of network usage. After analyzing experimental results, they found that the time of network internet usage by the user has positive relevance

with students failure rate and a negative relevance with obtaining a scholarship. It also helped to understand students behaviour and can be useful to guide them to add good habits. Moreover, the analysis helped to improve network bandwidth, application efficiency and performance.

Park and Park in [45] used a clustering algorithm to analyze Photovoltaic power data. The clustering method of photovoltaic data could only analyze the meaning of each cluster and not the behaviour. This problem made it difficult to find similarity between clusters. The authors used Hierarchical clustering algorithm to analyze the similarity of data between clusters to find the relationship between clusters. Results show that clustering helped to classify Photovoltaic power data into defective clusters and seasonal clusters.

Bu, Zhou, Zhou and Kong in [46] makes a comparison of three methods of hierarchical clustering namely Single Linkage Method, Group Average Method and Ward Method to analyze advantages, disadvantages and the applicability of each method in hydrochemical classification for BaYi tunnel. The results from the research indicate that the Single method is not applicable for complicated situations, Ward method is mainly applicable for samples in the narrow amount [46] and Group Average method is applicable in general scenarios.

Maulik, Mukhopadhyay in [50] used a combination of Simulated Annealing (VSA) based fuzzy clustering method and popular Artificial Neural Network (ANN) based classifier for analyzing microarray gene expression data from three real-life microarray data sets. The performance of this technique has proved to be superior and is demonstrated by comparing with existing clustering algorithms.

An unsupervised system is built by Wang et al. in [51] which used Divisive Hierarchical Clustering clustering to group similar users and identify user behaviour from clickstream data. The system is also able to identify new, previously unknown behaviours and their visualization tools can be used to interpret any identified behaviours

Evans, Pfahringer, and Holmes in [52] proposed a framework where they use various clustering algorithms like K-Means, Farthest first, Expectation Minimization, Bisecting K-Means for reducing the dataset and use it later for classification. The results show that clustering can beneficial when employing a classifier. The results also show that it is important to choose the right clustering method for each data set.

Coletta ar al. in [53] used a combination of clustering and classifier(SVM) called C3E-SL algorithm for tweet sentiment analysis. The results show that the classifier provides best results found, suggesting that this method is promising.

Ding and He in [22] analyzed Internet newsgroup and DNA gene expression and showed that PCA is solutions to discrete cluster belonging indicator for K-means which indicates that PCA is not just dimensionality reduction algorithm and they can perform data clustering too [22][62].

Wenskovitch et al. in [62] studied the combination of Dimension Reduction and Clustering in relation to visual analytics. Though the dimensionality reduction and clustering algorithms are implemented together in many visualization systems but they work independently and in parallel [62]. The authors discussed the challenges inherent in developing a visualization system that uses both families of algorithms [62].

4.1.2 Experiment

An experiment is chosen as a research method to answer RQ2 since working with quantitative data and experiment gives more control over variables. Other research methods like Survey or case-study can be rejected as they are descriptive methods [30].

From the literature review, it is concluded that K-means and Hierarchical clustering can be adopted for our study. The goal of the experiment is to evaluate the performance of the K-means and Hierarchical Agglomerative Clustering (HAC) algorithms on the reduced data (post t-SNE and post PCA) that is based on driving style and choose the best clustering method for the data.

Since clustering is an exploratory task, there is no distinction between independent variables and dependent variables.

The steps performed in the experiment are shown in the figure 4.2:

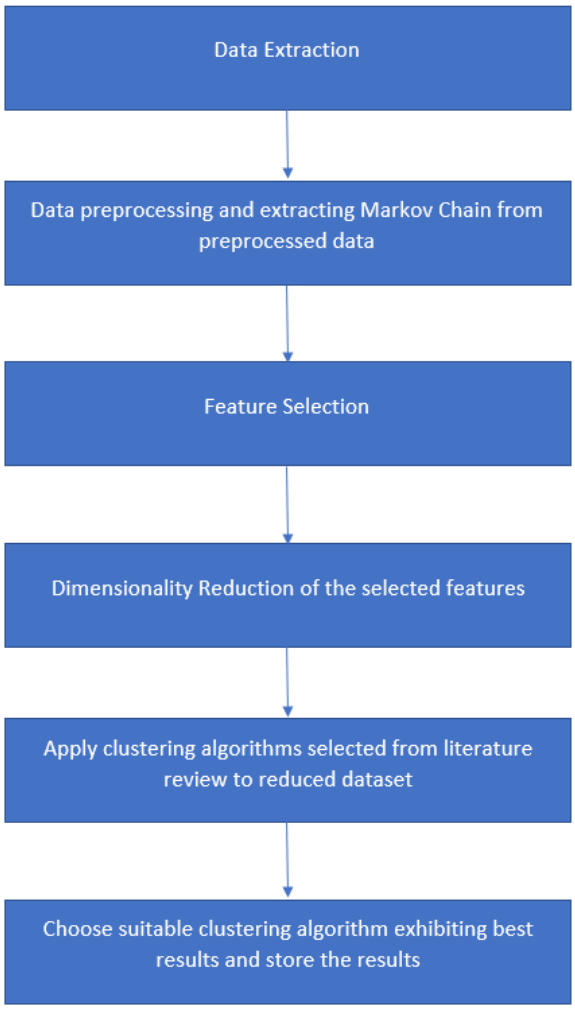

Figure 4.2: Steps performed during experiment

4.2 Software Environment

Software	Description	Version
Sympathy for Data	An open source platform that provides tools to work with complex data and perform data analysis [83]	1.5
Python	A high level, interpreted, general-purpose programming language created by Guido van Rossum [84].	2.7
Spyder	An open source, scientific python development environment [85]	3.3.0

Table 4.1: Software and their version

Libraries	Description
numpy	An open source library that adds support for large, multi-dimensional arrays and matrices, along with a large collection of high-level mathematical functions to operate on these arrays [80]
pandas	A BSD licensed, an open source library that provides high-performance, easy-to-use data structures and data analysis tools [81]
matplotlib	An open source plotting library for the Python programming language and its numerical mathematics extension NumPy [82]
sklearn	An open source library that provides tools for data mining and data analysis [79]

Table 4.2: Python libraries used and their description

4.3 Data Collection

The data is collected through an online portal accessible only to Volvo Cars' employees. Several filters are applied to data during the collection. Several different types of data are read out: Current condition of components and subsystems, Fault codes with additional sensor data and statistical loggers etc.

4.4 Dataset Used

The dataset consists of 24284 data points and 190 features meaning data is extracted from 24284 Volvo Cars driven in 58 countries. The data can be divided into two categories. Table 4.3 shows the categories of data and the feature number i.e., the range of categories in the dataset.

Category	**Feature number**	**Description**
1	1-159	Markov chain features
2	160-190	Meta-data

Table 4.3: Data category

Though we call the features from 1-159 as Markov chain features, the Markov chain is actually needed to be calculated (extracted) from the data after preprocessing is completed.

4.5 Data Preprocessing

A sample in the extracted data consists of features with a number of readouts recorded at end of the driving cycle. Each feature is a condition while driving which is logged only if that criteria is met. A feature may be read several times or even zero times depending on how many times the condition is met when the driver drives. Hence, there is a need to normalize the data in some way. In a simple way, it can be said that there is a need to calculate transition probabilities from the data that has number of readouts (i.e., occurrences).

Markov chain features are needed to be separated from meta-data and then normalized. The Markov Chain features are stored in a file called dataset1 and meta-data in dataset2. First step is to calculate the sum and percentage of the Markov chain features for each sample to normalize the data.

Calculate sum and percentage

Calculate the sum of all cells in a row then divide each cell of a row with the sum and then multiply by 100 to get the percentage of each feature.

For an instance, assume that we have three features in a dataset as shown in the table 4.4.

We calculate the sum of all features in a sample and divide each feature in that sample with the sum, then multiply the result by 100 to get the percentage of readout of each feature as shown in the table 4.5.

C2F1	C2F2	C2F3
1500	3000	4500
1100	3300	2200

Table 4.4: Dummy dataset

C2F1	C2F2	C2F3
(1500/9000)*100	(3000/9000)*100	(4500/9000)*100
(1100/6600)*100	(3300/6600)*100	(2200/6600)*100

Table 4.5: Dummy preprocessed dataset

4.6 Extracting Markov Chain

The preprocessed dataset1 is used to calculate the Markov Chain. The first step is to calculate the transition table for all the data points in dataset1. For each data point, we get one transition table.

Calculating transition matrix

Assume a simple Markov chain dataset with only one data point, three states and five possible transitions. The aim is to calculate the transition table for this data point. Transition table consists of the probability of transition of each state to all states (transition probability).

Let 0, 1 and 2 be the three states and 0–>0, 2–>1, 1–>2, 0–>2 and 1–>0 be the five possible transitions.

Transitions	**0–>0**	**2–>1**	**1–>2**	**0–>2**	**1–>0**
Number of transitions	16	4	2	6	8

Table 4.6: Transitions and number of transitions

The table 4.6 consists of transitions in the first row and number of transitions in the second row. 0–>0 implies the transition from 0 state to 0 state and number of transitions implies that there are 16 transitions from state 0 to 0. Note that there are two transitions from 0 to any states including itself(i.e., 0–>0, 0–>2), similary there are two transitions from 1 to any states including itself(i.e., 1–>2, 1–>0) and one transition from 2 to any states including itself(i.e., 2–>1).

In table 4.7, the total number of transitions from a particular state to any state including itself, for all states, is calculated. This is done by adding each transition count from a state to any state including itself. For example, from 0 there are 16 transitions to itself and 6 transitions to 2 state. Hence, by adding this we get total transitions count from the 0 state to any state including itself.

State	**Total transitions from the state to any state including itself**
0	16+6(=22)
1	2+8(=10)
2	4

Table 4.7: Total transitions count from the state to any state including itself

States	**0**	**1**	**2**
0	16/22(=0.72)	0	6/22(=0.28)
1	8/10(=0.8)	0	2/10(=0.2)
2	0	4/4(=1)	0

Table 4.8: Transition table

The table 4.8 is the transition table for the data point of our simple Markov chain dataset. The probability of transition from each state to all states is calculated in the table. The sum of each row in a transition table must be equal to 1 and there should be the same number of rows and columns in the table. The

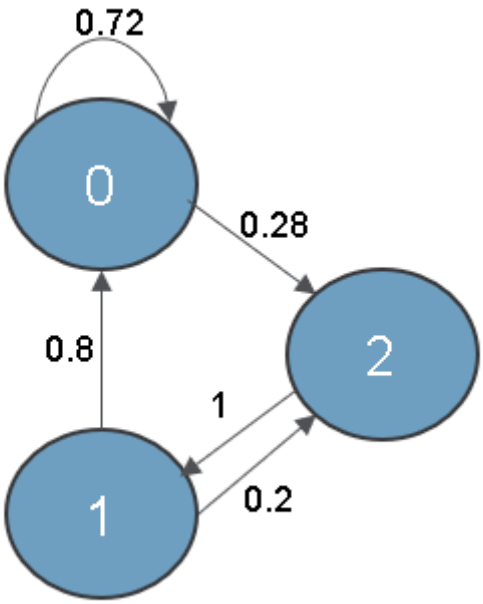

Figure 4.3: Transition diagram example

figure 4.3 is the transition diagram for the transition table 4.8. There are 3 states 0, 1 and 2. The probability of transition from 1 to 2 state is 0.2%. Also, there is a possible transition from 0 to 0 state with 0.72% probability. There is no possible

transition from 0 to 1 state and hence it is an impossible transition.

The dataset for which we calculated transition table is the simplest with only one data point, three states and five transitions. The real-life dataset which is used in the experiment consists of 24284 data points with 23 states and 159 transitions.

Markov Chain Model for the Markov features

The figure 4.4 is the Markov Chain model for dataset1 with all the possible transitions in orange coloured cells and impossible transitions in white coloured cells. The Markov chain obtained is 2D with 23 states in total and 159 possible transitions. The 23 states are present on both horizontal and vertical axis representing To and From respectively. For example, a transition from off to off is not possible and hence white coloured cell, a transition from off to StillStart is possible and hence orange coloured cell. The table 4.9 consists of 23 states of Markov Chain with the description:

States	**Description**
Off	Off state
StillStart	Engine is on and car not moving
StillEarly	Engine has been on for few mins and car is not moving
StillLate	Engine has been on for long time and car not moving
StillMid	Engine has been on for time greater than StillEarly and less than StillLate
Spd1Dec	Decelerating in Spd1
Spd1StSt	Steady State in Spd1, meaning driving at a steady speed in Spd1
Spd1Acc	Accelerating in Spd1
Spd2Dec	Decelerating in Spd2
Spd2StSt	Steady State in Spd2
Spd2Acc	Accelerating in Spd2
Spd3Dec	Decelerating in Spd3
Spd3StSt	Steady State in Spd3
Spd3Acc	Accelerating in Spd3
Spd4Dec	Decelerating in Spd4
Spd4StSt	Steady State in Spd4
Spd4Acc	Accelerating in Spd4
Spd5Dec	Decelerating in Spd5
Spd5StSt	Steady State in Spd5
Spd5Acc	Accelerating in Spd5
Spd6Dec	Decelerating in Spd6
Spd6StSt	Steady State in Spd6
Spd6Acc	Accelerating in Spd6

Table 4.9: States of Markov chain

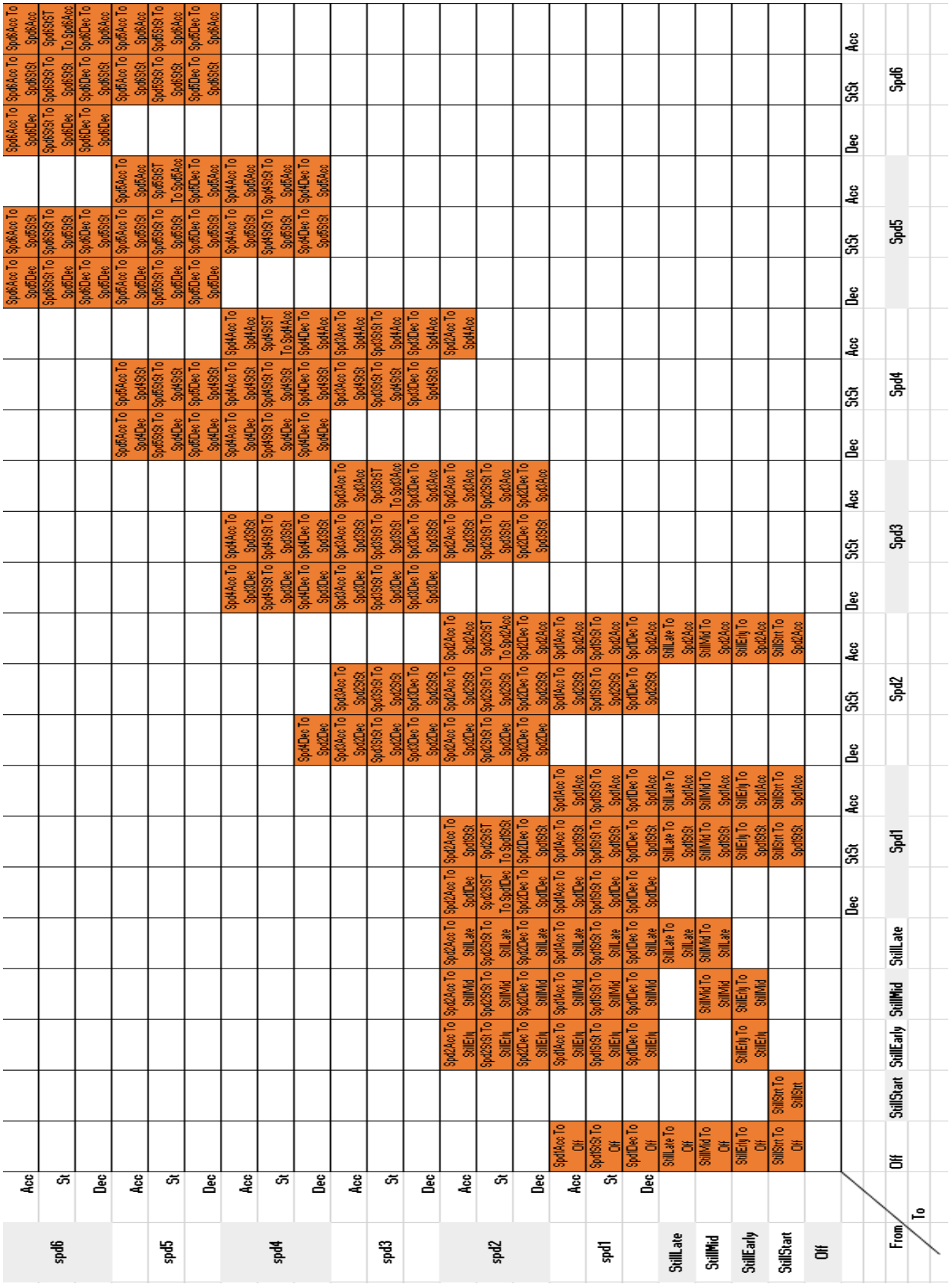

Figure 4.4: Markov chain model

From/To	Off	StillStrt	StillErly	StillMid	StillLate	Spd1Dec	Spd1Stst	Spd1Acc	Spd2Dec	Spd2Stst	Spd2Acc	Spd3Dec	Spd3Stst	Spd3Acc	Spd4Dec	Spd4Stst	Spd4Acc	Spd5Dec	Spd5Stst	Spd5Acc	Spd6Dec	Spd6Stst	Spd6Acc
StillStrt	0.0	1.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
StillErly	0.0	0.0	0.9	0.0	0.0	0.0	0.1	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
StillMid	0.0	0.0	0.0	0.9	0.0	0.0	0.1	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
StillLate	0.0	0.0	0.0	0.0	1.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
Spd1Dec	0.0	0.0	0.1	0.1	0.1	0.3	0.2	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.1	0.1	0.0	0.0	0.0	0.0	0.0	0.0
Spd1Stst	0.0	0.0	0.0	0.0	0.1	0.1	0.5	0.1	0.0	0.0	0.1	0.0	0.0	0.0	0.0	0.0	0.0	0.1	0.0	0.0	0.0	0.0	0.0
Spd1Acc	0.0	0.0	0.0	0.0	0.0	0.0	0.3	0.2	0.0	0.0	0.4	0.0	0.0	0.0	0.1	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
Spd2Dec	0.0	0.0	0.0	0.0	0.0	0.1	0.0	0.0	0.7	0.2	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
Spd2Stst	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.1	0.7	0.1	0.0	0.0	0.0	0.0	0.0	0.1	0.0	0.0	0.0	0.0	0.0	0.0
Spd2Acc	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.2	0.7	0.0	0.0	0.1	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
Spd3Dec	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.3	0.0	0.0	0.5	0.1	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.1	0.0
Spd3Stst	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.1	0.7	0.1	0.0	0.0	0.0	0.1	0.0	0.0	0.0	0.0	0.0
Spd3Acc	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.2	0.6	0.0	0.0	0.2	0.0	0.0	0.0	0.0	0.0	0.0
Spd4Dec	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.1	0.0	0.0	0.7	0.2	0.0	0.0	0.0	0.0	0.0	0.0	0.0
Spd4Stst	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.1	0.8	0.1	0.0	0.0	0.0	0.0	0.0	0.0
Spd4Acc	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.2	0.7	0.0	0.0	0.0	0.0	0.1	0.0
Spd5Dec	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.1	0.0	0.0	0.7	0.2	0.0	0.0	0.0	0.0
Spd5Stst	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.1	0.0	0.0	0.0	0.0	0.9	0.0	0.0	0.0	0.0
Spd5Acc	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.3	0.7	0.0	0.0	0.0
Spd6Dec	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.2	0.0	0.0	0.6	0.2	0.0
Spd6Stst	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.1	0.0	0.9	0.0
Spd6Acc	0.0	0.0	0.0	0.0	0.0	0.0	0.1	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.3	0.6

Figure 4.5: Transition table for a data point in dataset1

The figure 4.5 is the transition table for a data point(single data point) in dataset1. The table represents all the 23 states and all the transitions whether possible or not. Impossible transitions are marked with probability 0.0%. Since this transition table is only for one data point, there is a possibility that the driver must not have driven in all the 23 states. For example, a driver who drives slow will most likely not reach the Spd6Acc state since Spd6Acc indicates very high speed driving.

Datasets

Table 4.10 consists of the dataset after preprocessing.

Dataset name	**Feature range**	**Description**
dataset1	1-159	Markov Chain features
dataset2	160-190	Metadata

Table 4.10: Datasets

The dataset1 is the Markov Chain which will be used in the experiment The dataset2 is meta-data dataset, which consists information about country where car is driven, number of driving cycles, etc. This dataset2 will be used to analyze the driving styles.

4.7 Choosing an optimal number of clusters

Choosing an optimal number of clusters is a very crucial task in clustering. The experienced data analysis team expected at least ten different types of driving styles from the data to analyze i.e., ten clusters at least. To get the optimal number of clusters, the silhouette score for a number of clusters between ten and twenty is calculated. The Markov chain is used as input to the Silhouette method. It is found that ten clusters showed the best silhouette score among the others and hence, ten clusters are chosen as an optimal number of clusters. Elbow method which considers the percentage of explained variance as a function was also used to find an optimal number of clusters, but the elbow could not be seen properly. It is worth noting that the elbow method can sometimes be ambiguous [17, 18, 19, 20].

4.8 Feature Selection

All the dataset1 features, i.e., Markov chain is selected because each feature is based on driving style and the aim is to obtain the clusters to based on driving style.

4.9 Experimental Setup

- Perform K-means and Hierarchical clustering on the dataset obtained after reducing dimensions using Principal Component Analysis and t-distributed Stochastic Neighbor Embedding.

- The performance metrics are noted for each combination run of clustering algorithm and reduced dataset. Since there are two clustering algorithm and two datasets obtained by dimensionality reduction using PCA and t-SNE, there will be four possible combinations and runs in the experiment. The experimental results are analyzed and performance metrics scores are compared to find the suitable clustering algorithm for our problem.

4.9.1 Performance Metrics

An evaluation is needed to be performed on the model when ground truth labels are unknown.

Silhouette Coefficient

Silhouette Coefficient represents the similarity and dissimilarity between samples of each cluster. This technique provides a clear representation of how the sample lies within its cluster [16]. Silhouette coefficient also shows how well a sample is placed in its cluster and if any sample floats between two or more clusters. Given a range of a number of clusters, the best number of clusters can be estimated by computing the average of silhouettes [15].

Let us consider $a(i)$ as the average distance between sample i and other samples which are present in the same cluster as i. Let k be the total number of clusters and C_k represent k^{th} cluster. Let $d(i, C)$ be the average distance between sample i and other samples in C_k. The minimum value of $d(i, C)$ can be given by

$b := min\ d(i, C)$ is distance between i and nearest cluster. The silhouette score $s(i)$ can be computed as

$$s(i) := \frac{b(i) - a(i)}{\max(a(i) - b(i))}$$

Silhouette Coefficient score value lies within the range [-1,1]. +1 score means that the sample is far from its nearest cluster and very close to assigned cluster i.e. well clustered. -1 score means that the sample is close to neighbouring cluster than to currently assigned cluster. 0 score means that the sample is floating between two or more clusters. Hence a score close to 1 is better and to get $s(i)$ close to 1, $a(i)$ should be very small as compared to $b(i)$. It only happens when $a(i)$ is very close to the assigned cluster. A high value of $b(i)$ indicates that the sample is very far from its next nearest cluster. The clustering quality can be

estimated by computing the average value of silhouette scores $s(i)$ of associated samples [14][15].
Advantages:

- The higher the score, the denser and well-separated clusters which refer to cluster's standard concept [29]

Disadvantages:

- The Silhouette score is usually higher for convex clusters than any other cluster concept, like density-based clusters which could be obtained using the DBSCAN clustering algorithm [29]

Calinski-Harabasz Index or Variance Ratio Criterion

Higher Calinski-Harabasz Index implies that model has better defined clusters [31]. The cluster validity is evaluated based on the average between and within-cluster sum of squares [31][47]. For k number of clusters, Calinski-Harabasz Index score s is ratio of between-clusters dispersion mean and within-cluster dispersion [29].

$$s(k) = \frac{\mathrm{Tr}(B_k)}{\mathrm{Tr}(W_k)} \cdot \frac{N-k}{k-1}$$

where B_k is between group dispersion matrix, W_k is within-cluster dispersion matrix and both are defined below

$$W_k = \sum\nolimits_{q=1}^{k} \sum_{x \in C_q} (x - c_q)(x - c_q)^T$$

$$B_k = \sum_q n_q \ (x - c_q)(x - c_q)^T$$

where N is a number of data points in the data, c_q is the center of cluster q, C_q is set of points in q, n is the number of points in q.
Advantages:

- The higher the score, the denser and well-separated clusters which refer to cluster's standard concept [29]
- Calinski-Harabasz Index is fast to compute [29]

Disadvantages:

- The Calinski-Harabasz score is usually higher for convex clusters than any other cluster concept like density-based clusters which could be obtained using the DBSCAN clustering algorithm [29]

Davies-Bouldin Index(DBI)

DBI can be defined as average similarity between each cluster C_i for i=1,2,...k and their most similar cluster C_j [29]. In context of DBI, similarity can be defined as a measure R_{ij} which trades off [29][48]:

- s_i, known as cluster diameter is defined as the average distance between points in cluster i and the centroid of cluster i.
- d_{ij} is distance between the cluster centroid i and j.

$$R_{ij} = \frac{s_i + s_j}{d_{ij}}$$

The Davies-Bouldin Index is defined as:

$$DBI = \frac{1}{k}\sum_{i=1}^{k} \underset{i \neq j}{m} axR_{ij}$$

$$\sqrt{\sum_{i=1}^{K}(q_i - p_i)^2}$$

Advantages:

- Lower DBI relates to model with better separation or partition between the clusters [29]
- Computation of DBI is simpler than that of computation of Silhouette scores [29][48]

Disadvantages:

- Increasing k value(number of clusters) without any penalty will reduce the DBI amount in resulting clustering.
- DBI is not normalized and comparison of two DBI values with two different datasets is difficult
- The DBI score is usually higher for convex clusters than any other cluster concept like density-based clusters which could be obtained using the DBSCAN clustering algorithm [29]
- Centroid usage limits the distance to Euclidean only [29]
- Best score does not imply that the best information can be retrieved [29]

Chapter 5

Results

Based on literature review, KMeans and Hierarchical Clustering algorithms are chosen. After preprocessing is completed, the dimensionality reduction algorithms like PCA and t-SNE are applied to dataset1(Markov Chain). After reducing the dimensions, K-means clustering and HAC algorithm are applied to reduced dimensions dataset.

Principal Component Analysis (PCA)

In this thesis, 159 features that represent Markov Chain i.e., dataset1 are analyzed by principal component analysis. It is found that the first three principal components reflect more than 80% of the original variable information. You can find the new components contribution rate and the cumulative contribution rate tabulated in table 5.1.

New Components	Contribution	Accumulative
x0	0.4855	0.4855
x1	0.2257	0.7112
x2	0.1169	0.8281

Table 5.1: Principal component cumulative contribution rate

The PCA algorithm is used from the package sklearn.decomposition.PCA in python's sklearn library, where the algorithm is already implemented. We have applied PCA to Markov Chain dataset i.e., dataset1. The parameters in the PCA algorithm like [n_components, copy, whiten, svd_solver, tol, iterated_power, random_state] can be tuned. The resultant Principal Component Analysis algorithm had a variance of 80% with three components which were good to move ahead for clustering. The following parameters are tuned and rest are set to default.

Parameters tuned: $n\_components = 3$

t-Distributed Stochastic Neighbor Embedding (t-SNE)

The t-SNE algorithm is used from the package sklearn.manifold.TSNE in Python's sklearn library, where the algorithm is already implemented. We have

applied the t-SNE algorithm to Markov Chain dataset ie., dataset1. The parameters in t-SNE like [n_components, perplexity, early_exaggeration, learning_rate, n_iter, init, n_iter_without_progress, metric, method, verbose, min_grad_norm, angle, random_state] can be tuned. After many runs of the algorithm with a different combination of hyperparameters, we found the desirable result with default hyperparameters. The t-SNE algorithm was used to reduce the data in three dimensions. The following parameters are tuned and rest are set to default.

Parameters tuned: $n\_components = 3$

K-means Clustering

K-means clustering algorithm is applied to the reduced dimensions datasets to obtain the ten clusters (since $k = 10$). The K-means algorithm is used from the package sklearn.cluster.KMeans in Python's sklearn library, where the algorithm is already implemented. The following parameters are tuned and rest are set to default.

Parameters tuned: $n\_clusters = 10$

Hierarchical Agglomerative Clustering

Hierarchical Agglomerative Clustering algorithm is applied to the reduced dimensions datasets to obtain the ten clusters (since $k = 10$). The K-means algorithm is used from the package sklearn.cluster.AgglomerativeClustering in Python's sklearn library, where the algorithm is already implemented. The following parameters are tuned and rest are set to default.

Parameters tuned: $n\_clusters = 10$, $linkage = complete$

5.1 Clustering Visualization

The figure 5.1 shows the cluster labels of figure 5.2 and 5.3. The reason for attaching a separate figure of label legend is that the legend in the sub-figures of figure 5.2 and 5.3 are not clearly visible and figure 5.1 can help to identify the clusters. Similarly, the table 5.2 consists of a cluster label with the colour name for a even better understanding.

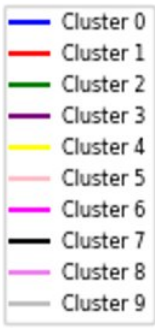

Figure 5.1: Cluster label legend

Cluster label	Colour
0	blue
1	red
2	green
3	purple
4	yellow
5	pink
6	magenta
7	black
8	violet
9	silver

Table 5.2: Cluster labels and colors

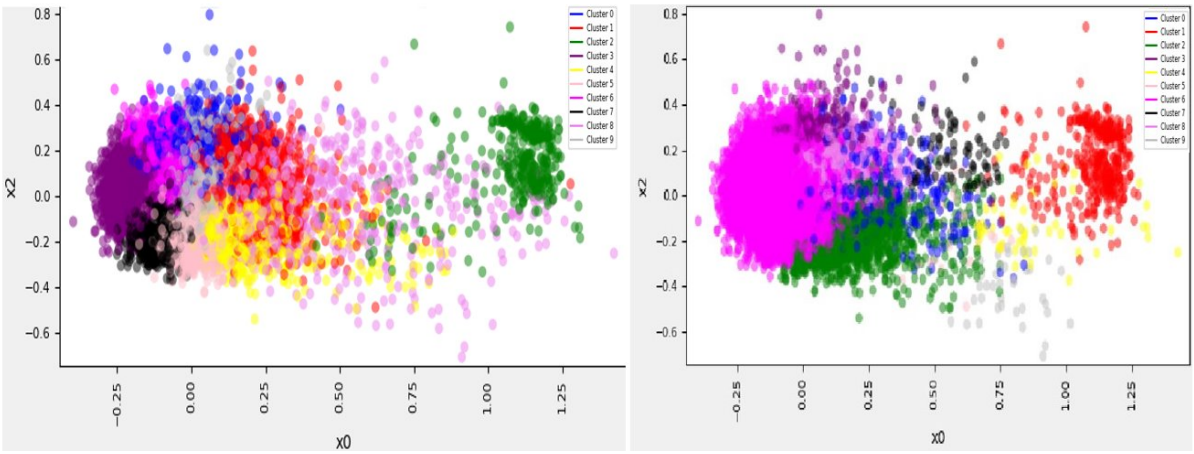

Figure 5.2: K-means and HAC clustering with PCA dimensions

The figure 5.2 is a combination of sub-figures showing the figures of clusters obtained by applying K-means and HAC algorithm to PCA reduced dimensions respectively. The sub-figure on the left shows K-Means and on the right shows HAC result, with PCA dimensions. Though the first two principal components gave the maximum information about the data, the first principal component and third component showed the best visualization view. The clusters in the sub-figures of the figure 5.2 looks very similar. The clusters look spherical in shape. The green colour clusters i.e., Cluster 2 in KMeans sub-figure and the red colour clusters i.e., Cluster 1 in HAC sub-figure looks similar and are far from other clusters in their respective sub-figures. The pink colour clusters near green colour in KMeans figure (left-side) looks very closer to green but are far away in reality. It looks closer because we are viewing a three-dimensional figure in two dimensions when viewed in three dimensions Green colour clusters are separated from others.

Similarly, yellow colour clusters near red colour in HAC figure (right-side) looks very closer to red but are far away in reality.

The figure 5.3 is a combination of sub-figures showing the clusters obtained by applying K-means and HAC algorithms to t-SNE reduced dimensions. The figure on the left is K-Means and on the right is HAC result with t-SNE dimensions. The clusters in the sub-figures of the figure 5.3 looks very similar. The clusters in the sub-figures looks non-spherical in shape. The green color clusters i.e., Cluster 2 in KMeans figure (left-side) looks like they are far from other clusters. Similarly, the black color clusters i.e., Cluster 7 in HAC figure(right-side) looks like they are far from other clusters. These clusters look closer because we are viewing a three dimensional figure in two dimensions.

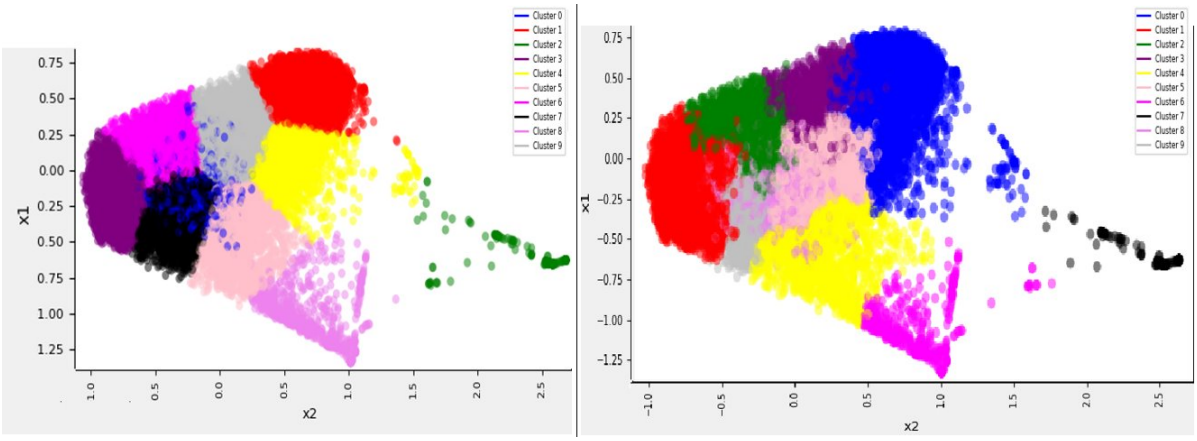

Figure 5.3: K-means and HAC clustering with t-SNE dimensions

The data could also have been reduced to two dimensions directly instead of three dimensions. Volvo Cars had a visualization tool which could view 3D data and hence that tool was used to visualize and analyze the clusters. The team preferred the cool 3D view rather than the old 2D view. Hence, the dimensions are reduced to 3D instead of 2D.

Chapter 6

Analysis

The clustering visualizations in chapter 5 shows that the K-means and HAC algorithms produce similar looking clusters when both algorithms applied to the same reduced dataset. The sub-figures in the figure 5.2 shows the visualization of clusters obtained by using K-means clustering and HAC respectively for PCA reduced dataset. The sub-figures in the figure 5.3 shows the visualization of clusters obtained by using K-means clustering and HAC respectively for the t-SNE reduced dataset. The sub-figures in figure 5.2 look similar to each other in terms of the shapes of the cluster, locations of data points on the graph, etc. Sub-figures in figure 5.3 also looks similar to each other in terms of the shapes of the cluster, locations of data points on the graph, etc. Hence, it will be valuable to understand the key differences between PCA and t-SNE algorithms.

Few differences between PCA and t-SNE dimensionality reduction algorithms that I found when working with them are tabulated in table 6.1

Performance Evaluation

Why do we evaluate clusters?

1. Comparing two or more clustering algorithms
2. Comparing two sets of clusters
3. Comparing two clusters

How do we evaluate clusters?

We have chosen three performance metrics to evaluate our clusters. They are:

1. Silhouette Coefficient
2. Calinski-Harabasz Index
3. Davies-Bouldin Index

The reduced dimensions dataset and the labels obtained as output from clustering algorithms are used as input to these performance metrics.

PCA	**t-SNE**
Linear dimensionality reduction algorithm	Non-linear dimensionality reduction algorithm
Easy to interpret what is happening at each step of the algorithm. Even though it is easy to interpret, we do have features which are not easily interpretable	Difficult to interpret what is happening at each step of algorithm
Finds principal components of the data as the name suggests	Similar objects are modelled by nearby points and dissimilar objects are modelled by distant points with high probability
Does not solve Crowding problem	Solves the Crowding problem
PCA is focused on saving the shape of data	t-SNE is more focused on saving the probability distribution of the neighboring points. If two points are far in original data then they will also be far in reduced data and vice-versa
Low computational requirements(Memory and Speed)	High computational requirements, usually very slow for large datasets
Not much sensitive to hyperparameters tuning	Very sensitive to hyperparameters tuning

Table 6.1: PCA vs t-SNE

6.0.1 Silhouette Coefficient

It evaluates the average distance between a sample and other samples which are present in same cluster. Silhouette Coefficient score value lies within the range [-1,1]. +1 score means that the sample is far from its nearest cluster and very close to assigned cluster i.e. well clustered. -1 score means that sample is close to neighboring cluster than to currently assigned cluster. 0 score means that sample is floating between two or more clusters.

	KMeans	**HAC**
PCA	0.29	0.27
t-SNE	0.39	0.35

Table 6.2: Silhouette coefficient scores

The table 6.2 shows the Silhouette Coefficient Index for different algorithms. The combination of PCA and K-means has the Silhouette Coefficient Index of

0.29, the combination of t-SNE and K-means has the Silhouette Coefficient Index of 0.39, the combination of PCA and HAC has the Silhouette Coefficient Index of 0.27 and the combination of t-SNE and HAC has the Silhouette Coefficient of 0.35.

6.0.2 Calinski-Harabasz Index

K-means is a non-deterministic algorithm. Hence, samples of the Calinski-Harabasz Index are stored for many runs of the algorithm with ten clusters and the average score of is calculated for Kmeans algorithm.

	KMeans	**HAC**
PCA	332.19	313.51
t-SNE	485.37	470.25

Table 6.3: Calinski-Harabasz Index

Calinski-Harabasz Index evaluates how denser and separated are the clusters. The higher the score, the denser and well separated clusters which refers to cluster's standard concept [29].

The table 6.3 shows the Calinski-Harabasz Index for different algorithms. The combination of PCA and K-means has the Calinski-Harabasz Index of 332.19, the combination of t-SNE and K-means has the Calinski-Harabasz Index of 485.37, the combination of PCA and HAC has the Calinski-Harabasz Index of 313.51 and the combination of t-SNE and HAC has the Calinski-Harabasz Index of 470.25.

6.0.3 Davies-Bouldin Index

Davies-Bouldin Index evaluates how well the clustering is performed based on features and quantities. Lower Davies-Bouldin Index relates to model with better separation or partition between the clusters [29].

	KMeans	**HAC**
PCA	0.722	0.747
t-SNE	0.571	0.603

Table 6.4: Davies-Bouldin Index

The table 6.4 shows the Davies-Bouldin Index for different algorithms. The combination of PCA and K-means has the Davies-Bouldin Index of 0.722, the

combination of t-SNE and K-means has the Davies-Bouldin Index of 0.571, the combination of PCA and HAC has the Davies-Bouldin Index of 0.747 and the combination of t-SNE and HAC has the Davies-Bouldin Index of 0.603.

6.0.4 Comparison of Performance Metrics

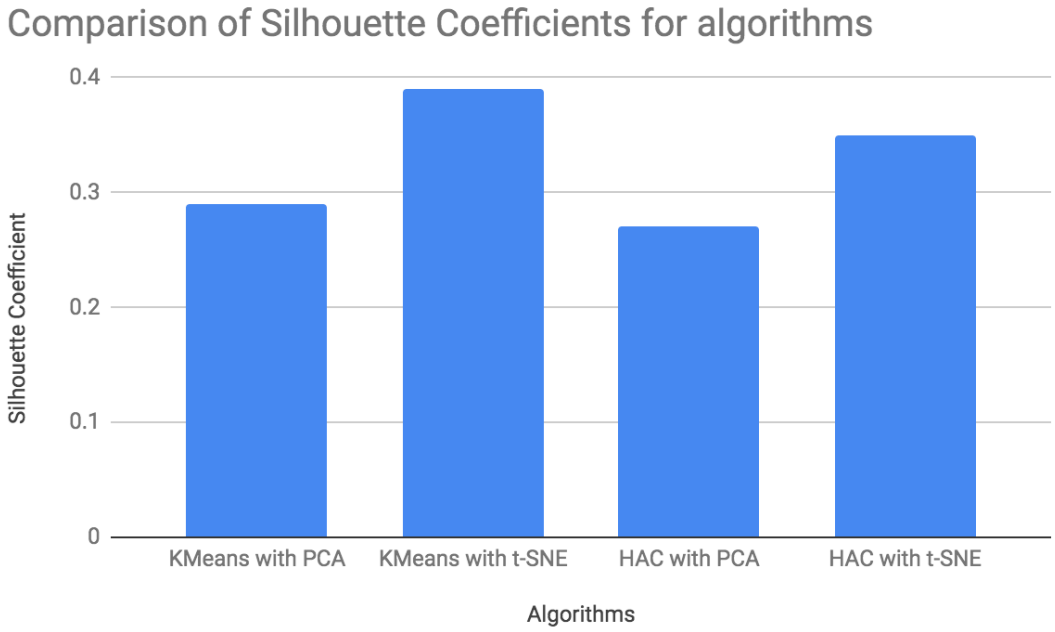

Figure 6.1: Silhouette coefficient vs algorithms

Figure 6.1 is a visualization for table 6.2. The bars in the figure represents the silhouette Coefficient for different algorithms. The KMeans with t-SNE algorithm has the highest silhouette coefficient. This indicates that KMeans with t-SNE produced the best clusters among all the algorithms.

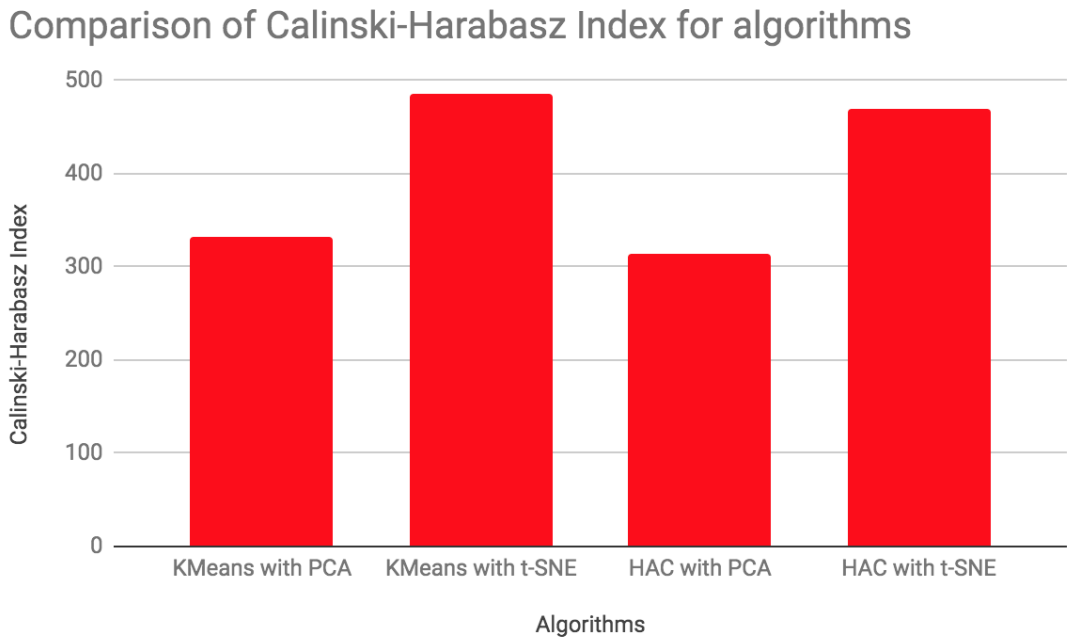

Figure 6.2: Calinski-Harabasz Index vs algorithms

Figure 6.2 is a visualization for table 6.3. The bars in the figure represents the Calinski-Harabasz Index for different algorithms. The KMeans with t-SNE algorithm has the highest Calinski-Harabasz Index. This indicates that KMeans with t-SNE algorithm produced denser and better-separated clusters among all the algorithms.

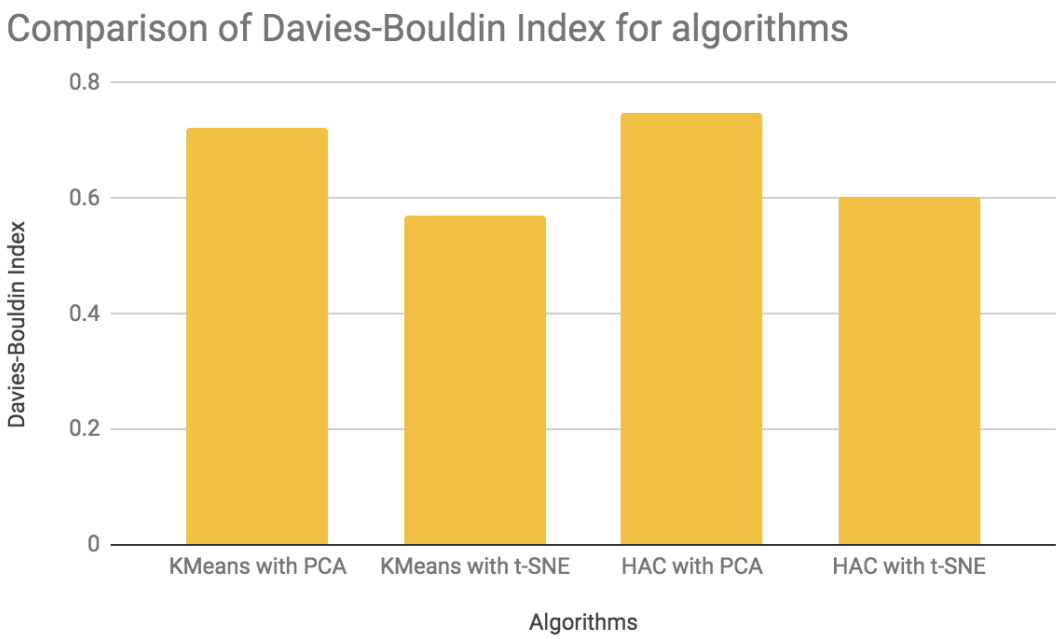

Figure 6.3: Davies-Bouldin Index vs algorithms

Figure 6.3 is a visualization for table 6.4. The bars in the figure represents the Davies-Bouldin Index for different algorithms. The KMeans with t-SNE algorithm has the lowest Davies-Bouldin Index. This indicates that KMeans with t-SNE algorithm performed the best clustering based on features and quantities among all the algorithms.

In figure 6.1, K-means for t-SNE dimensions has the highest Silhouette Coefficient Score when compared to others. In figure 6.2, K-means for t-SNE dimensions has the highest Calinski-Harabasz Index when compared to others. In figure 6.3, K-means for t-SNE dimensions has the lowest Davies-Bouldin Index when compared to others. Hence, it is reasonable to conclude that K-means with t-SNE is the suitable clustering algorithm for the problem.

6.0.5 Further analysis of clusters obtained from K-means with t-SNE dimensions

Distribution of data points per clusters

We start with analysis of number of data points or samples that lie inside each cluster. Let C = 0,1,2,3,4,5,6,7,8,9 be the set of ten clusters.

Cluster	Number of Data points
0	1156
1	4099
2	366
3	4468
4	1656
5	1841
6	3433
7	2867
8	698
9	3700

Table 6.5: Distribution of data points per clusters

Analyzing Speed Profile for Cluster 2

After analyzing the number of data points in cluster 2 and left sub-figure of figure 5.3, it can seen that cluster-2 is a unique cluster and is far from other clusters. Hence, we analyze cluster 2 separately and compare with other cluster based on speed profiles. Note that Spd7 used for speed profiling not used in Markov Chain since it is very high speed driving and anything beyond Spd6 is considered as very high speed driving. The speed profile of cluster 2 and other clusters are shown in the figures below:

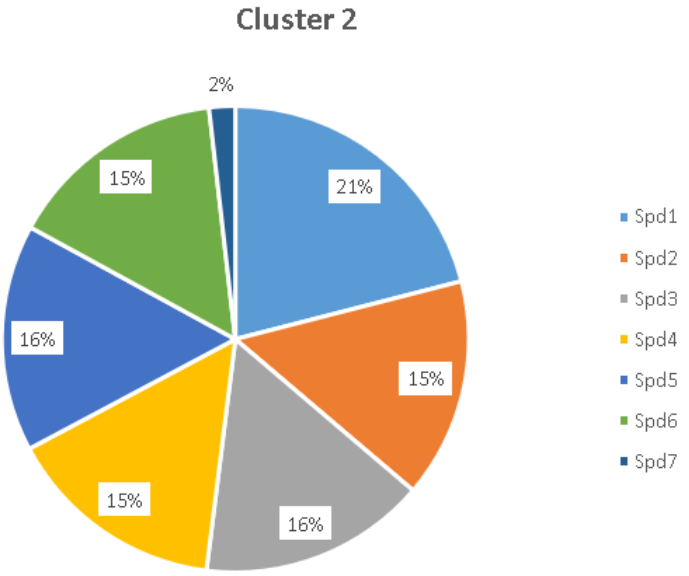

Figure 6.4: Speed profile of cluster 2

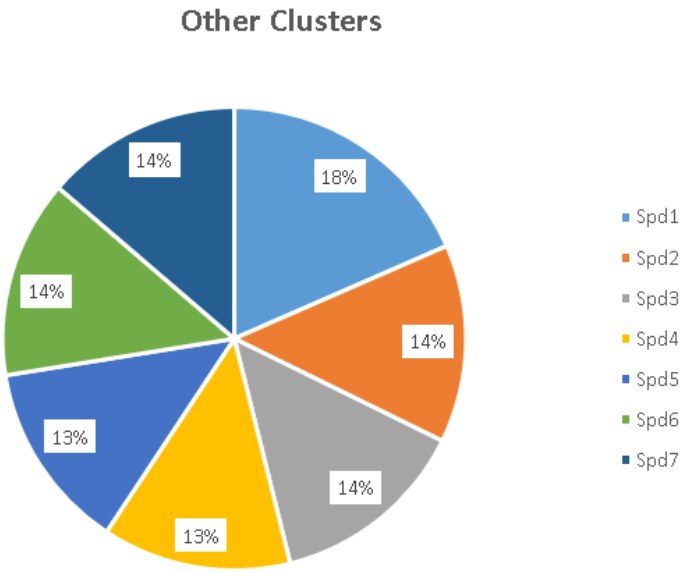

Figure 6.5: Speed profile of other clusters

The speed profile indicates the percentage of driving at that particular speed. Spd7 indicates the very highest speed and Spd1 indicates the lowest speed i.e., Spd7 > Spd6 > Spd5 > Spd4 > Spd3 > Spd2 > Spd1. For example, from figure 6.4, we can say that for data points in cluster 2, 21% of the driving was at Spd1, 15% at Spd2, 16% at Spd3, 15% at Spd4, 16% at Spd5, 15% at Spd6 and 2% at Spd7. From figure 6.4 and figure 6.5, we see that there is a big difference in driving at Spd7 i.e., during very high-speed driving. We can see that the percentage of driving at Spd7 for cluster 2 is only 2%, whereas for other clusters its around 14%. It can be concluded that drivers of cluster 2 rarely drive in very high speed.

Distribution of data points per country per cluster

Cluster	C12	C14	C23	C28	C35	C40	C42	C49	C51	C53	Others
0	1	0	3	1	54	2	0	0	2	1	36
1	4	10	1	0	0	4	1	1	10	64	5
2	2	1	1	0	1	0	45	3	1	33	13
3	16	0	6	13	1	43	1	4	4	0	12
4	3	4	2	0	4	4	31	7	15	13	17
5	1	1	1	0	3	3	55	12	4	2	18
6	21	1	8	2	1	36	1	2	14	3	11
7	6	1	6	3	2	22	9	19	9	2	21
8	0	1	0	0	1	0	76	3	1	3	15
9	11	4	6	0	2	21	2	3	22	18	11

Figure 6.6: Distribution of data points per country per cluster

The figure 6.6 shows the distribution of data points per countries per cluster. Due to confidentiality, country names cannot be disclosed and hence, names likes C1, C2....C58 are used since Cars are driven in 58 countries. The reason for choosing only these countries for analysis is that these are the top 10 countries in terms of the number of data points i.e., Country C12 has the highest number of data points (or Cars) for the given dataset, followed by C14, C23, C28 and so on.

When we(I and experts) analyzed the countries based on clusters, we found that for many clusters, countries in the same clusters has geographical similarities

and driving conditions. Especially, the countries in cluster two had similar driving conditions. It is worth noting that the driving conditions and traffic plays a key role in the driving style of a driver and we found driving conditions to be similar for countries in a cluster. Though this is not the aim of the experiment, it verifies the goodness of clusters in an indirect manner.

6.0.6 Analysis of driving styles for each cluster

Summary of analysis of clusters is presented in the table 6.6

Cluster label	**Analysis**
0	Continuous Speed Change
1	Less high speed and more steady speed driving
2	Special cluster with rare high-speed driving
3	High-speed driving and short driving cycles
4	Steady speed Driving
5	Less high speed; More speed change
6	More speed change; Less steady state
7	Less speed change; less high-speed driving; High Still state
8	More steady state in high speed; High Still state
9	Long driving cycles; Accelerates more and changes speed often

Table 6.6: Analysis of driving styles for each cluster

The table 6.6 consists of the analysis of each cluster based on driving style. The analysis is performed with the help of heat maps visualization of Markov Chain in figure 9.1 to 9.10 attached in the appendices. The values of a cell represent the average percentage of transition from one state to any state for a cluster. The white coloured cells represent impossible transitions. Note that all the values are rounded to two digits after the decimal. For example, a value like 0.005 might be rounded to 0.0. These heat maps figures are very helpful to analyze the driving styles of drivers of each cluster. However, heat maps are found difficult to understand and analyze with our data.

Chapter 7

Discussion

Clustering is an unsupervised learning problem and this makes it difficult. We are expected to infer structure from a given dataset (clusters/categories in our case). The main problem with clustering is that there is not a "correct" or ground truth solution necessarily that could be referred to check our solution. In contrast, for Classification problems, we do know the ground truth. A famous classification problem is to predict whether a given patient has a common disease or not based on a list of symptoms. To make this judgment, we can refer to past clinical records and gather more data to check our prediction. In this case, we have an observable self-evident ground truth i.e., a patient has disease X or does not have disease X. We lack this evidence for clustering. This makes clustering a very open problem where there is no good way to determine or conclude that the solution is correct.

7.0.1 Answer to Research Questions

RQ1: Which clustering algorithms should be used for grouping drivers based on driving style for the given dataset?
Answer: From the conducted literature review, we conclude that K-means and Hierarchical Clustering algorithms can be used for grouping drivers based on driving style for the given dataset.

RQ2: Which algorithm gives the best clustering results for the used dataset?
Answer: Based on performance metrics evaluation and expert's evaluation of clusters, it is concluded that the K-means algorithm gives the best driver groups for the used dataset. In this experiment, t-SNE dimensionality reduction algorithm with K-means clustering algorithm has the highest Silhouette Coefficient score, Calinski-Harabasz Index and the lowest Davies-Bouldin Index when compared to the other combinations. We have further performed analysis based on the labels obtained by using K-means for t-SNE dimensions.

7.0.2 Contribution

Although there has been a lot of research on clustering algorithms, there has been no particular research on clustering of a 2D Markov Chain to analyze driving behaviour of drivers driving Volvo Cars, to my knowledge. Also, the datasets used are very unique and has been never used to conduct an experiment. The thesis shows that clustering can also be applied to Markov Chain data based on driving styles.

7.0.3 Threats to validity

Validity is "an indication of how well an assessment actually measures what it is supposed to measure." [65]

Internal Validity refers to how well a research is performed [77]. The threat of losing or missing an observation/observations is mitigated by storing multiple copies of each logged observations on a cloud which is always watched by an observer.

External Validity refers to the extent to which the research results can be generalized to a larger group other than the group performing the research [77]. External validity is achieved in this study by using real-world data that has been collected from multiple cars (24284 cars driven in 58 countries) and can be used to evaluate the algorithm and performance of the algorithm.

Conclusional Validity checks whether the data obtained from the experiment and results are right [77]. This usually arises when improper evaluation metrics are chosen and the research is conducted without following proper methods and steps. This threat is mitigated by following proper methods and steps while conducting the research. Proper evaluation metrics have been chosen to analyze and evaluate the clustering algorithms in this study.

Chapter 8

Conclusion and Future Work

8.1 Conclusion

The primary focus of this research is to cluster the given dataset to obtain a group of drivers based on their driving styles. The important thing to know about the dataset used for clustering is that the dataset is a 2D Markov Chain. In other words, one could say that clustering is performed on a Markov Chain. Clustering gives valuable insights about driving patterns to data engineers. There are 159 features in the Markov Chain dataset and hence it is difficult to visualize the data with such a high number of dimensions. PCA and t-SNE algorithms have been used to reduce the dimensions of the data for visualization. After reducing the dimensions, K-means clustering and HAC algorithms have been evaluated thoroughly which are used to cluster the drivers based on the driving styles. The evaluation is conducted with performance metrics like Silhouette Coefficient Index, Calinski-Harabasz Index and Davies-Bouldin Index. It is found that K-means with t-SNE dimensions is the best clustering algorithm which can be used to cluster the given dataset, thus fulfilling the aim of this research.

t-SNE has the biggest impact on the outcome of the clustering in this study. The hyperparameters of t-SNE can be tuned with thousands of different combinations obtaining different result each time. It is very important to understand how t-SNE algorithms work. The original paper [25] of t-SNE can be intimidating to read and understand at first, but when reading a couple of times, it can be very helpful to understand t-SNE. The perplexity hyperparameter plays a major role in working of t-SNE. In clustering, there is not a "correct" or ground truth solution necessarily that could be referred to check our solution. The best that could be done is to verify the quality of clusters. There are different evaluation metrics for measuring purpose. The three metrics selected in this study helps to cover various measures like separation, density, etc. There is no general threshold for metrics that says clusters with this score should only be accepted. Real world data is dirty and needs to undergo through various stages of cleaning before using it to solve a problem. Clustering is all about exploring the data where it is important to make good decisions and make the best use of what data is available.

The problem with the data in the thesis is that the data points are numbers in high dimensional space which makes it more difficult to understand the data. It is possible to understand the meaning of each feature but not possible to understand a data point as a whole since there are 159 features. It is also not possible to handpick a few data points and place it in a suitable cluster and verify if clustering labels have resulted in the same. For example, if the problem was clustering twitter data to obtain sentiment of tweets (three groups ie positive, negative or neutral) then it could be easy to read few tweets and infer the sentiment and add it one of the three groups and validate the tweet against clustered label obtained. This kind of validation was also not possible with our data since it was not known what kind of drivers to expect. Even if a few driver types could be guessed, it would be very difficult to understand the data and infer meaning from it since it is high dimensional. Clustering plays a key role in the when you have to solve a problem this kind where there is no much information from the data and everything needs to be analyzed from scratch.

8.2 Future Work

The resultant label obtained after clustering along with the datasets can be used to build a classification model. This classification model can be used to the classify similar data in the future. The driver groups can also be used to analyze monitoring performance and develop a machine learning model to predict monitoring failures.

References

[1] P. Pelliccione et al., “Automotive Architecture Framework: The experience of Volvo Cars,” Journal of Systems Architecture, vol. 77, pp. 83–100, Jun. 2017.

[2] Wikipedia contributors. "Volvo Cars." Wikipedia, The Free Encyclopedia. Wikipedia, The Free Encyclopedia, 10 Mar. 2019. Web. 28 Apr. 2019.

[3] Qidong Wang, Hong Huo, Kebin He, Zhiliang Yao, Qiang Zhang, "Characterization of vehicle driving patterns and development driving cycles in Chinese cities", Transportation Research Part D, no. 13, pp. 289-297, 2008.

[4] Mengliang Li, Jianwei Zhang, Fuxing Zhang, "A Study on Real Driving Cycle of Passenger Cars in Typical Cities of China", Automotive Engineering, vol. 28, no. 6, pp. 554-557, 2006.

[5] Yanxiang Yang, Xioalin Cai, Qing Du, Changwen Liu, "Investigation into Vehicle Driving Mode in Tianjin", Automotive Engineering, vol. 24, no. 3, pp. 200-205, 2005.

[6] F. Zhang, F. Guo and H. Huang, "A research on driving cycle for electric cars in beijing," 2016 Chinese Control and Decision Conference (CCDC), Yinchuan, 2016, pp. 4450-4455.

[7] Wikipedia contributors. "Cluster analysis." Wikipedia, The Free Encyclopedia. Wikipedia, The Free Encyclopedia, 20 Apr. 2019. Web. 29 Apr. 2019.

[8] H. Wahidah, L.V. Pey et al., "Application of Data Mining Techniques for Improving Software Engineering", The 5th International Conference on Information Technology, vol. 2, pp. 1-5, 2011.

[9] Nisha and P. J. Kaur, "Cluster quality based performance evaluation of hierarchical clustering method," 2015 1st International Conference on Next Generation Computing Technologies (NGCT), Dehradun, 2015, pp. 649-653.

[10] R.R. Henrique, E.A.A. Ahmed, "Proposed Application of Data Mining Technique for Clustering Software Projects", INFOCOMP- special edition, pp. 43-48, Jul 2010.

[11] J. Aastha, K. Rajneet, "Review: Comparative Study of Various Clustering Techniques in Data Mining", International Journal of Advanced Research in Computer Science and Software Engineering, vol. 3, pp. 55-57, March 2013.

[12] Wikipedia contributors. "Markov chain." Wikipedia, The Free Encyclopedia. Wikipedia, The Free Encyclopedia, 2 May. 2019. Web. 6 May. 2019.

[13] Wikipedia contributors. "Markov property." Wikipedia, The Free Encyclopedia. Wikipedia, The Free Encyclopedia, 8 Jul. 2018. Web. 6 May. 2019.

[14] Peter J. Rousseeuw, Silhouettes: A graphical aid to the interpretation and validation of cluster analysis, Journal of Computational and Applied Mathematics, Volume 20, 1987, Pages 53-65, ISSN 0377-0427, https://doi.org/10.1016/0377-0427(87)90125-7.

[15] Camila Maione, Donald R. Nelson, Rommel Melgaço Barbosa, Research on social data by means of cluster analysis, Applied Computing and Informatics, 2018, ISSN 2210-8327, https://doi.org/10.1016/j.aci.2018.02.003.

[16] Wikipedia contributors. "Silhouette (clustering)." Wikipedia, The Free Encyclopedia. Wikipedia, The Free Encyclopedia, 20 Mar. 2019. Web. 9 May. 2019.

[17] Wikipedia contributors. "Determining the number of clusters in a data set." Wikipedia, The Free Encyclopedia. Wikipedia, The Free Encyclopedia, 23 Feb. 2019. Web. 29 Apr. 2019.

[18] David J. Ketchen Jr; Christopher L. Shook (1996). "The application of cluster analysis in Strategic Management Research: An analysis and critique". Strategic Management Journal. 17 (6): 441–458.

[19] E. Muningsih, A. B. S. I. Yogyakarta, "Optimasi jumlah cluster k-means dengan metode elbow untuk pemetaan pelanggan" in Pros. Semin. Nas. ELINVO, pp. 105-114, September 2017.

[20] D. Marutho, S. Hendra Handaka, E. Wijaya and Muljono, "The Determination of Cluster Number at k-Mean Using Elbow Method and Purity Evaluation on Headline News," 2018 International Seminar on Application for Technology of Information and Communication, Semarang, 2018, pp. 533-538.doi: 10.1109/ISEMANTIC.2018.8549751

[21] F. A. Espinoza, J. M. Oliver, B. S. Wilson, and S. L. Steinberg, “Using Hierarchical Clustering and Dendrograms to Quantify the Clustering of Membrane Proteins,” Bulletin of Mathematical Biology, vol. 74, no. 1, pp. 190–211, Jan. 2012.

[22] Chris Ding and Xiaofeng He. 2004. K-means clustering via principal component analysis. In Proceedings of the twenty-first international conference on Machine learning (ICML '04). ACM, New York, NY, USA, 29-.
DOI: https://doi.org/10.1145/1015330.1015408

[23] H. Abdi and L. J. Williams, "Principal component analysis," Wiley Interdisciplinary Reviews: Computational Statistics, vol. 2, no. 4, pp. 433–459, Jul. 2010.

[24] Wikipedia contributors. "Dimensionality reduction." Wikipedia, The Free Encyclopedia. Wikipedia, The Free Encyclopedia, 9 Mar. 2019. Web. 19 Apr. 2019.

[25] Maaten, Laurens van der, and Geoffrey Hinton. "Visualizing data using t-SNE." Journal of machine learning research 9.Nov (2008): 2579-2605.

[26] "Hierarchical Clustering" in , Cambridge University Press, April 2009.

[27] S. Patel, S. Sihmar and A. Jatain, "A study of hierarchical clustering algorithms," 2015 2nd International Conference on Computing for Sustainable Global Development (INDIACom), New Delhi, 2015, pp. 537-541.

[28] K. Sasirekha, P. Baby, "Agglomerative Hierarchical Clustering Algorithm-A Review", vol. 3, 2013.

[29] "Hierarchical Clustering", scikit learn. [Online]. Available https://scikit-learn.org/stable/modules/clustering.html#hierarchical-clustering [Accessed: 07-May-2018]

[30] J. L. Blanco, S. Fuchs, M. Parsons, and M. J. Ribeirinho, "Artificial intelligence: Construction technology's next frontier | McKinsey Company."

[31] Caliński, T., & Harabasz, J. (1974). "A dendrite method for cluster analysis". Communications in Statistics-theory and Methods 3: 1-27.
doi:10.1080/03610926.2011.560741

[32] Marcelo Maia, Jussara Almeida, and Virgílio Almeida. 2008. Identifying user behaviorin online social networks. In Proceedings of the 1st Workshop on Social Network Systems(SocialNets '08). ACM, New York, NY, USA, 1-6.

[33] Gang Wang, Xinyi Zhang, Shiliang Tang, Haitao Zheng, and Ben Y. Zhao. 2016. Unsupervised Clickstream Clustering for User Behavior Analysis. In Proceedings of the 2016 CHI Conference on Human Factors in Computing Systems (CHI '16). ACM, New York, NY, USA, 225-236.

[34] Alexander Hinneburg and Daniel A. Keim. Clustering methods for large databases: From the past to the future. In Alex Delis, Christos Faloutsos, and Shahram Ghandeharizadeh, editors, SIGMOD 1999, Proeedings ACM SIGMOD International Conference on Management of Data, June 1-3, 1999, Philadephia, Pennsylvania, USA. ACM Press, 1999.

[35] P. Schnell. A method for discovering data-groups. Biomertica, 6:47-488, 1964.

[36] R. Rojas. Neural Networks - A systematic introduction. Springer, Berlin, 1996.

[37] L. Portnoy, E. Eskin, and S. Stolfo, "Intrusion detection with unlabeled data using clustering," in In Proceedings of ACM CSS Workshop on Data Mining Applied to Security (DMSA-2001), 2001, pp. 5–8.

[38] M. Collins and Y. Singer, "Unsupervised Models for Named Entity Classification," in In Proceedings of the Joint SIGDAT Conference on Empirical Methods in Natural Language Processing and Very Large Corpora, 1999, pp. 100–110.

[39] A. J. Schuit, A. J. M. van Loon, M. Tijhuis, and M. C. Ocké, "Clustering of Lifestyle Risk Factors in a General Adult Population," Preventive Medicine, vol. 35, no. 3, pp. 219–224, Sep. 2002.

[40] R. Kalsoom and Z. Halim, "Clustering the driving features based on data streams," INMIC, Lahore, 2013, pp. 89-94.
doi: 10.1109/INMIC.2013.6731330

[41] N. Garg and R. Rani, "Analysis and visualization of Twitter data using k-means clustering," 2017 International Conference on Intelligent Computing and Control Systems (ICICCS), Madurai, 2017, pp. 670-675.
doi: 10.1109/ICCONS.2017.8250547

[42] R. Nijhawan, I. Srivastava and P. Shukla, "Land cover classification using super-vised and unsupervised learning techniques," 2017 International Conference on Computational Intelligence in Data Science(ICCIDS), Chennai, 2017, pp. 1-6.
doi: 10.1109/ICCIDS.2017.8272630

[43] S. Lu et al., "Clustering Method of Raw Meal Composition Based on PCA and Kmeans," 2018 37th Chinese Control Conference (CCC), Wuhan, 2018, pp. 9007-9010.
doi: 10.23919/ChiCC.2018.8482823

[44] Q. Shi, L. Xu, Z. Shi, Y. Chen and Y. Shao, "Analysis and Research of the Campus Network User's Behavior Based on k-Means Clustering Algorithm,"

2013 Fourth International Conference on Digital Manufacturing & Automation, Qingdao, 2013, pp. 196-201.
doi: 10.1109/ICDMA.2013.46

[45] S. Park and Y. B. Park, "Photovoltaic power data analysis using hierarchical clustering," 2018 International Conference on Information Networking (ICOIN), Chiang Mai, 2018, pp. 727-731.
doi: 10.1109/ICOIN.2018.8343214

[46] J. Bu, J. Zhou, A. Zhou and F. Kong, "The comparison of different methods in hydrochemical classification using hierarchical clustering analysis," 2011 International Conference on Remote Sensing, Environment and Transportation Engineering, Nanjing, 2011, pp. 1783-1787.

[47] Y. Liu, Z. Li, H. Xiong, X. Gao and J. Wu, "Understanding of Internal Clustering Validation Measures," 2010 IEEE International Conference on Data Mining, Sydney, NSW, 2010, pp. 911-916.
doi: 10.1109/ICDM.2010.35

[48] D. Davies, D. Bouldin, "A cluster separation measure", IEEE PAMI, vol. 1, no. 2, pp. 224-227, 1979.

[49] Hart, Chris (2018). Doing a Literature Review: Releasing the Research Imagination. SAGE Study Skills Series. SAGE. pp. xiii. ISBN 9781526423146.

[50] Ujjwal Maulik, Anirban Mukhopadhyay, "Simulated annealing based automatic fuzzy clustering combined with ANN classification for analyzing microarray data", Computers & Operations Research, Volume 37, Issue 8, 2010, Pages 1369-1380, ISSN 0305-0548.

[51] Gang Wang, Xinyi Zhang, Shiliang Tang, Haitao Zheng, and Ben Y. Zhao. 2016. Unsupervised Clickstream Clustering for User Behavior Analysis. In Proceedings of the 2016 CHI Conference on Human Factors in Computing Systems (CHI '16). ACM, New York, NY, USA, 225-236.

[52] R. Evans, B. Pfahringer and G. Holmes, "Clustering for classification," 2011 7th International Conference on Information Technology in Asia, Kuching, Sarawak, 2011, pp. 1-8.

[53] L. F. S. Coletta, N. F. F. d. Silva, E. R. Hruschka and E. R. Hruschka, "Combining Classification and Clustering for Tweet Sentiment Analysis," 2014 Brazilian Conference on Intelligent Systems, Sao Paulo, 2014, pp. 210-215.

[54] Abdul Wahab, Toh Guang Wen and Norhaslinda Kamaruddin, "Understanding Driver Behavior Using Multi-Dimensional CMAC", 6th International Conference on Information, Communications & Signal Processing, pp. 1-5, 2007.

[55] Abdul Wahab A, Quek C, Tan CK, Takeda K, "Driving Profile Modeling and Recognition Based on Soft Computing Approach. IEEE Transactions on Network, vol. 20/4. pp. 563-582, April 2009.

[56] Yi Lu Murphey, Robert Milton, Leonidas Kiliaris, "Driver's Style Classification Using Jerk Analysis", IEEE Workshop on Computational Intelligence in Vehicles and Vehicular Systems. CIVVS '09, pp. 23-28, 2009.

[57] Chuang-Wen You, Martha Montes-de-Oca, Thomas J. Bao1, Nicholas D. Lane, Giuseppe Cardone4, Lorenzo Torresani1, and Andrew T. Campbell, " CarSafe: A Driver Safety App that Detects Dangerous Driving Behavior using Dual-Cameras on Smartphones", ACM, 2009.

[58] Garima R. Singh and Snehlata S. Dongre, "Crash Prediction System for Mobile Device on Android by Using Data Stream Mining Techniques", Sixth Asia Modeling Symposium, 2012.

[59] Rygula A, "Driving Style Identification Method Based on Speed Graph Analysis", International Conference on Biometrics and Kansei Engineering, pp. 76-79, 2009.

[60] Maria E. Jabon, Jeremy N. Bailenson, Emmanuel Pontikakis, Leila TakayamaFacial. Expression Analysis for Predicting Unsafe Driving Behavior car driving simulator, IEEE Pervasive Computing, Vol. 10, pp.84-95, 2011.

[61] Duda, R. O., Hart, P. E., & Stork, D. G. (2000). Pattern classification, 2nd ed. Wiley

[62] J. Wenskovitch, I. Crandell, N. Ramakrishnan, L. House, S. Leman and C. North, "Towards a Systematic Combination of Dimension Reduction and Clustering in Visual Analytics," in IEEE Transactions on Visualization and Computer Graphics, vol. 24, no. 1, pp. 131-141, Jan. 2018.
doi: 10.1109/TVCG.2017.2745258

[63] Wikipedia contributors. "On-board diagnostics." Wikipedia, The Free Encyclopedia. Wikipedia, The Free Encyclopedia, 9 Apr. 2019. Web. 21 Apr. 2019

[64] Gagniuc, Paul A. (2017). Markov Chains: From Theory to Implementation and Experimentation. USA, NJ: John Wiley & Sons. pp. 1–235. ISBN 978-1-119-38755-8.

[65] I. N. Serbec, M. Strnad, and J. Rugelj, Assessment of wiki-supported collaborative learning in higher education. IEEE, 2010.

[66] Emerging Artificial Intelligence Applications in Computer Engineering : Real Word AI Systems with Applications in eHealth, HCI, Information Retrieval and Pervasive Technologies, edited by I. Maglogiannis, et al., IOS Press, 2007. ProQuest Ebook Central, https://ebookcentral.proquest.com/lib/bthbib-ebooks/action?docID=329926.

[67] H. Kusetogullari, Unsupervised Text Binarization in Handwritten Historical Documents Using k-Means Clustering. In: Bi Y., Kapoor S., Bhatia R. (eds) Proceedings of SAI Intelligent Systems Conference (IntelliSys) 2016, IntelliSys 2016, Lecture Notes in Networks and Systems, vol 16. Springer, Cham, 2018.

[68] H. Kusetogullari and A. Yavariabdi, Self-adaptive hybrid PSO-GA method for change detection under varying contrast conditions in satellite images, 2016 SAI Computing Conference (SAI), London, pp. 361-368, 2016.

[69] H. Kusetogullari and A. Yavariabdi, Unsupervised Change Detection in Landsat Images with Atmospheric Artifacts: A Fuzzy Multiobjective Approach, Mathematical Problems in Engineering, vol. 2018, Article ID 7274141, 2018.

[70] Pandit, Shraddha & Gupta, Suchita. (2011). A Comparative Study on Distance Measuring Approaches for Clustering. International Journal of Research in Computer Science. 2. 29. 10.7815/ijorcs.21.2011.011.

[71] Wikipedia contributors. "Euclidean distance." Wikipedia, The Free Encyclopedia. Wikipedia, The Free Encyclopedia, 1 Apr. 2019. Web. 6 Jun. 2019.

[72] Iffat A. Gheyas, Leslie S. Smith, Feature subset selection in large dimensionality domains, Pattern Recognition, Volume 43, Issue 1, 2010, Pages 5-13, ISSN 0031-3203, https://doi.org/10.1016/j.patcog.2009.06.009.

[73] Yan, Jun & Zhang, Benyu & Liu, Ning & Yan, Shuicheng & Cheng, Qiansheng & Fan, Weiguo & Yang, Qiang & Xi, Wensi & Chen, Zheng. (2006). Effective and Effcient Dimensionality Reduction for Large-Scale and Streaming Data Preprocessing. IEEE Transactions on Knowledge and Data Engineering. 18. 320-333. 10.1109/TKDE.2006.45.

[74] Kriegel, Hans-Peter, Peer Kröger, and Arthur Zimek. "Clustering high-dimensional data: A survey on subspace clustering, pattern-based clustering, and correlation clustering." ACM Transactions on Knowledge Discovery from Data (TKDD) 3.1 (2009): 1.

[75] G. Seif, "Principal Component Analysis: Your Tutorial and Code," Towards Data Science, 09-Nov-2018. [Online]. Available:

https://towardsdatascience.com/principal-component-analysis-your-tutorial-and-code-9719d3d3f376.

[76] A. keitakurita, "Paper Dissected: 'Visualizing Data using t-SNE' Explained," Machine Learning Explained, 14-Sep-2018. [Online]. Available: http://mlexplained.com/2018/09/14/paper-dissected-visualizing-data-using-t-sne-explained/

[77] Barbara Ann Kitchenham, David Budgen, and Pearl Brereton. Evidence-based software engineering and systematic reviews. Vol. 4. CRC press, 2015.

[78] "Introduction to K-means Clustering." [Online]. Available: https://www.datascience.com/blog/k-means-clustering

[79] F. Pedregosa et al. "Scikit-learn: Machine Learning in Python". In: Journal of Machine Learning Research 12 (2011), pp. 2825–2830.

[80] "NumPy — NumPy." [Online]. Available: https://www.numpy.org/. [Accessed: 09-Jun-2019].

[81] "Python Data Analysis Library — pandas: Python Data Analysis Library." [Online]. Available: https://pandas.pydata.org/. [Accessed: 09-Jun-2019].

[82] "Matplotlib: Python plotting — Matplotlib 3.1.0 documentation." [Online]. Available: https://matplotlib.org/. [Accessed: 09-Jun-2019].

[83] "Sympathy for data." [Online]. Available: https://www.sympathyfordata.com/. [Accessed: 09-Jun-2019].

[84] "Python Definition." [Online]. Available: https://techterms.com/definition/python. [Accessed: 09-Jun-2019].

[85] "Spyder Website." [Online]. Available: https://www.spyder-ide.org/. [Accessed: 09-Jun-2019].

[86] M. K. Pakhira, "A Linear Time-Complexity k-Means Algorithm Using Cluster Shifting," 2014 International Conference on Computational Intelligence and Communication Networks, Bhopal, 2014, pp. 1047-1051. doi: 10.1109/CICN.2014.220

Appendices

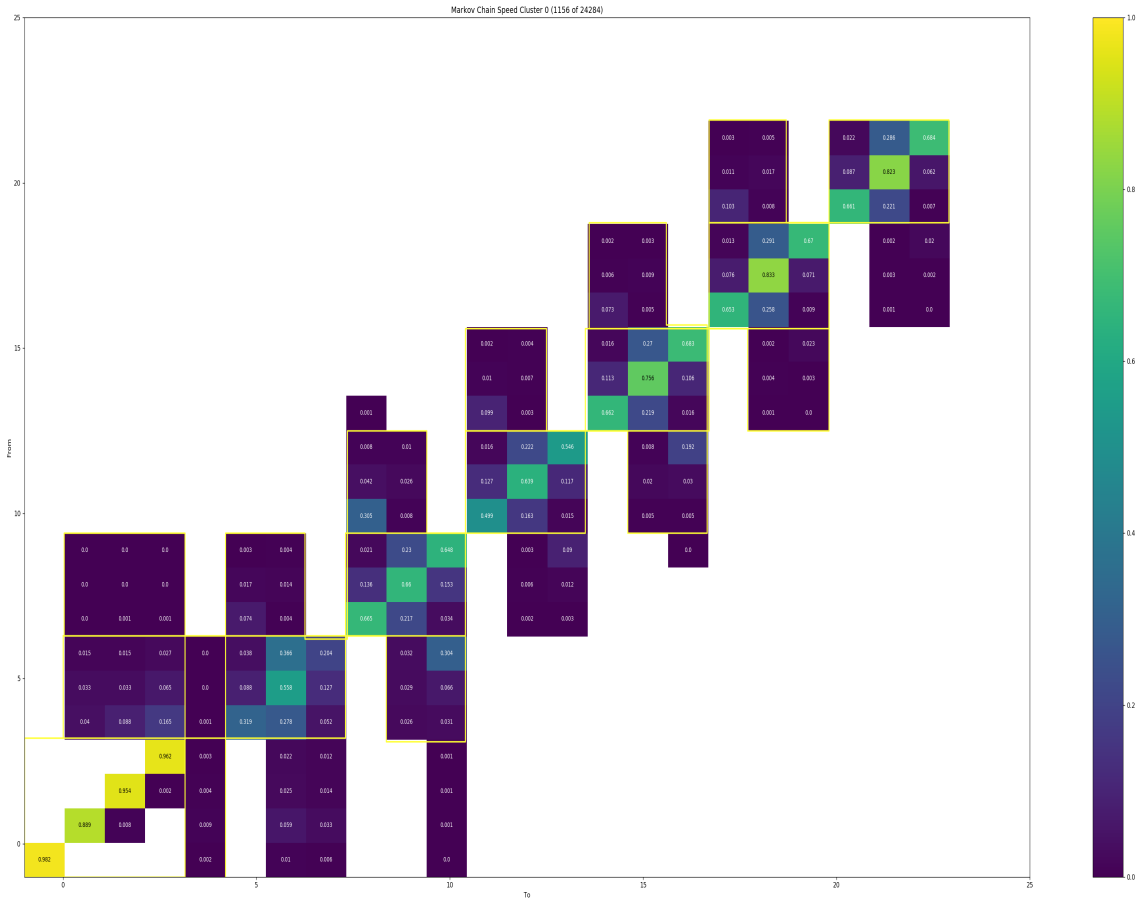

Figure 8.1: Analysis of cluster 0

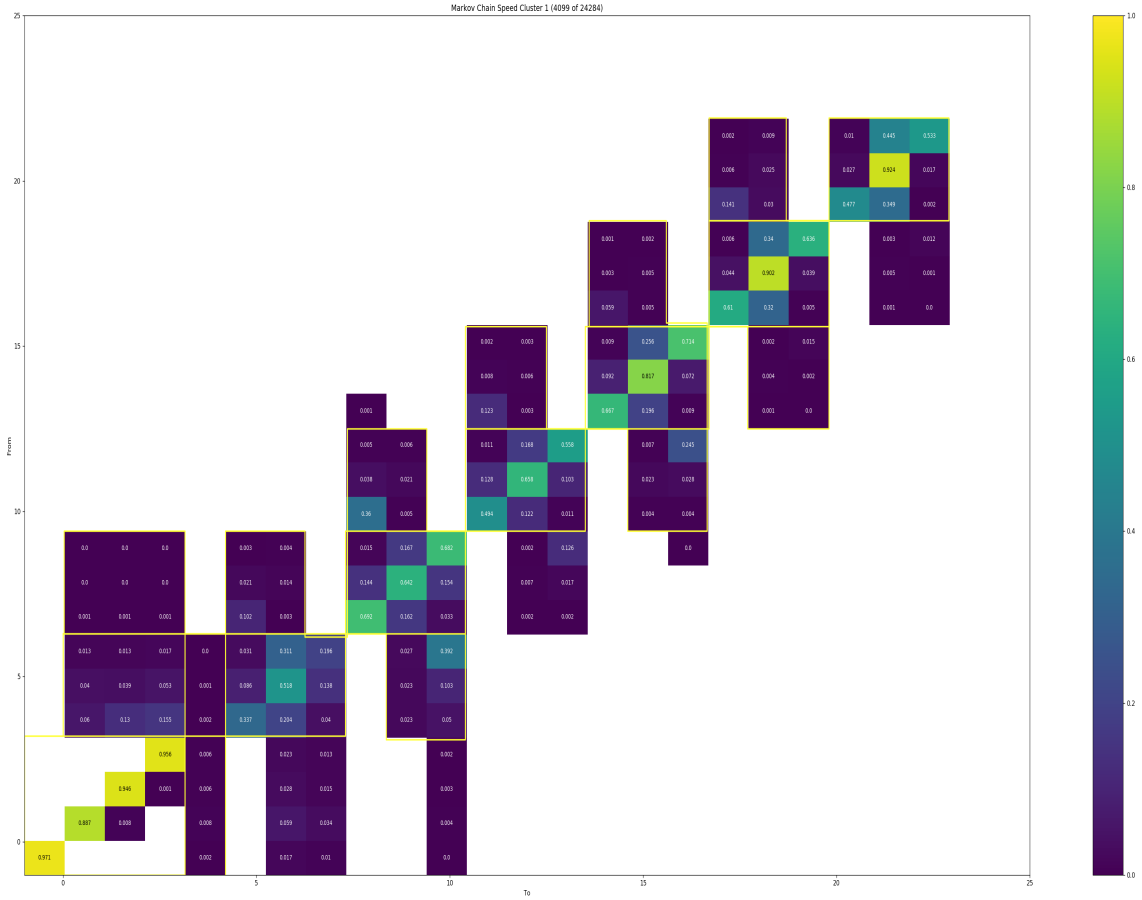

Figure 8.2: Analysis of cluster 1

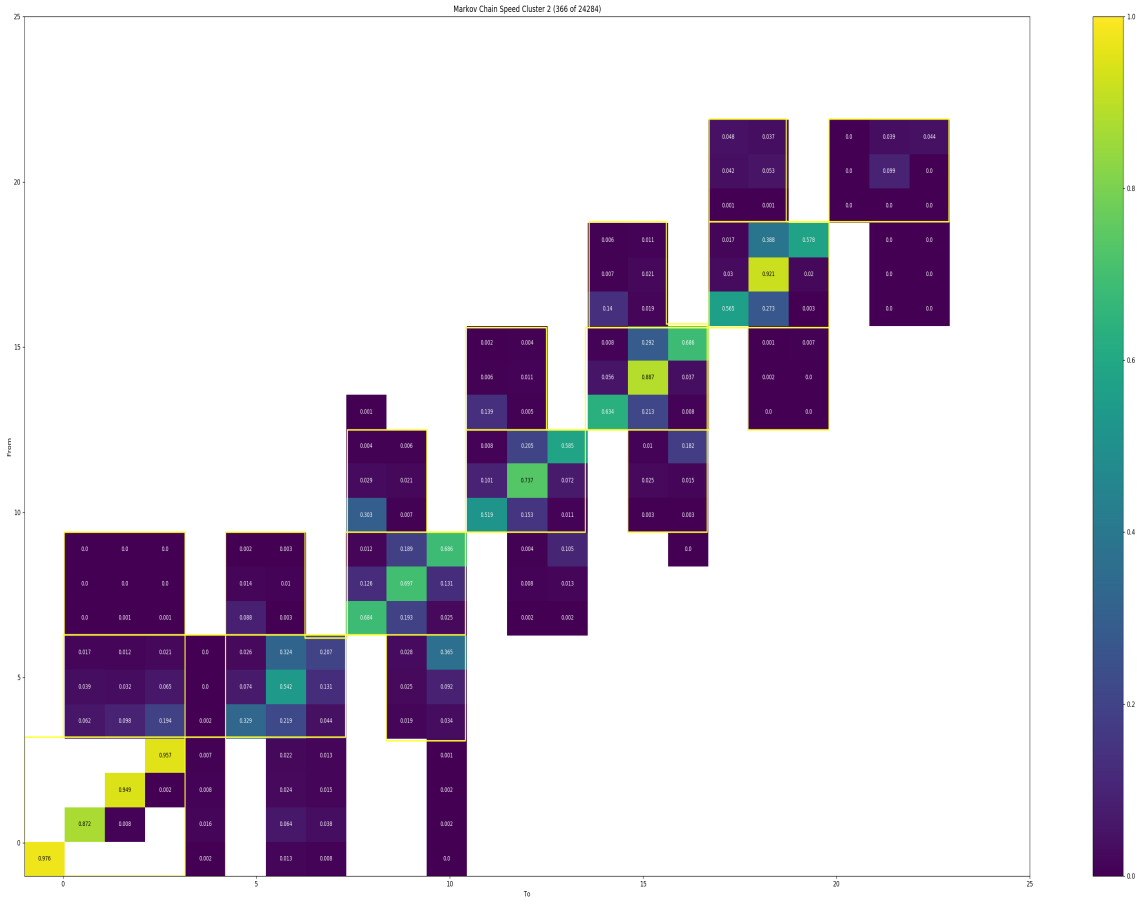

Figure 8.3: Analysis of cluster 2

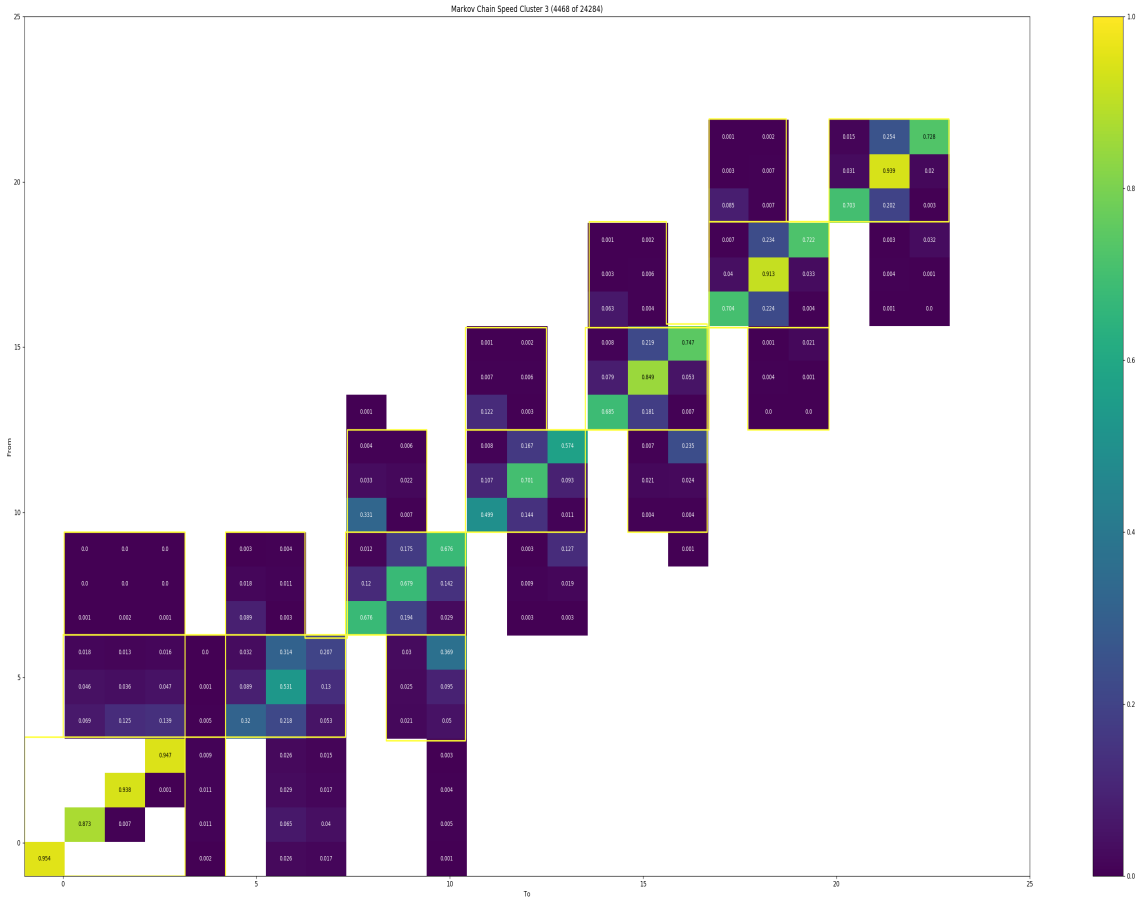

Figure 8.4: Analysis of cluster 3

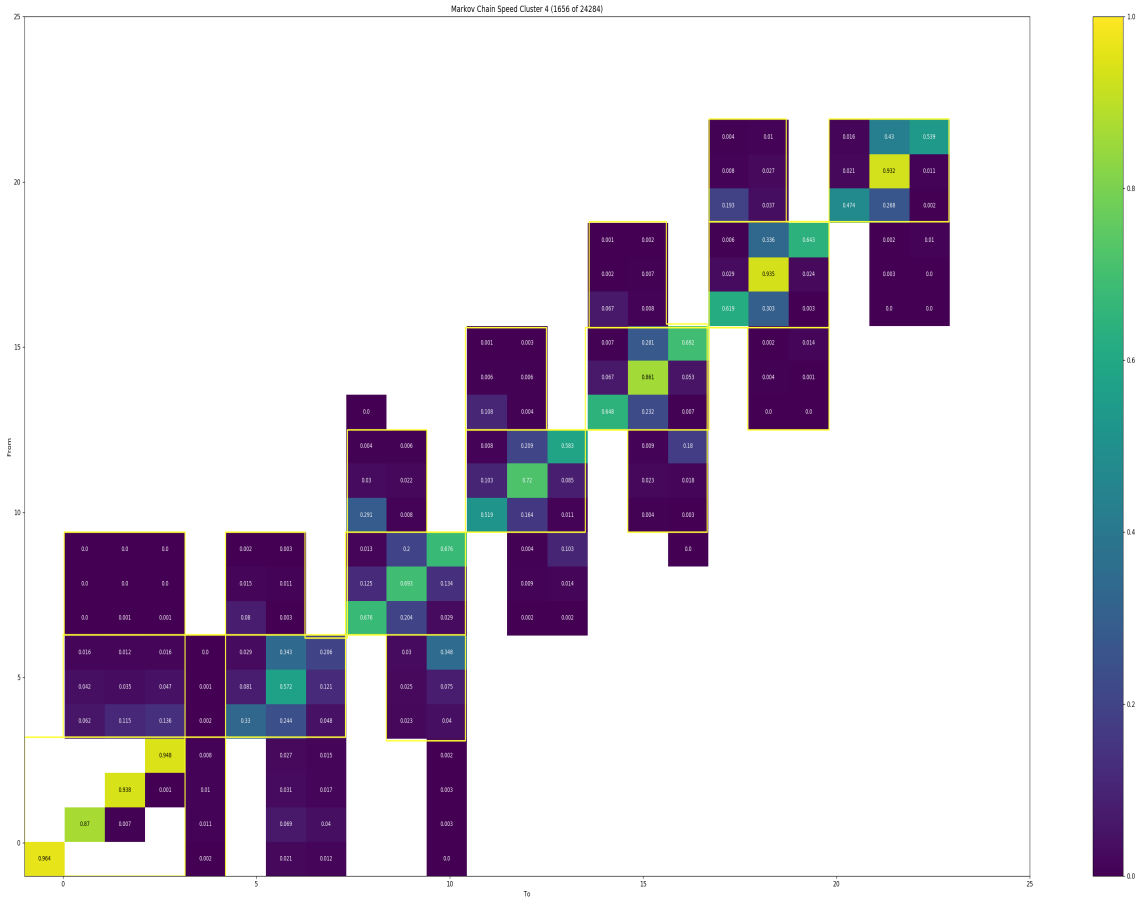

Figure 8.5: Analysis of cluster 4

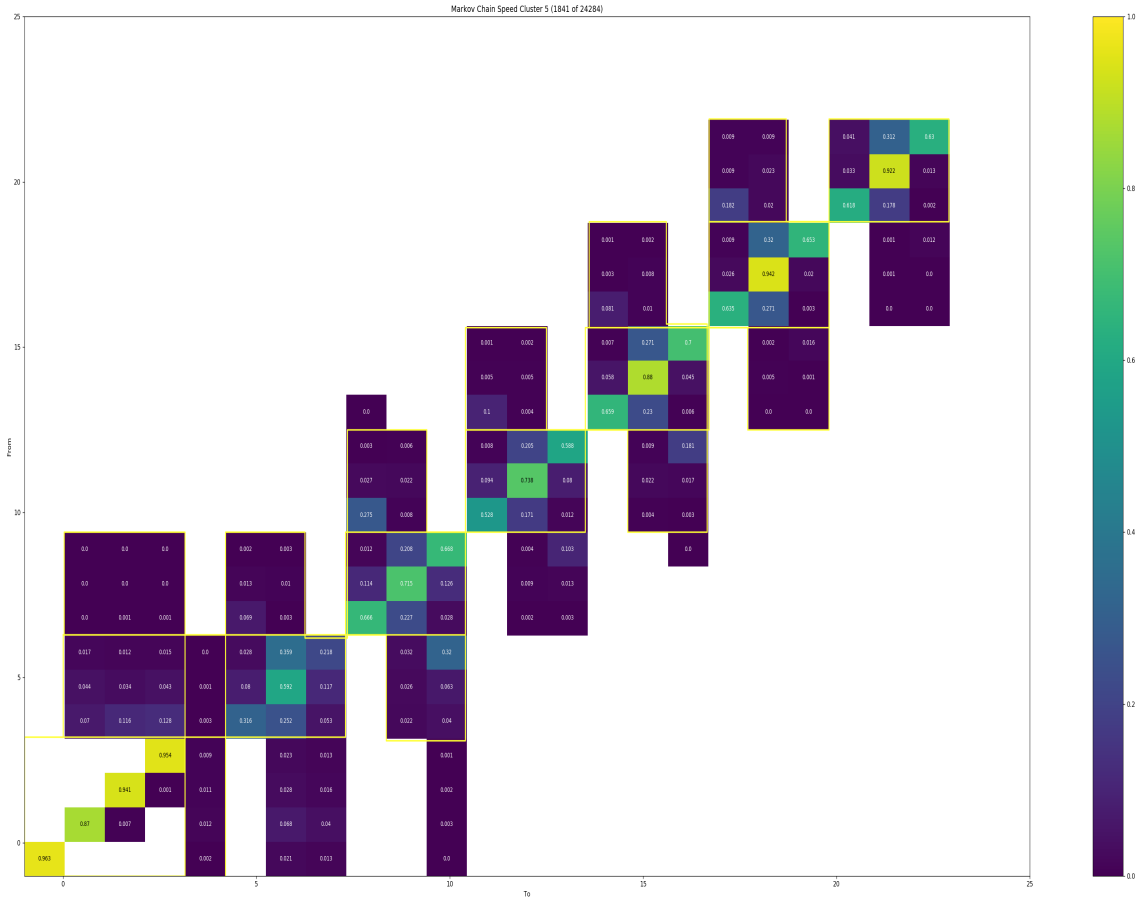

Figure 8.6: Analysis of cluster 5

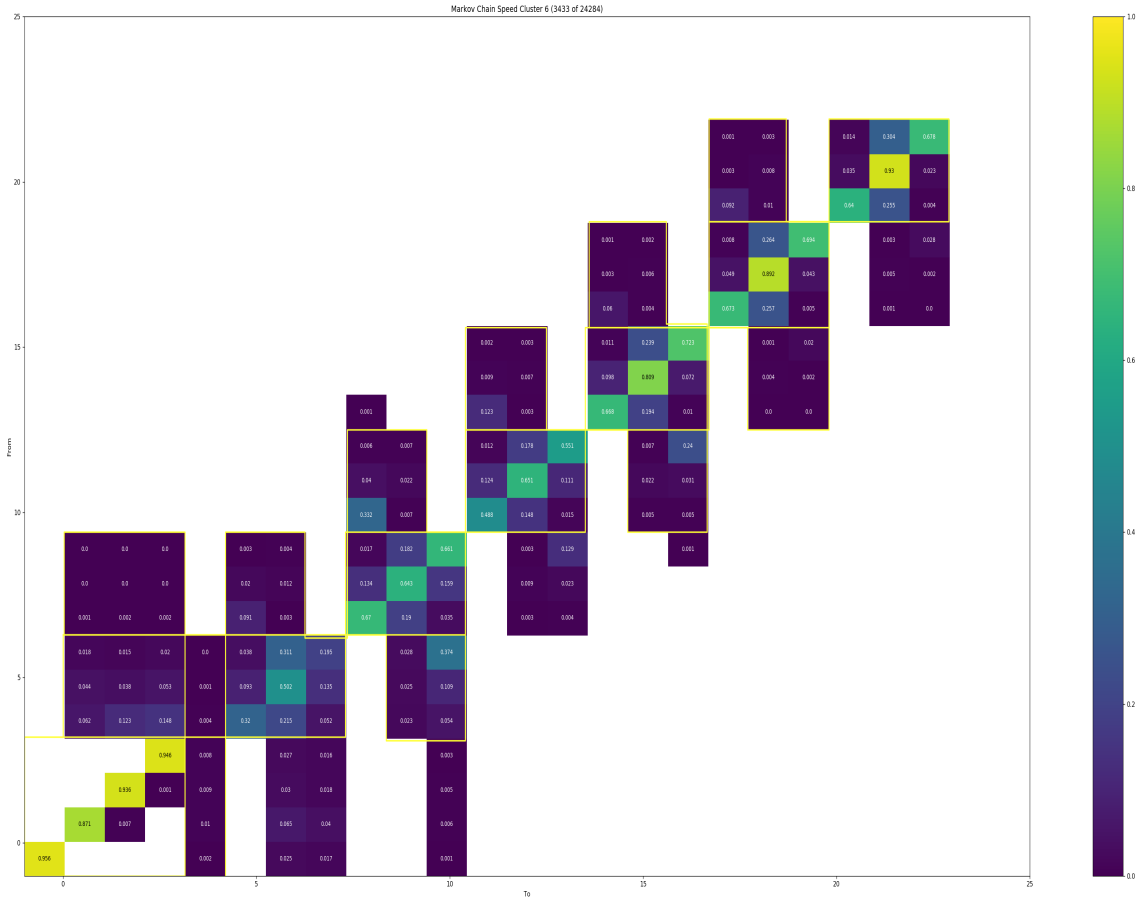

Figure 8.7: Analysis of cluster 6

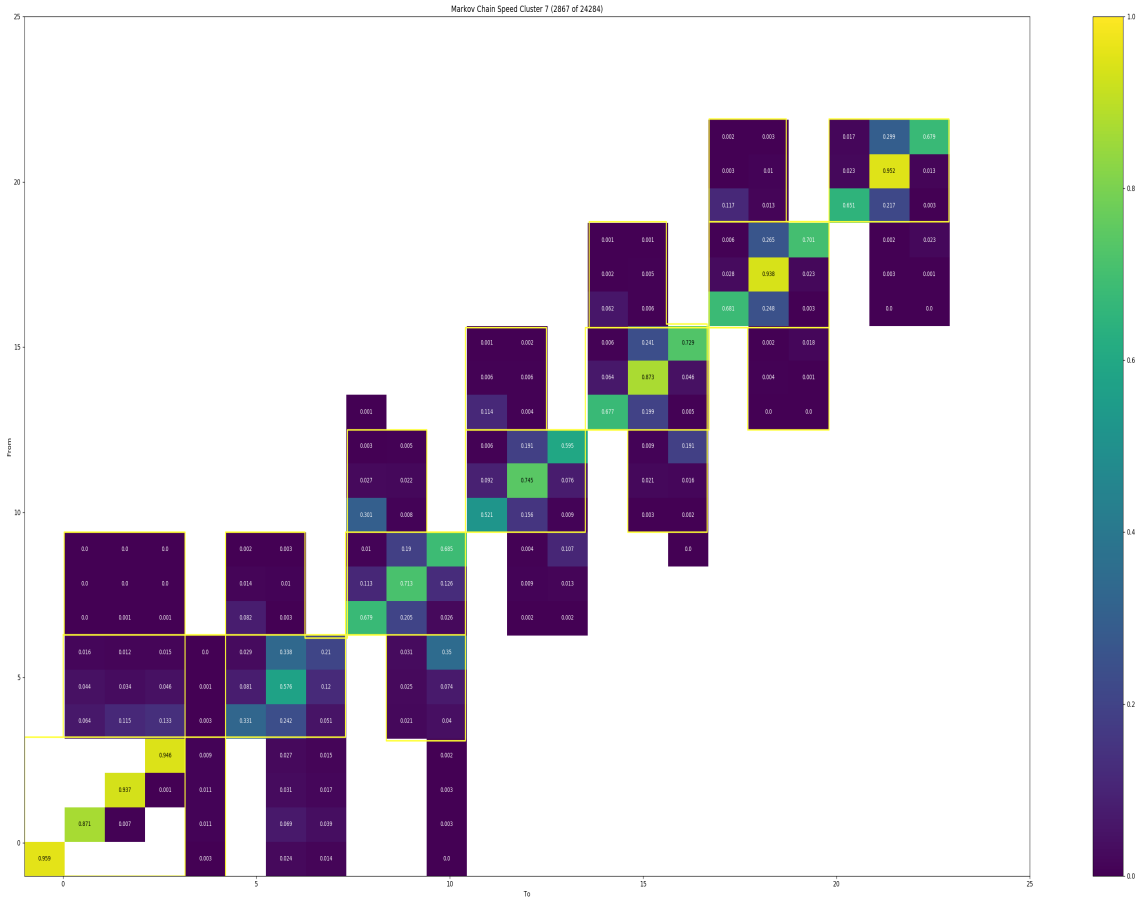

Figure 8.8: Analysis of cluster 7

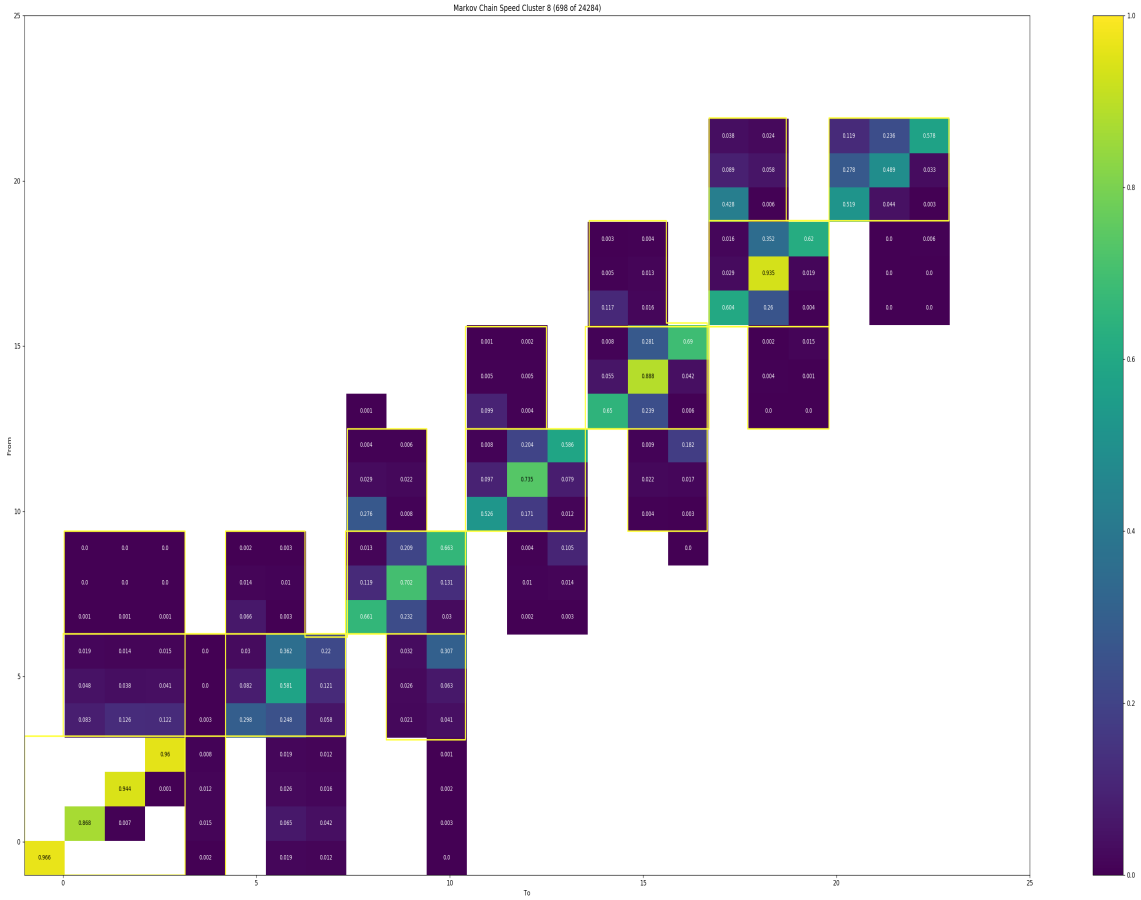

Figure 8.9: Analysis of cluster 8

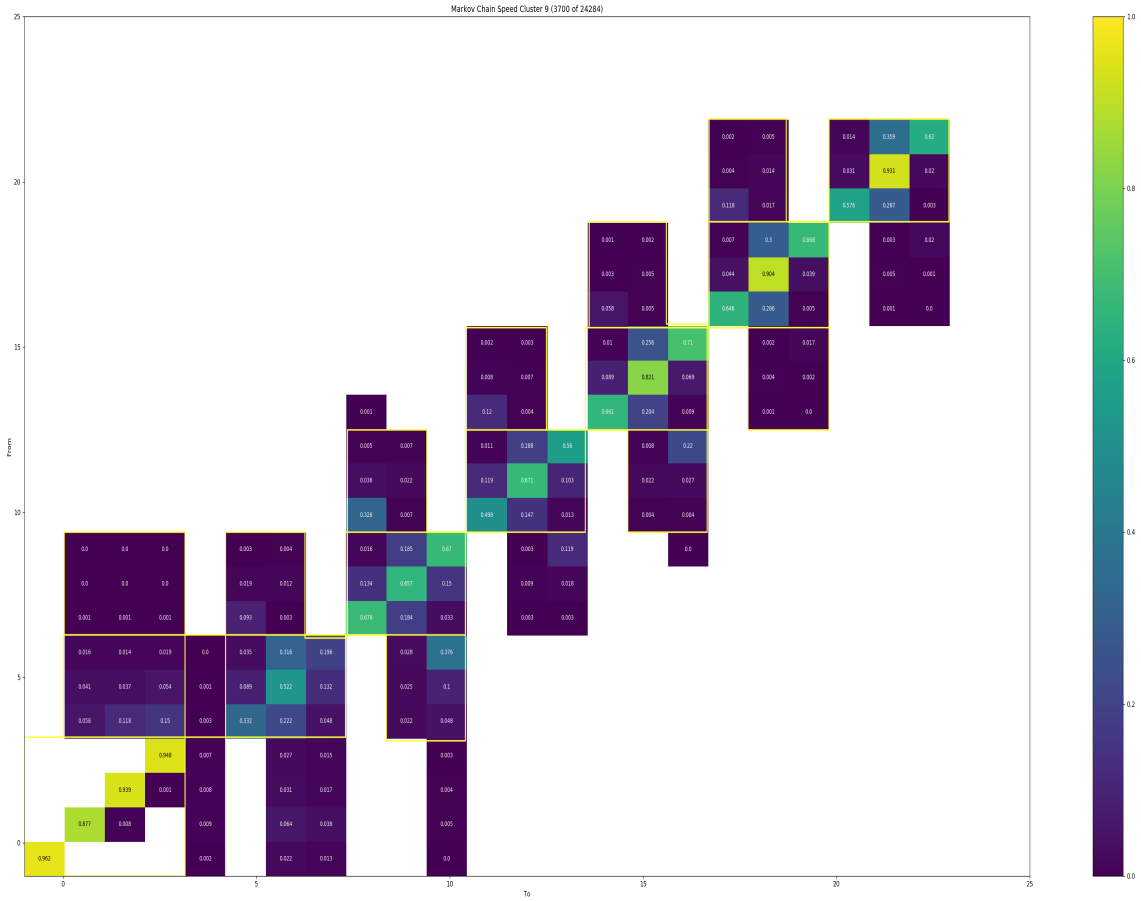

Figure 8.10: Analysis of cluster 9