CHALMERS
UNIVERSITY OF TECHNOLOGY

UNIVERSITY OF GOTHENBURG

Generating Representative Driving Cycles

A Comparative Analysis of Data Analytics Methods

Master's thesis in Computer Science and Engineering

Adam Eliasson

Felicia Hjalmarsson

Department of Computer Science and Engineering
CHALMERS UNIVERSITY OF TECHNOLOGY
UNIVERSITY OF GOTHENBURG
Gothenburg, Sweden 2024

Master's thesis 2024

Generating Representative Driving Cycles

A Comparative Analysis of Data Analytics Methods

Adam Eliasson

Felicia Hjalmarsson

UNIVERSITY OF
GOTHENBURG

CHALMERS
UNIVERSITY OF TECHNOLOGY

Department of Computer Science and Engineering
Division of Data Science and AI
Chalmers University of Technology
University of Gothenburg
Gothenburg, Sweden 2024

Generating Representative Driving Cycles
A Comparative Analysis of Data Analytics Methods
Adam Eliasson Felicia Hjalmarsson

Supervisor: Shirin Tavara, Computer Science and Engineering
Supervisor: Gabriel Angerd, Volvo Group
Examiner: Marina Axelson-Fisk, Mathematical Sciences

Master's Thesis 2024
Department of Computer Science and Engineering
Chalmers University of Technology and University of Gothenburg
SE-412 96 Gothenburg
Telephone +46 31 772 1000

Typeset in LaTeX
Gothenburg, Sweden 2024

Generating Representative Driving Cycles
A Comparative Analysis of Data Analytics Methods
Adam Eliasson
Felicia Hjalmarsson
Department of Computer Science and Engineering
Chalmers University of Technology and University of Gothenburg

Abstract

The transportation industry is rapidly evolving. Customer expectations and environmental sustainability demands are quickly shifting, making performance optimisation of Heavy-Duty Vehicles (HDVs) increasingly critical. An essential aspect of the optimisation process is having Driving Cycles (DCs) that are representative of real-world driving patterns for accurate vehicle verification and validation.

In this thesis, different advanced data analytics methods were implemented and evaluated on their ability to generate DCs representative of real-driving patterns of HDVs. The implemented methods belonged to three main categories explaining the general technique of how DCs are constructed: Speed Acceleration State (SAS) methods, Micro-Trip (MT) methods, and Kinematic Segment (KS) methods.

To quantitatively assess method effectiveness, two main metrics were used: Characteristic Parameters (CPs), and Speed Acceleration Probability Distribution (SAPD). The CPs describe different statistical characteristics of driving behaviour which were compared between generated cycles and the operational data. Additionally, a comparison between generated cycles and the Vehicle Energy Consumption Calculation Tool (VECTO) was also made to provide a performance baseline. CPs were evaluated using Relative Difference (RD), while SAPD was evaluated with RD and Earth Mover's Distance (EMD). EMD measures distribution dissimilarities, and its addition to the evaluation offers a more reliable SAPD assessment than what has been done in previous research.

The results showed that all implemented methods outperform the VECTO baseline both in terms of CP and SAPD representativeness, highlighting the need for fine-tuned cycles. The SAS and MT methods demonstrated superior performance compared to the KS methods. Especially in terms of SAPD representativeness and computational efficiency. While the KS methods showed significant limitations, the SAS and MT methods achieved highly promising CP and SAPD representations of the operational data. The SAS and MT methods were concluded as viable methods for DC generation and are the methods suggested to continue researching.

The thesis aims to serve as a DC generation framework that future researchers can follow, including detailed descriptions of all necessary methodological steps. The framework details a systematic approach for creating representative DCs adaptable for any driving profile or vehicle type.

Keywords: Driving Cycles, Machine Learning, Data Analytics, Heavy-Duty Vehicles, Master Thesis, Chalmers University of Technology.

Acknowledgements

First of all, we would like to express our sincerest gratitude to our supervisor Shirin Tavara for all her help and guidance throughout the thesis. A special thanks also to Gabriel Angerd, our supervisor at the Volvo Group, for his tremendous investment in our work and his daily support.

Thank you to Chalmers and Anthony Norman for the insightful writing seminars, as well as Lisa Eskilson and Alexander Lager Carvalho for their constructive feedback during the opposition(s). Thank you also to Marina Axelson-Fisk for her dedication and efforts as the thesis examiner. Finally, we would like to thank Johan Gustafsson, General Manager Customer Feature Data Analytics at the Volvo Group, and the Volvo Group as a whole for welcoming us to be part of their team and providing all the resources necessary to complete this thesis.

Adam Eliasson & Felicia Hjalmarsson, Gothenburg, 2024-06-28

Contents

List of Figures

List of Tables

1

Introduction

In the fast-evolving landscape of the transportation industry, improving the performance of Heavy-Duty Vehicles (HDVs) is becoming increasingly critical. As customer expectations continue to shift, optimising vehicles to meet these new expectations is essential. A key component of that is the verification and validation of the vehicles before they enter production. A crucial part of that process is to have representative Driving Cycles (DCs) [1].

To address the challenge of generating synthetic DCs representative of real-world driving patterns, this thesis researches the use of advanced data analytics. Various statistical analysis and Machine Learning (ML) techniques are applied and compared in their efficacy to generate representative DCs based on HDV operational data. Additionally, the thesis investigates how cycle representativeness is measured and quantified. The research efforts aim to provide a framework for future work in the field, comparing the suitability and effectiveness of different advanced data analytical techniques.

1.1 Driving Cycles

DCs are collections of data points representing the speed of a vehicle over time or distance [2]. These cycles are used to simulate a variety of real-world driving conditions to assess vehicle performance. Cycles are also commonly used for legislation and regulation of vehicle gas emission standards, for example, the Vehicle Energy Consumption Calculation Tool (VECTO) which was introduced as a part of the EU's CO_2 legislation in 2019 [3]. VECTO is utilised to simulate gas emission levels along with fuel consumption from HDVs, serving both as vehicle performance comparison and the regulation of gas emission standards.

Broadly, DCs are classified into two types: modal and transient. Modal cycles are characterised by constant driving stages such as cruising, idling, and constant acceleration or deceleration. A commonly referred to example of a modal cycle is the New European Driving Cycle (NEDC) [4]. The NEDC was developed to assess the emission level and fuel economy of passenger cars.

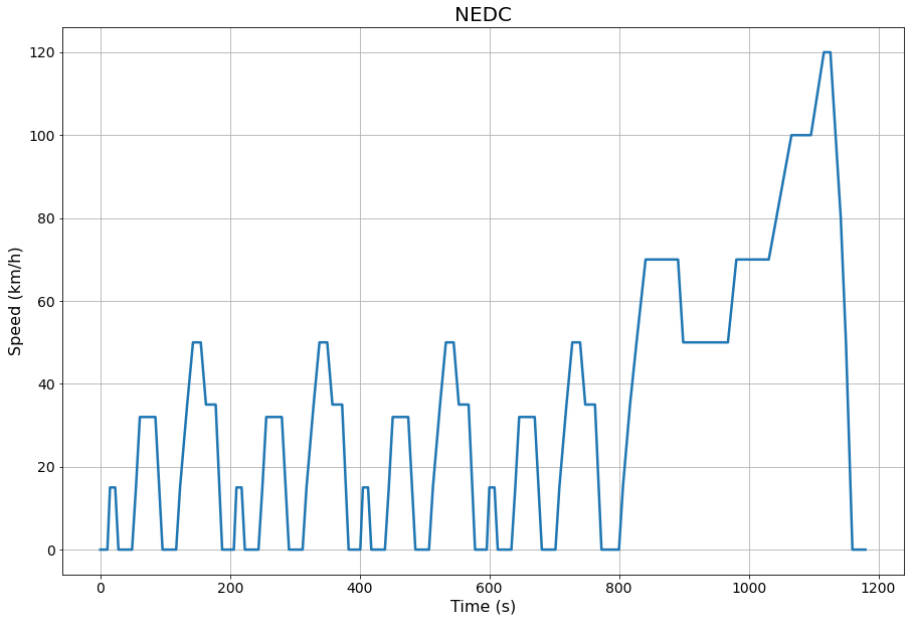


Figure 1.1: The New European Driving Cycle. The cycle consists of constant cruising, idling, acceleration and deceleration.

As is illustrated in Figure 1.1, the NEDC cycle is built by combining a set of the four different constant driving stages idling, constant acceleration, constant speed, and constant deceleration. Together, the states construct a complete driving cycle of twenty minutes, aiming to mimic a typical driving pattern. However, given the simplicity of the constant driving stages, the cycle may oversimplify driving tendencies and hence not capture some of the complexities of real-world driving. This is a limitation with modal cycles, which in many cases can result in underestimations of emissions and fuel consumption [2]. Due to this, modal cycles are viewed as unrealistic and are now generally regarded as outdated.

To avoid the limitations of the modal cycles, the industry has transitioned into designing the second driving cycle classification, transient cycles. Transient cycles are typically based on real driving data and are characterised by a wider range of speed and acceleration variations. Transient cycles thus capture more nuanced transitions in speed and acceleration, offering a more accurate representation of real driving conditions [2]. The Worldwide harmonized Light vehicles Test Cycles (WLTCs) are transient and have replaced the NEDC for type approval testing of light-duty vehicles since 2017.

The WLTCs have been developed on real-world driving data collected from different regions globally, designed to reflect diverse driving environments [5]. Based on the region and driving capabilities of vehicles the WLTCs have been segmented into different classes. The cycle of each class is hence representative of a certain vehicle population. Among these, one of the classes relevant for most passenger cars is the WLTC-3.

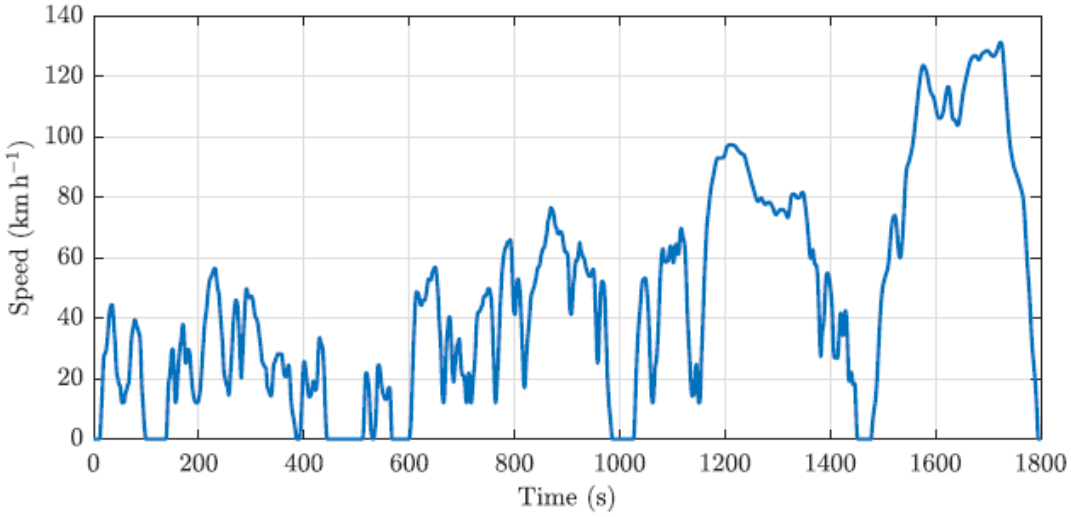


Figure 1.2: WLTC class 3 driving cycle [6].

Figure 1.2 illustrates the WLTC-3 cycle. The cycle highlights the more dynamic speed and acceleration transitions. These transitions more accurately represent real driving behaviour compared to the NEDC, and thus the simulation of transient cycles is a superior method for evaluating vehicle performance compared to modal cycles [2].

1.2 Data Analytics

Advanced data analytics is used for transforming and analysing data to gain valuable insights across various industries, including the automotive industry [7]–[9]. Data is often transformed into a standardised format which is then followed by some statistical analysis and the application of ML models for prescriptive and predictive modeling. These techniques are utilised to identify complex patterns and relationships within the data. The process is essential for making data-driven decisions that foster innovation and performance optimisation.

The role of advanced data analytics has become especially useful within DC generation. The complexity and variability of driving patterns call for more advanced methods that can analyse those patterns and capture meaningful insights. Advanced data analytics methods are thus applied on extensive datasets generated from real driving experiences [10]–[15]. This enables the generation of transient cycles that offer more realistic representations of real-world driving behaviors compared to modal cycles, narrowing the accuracy gap between standardised testing cycles and actual driving experiences [2].

Data analytics is also useful due to its adaptability to varying input data. As driving behaviors evolve and new data becomes available, existing DCs can be updated to reflect these changes. This capability of data analytics methods is essential for maintaining the validity and reliability of DC development over time. Thus, data

analytics tools are robust to changes in technology, infrastructure, or driving patterns. It allows cycle generation to be adapted to reflect changes and developments.

Advanced data analytics has made the generation of DCs for HDVs a more accurate and meaningful process. It enables the generation of data-driven transient DCs that more accurately reflect real-world driving behaviours. Furthermore, it ensures that cycles remain aligned with continuously evolving driving conditions. As a result, advanced data analytics enhances the reliability and relevance of vehicle performance assessments within the transportation industry.

1.3 Thesis Motivation

To unlock the full potential of HDVs, functionality, and performance evaluation are critical. However, there exists a discrepancy between real-world customer experiences and the DCs used for vehicle validation. This gap is acknowledged in many works of literature [16]–[18], raising concerns of lacking vehicle performance and reduced customer satisfaction. With growing customer expectations and increasing regulatory demands for more environmentally sustainable and efficient transportation, it is becoming increasingly critical to bridge this gap to better align vehicle validation with real-world customer experiences.

This thesis is motivated by the need to bridge this gap, thereby contributing to the improvement of the reliability and performance of HDVs. By generating synthetic DCs representative of real-world driving, this research aims to make a significant contribution towards achieving HDVs that are better tailored for customers while also improving their environmental sustainability.

Despite considerable previous research conducted on generating synthetic DCs, there remains an absence of a comprehensive comparison of these methods and a robust framework for their evaluation. Thus, the motivation for this thesis is twofold: (1) to bridge the existing gap in synthetic DC generation, and (2) to contribute to the field through an in-depth comparison of advanced data analytics methods. The thesis aims to establish a systematic approach for validating and comparing the effectiveness of different methodologies in generating representative cycles. This dual focus seeks to increase the relevance of the thesis by offering both practical benefits to HDV production and significant contributions to literature for future research.

1.4 Thesis Aim

In this section, the research questions and goals of this thesis are covered.

Research questions:

- How do different advanced data analytics methods compare in their accuracy for generating synthetic DCs that represent real-world driving patterns for HDVs?

- How can a systematic framework be implemented to generate, evaluate, and validate synthetic DCs from real-world driving data using advanced data analytics methods?

Project goals

- Make a suitability assessment of advanced data analytics methods for synthetic DC generation. The suitability criteria for each method are based on their accuracy in replicating real-world driving patterns and their efficiency and scalability for processing large datasets.
- Implement and hyperparameter tune the methods determined to be suitable. The primary goal is to generate synthetic DCs with the highest possible accuracy representing real-world driving patterns to maximize their usefulness in vehicle development. Simultaneously, it is also key to ensure a reasonable level of computational efficiency for the method to be suitable for practical use.
- Identify the method that generates DCs most accurately representing real-world driving patterns for HDVs. The evaluation is based on three main criteria: accuracy, computational demand, and overall performance. Success in the thesis is measured by the following criteria:

  1. Superior accuracy. Identifying a method that demonstrates significant accuracy superiority in reflecting real-world-driving patterns.
  2. Satisfactory computational efficiency. Identifying a method that shows satisfactory computational efficiency compared to other methods without significantly compromising accuracy.
  3. Superior accuracy and computational efficiency. Identifying a method that fulfills both criteria 1 and 2.

  Fulfilling these criteria would align with the aim of the thesis to improve the utility of synthetic DC generation for HDVs. While fulfilling criterion number 3 would be the best outcome, fulfilling criterion number 1 alone would be a significant achievement and great success in the thesis.
- Create a structured and comprehensive documentation of the cycle generation, evaluation, and validation process. Regardless of the specific outcome in achieving any of the criteria above, this thesis should provide a comprehensive analysis of the performance capabilities of various advanced data analytics methods. This analysis is meant to serve as a valuable foundation for future research, offering insights on how and why certain methods may outperform others.

1.5 Scope

This thesis acknowledges certain limitations in its scope and methodology. Limitations are mainly due to time and resource constraints. The thesis is limited to considering

a selected set of data analytics methods. While the existence of other potentially effective methods is recognised, the scope is limited to methods that have shown to be most relevant in the research area and are feasible for the research scope.

More limitations concern the data scope. While the aim is to incorporate a diverse set of data parameters to make the synthetic cycles represent as many dimensions of driving as possible, certain limitations had to be set considering the time and resource availability for this thesis. Therefore, the analysis is primarily based on speed and acceleration parameters, as these are heavily used in previous research, see Chapter 2. Making an extensive analysis of methods utilising these parameters is thus most important.

Furthermore, the findings and models developed in this thesis are specifically tailored to HDVs. As such, the results may not be directly applicable or transferable to other vehicle types without modifications and further research. Moreover, the cycles that are generated will be tailored for the long-haul driving profile which is of great interest in the HDV market, and especially for the Volvo Group who is the industrial partner of this thesis.

1.6 Thesis Outline

Chapter 1 consists of an introduction to DCs and the differences between modal and transient cycles. It also introduces the topic of data analytics and its relevance for data-driven decision-making. Moreover, the chapter includes the thesis motivation, research questions, goals, and scope.

Chapter 2 reviews related work within the field. The section focuses on existing research on various advanced data analytics methods for synthetic DC generation and evaluation metrics used to measure representativeness.

Chapter 3 moves into introducing and describing concepts that are important to sufficiently grasp when reading the thesis. It splits concepts into four main categories: HDVs, advanced analytics, performance evaluation, and cycle construction modeling. All material in this section are key aspects related to the thesis work and hence they are thoroughly described here to aid the reader.

Chapter 4 details the methodology followed during the thesis. It covers all key aspects including literature study, data collection and preparation, model implementation, and model evaluation. In Chapter 5, the results of the model evaluations are detailed. The results are then discussed in Chapter 6 along with a discussion regarding ethical and sustainability considerations.

Finally, in Chapter 7, suggestions for future work in the field are presented and the thesis is concluded with a summary of key aspects and goals achieved.

2

Related Work

This chapter reviews methods for generating representative DCs from existing research. It outlines the most common approaches in the field and the metrics used for evaluating their representativeness. This overview forms the foundation of this study and prepares for an in-depth examination and comparison of selected methods in later chapters.

2.1 Micro-Trip Based Methods

In the field of DC development, methods based on Micro-Trips (MTs) are commonly found in the literature [14], [19], [20]. These methods divide driving data into MTs, which are subtrips that start when a vehicle begins moving from zero speed and end when it returns to a standstill. The segmentation makes it possible to analyse driving behavior in more detail within these separate segments.

MTs are typically defined to start and end at zero speed. However, some researchers have experimented with using other speed thresholds to mark the start and end of an MT [1]. The motivation for this is to make the MTs better suited to different driving conditions and target profiles since long MTs can lose their representativeness. Adjusting this threshold could therefore improve the relevance and accuracy of the analysis.

The core of MT based methods is to analyse these MTs, select a subset based on various criteria, and link them together to construct a complete DC [20]. Among the several techniques that are used for DC construction using MTs, clustering MTs based on similarities in speed and acceleration is the most common one. The clustering approach helps to find distinct patterns reflecting different driving behaviors. The goal after the clustering becomes to carefully select the MTs that best capture these driving behaviors and combining them to construct a comprehensive DC that is an accurate representation of real-world driving.

The selection techniques used across various MT methods are very similar. Predominately, there are three existing strategies: random selection, best incremental (incrementally searching for and selecting a micro-trip with certain characteristics), and a hybrid combination of both [21].

2.2 Markov Chain Based Methods

One of the most adopted methods for DC generation is the Markov Chain (MC) method. The logic behind the method is based on the Markov property, each state in a DC only depends on the state immediately before it. The Markov property makes the statistical modeling of driving behavior quite simple and the method has proven to be well-performing in most prior research [10], [12], [15], [19].

Most studies, including those by Jia et al. [12] and Yang et al. [15], use driving parameters speed and acceleration which are discretized into independent states. The next step is to calculate the probability of transitioning from one state to another by analysing the sequence of states observed in actual driving data. These probabilities are then aggregated to form the state transition matrix which outlines all potential transitions between states.

To construct a DC, Monte Carlo simulation is commonly applied [10], [12], [19]. According to the probabilities in the state transition matrix, states are sampled to generate a sequence that mirrors realistic driving patterns. The probabilities in the transition matrix are histogram-based, meaning that the sampling of states with the MC method is limited to discrete values. Despite the effectiveness of the MC method, the discrete sampling might not capture the full spectrum of driving behaviors, pointing out some possible improvements [22].

Recently, alternative approaches have been explored to address certain limitations with MC DC generation. One study by Yang et al. [22] integrated Metropolis-Hastings Sampling (MHS) with Kernel Density Estimation (KDE) to estimate state transition probabilities in an attempt to overcome the limitations of discrete histogram-based estimations. According to their findings, the MHS method's ability to sample any value from a given Probability Density Function (PDF) more accurately captures the random variations in vehicle states [22], making the modeling of state transitions closer to actual driving data.

Most MC applications for DC generation focus on a 2D context, using speed and acceleration, due to the complexity involved in extending the method to three or more dimensions [19]. However, one study addressed this challenge by incorporating road slope, adapting the traditional 2D MC approach [12]. Road slope data was added with a matching algorithm to the conventional 2D framework, presenting an efficient and practical solution to incorporate a third parameter without the need for a more complex and time-consuming 3D MC model.

2.3 Machine Learning Based Methods

There is a growing trend in applying ML techniques to the development of DCs. Most of these applications have been focused on methods involving the clustering of MTs, with K-means being a common technique for this analysis [14], [19].

In addition to these methods, a study explored a hybrid clustering algorithm combining K-means and SVM [23]. Instead of using MTs, this method partitioned the driving

data into Kinematic Segments (KSs) representing different driving behaviours such as idling, acceleration, deceleration, and cruising. Eight Characteristic Parameters (CPs) were considered: maximum speed, minimum speed, average speed, standard deviation of speed, maximum acceleration, maximum deceleration, average acceleration, and standard deviation of acceleration. To manage the data complexity, Principal Component Analysis (PCA) was applied to reduce the dimensionality, focusing the analysis on the most important features. K-means clustering was then performed to group these kinematic segments into preliminary clusters. These clusters were further refined using SVM by leveraging its capability to handle complex nonlinear relationships.

Another study employed a similar structure but used different algorithms [20]. This approach clustered MTs instead of KSs and applied kernel Principal Component Analysis (kPCA) for dimensionality reduction and Random Forest (RF) to improve upon K-means clustering results. Unlike ordinary PCA which is based on the covariance matrix of the observed data and struggles with nonlinear patterns, kPCA uses kernel functions to effectively handle both linear and nonlinear patterns. This makes kPCA particularly effective for complex feature extraction. Additionally, this study analysed fourteen CPs instead of eight for a deeper insight into driving behaviors. These modifications demonstrates the breadth of ML techniques and highlights the importance of exploring and comparing different algorithms in order to find the most effective way to analyse complex data and uncover meaningful patterns.

The construction of the candidate DCs differed between the two approaches, each with its own criteria for how to select and combine the driving segments. The SVM hybrid clustering method prioritised segments close to cluster centers and ensured transitions were smooth by keeping speed differences between consecutive segments under 1 km/h. The duration of each segment class was determined using time proportions from the operational data to ensure the cycle reflected the observed distribution of driving behaviors. The final candidate DC was assembled by carefully selecting segments based on their representativeness, measured by their proximity to cluster centers, and ensuring continuity until the desired cycle length was reached.

Conversely, the RF method selected MTs based on their Pearson Correlation Coefficient (PCC) with the overall cluster characteristics, focusing on statistical representativeness [20]. Each selected MT was thus ensured to be statistically representative of the average cluster MT. The number of MTs selected from each cluster to be included in the generated cycle was determined by the clusters relative duration in the operational data and the duration of the selected MTs. With these steps, candidate DCs were constructed by splicing the carefully selected MTs together.

Expanding on the variety of ML techniques used in DC development, another study employed a Decision Tree Regressor (DTR) to incorporate more diverse parameters beyond the commonly used speed and acceleration into the analysis [11]. The added parameters included engine torque and fuel economy. By setting energy consumption as the regression target, the DTR ranks trips and determines the most representative of real-world driving behavior. This trip is selected as the resulting DC. The approach

differs from the previously mentioned segment-based methods by selecting a single trip, and by incorporating more parameters it aims to improve the accuracy and representativeness of the DC.

2.4 Evaluation Metrics for Driving Cycle Representativeness

Vehicle driving conditions can be described and quantified using CPs. These parameters are the criteria for evaluating whether a driving cycle can realistically represent vehicle behavior. Previous studies have used a wide array of CPs for constructing and evaluating representative DCs. Some of these include average speed, standard deviation of speed, average positive acceleration, along with the proportions of time spent accelerating, decelerating, cruising, and idling [12], [24].

To assess how well the generated cycles align with real-world data in terms of the selected parameters, the metric Relative Difference (RD) is widely used [10]. RD provides a quantitative measure of the similarity between the generated DCs and actual driving patterns. For each CP considered, an RD value is computed. Since most methods utilised for DC generation are stochastic processes and produce varying results in each run, a CP threshold value is usually pre-established to facilitate the selection of representative cycles from the generated candidates. Only cycles where all CPs fulfill this threshold will be selected as representative DCs [19]. The average RD across all CPs gives an overall measure of cycle representativeness, with acceptable differences typically ranging between 5% and 15%.

The Speed Acceleration Probability Distribution (SAPD) is a common alternative way of describing driving patterns and is also used for evaluating the representativeness of driving cycles. Usually visualised in a three-dimensional plot, SAPD measures the probability of the cycle being in certain speed and acceleration states. The similarity between the SAPD of a candidate cycle and actual driving data is evaluated using criteria such as the RD [19] or the smallest sum of squared differences (SSD) [25], aiding in the assessment and selection of representative DCs.

The choice of evaluation metrics ultimately depends on the specific research goals, such as accurately measuring fuel consumption. For instance, one study [22] used regression analysis to identify which CPs significantly impact fuel consumption. To evaluate the cycle, the study compared simulated fuel consumption rates derived from the generated DCs against real-world data. This comparison ensures that the DCs accurately reflect real-world vehicle energy usage, thereby validating their reliability.

3

Theory

This chapter presents the necessary theory to give readers sufficient technical depth to understand and follow the work done in the thesis. This is split into four main aspects: those related to HDVs, advanced analytics methods, performance evaluation, and cycle construction modeling.

3.1 Driving Profile Segmentation

HDV driving activities are commonly categorised into distinct 'driving profiles', a term paralleled by 'driving vocations' as identified in prior research [11]. The driving profile describes the HDVs primary operational tasks and typically indicates what type of routes it frequently drives. Among these categories are long-haul, regional-haul, and local distribution, yet the complete categorisation of HDVs spans a much broader spectrum of profiles.

There are typically significant variations in hardware specifications and operational environments across these different profiles. To analyse any specific profile there is a need to segment the data, the segmentation is essential for accurately modeling the varying and diverse driving patterns. By employing advanced data analytics on a dataset, DCs can be specifically tailored to represent specific characteristics of targeted vehicle populations, thereby improving the precision of individual cycles and simulations. Profile-tailored cycle generation is thus a powerful aid for enhancing vehicle validation processes on a closer truck-by-truck basis, contributing to HDV efficiency and sustainability advancements.

3.2 Kinematic Segments

Driving sequences can be split into a series of shorter segments according to the four primary states of driving: acceleration, cruising, deceleration, and idling [26]. These states represent distinct phases of vehicle operation characterised by the vehicle's changes in speed.

Acceleration and deceleration are characterised by sequences of driving where an absolute change in speed between logged values exceeds a certain predefined threshold. Conversely, cruising is characterised by a sequence of consecutive logged vehicle speeds where the absolute speed difference is below that same threshold and the

speed is non-zero. Idling is defined as sequences where the vehicle is standing still while the engine is still running. An example of a driving sequence segmented into the four different driving states can be seen in Figure 3.1.

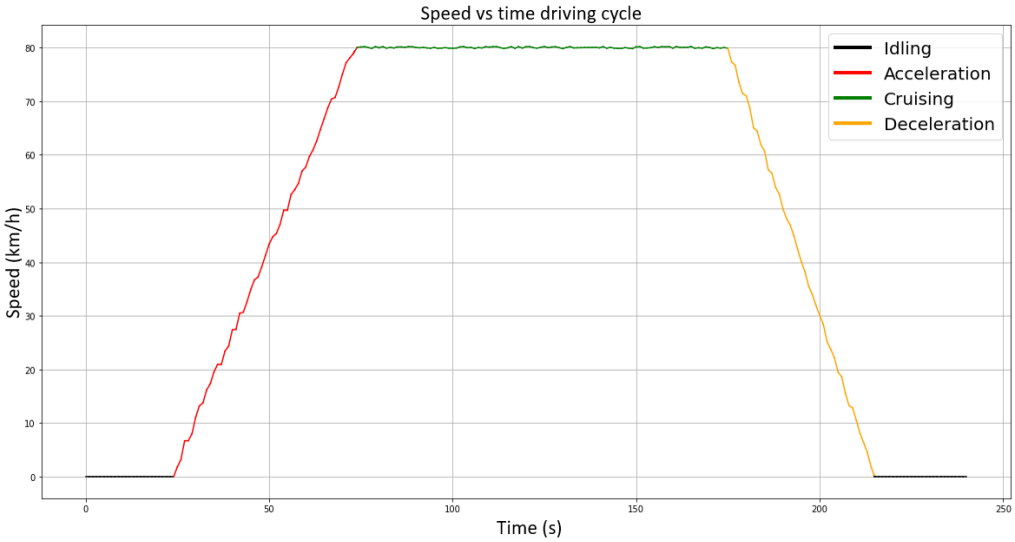


Figure 3.1: Driving sequence example segmented into the four primary of driving.

The proportion of time that vehicles spend in each state can significantly vary depending on their driving profile. For example, long-haul trucks usually spend a large portion of their operation in the cruising state, being able to maintain more consistent speed driving on highway roads with cruise control active. Other driving profiles, such as regional or local distribution, may spend much less time in the cruising state. These trucks may drive on smaller roads, idling when waiting at traffic lights, accelerating and decelerating more often and more aggressively. This variance in the proportion of time spent in different driving states plays a key role in influencing fuel consumption and overall driving efficiency of a vehicle [27]. Understanding these dynamics can be utilised for the generation of more representative DCs targeting certain driving behaviors, which can improve estimates of fuel economy and consumption [23].

In the context of DC generation and validation, KSs often serve a dual purpose. Firstly, they are utilised as a DC construction model technique to construct cycles that reflect the operational patterns of different HDV profiles [23]. A set of segments is sampled and concatenated to construct a complete DC, more details on this construction model technique are covered in Chapter 3.6.3. Secondly, the proportion of time spent in each distinct state (acceleration, cruising, deceleration, and idling) is a CP commonly used for DC evaluation [1], [19], [20], [22], [23], [28], [29]. More on this and other cycle CPs is covered in Section 3.5.1.

3.3 Machine Learning

ML is a branch of artificial intelligence that focuses on analysing data in order to learn and make predictions [30]. ML algorithms are designed to discover meaningful

patterns in data, storing this knowledge in models to be used for decision-making or predictions. With training they improve and become more accurate without being explicitly programmed for each task.

ML is widely used for generating representative DCs for its ability to handle the complexity and variability of real-world driving data. ML algorithms can identify patterns within large volumes of data which is useful for constructing the cycles with the use of these key patterns, making them a valuable tool for DC development.

3.3.1 Supervised Learning

Supervised learning is a type of ML where the algorithm learns a function from a set of training examples [31]. This set, often referred to as a labeled dataset, contains input-output pairs of an underlying unknown function where the desired output is known. The goal of supervised learning is ultimately to return a function f, given a set of examples of Y, that best approximates Y [31]. In its most basic form, a supervised learning algorithm can be represented by the equation $Y = f(x)$. Learning this function, or mapping from inputs to outputs, allows the model to predict the output for new unseen data.

In DC generation, supervised learning is applied to tasks where the targets or categories are known beforehand, specifically through regression or classification. For example, prior research used regression to predict continuous variables like vehicle energy consumption per mile based on driving data [11]. Classification, on the other hand, has been applied when refining the outputs of unsupervised methods [20], [23]. However, the effectiveness and applicability of supervised learning in DC generation depends on the availability and quality of labeled data which can limit its usefulness. While K-means and other unsupervised techniques are useful for exploring and segmenting the driving data, supervised learning is commonly applied for improving and refining the analysis.

3.3.1.1 Support Vector Machine

SVM is a supervised learning algorithm widely used for classification tasks [32]. The algorithm finds an optimal hyperplane that separates the different classes in the input-defined feature space and maximises the margin between those classes [31]. The margin is defined as the distance between the hyperplane and the nearest data points of each class.

SVMs are well known for their robustness, strong generalisation capabilities, and ability to find unique global optimum solutions [32]. They can handle both linear and nonlinear data due to their use of kernel functions. These functions transform the input data into a higher-dimensional space where linear separation of different classes becomes possible.

This capability has been useful for analysing driving data and refining inital K-means clustering, as demonstrated in a previous study [23]. While K-means is effective for identifying initial clusters, it typically does not achieve global optimization and struggles with nonlinear data. In such cases, SVM can take the preliminary clusters

from K-means and use its kernel functions to address the nonlinear patterns in the driving data. This approach can refine the cluster boundaries and potentially lead to more accurate DC models.

3.3.1.2 Random Forest

Another popular supervised learning algorithm is RF. RF is an ensemble learning algorithm that improves stability and accuracy over a single decision tree by creating and combining multiple trees [30]. It builds the decision trees using different subsets of the data and features and then aggregates their predictions. In RF classification, each decision tree makes its prediction and the final output is determined by taking the most frequent prediction among all the trees.

Decision trees are the foundation of RF. They model the relationship between input features and a target variable using a hierarchical tree structure. This structure includes nodes that make decisions based on input features, and branches that represents the outcomes of these decisions. The algorithm starts at the root node and splits data points based on criteria that maximise information gain for classification or minimises variance for regression. Information gain measures how well an input feature separates data points according to their classification target [33], while variance reduction in regression groups data points with similar target values [34]. With each split, new internal nodes are created until the leaf nodes are reached, which represent the predicted values of the output variable.

Similar to SVM, RF has proven useful in refining clusters initially identified by K-means in driving data analysis. RF addresses the non-linear relationships in the data and limitations of K-means in achieving global optimisation [20]. Additionally, because it can process large datasets efficiently with parallel training, it is especially well-suited for analysing large amounts of driving data.

3.3.2 Unsupervised Learning

Unsupervised learning is a type of ML involving training an algorithm on data without labels [30]. The goal is to discover hidden patterns and structures within the data with the desired output being unknown. When analysing driving data, unsupervised learning is useful for clustering and dimensionality reduction.

3.3.2.1 K-means Clustering

One of the most widely used ML algorithms for developing representative DCs is K-means clustering. The algorithm groups all data points into k clusters by assigning each point to the cluster with the closest mean [30]. This is repeated until the sum of the squared distances between data points and their cluster mean is minimised. The objective function of K-means can be written as [35]:

$$J_{\text{kmeans}} = \sum_{i=1}^{n} \min_{1 \leq k \leq K} \|x_i - f_k\|^2 = \sum_{k=1}^{K} \sum_{i \in C_k} \|x_i - f_k\|^2, \tag{3.1}$$

where x_i represents the data points and f_k represents the cluster centers.

K-means is a greedy algorithm that is guaranteed to converge to a local minimum, but the minimisation of its objective function is known to be NP-Hard [35]. Despite this limitation, K-means is effective for analysing large datasets and grouping similar driving patterns, which makes it popular for analysing driving data for developing representative DCs.

When evaluating the quality of the clusters various metrics can be used for evaluating different aspects. Two such metrics are Within-cluster Sum of Squares (WSS) and Between-cluster Sum of Squares (BSS) [36]. WSS measures how compact each cluster is, ensuring that points belonging to the same cluster are as close to each other as possible. Conversely, BSS evaluates the separation between clusters, aiming to keep them distinct with minimal overlap. These metrics are useful for independently assessing the compactness of each cluster and the separation between them.

In addition to WSS, another metric for evaluating the compactness of clusters is called Compactness. Compactness for the i-th cluster is defined as:

$$\widehat{Compactness}_i = \frac{1}{|C_i|} \sum_{x_j \in C_i} \|x_j - m_i\| \tag{3.2}$$

where C_i is the i-th cluster, m_i is the cluster center in C_i, x_j represents the data points in C_i, and $|C_i|$ represents the number of data points in C_i. To assess the overall compactness across all clusters, the average Compactness is calculated as:

$$Compactness = \frac{1}{K} \sum_{i=1}^{K} \widehat{CP}_i \tag{3.3}$$

This measure indicates how tightly the data points are grouped around the cluster center [23].

WCSS and Compactness are closely related metrics that measure how compact clusters are but in different ways. WCSS calculates the sum of squared distances between data points and their respective cluster centers. On the other hand, Compactness measures the average distance between data points and their cluster centers.

Separation is another metric for evaluating clustering quality, which measures the separation between different clusters. It is defined as:

$$Separation = \frac{2}{K^2 - K} \sum_{i=1}^{K} \sum_{j=i+1}^{K} \|w_i - w_j\|_2 \tag{3.4}$$

where w_i and w_j are the centers of clusters i and j respectively, and K is the total number of clusters. This metric calculates the average distance between the centers of all pairs of clusters, providing insights into how well-separated the clusters are from each other [23].

One important consideration when using K-means clustering is determining the optimal number of clusters, k. The Elbow Method is often used for this which finds the best value for k by identifying the point where the WSS starts to decrease more slowly, indicating that adding more clusters yields diminishing returns [37]. Another method for finding the optimal number of clusters is gap statistic [38]. This method compares the total within-cluster variation for different numbers of clusters with what would be expected under a null reference distribution of the data. The gap statistic measures the logarithmic difference between the mean distances of the reference datasets and those of the original dataset. The optimal number of clusters is where this gap is the largest.

3.3.2.2 Principal Component Analysis and Kernel Principal Component Analysis

PCA is a dimensionality reduction algorithm that simplifies datasets with many interrelated variables while retaining as much variance as possible [39]. The idea is to transform the data into new, uncorrelated variables known as principal components. This transformation aims to reduce the complexity of the dataset while still retaining the most important information, making it easier to analyse. Historically, PCA was introduced by Pearson in 1901 as a method for linear regression and later expanded by Hotelling in 1933 for analysing correlation structures among multiple random variables [40]. Because of its effectiveness and reliability, PCA is widely used for feature extraction and data visualization in many fields.

The kPCA algorithm is an extension to PCA designed to handle datasets that are not linearly separable [41]. It uses kernel methods to project the data into a different higher-dimensional space where it becomes linearly separable. This makes kPCA effective for uncovering patterns in complex datasets where traditional PCA might struggle. Both of these algorithms are useful for analyzing complex driving data as they can identify the key features that influence driving patterns.

An important step when using PCA and kPCA is deciding on the number of principal components to use. The decision is usually based on analysing the cumulative variance explained by the components against the number of components [20], [23]. The goal is to select the principal components that captures most of the variance in the original variables while using as few principal components as possible. Essentially striking a balance between simplifying the data and retaining important information.

When used properly PCA and kPCA can be powerful tools for simplifying complex datasets in order to make analysis more efficient and insightful. In DC development, these techniques can be applied to reduce a wide range of CPs computed from driving data to a smaller set of key principal components that are easier to manage.

3.3.2.3 Kernel Density Estimation

KDE is a method for estimating the PDF of a random variable [42]. Given a set of independently and identically distributed data points $(x_1, x_2, ..., x_n)$ drawn from a population X with an unknown probability distribution $f(x)$, the KDE is defined as

shown in Equation 3.5.

$$\hat{f}(x) = \frac{1}{nh} \sum_{i=1}^{n} K(\frac{x - x_i}{h}) \tag{3.5}$$

Where K is the kernel function, x_i are the data points, n is the number of data points, and h is the bandwidth parameter.

Several kernel functions can be used for KDE, such as Epanechnikov, Triangular and Gaussian [42]. These kernels are symmetric, simplifying their mathematical formulation. In this thesis Gaussian KDE was utilised, defined in Equation 3.6.

$$K(x) = \frac{1}{\sqrt{2\pi}} e^{-t^2/2} \tag{3.6}$$

The bandwidth parameter h, also known as the smoothing parameter or window width, controls the amount of smoothing applied to the estimated PDF. Too small values of h may result in an estimator sensitive to minor fluctuations in the data, while a too large bandwidth might result in excessive smoothing and loss of important characteristics of $f(x)$. Therefore, appropriate bandwidth selection is crucial to achieve the desired performance.

Several methods exist for bandwidth selection. In this thesis, Silverman's rule of thumb was used [43], which is defined in Equation 3.7.

$$h \approx 0.9 \cdot min(\hat{\sigma}, IQR/1.35) n^{-1/5} \tag{3.7}$$

Where $\hat{\sigma}$ is the standard deviation of the sample data, n is the number of data points, and IQR is the Interquartile Range which is the difference between the 0.75 and 0.25 quantiles of the data ($x_{0.75} - x_{0.25}$). Silverman's rule of thumb is straightforward to implement, making it a practical and effective method for bandwidth selection.

3.4 Markov Chains

MCs are mathematical models describing random transitions from one state to another within a finite set of states, represented by a series of random variables $X_0, ..., X_n$ [44]. They are defined by the probability distribution for the initial variable $p(X_0)$ and the transition probabilities for the following variables $p(X_{m+1}|X_m)$. A fundamental characteristic of MCs is the Markov property of memorylessness, which means the probability of transitioning to a state only depends on the current state and not on any states before it. Mathematically it can be expressed by equation 3.8, where X_{n+1} is the next state, X_n is the current state and $X_{n-1}, ..., X_0$ are the previous states [45].

$$P(X_{n+1} = x_{n+1}|X_n = x_n, X_{n-1} = x_{n-1}, ..., X_0 = x_0) = P(X_{n+1} = x_{n+1}|X_n = x_n) \tag{3.8}$$

MCs are widely used in many fields for their ability to model complex processes with discrete, probabilistic transitions. This makes them especially useful for analysing sequential data such as vehicle dynamics. For analysing driving data, MCs are used to model the probabilities of transitioning between different driving states, such as acceleration and speed, based on real-world data.

3.5 Evaluating Representativeness

To be able to determine if a DC is representative of real-world driving data, it is necessary to define what a representative DC means. Though a formal definition is yet to be established, most studies define a representative DC as one whose statistical properties closely mirror those of typical driving conditions [46]. In the following sections, key concepts regarding the evaluation process of DC representativeness are covered.

3.5.1 Characteristic Parameters

To validate if a DC is representative of a larger group of real driving data, CPs are defined. This is a wide range of parameters, each describing a certain characteristic of a driving segment. Some of the most adopted parameters that are utilised in existing research [1], [10], [12], can be seen in Table 3.1.

Group	**Parameter**	**Unit**
Speed	Average Vehicle Speed	kmh^{-1}
	Maximum Vehicle Speed	kmh^{-1}
	Average Vehicle Running Speed	kmh^{-1}
	Std speed	kmh^{-1}
	Std running speed	kmh^{-1}
	Percentage duration in certain speed levels	%
Driving State	Percentage duration in Acceleration	%
	Percentage duration in Deceleration	%
	Percentage duration in Cruise	%
	Percentage duration in Idling	%
Acceleration & Deceleration	Maximum Acceleration	ms^{-2}
	Average Acceleration	ms^{-2}
	Std Acceleration	ms^{-2}
	Maximum Deceleration	ms^{-2}
	Average Deceleration	ms^{-2}
	Std Deceleration	ms^{-2}
Dynamics	No. of stop-start per km	event/km
	Positive Kinetic Energy (PKE)	ms^{-2}

Table 3.1: CPs frequently utilised in existing research.

CPs are utilised for several different purposes in the DC construction process. First and foremost, their main application involves the comparison between the operational data and a candidate DC. A cycle is considered more representative of the real-world driving data if the RD between its CPs and the CPs of the real-world data is minimised. For more details, see Section 3.5.2. As such, if the CPs of a candidate DC have low RD to the CPs of the real operational data, it is concluded that the driving behavior of the cycle is a good representation of the average operation of the targeted driving data.

In addition to cycle evaluation, CPs also play a key factor in commonly used driving cycle construction methods. CPs are used to calculate similarities between MTs and driving sequences. Further details on these applications of CPs are covered in Section 3.6.

3.5.2 Relative Difference

The RD between CPs of the candidate DC and the actual driving data is a widely used metric to evaluate the representativeness of DCs [28]. The RD formula for a CP is defined in Equation 3.9.

$$RD_i = \left| \frac{CP_i^* - CP_i}{CP_i} \right| \tag{3.9}$$

Where CP_i^* denotes the CP i of the candidate DC and CP_i the CP i of the operational data.

In addition to RD, Mean Relative Difference (MRD) is used as a metric in existing research [22]. MRD is the average of the RDs across all the considered CPs. The MRD is hence a singular value representing the overall average deviation, utilised for overall representativeness comparisons between different DCs. The mathematical formula for MRD is defined in Equation 3.10.

$$MRD = \frac{1}{n} \sum_{i=1}^{n} RD_i \tag{3.10}$$

3.5.3 Speed Acceleration Probability Distribution

SAPD is another common metric for measuring DC representativeness. This metric divides vehicle speed and acceleration into discrete bins, enabling a comparison between the distribution of these bins between the candidate DC and actual driving patterns. A close match between the SAPD of the DCs and real-world data indicates a higher degree of representativeness. To quantify this similarity, several statistical measures are often used, including:

- Sum Square Difference (SSD) [25]:

$$SSD_{SAPD} = \sum_{i=1}^{m} \sum_{j=1}^{n} \left(P_{ij}^* - \bar{P_{ij}} \right)^2 \tag{3.11}$$

- SAPD RD [19]:

$$RD_{SAPD} = \frac{\sum_{i=1}^{m}\sum_{j=1}^{n}\left|P_{ij}^{*} - P_{ij}\right|}{2} \tag{3.12}$$

The probabilities P_{ij}^{*} and P_{ij} reflect the frequency of observations in each bin for the candidate DC and the real driving pattern, respectively. The index m represents the total number of speed bins, while n represents the total number of acceleration bins.

3.5.3.1 Earth Mover's Distance

The Earth Mover's Distance (EMD) is a metric that can be used to compare histograms, and therefore also evaluate the similarity between SAPDs. EMD provides a way to measure the distance between two probability distributions by considering the minimal amount of work required to transform one of the distributions into the other [47]. The definition of EMD for SAPD is shown in Equation 3.14.

$$EMD_{SAPD} = \min \sum_{i}\sum_{j} f_{ij}d_{ij} \tag{3.13}$$

$$\text{subject to} \quad \sum_{i} f_{ik} - \sum_{j} f_{kj} = flow(k) \tag{3.14}$$

where d_{ij} is the distance between bins, f_{ij} is the amount of mass to be moved between bins to minimise the total distance, and $flow(k)$ ensures the balance of flow for each bin k [47].

EMD has been successfully used in many applications including image retrieval by comparing color and texture histograms, which demonstrates its effectiveness in capturing meaningful differences between distributions [48].

Using EMD improves the evaluation of SAPD as it provides an additional measure of similarity. EMD can capture slight shifts in the distribution more reliably than other commonly used measures, such as RD, which measures exact differences and may excessively penalise these shifts. Incorporating EMD metric ensures a more accurate assessment of how well a synthetic DC represents real-world driving behavior in terms of the SAPD.

3.5.4 Pearson Correlation Coefficient

The PCC measures the linear relationship between two variables [49]. It quantifies the degree to which changes in one variable predict changes in another, ranging from -1 to 1, with 0 indicating no linear relationship.

The formula for the PCC is:

$$r_{xy} = \frac{\sum_{i=1}^{n}(x_i - \overline{x})(y_i - \overline{y})}{\sqrt{\sum_{i=1}^{n}(x_i - \overline{x})^2 \sum_{i=1}^{n}(y_i - \overline{y})^2}} \tag{3.15}$$

where x_i and y_i are the individual sample points, $\bar{x_i}$ and $\bar{y_i}$ are the means of the x and y variables, respectively, and n is the number of sample points.

While the PCC is not typically used for evaluating the representativeness of generated DCs, it has been effectively used in prior research for selecting representative MTs for constructing DCs [20]. In this approach, CPs for each MT are reduced in dimensionality and then clustered. The PCC is then used to select MTs that best represent the overall characteristics of their respective clusters, ensuring that the selected MTs accurately reflect the driving data within those clusters.

3.6 Construction Models

To effectively use analytics methods for DC construction, the input data must be processed so that various models can use it to construct the sequence of data points representing the cycle. This processing is what can be referred to as different construction models which are utilised to generate sequences of synthetic time series data. There are some extensively used construction models in DC development. The most extensively researched methods, and the ones which are utilised in this thesis, are (1) Micro-Trip (MT) construction, (2) Speed Acceleration State (SAS) construction, and (3) Kinematic Segment (KS) construction.

3.6.1 Micro-Trip Construction

One of the most researched and utilised construction models is the MT-based method [28]. As mentioned in Section 2.1, an MT is a segment of data where the vehicle speed leaves and returns to some cutoff speed. Essentially a shorter driving segment within some longer one. The main adopted definition is having a cutoff speed of zero, which means an MT is a segment of data between the start and stop of a vehicle [1]. In Figure 3.2 an example trip can be seen consisting of two stop-and-go driving segments, which with a zero-speed cut-off speed is segmented into two separate MTs.

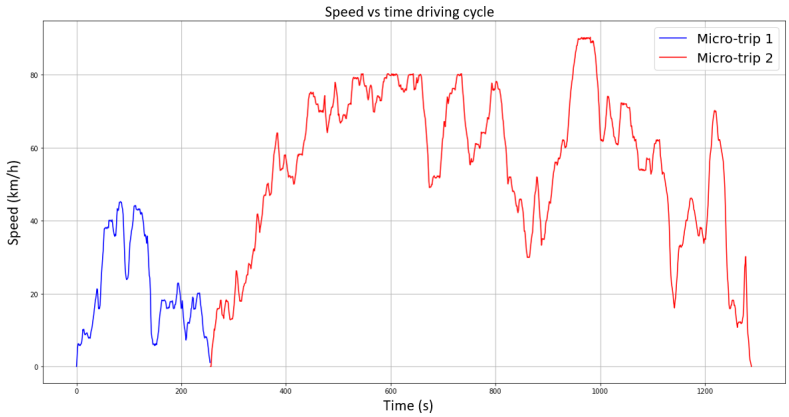


Figure 3.2: Example of a trip split into two separate MTs with a zero-speed cut-off threshold.

The input driving data, based on the cut-off speed, is segmented into different MTs. The construction of the DC is then based on sampling several MTs from the created population to construct a complete DC. As acknowledged in Chapter 2.1, there exist different types of techniques of how to sample MTs and constructing the complete cycle. The most adopted and researched approach is clustering trips based on a key set of CPs. Commonly, these are many of the same CPs used for the evaluation of DC representativeness.

The CPs of a trip describe its features and clustering methods utilise these as input. The resulting clustering of the trips describes both the similarities between different MTs and the distribution of MT types within the selected driving profile. In Figure 3.3 such an MT clustering can be seen for the CPs average speed and average positive acceleration.

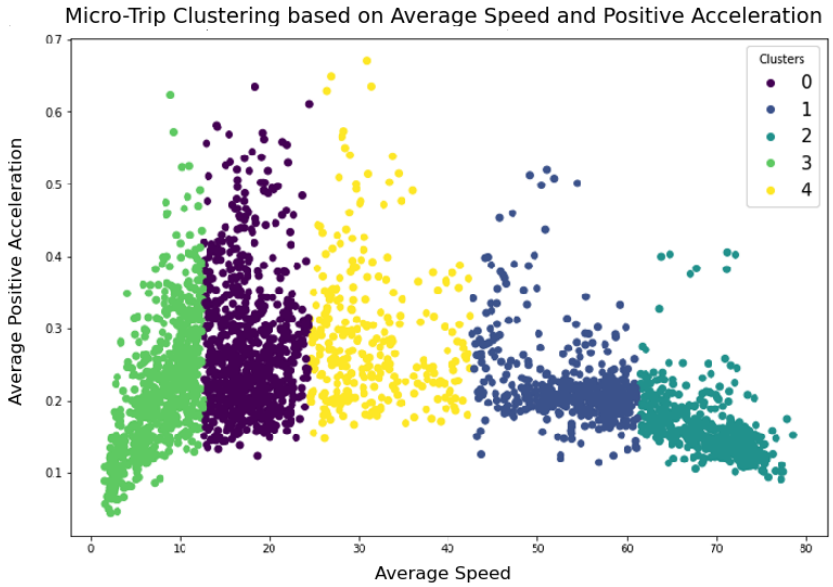


Figure 3.3: MT clustering on average speed and average acceleration.

The complete DC is then constructed by sampling MTs from these clusters. Most such strategies weigh in the frequency ratio of MTs between clusters, such that the final constructed trip represents an MT distribution as similar as possible to the one observed in the real data. Given the stochastic nature of sampling the MTs, typically a set number of candidate cycles are constructed [28]. The candidate cycle evaluated as the most accurate representation of the real-world driving data is selected as the final DC.

3.6.2 Speed Acceleration State Construction

SAS construction is another construction model approach heavily utilised in existing research [28]. This method considers vehicle speed and acceleration as distinct state variables, and the driving conditions in a cycle can be considered a stochastic process [50]. Vehicle speed and acceleration are discretised into segments of some interval.

The intervals can vary, however, in common practice, speeds are discretised with an interval of 1km/h, whereas accelerations are typically discretized with intervals of $0.1\text{m}/s^2$ [12], [51]. The discretisation transforms the continuous variables into a discrete set of states representing all possible speed and acceleration combinations.

With the state space discretised, probabilistic methods can be utilised for simulating a sequence of states to construct a complete DC. The MC model is the widely used approach where the Markov property is utilised for calculating transition probabilities between states [12], [22], [51]. Utilising this property, a Transition Probability Matrix (TPM) can be constructed where each state (v_i, a_j), corresponds to a probability vector $p_{ij} = [v_1, v_2, ..., v_max]$. This vector describes the probability of the next speed in the sequence given the current state:

$$TPM = [p_{ij}]_{I\times I} = \begin{bmatrix} p_{11} & \cdots & p_{1I} \\ \vdots & \ddots & \vdots \\ p_{I1} & \cdots & p_{II} \end{bmatrix}_{I\times I} \tag{3.16}$$

Where I is the total number of discretized speeds. A three-dimensional visualisation of a TPM can be seen in figure 3.4.

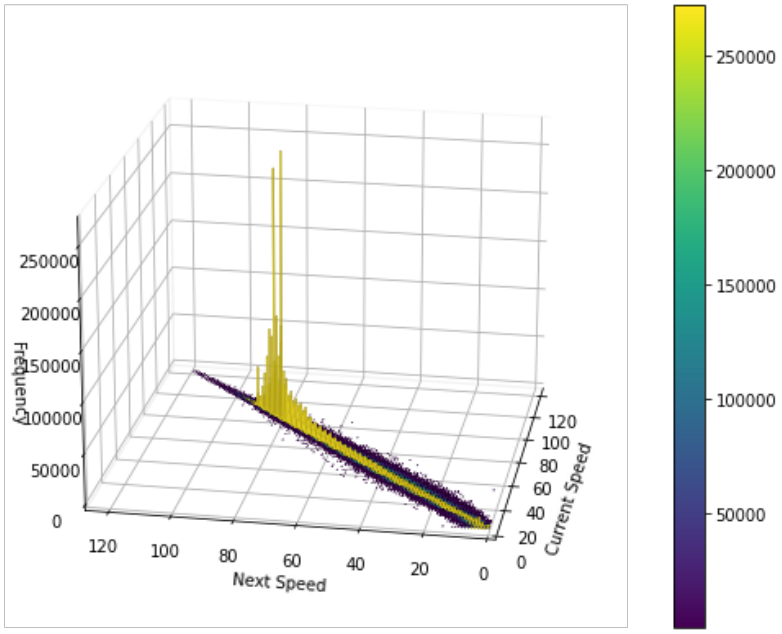


Figure 3.4: TPM three-dimensional visualisation.

The construction of a complete DC is then made by sampling states from the TPM. Each sampled sequence is initiated by a starting state, typically the idling state of zero speed and zero acceleration, and progresses to sample states according to the probabilities of the TPM. The procedure is iterative, continuing until it satisfies one or more predefined target criteria such as a minimum duration, and that the cycle ends back at zero speed.

The culminating sequence of sampled data points is the final synthetic DC. Similar to the MT construction model, this too is the result of a stochastic process. Therefore, typically multiple candidate cycles are generated. These candidates are all evaluated to determine the most representative cycle.

3.6.3 Kinematic Segment Construction

The KS construction method is based on calculating frequencies of certain driving scenarios occurring in the operational data. Formally it involves splitting data segments into the four primary states of driving: acceleration, cruising, deceleration, and idling. As with the other construction models, this data segmentation indicates what type of driving segments commonly appear in the targeted driving profile. For instance, in long-haul truck applications, one expects to see a larger proportion of cruising segments compared to for instance local distribution.

For each of the KS, CPs are calculated. The CPs are then utilised for clustering the segments, measuring similarities and differences of different segments. Although each distinct parameter characterises certain information about the driving segment, the parameters are not independent of each other. Therefore, PCA and kPCA, as described in Section 3.3.2.2, are quite commonly utilised to reduce the dimensionality of the parameters while still retaining a certain level of variance.

A complete DC is constructed by sampling and concatenating the most representative segments according to some predefined rules until the cycle duration limit is met. The process is stochastic as the other construction models, hence, the construction process is repeated to generate a set of candidate DCs which all are evaluated.

4

Method

This chapter goes into detail about all relevant methodological steps taken during the research. All practical implementations, including data preparation, method implementations, and cycle evaluations, were implemented and executed in Python. Data preparation was mainly conducted using the Spark library for handling large datasets. The Pandas library was used whenever the dataset size allowed for it.

4.1 Choosing Data Analytics Methods

The selection of data analytics methods for this research was informed by a review of existing methodologies for constructing representative DCs for HDVs, as detailed in Chapter 2. These methods were chosen for their demonstrated ability to produce accurate DCs, with each offering a different approach to represent driving behaviors.

The following methods were selected for further exploration and comparison:

1. SAS methods
    - Markov Chain (MC)
    - Kernel Density Estimation - Markov Chain (KDE-MC)
2. MT methods
    - Micro-Trip Clustering (MTC)
    - Micro-Trip Clustering with dimensionality reduction and Random Forest refinement (MTC-RF)
3. KS methods
    - Kinematic Segment Clustering with Support Vector Machine refinement (KSC-SVM)

With this selection, the objective was to identify the method that most accurately represents real-world HDV driving patterns. Details of method implementations are covered in Section 4.3.

4.2 Data Collection and Preparation

This section goes into details of the process undertaken to collect and prepare operational data from Volvo Group customers.

The commitment to upholding the highest standards of data integrity and compliance with privacy legislation was a key aspect throughout all steps of the thesis. Accordingly, all data included in this thesis report has been anonymised or aggregated. The anonymisation process involved excluding any data features or information that could potentially identify any individual or customer of the Volvo Group. These measures ensure that there are no possibilities of tracing any information back to specific individuals or organisations, thereby ensuring compliance with the General Data Protection Regulation (GDPR) standards.

The following subsections detail the steps undertaken in the data collection and preparation process, which included the following:

- Querying the data source: Collected long-haul operational data from the data source. Relevant signals necessary for the cycle construction process were determined and included in the input dataset.
- Data preprocessing and trip segmentation: The raw input data was cleaned and segmented into distinct trips. This was done to ensure data quality and transform it into a suitable format for applying data analytics methods.
- CP evaluations of trips: Computed CPs for each trip in the input data. enabling statistical comparisons between trips.
- Statistical selection of representative trips: A statistical selection of the most representative trips was made, eliminating unrepresentative trips and outlier behavior from the input.

By following these steps, the study aimed to create a comprehensive framework for analysing real-world driving data and preparing it for DC generation.

4.2.1 Data Source

The data used for the thesis was operational vehicle data in high-resolution time series format from Controller Area Network (CAN) bus recordings. This data offered a robust basis of real-world driving input data. The targeted driving profile of this thesis was the long-haul operation.

Within the collected dataset, a comprehensive set of signals was available, which were valuable for DC generation and validation purposes. The main signals that were utilised in this thesis were:

- Absolute Time (timestamp): The precise time of each recorded data point. Used for monitoring driving events and dynamics throughout trips.
- Vehicle Speed (km/h): Recordings of the vehicle's speed at each timestamp. Providing details of speed variations and driving states.

- Vehicle acceleration (m/s^2): The rate at which the vehicle's speed changes, providing insights into acceleration and deceleration patterns under different driving states.
- Engine speed (RPM): Monitors the engine's revolutions per minute. Examining the variations in engine speed across a trip can provide valuable insights into potential data inconsistencies or anomalies seen in the data as well as indications of engine shut-off events.
- Vehicle distance (m): Measures the total cumulative distance traveled by the vehicle, utilised for analysing the lengths of trips and shorter segments within each trip as well.

These signals were essential for either data cleaning or serving as inputs to data analytics methods. They were used in the cycle generation, CP calculations, and cycle evaluation processes. More details on how and where in the process the different signals were utilised are covered in Sections 4.2.2, 4.2.3 and 4.2.4.

Although the data at the source is stored with higher resolution, the data collected for this thesis was downsampled to 1-Hz frequency (1-second resolution). Downsampling to 1-Hz was done for computational feasibility purposes while still maintaining a reasonably high resolution. Moreover, 1-Hz is the most commonly adopted resolution utilised in existing research [10], [12], [19].

4.2.2 Data Preprocessing

The data preprocessing steps undertaken can be split into four main areas. The order of which they were completed was as follows below.

1. Filter out unrealistic values: Certain signals from the CAN-bus would at times report unrealistic values indicating an error has occurred in the logging. Such data points were removed from the input data to ensure data quality. The signals considered for this initial cleaning were vehicle- and engine speed. Thus, any data points with speed values of 180 km/h or above, as well as any engine speed value of 8000 rpm or above were removed.
2. Split continuous recordings into trips: The continuous time recordings were then split into distinct trips based on two different criteria, vehicle "key-off" events or prolonged idling segments. Vehicle "key-off" events were indicated by a gap in the recorded timestamps of consecutive logged data points. If the gap between timestamps exceeded 10 seconds, then this was considered to be a new trip. This split caused several different possible outcomes which were handled as follows:
    - If a trip started and ended in an idling segment this was considered a correctly identified trip.
    - If a trip started or ended with a vehicle speed above zero, this was handled with the following cases:

- If the trip started with a vehicle speed above 5km/h, then this was considered an unreliable trip to include and the trip was completely removed. If the trip started with a speed value less than or equal to 5km/h, then a vehicle start at 0km/h speed was imputed.
- If the trip ended with a vehicle speed above 10km/h, then this was considered an unreliable trip to include and the trip was completely removed. If the trip ended with a speed value less than or equal to 10km/h, then a vehicle ending at 0km/h speed was imputed.

Prolonged idling segments, which were classified as segments of logged zero-speed values for more than two straight hours, are typically not considered part of a single normal long-haul distribution operation. Therefore, trips including such segments were split into separate trips. Thus, no individual idling segment in a trip on its own exceeded two hours.

3. Removal of trips with a total distance shorter than 10km: This filter removed short trips that were not part of the typical long-haul operation. Examples could be refuelling the truck or maintenance work.

4. Speed change filter: Instances of sudden, unrealistic speed changes occurred in the data. These changes were also often coupled with an implausible change in the cumulative distance counter, suggesting some logging error. To ensure data reliability, a filtering function was implemented, analysing and modifying the second-by-second logs as follows:

   - *Speed spikes:* Speed spikes were detected by a rapid increase directly followed by a rapid decrease in speed exceeding 10km/h (or vice versa), accompanied by a realistic increase in cumulative distance counter (calculated by current speed and time passed). These types of instances were treated as faulty speed logs but the trip was overall treated as a reliable trip. The spike was removed and interpolated to maintain speed continuity within the trip. If such a speed spike was accompanied by an unrealistic change in the cumulative distance counter, then the trip was determined to be unreliable and was completely removed.
   - *Non-spike unrealistic speed changes:* If an unrealistic speed increase (over 10km/h) or decrease (over 15km/h) was identified but was not a spike in speed, then the trip was determined to be unreliable and was completely removed.
   - *Zero-speed drift:* During prolonged periods of idling, at times zero-speed drifting occurred indicated by small non-zero speeds being logged despite the vehicle not moving. To account for this, a filter was applied analysing all MTs which had all speed logs less than 2 km/h. If the cumulative distance counter of the vehicle increased during this MT, then the vehicle had been moving and hence no action was taken. However, if the distance counter remained unchanged then all the speed values were flattened down to zero.

This filter function significantly enhanced the quality of the speed data, ensuring that during suspicious speed changes the vehicle's covered distance remained within a plausible distance.

After completing all of the preprocessing steps described above, the raw data was refined and ready for analysis. The following Sections 4.2.3 and 4.2.4 go into additional data handling steps that were taken before the application of the data analytical methods.

4.2.3 Characteristic Parameter Evaluations of Trips

Following the completion of data cleaning and segmentation steps, CPs were calculated for each trip. The CPs are listed and detailed in Table 4.1. In total nineteen different CPs were calculated and included a wide range of metrics that cover various aspects of speed and acceleration patterns, driving state ratios, and energy utilisation. This broad set was selected to enable a quantitative analysis and comparison of trips. Incorporating parameters a large number of parameters to offer insights into operational patterns in numerous dimensions.

Parameter	**Symbol**	**Unit**	**Definition**
Average Vehicle Speed	V_{avg}	kmh^{-1}	Average vehicle speed including idling segments
Maximum Vehicle Speed	V_{max}	kmh^{-1}	Maximum vehicle speed
Average Running Vehicle Speed	Vr_{max}	kmh^{-1}	Average vehicle speed excluding idling segments
Std Vehicle Speed	V_{std}	kmh^{-1}	Standard deviation of vehicle speed including idling segments.
Std Running Vehicle Speed	Vr_{std}	kmh^{-1}	Standard deviation of vehicle speed excluding idling segments.
Acceleration Ratio	p_a	%	Proportion of time accelerating $a > 0.15m/s^2$.
Cruise Ratio	p_c	%	Proportion of time cruising $-0.15 \leq a \leq 0.15m/s^2, v > 0km/h$
Deceleration Ratio	p_d	%	Proportion of time decelerating $a < -0.15m/s^2$ or ($a < 0m/s^2$,$v = 0km/h$)
Idling Ratio	p_i	%	Proportion of time idling ($v = 0km/h$, $a = 0m/s^2$)
High-Speed Ratio	p_{hv}	%	Proportion of time with $v >= 70km/h$
Low-Speed Ratio	p_{lv}	%	Proportion of time with $v <= 20km/h$
Maximum Acceleration	a_{max}	ms^{-2}	Maximum vehicle acceleration
Average Acceleration	a_{avg}	ms^{-2}	Average vehicle acceleration
Std Acceleration	a_{std}	ms^{-2}	Standard deviation of vehicle acceleration
Maximum Deceleration	d_{max}	ms^{-2}	Maximum vehicle deceleration
Average Deceleration	d_{avg}	ms^{-2}	Average vehicle deceleration
Std Deceleration	d_{std}	ms^{-2}	Standard deviation of vehicle deceleration
No. of stop-start per km	SSPK	event/km	Vehicle stop and starts per kilometer
Positive Kinetic Energy	PKE	ms^{-2}	Vehicle PKE ($\frac{1}{D}\sum_{t=2}^{T}(v_t^2 - v_{t-1}^2)$), where $v_t > v_{t-1}$, D = Total trip distance, T = Total elapsed time

Table 4.1: Calculated CPs including their symbols, units, and definitions.

4.2.4 Statistical Selection of Trips

A statistical selection among the input trips to ensure that the DC generation was based on reliable and relevant data was then conducted. This selection was inspired by a methodology described by Kondaru et al. [1], whose research aimed to filter out trips that deviated significantly from typical driving patterns and ensure the analysis was based on data that accurately reflected the driving behaviors of HDVs under normal operating conditions.

The selection methodology was based on the CPs of all trips, the CPs previously listed in Table 4.1. This set of CPs provided insights into driving behavior, enabling a comparative analysis of trips. Through this comparison, outlier trips were identified and excluded from the dataset.

The selection was based on the statistical principle that data from experiments performed under consistent conditions tend to form a normal distribution, centering around the most common value or mean. However, given that the CPs of trips might not perfectly fit a normal distribution, it was not possible to directly apply mean and standard deviation values to calculate the acceptance range. Instead, a tailored statistical method was used for determining the acceptance range and selecting trips. That process was conducted as follows:

1. To determine what trips to include in the analysis, an acceptance criteria of 95% was established. This value was selected to be high enough to broadly capture typical driving behavior while still excluding the most divergent outliers.
2. A histogram was created for each CP and the most frequent bin was identified as the provisional mean.
3. Each CP acceptance range was identified as the following: Calculating each CP's cumulative frequency distribution, beginning from the provisional mean, the acceptance range was expanded outwards until 95% of trips were within the range. This established the range of values that were considered representative of typical driving behavior, ensuring the selected trips aligned with common patterns observed in the dataset. An example of this process for the percentage cruise parameter is shown in Figure 4.1.
4. A binary matrix was computed, each column representing a CP and each row representing a trip. A value of one in the matrix indicated a trip's CP value was within the acceptance range, and a value of zero indicated it was outside.
5. Trips were selected based on their scores from the binary matrix, which quantified how closely each trip matches the typical patterns of the dataset as a whole by the sum of its corresponding row in the matrix. Meaning, if a trip was within the acceptance threshold for all nineteen CPs then it received the maximum score of nineteen. The top 95% scoring trips were retained, ensuring the analysis focused on the most representative data.

This methodology, adapted from the statistical approach detailed by Kondaru et al., provided a framework for identifying and selecting the most representative trips for DC generation. The 95% acceptance threshold ensured that the trips least

representative of typical long-haul operational behaviour were excluded, while still keeping most of the data.

The resulting trip acceptance ranges for each of CPs are listed in Table 4.2.

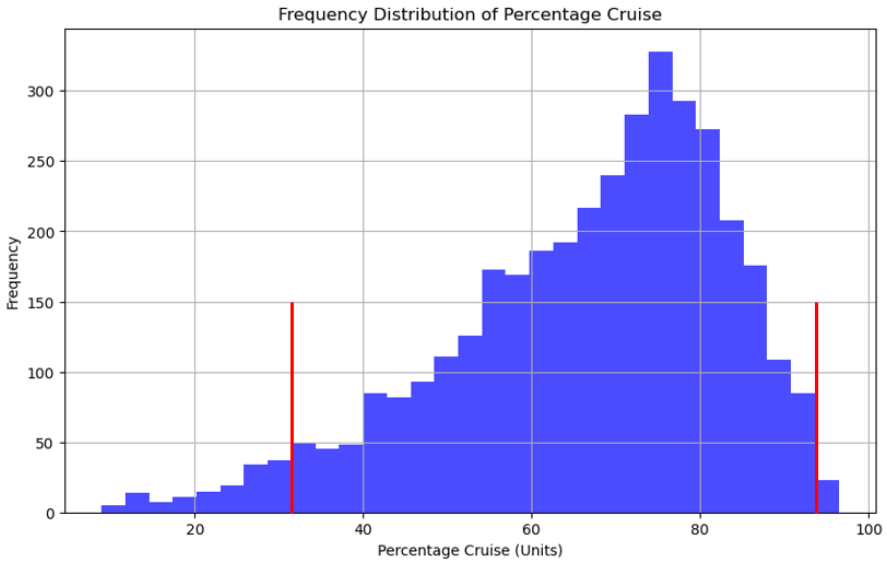


Figure 4.1: Frequency distribution of 'Percentage Cruise' with acceptance range marked as red lines

Parameter	**Acceptance Range**
Average Speed	$[36.44, 88.67]$
Maximum Speed	$[81.67, 104.03]$
Average Running Speed	$[47.63, 87.99]$
Stddev Speed	$[15.25, 40.74]$
Stddev Running Speed	$[10.70, 32.16]$
Percentage Acceleration	$[2.03, 23.10]$
Percentage Deceleration	$[2.48, 20.61]$
Percentage Cruise	$[31.50, 93.62]$
Percentage Idling	$[0.08, 3.19]$
Maximum Acceleration	$[0.70, 2.16]$
Maximum Deceleration	$[-2.81, -1.04]$
Average Acceleration	$[0.24, 0.52]$
Average Deceleration	$[-0.59, -0.26]$
Stddev Acceleration	$[0.09, 0.32]$
Stddev Deceleration	$[0.14, 0.46]$
Percentage Low Speed	$[0.52, 19.87]$
Percentage High Speed	$[23.62, 98.40]$
SSPK	$[4.29 \times 10^{-5}, 0.033]$
PKE	$[442.87, 2656.68]$

Table 4.2: CP acceptance ranges for the statistical selection of trips.

4.3 Application of Data Analytics Methods

This section covers the implementation and application parts of the DC generation methods. This is split into five main subsections, covering all technical implementation details of the methods outlined in Section 4.1.

4.3.1 Markov Chain

This section details the implementation of the Markov Chain (MC) method used to model vehicle dynamics through discrete state construction of speed and acceleration. Speed states v_i were defined at intervals of 0.2 km/h, and acceleration states a_j at intervals of 0.1 m/s^2.

The TPM, describing the probability of transitioning from a given state ($v_t \in v_s, a_t \in a_a$) to a new speed ($v_{t+1} \in v_{s'}$), was calculated using Equation 4.1. In the equation, $f(s, a, s')$ is the frequency of going from speed-acceleration state (v_s, a_a) to the next speed $v_{s'}$, and $f(s, a)$ represents the total frequency of being in the state (v_s, a_a). These frequencies were used to normalise the TPM. In Figure 4.2 an example plot is shown for the state $v = 80$ km/h and $a = 0$m/s ^{2}, showing the next speed transition probabilities for that state.

$$P(s, a, s') = P(v_{t+1} \in v_{s'} \mid v_t \in v_s, a_t \in a_a) = \frac{f(s, a, s')}{f(s, a)} \tag{4.1}$$

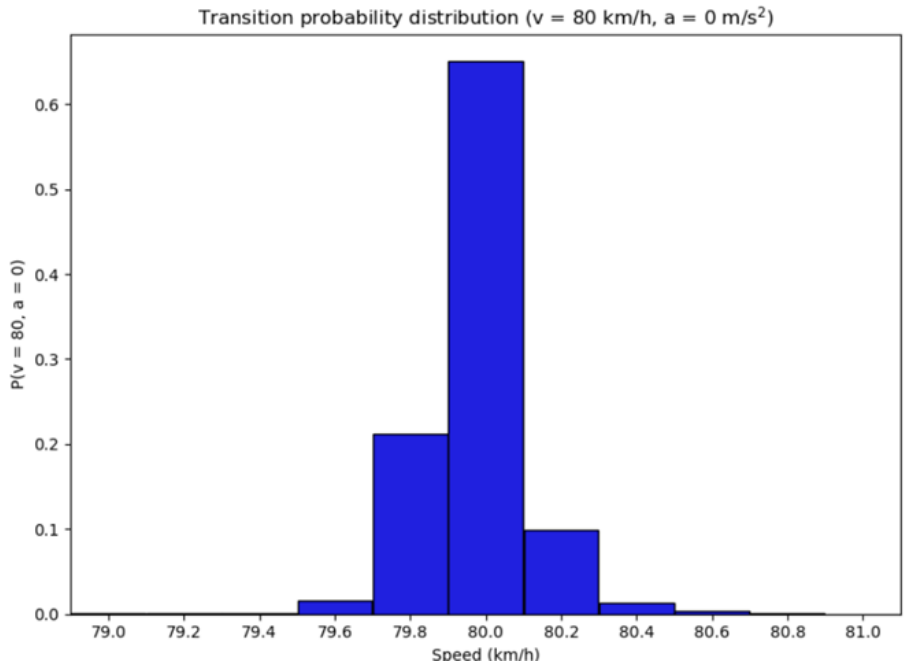


Figure 4.2: TPM next speed probabilities for state $v = 80$ km/h and $a = 0$m/s ^{2}

To construct DCs, a state of zero speed and zero acceleration ($v_0 = 0, a_0 = 0$) was initiated. The next speed (v_1) was then sampled from the probability vector in the

TPM associated with the state ($v_0 = 0, a_0 = 0$). The corresponding acceleration (a_1) was then calculated based on the change in speed ($a_1 = (v_1 - v_0)/3.6$), converting the acceleration to m/s^2. This process was repeated, sampling speeds from the TPM and calculating the corresponding acceleration.

The DC generation was completed when two termination criteria were met: (1) the cycle duration reached at least one hour, and (2) the cycle ended with a vehicle speed of zero. The complete construction process of an MC candidate cycle is described in pseudo-code in Algorithm 1.

Algorithm 1 MC DC construction

function MC(TPM, $v_{\text{current}} = 0, a_{\text{current}} = 0$)
  DC $\leftarrow (v_{\text{current}}, a_{\text{current}})$
  while True **do**
    $v_{\text{next}} \leftarrow$ SAMPLE_SPEED(TPM$[v_{\text{current}}][a_{\text{current}}]$)
    $a_{\text{next}} \leftarrow \frac{(v_{\text{next}} - v_{\text{current}})}{3.6}$
    DC $\leftarrow (v_{\text{next}}, a_{\text{next}})$
    $v_{\text{current}} \leftarrow v_{\text{next}}$
    $a_{\text{current}} \leftarrow a_{\text{next}}$
    if len(DC) ≥ 3600 and $v_{\text{current}} = 0$ **then**
      Break
    end if
  end while
  return DC
end function

Since the MC method is a stochastic process, there were no guarantees that a generated cycle would be back at zero speed after any predetermined duration. Therefore, generated cycles varied in length with the minimum possible length being one hour. Moreover, since each resulting candidate varies from the other, one hundred candidate cycles were generated.

4.3.2 Kernel Density Estimation - Markov Chain

This section details the implementation of the KDE-MC method. In this approach, a KDE was computed for each discrete state in the TPM to estimate a PDF for every combination of discrete speed and acceleration seen in the operational data. Unlike the MC method which sampled new speeds from discrete values, the KDE-MC method allowed for the sampling of continuous speed values from these PDFs. This method modification was implemented to explore whether this could result in a more realistic representation of vehicle speed changes given that the continuous next-speed sampling enables smoother, more transient speed transitions compared to the original MC method.

The KDE that was implemented was the Gaussian KDE, see mathematical formulation in Chapter 3.3.2.3. As such, each estimated PDF fits a Gaussian curve to the observed speed transitions. The observed next speed values in a given state

were the input data points, with their frequencies being their corresponding weights. Therefore, a commonly observed next speed was assigned a higher density compared to a less commonly observed speed. The bandwidth for each KDE was determined using Silverman's rule of thumb to ensure a bias and variance balance in the density estimation.

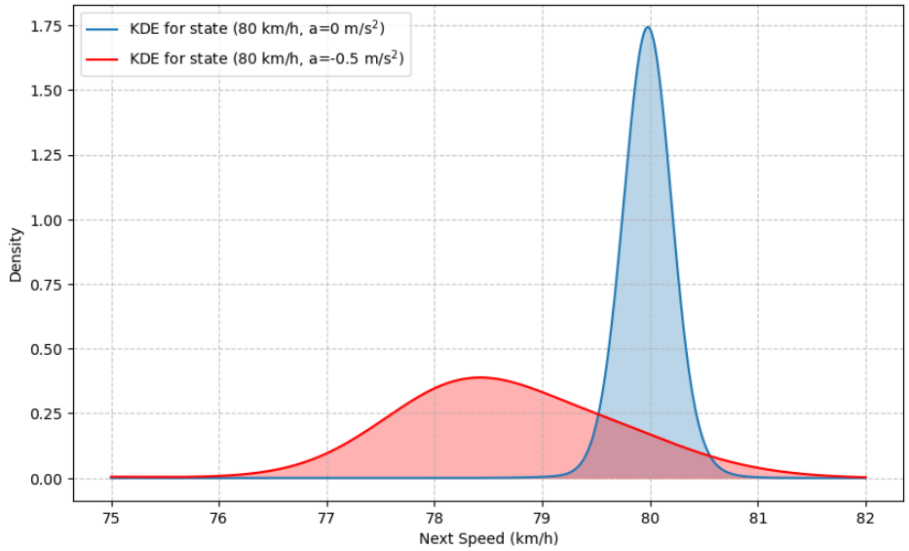


Figure 4.3: PDF comparison between states ($v = 80$ km/h, $a = 0$m/s ^{2}) and ($v = 80$ km/h, $a = -0.5$m/s ^{2})

Figure 4.3 illustrates and compares the PDFs for states ($v = 80$ km/h, $a = 0$ m/s^2) and ($v = 80$ km/h, $a = -0.5$ m/s^2). As shown in the figure, the PDFs exhibit distinct characteristics, with the state of ($v = 80$ km/h, $a = 0$ m/s^2) assigning higher probibalistic density to maintain a cruising speed, whereas the state of ($v = 80$ km/h, $a = -0.5$ m/s^2) assigns higher probibalistic density to reduced next speed values, indicating deceleration.

The construction process of the KDE-MC cycle was implemented to closely follow the original MC method. Starting in the initial state of ($v = 0$ km/h, $a = 0$ m/s^2), the next speed was sampled from the KDE of this initial idling state. The corresponding acceleration was computed based on the speed difference between the current speed and the newly sampled speed. Having the new speed state (v_1, a_1), the Euclidean distance between this state and the existing KDEs was calculated.

If the state was closer than a set distance threshold to one of the existing KDEs, then that KDE was selected to be the new current KDE. The next speed value (v_2) was then sampled from the PDF of that KDE. If more than one existing KDE was closer than the distance threshold, the one with minimum distance was selected. If no state was within the distance threshold, then a weighted interpolation between the four closest states was made. The search for close KDEs and weighted interpolation strategy can be seen in algorithm 2. The distance threshold utilised in this implementation was 0.1.

Algorithm 2 Closest KDE or KDE interpolation

function GET_INTERPOLATED_KDE(kde_models, new_state, threshold = 0.1)
  distances ← EUCLIDEAN_DISTANCE(kde_models, new_state)
  min_distance ← MIN(distances)
  if min_distance < threshold **then**
    closest_kde ← kde_models[ARGMIN(distances)]
    return closest_kde
  end if
  closest_indices ← ARGSORT(distances)[: 4]
  closest_kdes ← kde_models[ARGSORT(distances)[: 4]
  weights ← $\frac{1}{\text{closest\_distances}}$
  weights ← $\frac{\text{weights}}{\text{SUM(weights)}}$
  combined_data ← SAMPLE_AND_COMBINE(closest_kdes, weights)
  interpolated_kde ← GAUSSIAN_KDE(combined_data)
  return interpolated_kde
end function

Thus, in each iteration of sampling a new state, either an existing KDE was selected and sampled from or an interpolated one was used. An example of an interpolated KDE is shown in Figure 4.4.

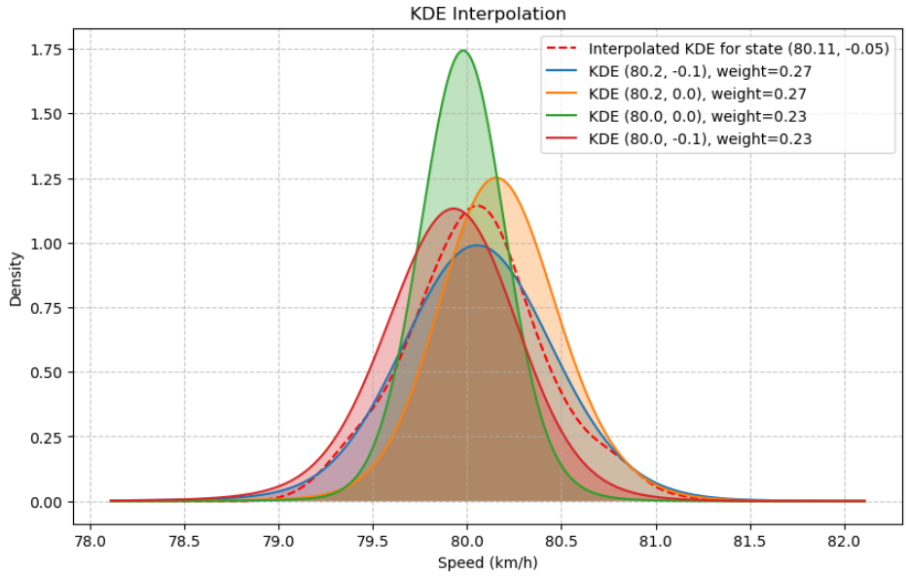


Figure 4.4: KDE interpolation between the four closest states to (v = 80.11 km/h, $a = -0.05\text{m/s}^2$), using a distance threshold of 0.1.

This iterative process continued until the termination criteria of at least one-hour cycle length and the cycle ending at zero speed were met. The complete generation process of a KDE-MC candidate cycle is described in the pseudo-code in Algorithm 3.

Just like the original MC implementation, the KDE-MC is also a stochastic process generating different candidate cycles in each run. Therefore, one hundred candidate

Algorithm 3 KDE-MC DC construction

function MC(TPM, $v_{\text{current}} = 0, a_{\text{current}} = 0$)
  DC $\leftarrow (v_{\text{current}}, a_{\text{current}})$
  $\text{KDE}_{\text{current}} \leftarrow$ KDE_models$[v_{\text{current}}][a_{\text{current}}]$
  while True **do**
    $v_{\text{next}} \leftarrow$ SAMPLE_SPEED($\text{KDE}_{\text{current}}$)
    $a_{\text{next}} \leftarrow \frac{(v_{\text{next}} - v_{\text{current}})}{3.6}$
    DC $\leftarrow (v_{\text{next}}, a_{\text{next}})$
    $v_{\text{current}} \leftarrow v_{\text{next}}$
    $a_{\text{current}} \leftarrow a_{\text{next}}$
    if len(DC) ≥ 3600 and $v_{\text{current}} = 0$ **then**
      Break
    end if
    $\text{KDE}_{\text{current}} \leftarrow$ GET_KDE(KDE_models, $(v_{\text{current}}, a_{\text{current}})$)
  end while
  return DC
end function

cycles were generated with this method as well.

When analysing the KDE-MC generated candidate cycles, it became clear that speed changes were excessive and aggressive. Due to this, the RDs for acceleration and deceleration correlated CPs were high. A Savitsky-Golay filter was applied to all non-idling speeds in an attempt to mitigate that issue. The filtering aimed to smooth out some of the speed transitions and analyse its impact on overall candidate cycle representativeness. The filter was applied with a window size of five and a polynomial order of one. The impact of the smoothing was then evaluated by comparing the smoothed and non-smoothed cycles.

4.3.3 Micro-Trip Clustering

This section describes the Micro-Trip Clustering (MTC) implementation. The method aimed to capture real driving behavior by clustering MTs based on statistical features, sampling a set of MTs from these clusters, and then splicing them together into a complete DC. All technical details are described below.

1. MT segmentation: All trips were segmented into MTs using the standard zero-speed cutoff threshold. However, contrary to the most standard implementations, idling was also included in the MTs. To include idling, idling preceding vehicle movement was included. MTs shorter than ten seconds, excluding initial idling, were filtered out.

2. Calculating MT features: For each MT, average speed and average acceleration $(a > 0 m/s^2)$ were calculated.

3. MT clustering: Using K-means clustering, MTs were clustered by their average speed and average acceleration features. The optimal number of clusters was determined using the Elbow method for evaluating WCSS.

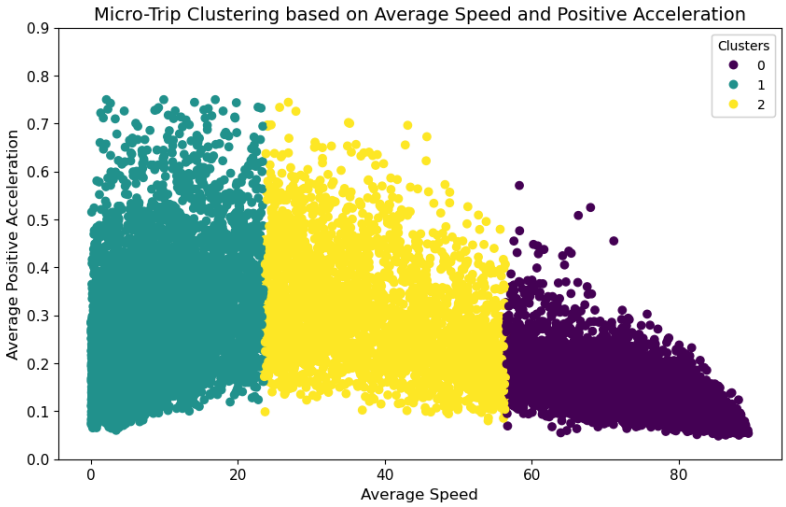


Figure 4.5: Clustering results of the MTC method.

4. DC generation: Random MTs were selected from each cluster based on the cluster frequency distribution, meaning more MTs were selected from a cluster of larger size. The logic behing this method was that the distribution of the MTs included in the generated cycle should align with the MT distribution of the operational data. Selected MTs were then spliced together to form a complete DC. This process was repeated one hundred times to construct one hundred candidate cycles. The complete MTC DC construction process is described in pseudo-code in Algorithm 4.

Algorithm 4 MTC DC construction

```
function MTC(c_labels, all_MTs, total_cycles = 100, MTs_per_cycle = 5)
    cluster_sizes ← COUNT(c_labels)
    total_MTs ← SUM(cluster_sizes)
    selection_sizes ← ROUND((cluster_sizes/total_MTs) × MTs_per_cycle))
    all_cycles ← []
    for i ← 1 to total_cycles do
        cycle_MTs ← []
        for each cluster_id, size in selection_sizes do
            cluster_MTs ← filter all_MTs by cluster_id
            if size > 0 and cluster_MTs is not empty then
                selected_MTs ← randomly sample size from cluster_MTs
                Add selected_MTs to cycle_MTs
            end if
        end for
        final_cycle ← CONCATENATE(cycle_MTs)
        add final_cycle to all_cycles
    end for
    return all_cycles
end function
```

4.3.4 Micro-Trip Clustering with Dimensionality Reduction and Random Forest Refinement

This section details another method based on clustering and sampling of MTs, the Micro-Trip Clustering with dimensionality reduction and Random Forest Refinement (MTC-RF) method. Just as with the MTC method, driving data was segmented into MTs, the MTs included idling, and MTs shorter than ten seconds were filtered out. For each MT, fourteen CPs were computed for describing the MTs, as listed in Table 4.3.

Characteristic Parameter	**Unit**
Time-related	
Driving time	s
Acceleration time	s
Deceleration time	s
Cruising time	s
Stopping time	s
Speed-related	
Maximum speed	km/h
Mean speed	km/h
Mean speed (excluding stops)	km/h
Speed standard deviation	km/h
Acceleration-related	
Maximum acceleration	m/s^2
Mean acceleration	m/s^2
Minimum deceleration	m/s^2
Mean deceleration	m/s^2
Acceleration standard deviation	m/s^2

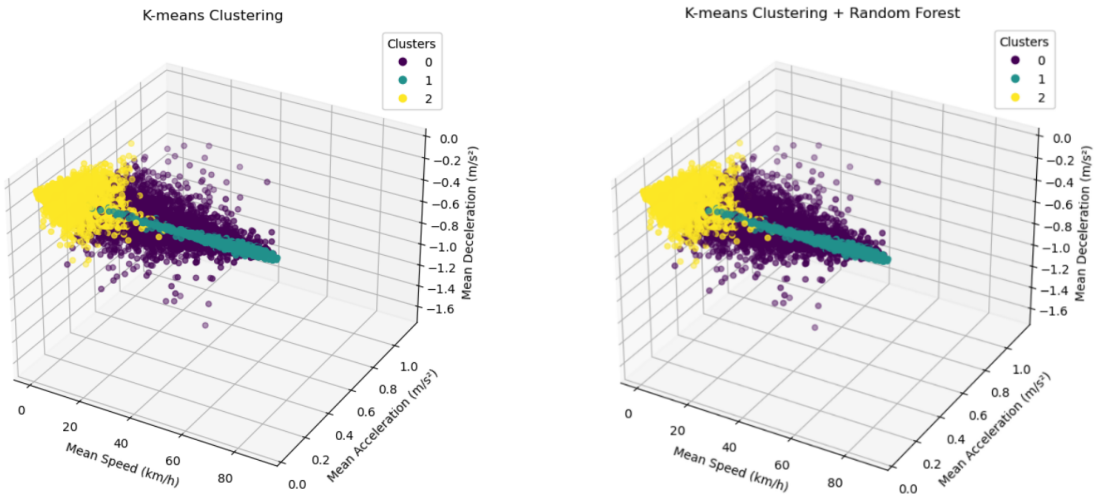
Table 4.3: CPs computed for each MT

Since these parameters were highly correlated, a necessary step was to reduce the dimensions to avoid redundancy and the curse of dimensionality. Two techniques were used for this: PCA and kPCA. Both techniques were applied, and their results were compared to determine the most effective approach. A threshold was set to achieve at least 85% explained variance. Using PCA, four components were selected, achieving a cumulative variance contribution of 89%. With kPCA, which can capture nonlinear patterns, four components were also used, achieving a cumulative variance contribution of 90%.

Subsequently, K-means clustering was applied to group the MTs into three clusters that capture distinct driving patterns. The three clusters represented low-speed, medium-speed, and high-speed categories. To refine the clustering, an RF classifier was utilised trying to enhance the optimisation process. A selection of 500 MTs from each cluster was made based on their proximity to the cluster center and their distance from other cluster centers. The selected samples were used to train the RF

model, which was then used to predict the cluster assignments for the remaining MTs.

In Figure 4.6, the clustering results of K-means are compared with those of the K-means with the RF approach, visualised by plotting the average speed, acceleration, and deceleration. The comparison shows that using the RF classifier did not result in significant visual changes to the cluster assignments. Additionally, Table 4.4 presents the compactness and separation metrics for the original and refined clustering. While the separation was slightly improved with the use of RF, it also marginally increased the compactness. This indicated a minor trade-off between these two metrics and overall minimal difference was achieved with the refinement.



(a) K-means Clustering

(b) K-means + RF Clustering

Figure 4.6: Comparison of clustering methods: (a) K-means, (b) K-means + RF

Method	**Compactness**	**Separation**
K-means	1.80163	4.64897
K-means + RF	1.80180	4.64966

Table 4.4: Comparison of compactness and separation values for K-means and K-means + RF methods

For the DC construction, a duration threshold of 4000 seconds was set. To then determine how many MTs from each of the three clusters were to be included in the generated DC, the duration threshold was multiplied by the time proportion of each cluster. The formula can be seen in Equation 4.2.

$$D_i = \left(\frac{D_{c_i}}{\sum_{j=1}^{n} D_{c_j}} \right) \times D_{\text{total}}, \tag{4.2}$$

where D_i represents the duration of cluster i, D_{c_i} is the total driving time of cluster i, and D_{total} the total desired duration of the driving cycle. Thus, D_i determines the minimum duration of MTs that should be included from cluster i. That could result in one or several MTs depending on the length of the MTs selected.

Additionally, the PCC for each MT with its respective cluster was computed. The larger the PCC is, the better the MT represents the overall characteristics of its cluster. Based on this coefficient and the duration of each MT, MTs from each cluster were selected to meet the proportional target durations. Finally, candidate cycles were generated by connecting the selected MTs. The construction process of MTC-RF candidate cycles is described in pseudo-code in Algorithm 5.

Algorithm 5 MTC-RF DC construction

```
function MTC-RF(all_MTs, total_cycles = 100, dc_target_duration = 4000)
    Select Best MTs:
        best_MTs ← Top 25 MTs per cluster from all_MTs based on:
          - Highest Pearson correlation scores
          - Duration ≤ 5000 seconds
    cl_durations ← SUM(group by cluster, driving_time from all_MTs)
    total_duration ← SUM(cl_durations)
    cl_target_durations ← (cl_durations/total_duration) × dc_target_duration
    all_cycles ← []
    for cycle_num ← 1 to total_cycles do
        cycle_MTs ← []
        cl_durations ← 0 for each cluster
        for each cluster_id in cl_target_durations do
            best_cluster_MTs ← filter best_MTs by cluster_id
            while cl_durations[cluster_id] < cl_target_durations[cluster_id] do
                selected_MT ← randomly sample 1 MT from best_cluster_MTs
                Add selected_MT to cycle_MTs
                cluster_durations[cluster_id] ← duration of selected_MT
                Remove selected_MT from best_cluster_MTs
            end while
        end for
        dc ← shuffle and concatenate cycle_MTs
        Add dc to all_cycles
    end for
    return all_cycles
end function
```

To compare the effectiveness of PCA, kPCA, and RF within the method, candidate cycles were generated for four different method variations: (1) PCA dimension reduction, (2) kPCA dimension reduction, (3) PCA dimension reduction with RF refinement, and (4) kPCA dimension reduction with RF refinement. One hundred candidate cycles were generated for each method variation.

4.3.5 Kinematic Segment Clustering with Support Vector Machine refinement

This section covers the implementation of the KSC-SVM method. The first step of the implementation was to segment the driving data into KSs. The segmentation criteria were the following:

- Acceleration: $a \geq 0.15\,\mathrm{m/s^2}$
- Deceleration: $a \leq -0.15\,\mathrm{m/s^2}$
- Cruising: $v \geq 2\,\mathrm{km/h}$ and $-0.15\,\mathrm{m/s^2} < a < 0.15\,\mathrm{m/s^2}$
- Idling: $v < 2\,\mathrm{km/h}$ and $-0.15\,\mathrm{m/s^2} < a < 0.15\,\mathrm{m/s^2}$

where a is the acceleration and v is the speed.

Each data point was classified into a KS type according to the criteria above. This classification organised the driving data into distinct segments. A unique segment identifier was assigned to each series of consecutive data points sharing the same segment type. This identifier marked the start and end of each KS.

After the segmentation of driving data CPs were computed for each segment. The following eight CPs were computed: (1) the maximum speed, (2) minimum speed, (3) average speed, (4) standard deviation of speed, (5) maximum acceleration, (6) maximum deceleration, (7) average acceleration, and (8) standard deviation of acceleration.

PCA was used for dimensionality reduction of the CPs. The number of principal components was determined by computing the cumulative variance contribution ratio and obtaining the principal components exceeding 90%. The principal component variance contribution ratio and the cumulative variance contribution ratio are listed in Table 4.5. Using three principal components, the cumulative contribution ratio reached 96%, and thus three components were used for the clustering analysis.

Number of principal component	**Variance contribution rate (%)**	**Cumulative variance contribution rate (%)**
1	39.61209	39.61209
2	34.88043	74.43234
3	21.51063	96.00316
4	3.81902	99.82218
5	0.10066	99.92284
6	0.07443	99.99727
7	0.00169	99.99896
8	0.00103	1.00000

Table 4.5: PCA results summary

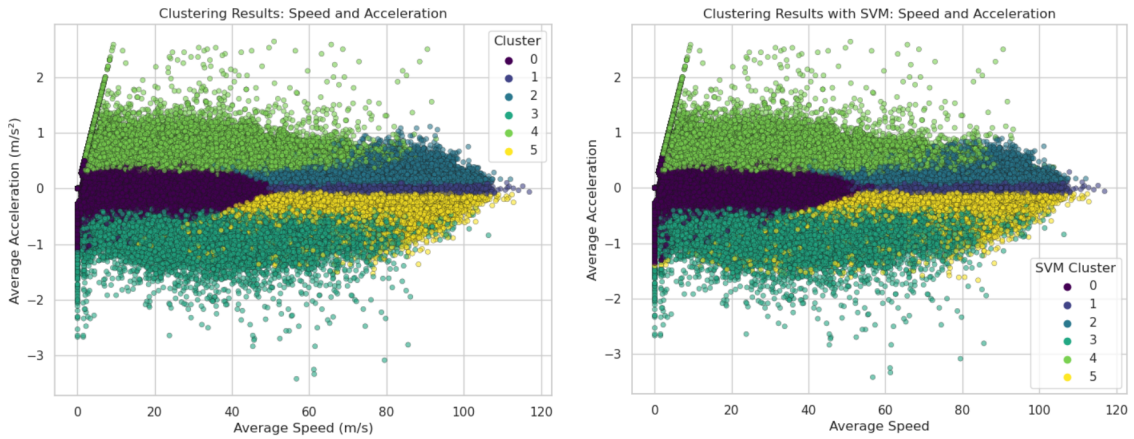
The principal components were then clustered using K-means clustering. To determine the optimal number of clusters, k, a combination of the Elbow Method for WCSS and gap statistics was used for evaluating cluster compactness. Combining these approaches provided a more comprehensive evaluation, as each method offers different insights into the clustering quality. The chosen k value of six was determined based on this combined approach. The characteristics of the clustered driving segments are displayed in Table 4.6.

Cluster	**Average speed**	**Average acceleration**	**Description**
1	20.74	-0.01262	Low-speed segments, idling and cruising
2	79.42	0.00085	High-speed segments, steady cruising
3	77.43	0.19185	High-speed segments, slight acceleration
4	39.03	-0.65728	Medium-speed segments, strong deceleration
5	31.78	0.50569	Medium-speed segments, strong acceleration
6	77.19	-0.20282	High-speed segments, slight acceleration

Table 4.6: Cluster characteristics based on average speed and average acceleration.

To potentially further refine the clustering results the SVM algorithm was utilised. SVM offers advantages such as its robustness and ability to effectively classify complex, non-linear data. The process began by creating a training set for the SVM algorithm. The most representative segments were selected based on their proximity to the cluster centers and their distance from other cluster centers. This method ensured that the training set consisted of segments that accurately reflected the core characteristics of each cluster and being distinct from other clusters.

In Figure 4.7, the clustering results of K-means are compared with those of the K-means with SVM approach, visualised by plotting the average speed and acceleration. The comparison indicates that using an SVM classifier did not lead to notable visual changes in the cluster assignments. Additionally, Table 4.7 presents the compactness and separation metrics for both methods. Both separation and compactness were slightly improved with the use of SVM, suggesting a small enhancement in clustering quality.



(a) K-means Clustering

(b) K-means + SVM Clustering

Figure 4.7: Comparison of clustering methods: (a) K-means, (b) K-means + SVM

Method	**Compactness**	**Separation**
K-means	1.704	5.410
K-means + SVM	1.699	5.641

Table 4.7: Comparison of compactness and separation values for K-means and K-means + SVM methods

To construct the DC, the target length for the cycle was first determined to be 4000 seconds. Time proportions for each of the six clusters based on the driving data were then calculated. These proportions determine how much time each cluster would contribute to the overall driving cycle. The time allocated to each cluster was computed by multiplying the proportions by the target length of the cycle and was as follows: 733 s, 2577 s, 214 s, 114 s, 154 s, and 208 s.

The construction of the DC involved several steps. An initial segment was randomly sampled that was less or equal to five seconds long and had a start speed of zero. This segment served as the starting point for the DC. Subsequent segments were selected based on their proximity to the cluster center to ensure representativeness. Additionally, the difference between the final speed of the previous segment and the initial speed of the current segment had to be less than one kilometer per hour to maintain continuity. Segments were selected without replacement to avoid repetition. Moreover, the remaining duration for each cluster was also considered when selecting the next segment to ensure that the target durations for each cluster were met as closely as possible.

Segments were linked together until the total DC duration of 4000 seconds was reached, resulting in a complete candidate cycle. If the final speed of the cycle was greater than zero, a deceleration path was found and added to bring the cycle to a

complete stop. The construction process of KSC-SVM candidate cycles is described in pseudo-code in Algorithm 6.

Just as with the MTC-RF implementation, the KSC-SVM implementation was evaluated with different method variations. These variations were: (1) KSC without SVM refinement, and (2) KSC with SVM refinement. Thirty candidate cycles were generated for each method variation. This number was significantly lower than the one hundred candidates generated with other methods. The number of candidate cycles generated was reduced due to the computational load of the KS methods.

Algorithm 6 KSC-SVM DC Construction

```
Input: segments, cl_durations, dc_target_duration, total_cycles
Output: all_cycles

total_duration ← SUM(cl_durations)
target_durations ← (cl_durations / total_duration) * dc_target_duration
all_cycles ← []

for cycle_num ← 1 to total_cycles do
    dc ← []
    current_duration ← 0
    used_durations ← 0 for each cluster
    Select initial segment:
        Filter for segments with:
          - Duration ≤ 5 seconds
          - Start speed ≈ 0
        initial_segment ← Randomly select one segment from the filtered segments
        Add initial_segment to dc
        Update current_duration and used_durations

    while current_duration < dc_target_duration do
        last_end_speed ← End speed of the last segment in dc
        Filter candidates:
            candidates ← Filter segments for:
              - Start speed within 1 km/h of last_end_speed
              - Not used
              - Remaining duration in target ≥ 0
            Sort candidates by remaining duration and min distance to cluster center
        selected_candidate ← First candidate in sorted candidates
        Add selected_candidate to dc
        Update current_duration and used_durations
    end while

    final_speed ← End speed of the last segment in dc
    if final_speed > 0 then
        Add deceleration path:
            Filter segments for deceleration path
            Add deceleration segments to dc
    end if
    Add dc to all_cycles
end for
return all_cycles
```

4.4 Statistical Evaluation of Candidate Cycles

The resulting candidate cycles of each cycle generation method were statistically evaluated in two different categories:

1. The same nineteen CPs that were used in the statistical selection of trips were calculated for each candidate cycle. The RD of each CP of the candidate cycles were calculated and analysed. The candidate cycle with the overall lowest mean CP RD was selected to be the best CP representation of the operational data.

2. The SAPD of each candidate cycle was computed and evaluated against the SAPD of the operational data. The SAPD evaluation was performed using two statistical measures:

    - EMD: The primary metric for SAPD evaluation, quantifying the minimum cost to transform one probability distribution into another. When measuring dissimilarity, EMD considers both the amount of probability mass that needs to be moved and the distance it needs to be moved, which makes it less sensitive to small shifts in the distributions. This makes EMD reliable for comparing SAPDs.

    - RD: Included for making comparisons with other studies, RD measures the exact differences between distributions and is easier to interpret. However, RD treats all deviations equally and does not account for the positional differences which can give high difference scores for distributions that are simply shifted. This limitation means that RD might not as effectively capture the overall similarity between distributions when the differences are subtle.

Combining EMD with RD improved the validity of the result. While RD as a metric is more straightforward to understand and is useful for comparison, EMD's consideration of the overall distribution made it the primary evaluation metric. An SAPD achieving both low RD and low EMD was assured to be a good representation. Ultimately it was the candidate cycle with the lowest EMD score that was evaluated as the most accurate SAPD representation.

The CPs of the operational data which all candidate cycles are evaluated against can be seen in Table 4.8. The SAPD of the operational data can be seen in Figure 4.8, which all candidate cycles are evaluated against.

	Operational Data
V_{avg}	64.81 km/h
V_{max}	101.4 km/h
Vr_{avg}	72.30 km/h
V_{std}	27.38 km/h
Vr_{std}	20.26 km/h
p_a	11.09%
p_d	9.771%
p_c	68.43%
p_i	10.71%
a_{max}	2.028 m/s^2
d_{max}	2.666 m/s^2
a_{avg}	0.3572 m/s^2
d_{avg}	0.4136 m/s^2
a_{std}	0.1984 m/s^2
d_{std}	0.3020 m/s^2
p_{lv}	16.37%
p_{hv}	65.98%
SSPK	0.1053 event/km
PKE	1336 ms^{-2}

Table 4.8: CPs of the operational data.

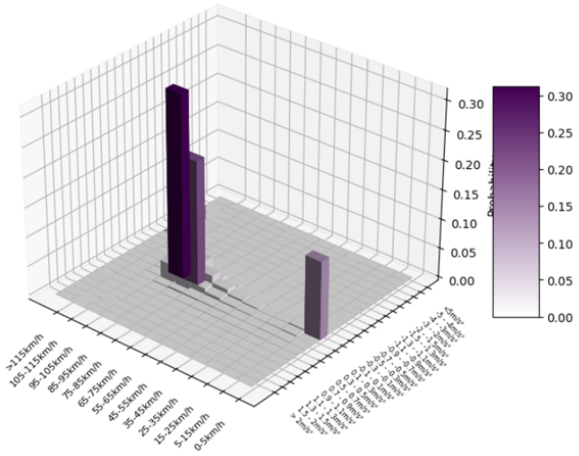


Figure 4.8: SAPD plot of the operational data.

In addition to comparing the implemented DC generation methods, a simulation of the VECTO cycle was analysed to provide a baseline cycle for measuring performance. The VECTO cycle is a standard cycle against which all HDVs are tested, making it a valuable reference. The CP analysis of the VECTO cycle is shown in Table 4.9 and Figure 4.9. The SAPD plot of the VECTO cycle is presented in Figure 4.10.

	VECTO
V_{avg}	78.61 km/h
V_{max}	90.00 km/h
Vr_{avg}	79.74 km/h
V_{std}	15.33 km/h
Vr_{std}	12.18 km/h
p_a	5.994%
p_d	4.991%
p_c	87.69%
p_i	1.330%
a_{max}	1.000 m/s^2
d_{max}	0.9996 m/s^2
a_{avg}	0.3483 m/s^2
d_{avg}	0.4251 m/s^2
a_{std}	0.2237 m/s^2
d_{std}	0.2992 m/s^2
p_{lv}	3.008%
p_{hv}	89.54%
SSPK	0.03993 event/km
PKE	564 ms^{-2}

Table 4.9: CPs of the MC candidate cycle with lowest mean CP RD.

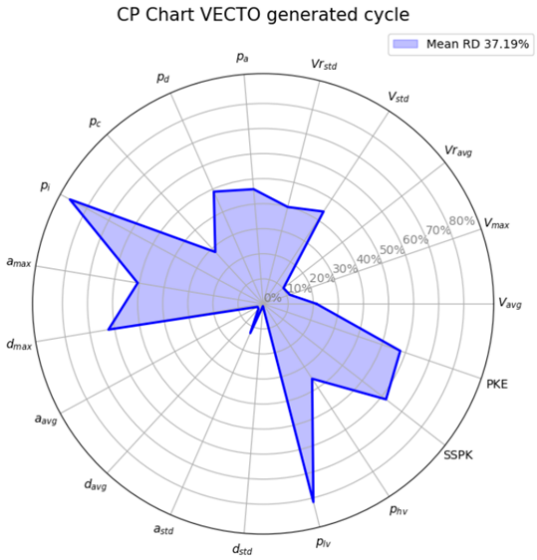


Figure 4.9: CP RDs of the VECTO cycle compared to the operational data.

EMD (10^{-4})	**SAPD RD**
37.76	48.62%

Table 4.10: SAPD metrics Vecto cycle.

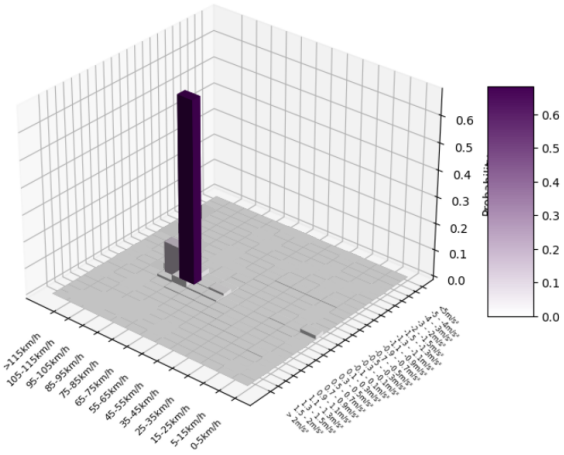


Figure 4.10: SAPD plot of Vecto cycle.

The analysis of the VECTO cycle revealed that it is a poor representation of the targeted operational data, with a mean CP RD of 37.19%, as well as EMD score of 37.76×10^{-4} and SAPD RD of 48.62%. These metrics indicate that while the VECTO cycle is efficient for general analysis of HDV performance, it is not well-tuned for specific long-haul vehicle populations or customers.

5

Results

In this chapter, the results of the implemented data analytics methods are detailed. The results are structured according to the construction model methodologies: SAS methods, MT methods, and KS methods. This structure highlights how the construction methods differ in their ability to generate representative DCs. Within each construction method, the results of the specific data analytical approaches are presented individually, emphasising the differences between each distinct approach.

5.1 Speed Acceleration State Methods

This section presents the results achieved with SAS methods. Two main methods were evaluated: MC and KDE-MC. The performance of each method is detailed in Sections 5.1.1 and 5.1.2.

5.1.1 Markov Chain

The MC candidate cycle with the best mean CP RD is shown in Figure 5.1. Its corresponding CPs are listed in Table 5.1, and CP RDs are displayed in Figure 5.2. Key results include:

- RD is below 10% for 13 out of 19 CPs, indicating high accuracy in replicating most of the operational data CPs.
- The mean CP RD of 10.44% is the second lowest among all candidate cycles from all methods.
- The Start-Stop Per Kilometer (SSPK) CP is significantly lower than that of the operational data, resulting in an SSPK RD of about 75.80%. This significantly increased the mean CP RD.
- As depicted in the candidate cycle plot, the MC method can generate cycles that mimic the driving behavior expected in long-haul driving, with significant periods spent in high speed cruising and a low percentage of time spent in medium and low speeds.

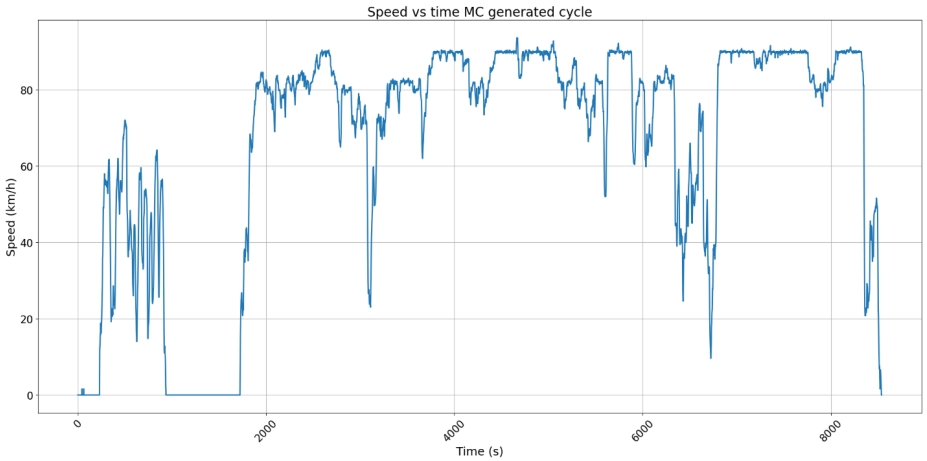


Figure 5.1: MC candidate cycle with the lowest mean CP RD.

	MC
V_{avg}	66.16 km/h
V_{max}	93.60 km/h
Vr_{avg}	75.10 km/h
V_{std}	30.29 km/h
Vr_{std}	19.23 km/h
p_a	11.06%
p_d	9.919%
p_c	67.16%
p_i	11.86%
a_{max}	1.700 m/s^2
d_{max}	2.100 m/s^2
a_{avg}	0.3367 m/s^2
d_{avg}	0.3841 m/s^2
a_{std}	0.2010 m/s^2
d_{std}	0.3036 m/s^2
p_{lv}	13.43%
p_{hv}	66.92%
SSPK	0.02549 event/km
PKE	1237 ms^{-2}

Table 5.1: CPs of the MC candidate cycle with lowest mean CP RD.

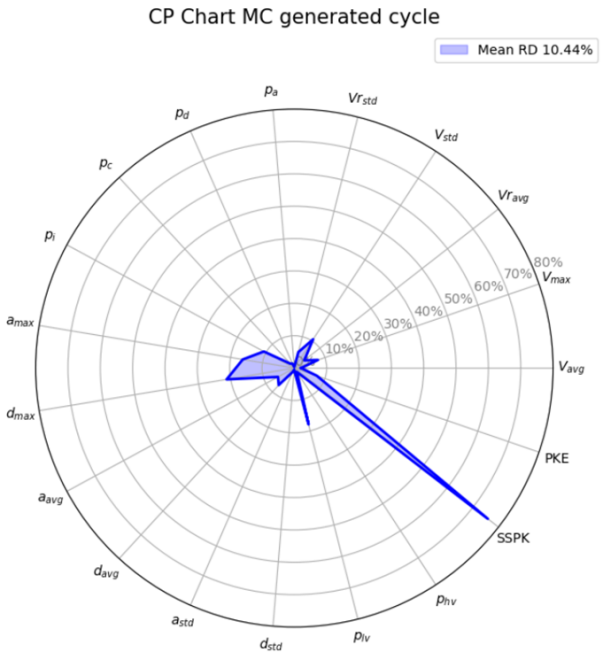


Figure 5.2: CP RDs of the MC candidate cycle with lowest mean CP RD.

The best MC SAPD candidate cycle is shown in Figure 5.3, with its SAPD metrics listed in Table 5.2 and the SAPD plot in Figure 5.4. The SAPD of the MC cycle achieved an EMD cost of 14.35×10^{-4} and a SAPD RD of 17.73%. This result is relatively average compared to all other implementations. In general, the SAPD appears to be a decent representation of the operational data, capturing the general driving pattern of long-haul operations, characterised by a high proportion of time spent cruising at high speeds.

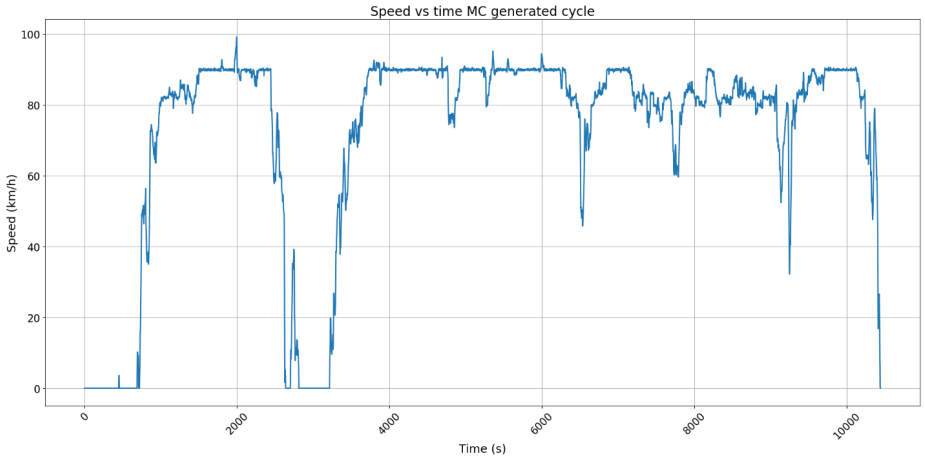


Figure 5.3: MC Candidate cycle with the best SAPD representation.

EMD (10^{-4})	**SAPD RD**
14.35	17.73%

Table 5.2: SAPD metrics MC candidate cycle with the best SAPD representation.

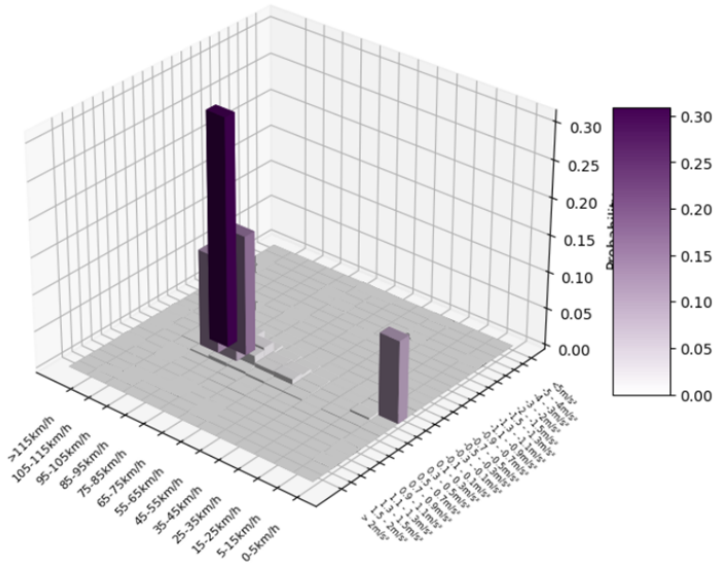


Figure 5.4: SAPD plot of the MC candidate cycle with the best SAPD representation.

5.1.2 Kernel Density Estimation - Markov Chain

The KDE-MC method, while being computationally more intensive than the original MC due to its more complex sampling technique, initially also showed large deviations in acceleration and deceleration correlated CPs. The KDE-MC candidate cycle with the lowest mean CP RD is shown in Figure 5.5, with its corresponding CPs listed in Table 5.3, and CP RDs displayed in Figure 5.6.

These high RDs are largely impacted by speed changes being too aggressive, resulting in a reduced percentage of time spent cruising and an increased percentage of time spent in acceleration and deceleration. The RDs of percentage acceleration (p_a) and percentage deceleration (p_d) are both more than 80%. The excessive acceleration

also impacts the Positive Kinetic Energy (PKE) CP, being 2107 ms^{-2} compared to 1336 ms^{-2} in the operational data.

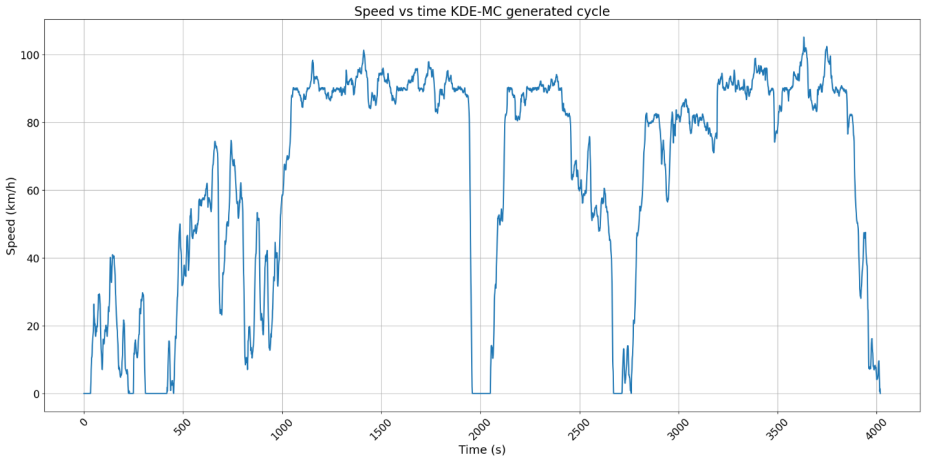


Figure 5.5: KDE-MC candidate cycle with the lowest mean CP RD.

	KDE-MC
V_{avg}	63.59 km/h
V_{max}	105.2 km/h
Vr_{avg}	68.74 km/h
V_{std}	32.82 km/h
Vr_{std}	28.46 km/h
p_a	20.63 %
p_d	18.54 %
p_c	53.52 %
p_i	7.315 %
a_{max}	2.626 m/s^2
d_{max}	2.222 m/s^2
a_{avg}	0.3690 m/s^2
d_{avg}	0.4139 m/s^2
a_{std}	0.2329 m/s^2
d_{std}	0.3377 m/s^2
p_{lv}	17.96 %
p_{hv}	57.78 %
SSPK	0.09860 event/km
PKE	2107 ms^{-2}

Table 5.3: CPs of the KDE-MC candidate cycle with the lowest mean CP RD.

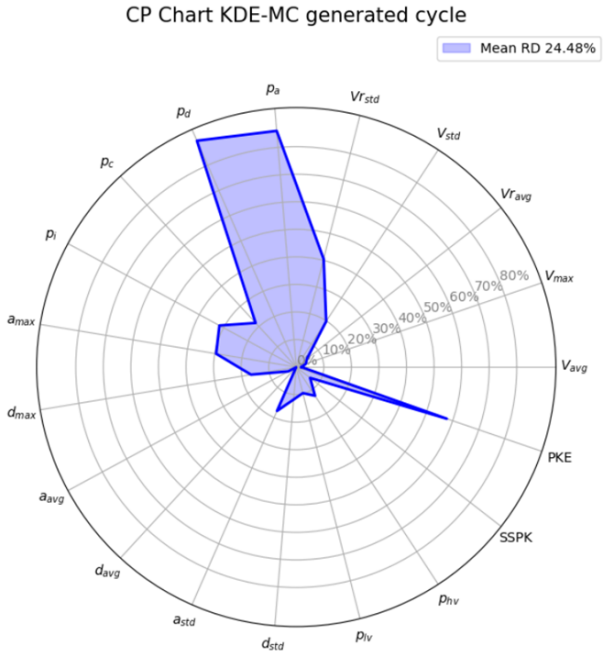


Figure 5.6: CP RDs of the KDE-MC candidate cycle with lowest mean CP RD.

The results of the Savitsky-Golay smoothed cycle showed significant CP improvements. The smoothed candidate cycle is shown in Figure 5.7, with its CPs listed in Table 5.4, and its CP RDs compared to the original KDE-MC in Figure 5.8. The smoothed cycle reduced the percentage of time spent in acceleration and deceleration, reduced the PKE, and increased the percentage of cruise. This reduced these CP RDs quite

significantly which ended up decreasing the mean CP RD by approximately 9.48 percentage points from 24.48% down to 15%.

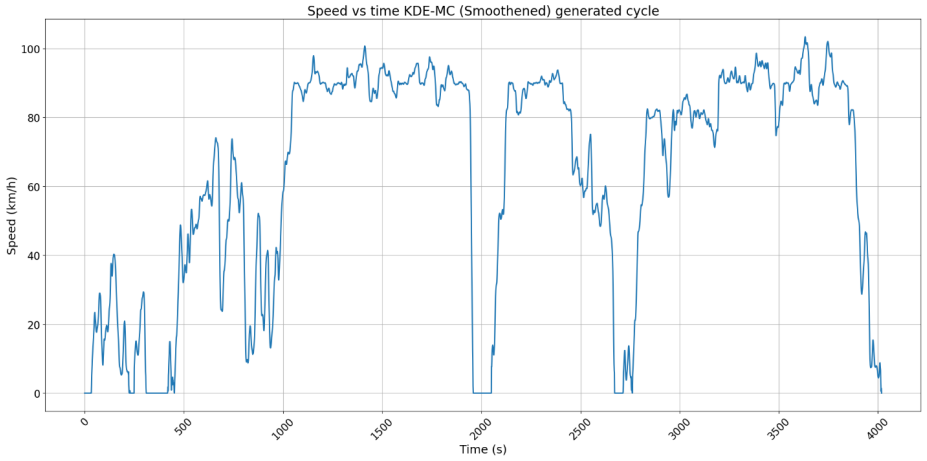


Figure 5.7: KDE-MC (Smoothed) candidate cycle with the lowest mean CP RD.

	KDE-MC (Smoothed)
V_{avg}	63.59 km/h
V_{max}	103.3 km/h
Vr_{avg}	68.74 km/h
V_{std}	32.79 km/h
Vr_{std}	28.43 km/h
p_a	16.22 %
p_d	13.54 %
p_c	62.93 %
p_i	7.315 %
a_{max}	2.145 m/s^2
d_{max}	2.044 m/s^2
a_{avg}	0.3496 m/s^2
d_{avg}	0.4288 m/s^2
a_{std}	0.1957 m/s^2
d_{std}	0.3308 m/s^2
p_{lv}	17.87 %
p_{hv}	57.68 %
SSPK	0.09860 event/km
PKE	1571 ms^{-2}

Table 5.4: CPs of the smoothed KDE-MC candidate cycle with the lowest mean CP RD.

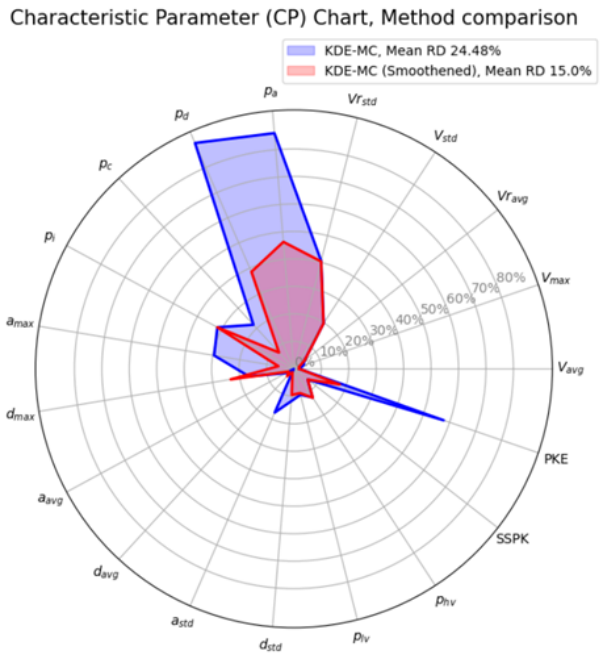


Figure 5.8: CP comparison between the non-smoothed and smoothed KDE-MC candidate cycles with lowest mean CP RD.

The smoothing filter demonstrated similar performance improvements in terms of SAPD. The SAPD metrics of the non-smoothed and smoothed candidate cycles with the best SAPD representation are compared in table 5.5, their SAPD plots can be seen in Figure 5.9 and 5.10. The smoothed cycle achieved an EMD cost of

11.98×10^{-4}, about one-third lower compared to the 18.20×10^{-4} of the non-smoothed cycle. In terms of SAPD RD, the decrease was about 7.61 percentage points, which is also a significant decrease.

The SAPD metrics of the smoothed KDE-MC candidate cycle showed a slight improvement compared to the original MC method. Having an EMD cost of 11.98×10^{-4} compared to 14.35×10^{-4} of the original MC, and an SAPD RD of 15.81% compared to 17.73% of the original MC. This indicates that while the KDE-MC method had issues with excessive speed transitions, it still managed to capture much of the underlying patterns commonly seen in long-haul driving. The smoothed candidate cycle with the best SAPD representation can be seen in Figure 5.11.

	KDE-MC	**KDE-MC (Smoothed)**
EMD (10^{-4})	18.20	11.98
SAPD RD	23.43%	15.81%

Table 5.5: SAPD metrics KDE-MC candidate cycles with the best SAPD representation.

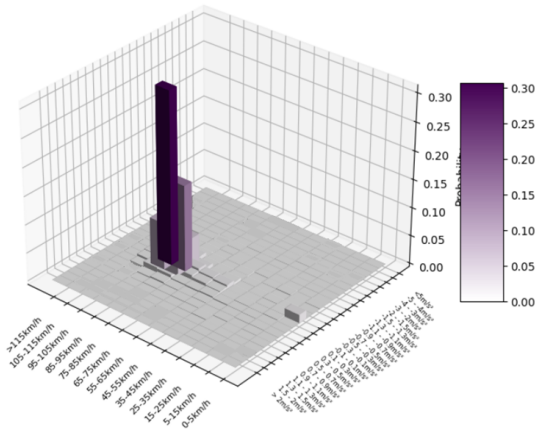


Figure 5.9: SAPD plot KDE-MC candidate cycle with the best SAPD representation.

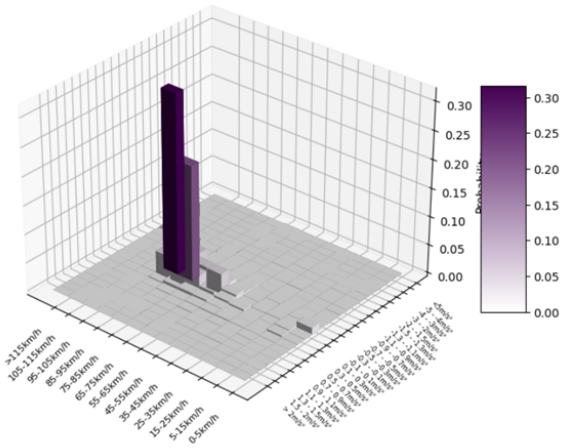


Figure 5.10: SAPD plot KDE-MC (Smoothed) candidate cycle with the best SAPD representation.

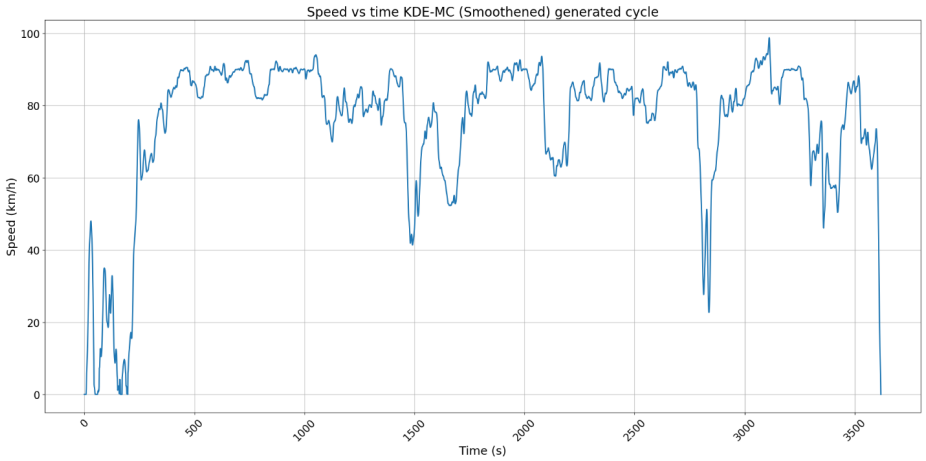


Figure 5.11: KDE-MC (Smoothed) candidate cycle with the best SAPD representation.

5.2 Micro-Trip Methods

The following sections detail the results achieved using MT-methods. This includes two main methods: MTC and MTC-RF, which are two different MT clustering methods with varying computational complexity. The results achieved with the MTC method are detailed in Section 5.2.1 , and the MTC-RF results in Section 5.2.2.

5.2.1 Micro-Trip Clustering

The MTC candidate cycle with best mean CP RD is shown in Figure 5.12, with its corresponding CP values listed in Table 5.6, and CP RDs displayed in Figure 5.13. Key results include:

- The mean CP RD of 10.14% is the lowest among all methods.
- RD is below 10% for 12 out of 19 CPs, and below 20% for 18 out of 19 CPs, indicating strong overall accuracy in replicating most CPs of the operational data.
- SSPK of 0.04676 is significantly lower than the 0.1053 of the operational data, resulting in a SSPK RD of 55.60%, significantly increasing the overall mean CP RD.

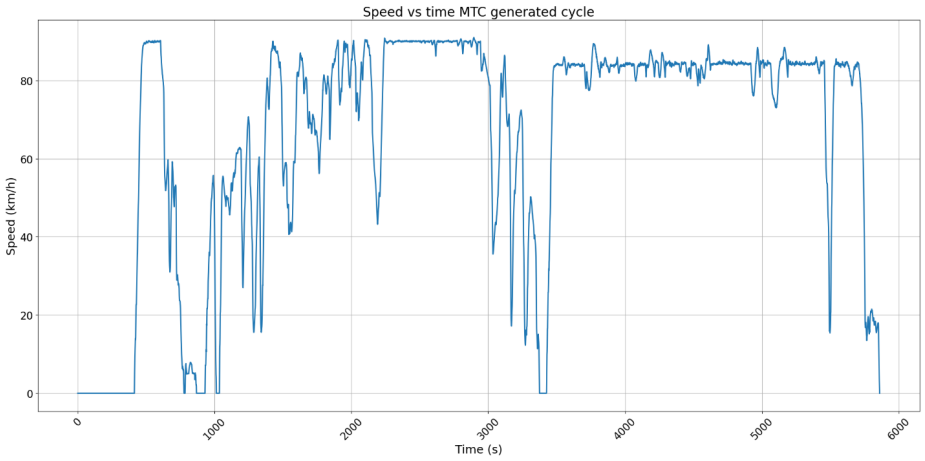


Figure 5.12: MTC candidate cycle with the lowest mean CP RD.

	MTC
V_{avg}	65.68 km/h
V_{max}	90.90 km/h
Vr_{avg}	72.65 km/h
V_{std}	29.99 km/h
Vr_{std}	22.10 km/h
p_a	13.07 %
p_d	10.70 %
p_c	66.73 %
p_i	9.503 %
a_{max}	1.667 m/s^2
d_{max}	2.222 m/s^2
a_{avg}	0.3595 m/s^2
d_{avg}	0.4265 m/s^2
a_{std}	0.1974 m/s^2
d_{std}	0.3366 m/s^2
p_{lv}	15.07%
p_{hv}	65.57%
SSPK	0.04676 event/km
PKE	1422 ms^{-2}

Table 5.6: CPs of the MTC candidate cycle with lowest mean CP RD.

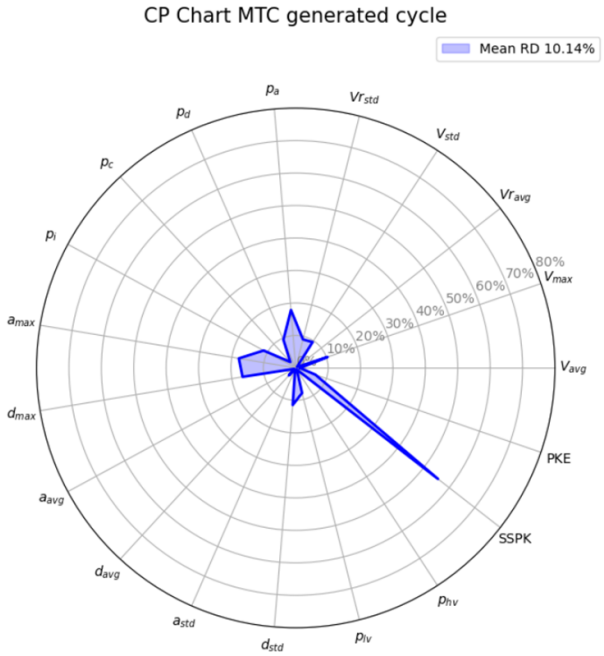


Figure 5.13: CP RDs of the MTC candidate cycle with lowest mean CP RD.

The best MTC SAPD candidate cycle is shown in Figure 5.14, with its SAPD metrics listed in Table 5.7 and SAPD plot in Figure 5.15. The EMD score of 9.420×10^{-4} and SAPD RD of 12.21% are substantially lower than those achieved with any of the SAS methods. The cycle is a good overall SAPD representation of the operational data, characterised by a high probability of high-speed cruising, a low probability of medium speed, and a relatively high probability of very low speed or idling.

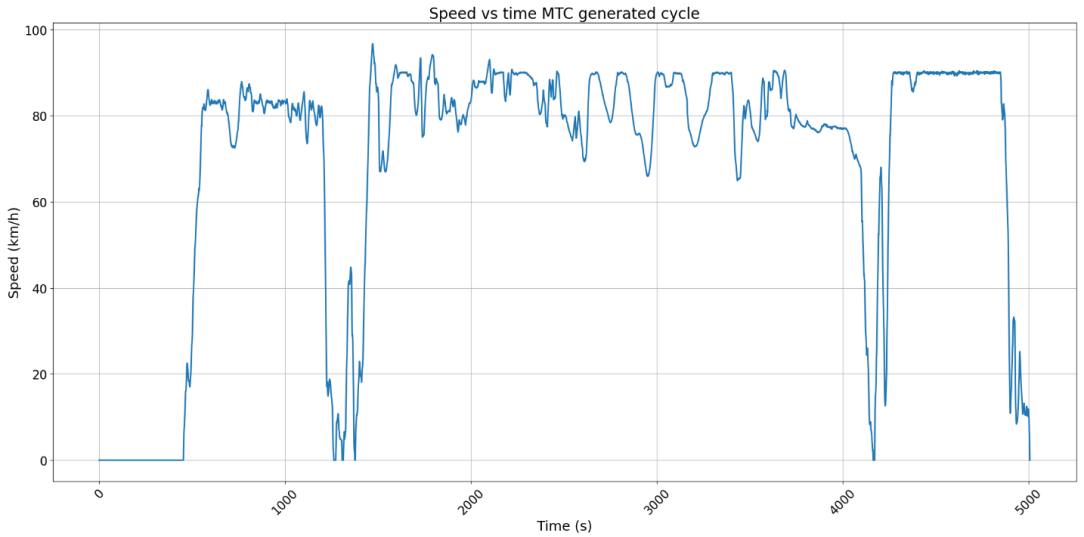


Figure 5.14: MTC candidate cycle with the best SAPD representation.

EMD (10^{-4})	**SAPD RD**
9.420	12.21%

Table 5.7: SAPD metrics MTC candidate cycle with the best SAPD representation.

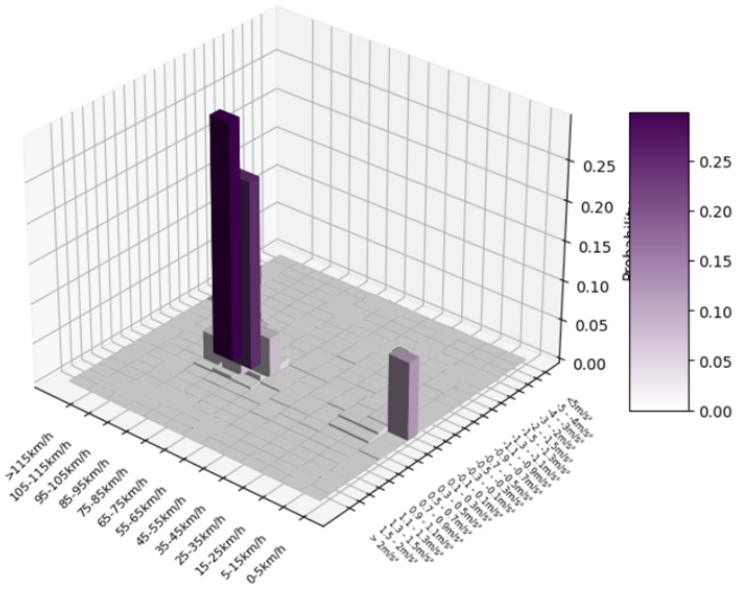


Figure 5.15: SAPD plot of the MTC candidate cycle with best SAPD representation.

5.2.2 Micro-Trip Clustering with Dimensionality Reduction and Random Forest Refinement

As detailed in the methodology chapter, the MTC-RF method was implemented with two different dimension reduction techniques, PCA and kPCA. Therefore, the results are reported using both techniques. Moreover, results with and without RF refinement were also compared. Hence, results from four method variations are reported: MTC-NORF (PCA), MTC-NORF (kPCA), MTC-RF (PCA), and

MTC-RF (kPCA).

In table 5.8, the CPs of each method variation's best candidate in terms of mean CP RD are shown, and their CP RDs are compared and displayed in Figure 5.16. As can be concluded from the table and figure, each of these methods produces very similar best CP candidates. MTC-NORF (kPCA) and MTC-RF (kPCA) produce the same candidate cycle. Across all the method variations, there is little difference in CP values, except for the MTC-RF (PCA) candidate having a slightly too low maximum acceleration compared to the other candidates resulting in a high RD.

Key findings from the CPs are:

- None of the method variations achieved a mean CP RD that was lower than the original MTC method.
- All the method variations showed very high RD for the SSPK CP, with three of the methods having the same SSPK and the fourth deviating from the other three by only 7×10^{-6} event/km.
- The other CPs across the method variations are also very similar, indicating robustness in replicating similar outputs with less variation compared to MTC or SAS methods, which have more randomness in their sampling.

	MTC-NORF (PCA)	**MTC-NORF (kPCA)**	**MTC-RF (PCA)**	**MTC-RF (kPCA)**
V_{avg}	67.81 km/h	67.34 km/h	66.58 km/h	67.34 km/h
V_{max}	99.10 km/h	99.10 km/h	91.90 km/h	99.10 km/h
Vr_{avg}	74.51 km/h	74.47 km/h	72.36 km/h	74.47 km/h
V_{std}	27.71 km/h	28.04 km/h	26.48 km/h	28.04 km/h
Vr_{std}	18.58 km/h	18.39 km/h	18.56 km/h	18.39 km/h
p_a	10.06%	9.950%	9.449%	9.950%
p_d	8.386%	8.392%	8.922%	8.392%
p_c	72.61%	72.11%	73.69%	72.11%
p_i	8.945%	9.544%	7.940%	9.544%
a_{max}	1.528 m/s^2	1.528 m/s^2	1.111 m/s^2	1.528 m/s^2
d_{max}	1.889 m/s^2	1.889 m/s^2	2.555 m/s^2	1.889 m/s^2
a_{avg}	0.3240 m/s^2	0.3218 m/s^2	0.3479 m/s^2	0.3218 m/s^2
d_{avg}	0.3860 m/s^2	0.3801 m/s^2	0.3821 m/s^2	0.3801 m/s^2
a_{std}	0.1946 m/s^2	0.1951 m/s^2	0.1989 m/s^2	0.1951 m/s^2
d_{std}	0.3055 m/s^2	0.2966 m/s^2	0.2827 m/s^2	0.2966 m/s^2
p_{lv}	11.26%	11.37%	11.18%	11.37%
p_{hv}	70.94%	70.20%	71.14%	70.20%
SSPK	0.01979 event/km	0.01972 event/km	0.01965 event/km	0.01972 event/km
PKE	1164 ms^{-2}	1156 ms^{-2}	1130 ms^{-2}	1156 ms^{-2}

Table 5.8: CPs of the MTC-NORF and MTC-RF candidate cycles with lowest mean CP RD.

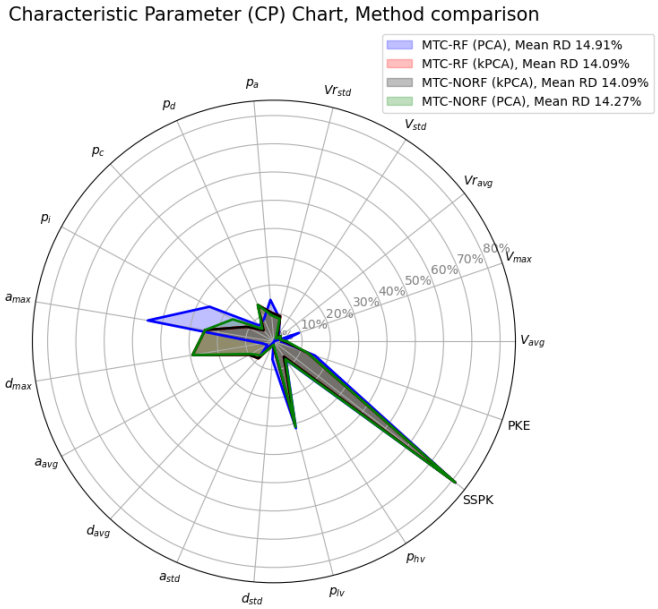


Figure 5.16: CP RD comparison between the MTC-NORF and MTC-RF candidate cycles with lowest mean CP RD.

Due to the similarity of the candidate cycle CPs, it is difficult to determine any method variation as significantly superior in terms of generating CP-accurate cycles. However, due to the technique of selecting MTs with high PCC, the results are much more likely to be consistent when generating new candidates. Therefore, these results are not expected to deviate significantly if more candidate cycles were to be generated. The candidate cycle with the lowest mean CP RD, the one generated by the MTC-NORF (kPCA) and MTC-RF (kPCA) method, is shown in Figure 5.17.

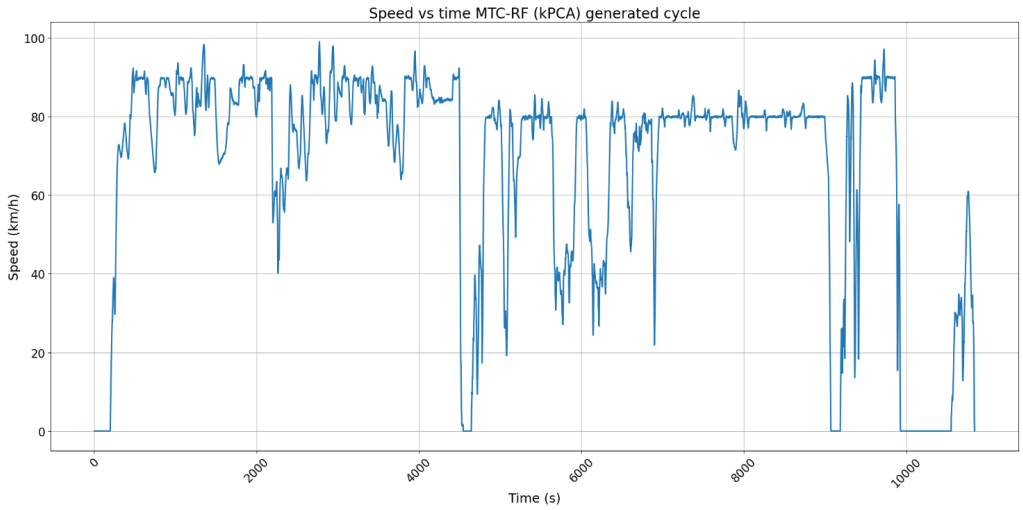


Figure 5.17: MTC-NORF (kPCA) and MTC-RF (kPCA) candidate cycle with lowest mean CP RD.

In Table 5.9, the SAPD metrics of all the best SAPD representations of the method

variations are listed. As shown in the table, the MTC-NORF (PCA) method achieved the lowest EMD cost of 7.642×10^{-4}, while the MTC-RF (PCA) method achieved the second lowest at 7.779×10^{-4}, indicating slightly better performance with PCA reduction compared to kPCA reduction in terms of SAPD representation. Interestingly, the best EMD cost and the best SAPD RD candidates are not the same. In terms of SAPD RD, the MTC-RF (PCA) achieved a slightly lower SAPD RD of 9.935% while the MTC-NORF (PCA) had a SAPD RD of 10.13%.

	MTC-NORF (PCA)	**MTC-NORF (kPCA)**	**MTC-RF (PCA)**	**MTC-RF (kPCA)**
EMD (10^{-4})	7.642	8.549	7.779	8.549
SAPD RD	10.13%	10.99%	9.935%	10.99%

Table 5.9: SAPD metrics MTC-NORF and MTC-RF candidate cycles with best SAPD representation.

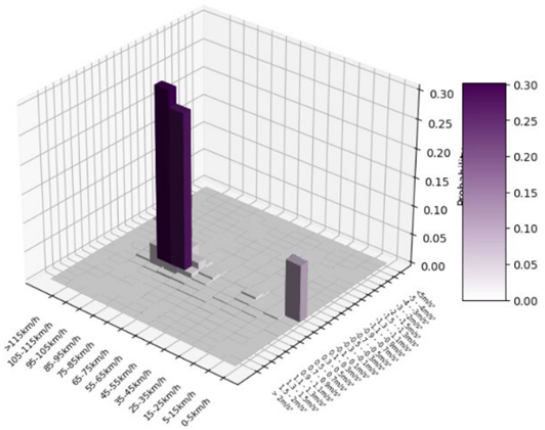


Figure 5.18: SAPD plot MTC-NORF (PCA) candidate cycle with best SAPD representation.

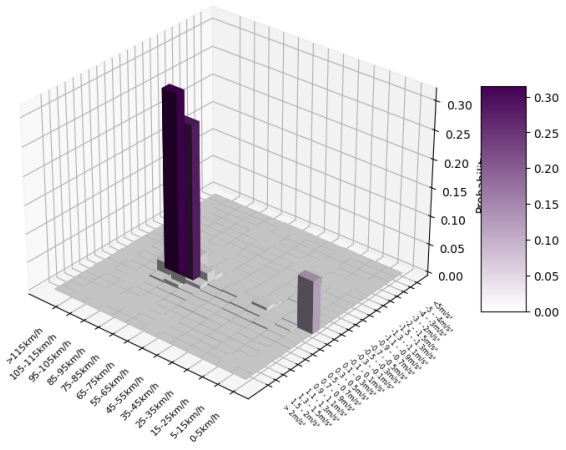


Figure 5.19: SAPD plot MTC-RF (PCA) candidate cycle with best SAPD representation.

Figures 5.18 and 5.19 show the SAPD plots of the MTC-NORF (PCA) and MTC-RF (PCA) candidate cycles next to each other. Both are very good SAPD representations. However, since the EMD metric is a better measurement for the dissimilarity between distributions, the MTC-NORF (PCA) candidate is determined to be a slightly better SAPD representation. It is the best SAPD representation across all implemented methods, not only just among MT methods. The candidate cycle is shown in Figure 5.20.

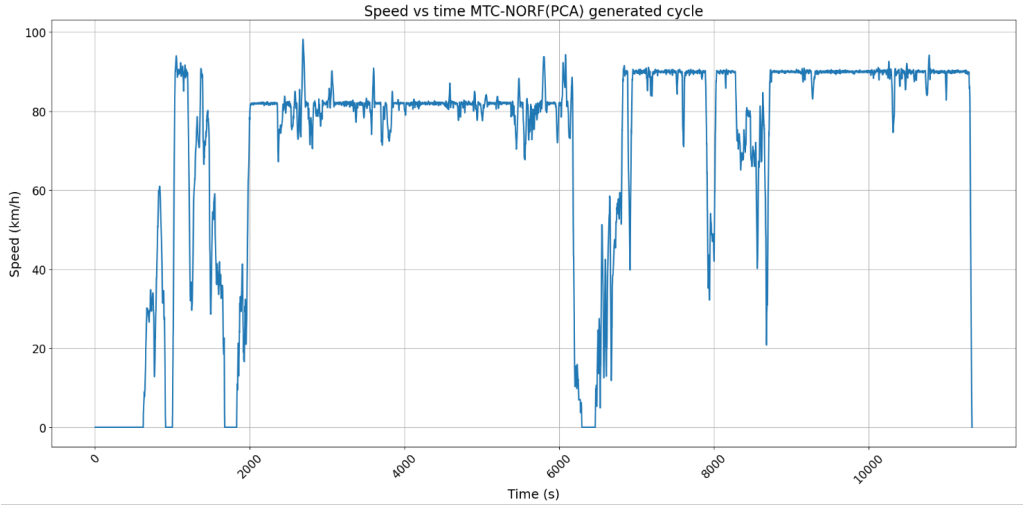


Figure 5.20: MTC-NORF (PCA) candidate cycle with the best SAPD representation.

Another notable finding regarding the candidate cycles achieved with the MTC-RF and MTC-NORF method variations is that they tend to be significantly longer than most other candidates generated with other methods. The candidate cycles shown previously in this section are both longer than 10 000 seconds, this can be compared to for instance the candidate cycles shown for the original MTC cycle which were both shorter than 6000 seconds.

5.3 Kinematic Segment Methods

As detailed in the methodology chapter the KS method was implemented with PCA dimensionality reduction, clustering segments based on the principal components, and then refining the clusters using SVM. The results from this method are reported through two different method variations: Kinematic Segment Clustering (KSC), which is the method excluding the SVM cluster refinement, and Kinematic Segment Clustering with SVM refinement (KSC-SVM).

In Table 5.8, the CPs of the KSC and KSC-SVM best candidates in terms of mean CP RD are shown. Their CP RDs are compared and displayed in Figure 5.21. As can be seen, none of the method variations perform very well, achieving mean CP RDs of 16.9% and 15.79% respectively. The KSC method achieved especially poor RD for the parameters max deceleration (d_{max}), max acceleration (a_{max}), percentage idling (p_i), and positive kinetic energy (PKE), all of which had RDs above 30%.

	KSC	KSC-SVM
V_{avg}	66.25 km/h	60.50 km/h
V_{max}	88.60 km/h	88.50 km/h
Vr_{avg}	71.50 km/h	64.13 km/h
V_{std}	28.79 km/h	28.56 km/h
Vr_{std}	22.78 km/h	25.14 km/h
p_a	7.800 %	9.829 %
p_d	6.904 %	9.644 %
p_c	78.15 %	74.93 %
p_i	7.149 %	5.597 %
a_{max}	1.332 m/s^2	1.610 m/s^2
d_{max}	1.806 m/s^2	1.806 m/s^2
a_{avg}	0.2846 m/s^2	0.3291 m/s^2
d_{avg}	0.3588 m/s^2	0.3865 m/s^2
a_{std}	0.1732 m/s^2	0.2043 m/s^2
d_{std}	0.2684 m/s^2	0.2745 m/s^2
p_{lv}	15.49 %	16.49 %
p_{hv}	75.77 %	56.01 %
SSPK	0.1062 event/km	0.04128 event/km
PKE	852.2 ms^{-2}	1133 ms^{-2}

Table 5.10: CPs of the KS candidate cycles with lowest mean CP RD.

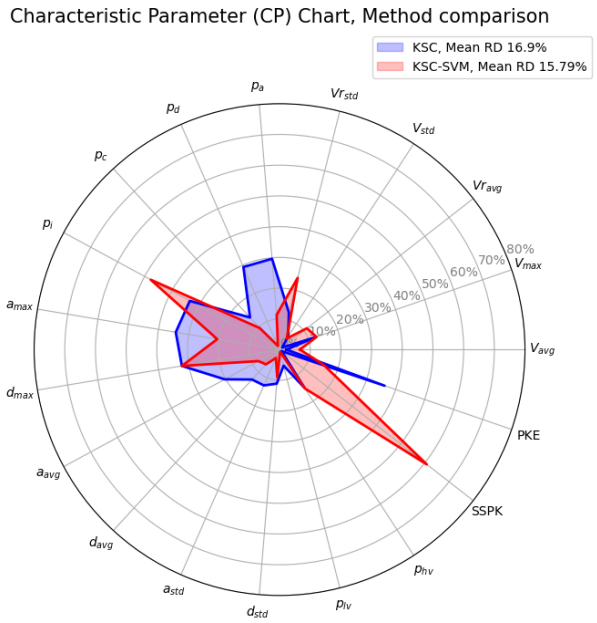


Figure 5.21: CP RDs comparison between the KS candidate cycles with the lowest mean CP RD.

The KSC-SVM method variation does not perform quite as poorly as the KSC, with nine out of nineteen CP RDs below 10%, showing that nearly half of the parameters were well-represented. However, having five out of nineteen CP RDs above 20% and three above 30%, with the highest reaching 60.80%, indicates significant issues as well. These high RDs demonstrate that the method struggles in several areas, resulting in an overall less reliable performance. The KSC-SVM candidate with the lowest mean CP RD is shown in Figure 5.22.

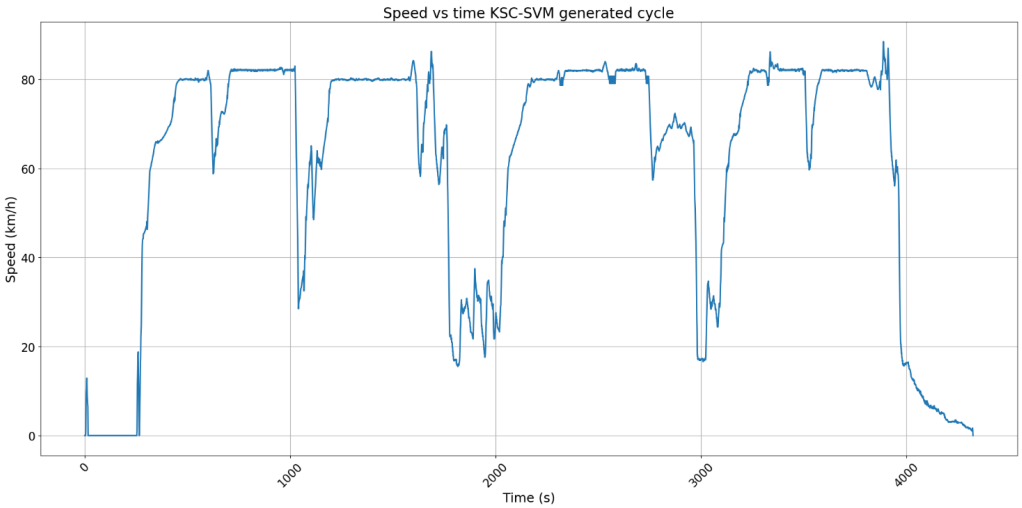


Figure 5.22: KSC-SVM candidate cycle with the lowest mean CP RD.

In Table 5.11, the SAPD metrics of best SAPD representations achieved with the KSC

and KSC-SVM method variations are listed. Their corresponding SAPD plots can be seen in Figures 5.23 and 5.24. Evident by the metrics the SAPD representations with the methods were poor, with EMD costs of 35.13×10^{-4} and 32.16×10^{-4}, and SAPD RDs of 44.72% and 41.45%. These were the highest costs and deviations compared to all the other methods. The KSC-SVM SAPD candidate can be seen in Figure 5.25.

	KSC	**KSC-SVM**
EMD (10^{-4})	35.13	32.16
SAPD RD	44.72%	41.45%

Table 5.11: SAPD metrics KS candidate cycles with the best SAPD representation.

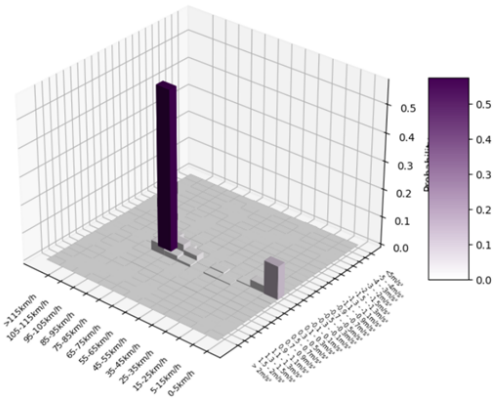


Figure 5.23: SAPD plot KSC candidate cycle with the best SAPD representation.

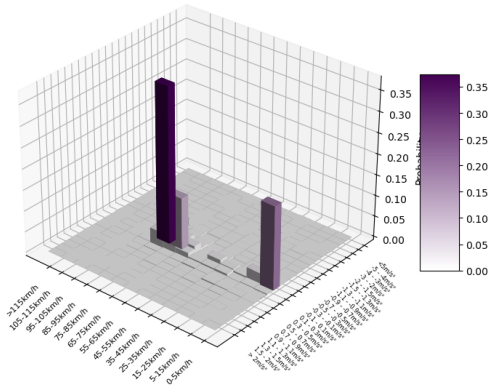


Figure 5.24: SAPD plot KSC-SVM candidate cycle with the best SAPD representation.

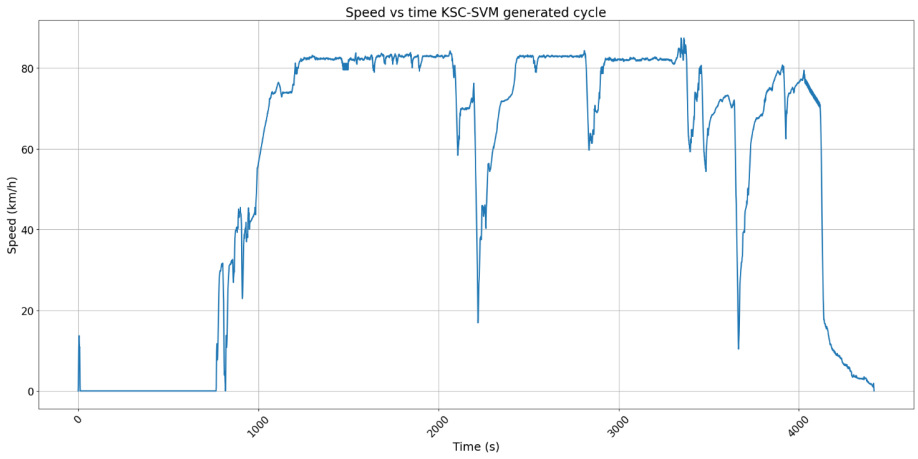


Figure 5.25: KSC-SVM candidate cycle with the best SAPD representation.

The CP and SAPD evaluations of the KS methods indicate that they are significantly less effective at generating representative cycles compared to the other methods.

Furthermore, analysing the cycles in the plots reveals that these methods tend to concatenate strange kinematic sequences, leading to unnatural acceleration and deceleration patterns not typically seen in real world driving patterns.

5.4 Characteristic Parameter Summary

In Table 5.12, the CP and CP RDs for a selection of the best-performing method variations in each construction model category are listed. Their CP RDs are also compared and displayed in Figure 5.26. The table clearly shows that the most basic implementations, MC and MTC, despite their simplicity, achieved the best overall CP candidates in terms of mean CP RD. Their CP means of 10.44% and 10.14% are more than three percentage points lower than the next closest candidate cycle, the cycle achieved by MTC-NORF (kPCA) and MTC-RF (kPCA) methods which achieved a mean CP RD of 14.09%.

Consistent with the original MC method and all the MT methods, there were considerable difficulties with the SSPK CP. Out of their CP candidates, only the MTC candidate cycle achieved an SSPK RD below 70% with a value of 55.60%, while the other methods had an SSPK RD of 75.80% or higher. Despite the high RDs for the SSPK parameter, these methods still turned out to have the best mean CP RDs. Excluding the SSPK parameter, the MC, MTC, and MRC-NORF (kPCA)/MRC-RF (kPCA) CP candidates would have had mean CP RDs of 6.809%, 7.614%, and 10.36%, indicating their strong overall accuracy in representing most CPs of the operational data.

		Operational data	VECTO	MC	MTC	KDE-MC (Smooth)	MTC-RF/MTC-NORF (kPCA)	KSC-SVM
V_{avg}	Data	64.81 km/h	78.61 km/h	66.16 km/h	65.68 km/h	63.59 km/h	67.34 km/h	60.50 km/h
	RD	0 %	21.29 %	2.075 %	1.335 %	1.889 %	3.893 %	6.655 %
V_{max}	Data	101.4 km/h	90.00 km/h	93.60 km/h	90.90 km/h	103.3 km/h	99.10 km/h	88.50 km/h
	RD	0 %	11.24 %	7.691 %	10.36 %	1.922 %	2.269 %	12.72 %
Vr_{avg}	Data	72.30 km/h	79.74 km/h	75.10 km/h	72.65 km/h	68.74 km/h	74.47 km/h	64.13 km/h
	RD	0 %	10.29 %	3.875 %	0.475 %	4.929 %	3.003 %	11.296 %
V_{std}	Data	27.38 km/h	15.33 km/h	30.29 km/h	29.99 km/h	32.79 km/h	28.04 km/h	28.56 km/h
	RD	0 %	44.02 %	10.62 %	9.505 %	19.74 %	2.409 %	4.314 %
Vr_{std}	Data	20.26 km/h	12.18 km/h	19.23 km/h	22.10 km/h	28.43 km/h	18.39 km/h	25.14 km/h
	RD	0 %	39.91 %	5.104 %	9.073 %	40.31 %	9.223 %	24.06 %
p_a	Data	11.09 %	5.994 %	11.06 %	13.07 %	16.22 %	9.950 %	9.83 %
	RD	0 %	45.94 %	0.2857 %	17.88 %	46.33 %	10.25 %	11.35 %
p_d	Data	9.771 %	4.991 %	9.919 %	10.70 %	13.54 %	8.392 %	9.64 %
	RD	0 %	48.92 %	1.518 %	9.488 %	38.53 %	14.11 %	1.30 %
p_c	Data	68.43 %	87.69 %	67.16 %	66.73 %	62.93 %	72.11 %	74.93 %
	RD	0 %	28.14 %	1.852 %	2.485 %	8.043 %	5.384 %	9.50 %
p_i	Data	10.71 %	1.330 %	11.86 %	9.503 %	7.315 %	9.544 %	5.60 %
	RD	0 %	87.59 %	10.74 %	11.29 %	31.71 %	10.91 %	47.76 %
a_{max}	Data	2.028 m/s²	1.000 m/s²	1.700 m/s²	1.667 m/s²	2.145 m/s²	1.528 m/s²	1.610 m/s²
	RD	0 %	50.69 %	16.17 %	17.82 %	5.747 %	24.67 %	20.60 %
d_{max}	Data	2.666 m/s²	0.9996 m/s²	2.100 m/s²	2.222 m/s²	2.044 m/s²	1.889 m/s²	1.806 m/s²
	RD	0 %	62.50 %	21.23 %	16.65 %	23.35 %	29.14 %	32.28 %
a_{avg}	Data	0.3572 m/s²	0.3483 m/s²	0.3367 m/s²	0.3595 m/s²	0.3496 m/s²	0.3218 m/s²	0.3291 m/s²
	RD	0 %	2.484 %	5.755 %	0.6334 %	2.124 %	9.921 %	7.87 %
d_{avg}	Data	0.4136 m/s²	0.4251 m/s²	0.3840 m/s²	0.4265 m/s²	0.4288 m/s²	0.3801 m/s²	0.3865 m/s²
	RD	0 %	2.777 %	7.139 %	3.133 %	3.671 %	8.101 %	6.54 %
a_{std}	Data	0.1984 m/s²	0.2237 m/s²	0.2010 m/s²	0.1974 m/s²	0.1957 m/s²	0.1951 m/s²	0.2043 m/s²
	RD	0 %	12.73 %	1.308 %	0.5043 %	1.359 %	1.661 %	2.96 %
d_{std}	Data	0.3020 m/s²	0.2992 m/s²	0.3036 m/s²	0.3366 m/s²	0.3308 m/s²	0.2966 m/s²	0.2745 m/s²
	RD	0 %	0.9083 %	0.5193 %	11.45 %	9.549 %	1.787 %	9.08 %
p_{lv}	Data	16.37 %	3.008 %	13.43 %	15.07 %	17.87 %	11.37 %	16.49 %
	RD	0 %	81.62 %	17.93 %	7.954 %	9.149 %	30.53 %	0.74 %
p_{hv}	Data	65.98 %	89.54 %	66.91 %	65.57 %	57.68 %	70.20 %	56.01 %
	RD	0 %	35.70 %	1.417 %	0.6248 %	12.59 %	6.401 %	15.11 %
SSPK	Data	0.1053 event/km	0.03993 event/km	0.02549 event/km	0.04676 event/km	0.09860 event/km	0.01972 event/km	0.0413 event/km
	RD	0 %	62.09 %	75.80 %	55.60 %	6.381 %	81.28 %	60.80 %
PKE	Data	1336 ms⁻²	563.6 ms⁻²	1237 ms⁻²	1422 ms⁻²	1571 ms⁻²	1156 ms⁻²	1133 ms⁻²
	RD	0 %	57.80 %	7.376 %	6.438 %	17.60 %	12.76 %	15.16 %
Mean RD		**0 %**	**37.19 %**	**10.44 %**	**10.14 %**	**15.00 %**	**14.09 %**	**15.79 %**

Table 5.12: Comparison of CP values and CP RDs achieved with different methods.

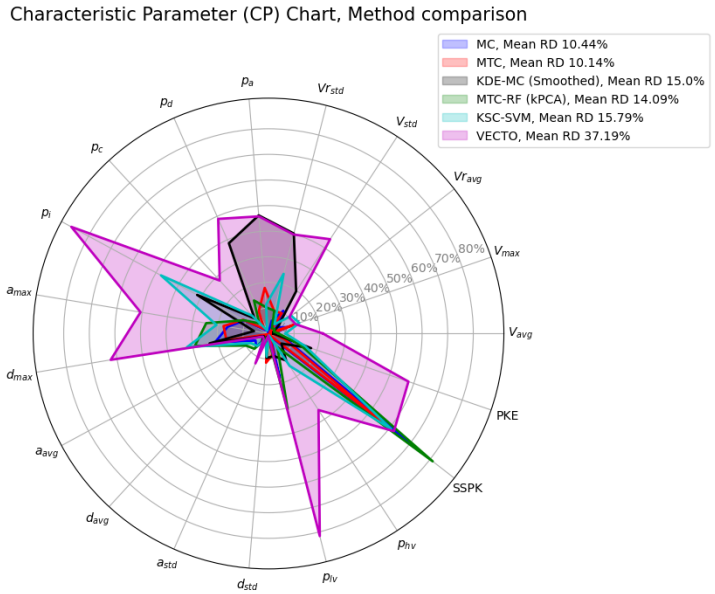


Figure 5.26: Comparison CP RDs achieved with different methods.

Despite the KS methods' inability to produce as accurate CP cycles as the SAS and MT methods, the KS methods were still able to generate candidate cycles with considerably improved mean CP RD compared to the VECTO cycle. The KSC-SVM method produced a candidate cycle with 15.79% mean CP RD, a 21.9 percentage point improvement compared to the 36.69% mean CP RD of the VECTO cycle.

5.5 Speed Acceleration Probability Distribution Summary

In Table 5.13, all the SAPD metrics are summarised for each method, showing the best SAPD representations sorted by lowest EMD cost. The MT methods stand out as clearly superior in SAPD representativeness compared to the SAS and KS methods. The top five lowest EMD costs and SAPD RDs are all MT method candidate cycles.

Method	**EMD** (10^{-4})	**SAPD RD**
MTC-NORF (PCA)	7.642	10.13%
MTC-RF (PCA)	7.779	9.935%
MTC-RF (kPCA)	8.549	10.99%
MTC-NORF (kPCA)	8.549	10.99%
MTC	9.420	12.21%
KDE-MC (Smoothed)	11.98	15.81%
MC	14.35	17.73%
KDE-MC	18.20	23.42%
KSC-SVM	32.16	41.45%
KSC	35.13	44.72%
VECTO	37.76	48.62%

Table 5.13: SAPD metrics compared between methods.

Following the MT methods are the SAS methods. Although they did not achieve the same level of SAPD representativeness as the MT methods, they still performed reasonably well, significantly better than the KS methods and the baseline VECTO cycle. The KS cycles, despite having significantly worse SAPD representation compared to the MT and SAS cycles, still managed to achieve slightly improved SAPD metrics compared to the VECTO cycle.

6

Discussion

This chapter discusses the performance of the implemented DC generation methods. One of the main goals of this thesis was to evaluate, compare, and asses the suitability of different advanced data analytics methods for DC generation. The suitability criteria being their accuracy in replicating real-world driving patterns, efficiency, and scalability for processing large datasets.

The section begins with an analysis and discussion of each construction model methodology: SAS methods, MT methods, and KS methods. This is followed by a discussion on the developed and proposed framework for DC generation and its effectiveness. Finally, ethical and sustainability considerations of the research are discussed.

6.1 Performance of Speed Acceleration State Methods

The SAS methods implemented demonstrated many strengths but also certain limitations in DC generation. One of the main advantages of SAS methods compared to other methods evaluated in this thesis, especially the traditional MC method, was the computational efficiency. The discrete TPM logic used to model speed change probabilities is logically straightforward, the TPM can be easily stored for later use and is computationally efficient to sample from. This makes it possible to sample numerous candidate cycles quickly, which through the law of averages increases the likelihood of achieving a highly representative DC. Moreover, expanding the TPM to include more data is also efficient and straightforward. This makes the SAS methods with TPM logic highly scalable for handling large and growing datasets. This is an aspect that should not be overlooked since the industry is always changing and having cycles that are easily modified adds great value.

The availability of operational data had a significant impact on the choice of method parameters, particularly the discretisation intervals of the states. With a large amount of operational data available, as in this thesis, increasing state granularity allowed for more subtle speed transitions, better reflecting the nuanced variations of real-world driving patterns and increasing cycle transientness. In this thesis, a speed interval discretisation of 0.2 km/h was used compared to the more commonly adopted integer interval as discussed in Chapter 2. However, excessive granularity

can lead to diminished performance when the operational data is sparse. Such a case may result in having insufficient data points for each state. This worsens driving pattern recognition and is more than likely to negatively impact performance. In such cases, having less granular states might be more beneficial. The resulting cycles might tend to be less transient but still overall better capture the driving patterns of the targeted operational data. This would in turn likely lead to suboptimal CP evaluation as certain CPs would need more transient behavior to be accurately represented. However, with a general driving pattern similar to the operational data the SAPD might still be highly representative. This highlights the importance of balancing method parameter choices with the operational data availability to optimise method performance.

Exploring SAS method optimisations through continuous-valued speed sampling can further enhance the transientness of SAS generated cycles. Although the KDE-MC method did not perform as well as anticipated in this thesis, its methodology of KDE sampling and KDE interpolation provides a more nuanced approach to speed sampling, reducing bias and increasing variance in the resulting cycles. This can be particularly useful in regions where operational data is sparse, as KDE interpolation can estimate driving patterns, potentially leading to better pattern recognition and more representative cycles.

The synthetic nature of SAS-generated cycles is another significant method advantage. This allows a certain level of flexibility in controlling cycle length by adjusting termination criteria. While the generation is stochastic and controlling exactly when a generated cycle should be idling is not feasible, adjusting the minimum cycle length criteria can significantly impact the average candidate cycle length. This modularity is particularly valuable in practical applications such as simulations, where specific needs or constraints on cycle length often exist. Cycles that are too long are less practical for these purposes.

Overall, SAS methods have proven to be highly effective methods. The candidate cycles generated maintain realistic driving patterns and exhibit strong similarities to the operational data. This is especially true in terms CPs, indicating that the methods can statistically represent many dimensions of operational data accurately and exhibit strong transientness. Despite its simplicity, the traditional MC method is currently determined to be the overall best approach. However, continuing to explore method optimisations with the KDE-MC method or other continuous-valued speed sampling approaches could lead to enhancements that improve upon the traditional MC method and make SAS methods even more suitable for DC generation.

6.2 Performance of Micro-Trip Methods

MT methods proved to generate representative DCs. The main strength was the SAPD cycle representativeness. All the implemented MT methods generated candidate cycles being highly accurate SAPD representations, indicating that they can generate cycles that closely mimic the overall driving behaviour of the operational data. In terms of CP evaluation, all the methods consistently performed solid as well.

The MTC method achieved the lowest overall mean CP RD across all implemented methods, not just among MT methods. The other methods, while not achieving quite as low mean CP RD, still consistently produced accurate candidates with no significant outliers.

Efficiency and scalability are also strengths of the MT methods. All variations, except those incorporating kPCA for dimension reduction, demonstrated high computational efficiency in the cycle generation. K-means clustering works well with large datasets, as does PCA for dimension reduction and RF for clustering refinement. Although methods incorporating kPCA achieved marginally improved accuracy in terms of mean CP RD, kPCA is significantly more computationally intensive compared to PCA. Given that PCA achieved comparable CP representativeness and slightly improved SAPD representativeness, PCA dimension reduction is determined to be the more practical and better option for dimension reduction.

The MTC-NORF and MTC-RF methods selected MTs with high PCC to be included in their candidate cycles. Hence there was less randomness in what MTs were selected. This ended up reducing result variance and improving method robustness. Out of the one hundred candidate cycles, most were at least decently representative, and fewer candidates were required to ensure that a highly representative one was generated. However, the original MTC method still achieved the lowest mean CP RD, suggesting that having more variation in the output candidates can also be beneficial. Exploring different combinations of MT concatenations can lead to better MT combinations.

RF refinement should theoretically improve the MT clustering, however, in this implementation, it ended up not adding any substantial value. The compactness and separation of the clustering pretty much remained the same, highlighting that it is important to evaluate the practical benefits of applying more advanced techniques versus the computational cost that they add. However, given that RF handles large datasets well and does not add much computational cost, it remains a valid method addition. In a different method application with worse initial clustering, RF refinement could potentially add more value.

An interesting aspect of MT methods which is different from the SAS and KS methods is that the generated cycles are not entirely synthetic. The candidates are just concatenations of shorter real-driven trips, which in theory could make a complete DC just be one long driven MT. That would not then be considered a synthetic cycle but instead just logged operational data.

This aspect of MT candidates being less synthetic has both advantages and disadvantages. The superior SAPD representativeness observed with MT methods indicates that sampling real-driven trips results in overall cycle behavior similar to the real operational data. Additionally, using real MTs means that additional features such as road slope, air temperature, engine speed, cruise control activation, and so on are inherently included in each sampled MT. This makes adding such additional features to the cycles logically more straightforward compared to SAS and KS methods which generate entirely synthetic cycles. This makes MT methods intriguing for further research to investigate how such additional parameters could be incorporated into the cycles and enhance their multidimensional representativeness.

However, the approach of sampling real MTs also comes with some limitations in adjusting the candidate cycle length, which is dependent on the length of the selected MTs. This can at times result in excessively long candidate cycles that may be impractical for certain applications. This issue was especially evident with the MTC-NORF and MTC-RF generated cycles, which all exceeded 10 000 seconds. While such cycles may be statistically accurate, they are not practical for simulation scenarios requiring shorter cycle lengths. Therefore, MT methods might be a less viable cycle generation option when targeting the long-haul driving profile. However, they may instead be more suitable for a profile like local distribution where MTs tend to be significantly shorter.

Overall, MT methods have proven to be highly effective for generating representative DCs, particularly in terms of SAPD. The fact that the cycles consist of real-driven trips adds validity to the results and ensures that the cycles will always exhibit realistic driving patterns. However, the cycles consisting of real-driven trips also come with some limitations regarding cycle length adjustments. Despite that limitation, MT methods are efficient and valuable for generating representative DCs suitable for HDV development and testing.

6.3 Performance of Kinematic Segment Methods

KS methods achieved the worst outcomes in both SAPD and CP evaluations. Additionally, the KS methods were also the most computationally intensive making them not at all suitable to apply to the large dataset used in this thesis.

The most significant limitation, aside from perhaps the computational intensity, was the frequent occurrence of unrealistic transitions between segments. The segment splicing often led to non-realistic speed changes. This issue was not only evident statistically but also visually when analysing the cycle results. Although the CP evaluation of the KS methods suggested that the cycles were not substantially worse representations of the operational data compared to the other methods, the SAPD and visual analysis clearly showed otherwise.

While further refinements could potentially, and likely, improve upon the performance of the KS methods, they are still less promising compared to other approaches due to the significant performance gap that exists compared to SAS and MT methods. The primary cause for the poor performance is most likely the segment sampling logic. With the current method, the selection of the next segment to add to the cycle is not computationally efficient enough, making the cycle generation process very slow. Additionally, the logic for determining a suitable next segment needs to be adjusted. For instance, a possible refinement could be implementing a segment transition probability logic similar to the TPM of the SAS methods. Incorporating both the speed and acceleration of segments and computing transition probabilities to following segments might help address the issue of unrealistic transitions. Moreover, a TPM logic could also make the sampling process more computationally efficient. However, this would require significant further research to determine the possibility of such an approach.

The SVM cluster refinement, although theoretically beneficial, did not provide significant performance improvement. It also turned out to be very computationally intensive with the dataset size used for the thesis, making it an impractical addition. The marginal performance benefits achieved do not justify the additional computational cost. Instead, implementing a cluster refinement technique better suited for large datasets, such as RF, is likely a better refinement technique to investigate further.

The overall performance of KS methods was poor, significantly worse than the SAS and MT methods. However, there were some positive aspects of KS methods. The CPs of the KS methods were not substantially worse than many of the other SAS and MT methods. The KS methods were able to represent high-speed cruising segments from the operational data. Additionally, similar to the SAS methods, KS methods generate completely synthetic cycles which allows for more control and flexibility in candidate cycle lengths. However, these aspects do not outweigh the limitations, and thus KS methods are concluded to be a less viable option for DC generation.

6.4 DC Generation Framework

The development of a systematic framework for generating, evaluating, and validating synthetic DCs was also a key goal of this thesis work. The effectiveness and adaptability of this framework are discussed here to provide a foundation for future research.

The data preparation steps undertaken were fundamental in ensuring the quality and representativeness of the driving data, involving careful cleaning, preprocessing, and trip segmentation. Effective trip segmentation was critical to ensure each segment represented typical driving conditions, which was crucial for generating synthetic DCs that accurately replicate real-world trips. Inaccurate or incomplete data could lead to driving cycles that fail to represent actual driving patterns, highlighting the importance of thorough preparation. The preprocessing steps implemented in this research ensured sufficient data quality and should be considered the minimum required to generate reliable and representative driving cycles. However, additional preprocessing steps could be added to further refine the data and improve its overall quality.

The statistical selection of trips was crucial for focusing on relevant data and filtering out outliers, a practice that is not very common in existing research. This step ensured that the input data used for generating DCs represented typical driving patterns rather than anomalies, thereby helping to make the synthetic DCs more representative and reliable. Although nineteen of the most commonly utilised CPs were used in this thesis to aid in this selection, more effort could be directed towards analysing the most important CPs for specific target profiles by employing advanced statistical techniques or machine learning algorithms and weighting them accordingly. Currently, all nineteen CPs were weighted equally. This approach could further refine the selection process and improve the overall quality of the synthetic DCs.

Evaluating the synthetic DCs using both CPs and SAPD provided a comprehensive

assessment of their representativeness. CP evaluation offers detailed insights into specific driving behaviors, ensuring that key characteristics of the driving data are captured. On the other hand, SAPD provides a broader view by analysing the probability of different speed and acceleration states, which helps in understanding the overall driving conditions. Using the EMD metric for evaluating SAPD proved particularly robust in measuring differences between distributions. Although not commonly used in existing research, EMD is recommended due to its ability to capture broader shifts in data distribution. This makes it more reliable than RD, which measures exact differences may excessively penalize these shifts. Evaluating SAPD with both EMD and RD was however beneficial as it ensured a more comprehensive assessment without relying on a single evaluation method.

Regarding the CP selection in general, further effort could be dedicated to reviewing the selected CPs to determine their relevance, potentially adding or excluding some to better target specific driving profiles. This could lead to more accurate and representative synthetic DCs. For instance, most methods struggled with the SSPK parameter, raising questions about its validity. The current approach for computing this CP involved counting all start and stop events, including very brief stops lasting only a few seconds or meters, which may not be meaningful. Revising the definition of SSPK, especially when computed for the operational data, could improve the relevance of this parameter.

The DC generation framework is highly adaptable which makes it applicable to a wide range of driving data and vehicle types. Although this study focused on long-haul operational data, the methodological steps are equally relevant for various other driving profiles, such as local distribution, and for different vehicle types, including combustion engine vehicles, electric vehicles and passenger cars. The framework can easily be adapted by adjusting parameters and thresholds to suit different driving conditions and vehicle characteristics. For instance, parameters for filtering short trips or MTs can be customised for long-haul as well as for any other driving profile. Similarly, threshold values used for data cleaning, such as those for reasonable speed changes, and other parameters for data preparation and CP evaluation, can also easily be adjusted to fit other automotive research and development projects.

6.5 Ethics and Sustainability

This section covers the ethics and sustainability considerations of this thesis. Given the thesis domain, field, and collaboration with the Volvo Group, there was a considerable amount of important aspects to consider to maintain research integrity. Although this section will only cover some of the more critical aspects, the existence of many more is also acknowledged.

The thesis involved working with real-world driving data, which raised concerns regarding data privacy and security. To that effect, measures were taken to ensure that any data published has been anonymised, confidentiality has been maintained and no personal information is traceable. Ensuring GDPR compliance.

In addition to individual privacy concerns, the protection of the intellectual property

rights of the Volvo Group was also critical to consider. Accordingly, measures were taken to prevent any disclosure of proprietary information that could be exploited by competitors or other entities.

Ensuring the quality of the data used was critical to prevent any biases. Bias could skew the outputs of data analytics methods and invalidate the results. These obligations are especially important to align with when validating these cycles against VECTO. Using unreliable or unrepresentative data as input would lead to generating DCs that could be mistakenly evaluated as representative of real-world driving. This would compromise the scientific integrity of the research, and possibly result in non-compliance with environmental standards and other ethical aspects. Therefore, a thorough analysis of the data quality and data preprocessing was conducted to ensure the reliability and true representativeness of the operational data.

7

Conclusion

This chapter presents suggestions for future work in the field and concludes the thesis with a summary of key aspects and goals achieved.

7.1 Future Work

This research focused on the long-haul driving profile. For future work it would be useful to compare the performance of these methods across different driving profiles to determine their effectiveness for each specific profile. Long-haul driving profiles generally have longer MTs which can make effective sampling from all clusters challenging without producing overly long DCs. Long MTs can also reduce the representativeness and control over DC duration. Other driving profiles with shorter MTs, such as local distribution, could enable more effective sampling and potentially better performance of MT-based methods. Comparing the methods across different driving profiles would reveal if some methods are better suited for other driving profiles, even if they are less effective for long-haul driving. Making this kind of comparison could give more insight into the strengths and limitations of each method.

Further optimisation of the methods compared in this research is another potential area for future work. For instance the performance of the KDE-MC method could improve with some additional fine-tuning of the bandwidth parameter. For the KS methods, exploring other ways to connect segments could mitigate the issue with unrealistic transitions between segments. Making these optimisation efforts could boost the performance of the methods and make them more reliable for generating representative DCs.

Additionally, incorporating more parameters for generating DCs could make them more representative. This research only considered parameters derived from vehicle speed. Adding parameters such as road gradient which greatly affect fuel consumption and emissions, could make the synthetic DCs more accurate. With these added parameters, the DCs might better reflect real-world conditions which will result in more realistic performance estimations.

Finally, future research could improve the DC generation framework by evaluating the relevance of the CPs used for both the evaluation and the statistical selection of trips. This includes both assessing the effectiveness of currently used CPs and also identifying if additional parameters should be included. Furthermore, assigning

weights to these CPs based on the target driving profile could improve the overall framework. Currently, all CPs are treated equally. By adjusting their weights to reflect their importance for different profiles, the framework could better capture the specific driving characteristics of each profile. For instance, CPs critical for long-haul driving would have higher weights during the evaluation and trip selection process for a long-haul driving profile, while those same CPs might receive lower weights for a local distribution driving profile. Using weighted CPs would make the framework more adaptable and effective.

7.2 Conclusion

In this thesis, advanced data analytics methods were selected, implemented, evaluated, and compared in their performance in generating synthetic DCs representative of real-world driving. The methods covered three main construction model methodologies: SAS, MT, and KS, adding depth to the method comparison by analysing how these construction choices impact the capability to generate synthetic time-series cycles that are accurate representations of operational data while remaining practical for industry applications.

All implemented methods successfully generated DCs that statistically represented the operational data better than the VECTO baseline cycle. This demonstrates that DCs generated by advanced data analytical methods and fine-tuned for specific vehicle populations can enhance the statistical accuracy compared to standardised regulatory cycles. Particularly promising results were achieved with SAS and MT methods, both of which produced statistically accurate cycles while maintaining high computational efficiency, making them suitable for industry applications. Moving forward, further research should focus on optimising these methods to enhance performance and incorporate additional parameters beyond those derived from vehicle speed to increase the multidimensional representativeness of the cycles.

The knowledge and experience acquired during this process were used to formulate a framework to aid future work in DC development. The framework aims to provide a standardised approach to data preparation, cycle generation, evaluation and validation. Although this thesis focused on HDV long-haul operational data, the methodological steps are applicable to any type of operation or vehicle as long as the operational data is in time-series format and stored with reasonable resolution.

Bibliography

[1] M. K. Kondaru, K. T. Prasad, S. V. Thimmalapura, and N. K. Pandey, "Generating a real world drive cycle–a statistical approach," *SAE Technical Paper*, 2018, ISSN: 0148-7191. DOI: 10.4271/2018-01-0325. [Online]. Available: https://doi.org/10.4271/2018-01-0325.

[2] R. Ma, E. Breaz, and F. Gao, "Chapter 10 - power demand for fuel cell system in hybrid vehicles," in *Fuel Cells for Transportation*, P. K. Das, K. Jiao, Y. Wang, B. Frano, and X. Li, Eds., Woodhead Publishing, 2023, pp. 279–303, ISBN: 978-0-323-99485-9. DOI: https://doi.org/10.1016/B978-0-323-99485-9.00016-2. [Online]. Available: https://www.sciencedirect.com/science/article/pii/B9780323994859000162.

[3] European Commission, *Reducing co2 emissions from heavy-duty vehicles*, https://climate.ec.europa.eu/eu-action/transport/road-transport-reducing-co2-emissions-vehicles/reducing-co2-emissions-heavy-duty-vehicles_en, Accessed: 2024-03-18, 2023.

[4] E. kommissionen, G. forskningscentrumet, G. Fontaras, *et al.*, *From NEDC to WLTP : effect on the type-approval CO2 emissions of light-duty vehicles*. Publications Office, 2017. DOI: doi/10.2760/93419.

[5] M. Tutuianu, P. Bonnel, B. Ciuffo, *et al.*, "Development of the world-wide harmonized light duty test cycle (wltc) and a possible pathway for its introduction in the european legislation," *Transportation Research Part D: Transport and Environment*, vol. 40, pp. 61–75, 2015, ISSN: 1361-9209. DOI: https://doi.org/10.1016/j.trd.2015.07.011. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S1361920915001030.

[6] L. Romano, *The Operating Cycle Representation of Road Transport Missions* (Doktorsavhandlingar vid Chalmers tekniska högskola. Ny serie). Chalmers University of Technology, 2023, ISBN: 9789179058883. [Online]. Available: https://books.google.se/books?id=tJBF0AEACAAJ.

[7] D. Farmer, *Advanced analytics*, Accessed: 2024-03-18, 2024. [Online]. Available: https://www.techtarget.com/searchbusinessanalytics/definition/advanced-analytics.

[8] Coursera, *What is advanced analytics?* https://www.coursera.org/articles/advanced-analytics, Accessed: 2024-03-18, 2024.

[9] Amazon Web Services, *What is advanced analytics?* https://aws.amazon.com/what-is/advanced-analytics/, Accessed: 2024-03-18, 2024.

[10] X. Zhao, X. Zhao, Q. Yu, Y. Ye, and M. Yu, "Development of a representative urban driving cycle construction methodology for electric vehicles: A case

study in xi'an," *Transportation Research Part D: Transport and Environment*, vol. 81, p. 102 279, 2020, ISSN: 1361-9209. DOI: https://doi.org/10.1016/j.trd.2020.102279. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S1361920918303274.

[11] C. Zhang, A. Kotz, K. Kelly, and L. Rippelmeyer, "Development of heavy-duty vehicle representative driving cycles via decision tree regression," *Transportation Research Part D: Transport and Environment*, vol. 95, p. 102 843, 2021, ISSN: 1361-9209. DOI: https://doi.org/10.1016/j.trd.2021.102843. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S1361920921001462.

[12] X. Jia, H. Wang, L. Xu, *et al.*, "Constructing representative driving cycle for heavy duty vehicle based on markov chain method considering road slope," *Energy and AI*, vol. 6, p. 100 115, 2021, ISSN: 2666-5468. DOI: https://doi.org/10.1016/j.egyai.2021.100115. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S2666546821000641.

[13] J. Peng, J. Jiang, F. Ding, and H. Tan, "Development of driving cycle construction for hybrid electric bus: A case study in zhengzhou, china," *Sustainability*, vol. 12, no. 17, 2020, ISSN: 2071-1050. DOI: 10.3390/su12177188. [Online]. Available: https://www.mdpi.com/2071-1050/12/17/7188.

[14] R. M. Desineedi, S. Mahesh, and G. Ramadurai, "Developing driving cycles using k-means clustering and determining their optimal duration," *Transportation Research Procedia*, vol. 48, pp. 2083–2095, 2020, ISSN: 2352-1465. DOI: 10.1016/j.trpro.2020.08.268.

[15] Y. Yang, Q. Zhang, Z. Wang, Z. Chen, and X. Cai, "Markov chain-based approach of the driving cycle development for electric vehicle application," *Energy Procedia*, vol. 152, pp. 502–507, 2018, Cleaner Energy for Cleaner Cities, ISSN: 1876-6102. DOI: https://doi.org/10.1016/j.egypro.2018.09.201. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S187661021830746X.

[16] S. K. Pathak, Y. Singh, V. sood, and S. A. Channiwala, "On-road vehicle driving and energy requirements and impact on unregulated exhaust emissions under urban driving conditions," *SAE International Journal of Engines*, vol. 10, no. 4, pp. 1866–1879, 2017, ISSN: 1946-3936. DOI: 10.4271/2017-01-1013. [Online]. Available: https://doi.org/10.4271/2017-01-1013.

[17] S. Saxena, B. Kudachi, S. Pasupathi, and G. Bergsieker, "Commercial vehicle – drive cycle development and validation using gt-realdrive & 1d gt-suite electric vehicle models," *SAE Technical Paper*, 2023, ISSN: 0148-7191. DOI: 10.4271/2023-01-0472. [Online]. Available: https://doi.org/10.4271/2023-01-0472.

[18] S. Kamguia Simeu and N. Kim, "Standard driving cycles comparison (iea) & impacts on the ownership cost," *SAE Technical Paper*, 2018, ISSN: 0148-7191. DOI: 10.4271/2018-01-0423. [Online]. Available: https://doi.org/10.4271/2018-01-0423.

[19] J. I. Huertas, L. F. Quirama, M. Giraldo, and J. Díaz, "Comparison of three methods for constructing real driving cycles," *Energies*, vol. 12, no. 4, 2019,

ISSN: 1996-1073. DOI: 10.3390/en12040665. [Online]. Available: https://www.mdpi.com/1996-1073/12/4/665.

[20] L. Wang, J. Ma, X. Zhao, and X. Li, "Development of a typical urban driving cycle for battery electric vehicles based on kernel principal component analysis and random forest," *IEEE Access*, vol. 9, pp. 15 053–15 065, Jan. 2021. DOI: 10.1109/ACCESS.2021.3052820.

[21] Z. Dai, D. Niemeier, and D. Eisinger, "Driving cycles: A new cycle-building method that better represents real-world emissions," Jan. 2008.

[22] D. Yang, T. Liu, X. Zhang, X. Zeng, and D. Song, "Construction of high-precision driving cycle based on metropolis-hastings sampling and genetic algorithm," *Transportation Research Part D: Transport and Environment*, vol. 118, p. 103 715, 2023, ISSN: 1361-9209. DOI: https://doi.org/10.1016/j.trd.2023.103715. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S1361920923001128.

[23] X. Zhao, Q. Yu, J. Ma, *et al.*, "Development of a representative ev urban driving cycle based on a k-means and svm hybrid clustering algorithm," *Journal of Advanced Transportation*, vol. 2018, p. 1 890 753, 2018. DOI: 10.1155/2018/1890753. [Online]. Available: https://doi.org/10.1155/2018/1890753.

[24] J. Topić, B. Škugor, and J. Deur, "Synthesis and feature selection-supported validation of multidimensional driving cycles," *Sustainability*, vol. 13, no. 9, 2021, ISSN: 2071-1050. DOI: 10.3390/su13094704. [Online]. Available: https://www.mdpi.com/2071-1050/13/9/4704.

[25] L. Berzi, M. Delogu, and M. Pierini, "Development of driving cycles for electric vehicles in the context of the city of florence," *Transportation Research Part D: Transport and Environment*, vol. 47, pp. 299–322, 2016, ISSN: 1361-9209. DOI: https://doi.org/10.1016/j.trd.2016.05.010. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S1361920916303017.

[26] H.-Y. T. Wing-Tat Hung and C.-S. Cheung, "A modal approach to vehicular emissions and fuel consumption model development," *Journal of the Air & Waste Management Association*, vol. 55, no. 10, pp. 1431–1440, 2005, PMID: 28086044. DOI: 10.1080/10473289.2005.10464747. eprint: https://doi.org/10.1080/10473289.2005.10464747. [Online]. Available: https://doi.org/10.1080/10473289.2005.10464747.

[27] J. C. Ferreira, J. de Almeida, and A. R. da Silva, "The impact of driving styles on fuel consumption: A data-warehouse-and-data-mining-based discovery process," *IEEE Transactions on Intelligent Transportation Systems*, vol. 16, no. 5, pp. 2653–2662, 2015. DOI: 10.1109/TITS.2015.2414663.

[28] L. F. Quirama, M. Giraldo, J. I. Huertas, J. E. Tibaquirá, and D. Cordero-Moreno, "Main characteristic parameters to describe driving patterns and construct driving cycles," *Transportation Research Part D: Transport and Environment*, vol. 97, p. 102 959, 2021, ISSN: 1361-9209. DOI: https://doi.org/10.1016/j.trd.2021.102959. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S1361920921002571.

[29] J. I. Huertas, M. Giraldo, L. F. Quirama, and J. Díaz, "Driving cycles based on fuel consumption," *Energies*, vol. 11, no. 11, 2018, ISSN: 1996-1073. DOI:

10.3390/en11113064. [Online]. Available: https://www.mdpi.com/1996-1073/11/11/3064.

[30] A. Gaikwad, *The Fundamentals of Machine Learning*. Jun. 2023, ISBN: ISBN:978-620-6-16611-5.

[31] A. Ławrynowicz and V. Tresp, "Introducing machine learning," in *Perspectives on Ontology Learning*. 2014, vol. 18, pp. 35–50.

[32] M. Awad and R. Khanna, "Support vector machines for classification," in Apr. 2015, pp. 39–66, ISBN: 978-1-4302-5989-3. DOI: 10.1007/978-1-4302-5990-9_3.

[33] T. M. Mitchell, *Machine learning*. McGraw-hill New York, 1997, vol. 1.

[34] L. Rokach and O. Maimon, *Data Mining With Decision Trees: Theory and Applications*, 2nd. USA: World Scientific Publishing Co., Inc., 2014, ISBN: 9789814590075.

[35] C. C. Aggarwal and C. K. Reddy, *Data Clustering: Algorithms and Applications*, 1st. Chapman & Hall/CRC, 2013, ISBN: 1466558210.

[36] D. SAPUTRA, D. Saputra, and L. Oswari, "Effect of distance metrics in determining k-value in k-means clustering using elbow and silhouette method," Jan. 2020. DOI: 10.2991/aisr.k.200424.051.

[37] E. Umargono, J. Suseno, and S. Gunawan, "K-means clustering optimization using the elbow method and early centroid determination based on mean and median formula," Jan. 2020. DOI: 10.2991/assehr.k.201010.019.

[38] A. El-Mandouh, L. A., A. Hamdi, and M. H., "Optimized k-means clustering model based on gap statistic," *International Journal of Advanced Computer Science and Applications*, vol. 10, Jan. 2019. DOI: 10.14569/IJACSA.2019.0100124.

[39] I. Jolliffe, *Principal component analysis*. New York: Springer Verlag, 2002.

[40] S. Y. Kung, "Pca and kernel pca," in *Kernel Methods and Machine Learning*. Cambridge University Press, 2014, pp. 79–117.

[41] G. Bonaccorso, "Machine learning algorithms second edition," 2018.

[42] S. Weglarczyk, "Kernel density estimation and its application," *ITM Web of Conferences*, vol. 23, p. 00 037, Jan. 2018. DOI: 10.1051/itmconf/20182300037.

[43] B. W. Silverman, "Density estimation for statistics and data analysis," 1986.

[44] C. M. Bishop, *Pattern Recognition and Machine Learning (Information Science and Statistics)*, 1st ed. Springer, 2007, ISBN: 0387310738.

[45] "Markov chains: First steps," in *Introduction to Stochastic Processes With R*. John Wiley & Sons, Ltd, 2016, ch. 2, pp. 40–75, ISBN: 9781118740712. DOI: https://doi.org/10.1002/9781118740712.ch2. eprint: https://onlinelibrary.wiley.com/doi/pdf/10.1002/9781118740712.ch2. [Online]. Available: https://onlinelibrary.wiley.com/doi/abs/10.1002/9781118740712.ch2.

[46] P. Nyberg, E. Frisk, and L. Nielsen, "Driving cycle equivalence and transformation," *IEEE Transactions on Vehicular Technology*, vol. 66, no. 3, pp. 1963–1974, 2017. DOI: 10.1109/TVT.2016.2582079.

[47] S. Shirdhonkar and D. W. Jacobs, "Approximate earth mover's distance in linear time," in *2008 IEEE Conference on Computer Vision and Pattern Recognition*, 2008, pp. 1–8. DOI: 10.1109/CVPR.2008.4587662.

[48] Y. Rubner, C. Tomasi, and L. Guibas, "The earth mover's distance as a metric for image retrieval," *International Journal of Computer Vision*, vol. 40, pp. 99–121, Nov. 2000. DOI: 10.1023/A:1026543900054.

[49] P. Schober, C. Boer, and L. Schwarte, "Correlation coefficients: Appropriate use and interpretation," *Anesthesia & Analgesia*, vol. 126, p. 1, Feb. 2018. DOI: 10.1213/ANE.0000000000002864.

[50] S. Shi, N. Lin, Y. Zhang, *et al.*, "Research on markov property analysis of driving cycle," in *2013 IEEE Vehicle Power and Propulsion Conference (VPPC)*, 2013, pp. 1–5. DOI: 10.1109/VPPC.2013.6671737.

[51] P. Nyberg, E. Frisk, and L. Nielsen, "Generation of equivalent driving cycles using markov chains and mean tractive force components," *IFAC Proceedings Volumes (IFAC-PapersOnline)*, vol. 19, pp. 8787–8792, Jan. 2014.

A

Appendix 1