VOCATION CLUSTERING FOR HEAVY-DUTY VEHICLES

by

Daniel Kobold, Junior

A Thesis

Submitted to the Faculty of Purdue University

In Partial Fulfillment of the Requirements for the degree of

Master of Science in Electrical and Computer Engineering

Department of Electrical and Computer Engineering

Indianapolis, Indiana

December 2020

THE PURDUE UNIVERSITY GRADUATE SCHOOL
STATEMENT OF COMMITTEE APPROVAL

Dr. Zina Ben-Miled, Chair

Department of Electrical and Computer Engineering

Dr. Brian S. King

Department of Electrical and Computer Engineering

Dr. Euzeli C. Dos Santos

Department of Electrical and Computer Engineering

Approved by:

Dr. Brian S. King

To my family and friends,
colleagues and mentors,
from every walk of life

ACKNOWLEDGMENTS

I would like to extend my thanks to Dr. Zina Ben Miled and Dr. Euzeli dos Santos for their assistance and advice throughout this project. Also, my sincere appreciation goes to Mr. Andy Byerly and Allison Transmission, Inc. for their support and technical guidance. The insights offered by my teammates, Mr. Nathaniel Cantwell and Mr. Rishikesh Bagwe were invaluable. In addition, I would like to thank Dr. Brian King for serving on my committee, and Mrs. Sherrie Tucker for guiding me through the administrative processes during my master's degree.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

ABBREVIATIONS

Algorithms

KM	K-Means
EM	Expectation-Maximization

Vocations

BT	Bucket Truck
CT	Class 8 Tractor
DT	Delivery Truck
DV	Delivery Van
TB	Transit Bus

ABSTRACT

The identification of the vocation of an unknown heavy-duty vehicle is valuable to parts manufacturers who may not have otherwise access to this information on a consistent basis. This study proposes a methodology for vocation identification that is based on clustering techniques. Two clustering algorithms are considered: K-Means and Expectation Maximization. These algorithms are used to first construct the operating profile of each vocation from a set of vehicles with known vocations. The vocation of an unknown vehicle is then determined using different assignment methods.

These methods fall under two main categories: one-versus-all and one-versus-one. The one-versus-all approach compares an unknown vehicle to all potential vocations. The one-versus-one approach compares the unknown vehicle to two vocations at a time in a tournament fashion. Two types of tournaments are investigated: round-robin and bracket. The accuracy and efficiency of each of the methods is evaluated using the NREL FleetDNA dataset.

The study revealed that some of the vocations may have unique operating profiles and are therefore easily distinguishable from others. Other vocations, however, can have confounding profiles. This indicates that different vocations may benefit from profiles with varying number of clusters. Determining the optimal number of clusters for each vocation can not only improve the assignment accuracy, but also enhance the computational efficiency of the application. The optimal number of clusters for each vocation is determined using both static and dynamic techniques. Static approaches refer to methods that are completed prior to training and may require multiple iterations. Dynamic techniques involve clusters being split or removed during training. The results show that the accuracy of dynamic techniques is comparable to that of static approaches while benefiting from a reduced computational time.

1. INTRODUCTION

The ability to identify the vocation of a heavy-duty vehicle from drive cycle data collected during the vehicle's daily operation is valuable to many parts' manufacturers in order to track the end-use of the vehicle and subsequently improve parts' design and configuration. Electronic components and sensors in vehicles are becoming increasingly pervasive. This evolution led to the emergence of new sources of data that can help inform improved designs. In fact, both OEMs and parts' manufacturers now have access to a large stream of operational data that can be acquired during maintenance or configuration updates, with the vehicle owner's consent. However, as opposed to OEMs, parts' manufacturers do not typically have knowledge of the actual use, application or vocation of the vehicle. Moreover, the parts can be deployed in a large number of varying vocations. Finally, data available to the parts' manufacturer for each individual vehicle may be limited. Vocation information is needed to leverage the collected data for vocation-targeted design or configuration enhancements.

The proposed vocation identification methodology follows a two-step approach. First, the profile of each vocation of interest is established using a set of vehicles with known vocations. Second, the daily drive cycle measurements collected from the unknown vehicle are compared to all vocation profiles and the most likely vocation is selected.

In machine learning, this type of application is typically solved by using either classification or clustering techniques. The major difference between the two being that classification aims at assigning a vehicle to a vocation from a pre-defined set of vocations whereas clustering aims at grouping similar vehicles into one vocation. Because of this underlying difference, classification techniques rely primarily on supervised learning whereas clustering techniques use unsupervised learning[1][2]. This thesis investigates the use of clustering techniques for the vocation identification of heavy-duty vehicles.

Two clustering algorithms are used to demonstrate the proposed vocation identifier: K-means (KM)[3] and Expectation maximization (EM)[4]. It is possible to use other clustering algorithms, such as Particle swarm optimization (PSO)[5], Density-based spatial clustering

(DBSCAN)[6], or Hierarchical DBSCAN (HDBSCAN)[7] in conjunction with the proposed methodology. KM and EM have been selected because they are widely used in various applications [8][9].

Most classification algorithms are best at handling two classes[10]: a positive and a negative class. These binary classifiers have been extended to multi-class models using the one-versus-all[11] and the one-versus-one methodology[12]. The one-versus-all consists of an ensemble of classifiers where each classifier is trained to correctly predict one positive class while considering all the remaining classes as negative. This method has a linear complexity with respect to the number of vocations. The one-versus-one is also an ensemble of classifiers. However, in this case a classifier is developed for each pair of classes leading to a quadratic complexity with respect to the number of vocations.

In this thesis, a clustering approach as opposed to a classification approach is adopted. The proposed methodology follows a multi-class that was inspired by the one-versus-one classification approach which is able to accommodate a large number of vocations. The daily measurements of the unknown vehicle are compared to two vocations at a time in a tournament bracket. In each round, a vocation is eliminated, making the approach linearly scalable with respect to the number of vocations. This approach is compared to a one-versus-all assignment (where the unknown vehicle is compared to all the vocations at the same time) as well as to the round-robin assignment.

An important hyper-parameter of any of the clustering methodologies mentioned above is the number of clusters allowed for each vocation. This number can be pre-specified prior to training. However, selecting the appropriate number of clusters for each vocation requires prior knowledge of the vocations' operating profiles in addition to significant fine tuning (using an iterative procedure). Several methods that can be used to determine the optimal number of clusters per vocation have been proposed in the literature. These include, for example, the Elbow [13] and Silhouette [14] methods. Most of these previous approaches are iterative and require several iterations of development and validation until the optimal number of clusters for a vocation is established. In this thesis, these previous methods are evaluated and compared to a newly proposed dynamic approach. This approach attempts to determine the optimal number of clusters for each vocation during training. After a set of training epochs, the clusters established for the vocation are evaluated. This evaluation determines whether each cluster should be retained as is, removed, or split into two different clusters. The abovementioned clustering methodologies are demonstrated on 5 vocations from the NREL Fleet DNA[15][16] dataset.

The remainder of this thesis is organized as follows: Chapter 2 includes a review of previous work related to clustering and vehicular applications. Chapter 3 investigates methodologies for the assignment of records and vehicles to vocations. Chapter 4 describes various methods that can be used to determine the optimal number of clusters for a vocation. Chapter 5 concludes the thesis with a summary of the main findings and suggested direction for future work.

2. RELATED WORK

Vocation identification is a classification problem. The objective is to assign a record collected from an unknown vehicle to one of the potential vocations. In this study, a clustering methodology is used to first establish the operating profile of each vocation. The record collected from the unknown vehicle is then compared to these established profiles. This chapter introduces a brief review of previous work related to clustering and classification, and the use of these techniques in support of vehicular applications.

2.1 Classification and Clustering

Since vocations for the training dataset are known a-priori in our study, using a classifier model with supervised learning would be expected. Some of the widely recognized classification algorithms include support vector machine (SVM)[11], random forest (RF)[17], and neural networks. Most of these algorithms are inherently two-class (binary) classifiers. However, they have been extended to accommodate multi-class applications. For instance, SVM maps input records to a higher-dimensional input space using kernel transformations where they can be linearly separated by a hyperplane into a positive and a negative class. It was extended to multi-classes using one-versus-one and one-versus-all ensemble learners[11]. Similarly, neural networks can use multiple nodes in the output layer where each node corresponds to a class[18]. This architecture is equivalent to using an ensemble of one-versus-all neural networks. RF can also support multiple classes if multiway trees are used instead of the traditional binary decision trees[19]. These multi-class RF classifiers would also need a modified consensus rule when trees in the RF indicate different class predictions.

Compared to a classifier, the purpose of a clustering algorithm is to: (a) identify clusters with similar records, (b) select a representative member for each cluster and (c) adequately assign a record to a cluster. These three aspects vary from one clustering algorithm to the next. As opposed to a classifier, the first step is performed using unsupervised learning. For example, KM defines the similarity between two records according to a distance measure (e.g., Euclidean distance or cosine distance). The smaller the distances the more similar are the records. Other similarity criteria that are optimized to specific applications are proposed in [20][21][22].

Once a cluster is identified, a representative member, called the centroid is selected and refined iteratively as more members are added to or removed from the cluster. Under the KM and EM algorithms, the centroid is typically calculated by averaging across all the members of the cluster. A variant of this approach requires that the representative be an actual member of the cluster and this representative is called the medoid to distinguish it from the centroid. Other clustering algorithms, such as PSO[5], derive their efficiency from the selection of appropriate centroids. Centroids are mapped to particles in PSO[23]. Each particle moves in the feature space according to its velocity which is updated based on the best position that the particle has found so far and the current global best position across all particles. These positions are referred to as local conscience and global conscience, respectively. The particles are updated iteratively until the best centroids are found.

The assignment of a record to a cluster also varies from one clustering algorithm to the next. For KM, each record is assigned to exactly one cluster based on the distance between the record and the centroid of the cluster. This assignment is referred to as a ”hard” assignment. EM uses a ”soft” assignment[24]. That is, each record has a probability of belonging to each cluster.

2.2 Number of Clusters

Other important aspects of clustering algorithms include the relationship among the clusters and the appropriate number of clusters. Most clustering algorithms assume that all clusters are at the same level. This type of clustering is referred to as partitioning[6]. This is also the type of clustering being used in this study. In contrast, hierarchical clustering[7] allows some clusters to be a subset of others.

Silhouette[14] and Elbow[25] are methods for finding the optimal number of clusters. Silhouette[14] takes into consideration the tightness of a cluster and its separation from other clusters. Elbow[25] uses a recursive doubling approach. It starts with two clusters and splits these clusters until an accuracy threshold is achieved. The Elbow Method uses a measure such as inertia or sum square error (SSE) ratio to determine the point where additional clusters do not improve the accuracy of the clustering model. The elbow point is defined by a significant decrease in the inertia or SSE ratio, followed by a lack of significant change [13]. The drawback of this approach is the ability to accurately determine the elbow point, which requires manual evaluation of the inertia graph for clustering models with varying number of clusters, or an accurate mathematical formulation of what qualifies as an elbow point.

In this thesis, two methods for selecting an adequate number of clusters are investigated. The first method compares the average of the standard deviations of each feature to a threshold in order to decide whether the cluster should be split or removed. The second method is inspired from simulated annealing[26]. Simulated annealing allows some of the split or removal decisions to be arbitrary in the beginning of the training. However, this flexibility decreases as the training progresses.

2.3 Vehicular Applications

Clustering has been used in several vehicular applications. For example, in [21] location-based clustering was shown to enhance routing for vehicle-to-vehicle communication. Clustering was also used in [20] for sharing of traffic congestion information. Each cluster of vehicles was used to represent a given traffic flow thereby allowing the vehicle at the head of the flow to inform the vehicle at the tail of the flow of any traffic congestion. In [22], clustering was used to detect anomalous cab trajectories. Five clusters are established, namely, normal trajectory, global short cut, local short cut, global detour and local detour. Each of the above applications innovate by proposing a customized similarity measure for the target application.

The fleet DNA dataset used in this study was introduced and extensively analyzed in [16]. Indeed, dimension reduction was performed on the dataset using principal component analysis (PCA) and cross-correlation to identify the eight most expressive features in the dataset. These were found to be aerodynamic speed, characteristic acceleration, percent of total cycle distance accumulated at speeds below 55 mph, percent of total cycle time duration accumulated at vehicle speeds of 0 mph, number of vehicle stops per mile, mean (nonzero) driving speed, maximum driving speed and standard deviation of (nonzero) driving speed.

Using these eight features, the study found that the first 6 components of PCA were able to describe 99% of the variance in the data. K-medoid was also used to cluster all the drive cycles in the fleet DNA dataset into three clusters. The optimal number of clusters was established using Silhouette[14].

The above study helped guide the methodology proposed in this thesis. However, this thesis addresses a different problem and uses different algorithms. The NREL study[16] aims at identifying a limited number of representative drive cycles across all US commercial fleets. The aim of this thesis is to identify the specific vocation of an unknown vehicle. The methodology is also different since it demonstrates the use of a clustering algorithm for vocation identification. In fact, while targeting a different application, the methodology proposed in this thesis shares this aspect with the approach for the detection of anomalous cab trajectories proposed in [22]. The algorithm proposed in this thesis enhances this methodology by showing that a one-versus-one bracket assignment can be efficiently applied to a large number of vocations.

3. VOCATION IDENTIFICATION

This chapter describes the methodology used to develop the proposed vocation identification model. The profile of the vocation in the model is established using K-means (KM)[3] or Expectation-maximization (EM)[4]. Two primary methods are then used to assign a vehicle to a given vocation: one-versus-all assignment and one-versus-one assignment. One-versus-one assignment methods include the round-robin approach and the tournament bracket approach. Results are presented for each method.

3.1 Methodology

The proposed methodology is able to create a model that identifies the vocation of an unknown vehicle. In the next subsections, we describe the dataset, the training phase of the model which establishes the operating profile of each vocation, and three vocation assignment algorithms.

Table 3.1. Feature List.

	Label	Feature	Unit
1	Max spd	Max Speed	mph
2	Total avg spd	Total Average Speed	mph
3	*Total spd std	Total Speed Standard Deviation	mph
4	Drive avg spd	Driving Average Speed	mph
5	Drive spd std	Driving Speed Standard Deviation	mph
6	Zero seconds	Zero Seconds	$1000s$
7	Distance total	Distance Total	$miles$
8	Total stops	Total Stops	$count$
9	*Avg kin pwr density demand	Average Kinetic Power Density Demand	W/kg
10	*Cuml instant KE density	Cumulative Instantaneous Kinetic Energy Density	MJ/kg
11	*Char accel	Characteristic Acceleration	m/s^2
12	*Aero spd	Aerodynamic Speed	m/s
13	Max accel	Max Acceleration	ft/s^2
14	Avg accel	Average Acceleration	ft/s^2
15	*Max decel	Max Deceleration	ft/s^2

3.1.1 Dataset

Each vehicle in the Fleet DNA[15] dataset is represented by a set of records where every record is an aggregation of the drive cycle measurements over a single day. The features of the records used in this study are shown in Table 3.1. Their definitions are available in [16] and references therein. For convenience, some of these definitions are reproduced below:

- Max Speed: Maximum speed observed during the trip.
- Total Average Speed: Average speed over the trip (including zero speed points).
- Driving Average Speed: Average speed over the trip not including the zero speeds.
- Zero Seconds: Number of 1000 seconds at zero speed.
- Distance Total: Total distance traveled in miles.
- Total Stops: Number of vehicle stops.
- Average Kinetic Power Density Demand: Mean of the kinetic power density demand (with respect to mass).

The list of the 15 features shown in Table 3.1 was selected among the 350 available parameters in the original dataset using dimension reduction. Some of the features in the original data identify the vehicle, the deployment or the vocation. These were used to label the data. A large number of parameters were removed because they had a linear or an inverse relationship with another parameter (e.g., Characteristic Acceleration and Characteristic Deceleration, Average Acceleration and Average Deceleration). Parameters related to potential energy (e.g., Cumulative Instantaneous Potential Energy Density and Average Potential Power Density Demand) were also removed because they are more dependent on the road elevation than on the vocation of the vehicle. Moreover, daily records with Zero Seconds $>$ 18,000s were removed from all the vehicles because this is an indication that the vehicle was not in operation for more than 5 hours in the given day.

The original FleetDNA dataset includes eight vocations: Bucket Trucks, Class 8 Tractors, Delivery Vans, Delivery Trucks, Transit Buses, Refuse Trucks, School Buses, and Service Vans. The latter three vocations were eliminated because they did not include sufficient data.

Table 3.2. Number of vehicles used for testing for each vocation.

Vocation	Label	Total number of vehicles	Number of test vehicles
Bucket Truck	BT	12	2
Class 8 Tractor	CT	43	33
Delivery Truck	DT	29	19
Delivery Van	DV	26	16
Transit Bus	TB	21	11

For the remaining vocations, the vocation identification model followed a training/testing split at the vehicle level. This prevents information leakage that may result from allowing records from the same vehicle to participate in both the training and the testing of the vocation classification. After assigning a vehicle from the original dataset to either training or testing, 13 records were randomly sampled without replacement from each vehicle. Each random selection was considered as a separate vehicle in either the training or testing pool of vehicles. Moreover, to keep the training records balanced across vocations, 10 vehicles were selected per vocation for training. The remaining vehicles were used for testing. This split approach led to variations in the number of vehicles available for testing across the vocations (Table 3.2). In total, 50 vehicles are used for training and 81 vehicles are used for testing across the 5 vocations.

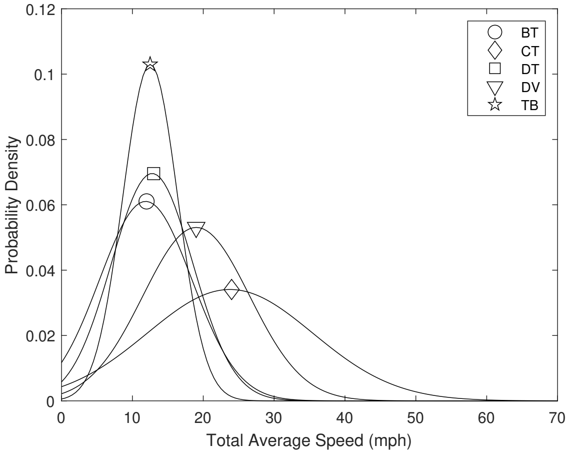

Figure 3.1. Probability Density of Total Average Speed for each vocation.

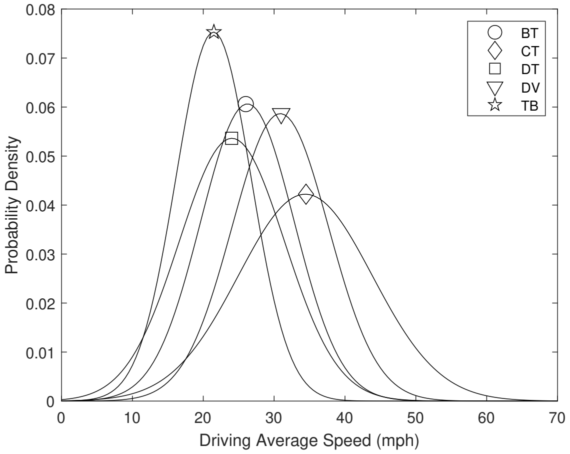

Figure 3.2. Probability Density of Driving Average Speed for each vocation.

Each vocation represents a group of vehicles that perform similar tasks. A detailed description of each vocation in the fleet DNA is provided in [15]. Figures 3.1 and 3.2, show the probability density functions under the assumption of uni-modal normal distribution for two example features from the dataset: Total Average Speed and Driving Average Speed for all the vocations. These figures illustrate the complexity of vocation identification. Some of the vocations (e.g., TB) have a distinct operational profile while others have an operational profile that can be confounded with the remaining vocations. The similarity in operating profiles between BT and DT is notable in both figures. Delivery Vans (DV) and Delivery Trucks (DT) are also expected to have a high level of similarity since the main difference between these two vocations is the vehicle weight, with DT vehicles being typically heavier than DV vehicles.

Bucket Trucks (BT) perform tasks at the job site and will possibly spend less time driving from one point to another. Therefore, compared to DT, DV and TB vehicles, their operational profile will show lower distances traveled and lower average speeds (Figure 3.1). Class 8 Tractors (CT) are typically used to haul a trailer from a source (e.g., distribution center) to a destination (e.g., customer site). Therefore, CT vehicles are expected to travel long distances over highways compared to DT or DV vehicles. However, according to [15], the CT vocation consists of various types of class 7 and 8 vehicles that can be used for different tasks ranging from food delivery to long-hauling tasks. This variation in the CT profile explains some of the results later discussed in Section 3.2. Lastly, the Transit Bus (TB) profile exhibits frequent stops with low average speeds since TB vehicles are used for short-distance public transportation.

3.1.2 Model Development

Each model is developed using either the KM or EM training algorithms. The training is executed for each vocation independently. It starts by randomly selecting a set of initial centroids for the target vocation from the available training data. During each training iteration, records from the training data are compared to each centroid of the vocation. After processing all records, the centroids are updated and a new training iteration begins. Both the KM and EM algorithms are well studied in the literature. However, for the purpose of completeness and in order to support the discussion of the proposed methodology and results, a summary of the main steps in these algorithms is provided using the following notation:

- $\mathbf{r}_i = (r_i[1], r_i[2], ..., r_i[n])$ represents a record where each element $r_i[.]$ of the input vector $\mathbf{r}_i$ is the value of one of the input features f and n is the total number of features.
- $\mathbb{C}_v = \{\mathbf{cv}_1, \mathbf{cv}_2, ..., \mathbf{cv}_m\}$ is the set of centroids of vocation $\mathbf{v}$ where each centroid represents a cluster of the vocation $\mathbf{v} \in \mathbb{V} = \{BT, CT, DV, DT, TB\}$. In the first implementation of the vocation identification application, the number of centroids, m, is fixed for each vocation. Chapter 4 introduces techniques that vary m.

Under the KM algorithm, the conditional probability of a record $\mathbf{r}_i$ being assigned to $\mathbf{cv}_\mathrm{j}$ is binary and based on the selected distance measure (e.g., Euclidean or cosine distance) $d(\mathbf{r}_i, \mathbf{cv}_\mathrm{j})$ as defined in (3.1)

$$P_{KM}(\mathbf{cv}_\mathrm{j}/\mathbf{r}_i) = \begin{cases} 1, & \text{if } d(\mathbf{r}_i, \mathbf{cv}_\mathrm{j}) = \underset{1 \leq k \leq m}{\operatorname{argmin}} \{d(\mathbf{r}_i, \mathbf{cv}_k)\} \\ 0, & \text{otherwise.} \end{cases} \tag{3.1}$$

In the case of EM, the evaluation of this probability is referred to as the *Expectation* step. Using Bayes' rule, with the assumption that each feature has a normal distribution and that all the features are independent, the conditional probability $P_{EM}(\mathbf{cv}_{\mathrm{j}}/\mathbf{r}_i)$ is given by (3.2)

$$\begin{aligned} P_{EM}(\mathbf{cv}_{\mathrm{j}}/\mathbf{r}_i) &= \frac{P(\mathbf{cv}_{\mathrm{j}})P(\mathbf{r}_i/\mathbf{cv}_{\mathrm{j}})}{\sum\limits_{k=1}^{m} P(\mathbf{cv}_k)P(\mathbf{r}_i/\mathbf{cv}_k)} \\ &= \frac{P(\mathbf{cv}_{\mathrm{j}}) \prod\limits_{f=1}^{n} P(r_i[f]/\mathbf{cv}_{\mathrm{j}})}{\sum\limits_{k=1}^{m} P(\mathbf{cv}_k) \prod\limits_{f=1}^{n} P(r_i[f]/\mathbf{cv}_k)} \end{aligned} . \tag{3.2}$$

where each term of the form $P(r_i[f]/\mathbf{cv}_{\mathrm{j}})$ is derived from a normal distribution with a mean $\mu(cv_{\mathrm{j}}[f])$ and a standard deviation $\sigma(cv_{\mathrm{j}}[f])$ according to (3.3)

$$\begin{aligned} P(r_i[f]/\mathbf{cv}_{\mathrm{j}}) &= \mathcal{N}(\mu(cv_{\mathrm{j}}[f]),\, \sigma(cv_{\mathrm{j}}[f])^2) \\ &= \frac{1}{\sigma(cv_{\mathrm{j}}[f])\sqrt{2\pi}} \cdot \mathrm{e}^{-\frac{1}{2}\left(\frac{r_i[f]-\mu(cv_{\mathrm{j}}[f])}{\sigma(cv_{\mathrm{j}}[f])}\right)^2}. \end{aligned} \tag{3.3}$$

For illustration purposes, Figures 3.1 and 3.2 show this probability density when all the data for a given vocation are assigned to one cluster ($m = 1$) and where f is the Total Average Speed or the Driving Average Speed, respectively. These figures also depict the differences in mean (μ) and standard deviation (σ) among the vocations for each respective feature.

At the end of each iteration of either the KM or EM algorithms, the variables $P(\mathbf{cv}_{\mathrm{j}})$, $\mu(cv_{\mathrm{j}}[f])$, and $\sigma(cv_{\mathrm{j}}[f])$ are updated for each feature f and centroid j of vocation $\mathbf{v}$ using (3.4), (3.5) and (3.6), respectively

$$P(\mathbf{cv}_{\mathrm{j}}) = \frac{1}{N}\sum_{i=1}^{N} P(\mathbf{cv}_{\mathrm{j}}/\mathbf{r}_i) \tag{3.4}$$

$$\mu(cv_{\mathrm{j}}[f]) = \frac{1}{N \cdot P(\mathbf{cv}_{\mathrm{j}})}\sum_{i=1}^{N} P(\mathbf{cv}_{\mathrm{j}}/\mathbf{r}_i) \cdot r_i[f] \tag{3.5}$$

$$\sigma(cv_{\mathrm{j}}[f])^2 = \frac{1}{N \cdot P(\mathbf{cv}_{\mathrm{j}})}\sum_{i=1}^{N} P(\mathbf{cv}_{\mathrm{j}}/\mathbf{r}_i) \cdot (r_i[f] - \mu(cv_{\mathrm{j}}[f]))^2 \tag{3.6}$$

where N is the number of records in the training set of vocation $\mathbf{v}$. Equations (3.4) and (3.5) can be simplified considerably in the case of the KM algorithm. The equations were kept in this form in order to highlight the alignment in the update procedure between the two algorithms. This update procedure for EM is referred to as the *Maximization* step. Moreover, (3.6) is not needed in the KM algorithm. It is only calculated at the end of the training or testing phases in order to support feature reduction as discussed next.

3.1.3 Feature Reduction

Even though the starting dataset was manually reduced from 350 parameters to 15 features as described in Section 3.1.1, a minimalist model is desirable in order to limit the deployment cost of the vocation identifier and increase its applicability in production. This minimalist model should only include the features that are necessary and practical for vocation identification. Feature reduction was performed using the wrapper induction method[27] which was used to remove any feature that does not contribute to vocation identification. During each iteration of the feature reduction process, the standard deviation of each feature is evaluated and the feature is removed if the standard deviation as defined in (3.6) is below a certain pre-set threshold across all the clusters. One feature was considered per iteration until none of the features had a standard deviation below this threshold. In addition, features that are easier to collect (e.g., vehicle speed) were favored over features that may not be readily available (e.g., characteristic acceleration, kinetic energy density).

In the remainder of the thesis, the model with the full feature set is labeled FFmodel and the reduced feature model is labeled RFmodel.

3.1.4 Vocation Assignment

Once the model is trained, it is exposed to a record $\mathbf{r}_i$ from an unknown vehicle. That is, for each vocation $\mathbf{v}$ and centroid $\mathbf{cv}_{\mathrm{j}}$ of $\mathbf{v}$, the probability $P(\mathbf{cv}_{\mathrm{j}}/\mathbf{r}_i)$ is calculated using either (3.1) or (3.2) for KM or EM, respectively. The record is then assigned to the vocation - $v^T(\mathbf{r}_i)$ - with the largest probability according to the following equation:

$$v^T(\mathbf{r}_i) = \operatorname*{argmax}_{\mathbf{v}\in\mathbb{V}} \left\{ \operatorname*{argmax}_{1\leq \mathrm{j}\leq m} \{P(\mathbf{cv}_\mathrm{j}/\mathbf{r}_i)\} \right\}.. \tag{3.7}$$

Equation 3.7 is used for a single daily record from an unknown vehicle. When the unknown vehicle has multiple records, each record can be assigned to a different vocation and a consensus is needed to select the winning vocation. Let $\mathbb{R} = \{\mathbf{r}_1, \mathbf{r}_2, ..., \mathbf{r}_p\}$ represent the set of records of the unknown vehicle. The winning vocation of the unknown vehicle is the vocation that is assigned the highest number of records. This process is defined by the following equation:

$$voc^T(\mathbb{R}) = \operatorname*{argmax}_{\mathbf{v}\in\mathbb{V}} \left\{ \sum_{i=1}^{p} v^T(\mathbf{r}_i) = \mathbf{v} \right\}. \tag{3.8}$$

Equations 3.7 and 3.8 show the *traditional one-versus-all* (T) assignment where all the vocations compete for the same vehicle at once. While computationally efficient, this assignment has an important limitation since the wrong vocations may weaken the chances of the correct vocation by acquiring several of the records of the unknown vehicle. This aspect is particularly important for the current application because the number of vocations can be large and the number of daily records available for each unknown vehicle is small.

In order to mitigate this potential limitation, the one-versus-one *round-robin tournament* (R) assignment was investigated. This assignment consists of multiple rounds where each vehicle is exposed to every combination of two vocations. The vocation of choice is the one that is assigned the most records across all of the rounds for a given vehicle as defined in (3.9) and (3.10)

$$v^R(\mathbf{r}_i, \mathbf{a}, \mathbf{b}) = \operatorname*{argmax}_{\mathbf{v}\in\{\mathbf{a},\mathbf{b}\}} \left\{ \operatorname*{argmax}_{1\leq k\leq m} \{P(\mathbf{cv}_k/\mathbf{r}_i)\} \right\} \tag{3.9}$$

$$voc^R(\mathbb{R}) = \operatorname*{argmax}_{\mathbf{v}\in\mathbb{V}} \left\{ \sum_{\mathbf{a}\neq\mathbf{v}} \sum_{i=1}^{p} v^R(\mathbf{r}_i, \mathbf{a}, \mathbf{v}) = \mathbf{v} \right\}. \tag{3.10}$$

Unfortunately, the round-robin assignment has a quadratic time complexity with respect to the number of vocations. The *tournament bracket* (B) also follows the one-versus-one assignment and consists of multiple rounds where the unknown vehicle is only exposed to two vocations in each round. However, in the bracket assignment, a vocation is eliminated in each round. The vocation that is retained is the one that collects the highest number of records from the unknown vehicle in the round and this vocation proceeds to the next round. The assignment concludes when only one vocation remains.

Equation (3.11) shows the selection between two vocations **a** and **b** for one round of the bracket assignment. This equation is applied recursively in order to determine the winning vocation as shown in (3.12)

$$w^B(\mathbb{R}, \mathbf{a}, \mathbf{b}) = \underset{\mathbf{v} \in \{\mathbf{a},\mathbf{b}\}}{\operatorname{argmax}} \left\{ \sum_{i=1}^{p} v^R(\mathbf{r}_i, \mathbf{a}, \mathbf{b}) = \mathbf{v} \right\} \tag{3.11}$$

$$voc^B(\mathbb{R}) = w^B\left(\mathbb{R}, \mathbf{v}_c, w^B(\mathbb{R}, \mathbf{v}_{c-1}, \mathbf{v}_{c-2})\right) \tag{3.12}$$

where c is the number of vocations in $\mathbb{V}$. As opposed to the round-robin assignment, Equation (3.12) is only executed $c - 1$ times allowing the bracket assignment to have a linear time complexity with respect to the number of vocations.

3.2 Results and Discussion

The traditional one-versus-all, round-robin and bracket assignments are applied to the dataset described in Table 3.2. During training, the centroids of each vocation are determined using 130 daily records from each vocation. The model is then exposed to the testing vehicles. As indicated earlier, the training and testing vehicles are distinct and the number of testing vehicles varies per vocation as available in the dataset. However, the number of records for each test vehicle is fixed to 13. The next subsections compare the KM and EM algorithms with and without feature reduction using the three assignment methods.

Table 3.3. Vocation assignment of the test vehicles using the traditional one-versus-all KM and EM FFmodels.

	Vocation	BT	CT	DT	DV	TB
KM	BT	1	0	1	0	0
	CT	2(1)	20	2(2)	4(1)	2(2)
	DT	0	0	8	4	7
	DV	4	1	4	6	1
	TB	0	0	0	0	11
EM	BT	2	0	0	0	0
	CT	4(1)	19(1)	5(1)	0	3(1)
	DT	1(1)	0	15	0(1)	2
	DV	3	0(1)	3(1)	8	1
	TB	0	0	0	0	11

3.2.1 One-Versus-All Assignment

Table 3.3 shows the confusion matrix of the one-versus-all multi-class vocation identification FFmodel with KM and EM clustering. The results are presented in this manner in order to facilitate the analysis of confounding vocations and the identification of vocations with unique profiles.

The assignment of a vehicle to a vocation follows (3.8). Each row in the table represents the test vehicles of a vocation. The entries represent the number of vehicles of the target vocation (row) that are assigned to a given vocation (column). The numbers in between parenthesis represent the number of ties for each vocation. For example, the CT vocation has a total of 33 test vehicles (Table 3.2). Using the KM algorithm, 20 of these vehicles were correctly assigned to the CT vocation, 2 vehicles were assigned to the DT vocation, and all vocations except CT included at least one tie. The number of test vehicles that are correctly assigned (i.e., true positives) for each vocation is shown across the diagonal of the table. The KM FFmodel was able to correctly classify 46 out of the 81 test vehicles whereas the EM FF model shows 55 true positives.

Only one of the BT test vehicles was assigned to a different vocation under the two FFmodels. Despite the low number of test vehicles in this vocation (Table 3.2), this is still an indication of the unique profile of the vocation. TB is another vocation with a distinct operational profile with no vehicles incorrectly classified under both FFmodels. The large number of DT vehicles that are assigned to the DV vocation indicates that the two vocations may be similar.

Feature reduction using the approach described in Section 3.1.3 was performed on the models. The features that were eliminated include: Total Speed Standard Deviation, Average Kinetic Power Density Demand, Cumulative Instantaneous Kinetic Energy Density, Characteristic Acceleration, Aerodynamic Speed, and Max Deceleration. These are indicated by a ’*’ in Table 3.1. The full feature model (FFmodel) includes 15 features whereas the reduced feature model (RFmodel) includes only 9 features which can all be derived from two parameters: speed and distance traveled. These parameters are readily available for all vehicles without the need for additional instrumentation.

Table 3.4. Vocation assignment of the test vehicles using the traditional one-versus-all KM and EM RFmodels.

	Vocation	BT	CT	DT	DV	TB
KM	BT	1	0	0	1	0
	CT	2(2)	19(2)	2(2)	4(1)	2(2)
	DT	0	0	11	3	5
	DV	4(1)	1	3(1)	7	0
	TB	0	0	0	0	11
EM	BT	2	0	0	0	0
	CT	2(1)	19	7(1)	0	4
	DT	0	0	15(1)	0	3(1)
	DV	5(1)	0	1(1)	9(1)	0
	TB	0	0	0	0	11

Table 3.4 shows the confusion matrix of the RFmodel under KM and EM. The model generated 49 and 56 true positives with KM and EM, respectively. The number of true positives for the reduced and full feature models are similar. One limitation of traditional clustering assignment is the number of tie assignments that result. The following one-versus-one assignment techniques are introduced in order to improve the number of true positives and reduce the number of ties.

3.2.2 Round-Robin Assignment

Table 3.5 shows the confusion matrix of the FFmodel with the round-robin assignment. The KM and EM models correctly classified 44 and 54 test vehicles, respectively. The number of true positives is comparable to that of the corresponding traditional one-versus-all model. However, the round-robin assignment does not suffer from ties. The numbers of true positives for the KM and EM RFmodels with round-robin assignment are 51 and 57, respectively (Table 3.6).

Table 3.5. Vocation assignment of the test vehicles using the round-robin one-vs-one KM and EM FFmodels.

	Vocation	BT	CT	DT	DV	TB
KM	BT	1	0	1	0	0
	CT	6	18	3	4	2
	DT	2	0	8	4	5
	DV	5	1	3	6	1
	TB	0	0	0	0	11
EM	BT	2	0	0	0	0
	CT	4	18	5	0	6
	DT	2	0	15	0	2
	DV	2	0	5	8	1
	TB	0	0	0	0	11

Table 3.6. Vocation assignment of the test vehicles using the round-robin one-vs-one KM and EM RFmodels.

	Vocation	BT	CT	DT	DV	TB
KM	BT	2	0	0	0	0
	CT	5	18	3	5	2
	DT	0	0	11	3	5
	DV	3	1	3	9	0
	TB	0	0	0	0	11
EM	BT	2	0	0	0	0
	CT	1	18	9	0	5
	DT	0	0	16	0	3
	DV	5	0	1	10	0
	TB	0	0	0	0	11

As in the case of the one-versus-all assignment, EM performs better than KM for the round-robin models. The results for the round-robin approach are comparable to the traditional approach, but vocations (such as CT) consistently have less true positives when the round-robin approach is used. We speculate that this is the case because the CT vocation, as discussed in Section 3.1.1, is actually a combination of two or more vocations. When pairwise comparisons are performed during each round of the round-robin, one of the wrong vocations can eliminate the CT vocation depending on whether the test vehicle belongs to one of the sub-vocations or the other.

3.2.3 Bracket Assignment

Tables 3.7 and 3.8 include the result of the bracket assignment for the FFmodel and RFmodel, respectively. As in the case of the round-robin assignment, the bracket assignment does not suffer from ties and the number of true positives generated by the respective models is nearly the same. In fact, the model with the highest number of true positives is the bracket RFmodel. While the difference in performance may be marginal, the bracket RF model offers several practical advantages: It scales linearly with respect to the number of vocations; it is less susceptible to an increasing number of vocations since only two vocations are compared at a time; and it uses a reduced feature set that is readily available.

Table 3.7. Vocation assignment of the test vehicles using the bracket one-vs-one KM and EM FFmodels.

	Vocation	BT	CT	DT	DV	TB
KM	BT	1	0	1	0	0
	CT	4	18	4	5	2
	DT	2	0	8	3	6
	DV	6	1	3	5	1
	TB	0	0	0	0	11
EM	BT	2	0	0	0	0
	CT	5	19	4	0	5
	DT	1	0	15	1	2
	DV	3	0	4	8	1
	TB	0	0	0	0	11

Table 3.8. Vocation assignment of the test vehicles using the bracket one-vs-one KM and EM RFmodels.

	Vocation	BT	CT	DT	DV	TB
KM	BT	2	0	0	0	0
	CT	4	18	3	6	2
	DT	0	0	11	3	5
	DV	4	1	4	7	0
	TB	0	0	0	0	11
EM	BT	2	0	0	0	0
	CT	1	19	7	0	6
	DT	0	0	16	0	3
	DV	5	0	1	10	0
	TB	0	0	0	0	11

The above results focus on the true positive assignments generated by each model. They show that the bracket model delivers the same or higher number of correct assignments compared to the other models while being computationally more efficient than the round-robin model and more scalable than the one-versus-all model.

3.2.4 Centroids

The results improved when the RFmodel was used instead of the FFmodel, which indicates that certain features have a different level of importance to some vocations. By removing 6 of the features, some vocations (such as BT) were able to achieve more true positives, suggesting that those removed features were unimportant or even confounding to the identification of the correct vocation. The development of centroids is also affected by the removal of features. Analysis of the centroids developed using the FFmodel and RFmodel will illustrate how the removed features affect the vocational profile of each vocation. Centroid analysis is completed for the BT and CT vocations. BT is selected because of the increase in performance from the KM FFmodel to the KM RFmodel, and CT is selected because it is the vocation with the highest number of test vehicles.

The values for the BT and CT centroids for the FFmodel are shown in Tables 3.9 and 3.10, respectively. The values for the BT and CT centroids for the RFmodel are shown in Tables 3.11 and 3.12. These tables show the mean and the standard deviation for each feature and cluster of the target vocation after training is completed. The mean represents the centroid of the cluster. The standard deviation is an indication of the tightness of the cluster. The units of the features are included in Table 3.1. During training, the initial centroids (for the FFmodel or RFmodel) are set to specific training records. This means that the developed centroids for the different models could provide some insight regarding the operation modes of the respective vocations. For example, for both models, the CT centroids have a higher mean value for max speed and total average speed. This is expected because of the long distance deliveries that CT vehicles must complete.

In general, the FF and RF models generate similar centroids. This is anticipated since the two models use the same training data. However, some differences occur. For example, the CT centroids generated using EM clustering and the FFmodel have a range of mean total stops of 24.52 to 39.34. On the other hand, the CT centroids generated using EM clustering and the RFmodel have a range of mean total stops of 20.1 to 51.24. This range can be attributed to some clusters having more of the vehicles than others. With fewer features, each feature in the RFmodel has more bearing on the final classification. Therefore, features (like total stops) must accurately cover a wider range of values than for the FFmodel.

Table 3.9. BT Centroids generated with the EM Clustering FFmodel.

Features	Metric	BT0	BT1	BT2	BT3	BT4
max spd	Mean	64.99	48.92	60.69	66.14	45.97
	STD	4.26	6.99	4.83	4.14	2.96
total avg spd	Mean	10.67	6.04	20.55	14.4	13.6
	STD	4.27	2.64	8.83	0.49	3.01
total spd std	Mean	18.27	11.03	19.58	18.11	14.99
	STD	2.95	2.11	2.52	1.27	1.14
drive avg spd	Mean	34.8	19.41	33.16	26.99	24.18
	STD	4.47	2.52	4.75	1.6	2.34
drive spd std	Mean	17.92	12.95	17.28	16.63	12.52
	STD	1.49	1.93	1.77	1.87	0.94
zero seconds	Mean	10.13	6.91	2.29	6.19	2.02
	STD	4.09	3.93	2.02	1.23	1.45
distance total	Mean	39.56	14.82	23.72	51.36	15.01
	STD	14.51	8.14	12.68	11.54	7.29
total stops	Mean	22.98	29.97	11.73	65.7	29.36
	STD	7.97	22.69	6.58	27.77	14.16
avg kin pwr density demand	Mean	3.16	2.3	2.6	3.31	4.26
	STD	0.5	0.5	0.57	0.48	0.41
cuml instant KE density	Mean	0.63	0.15	0.37	0.72	0.16
	STD	0.25	0.11	0.21	0.15	0.11
char accel	Mean	0.17	0.18	0.15	0.18	0.22
	STD	0.02	0.03	0.02	0.02	0.02
aero spd	Mean	17.95	11.83	17.23	15.71	12.49
	STD	1.35	1.54	1.42	1.45	0.92
max accel	Mean	5.28	5.62	4.23	11.43	8.85
	STD	0.61	2.43	0.87	7.4	2.69
avg accel	Mean	0.92	1.02	0.76	1.24	1.65
	STD	0.16	0.24	0.15	0.22	0.12
max decel	Mean	-7.23	-7.54	-6.16	-11.84	-8.51
	STD	0.96	2.99	1.19	4.81	1.7

Table 3.10. CT Centroids generated with the EM Clustering FFmodel.

Features	Metric	CT0	CT1	CT2	CT3	CT4
max spd	Mean	65.23	66.32	70.22	72.11	70.33
	STD	5.89	4.63	1.17	1.11	1.05
total avg spd	Mean	33.49	28.9	40.55	42.55	37.4
	STD	2.72	4.89	3.55	2.62	1.28
total spd std	Mean	23.54	24.2	22.86	24.64	21.15
	STD	1.7	2.09	1.18	0.94	0.6
drive avg spd	Mean	41.44	36.9	45.88	48.07	42.01
	STD	2.4	4.35	3.6	2.06	1
drive spd std	Mean	19.06	21.5	18.63	20.59	17.6
	STD	1.34	1.76	1.47	0.87	0.4
zero seconds	Mean	3.16	2.1	1.76	1.68	2.05
	STD	1.18	1.07	6.39	5.34	4.33
distance total	Mean	155.59	73.01	170.23	169.04	192.57
	STD	38.96	22.13	40.73	10.71	5.63
total stops	Mean	39.34	34.54	24.52	25.12	35.21
	STD	11.14	16.84	8.09	6.58	6.58
avg kin pwr density demand	Mean	2.25	2.11	2.12	2.21	2.53
	STD	0.69	0.36	0.28	0.09	0.18
cuml instant KE density	Mean	2.82	1.31	3.28	3.46	3.42
	STD	0.75	0.39	0.81	0.16	0.11
char accel	Mean	0.12	0.12	0.14	0.13	0.15
	STD	0.04	0.02	0.01	0	0
aero spd	Mean	20.1	20.06	21.35	22.72	19.75
	STD	1.06	1.53	0.87	0.43	0.3
max accel	Mean	4.65	4.11	3.83	3.83	4.06
	STD	0.83	0.61	0.26	0.38	0.53
avg accel	Mean	0.58	0.65	0.48	0.5	0.57
	STD	0.11	0.11	0.07	0.03	0.03
max decel	Mean	-6.73	-6.13	-6.24	-6.77	-6.39
	STD	0.99	1.06	1.08	1.02	0.95

Table 3.11. BT Centroids generated with the EM Clustering RFmodel.

Features	Metric	BT0	BT1	BT2	BT3	BT4
max spd	Mean	47.9	64.11	55.66	43.19	55.76
	STD	4.03	4.03	2.12	3.1	10.83
total avg spd	Mean	4.71	16.28	8.99	10.66	11.41
	STD	2.31	8.81	3.19	4.01	3.44
drive avg spd	Mean	19.26	34.96	24.46	20.37	23.09
	STD	2.95	4	2.59	3.53	3.83
drive spd std	Mean	12.93	17.78	16.24	11.31	13.73
	STD	1.21	1.6	1.03	1.12	2.25
zero seconds	Mean	7.66	6.6	4.32	3.5	6.36
	STD	4.77	5.23	2.08	2.56	2.63
distance total	Mean	10.04	33.9	15.45	15.33	36.13
	STD	5.63	15.44	6.42	6.24	13.85
total stops	Mean	17.51	18.18	16.43	35.51	66.32
	STD	8.52	9.59	6.58	17.27	23.75
max accel	Mean	4.25	4.91	4.11	7.67	11.36
	STD	0.94	0.87	0.68	1.22	4.78
avg accel	Mean	0.9	0.86	0.87	1.44	1.36
	STD	0.17	0.18	0.26	0.29	0.18

Table 3.12. CT Centroids generated with the EM Clustering RFmodel.

Features	Metric	CT0	CT1	CT2	CT3	CT4
max spd	Mean	70.27	66.97	68.39	71.23	56.14
	STD	1.52	3.6	0.59	1.35	0.8
total avg spd	Mean	37.5	27.87	30.71	44.06	27.73
	STD	2.53	7.59	2.82	2.34	4.85
drive avg spd	Mean	43.09	36.63	38.5	49.28	36.72
	STD	2.41	7.7	2.79	2.18	4.32
drive spd std	Mean	19.07	19.88	22.05	19.01	18.78
	STD	1.37	2.67	1.18	1.89	0.75
zero seconds	Mean	2.19	3.31	1.78	1.39	3.43
	STD	0.71	1.79	0.59	0.35	1.12
distance total	Mean	176.96	112.24	73.34	164.41	110.26
	STD	30.48	74.98	15.72	31.96	34.66
total stops	Mean	32.13	43.27	28.74	20.1	51.24
	STD	6.58	24.23	8.08	6.58	14.82
max accel	Mean	4.04	4.26	3.99	3.78	5.02
	STD	0.53	0.29	0.55	0.27	0.88
avg accel	Mean	0.55	0.71	0.62	0.45	0.6
	STD	0.06	0.2	0.06	0.06	0.15

4. CLUSTERS PER VOCATION

The number of clusters needed for each vocation may be different depending on the operating profile of the vehicles. In the previous chapter, all vocation identification models used a fixed number of clusters across all vocations. This number was established manually and was set to 5. This chapter investigates the use of three algorithms that can establish the optimal number of clusters for each vocation. These algorithms are:

- Elbow method, which was previously introduced in [13] and described in Section 2.2,
- Dynamic clustering, which decides on the split or removal of a cluster during training using a fixed threshold, and
- Dynamic clustering with simulated annealing, which splits and removes clusters based on a variable threshold during training.

4.1 Elbow Method

Finding the optimal number of clusters can be completed entirely prior to training, with an approach such as the Elbow method. The Elbow method involves some measure of cluster compactness, in this case the inertia, which is given by:

$$Inertia = \sum_{i=1}^{N} d(\mathbf{r}_i, \mathbf{cv}_{\mathrm{j}}) \tag{4.1}$$

where N is the number of records in the cluster and d is the Euclidean distance. The Elbow method is executed multiple times, increasing the number of clusters in each vocation by one after each training. Following each training, the average inertia value for each vocation is recorded and plotted on a graph, as shown in Figures 4.1 and 4.2. The elbow point is defined by the change from exponential to linear decrease in inertia. This represents the point where adding clusters no longer significantly impacts the inertia of the vocation. This definition introduces one of the limitations of the approach. The definition of ”significantly” can be subjective and identifying it automatically can be difficult.

Another issue with the Elbow approach is the selection of centroids. Since the number of centroids increases, at least one new centroid must be randomly introduced after each iteration. This can lead to an increase or decrease in inertia, which could make identifying the elbow point more difficult. This behavior is shown in Figures 4.1 and 4.2 where the inertia may increase for a given vocation with a higher number of clusters. For example, the inertia for the DT vocation in Figure 4.1 increases when the number of clusters increases from 3 to 4.

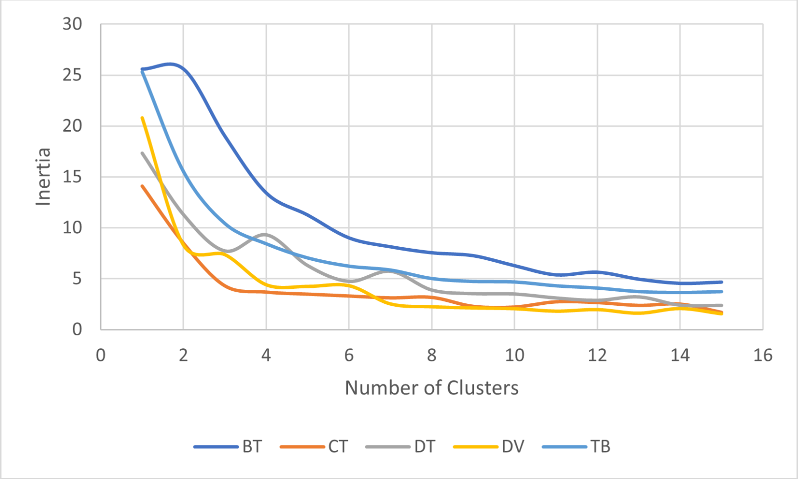

Figure 4.1. KM Elbow Method.

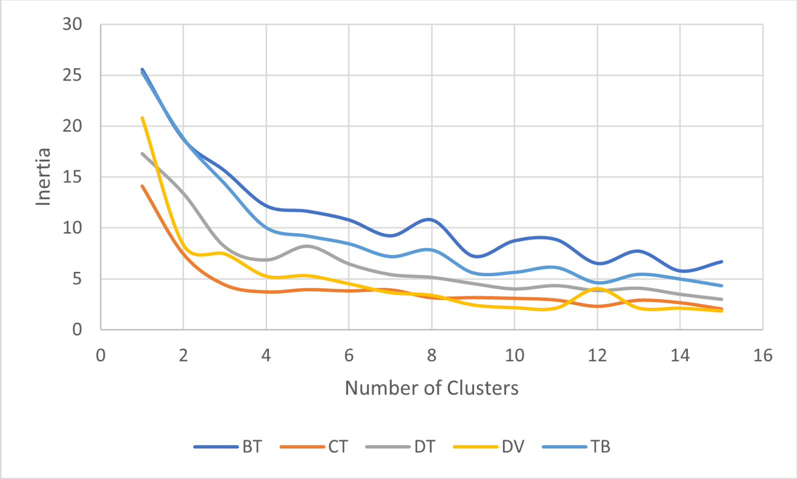

Figure 4.2. EM Elbow Method.

In this study, the elbow point was selected as the last local maxima for the slope of the elbow graph. This was used to avoid undershooting on the number of clusters needed for a vocation. The last local maximum system was also used for the automatic selection of the elbow point. In other words, a threshold was set to determine whether a local minimum or maximum was significant enough to be considered as a candidate for being an elbow point. The final significant maximum point for the derivative of the inertia graph (representing a large decrease in inertia) was selected as the last elbow point. This allowed the automatic elbow point selection to choose a higher number of clusters, which would (in theory) lead to more true positive results.

4.2 Dynamic Clustering

The number of clusters can also be adjusted dynamically during training. Two dynamic methods are considered. One of these methods splits and removes clusters based on the cluster's average standard deviation, in comparison to the standard deviation of all of the clusters of that vocation. Two pre-defined constants *Rem* and *Spl* specify how much lower or higher the cluster's standard deviation must be to merit removal or splitting. The ranges for these constants are $0 < Rem < 1$ and $1 < Spl < 2$. The values for *Rem* and *Spl* are used as a multiplier to determine the threshold for removal or splitting as shown below

$$Thresh_{Rem} = Rem * \overline{\mu_\sigma} \tag{4.2}$$

$$Thresh_{Spl} = Spl * \overline{\mu_\sigma} \tag{4.3}$$

where $\overline{\mu_\sigma}$ is the average standard deviation of all of the clusters. This average is calculated using the following equations:

$$\mu_\sigma(c) = \frac{\sum_{i=1}^{n} \sigma_i}{n} \tag{4.4}$$

where n is the number of features. In 4.4, σ_i represents the standard deviation of feature i.

$$\overline{\mu_\sigma} = \frac{\sum_{\mathrm{j}=1}^{m} \mu_\sigma(\mathrm{j})}{m} \tag{4.5}$$

where m is the number of clusters. In 4.5, $\mu_\sigma(\mathrm{j})$ is the value for each cluster, which is calculated with Equation 4.4. The ranges for Rem and Spl are set based on decreasing or increasing the standard deviation threshold value, respectively.

A very small standard deviation corresponds to a small cluster that does not include many records, whereas a very large standard deviation would indicate the cluster contains a wide range of records. As a result, Rem must be less than 1 to make the removal threshold ($Thresh_{Rem}$) lower than the average of the cluster standard deviations ($\overline{\mu_\sigma}$ from 4.5). Additionally, Spl must be between 1 and 2 to ensure the split threshold ($Thresh_{Spl}$) will be higher than μ_σ.

This approach keeps the criteria for splitting and removal constant, meaning that the algorithm is not allowed to make ”mistakes”. Therefore, the algorithm may be unable to seek a globally optimal solution.

Simulated Annealing[26] presents a solution to this issue, allowing the clusters to be split and removed as above, but also allowing mistakes with a probability that gets smaller as the training process continues. The simulated annealing method introduces some additional parameters defined as follows:

$$Temp = \mathrm{e}^{-k*Iter} \tag{4.6}$$

$$Pr = \mathrm{e}^{\frac{-|\mu_\sigma(c)-\overline{\mu_\sigma}|}{Temp}} \tag{4.7}$$

where k is a constant that determines how quickly the probability (Pr) of a split or removal is reduced for every training iteration, $Temp$ represents the temperature value, and $Iter$ corresponds to the current training iteration number.

For all of the following tests, the initial centroids, which would usually be selected randomly, are set so that each run of the clustering program will start with the same initial centroids. Other parameters that stay constant are the number of iterations (1000 iterations), the feature set (all of the features in the Fleet DNA RFmodel), and the number of initial centroids (5 per vocation). The period of evaluations is set at 100 iterations, meaning that every 100 training iterations the algorithm uses the Dynamic Method or Simulated

Annealing to check for cluster splits or removals. However, following the final iteration there is no evaluation, as introducing new clusters or removing established ones immediately before testing would make results worse. Therefore, with a period of 100 iterations, there are 1000/100 - 1 = 9 evaluations. The process is shown in Algorithm 1.

Algorithm 1: Dynamic cluster split and removal algorithm.

```
if (Iter+1) % Period == 0 and (Iter+1) != NumIterations then
    ▷ Equation 4.6
    $Temp = exp(-k * Iter)$
    for $v$ in $Vocations$ do
        for $c$ in $v$ do
            if $\mu_\sigma(c) < Rem * \overline{\mu_\sigma}$ then
                delete($c$)
            else if $\mu_\sigma(c) > Spl * \overline{\mu_\sigma}$ then
                split($c$)
            else
                $Pr = exp(-abs(std(c) - \overline{\mu_\sigma})/Temp)$
                if $random() < Pr$ then
                    if random() < 0.5 or $m$ < 2 then
                        split($c$)
                    else
                        delete($c$)
                    end
                end
            end
        end
    end
end
```

4.3 Results and Discussion

In order to compare the performance of the Elbow and the Dynamic algorithms, a baseline must be established. The results of the baseline are identical to those in Table 3.4 and are repeated in Table 4.1 for convenience.

Table 4.1. Vocation assignment of the test vehicles using the traditional one-versus-all KM and EM RFmodels with a fixed number of clusters (5).

Clustering	Vocation	BT	CT	DT	DV	TB
KM	BT	1	0	0	1	0
	CT	2(2)	19(2)	2(2)	4(1)	2(2)
	DT	0	0	11	3	5
	DV	4(1)	1	3(1)	7	0
	TB	0	0	0	0	11
EM	BT	2	0	0	0	0
	CT	2(1)	19	7(1)	0	4
	DT	0	0	15(1)	0	3(1)
	DV	5(1)	0	1(1)	9(1)	0
	TB	0	0	0	0	11

Using the Elbow method resulted in a number of clusters shown in Table 4.2. The table includes both the automatically set number of clusters and the cluster number manually selected. The number of true positives for each vocation is shown in Table 4.3.

Table 4.2. Number of Clusters derived using the Elbow Method with traditional KM and EM clustering.

Clustering	Method	BT	CT	DT	DV	TB
KM	Auto	10	2	7	6	2
KM	Manual	7	3	8	7	4
EM	Auto	13	11	3	12	11
EM	Manual	12	4	6	4	4

As shown in 4.3, the best results with the Elbow method occurred when EM clustering was used and the elbow points were selected manually. For that test, the results showed an improvement over both classical KM and EM. The other methods all had results worse than those of the corresponding classical methods. The most notable improvement was achieved

Table 4.3. Vocation assignment of the test vehicles using the one-versus-all RFmodel with the number of clusters generated by the Elbow method.

Clustering	Method	BT	CT	DT	DV	TB	Total
KM	Auto	2	20	10	10	10	52
KM	Manual	1(1)	18(3)	9(1)	10	11	49(5)
EM	Auto	2	17(1)	11	8(1)	11	49(2)
EM	Manual	2	28	6	10(1)	11	57(1)

for the classification of the CT vehicle. The EM manual elbow selection method correctly identified the vocation of 28 vehicles, a significant improvement over the 19 correct (with 2 tie-correct) identification generated by the classical KM and the 19 correct generated by the classical EM. Notably, the EM manual elbow method only required 4 clusters for CT and performed better than the EM automatic elbow method, which had 11 clusters for CT. While this may appear to confirm that less clusters is better for CT, it is important to note that some of the increase in accuracy could be due to the decrease in number of clusters for other vocations (such as DV and TB). The only vocation that had a decrease in true positives from classical EM to EM with the manual elbow cluster selection method was DT. This vocation also had one more cluster under the EM manual elbow method. Clearly, the most accurate conclusion is that the selection of the elbow point (whether automatic or manual) and the initial selection of those centroids greatly affects the final accuracy, making this method difficult to apply in practice without significant calibration.

4.3.1 Dynamic Method

For the Dynamic Method, the parameters are set to $Rem = 0.3$ and $Spl = 1.7$ for every test. These values were selected following iterative testing to determine how much removal and splitting of clusters should be permitted. Too strict a threshold would prevent enough change to increase the number of true positives, while too loose a threshold would actually decrease the accuracy because clusters would not have time to develop due to early removal or splitting. Tables 4.5 and 4.6 show the confusion matrices of the results when Dynamic Clustering is used with KM or EM clustering algorithms. Table 4.5 shows that using the

Dynamic Method with KM Clustering resulted in a slightly higher number of true positives. The small increase can be attributed to the fact that some of the newly created clusters do not have a sufficient chance to develop before they are removed.

Table 4.6 shows that EM clustering also improves slightly when the Dynamic Method is used. Moreover, EM generally performs better than KM, and this is reflected again when the Dynamic Method is used. However, improvement of the Dynamic Method compared to the fixed number of clusters is only applicable for some of the vocations. This suggests that certain methods may perform better for certain vocations.

Table 4.4. Number of Clusters generated by the use of Dynamic Methods.

Clustering	Number of Clusters Selection Method	BT	CT	DT	DV	TB
KM	Dynamic Clustering	7	5	5	8	5
KM	Simulated Annealing	2	3	6	4	4
EM	Dynamic Clustering	5	3	5	5	2
EM	Simulated Annealing	4	3	4	2	4

Table 4.5. Results for Dynamic KM Training with Classical KM Testing.

Vocation	BT	CT	DT	DV	TB
BT	2	0	0	0	0
CT	3(1)	19(2)	3(2)	3(1)	2(1)
DT	1	0(1)	10(2)	4	2(1)
DV	4	1	2	9	0
TB	0	0	0(1)	0	10(1)

Table 4.6. Results for Dynamic EM Training with Classical EM Testing.

Vocation	BT	CT	DT	DV	TB
BT	2	0	0	0	0
CT	1	19(1)	6(1)	1	5
DT	1	0	16	1	1
DV	5	0	2	9	0
TB	0	0	0	0	11

4.3.2 Simulated Annealing

When using Simulated Annealing, the parameters were set as $Rem = 0.2$, $Spl = 1.8$, and $k = 0.025$ for all tests. The selection of these parameters also required some iterative tuning,

As previously mentioned, Simulated Annealing allows splitting and removal by chance, with the probability of these changes getting smaller based on the number of training iterations that have been completed.

When using Simulated Annealing with KM clustering, the results had the same number of true positives as classical KM, as shown in Table 4.7. However, the individual numbers of true positives for each vocation is not the same. For example, when Simulated Annealing was used with KM, only 1 DV vehicle was correctly clustered. The Simulated Annealing KM made up for this decrease through an increase in true positives for the BT and DT vocations. This indicates that Simulated Annealing was not able to find the appropriate number of clusters for the DV vocation.

Table 4.9 compares Dynamic and Simulated Annealing for KM and EM clustering. For EM, both Dynamic and Simulated Annealing show an improvement over the Classical method which suggests that the use of a varying number of clusters has higher impact on EM than KM. The primary contributor to the improvement for EM was the DT vocation. However, for KM the DV vocation reported far fewer true positives than for the classical approach.

Table 4.7. Results for Simulated Annealing KM Training with Classical KM Testing.

Vocation	BT	CT	DT	DV	TB
BT	2	0	0	0	0
CT	1(1)	19(1)	10(2)	1	0
DT	2	0	17	0	0
DV	5	2	8	1	0
TB	0	0	0(1)	0	10(1)

Table 4.8. Results for Simulated Annealing EM Training with Classical EM Testing.

Vocation	BT	CT	DT	DV	TB
BT	2	0	0	0	0
CT	2	18(1)	7(1)	0(1)	4(1)
DT	2	0	17	0	0
DV	3	1	2	10	0
TB	0	0	0	0	11

Table 4.9. Combined Results from Chapter 4.

Clustering	Number of Clusters Selection Method	BT	CT	DT	DV	TB	Total
KM	Classical Approach	1	19(2)	11	7	11	49(2)
	Dynamic Clustering	2	19(2)	10(2)	9	10(1)	50(5)
	Simulated Annealing	2	19(1)	17	1	10(1)	49(2)
EM	Classical Approach	2	19	15(1)	9(1)	11	56(2)
	Dynamic Clustering	2	19(1)	16	9	11	57(1)
	Simulated Annealing	2	18(1)	17	10	11	58(1)

4.3.3 Centroids

While the centroids of a given vocation have similarities across the methods, there are still noticeable differences. Moreover, increasing the number of clusters does not always increase the number of true positives and may not lead to a reduction in the standard deviations of individual features for each cluster either.

As previously described, additional clusters can confound other vocations and lead to a lower number of true positives. For example, Table 4.9 shows that the test with the highest number of true positives was based on EM clustering and Simulated Annealing. This clustering only needed 17 clusters for all vocations, whereas KM clustering with the Dynamic Method required 30 clusters and resulted in a lower number of true positives.

As mentioned above, increasing the number of clusters does not guarantee a reduced range in the feature value. Table 4.10 shows that, even with 7 clusters, BT6 has a max speed standard deviation of 105.99, which is very high. In contrast, Table 4.12 shows (for a test using Simulated Annealing rather than Dynamic Method) that with 3 clusters, the highest standard deviation for the max speed feature was 82.48. This can be due to the presence of outlier vehicles. Ideally, the splitting and removal of clusters should result in a more even distribution of the vehicles across the clusters.

Centroid differences likely account for the small increases and decreases in number of true positives. However, there are still many vehicles that are not clustered correctly in any of the experiments. While this chapter has proven that the number of clusters has a significant impact on the number of true positives, the method for establishing the appropriate number of clusters for each vocation requires additional improvement.

Table 4.10. BT centroids generated with KM clustering using the Dynamic Method.

Features	Metric	BT0	BT1	BT2	BT3	BT4	BT5	BT6
max spd	Mean	62.71	20.15	50.29	47.92	63.9	46.82	60.3
	STD	21.27	0	58.28	62.5	56.85	94.36	105.99
total avg spd	Mean	26.89	1.75	6.37	13.45	13.31	7.94	6.18
	STD	37.15	0	28.66	22.75	44.86	25.42	39.37
drive avg spd	Mean	35.03	8.45	20.94	23.58	33.86	18.65	29.35
	STD	21.66	0	40.36	23.02	65.87	22.73	104.12
drive spd std	Mean	17.34	4.56	14.21	12.61	18.09	11.32	16.1
	STD	9.91	0	20.04	7.92	19.55	15.37	34.84
zero seconds	Mean	0.93	0.01	4.9	2.22	6.2	6.53	13.92
	STD	5.34	0	28.21	13.06	29.33	14.01	27.78
distance total	Mean	23.37	5.13	10.65	17.42	37.81	26.03	30.8
	STD	69.06	0	48.11	87.74	207.55	129.78	224.6
total stops	Mean	10.95	2	14.97	33	22.43	62.07	24.52
	STD	38.23	0	63.91	151	173.54	255.87	114.32
max accel	Mean	4.6	2.79	4.02	8.7	4.91	9.77	5.36
	STD	6.27	0	8.14	20.5	17.32	48.48	11.41
avg accel	Mean	0.82	0.26	0.85	1.56	0.84	1.29	0.99
	STD	0.92	0	1.99	1.88	2.29	1.61	2.62

Table 4.11. CT centroids generated with KM clustering using the Dynamic Method.

Features	Metric	CT0	CT1	CT2	CT3	CT4
max spd	Mean	68.86	70.1	68.3	56.26	71.11
	STD	7.33	15.97	5.5	7.44	20.43
total avg spd	Mean	33.9	36.97	28.52	26.53	43.04
	STD	13.13	21.31	23.59	53.01	36.48
drive avg spd	Mean	41.02	42.61	36.66	35.48	48.42
	STD	12.65	17.91	22.84	51.11	34.89
drive spd std	Mean	20.66	18.61	22.51	18.6	19.31
	STD	9.77	13.91	7.07	8.09	25.47
zero seconds	Mean	1.65	2.46	2.17	3.28	1.57
	STD	5.02	10.81	7.55	10.15	7.79
distance total	Mean	87.37	189.65	75.92	103.03	168.75
	STD	207.2	203.18	156.25	353.94	437.08
total stops	Mean	24.71	34.41	36.33	49.07	22.56
	STD	60.37	78.36	133.2	136.11	94.44
max accel	Mean	3.91	4.07	4.07	4.95	3.89
	STD	2.78	5.34	5.73	7.41	6.06
avg accel	Mean	0.59	0.56	0.64	0.64	0.48
	STD	0.37	0.81	0.7	1.68	1

Table 4.12. BT centroids generated with KM clustering using Simulated Annealing.

Features	Metric	BT0	BT1	BT2
max spd	Mean	49.19	63.37	70.58
	STD	82.48	63.26	18.71
total avg spd	Mean	7.77	15.84	39.75
	STD	46.13	118.63	46.07
drive avg spd	Mean	20.72	34.09	45.31
	STD	38.13	63.4	42.91
drive spd std	Mean	13.04	17.71	19
	STD	21.21	22.05	19.42
zero seconds	Mean	5.84	6.36	2.03
	STD	45.03	70.67	11.42
distance total	Mean	15.97	33.22	177.05
	STD	118.66	221.91	363.19
total stops	Mean	30.87	18.92	28.65
	STD	268.48	161.95	110.32
max accel	Mean	6.47	4.89	3.99
	STD	41.1	15.83	5.94
avg accel	Mean	1.13	0.85	0.52
	STD	3.79	2.53	1.01

Table 4.13. CT centroids generated with KM clustering using Simulated Annealing.

Features	Metric	CT0	CT1
max spd	Mean	56.26	70.58
	STD	5.76	26.46
total avg spd	Mean	26.53	39.75
	STD	41.06	65.15
drive avg spd	Mean	35.48	45.31
	STD	39.59	60.68
drive spd std	Mean	18.6	19
	STD	6.27	27.47
zero seconds	Mean	3.28	2.03
	STD	7.86	16.16
distance total	Mean	103.03	177.05
	STD	274.16	513.63
total stops	Mean	49.07	28.65
	STD	105.43	156.01
max accel	Mean	4.95	3.99
	STD	5.74	8.4
avg accel	Mean	0.64	0.52
	STD	1.3	1.43

Table 4.14. BT centroids generated with EM clustering using the Dynamic Method.

Features	Metric	BT0	BT1	BT2	BT3	BT4
max spd	Mean	54.13	48.88	47.38	62.12	66.53
	STD	4.47	9.84	4.39	4.03	2.5
total avg spd	Mean	8.6	11.33	3.97	20.85	10.49
	STD	2.95	3.5	1.59	8.58	4.07
drive avg spd	Mean	23.96	21.84	18.21	35.39	34.31
	STD	2.91	3.7	1.84	3.23	4.67
drive spd std	Mean	15.84	12.34	12.5	17.35	18.25
	STD	1.59	1.86	1.33	1.75	1.1
zero seconds	Mean	4.05	4.58	8.56	3.76	10.33
	STD	2.09	3.04	4.43	4.02	4.11
distance total	Mean	13.54	24.07	10.4	30.49	40.29
	STD	5.63	12.83	5.63	15.35	14.16
total stops	Mean	14.85	50.12	19.95	13.01	27.24
	STD	6.58	25.83	8.22	6.58	12.44
max accel	Mean	4	9.67	4.63	4.44	5.55
	STD	0.72	3.84	1.1	0.85	0.35
avg accel	Mean	0.84	1.43	0.97	0.77	0.98
	STD	0.23	0.23	0.21	0.15	0.12

Table 4.15. CT centroids generated with EM clustering using the Dynamic Method.

Features	Metric	CT0	CT1	CT2
max spd	Mean	63.32	70.79	68.39
	STD	6.36	1.41	0.59
total avg spd	Mean	30.32	40.27	30.68
	STD	6.35	3.66	2.79
drive avg spd	Mean	38.5	45.53	38.48
	STD	5.63	3.69	2.77
drive spd std	Mean	19.28	18.99	22.06
	STD	1.69	1.59	1.17
zero seconds	Mean	3.16	1.8	1.79
	STD	1.32	0.58	0.59
distance total	Mean	130.32	173.35	73.42
	STD	55.24	31.77	15.97
total stops	Mean	44.12	27.01	28.81
	STD	17.22	7.95	8.06
max accel	Mean	4.72	3.84	3.98
	STD	0.75	0.32	0.54
avg accel	Mean	0.63	0.51	0.62
	STD	0.15	0.06	0.06

Table 4.16. BT centroids generated with EM clustering using Simulated Annealing.

Features	Metric	BT0	BT1	BT2	BT3
max spd	Mean	50.05	49.03	61.87	64.44
	STD	5.24	9.75	4.26	4.45
total avg spd	Mean	5.31	11.15	22.8	11.13
	STD	2.23	3.64	8.17	4.37
drive avg spd	Mean	20.39	21.84	33.8	34.44
	STD	3.39	3.88	4.4	4.72
drive spd std	Mean	13.7	12.5	17.65	17.77
	STD	1.77	2.12	1.51	1.57
zero seconds	Mean	6.88	4.68	1.66	9.6
	STD	4.27	2.93	1.46	4.14
distance total	Mean	10.86	24.69	22.75	39.85
	STD	5.63	14.61	10.15	14.46
total stops	Mean	17.15	49.2	11.42	23
	STD	7.8	25.81	6.58	8.48
max accel	Mean	4.27	9.32	4.35	5.15
	STD	0.89	3.81	0.93	0.74
avg accel	Mean	0.91	1.4	0.79	0.9
	STD	0.22	0.25	0.13	0.19

Table 4.17. CT centroids generated with EM clustering using Simulated Annealing.

Features	Metric	CT0	CT1	CT2
max spd	Mean	70.79	66.05	68.76
	STD	1.4	5.16	0.59
total avg spd	Mean	40.55	30.65	29.59
	STD	3.45	4.88	0.49
drive avg spd	Mean	45.76	38.5	42.46
	STD	3.54	4.38	0.46
drive spd std	Mean	18.96	20.61	23.05
	STD	1.58	2	0.21
zero seconds	Mean	1.78	2.45	2.95
	STD	0.57	1.22	0.18
distance total	Mean	174.33	103.76	79.98
	STD	30.99	50.31	5.63
total stops	Mean	26.58	36.53	24
	STD	7.73	15.19	6.58
max accel	Mean	3.85	4.31	4.55
	STD	0.31	0.75	0.23
avg accel	Mean	0.51	0.62	0.67
	STD	0.06	0.12	0.02

5. CONCLUSION

This thesis introduces a methodology for vocation identification of heavy duty vehicles when the number of vocations is expected to be large and the number of records available for each unknown vehicle is small. The methodology consists of two phases. In the first phase, the profile of the vocation is developed using a set of training vehicles. This profile consists of a set of centroids that represent the operating modes of the vocation and are developed using a clustering technique such as K-Means (KM) or Expectation Maximization (EM). In the second phase, the unknown vehicle is assigned to a vocation using a tournament bracket. In each round, two vocations are compared to the unknown vehicle and the unlikely vocation is eliminated. This assignment was compared to both the one-versus-all assignment and the round-robin assignment. Moreover, two models were considered. The first model was based on 15 features. Some of these features included complex variables such as Average Kinetic Power Density Demand and Cumulative Instantaneous Kinetic Energy Density which may not be accessible to the parts' manufacturer for all the vehicles. The second model is more practical and was limited to 9 features that can be derived solely from speed and distance traveled. Compared to the full feature model, the results show that the reduced feature model had the same or higher number of true positives.

With the exception of the CT vocation, the number of true positives for each vocation using the bracket assignment is also either the same or higher than the corresponding number for the one-versus-all and round-robin assignments. The bracket assignment was introduced in order to avoid some of the drawbacks of the one-versus-all assignment for this application. Indeed, the one-versus-all assignment inherently implies the availability of a large number of records for the unknown vehicles as these records are exposed to all the clusters of all the vocations at once. The bracket assignment overcomes this limitation by comparing two vocations at a time and was shown in this study to have a comparable performance to that of the one-versus-all assignment. The bracket assignment was also compared to a round-robin assignment which can also scale with an increasing number of vocations. The results show that the bracket assignment has a higher number of true positives, but more importantly has lower time complexity than the round-robin assignment.

Several methods for establishing the adequate number of clusters for each vocation were investigated. These methods were compared in conjunction with the classical one-versus-all assignment method. The results show that the Dynamic (set-threshold) method and the Simulated Annealing (variable-threshold) method resulted in an increased number of true positives compared to the baseline assignment with a fixed number of clusters per vocation. The number of clusters in these dynamic methods stayed relatively close to 5, but even these small changes led to an increase in true positives for some of the vocations. The Dynamic Method generally resulted in more clusters for each vocation, compared to Simulated Annealing. The removal of one cluster may not seem important, but to the remaining vocations that removal can reduce confounding with other vocations. A varying number of clusters per vocation has more impact on EM compared to KM for all vocations.

There are several directions that are being considered for future work including exploring the possibility of reducing vocation confounding by applying weights to specific features. In addition, the proposed vocation identification algorithm relies on features aggregated daily from the duty cycle of the vehicle over a period of 13 days. Using data points collected over shorter sample periods (e.g., every 20 miles) will enhance the applicability of the algorithm to a wide range of vehicles. Another direction for future work is to use the bracket assignment method with the dynamic selection of the number of clusters.

REFERENCES

[1] K. Wong, "A short survey on data clustering algorithms," in *2015 Second International Conference on Soft Computing and Machine Intelligence (ISCMI)*, 2015, pp. 64–68.

[2] R. Xu and D. Wunsch, "Survey of clustering algorithms," *IEEE Transactions on Neural Networks*, vol. 16, no. 3, pp. 645–678, 2005.

[3] A. Chakraborty, N. Faujdar, A. Punhani, and S. Saraswat, "Comparative study of k-means clustering using iris data set for various distances.," *2020 10th International Conference on Cloud Computing, Data Science & Engineering (Confluence)*, pp. 332–335, 2020, ISSN: 978-1-7281-2790-3.

[4] Y. Shin, Y. Goh, C. Lee, and J.-M. Chung, "Effective data structure for smart big data systems applying an expectation-maximization algorithm.," *2019 Third World Conference on Smart Trends in Systems Security and Sustainablity (WorldS4)*, pp. 136–140, 2019, ISSN: 978-1-7281-3780-3.

[5] J. Kennedy and R. Eberhart, "Particle swarm optimization," in *Proceedings of ICNN'95-International Conference on Neural Networks*, IEEE, vol. 4, 1995, pp. 1942–1948.

[6] M. Ester, H.-P. Kriegel, J. Sander, X. Xu, *et al.*, "A density-based algorithm for discovering clusters in large spatial databases with noise.," in *Kdd*, vol. 96, 1996, pp. 226–231.

[7] L. McInnes and J. Healy, "Accelerated hierarchical density based clustering," in *2017 IEEE International Conference on Data Mining Workshops (ICDMW)*, IEEE, 2017, pp. 33–42.

[8] L. Chen, F. Ye, Y. Ruan, H. Fan, and Q. Chen, "An algorithm for highway vehicle detection based on convolutional neural network.," *EURASIP Journal on Image & Video Processing*, vol. 2018, no. 1, p. 1, 2018, ISSN: 16875176.

[9] S. Pumrin and D. Dailey, "Vehicle image classification via expectation-maximization algorithm.," *2003 IEEE International Symposium on Circuits and Systems (ISCAS), Circuits and Systems (ISCAS), 2003 IEEE International Symposium on, Circuits and systems*, vol. 2, 2003, ISSN: 0-7803-7761-3.

[10] M. Athimethphat and B. Lerteerawong, "Binary classification tree for multiclass classification with observation-based clustering.," *2012 9th International Conference on Electrical Engineering/Electronics, Computer, Telecommunications and Information Technology, Electrical Engineering/Electronics, Computer, Telecommunications and Information Technology (ECTI-CON)*, pp. 1–4, 2012, ISSN: 978-1-4673-2026-9.

[11] B. Scholkopf and A. J. Smola, *Learning with kernels: support vector machines, regularization, optimization, and beyond.* MIT press, 2001.

[12] S. Daengduang and P. Vateekul, “Applying one-versus-one svms to classify multi-label data with large labels using spark.,” *2017 9th International Conference on Knowledge and Smart Technology (KST), Knowledge and Smart Technology (KST)*, pp. 72–77, 2017, ISSN: 978-1-4673-9077-4.

[13] D. Marutho, S. Hendra Handaka, E. Wijaya, and Muljono, “The determination of cluster number at k-mean using elbow method and purity evaluation on headline news.,” *2018 International Seminar on Application for Technology of Information and Communication, Application for Technology of Information and Communication (iSemantic)*, pp. 533–538, 2018, ISSN: 978-1-5386-7486-4. [Online]. Available: `https://www.ulib.iupui.edu/cgi-bin/proxy.pl?url=https://search-ebscohost-com.proxy.ulib.uits.iu.edu/login.aspx?direct=true&db=edseee&AN=edseee.8549751&site=eds-live` (visited on 10/25/2020).

[14] P. J. Rousseeuw, “Silhouettes: A graphical aid to the interpretation and validation of cluster analysis,” *Journal of computational and applied mathematics*, vol. 20, pp. 53–65, 1987.

[15] *Fleet dna project data*, National Renewable Energy Laboratory, 2019. [Online]. Available: `www.nrel.gov/fleetdna` (visited on 11/11/2020).

[16] A. Duran, C. Phillips, J. Perr-Sauer, K. Kelly, and A. Konan, “Leveraging big data analysis techniques for us vocational vehicle drive cycle characterization, segmentation, and development,” SAE Technical Paper, Tech. Rep., 2018.

[17] L. Breiman, “Random forests,” *Machine learning*, vol. 45, no. 1, pp. 5–32, 2001.

[18] O. Sagi and l. Rokach Lior1, “Ensemble learning: A survey.,” *WIREs: Data Mining & Knowledge Discovery*, vol. 8, no. 4, p. 1, 2018, ISSN: 19424787.

[19] M. Patrick M. and P. Michael J., “Id2-of-3: Constructive induction of m-of-n concepts for discriminators in decision trees.,” *Machine Learning: Proceedings of the Eighth International Workshop (ML91)*, 1991.

[20] Y. Kanemaru, S. Matsuura, M. Kakiuchi, S. Noguchi, A. Inomata, and K. Fujikawa, “Vehicle clustering algorithm for sharing information on traffic congestion,” in *2013 13th International Conference on ITS Telecommunications (ITST)*, IEEE, 2013, pp. 38–43.

[21] R. Santos, R. Edwards, and A. Edwards, “Cluster-based location routing algorithm for vehicle to vehicle communication,” in *Proceedings. 2004 IEEE Radio and Wireless Conference (IEEE Cat. No. 04TH8746)*, IEEE, 2004, pp. 39–42.

[22] J. Wang, Y. Yuan, T. Ni, Y. Ma, M. Liu, G. Xu, and W. Shen, “Anomalous trajectory detection and classification based on difference and intersection set distance,” *IEEE Transactions on Vehicular Technology*, vol. 69, no. 3, pp. 2487–2500, 2020.

[23] X. Cui, T. E. Potok, and P. Palathingal, “Document clustering using particle swarm optimization,” in *Proceedings 2005 IEEE Swarm Intelligence Symposium, 2005. SIS 2005.*, IEEE, 2005, pp. 185–191.

[24] G. Wahba, “Soft and hard classification by reproducing kernel hilbert space methods,” *Proceedings of the National Academy of Sciences*, vol. 99, no. 26, pp. 16 524–16 530, 2002.

[25] T. M. Kodinariya and P. R. Makwana, “Review on determining number of cluster in k-means clustering,” *International Journal*, vol. 1, no. 6, pp. 90–95, 2013.

[26] I. Saha and A. Mukhopadhyay, “Genetic algorithm and simulated annealing based approaches to categorical data clustering.,” *International MultiConference of Engineers & Computer Scientists 2008*, pp. 534–539, 2008. [Online]. Available: `https://www.ulib.iupui.edu/cgi-bin/proxy.pl?url=https://search-ebscohost-com.proxy.ulib.uits.iu.edu/login.aspx?direct=true&db=aci&AN=41020369&site=eds-live` (visited on 10/27/2020).

[27] S. Khalid, T. Khalil, and S. Nasreen, “A survey of feature selection and feature extraction techniques in machine learning,” in *2014 Science and Information Conference*, IEEE, 2014, pp. 372–378.