Detecting Anomalous Trajectories of Vehicles For Support Traffic Transportation Services

Ing. Mazen Ismael

Department of Information Systems

Supervised by: Prof Tomaš Hruška

Doc. Ing. Jaroslav Zendulka

Abstract

Detecting anomalous behaviours in transportation has many benefits, some of these studies oriented to the security of people, safety of people and vehicles, or even to the traffic services, and support of traffic transportation systems. In previous studies, anomalous behaviour detection frameworks have used a combination of many techniques and algorithms. Although these frameworks have had interesting results, additional improvements of the performance are still necessary to obtain efficient results. In this paper, detecting anomalous trajectories frameworks was reviewed and analyzed. In actuality, this study focused on some of the methods of these frameworks which would be the basis of future research to develop an unsupervised transportation framework based on the detection of trajectories anomalous behaviour. This framework should be able to detect anomalous trajectories in order to predict changes in the roads structure and the suggestion of additional roads.

Keywords: clustering, data mining, transportation, detecting anomalous trajectories, outliers, HMM, scene partitioning, detecting roads changes, updates maps

1. Table of content

1.0 Introduction:

The recent advanced studies of the traffic trajectories data were performed for the purpose of increasing the safety and security of people and vehicles, and for improving the traffic services. The continued development in surveillance and GPS systems has added to the validity of these studies. Traffic safety is one of the main health problems in cities; reducing the injuries caused by traffic collisions has become a primary challenge for researchers. Improving security on the roads has led to increase proliferation of cameras. The use of recorded trajectories' data collected from GPS devices and identified from telephone signals and wireless networks has aided in this, as well. Moreover, current studies have focused on improving traffic quality and reducing the number of traffic collisions. Data Mining algorithms have been used for more robust clustering, as well as being utilized in the frameworks for detecting outlier trajectories. Many frameworks have been designed for detecting vehicles' abnormal behaviour, and many of these frameworks have positive results. However, they still need development to increase the accuracy of the results. This paper focuses on these studies and reviews the current state of the art frameworks designed to detect outliers of the transportation's trajectories. This paper contains 8 main Sections, organized as follows:
Section 1 - Introduction, Basic Definitions and Preprocessing Steps; **Section 2** - Study Current State of the Art in the Area of Detecting Abnormal Behaviour of Traffic Trajectories and Transportation Services; **Section 3 -** Results and Discussion; **Section 4 -** Conclusion and Future Work; **Section 5 –** Current Results; **Section 6 –** References; **Section 7 -** Appendices.

1.1. Preprocessing trajectories

Nowadays, many different technologies for detecting trajectories are used, GPS devices, Wi-Fi, Bluetooth, surveillance systems (static cameras or cameras carried by quadcopter) and sensors to name but a few. Although these intelligent devices can produce good, accurate measurements, we still suffer from errors in measuring which affect the drawing's quality. Moreover, due to the huge amount of data and the need to retain performance, saving the trajectories without using many compressions for most of the recording time could be impossible. Because of this, the importance of preprocessing actions rose, which contain: Generating and reduce storage, reduce errors and filtering.
It is useful here to clarify that the outlier in our study is different from the error in measuring (generating trajectories and reducing storage steps like the noise).

1.1.1.*Generating and reduce storage*

The process of generating and reducing data of trajectories oriented by the limitation of the errors. For that, most of the techniques use measuring error action.

1.1.1.1. Batched Compression Techniques (off-line)

Techniques of the data reduction type can run in a batch mode after the data are collected, or in an on-line mode as the data are collected. Popular algorithms:

i. ***Uniform sampling*** algorithm may keep the every i-th location points (not sensitive to the occurrence of the errors) [1].
ii. ***Douglas-Peucker (DP)*** recursively partitions splits the original problem into two sub-problems using split point, the time complexity $O(N^2)$ improved to $O(NlogN)$ [2][3]
iii. ***Top-down time-ratio (TD-TR)*** decomposes the trajectory [4]approximation problem in a top-down fashion [5](consideration both geometric and temporal Properties).

iv. ***Bellman*** applies the dynamic programming technique to guarantee split points are the best choices, the time complexity $O(N^3)$ improved to $O(N^2)$ [6].

Bellman's algorithm is very suitable for off-line uploading and analysis of trajectories. It can be used to reduce the transmission and storage overheads of data, if the time not priority (complexity $O(N^2)$).

1.1.1.2. On-Line Data Reduction Techniques

The major requirement for on-line processing algorithms is to be able to do effective decision in the on-line state. Popular algorithms:

i. ***Reservoir sampling*** maintains a reservoir of size R which are used to generate an approximated trajectory of size R. The time complexity $O(R(1+logN/R))$ [7].
ii. ***Sliding window*** fits the location points in a growing sliding window with a valid line segment and continue to grow the sliding window until the approximation error exceeds some error bound [4][5].
iii. ***Before Open Window (BOPW):*** the location points included in the final approximate trajectory are located before those that result in an excessive error [5].
iv. ***Normal Opening Window (NOPW)*** chooses location points with the highest error within their sliding window as the *closing point* of the approximating line segment as well as the new anchor point [5].

1.1.1.3. Trajectory Data Reduction Based on Speed and Direction

The changes in speed and direction for predicting the location of approximated trajectories considered as the key for these algorithms. Popular algorithms include:

i. ***Threshold-guided sampling*** **algorithm** reduces redundant data points in trajectories based on speed and direction changing, by using a safe area derived from the last two locations and a given thresholds to efficiently determine whether a newly location point contains important information [8].
ii. ***STTrace*** The idea is to insert data points into the sample memory of known and constant size, based on the movement features (e.g., speed and direction) [5][8].

1.1.2.*Filtering*

Accuracy remains the main focus for all the processes, errors have arisen in the generation steps, but it is not the only problem. Sensor noise, wireless signal, smoke in front the camera and other factors affect the accuracy. Filtering techniques smooth the noise, and potentially decrease the error in the measurements some of the popular filters include:

I. Mean and Median Filters:

The mean filter called "causal" filter, because it only depends on values in the past to compute the estimate, Median filter like the mean filter but is less affected by outliers, it estimates what is directly measured.

II. Kalman Filter:

Most widely used variant of Bayes filters [9], contain Measurement Model and Dynamic Model, Estimate directly measured, speed and acceleration by estimates for the state vector.

III. Particle Filter:

Use a measurement model and a dynamic model too, more general than Kalman filter, but less efficient.
A potential disadvantage of the particle filter is computation time, which is affected by the number of particles. However, more particles, generally give a better result [9][10].

2. State Of the Art

The quick advance in geolocation domain and observation systems, offer us a huge amount of moving objects data, which could be a potential source of important information for Safety, Security and Support for the Traffic Services. Mining this data to execute important patterns was a reason for stimulating the researchers to improve effective methods. Nowadays, studies in the domain of detecting the anomalous behavior of traffic trajectories are becoming widely common and the variety of these studies are a result of the many traffic problems. Most of these studies have built their own frameworks based on Hidden Markov Model (HMM), whilst other methods used graphs or built another creative framework.

2.1. Frameworks used Hidden Markov Model (HMM) Algorithms for Detecting Trajectories Anomalous Behaviour

2.1.1. Main Flow Direction (MFD) and Hidden Markov Model (HMM) for detect abnormal trajectories

Yingfeng Cai1 *et al* [11], present an efficient framework for detection anomalous behaviour. Such framework includes the trajectory pattern learning module and the online abnormal detection module. In the pattern learning module, a coarse-to-fine clustering strategy has utilized. Vehicle trajectories are coarsely grouped into coherent clusters according to the main flow direction (MFD) vectors followed by a three-stage filtering algorithm. Then, a robust K-means clustering algorithm is used in each coarse cluster to get fine classification by which the outliers are distinguished. Finally, the hidden Markov model (HMM) is used to establish the path pattern within each cluster. In the online detection module, the new vehicle trajectory is compared against all the MFD distributions and the HMMs so that the coherence with common motion patterns can be evaluated. Besides that, a real-time abnormal detection method is proposed. The abnormal behaviour should be detected when happening. Experimental results illustrate that the detection rate of the proposed algorithm is close to the state-of-the-art abnormal event detection systems. In addition, the proposed system provides the lowest false detection rate among selected methods. It is suitable for intelligent surveillance applications.

Yingfeng Cai1 *et al*. [11] proposed a study using the coarse-to-fine method to learn the trajectory patterns, and then develop a trajectory-based anomalous behaviour detection framework for intelligent traffic surveillance.
This framework includes the trajectory pattern learning part and the online abnormal detection part.
The first part, trajectory pattern learning: Main Flow Direction (MFD) vector is used to find a coarse number of trajectory groups and follow with a three-stage filtering algorithm. The robust K-means algorithm is applied to finish fine classification which can automatically distinguish outliers inside each group. Followed by using HMM to create the route pattern for each cluster. In the second part abnormal detection: each new trajectory is compared against all the HMMs and the anomaly is determined if the maximum of probability is lower than the abnormality threshold obtained in the learning part. The architecture described in *Figure **1***.

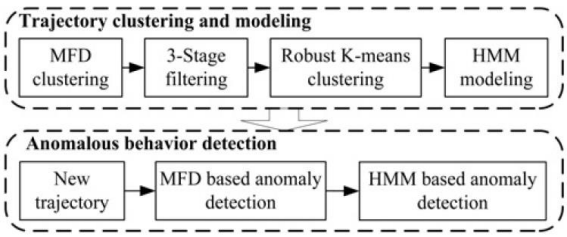

Figure 1. Architecture of the proposed framework

TRAJECTORY CLUSTERING

1. Coarse classification of trajectories

   There is a particular association between the starting and ending points in the same class of trajectories, and MFD defined a vector of a certain trajectory.

   Similar feature vectors are generated in the 2D space from a coherent trajectories cluster. And each cluster is described by a Gaussian function with the mean μk and the covariance matrix σk. All MFD vectors are considered by the overall distribution, that can be p(x) modelled as a Mixture of Gaussian (MoG).

$$p(x) = \sum_{k=1}^{k_{max}} \omega_k p_k(x) \quad (1)$$

Where k_{max} is the number of classes, ω_k is the prior probability and $p_k(x)$ is the normal distribution associated with the *k*th class.

2. Filtering of classification

   The accuracy of MFD clustering improved by three-stage filtering method:

   a) Remove clusters that are much wider than the other clusters. This wrong cluster, caused by the outliers is shown in *Figure 2*b.
   b) Merge clusters that have obvious overlaps (cluster 5 and 7 shown in *Figure 2*b)
   c) Remove the isolated outliers. Such error is generated by incorrect tracking because of the occlusion or those formed when overlap splits. In these cases, the maximum and minimum peak coefficient-based adaptive mean and variance estimation methods can be used [17].

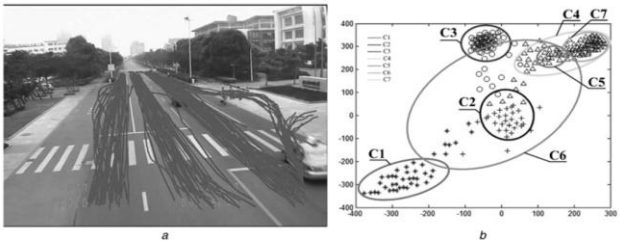

Figure 2. Architecture of the proposed framework. ***a*** *Detected trajectories in a certain time of a certain scene.* ***b*** *Corresponding clustering result of the MFD vectors.*

The result of the filtering of *Figure 2*b, can be seen clearly in *Figure 3a*. And the corresponding MFD distribution is shown in *Figure 3b* where the blue ellipses represent as ending areas and the pink ellipses are the beginning areas of trajectories. *Figure 3* indicates that the overlapping classes were efficiently merged and the large cluster was dissolved correctly.

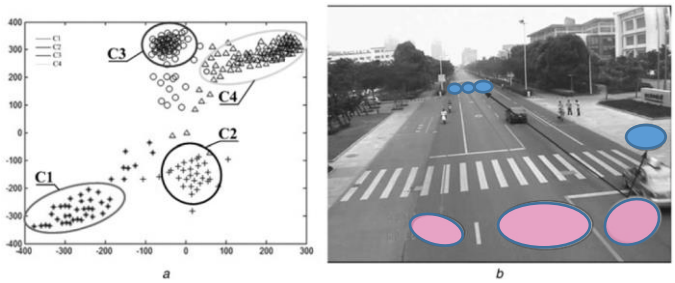

Figure 3. Improved clustering result of MFD vectors. a Improved clustering result of the MFD vectors b Corresponding MFD distribution of trajectories

3. Fine classification of trajectories

The previous two sections dealt with trajectories as a set of "coarse clusters" each cluster has similar direction features. So it is necessary to make more clustering to the trajectories to ensure that these trajectories have similar routes. A robust K-means clustering algorithm is used to improve classification of the trajectories in each coarse cluster. The algorithm clusters the data and will identify the outliers Yingfeng Cai1 et al. [11].

4. Since the behaviours of objects in traveling along the given path is requested. And the location of objects in trajectories certain by robust K-means. HMMs came to encode the spatiotemporal properties of every route, and identify the dynamics and locations of objects. [17].

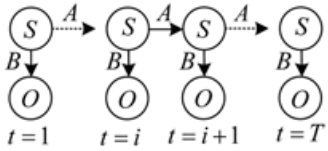

Figure 4. Basic structure of an HMM

There is a relation between the number of HMMs and the number of clusters.
An HMM is trained with all the trajectories from one cluster. The basic structure [18] of the HMM is shown in *Figure 4*.

5. Performance analysis and anomaly detection:
   a) MFD-based anomaly detection:
   MFD clustering can fetch anomalies like wrong turn and illegal entering.
   The new trajectory is labelled as an abnormal event if the distance between MFD vector x_{new}, is always two times greater than the determinant of the cluster's covariance matrices to any cluster center.
   b) b) HMM-based anomaly detection:

   Given a new trajectory:

$$T_{new} = \{(x_1, y_1, \delta x_1, \delta y_1), \ldots, (x_t, y_t, \delta x_t, \delta y_t)\} \quad (2)$$

   And the observed sequence is: O= [o_i], i = 1. . . m, $o_i=(x_i, y_i, \delta x_i, \delta y_i)$ where (x, y) is the position vector and (δx, δy) is the velocity vector.

- For all the HMMs $\lambda = \{\lambda^1, \lambda^2, \lambda^3, \dots, \lambda^K\}$, we calculate the probability, under each λ^k, k=1,…,K

Of T_{new} by following equation:

$$P(O|\lambda^k) = \sum_{i}^{N} \alpha_m(i) \quad (3)$$

The path model best explains the given trajectory T_{new} is assigned as follows:

$$\lambda^* = argmax_k P(O|\lambda_k) \quad (4)$$

Hence, every track will be classified into a path. If the probability λ^* is less than a threshold, the trajectory is defined as abnormal.

Finding the abnormal behaviours from offline trajectories is important, and detecting them when happening is important too. The HMM method can carry out real-time abnormal detection.

The coarse-to-fine clustering algorithm used to generate common routes, and MFD vectors of trajectories are modelled as a Mixture of Gaussian (MoG). The number of Gaussians in the mixture match to the number of coarse clusters. Then a three-stage filtering method used to improve the fineness of MFD clustering.

A robust K-means clustering algorithm used in each coarse cluster to get fine classification by Isolate the outliers.

After that, Hidden Markov Model (HMMs) encoded the spatiotemporal properties of each cluster. Depending on that, by the coherence with common motion patterns in online detection part, can distinguish abnormal behaviours.

- **Strengths and weaknesses**

  - ✓ The experimental results show that the approach can effectively detect the abnormal events in the outdoor and indoor environments.
  - ✓ Framework performed well in detecting a range of anomalous events under unsupervised working conditions.
  - ✓ This framework is considered as an unsupervised framework
  - ✗ In learning part choosing Partitioning Around Medoids (PAM) algorithm instead of K-means (in the case of small data set), will be less effective of an outlier, and in this case, there is no need to specify the number of clusters in the data a priori.

2.1.2. Using HMM with K-means in purpose of detecting anomalous trajectories

Saunier et al [23] has used the most common HMM. The framework built based on k-means approach with **HMM** and a simple heuristic to find the number of clusters automatically. The target was to develop a method for automated road safety analysis. The traffic conflicts could be detected by identifying pairs of models of conflicting trajectories.

The presented framework able to: Interaction detect if two vehicles are close enough (threshold on their distance) and nearing each other (their distance decreases). Each interacting vehicle trajectory is

assigned to a HMM. If the HMMs of both interaction trajectories were memorized as conflicting (the pairs of models), a traffic conflict between these two vehicles will be detected.

- **Strengths and weaknesses**

  - ✓ Although it could not consider so new method (2006), but there are some good ideas to improve the actions.
  - × The method can be refined by using the labeled data to guide the clustering process and by investigating more complex Dynamic Bayesian Networks (DBNs).
  - × To make the process more robust, other clustering techniques such as mean shift clustering could be used instead of K-means.

2.2. Frameworks for detecting Anomalous Trajectories Did Not used HMM

2.2.1. Zones portioning and by unsupervised learning of graphs

L. Brun *et al* [19], proposed unsupervised framework depending on graphs; As *Figure 5* shows, that during the first step learning phase comes to build the zones from a training set of trajectories, the scene is quantized into a fixed number of zones by exploiting the distribution of trajectories belonging to the training set. Each trajectory is represented at a high level through a graph, the vertices of the graph encode the zones and the edges the probability to move from zone to another one.

The trajectory can be considered as normal, if it corresponds to a path of the graph crossed by a large amount of trajectories belonging to the training set. Otherwise, the trajectory detected as abnormal and an alert will be sent to the tester.

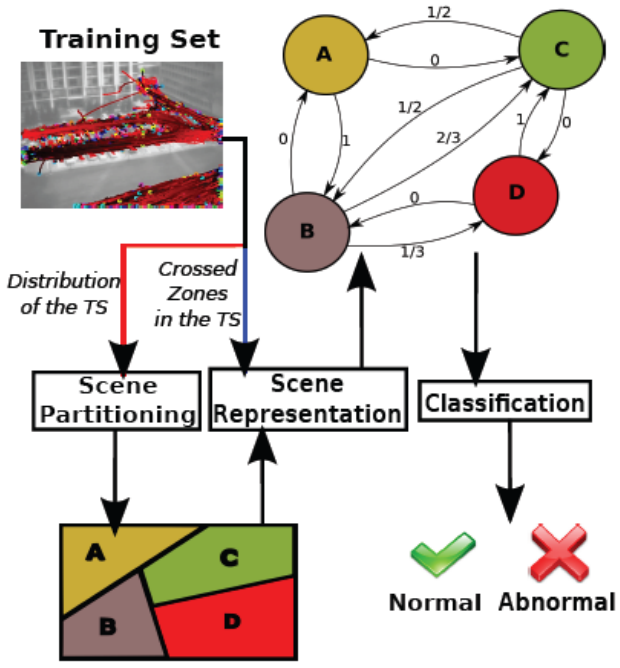

Figure 5. An overview of the framework

2.2.1.1. Method scenes

Scene Partitioning

This approach is strongly dependent on the weight of the graph; during the training phase the extracted trajectories embed in the graph.

Since the mapping of each node into one pixel of the scene is not practical in the reality; so dynamically partition the scene has been proposed by evaluating the distribution depending on this methodology which L. Brun *et al* presented in [20].

The algorithm selects one zone starting from the entire scene, and hierarchically partitions it into two zones by the density of trajectories crossing it.

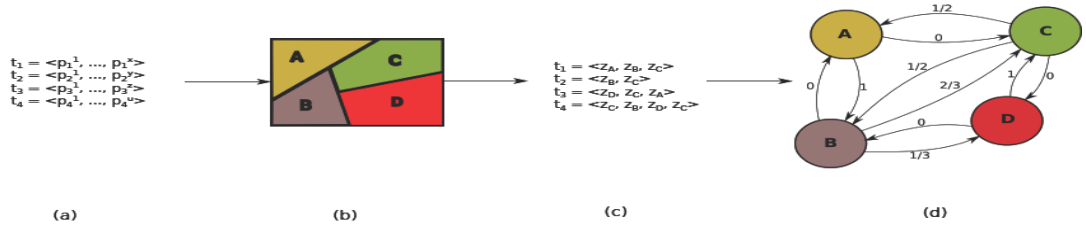

Figure 6. The training set in (a) raises the partitioning of the scene depicted in (b) and then the oriented-weighted graph in (d), whose weights have been obtained according to Equation 1.

Scene representing

After partitioning, to embed the trajectories, the scene represents as weighted and oriented graph. In this graph, $G = (V, E)$ each vertex belonging to $V = \{z_1, .., z_L\}$ is associated with a zone and each edge e_{ij} belonging to E identifies the adjacency between two zones z_i and z_j.

Creating e_{ij} is depending on the weight $w(e_{ij})$ which is the probability of trajectories are from training set from zone z_i to zone z_j.

$$w(e_{ij}) = \frac{g(i,j)}{\sum_{l=1}^{L} g(i,l)} \quad (5)$$

Figure 6 clarify the process: in the data set, we have four trajectories, {t1, t2, t3, t4} (a). By the previous algorithm, the scene partitioned to 4 zones L = 4 (b); the trajectories can then be seen as the sequence of zones crossed during their life (c). Finally, the graph based representation of the scene is obtained: for instance, all the trajectories exiting from the zone z_A go to z_B while no trajectory passes from z_A to z_C, then $w(e_{AB}) = 1$ and $w(e_{AC}) = 0$.

2.2.1.1.1. Classification

Classification come to identify abnormal trajectories by evaluating how much a new trajectory t belongs to the graph G previously defined.

Where trajectory t present as a sequence of zones symbols $t_j =< z_1, z_2, \dots, z_{|t|} >$; where |t| is the count of zones crossed by t. Then, the approach proposed by two steps:

For each sub-trajectory t_j, we consider $t_j =< z_1, z_2, \dots, z_H >$, where H<|t| with fixed length, so H should be selected before know if the sub-trajectory consider as abnormal.

For each sub-trajectory t_j it's probability given by:

The cross-probability $P(t_j)$ of the trajectory which seen as the path on the graph is evaluated by multiplying the weight of each edge that is on the path.

$$\mathrm{P}(t_j) = \mathrm{w}(e_{z_1;z_2}) * \dots * \mathrm{w}(e_{z_{H-1};z_H}) \tag{6}$$

The thershold used to compare with $P(t_j)$ to evaluate wither t_j is normal or abnormal.

Several strategies have been used for classification, each one especially suited for a particular application domain. Hence, choosing the strategy related to the three features: the time required for the computation, spatial memory required to store the structures used, and the results.

I. Thresholding

Single threshold α should be select

$$P(t_i) > \alpha \quad Normal \;\&\; P(t_i) < \alpha \quad abnormal \tag{7}$$

It is so important to partition the trajectories to sub trajectories in selecting the threshold independent of the trajectories length.

II. A Posteriori Probability Ratio Test

In this strategy, two conditional probability of test types are considered $P(N|x)$ and$P(\bar{N}|x)$, where N denotes that a trajectory seems normal, $\bar{N}$ denotes that a trajectory seem abnormal, and x is the path of the trajectory. Bayesian formulation of the problem has been exploited in this test, and the conditional probability could be written as:

$$P(N|x) = \frac{P(x|N) * P(N)}{P(x)}$$
$$P(\bar{N}|x) = \frac{P(x|\bar{N}) * P(\bar{N})}{P(x)} \tag{8}$$

For each path (x), the next ratio evaluate whether it is normal or abnormal trajectory:

$$\lambda(x) = \frac{P(x|N) * P(N)}{P(x|\bar{N}) * P(\bar{N})} \tag{9}$$

By the condition:

$$Class(x) = \begin{cases} N & if \quad \lambda(x) > \alpha \\ \bar{N} & if \quad \lambda(x) < \alpha \end{cases} \tag{10}$$

And since $\frac{P(N)}{P(\bar{N})}$ is not depend on x, the mass of conditional probability functions depends only on the path of a single sub trajectory of the test set so

$$P(x|N) = P(t_j) \;\&\; P(x|\bar{N}) = P^*(t_j) \tag{11}$$

The two value in equation (11) evaluate, if $\mathrm{P(x|N)}$ bigger than $\mathrm{P(x|\bar{N})}$ the sub-trajectory consider as normal, otherwise consider as abnormal.

Two different topologies used for computing$P^*(t_j)$:

- Vertex degree: do not consider edge's weights, and $P^*(t_j)$ set as the one of a random walker.

$$P^*(t_j) = p(1) * \dots * p(H-1)$$

Where

$$p(y) = \frac{1}{|z_y|} \quad (12)$$

And $|z_y|$ represents the degree of the vertex z_y.

- Inverted Weight

The wieght of edges have been recognized in computing $P^*(t_j)$.

$$P^*(t_j) = \bar{\omega}(e_{z_1;z_2}) * \omega(e_{z_2;z_3}) *..* \omega(e_{z_{H-1};z_H}) \quad (13)$$

Where

$$\bar{\omega}\left(e_{z_y;z_{y+1}}\right) = \frac{1-\omega\left(e_{z_y;z_{y+1}}\right)}{N-1} \quad (14)$$

With $\sum_{z=1}^{L} \bar{\omega}\left(e_{ij}\right) = 1 \; and \; \forall i$

III. Detecting using string kernels

The partitioning to sub-trajectories allowed online detection for abnormal behavior, without waiting for object to exit the scene. Depending on representation in [20] also string considered as a strategy could be used for representation. Then each sub-trajectory represent as a string of zones crossed during its life, and cluster algorithms are used for identifying the prototypes of normal trajectories. After that fast global alignment kernel [21] used to determine the similarity between the trajectories. Moreover, to investigate whether sub-trajectory $t_i$belong to the cluster C_i, in order each cluster should be split into a set of clusters C_{ij} which contains only sub-trajectories visit the zone j. So for each divider cluster C_{ij} we compare sub-trajectory t_i either length H by computing d(C_{ij} ; t_j), which belong to t, and that by calculation in order for determine if the cluster C_i^* closer to t_j . Then the distances d(C_i^*; t_j) are evaluated according to a threshold α in order to verify if t is normal or abnormal.

2.2.1.2. Experiment results

This method has been tested on the MIT Trajectory dataset taken by a tracking algorithm, (hereinafter MIT) [22], contain 40.453 trajectories during five days.

The results affected by influential factors:

ii. Short sub-trajectory length is used, a small number of zones, and use of a small fixed length H is a good choice for the classification.
iii. The results proved that a high number of zones is more effective with the simple threshold, while posteriori probability test works better with the low number of zones.

The training set used to create scene partitions, after that the scene represented graph which used to identify abnormal trajectories by evaluating how much a new trajectory t belongs to the graph.

- **Strengths and weaknesses**

  - ✓ In this method, abnormal behaviors could be detected without waiting if object exited the scene.
  - ✓ The proposed method does not require a time-consuming learning.
  - ✗ Although the method does not consume long learning time, but still need time for scene partitioning.

2.2.2. Grid-Cells *Scene portioning with Isolation Based Anomalous Trajectories (iBAT)*

Zhang *et al* [24], proposed creative detection framework for isolate anomalous trajectories, for discover anomalous driving patterns from taxi's trajectories. The practical implementation of this framework used for develop two applications (Detect Fraud of Taxi Drivers and Road network Change Detection). The framework contains two parts. First extracting trajectories for occupier's taxis, and spilt the scene into equal size's **grid-cells**, and group all trajectories which have same Source and Destination (study noticed only the valid trajectories of occupied taxis), and represent the trajectories as sequence of symbols. Second part, present an Isolation Based Anomalous Trajectory **iBAT** detection method.

The study differentiate between the anomalous instances and the new road, by discover the anomalous trajectories witch are "Few and Different" (easier to separate), for that, "different and many" meaning new road. By this mechanism, the study used Isolation Forest method (iForest) [26]. iBAT applied Isolation in a Lazy Learning Manner, to detect the fraud status which based on using random test sample (trajectory).

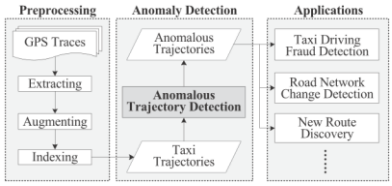

Figure 7. iBAT Overview

In trajectories preprocessing period *Figure 7*, problem of low-sampling-rate has solved by used method to argument the trajectories, this method insert pseudo cells between the far cells which belong to trajectory, another affective ways could be used to get more trajectories by use more historical data.

Isolation Forest method (iForest) used in iBAT via Lazy Learning Manner [25][26], and defined n(t) as the number of cells used for isolating the trajectory, and **Anomaly Score** of trajectory t as a normalization of the average number of used cells:

$$s(t,N) = 2^{-\frac{E(n(t))}{c(N)}} \tag{15}$$

Where $E(n(t))$ is average of cells used for isolating t, N is number of trajectories from witch t has separated, C(N) is the average of n(t) given N: $C(N)= 2H(N - 1) – 2(N – 1)/N$

H(i) is harmonic number that can be estimated as ln(i) +0.5771566 (Euler's constant). So if E(n(t)) →0, s(n) →1, and that is meaning t surely anomalous trajectory, and when E(n(t)) > c(N), s(t , N) < 0.5, and that is meaning t detected as normal trajectory. Even in large number of trajectories, iBAT effectively use Equation15 to reach anomaly scour and repeats the isolation process on different subsamples of all trajectories.

Input: t -test trajectory
T - set of trajectories to be separated from
m -number of running trials.
ψ -sub-sample size
Process:
Let n be a vector of m zeros (n 's are zeros)
For i = 1 to m **do**
randomly sample ψ trajectories from $T \rightarrow T'$
Repeat
$n_i + 1 \rightarrow n_i$
Randomly choose a cell p from t
Select the trajectories that include p from $T' \rightarrow T'$
Until T' is empty
End For
Compute s(t , ψ) according to Equation 15
Output: anomaly score s(t , ψ)

***Algorithm 1**: The iBAT method*

Figure 8. (a) Considered trajectories *(b) Similar trajectories with* $\boldsymbol{t_0}$

iBAT able to naturally incorporate newly-generated trajectories, and detect new anomalous trajectories clusters, and then catch the changes of roads.

- Example how could the new roads detected from anomalous trajectories:
  As the *Figure 8* shows up, two anomalous trajectories t_0, t_1, and one normal trajectory t_i, by detect 160 trajectories and similar to t_0, then based on Equation 15, anomaly score of t_0 change from about 0.9 to 0.3 (as *Figure 9* show), and in the result that means $t_0$became normal trajectory.

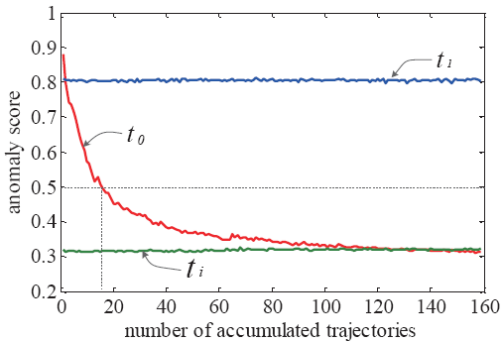

Figure 9. the anomaly score of $\boldsymbol{t_0}$ *decrease rapidly when similar trajectories are accumulating, but that of* $\boldsymbol{t_i}$ *and* $\boldsymbol{t_1}$ *keeps almost the same*

Actually, the score changed quickly after detecting only 18 trajectories from 0.9 to 0.5 and started to look like normal trajectory, and that mean the network changed, and new road open up.

- **Strengths and weaknesses**

  - ✓ The great performance of this method achieved by iBAT (Area Under the Curve AUC>0.99, over 90% detecting rate and the false alarm rate of less than 2%).
  - ✓ iBAT developed later to use online, and called iBOAT [27].
  - ✓ The algorithm counts the trajectories which pass through the source destination points.
  - ✓ No distance or density measures are needed in this effective framework, by applying the isolation mechanism.
  - ✗ Using grid-cell is ideal solution, but it could be not feasible in real contexts.
  - ✗ The framework did not explain how to update maps as result of its work.

2.2.3. Scene Partitioning Method in Online Detection framework of Anomalous Trajectories Based on the Isolation (iBOAT)

Chen *et al* [27], propose framework for online detection of anomalous trajectories, this framework continue after the previous offline framework (iBAT) with many improvements, to be able to work online and detect the abnormal behaviours even for sub-trajectories. Moreover, this framework covered many requirements for accuracy and performance (The area under the curve (AUC)>0.99), and working online, and characterize the anomaly degree, beside of that, iBOAT used time and distance to judge in real-time whether the sub-trajectories are anomalous or not.

Two interested ideas have been used to detect the outlier trajectories, first to select the **few** and **different** trajectories, second to use the **score** threshold for quantifying the degree of anomalousness of each trajectory.
In the other side, partitioning method developed for purpose of improve the performance.
In the following we explain how the method work:

Every trajectory consents from points or even checking points$\langle p_1, p_2 .. p_n \rangle$, where $p_i \in R^2$ so for every trajectory (t); i<j<n, $t_{i \to j}$ meaning subtrajectory $\langle p_i, p_2 .. p_j \rangle$.

The method split the scene to grid cells (matrix *G*), function $\alpha: R^2 \to G$ maps locations to grid cells. The method used **maximizing the grid size** to ensure the accuracy, and chosen an experimentally grid cells sizes (250m x 250m).

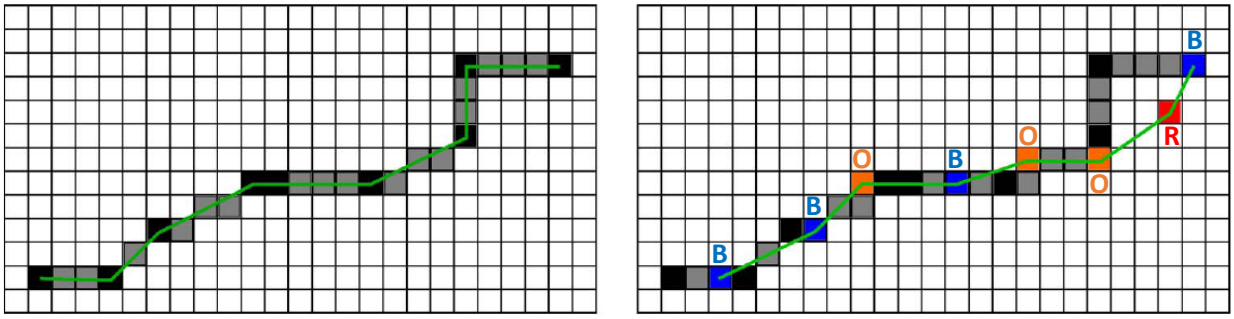

Figure 10. (Left) Example of a trajectory with augmented cells. (Right) Comparing existing trajectory with a new trajectory. B=Blue, R=Red, O=Orange

To deal only with mapped trajectories, mapped qualifier has dropped. But another challenge raised by the low sampling rates (GPS checking time), and the gaps between the mapped cells. Because of that the method has solved the problem in two steps:

First: augmented all trajectories by adding **pseudo** points (cells), and for that reason function $pos: T \times G \to N^+$ defined, where T denote the set of all mapped and augmented trajectories:

$$pos(t,g) = \begin{cases} \arg min_{i \in N^+}\{t_i = g\} & ,if\ g \in t \\ \infty, & Otherwise \end{cases} \quad (16)$$

The function in equation (17) retunes the first index in ***t***, which equal ***g***.

For example, let $t = \langle g_1, g_2, g_3, g_6, g_2, g_9 \rangle$ then $pos(t, g_2) = 2$ and the $pos(t, g_8) = \infty$.

Second: Because of the difference between the sampling of whether two trajectories pass from same path, the two trajectories seem to have different paths. Moreover, as *Figure 10* shows on the right hand side the trajectory with augmented cells, on the other side, the augmented trajectory with the new trajectory, some of the new trajectory share the points (cells) with the augmented trajectory (Blue (*B*)) cells, and some of new trajectory cells fall in empty cells (Orange (*O*) and Red (*R*) cells), the aim is to know if the new trajectory pass the same path, or have a new path.

In this part method consider if the trajectory has the same path. If all its cells are shared with the augmented cells (the Blue cells) or are adjacent to the augmented cell (The Orange cells). Otherwise, the cells are considered as Red and the trajectory have new path. For that the function $N : G \to P(G)$ is used to return the **adjacent neighbors** of a grid cell, so for **cell *g*** and **trajectory *t***, **N(g)** belong to t means at least one of the g neighbors belong to t. For that and using Equation 16, the method can select if there is neighbors to g belong to t, if $\mathrm{pos}(\mathrm{t}, \mathrm{N}(\mathrm{g})) \neq \infty$, and in this case the function return first index in t, that is equal to one of the neighbors of g.

In following, *Figure 11* presents example:

$$t = \langle g_1, g_2, g_3, g_4, g_{11}, g_{12} \rangle;\ \mathrm{pos}\big(\mathrm{t}, \mathrm{N}(g_9)\big) = 2$$

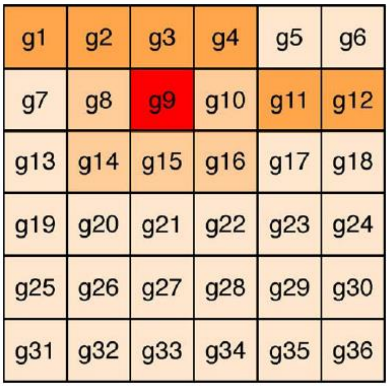

Figure 11. Sample trajectory used to illustrate a cell's neighbors.

- **Strengths and weaknesses**

  - ✓ The great performance of this method (Area Under the Curve AUC>0.99).
  - ✓ The ability to work in online environment.
  - ✓ The improvements in partitioning.
  - ✓ The algorithm used a method for acting with low sampling rates problem, by considers the adjacent neighbors points, as same as the main points.
  - × The algorithm chooses the grid size in manual way.
  - × No clear method used for adding pseudo points.

2.3. Partition the Scene of tested area, using Zone's Partitioning Method

L. Brun *et al* [20] proposed unsupervised learning technique, to extract normal trajectories which partition the scene to crossed zones using distribution of trajectories training set. The main steps in the algorithm used for scene partition summarize in: consider the entire scene as one zone, and then divide the zone to L fixed number of zones and use the distribution of training set. After that using statistical properties (mean, major axis and covariance matrix) for represent each zone. The algorithm use set of planes for repeat splitting the zones (cutting planes), from chosen positions (cutting positions).

For more explaining of Scene Partitioning step in this algorithm: let p be a generic point in the scene, then the function *f(p)* is the number of trajectories passing through p, three statistical values of zone Z_i (cardinality$|Z_i|$, Mean μ_i and Covariance Cov_i):

$$|Z_i| = \sum_{p \in Z_i} f(p) \qquad \mu_i = \frac{\sum_{p \in Z_i} f(p) p}{|Z_i|} \qquad Cov_i = \frac{\sum_{p \in Z_i} f(p) p * p^t}{|Z_i|} - \mu_i * {\mu_i}^t \tag{17}$$

And the property archived by minimizes the Total Squared Error (TSE) for all zones:

$$\mathrm{TSE(P)} = \sum_{i=1}^{L} SE(Z_i)$$

Where $SE(Z_i) = \sum_{p \in Z_i} \|\mu_i - p\|^2$

Some partitioning heuristic used in this framework (*Splitting strategy*) as *Figure 12* shows, three quantization steps recognized: selecting the cluster, selecting the cutting axis, selecting the cutting position.

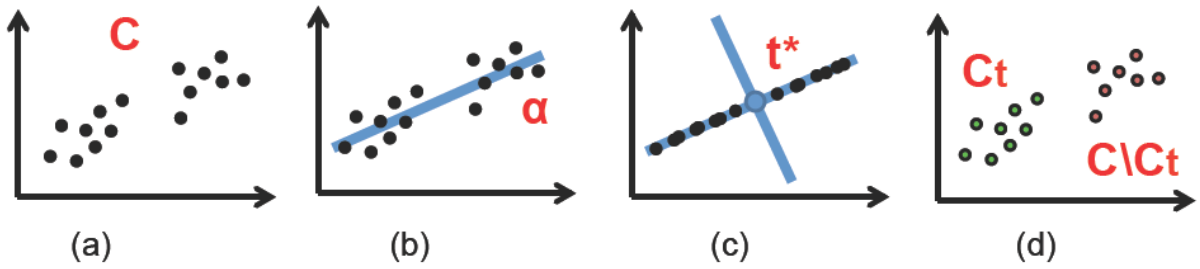

Figure 12. Simple example of the proposed clustering algorithm: once selected the cluster C (a), the cutting axis and the cutting position are computed ((b) and (c) respectively) and the new clusters Ct and (C\Ct) are obtained.

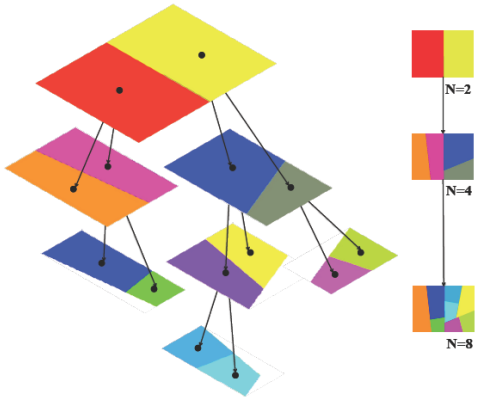

Figure 13. Example of a tree-structured vector quantization obtained recursively partitioning the scene into L = 8 zones.

Strategy as shown in *Figure 13*, generates a tree-structured vector quantization (leafs represent zones). Three steps used in this strategy:

Cluster Selection: Chosen the right cluster for split is important the strategy work. Since the squared error effect on TSE (P) very well, and for achieve suitable compromise between final quantization error and computational time, the strategy minimize the TSE (P) by split the zones based on largest squared error.

Cutting Axis: The strategy aiming to split along the axis with the greatest variance.

Cutting Position: The cutting position t^* chosen in place decreases the total squared error induced by the split:

$$SE(Z_i) = [SE(Z_i^1) + SE(Z_i^2)] \quad (18)$$

Where Z_i^1 and Z_i^2 are the two spilt zones.
And the maximization of equation (18) is achieved by the following:

$$t^* = \arg\max_{t\in[m,M]} \left[\frac{\delta(t)}{1-\delta(t)} \left\|\mu_i - \mu_i^1\right\|^2\right] \quad (19)$$

Where μ_i, μ_i^1 are the mean of Z_i , Z_i^1 and $\delta(t) = \frac{|Z_i^1|}{|Z_i|}$, and in case of $\delta(t) = \frac{1}{2}$ the two subzones are divided from same cardinality.

Figure 14, shown that this strategy can partition the scene by using different values of zones numbers L= {20, 50}.

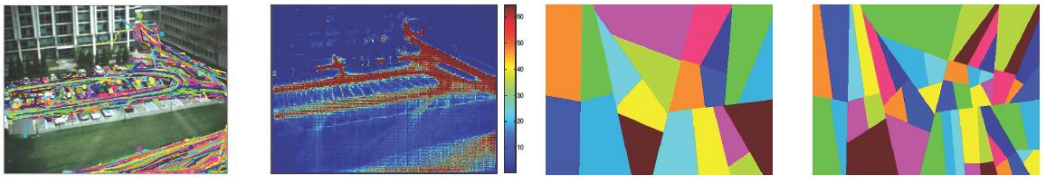

Figure 14. Partition of the scene starting from the training set depicted in (a) represented by the frequency map in (b). The quantization algorithm is applied with different values of L: (c) L=20, (d) L=50.

- **Strengths and weaknesses**

✓ The partitioning method is unsupervised learning technique.
✓ The scene could partition to selected number of zones.
✓ Zones partitioning method would help in reduce the count of points, and save data storages.

3. Summary of previous studies:

This paragraph summarizes previous studies to review the advantage and disadvantages of this methods and frameworks. For detecting anomalous trajectories many frameworks and methods built based on Hidden Markov Model (HMM): the unsupervised presented framework[11], can detect abnormal events in outdoor and indoor environments. In case of the dataset was not huge, the learning method could be improved, by using Partitioning Around Medoids (PAM) algorithm instead of K-means clustering algorithms. And then no need to specify the number of clusters in the data a priori. The second framework [12] using HMM, method can be refined by using the labeled data to guide the clustering process and by investigating more complex Dynamic Bayesian Networks (DBNs). And using mean shift clustering algorithms instead of K-means, could make the process more robust.

In the other side, another frameworks used graphs or built another creative framework. The framework [19] detected trajectories abnormal based on graph. Moreover, no need to wait object to exit the scene. However, although the method do not consuming long learning time, but still need time for scene partitioning. Another feature presented in this framework; the scene partitioning method, which divide the scene to zones based on dataset.
Isolation Based Anomalous Trajectories framework (iBAT) in [24] presented another method which partition the scene to grid-cells as preprocessing action, then selecting trajectories passes from source and destination cells. Finally, to recognize the few and different from the trajectories as anomalous trajectories, based on the Fact: To detect abnormal behaviour you need to recognize few outliers, but for normal behaviour, you need to detect most of the trajectory points. This framework achieved a great performance (Area Under the Curve AUC>0.99, over 90% detecting rate and the false alarm rate of less than 2%). Moreover, the algorithm counts the trajectories which pass through the source destination points. But in the other hand, the framework did not explain how to update maps as result of its work.
A good improvements to the partitioning method achieved by Isolation Based Online Anomalous detection Trajectories framework **(**iBOAT) [27], the mainly improvement focused on maximize the cells size by choose practical tested size, and considers the adjacent neighbors points, as same as the main points for solve low sampling rates problem and guarantee a smooth matching of the smeller normal trajectories. Beside chosen the cells size manually, no clear method presented in this framework. However, the framework still have great performance (Area Under the Curve AUC>0.99).
Zone based scene partitioning method presented in deep and more details in [20], the framework use unsupervised learning technique, and scene could partitioned to selected number of zones. And thus, zones partitioning method would help in reduce the count of points, and save the data storages.

4. *Conclusion and future work*

In this paper, frameworks for detecting anomalous trajectories were studied, and some of advantage and disadvantage of these frameworks was reviewed. In fact, this study focused on some methods of these frameworks which would be the core of **future research** study to develop unsupervised transportation framework based on detecting anomalous trajectories. This framework should be able to detect anomalous trajectories, and predict from the results the changes in the roads structure and suggesting new roads to add.
Previous summary (3), showed that iBAT/iBOAT frameworks had a good results working in online and offline environments. Although iBOAT did a good improvements to the scene partitioning method, but still maximize the cells size based on experimental results. Beside of that, a method to find main cells neighbor used in this framework for matching similar trajectories.
Using zones' based partitioning method with compute outliers trajectories by anomaly score (Equation 15), should improve the performance. In the results maps should updated based on count of discovered anomalous trajectories over threshold (Not few but different).

4.1. Future work:

The specific objectives supporting achievement the general objective by:

- Studying and evaluating the most recent related frameworks, which acting with transportations and the outliers' trajectories.
- Analyse the points of strength and weakness of related frameworks.
- Designing robust framework with good performance.
- Implement experiments over the data (which already processed) for evaluating the design framework.
- Projection the data to map data (GIS map).

4.2. Expected Contributions

The expected results for the future research could summarize as following:

- Develop robust framework with maximum performance and minimum rate of false alarms.
- The framework able to detect anomalous trajectories.
- Depending on the potential results; build application for detecting of roads changes based on the data of anomalous trajectories.
- Publishing papers on the results, and doing more in-depth research in the field.

5. *Current Results*

- Request was sent to the author of (iBAT and iBOAT) studies, for offering data had used in the experiments which described in the studies, and/or the implementation of the framework.
- Agree with private company (RCE Systems), which working on developing system for detects vehicles' moving, to offer Data describe moving of /4005/ vehicles (trajectories) by /1531950/ points.
- Preprocess was done to the data sent from RCE System, each row in original version was containing all the points of trajectory.
    - The structure of the data was redesigned by dividing the data to two main tables: Header (contain the main information about the moving object) and Transactions (contain information about the transactions).
    - The redesigning work was implemented using python scripting code **7.2.**

6. References:

[1] Potamias, *et al*.: Sampling Trajectory Streams with Spatio-Temporal Criteria. In: International Conference on Scientific and Statistical Database Management (SSDBM), pp. 275–284 (2006).

[2] Douglas, *et al*: Algorithms for the Reduction of the Number of Points Required to Represent a Line or its Caricature. The Canadian Cartographer 10(2), 112–122 (1973).

[3] Hershberger, *et al:* Speeding up the Douglas-Peucker Line simplification Algorithm. In: International Symposium on Spatial Data Handling, pp. 134–143 (1992).

[4] Maratnia, *et al*.: Spatio-Temporal Compression Techniques for Moving Point Objects. In: International Conference on Extending Database Technology (EDBT), pp. 765–782 (2004).

[5] Keogh, E., Chu, S., Hart, D., Pazzani, M.: An On-Line Algorithm for Segmenting Time Series. In: International Conference on Data Mining (ICDM), pp. 289–296 (2001).

[6] Bellman, R.: On the Approximation of Curves by Line Segments Using Dynamic Programming. Communications of the ACM 4(6), 284 (1961).

[7] Vitter, J.: Random sampling with a reservoir. ACM Transactions on Mathematical Software (TOMS) 11(1) (1985).

[8] Potamias, M., Patroumpas, K., Sellis, T.: Sampling Trajectory Streams with Spatio-Temporal Criteria. In: International Conference on Scientific and Statistical Database Management (SSDBM), pp. 275–284 (2006).

[9] Fox, D.: Adapting the Sample Size in Particle Filters Through KLD-Sampling. The International Journal of Robotics Research 22(12), 985–1003 (2003).

[10] Hightower, J., Borriello, G.: Particle Filters for Location Estimation in Ubiquitous Computing: A Case Study. In: in 6th International Conference on Ubiquitous Computing., pp. 88–106 (2004).

[11] Yingfeng Cai1, Hai Wang2 , Xiaobo Chen1, Haobin Jiang2: Trajectory-based anomalous behaviour detection for intelligent traffic surveillance. IET Intell. Transp. Syst. (2015).

[12] Piciarelli, C., Micheloni, C., Foresti, G.L.: 'Trajectory-based anomalous event detection', IEEE Trans. Circuits Syst. Video Technol. , 18, (11), pp. 1544–1554, (2008).

[13] Karavasilis, V., Blekas, K., Nikou, C.: 'A novel framework for motion segmentation and tracking by clustering incomplete trajectories', Comput. Vis.Image Understand., (2012).

[14] Hu, W., Xiao, X., Xie, D., et al.: 'Traffic accident prediction using 3-D model-based vehicle tracking', IEEE Trans. Veh. Technol., (2004).

[15] Hu, W., Li, X., Tian, G., et al.: 'An incremental DPMM-based method for trajectory clustering, modeling, and retrieval', IEEE Trans. Pattern Anal. Mach.Intell., (2013).

[16] Makris, D., Ellis, T.: 'Learning semantic scene models from observing activity in visual surveillance', IEEE Trans. Syst. Man Cybern., (2005).

[17] Singh, A., Pokharel, R., Principe, J.: 'The C-loss function for pattern classification',IEEE Trans. Pattern Recogn., (2014).

[18] Morris, B.T., Trivedi, M.M.: 'Learning, modeling, and classification of vehicle track patterns from live', IEEE Trans. Intell. Transp. Syst., [2008].

[19] L. Brun, B. Cappellania, A. Saggese, M. Vento. "Detection of anomalous driving behaviors by unsupervised learning of graphs". International Conference on Advanced Video and Signal Based Surveillance (AVSS), pages 405-410, (2014).

[20] L. Brun, A. Saggese, and M. Vento. Dynamic scene understanding for behavior analysis based on string kernels. IEEE Trans. Circuits Syst. Video Technol, PP(99):1–1, (2014). Isbn 1051-8215.

[21] M. Cuturi. Fast global alignment kernels. In L. Getoor andT. Scheffer, editors, ICML, pages 929–936. ACM, June 2011.

[22] X. Wang, K. T. Ma, G.-W. Ng, and W. E. Grimson. Trajectory analysis and semantic region modeling using nonparametric hierarchical bayesian models. Int. J. Comput. Vision, 95(3):287–312, Dec. (2011).

[23] N. Saunier and T. Sayed. Clustering vehicle trajectories with hidden markov models application to automated traffic safety analysis. In IJCNN 2006 pages 4132-4138 2006.

[24] D. Zhang, N. Li, Z.-H. Zhou, C. Chen, L. Sun, and S. Li, "iBAT: Detecting anomalous taxi trajectories from GPS traces," in Proc. 13th Int. Conf. Ubiquitous Comput., 2011, pp. 99–108.

[25] D. Aha. Lazy Learning. Springer, 1997.

[26] F. T. Liu, K. M. Ting, and Z.-H. Zhou. Isolation forest. In Proc. ICDM 2008, pages 413–422, 2008.

[27] C. Chen, D. Zhang , "iBOAT: Isolation-Based Online Anomalous Trajectory Detection," IEEE., 2013

7. *Appendices*

7.1. *Table of Figures*

7.2. *Preprocessing data code by Python*

```
head = None
startIndex = 7
count = 6
CSVInput=""
with open("C:\\Data Viewing\dfs_elis.csv") as f:
    query = ""
    HederQuery=""
    lineNum=0
    for line in f:
        lineNum+=1
        if lineNum <-1:
            print("No Lines")
            break#
        else:
            columns = line[:-1].split(";")
            if head is None:
                head = columns
            else:
                j = 0
                for i in range(startIndex, len(columns)):
                    query += " " + columns[i] + ","
                    j+=1
                    if j%count == 0:
                        query=columns[0] + ", "+query+"\n"
                        print(query)
                        CSVInput+=" "+query
                        query = ""
                HederQuery += " "+ columns[0]+ ","+ columns[1]+ ","+
columns[2]+ ","+ columns[3]+ ","+ columns[4]+ ","+ columns[5]+ "," +
columns[6]+ "," + columns[6]+ " \n"
                print(HederQuery)
                f = open('TheTrans.csv', 'w')
                s = str(CSVInput)
                f.write(s)
                f.close()
                f = open('TheHeader.csv', 'w')
                s = str(HederQuery)
                f.write(s)
                f.close()
```