Road Traffic Data analysis: Clustering and Prediction

Nicola Ronzoni, Paola Goatin

Road Traffic Data analysis: Clustering and Prediction

Nicola Ronzoni, Paola Goatin

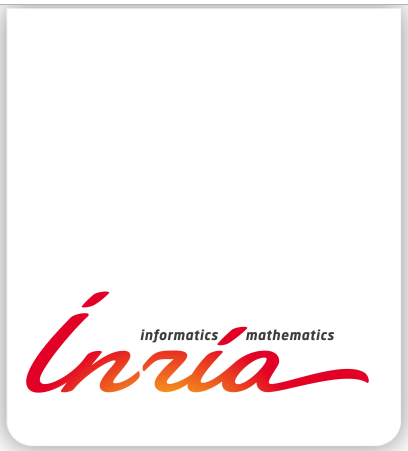
RESEARCH REPORT

N° 9426

Octobre 2021

Project-Teams ACUMES

ISSN 0249-6399 ISRN INRIA/RR--9426--FR+ENG



Road Traffic Data analysis: Clustering and Prediction

Nicola Ronzoni*, Paola Goatin*

Project-Teams ACUMES

Research Report n° 9426 — Octobre 2021 — 38 pages

Abstract: This document presents a clustering and prediction analysis on daily time series of traffic data taken from loop detector measurement at different location (France and USA). It shows the effectiveness of Soft Dynamic Time Warping and K-means algorithm for clustering and Support Vector Regression for prediction on the selected data sets. Results are commented to get information on specific traffic dynamics.

Key-words: Road traffic data, clustering, prediction

This work has been supported by the French government, through the 3IA Côte d'Azur Investments in the Future project managed by the National Research Agency (ANR) with the reference number ANR-19-P3IA-0002

* Université Côte d'Azur, Inria, CNRS, LJAD, France

RESEARCH CENTRE
SOPHIA ANTIPOLIS – MÉDITERRANÉE

2004 route des Lucioles - BP 93
06902 Sophia Antipolis Cedex

Analyse de données de trafic: clustering et prédiction

Résumé : Ce document présente une analyse de clustering et de prédiction sur des séries chronologiques quotidiennes de données de trafic obtenues à partir de mesures de détecteurs à boucle à différents endroits (France et États-Unis). Il montre l'efficacité de l'algorithme Soft Dynamic Time Warping et K-means pour le clustering et Support Vector Regression pour la prédiction sur les ensembles de données sélectionnés. Les résultats sont commentés pour obtenir des informations sur des dynamiques de trafic spécifiques.

Mots-clés : Données de trafic routier, clustering, prédiction

Contents

1 Introduction **4**

2 Tools and Methods **4**
2.1 Soft Dynamic Time Warping and K-means algorithm 4
2.2 Support Vector Regression 8
2.3 Clustering approach 11
2.4 Forecasting approach 12
2.4.1 One step predictions 12
2.4.2 Multistep predictions 13
2.4.3 Error metric of the prediction 14

3 Results **14**
3.1 Nice Data: Voie Mathis 14
3.1.1 Data description 15
3.1.2 Data preparation 16
3.1.3 Clustering on both directions 16
3.1.4 Clustering on south direction 19
3.1.5 Clustering on the north direction 22
3.1.6 Walk forward predictions on the north direction 24
3.2 Minnesota data: I-35 north bound direction to Minneapolis 26
3.2.1 Data description 27
3.2.2 Data preparation 29
3.2.3 Clustering 29
3.2.4 Prediction days and time: 31
3.2.5 Speed predictions 31
3.2.6 Flow predictions 33

4 Conclusions and outlooks **35**

1 Introduction

Road traffic data sources such as magnetic loop detectors can be analyzed by considering information on characteristic spatial and/or temporal traffic patterns. Indeed, traffic conditions may be time and/or location dependent, as some traffic conditions and driver behaviors may be correlated to specific time ranges, or to driving styles which may differ from place to place. This information can be recovered through suitable data analysis tools. In order to study the traffic patterns in different spatial and temporal locations a clustering procedure for time series is adopted. Functional classification methods are mainly used with cross sectional data. The purpose of the internship is to identify metrics that could be used with time and space dependent data such as road traffic data for both clustering and prediction tasks.

2 Tools and Methods

The research relies on three commonly accepted assumptions. First, there are a fixed number of traffic patterns that could be recognized by the clustering procedure. The number of clusters depends on the number of available loops, the variables analyzed, the length of the road segment considered and the resolution of the data. Second, short term traffic forecast mostly depends on what just happened in the road segment. To predict the flow or the speed at a certain time of a single loop, we should consider only a small amount of past observations of that loop or of all loops in the road segment [11]. Third, Support Vector Machine appears very robust to the overfitting problem, it can be used to map complex input-output relationship for the nonlinear system, thus it could be applied in solving various time series forecasting problems [4].

Clustering is the task of dividing the time series data into a number of groups such that data in the same group share some similarity. Each day that belongs to the same cluster presents analogous behaviours during the day. To better identify the trajectories represented by a group (cluster), a barycenter, also call centroid, is computed. Since we are dealing with temporal data, we have to define a strategy in order to compare different series both to assign each series in a cluster and update the centroid. We need a dissimilarity measure that is able to "match" a point of a time series x even with "surrounding" points of time series y. The Euclidean distance assumes the $i-th$ point of the x series is aligned with the $i-th$ point of the y series. It will produce a pessimistic dissimilarity measure. In the nonlinear traffic prediction, the support vector machine (SVM) regression model can solve the learning problems of the input-output relation by applying the risk minimization principle and introducing a kernel function method. This kernel should be able to compare time series and be efficient in classification tasks such as SVM. Three functions are discussed below: Soft-Dynamic Time Warping [7] used with K-means algorithm in the clustering procedure, the Radial basis function used as kernel function in the Support vector machine regression in the one step and multistep predictions and Dynamic Time Warping used for the selection of the closest day to the target in a multistep prediction approach.

2.1 Soft Dynamic Time Warping and K-means algorithm

Dynamic Time Warping is a technique to measure similarity between two temporal sequences considering not only the temporal alignment but every binary alignment of the two series. For example, a similar traffic condition could be recognized in different hours of the day in two different series. The calculation of the DTW similarity involves a dynamic programming algorithm that tries to find the optimum warping path between two series under certain constraints.

Given two multivariate series that corresponds to two different days:

$x \in \mathbb{R}^{d \times n}$ and $y \in \mathbb{R}^{d \times n}$ valued in $\mathbb{R}^d$ (where d is the dimension of the multivariate time series namely the number of loop detectors in the road segment, n is the number of daily observations in the series).

In order to compare different points of the two series ($x_i \in \mathbb{R}^d$ and $y_j \in \mathbb{R}^d$), consider a function $\delta : \mathbb{R}^d \times \mathbb{R}^d \to \mathbb{R}$, such as $\delta(x_i, y_j) = (\sum_{i,j=1}^{n} (|x_i - y_j|^p)$, where usually $p = 2$ and $d(x_i, y_j)$ is the quadratic Euclidean distance between two vectors.

A matrix of similarity is computed as

$$\Delta(x, y) := [\delta(x_i, y_j)]_{i,j} \in \mathbb{R}^{n \times n}.$$

$\Delta(x, y)$ can also be defined as local cost matrix, which must be created for every pair of series compared.

The DTW algorithm finds the path that minimizes the alignment between x and y by iteratively stepping through $\Delta(x, y)$, starting at $[\delta(x_i, y_j)]_{1,1}$ (bottom left) and finishing at $[\delta(x_i, y_j)]_{n,n}$ (upper right) and aggregating the cost [18]. At each step, the algorithm finds the direction in which the cost increases the least allowing 3 elementary moves $\rightarrow$, $\uparrow$, $\nearrow$. To improve the discrimination between different warped paths a chosen constraint can be specified. This constraint typically consists in forcing paths to lie close to the diagonal of the local cost matrix [17].

By considering $A_{n,n}$ the set of matrices that includes all possible binary alignment $\{0, 1\}^{n,n}$ between two time series x and y of the same length n, the DTW similarity measure reads as follows:

$$DTW(x, y) = \min_{A \in A_{n,n}} \langle A, \Delta(x, y) \rangle. \tag{2.1}$$

This creates a warped "path" between x and y that aligns each point in x to the nearest point in y.

However, dynamic time warping is not a distance because it does not satisfy the triangular inequality. Moreover, it is not differentiable everywhere due to the min operator.

Soft-Dynamic Time Warping is a variant of DTW that is differentiable. It uses the log-sum-exp formulation [7]:

$$DTW^{\gamma}(x, y) = -\gamma \log \sum_{A \in A_{n,n}} \exp\left(-\frac{\langle A, \Delta(x, y) \rangle}{\gamma}\right), \quad \text{where } \gamma \geq 0. \tag{2.2}$$

Despite considering all alignments and not just the optimal one, soft-DTW can be computed in quadratic time $O(d \times n^2)$ as DTW. However, like DTW, soft-DTW does not satisfy the triangular inequality. Soft-DTW is a symmetric similarity measure, it supports multivariate series as DTW, and it can provide differently smoothed results by means of the hyper-parameter γ that controls the smoothing. As showed in (2.3), DTW corresponds to the limit case when $\gamma = 0$:

$$DTW^{\gamma}(x, y) = \begin{cases} \min_{A \in A_{n,n}} \langle A, \Delta(x, y) \rangle, & \gamma = 0, \\ -\gamma \log \sum_{A \in A_{n,n}} \exp\left(-\frac{\langle A, \Delta(x,y) \rangle}{\gamma}\right), & \gamma \geq 0. \end{cases} \tag{2.3}$$

Figure 1 shows the optimal soft alignment corresponding to three different values of γ. The first case represents the DTW as showed in (2.3). Smoothing the path, by the hyper-parameter γ, can be considered as a way to make convex DTW^{γ} . In fact, DTW^{γ} converges to the sum of all costs as $\gamma \to \infty$. Thus, if δ is convex, DTW^{γ} will progressively become convex as γ grows; this is particularly visible in the third case of Figure 1 where the path becomes fuzzy.

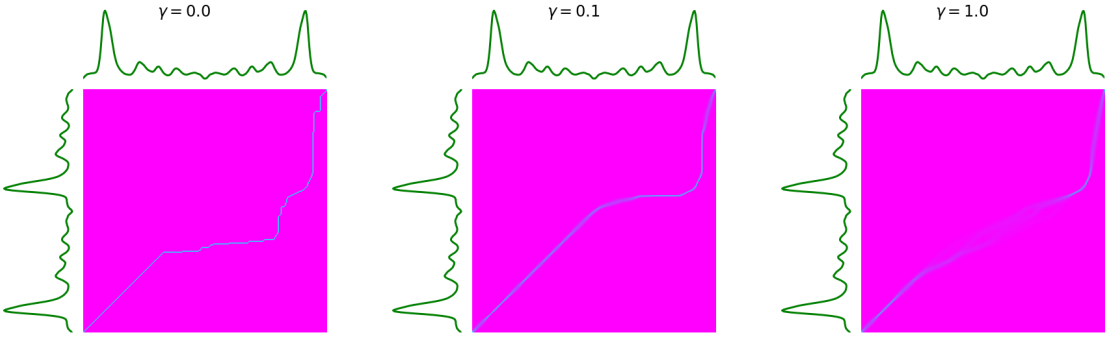


Figure 1: Soft-DTW hyperparameter γ behaviour

The Global Alignment Kernel (GAK) is a kernel that operates on time series and it is closely related to Soft-Dynamic Time Warping. The GAK can be defined as the exponentiated soft-minimum of all aligment distances:

$$k^{\gamma}_{GAK}(x,y) = \sum_{A \in A_{n,n}} \exp\left(-\frac{DTW^{\gamma}(x,y)}{2\sigma^2}\right), \tag{2.4}$$

where the hyperparameter γ that controls the smoothness in softDTW is linked to σ by

$$\gamma = 2\sigma^2. \tag{2.5}$$

Through this relation, GAK can be used to estimate the γ hyperparameter of softDTW.
As reported in [6], the bandwidth σ can be set as a simple estimate of the median distance of different points observed in different time-series of the training set, scaled by the square root of the length of time-series in the training set. In our case all daily multivariate time series have the same lenght previously indicated with n. The suggested estimation is then

$$\sigma = median||x - y||\sqrt{n} \tag{2.6}$$

and it is available in **tslearn** [21]. The number of points taken into account for estimating σ is set to 200.

Both GAK and softDTW scales in $O(d \times n^2)$. They incorporate the set of all possible alignments $A \in A_{n,n}$ in the cost matrix $\Delta(x,y)$. For this reason they provide richer statistics than the minimum of that set, which is instead the unique quantity considered by DTW. However only Global Alignment Kernel (GAK) is positive definite [1].

SoftDTW is then used with a Centroid-based clustering, such as K-means. This algorithm uses an iterative way to create the clusters by moving data points from one cluster to another, based on a distance measure, starting from an initial partitioning.

The soft-DTW is used in K-means algorithm to assign the series to the clusters $\{C_j\}_{j=1}^{k}$ and to upload the centroid of the clusters c_j. The centroid in a cluster corresponds to the multivariate time series $\in R^{d \times n}$ that minimizes the sum of the similarity measures between that time series and all time series inside the cluster. The centroid in a cluster is computed artificially, it does not correspond to a time series of the cluster. Given the ts multivariate time series (ts number of time series), each of them composed by n daily observations, the algorithm works as follows:

Algorithm 1 K-means clustering (T, k)

Require: $T = \{t_1, t_2, ..., t_{ts}\}$ set of daily time series where the generic series $t_i \in \mathbb{R}^{d \times n}$
Require: k number of clusters
Initialization: $p = 0$
Randomly choose k series in the data set and set them as initial centroids $\{c_1^{(0)}, c_2^{(0)}, \ldots, c_k^{(0)}\}$
while $c_j^{(p)} \not\approx c_j^{(p-1)} j = 1, 2, ..., k$ (inertia variation threshold) **do**
  Assign each time series t_i to the nearest cluster using $\arg\min_j DTW^{\gamma}(c_j^{(p)}, t_i)$
  $p = p + 1$
  for j=1,. . . , k **do**
    $c_j^{(p)} = \arg\min_j \sum_{i=1}^{t} DTW^{\gamma}(c_j, t_i)$ where $\{t_1, t_2, ..., t_t\} \in C_j$
  end for
end while
Output: partition of time series in clusters: $\{C_j\}_{j=1}^{k}$, centroid of each cluster $c_j \in R^{d \times n}$, $j = 1, 2, \ldots, k$.

Sometimes different initializations of the centroids lead to very different final clustering results. To overcome this problem the K-means algorithm is run 5 times with differents centroids randomly placed at different initial positions. The final result will be the best output of the five consecutive runs in terms of inertia [1]. Inertia can be recognized as a measure of how internally coherent clusters are:

$$\sum_{j=1}^{k} \sum_{i=1}^{r} \min_{c_j \in C_j} (\|t_i - c_j\|^2), \tag{2.7}$$

where r is the number of time series in each cluster.
Inertia is not only used for the initial guess of the centroids but also as a measure of variation between two consecutive runs. If at some point, inertia varies less than 1e−6 between two consecutive iterations, the model is considered to have converged and the algorithm stops.

Time series for each loop detectors are pre-processed using normalization over the whole period, both for the clustering and the prediction tasks. This normalization factor is such that each output time series is in the range [0,1], allowing to have identical scales for time series with originally different scales (the level of traffic measures, in a loop of the road segment could be different from the other ones): given a variables Y that could be the flow (veh/h), the occupancy rate (%), the speed (km/h) for each loop $d \in \{1, 2, 3, \ldots\}$ the normalization reads as:

$$\tilde{Y}_i^{(d)} = \frac{Y_i^{(d)} - MIN^{(d)}}{MAX^{(d)} - MIN^{(d)}},$$

where $i \in \{1, 2, 3 \ldots, ts \times n\}$, $MAX^d = \max\{Y_i^{(d)}\}_{i=1}^{ts \times n}$ and $MIN^d = \min\{Y_i^{(d)}\}_{i=1}^{ts \times n}$.

The choice of scaling the variable for each loop with respect to the entire set and not with respect to the single daily time series is done to preserve variability with respect to different days of the week. For example, we assume that the maximal flow reached on Sundays or Saturdays is low compared to maximal flow reached in working days [23]. The inverse transform method, that undo the scaling of a data point according to feature range $[MIN^{(d)}, MAX^{(d)}]$, is used in

[1] 5 differents sets of randomly chosen centroids are used in the algorithm. For each different centroid, a comparison is made about how much distance did the clusters move with respect to other centroids. The metric used for this comparison is the within-cluster sum-of-squares criterion (sum of square distance to their closest cluster center)

the centroid representations to have a better comprehension of paths and in the evaluation of the predictions on the test set.

2.2 Support Vector Regression

Support vector machines (SVMs) are a set of supervised learning methods used for classification and regression. The advantages of support vector machines are: effective in high dimensional spaces by using the so called "kernel trick", adaptable to time series when the number of dimensions is greater than the number of samples, memory efficient using a subset of training points in the decision function (called support vectors). After having defined a kernel function that corresponds to a dot product in some feature space ν, Support Vector Regression problem is a constrained minimization problem solved by using Lagrange Multipliers. To derive the decision function, we just need to consider dot products of support vectors and the samples.
Kernel methods use a kernel function in order to separate high dimensional data. Given two time series x_i,$y_i \in \mathbb{R}^{d\times n}$ a kernel $k(\cdot,\cdot)$ is defined as

$$k(x_i, y_i) = \langle\, \rho(x_i), \rho(y_i)\rangle_\nu, \tag{2.8}$$

where an appropriate non linear mapping $\rho : \mathbb{R}^{d\times n} \to \nu$ is used to compute the dissimilarity measure $k(x_i, y_i)$ in some (unknown) embedding space ν (also known as latent space). This approach is called the "kernel trick". It is used to map time series into a higher-dimensional feature space ν by computing the inner products between the images of pairs of data using a nonlinear function ρ. The "kernel trick" performs computations in the embedding space ν by the inner product of the transformed vectors $\rho(x_i), \rho(y_i)$. A kernel function must satisfy the Mercer's theorem: the kernel function must be symmetric; since a kernel function is the inner product of the mapping function ρ, the kernel function must be positive semidefinite. This last statement is translated in practice into the positive semidefiniteness of the so called Kernel matrix K [5]. For example, one can compute the distance between x_i and y_i in ν as

$$\begin{aligned} \|\rho(x_i) - \rho(y_i)\|_\nu^2 &= \big\langle \rho(x_i) - \rho(y_i), \rho(x_i) - \rho(y_i)\big\rangle_\nu \\ &= \big\langle \rho(x_i), \rho(x_i)\big\rangle_\nu + \big\langle \rho(y_i), \rho(y_i)\big\rangle_\nu - 2\big\langle \rho(x_i), \rho(y_i)\big\rangle_\nu \\ &= k(x_i, x_i) + k(y_i, y_i) - 2k(x_i, y_i) \end{aligned} \tag{2.9}$$

The idea of Support Vector Machine is to map training data from the input space into a high dimensional feature space also called (embedding space) ν by a function ρ and then build a separate hyperplane with maximum margin in the feature space [26]. Given a set of training data $x_i \in \mathbb{R}^{d\times n}, i = 1, ..., ts$ and $y_i = \pm 1$ class labels, SVM finds a hyperplane direction w and an offset scalar b such that:

$$\begin{cases} f(x_i) = w^T\rho(x_i) + b \geq 0 & \text{for } y_i > 0, \\ f(x_i) = w^T\rho(x_i) + b \leq 0 & \text{for } y_i \leq 0, \end{cases}$$

where $x = (x_1, x_2, ..., x_{ts})$ and $y = (y_1, y_2, ..., y_{ts})$.

SVMs have originally been used for classification tasks, but their principle can be extended to regression purpose and time series prediction. The regression task is to determine a linear function

$$f(x) = w^T\rho(x) + b \tag{2.10}$$

that can be used to estimate future values accurately. The aim is to find an hyperplane in the embedding space ν (a line) which minimises the following cost function [4]:

$$\min_{(b,w,\xi_i,\xi_i^*)} \frac{1}{2}\|w\|^2 + C\sum_{i=1}^{ts}(\xi_i + \xi_i^*) \qquad \text{subject to } \begin{cases} y_i - w^T\rho(x_i) - b \leq \epsilon + \xi_i, \\ w^T\rho(x_i) + b - y_i \leq \epsilon + \xi_i^*, \\ \xi_i, \xi_i^* \geq 0, \end{cases} \tag{2.11}$$

where $w \in \mathbb{R}^{d\times n}$, $b \in \mathbb{R}$ and ρ represent the non-linear transformation from $\mathbb{R}^{d\times n}$ to the high dimensional space ν, in which the function $f(x)$ becomes linear. The first term in (2.11) is a regularization term, which makes the function $f(x)$ flat and improve the generalization ability [13]. The second term is named as the experience risk, it penalizes errors larger than $\pm\epsilon$ using the ϵ-insensitive loss function $|\xi|_\epsilon$. For each time series in the training set, the loss function is in the form:

$$|\xi|_\epsilon = \begin{cases} |\xi_i| - \epsilon = |f(x_i) - y_i| - \epsilon, & \text{for } |f(x_i) - y_i| \geq \epsilon, \\ 0, & \text{otherwise.} \end{cases}$$

The constant $C > 0$ in (2.11) determines the trade-off between the flatness of f and the amount up to which deviations larger than ϵ are tolerated [2]. A large C assigns higher penalties to the error, so the regression is trained to minimize error with lower generalization, while a small C assigns lower penalties to errors. Errors larger than $\pm\epsilon$ are denoted with slack variables ξ_i (above ϵ) and ξ_i^* (below ϵ) as showed in Figure 2. The slack variables are introduced into the optmization problem in two ways. First, slack variables are introduced to allow certain constraints to be violated. Second, by adding slack variables to the cost function (2.11) we are aiming to simultaneously minimize them.

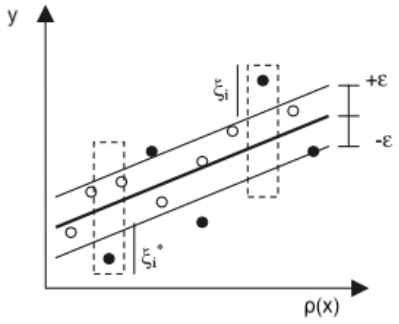


Figure 2: The tube of ϵ-insensitive loss function and points that do not meet this accuracy. The black dots located on or outside the tube are potential support vectors.

The minimisation of (2.11), with respect to (b, w, ξ_i, ξ_i^*), can be solved by applying Lagrangian theory that allows to derive the weights w [22]. The Lagrange multipliers represent solutions of the quadratic problem (2.11); they are also know as support vectors, used to forecast the regression line, pushing prediction $f(x_i)$ towards the target value y_i. Only if $|f(x_i) - y_i| \geq \epsilon$ is satisfied, Lagrange multipliers are used as support vectors. The model produced by support vector regression depends only on a subset $m \leq ts$ of the training time series [3], because the cost function (Q) for building the model does not care about training samples whose prediction is close to their target as showed in Figure 2. The key idea is to construct a Lagrange function

from (2.11) and the corresponding constraints, by introducing a dual set of variables [19]. We proceed as follows:

$$\begin{aligned} L =& \frac{1}{2}\|w\|^2 + C\sum_{i=1}^{m}(\xi_i + \xi_i^*) - \sum_{i=1}^{m}(\eta_i\xi_i + \eta_i^*\xi_i^*) \\ &- \sum_{i=1}^{m}\alpha_i(\epsilon + \xi_i - y_i + w^T\rho(x_i) + b) \\ &- \sum_{i=1}^{m}\alpha_i^*(\epsilon + \xi_i^* + y_i - w^T\rho(x_i) - b) \end{aligned} \tag{2.12}$$

In (2.12) L is the Lagrangian and $\eta_i, \eta_i^*, \alpha_i, \alpha_i^*$ are Lagrange multipliers. The dual variables have to satisfy positivity constraints, namely $\alpha_i^{(*)}, \eta_i^{(*)} \geq 0$ (where $\alpha_i^{(*)}, \eta_i^{(*)}$ represent α_i, α_i^* and η_i, η_i^* respectively).
It follows that the partial derivatives of L with respect to (b, w, ξ_i, ξ_i^*), for which we want to to minimize (2.11), have to disappear for optimality:

$$\partial_b L = \sum_{i=1}^{m}(\alpha_i^* - \alpha_i) = 0, \tag{2.13}$$

$$\partial_w L = w - \sum_{i=1}^{m}(\alpha_i - \alpha_i^*)\rho(x_i) = 0, \tag{2.14}$$

$$\partial_{\xi_i^{(*)}} L = C - \alpha_i^{(*)} - \eta_i^{(*)} = 0, \text{ which can be reformulated as } \eta_i^{(*)} = C - \alpha_i^{(*)}. \tag{2.15}$$

Substituting (2.13),(2.14),(2.15) into (2.12) yields the dual optimization problem

$$\begin{aligned} &\text{maximize } -\frac{1}{2}\sum_{i,j=1}^{m}(\alpha_i - \alpha_i^*)(\alpha_j - \alpha_j^*)\langle\, \rho(x_i), \rho(x_j)\rangle_\nu - \epsilon\sum_{i=1}^{m}(\alpha_i + \alpha_i^*) + \sum_{i=1}^{m} y_i(\alpha_i - \alpha_i^*) \\ &\text{subject to } \sum_{i=1}^{m}(\alpha_i^* - \alpha_i) = 0 \text{ and } \alpha_i, \alpha_i^* \in [0, C]. \end{aligned} \tag{2.16}$$

In (2.16), we already eliminated η_i, η_i^* through (2.15), thus (2.14) can be written as

$$w = \sum_{i=1}^{m}(\alpha_i - \alpha_i^*)\rho(x_i). \tag{2.17}$$

By substituting Equation (2.17) inside (2.10), the equation of the linear function in the embedding space ν can be written as:

$$f(x) = \sum_{i=1}^{m}(\alpha_i - \alpha_i^*)\langle\rho(x_i), \rho(x)\rangle_\nu + b \quad = \sum_{i=1}^{m}(\alpha_i - \alpha_i^*)K(x_i, x) + b, \tag{2.18}$$

where K is the positive semidefinite Kernel matrix, $K(x_i, x_j)$ could be seen as

$$\langle\rho(x_i), \rho(x_j)\rangle_\nu. \tag{2.19}$$

Since the kernel function defines the inner product in the transformed space ν, as shown in (2.8) and (2.19), Support Vector regression requires only a user specified kernel $k(\cdot,\cdot)$, such as a similarity function over pairs of data in the input dimension.
Once we have solved the value of w in terms of Lagrange multipliers, the variable b can be computed by applying the Karush-Kuhn-Tucker (KKT) conditions [8]. In this case, (KKT) conditions imply that the product of the Lagrange multipliers and constraints has to be 0:

$$\begin{aligned}
&\alpha_i(\epsilon + \xi_i - y_i + w^T\rho(x_i) + b) = 0,\\
&\alpha_i^*(\epsilon + \xi_i^* + y_i - w^T\rho(x_i) - b) = 0,\\
&\text{and}\\
&(C - \alpha_i)\xi_i = 0,\\
&(C - \alpha_i^*)\xi_i^* = 0.
\end{aligned} \tag{2.20}$$

This allows us to make useful conclusions. First, only samples (x_i, y_i) with corresponding $\alpha_i^{(*)}$ $= C$ lie outside the ϵ-insensitive tube. Second, $\alpha_i\alpha_i^* = 0$, thus there can never be a set of dual variables α_i, α_i^* which are both simultaneously nonzero. Third, b can be computed as:

$$\begin{aligned}
b &= y_i - w^T\rho(x_i) - \epsilon \text{ for } \alpha_i \in (0, C),\\
b &= y_i - w^T\rho(x_i) + \epsilon \text{ for } \alpha_i^* \in (0, C).
\end{aligned} \tag{2.21}$$

Putting all together we can use SVR by specifying the kernel function, the penalty C and the radius ϵ which determines the data inside the ϵ-tube to be ignored in the regression. The radial basis function kernel (RBF) is commonly used as kernel for regression [25]:

$$k(x, y) = \exp\{-\gamma\|x - y\|^2\}. \tag{2.22}$$

The parameter γ in the RBF kernel indicates how far the influence of a single training observation (day) goes, low values of γ meaning "far" and high values meaning "close". The γ can be seen as the inverse of the influence of the observations selected by the model as support vectors. If γ is too large, the radius of the area of influence of support vectors only includes the support vectors and no amount of the regularization parameter C will be able to prevent overfitting. If γ is too small, the model is too constrained, thus it can not capture the complexity of the data.

2.3 Clustering approach

K-means requires the number of clusters k, as clustering parameter. Getting the optimal number of clusters is very significant in the analysis. There is no unique approach for finding the right number of clusters. Since Soft-DTW is differentiable it could be also used as a function to evaluate the cohesion inside each cluster and the separation with respect to the nearest cluster. The silhouette coefficient [16] is a measure of how similar a time series is to its own cluster (cohesion) compared to other clusters (separation). The silhouette can be computed with the soft-DTW metric and it takes values in the range [-1, 1]. However the highest value of this coefficient is reached when the number of clusters is set to 2. If k is too low some characteristics of the traffic behaviour can not be separated. On the contrary, if the number of clusters is too high, each time series starts representing its own cluster. In addition, as the number of clusters increases, it is more difficult to interpret the traffic behaviour captured by each cluster. To deal with this problem, following an empirical method [27], we fixed a lower bound for the minimum number of observations in each cluster approximately at $\sqrt{\frac{3}{2} \times ts}$, where ts is the number of daily time series in the train set. If the number of time series within a cluster is lower than the

fixed bound the cluster does not generalize well a particular path, thus the interpretability of the cluster is too difficult. We remark that all clusters must appear on the test set, otherwise the number of clusters have to be reduced.

2.4 Forecasting approach

For the implementations of SVR for time series prediction, we used two different prediction horizons: one step and multistep predictions. For both prediction horizons, we develop two options: prediction using not only data from the loop detector that we would like to predict, but all loops considered in the road segment and prediction using only data from the loop detector that we would like to predict. Therefore we have 2 options for SVR one step predictions and 2 options for SVR multistep predictions. The multistep prediction aims to forecast h steps in the future given the prediction origin t, while the one step prediction wants to predict in the short future accordingly to the data resolution. Since we are assuming that only the most recent observations influence future values of traffic condition we use a fixed numbers of observations before the prediction origin t (indicated with s). In order to compare the two options of multistep SVR predictions with another multistep approach we use also a classification method based on Dynamic Time Warping metric.

2.4.1 One step predictions

The approach for SVR one step prediction is called "Walk forward prediction". For "Walk forward prediction" we develop two options: "Walk forward SVR all loops" that consider not only the data from the loop that we want to predict, but all loops in the road segment and "Walk forward SVR single loop" that uses only data from the loop that we want to predict.

Walk Forward SVR all loops Given a set of multivariate time series, only recent past observations are used to predict one step in the future. The numbers of more recent observations used is called window size. After having select the window size and a prediction origin t, we construct a set of training samples $X_{train}(t) = \{x(t), x(t-1), ..., x(t-s)\}$ to predict $x(t+1) = Y_{train}(t)$. After having trained the predictor and used it in the forecasting of $\hat{Y_{test}}(t)$ given the test day $X_{test}(t)$, once $Y_{test}(t) = x(t+1)$ becomes available, we update the prediction origin t and repeat the procedure iteratively. As the origin moves forward, the training set does not increase since we consider only a fixed amount of past observations s. This leads to a less expensive computation. The walk forward predicts one loop each time by using all other loops in the road segment. At every iteration $X_{train}(t)$ is a 3-D array of size $[ts, s, d]$, while $Y_{train}(t)$, the loop that we would like to predict becomes, a 1-D array $[ts]$, since the last two dimensions are equal to 1 (1 step in the future, 1 loop). As a consequence, $X_{test}(t)$ is a 3-D array of size $[1, s, d]$ and $Y_{test}(t)$ is a 1-D array $[1]$. Since SVR supported format is 2-D arrays for the input, we transform $X_{train}(t)$ a 3-D array of size $[ts, s, d]$ and $X_{test}(t)$ a 3-D array of size $[1, s, d]$ in 2-D arrays respectively of size $[ts, s \times d]$ and $[1, s \times d]$. In order to tune the hyperparameters C and ϵ of the support vector regression a Gridsearch [20] is performed at every iteration for the new prediction in the training set. The set for C is $[0.1, 1, 10, 100]$, the set for ϵ is $[0.01, 0.1, 1, 10]$. The number of groups (folds) considered in the cross validation is 3 and the metric used to select the best combination of C and ϵ is the mean squared error. For the parameter γ for RBF kernel we use $\frac{1}{s \times d}$ formulation, which is the inverse of the number of features in the train set.

Walk Forward SVR single loops The "Walk forward SVR single loop" predicts one loop each time by using data only from the loop that we want to predict. At every iteration $X_{train}(t)$ is

a 2-D array of size $[ts, s]$, the last dimension is equal to 1 since we use only 1 loop, while $Y_{train}(t)$ becomes a 1-D array $[ts]$ since the last two dimensions are equal to 1 (1 step in the future, 1 loop). As a consequence we do not need to reshape our array since $X_{train}(t)$ and $X_{test}(t)$ are already 2-D arrays respectively of size $[ts, s]$ and $[1, s]$. In order to tune the hyperparameters C and ϵ we use the same approach of "Walk forward SVR all loops". The parameter γ for RBF kernel is set to $\frac{1}{s}$, which is again the inverse of the number of features in the train set.

2.4.2 Multistep predictions

The approaches for SVR multistep predictions are: "SVR all loops" and "SVR single loop". As for the single loop the difference between the two options are the input data from all loops of the road segment or from the single loop that we want to predict. In addition we add a third approach, an empirical method called "Classification", based on the recognition of the closest day to the target in the train set based on past observations. In general in the multistep predictions the prediction origin t does not move forward as in the "Walk forward approach", this leads to higher temporal distances between the target and the input data as h becomes greater.

SVR all loops Starting from the train set of multivariate time series $[ts, n, d]$, we construct a set of training samples $X_{train}(t) = \{x(t), x(t-1), ..., x(t-s)\}$ to predict the future of observations $Y_{train}(t) = \{x(t+1), x(t+2), ..., x(t+h)\}$. In our case the numbers of observations in the past is lower than the numbers of observations that we would like to predict in the future: $s \leq h$. The "SVR all loops" predicts one loop each time by using all other loops in the road segment. Support Vector Regression supports only time series single target which could be express as a 1-D array $[ts]$, to extend SVR to multisteps prediction we used multioutput regression which is available in [15]. The strategy consists of fitting one regressor per target: one regressor per observation in the future that we would like to predict. This is a simple strategy for extending regressors that do not natively support multi-target regression as SVR. Since we use multioutput regression, we have to reshape the input-output arrays as the only supported format is 2-D arrays for both input and output. Thus we transform $X_{train}(t)$, a 3-D array of size $[ts, s, d]$, in a 2-D array of size $[ts, s \times d]$ and the $Y_{train}(t)$, a 3-D array of size $[ts, h, d]$, in a 2-D array of size $[ts, h]$ by selecting the loop that we would like to predict. In order to tune the hyperparameters C and ϵ of the support vector regressions a Gridsearch over multioutput regression is computed using a pipeline. The purpose of the pipeline is to assemble several steps that can be cross-validated together while setting different parameters. In this case the cross validation, for each possible combination of hyperparameters, takes into account multisteps prediction. The last passage, after having fit the model with $X_{train}(t)$ and $Y_{train}(t)$ using the best combination of hyperarameters, is to predict $\hat{Y_{test}}(t)$ given the test day $X_{test}(t)$, which is a 2-D array of size $[1, s]$.

SVR single loop The difference between "SVR all loop" and "SVR single loop" is the reshape phase needed in the first case in order to consider the data from the other loop. For "SVR single loop" the reshape phase is not requested. $X_{train}(t)$ and $X_{test}(t)$ are 2-D arrays respectively of size $[ts, s]$, and $[1, s]$: the last dimension is equal to one since we use only 1 loop.

Classification approach To Compare the multisteps SVR predictions with another method, we implement the "Classification" approach. For the "Classification" approach, we consider as in the SVR only most recent past observation s. The considered information of the test day before the prediction origin is a 3-D array of size $[1, s, d]$, while the information of the train set chosen for the "Classification" approach is a 3-D array of size $[ts, s, d]$. In order to predict h future observations of the test day, starting from the prediction origin t, we use the dynamic

time warping metric to compare the traffic in the road segment of the test set and each day of the train set. We compare the s observations before the prediction origin t of the test day with the traffic in the road segment of each day of the train set in the same range. Then we select the h observations after the prediction origin t of the day in the train set that is closest to the target day. We use a window size of s observations in the past to predict h observations in the future. Suppose we would like to predict the traffic from 8:00 to 8:54 of the test day. The classification approach uses the dynamic time warping metric to compare the traffic in the road segment of the test set from 6:00 to 7:54 with the traffic in the road segment of each day of the train set from 6:00 to 7:54 (in this example the window size in the past s is set to 2 hours). The prediction from 8:00 to 8:54, will be the traffic in the range 8:00 to 8:54 of day in the train set closest to the test day in the range 6:00 to 7:54. The "Classification" approach returns a prediction for every loop and not a single loop at each time as SVR.

2.4.3 Error metric of the prediction

To evaluate the prediction we use the following metric, that is the root mean squared error of all the predictions in the road segment:

$$\hat{E}^k = \sqrt{\frac{1}{h \times d} \sum_{i=1}^{h} \sum_{j=1}^{d} |Y_i^{(j)} - \hat{Y}_i^{(j)}|^2}, \tag{2.23}$$

for $k \in \{\text{flow}, \text{speed}\}$ and where $\hat{Y}_i^{(j)}$ represents the predicted values for a specific loop and $Y_i^{(j)}$ is the corresponding ground truth. We do not evaluate the prediction for every loop detector but for the entire road segment. We predict h observations in the future, for both one step or multistep predictions, and the number of loops detector in the road segment is d thus we evaluate the performance of the predictions on $h \times d$ observations.

3 Results

3.1 Nice Data: Voie Mathis

With this data set, we study the clustering procedure with different type of traffic variables: occupancy rate (%), flow (veh) and speed (km/h). We find out that considering the ramps in the clustering of the flow does not change significantly the classification of the test days. Moreover, we apply one step ahead prediction to the speed variable. We use three different approaches for SVR predictions:" Walk Forward SVR single loop", "Walk Forward SVR all loops" and "Walk Forward SVR all loops" with the clustering procedure computed before the prediction phase. Standard SVR training has $O(m^3)$ time complexity, but its compute requirements increase rapidly with the number of training vectors [24]. The clustering procedure before applying the prediction procedure can help the training phase, avoiding daily time series that are really far from the target, speeding up the computation. We used a resolution of 6 minutes, namely a 6-minute average for each variable. We try to study the impact of the clustering procedure applied before the one step forward prediction with SVR by comparing the error of the 3 types of predictions.

3.1.1 Data description

The data at our disposal (courtesy of Métropole Nice Côte d'Azur [12]) are described below.

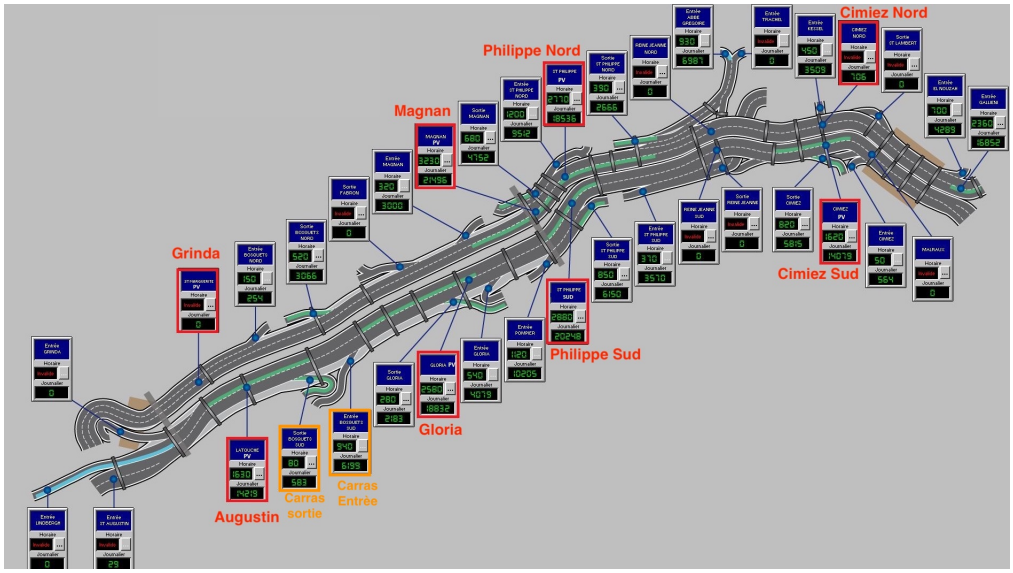


Figure 3: Map of loops and ramps on Voie Mathis

- Voie Mathis, north direction:
  - 5.7 km long road stretch
  - lanes per loop detector stations, $l \in \{1, 2, 3\}$
  - 4 aggregate loop detector stations (IDs: Cimiez Nord, Philippe Nord, Magnan, Grinda $d \in \{1, 2, 3, 4\}$
  - 1 on-ramps (ID: Bosquets Entrée)
  - 1 off-ramps (ID: Fabron Sortie)

ID	Cimiez Nord	Philippe Nord	Magnan	Grinda
location in km	0	1.7	2.1	5.7

Table 1: Location and arrangement of loop detectors, north direction.

- Voie Mathis, south direction [2]:
  - 5.7 km long road stretch
  - 4 aggregate loop detector stations (IDs: Augustin, Gloria, Philippe Sud, Cimiez Sud $d \in \{1, 2, 3, 4\}$
  - 1 on-ramps (ID: Carras Entrée)
  - 1 off-ramps (ID: Carras Sortie)

[2] Augustin loop detector was out of order, Philippe Sud detector does not contain speed information

ID	Augustin	Carras	Carras	Gloria	Philippe Sud	Cimiez Sud
location in km	0	2.1	2.2	2.5	3.6	5.7

Table 2: Location and arrangement of loop detectors, south direction: main road loop detectors (black), off-ramps (red), on-ramps (green).

We use the following notations:

- $Y_{\mathrm{k}}^{(d)}(t)$: $k \in \{\text{flow} \quad (veh), \quad \text{speed} \quad (km/h), \quad \text{occupancy rate} \quad (\%)\}$, measurement at time t, for a loop detector d,
- $Y_{\mathrm{k}}^{(r)}(t)$:$k \in \{\text{flow} \quad (veh), \quad \text{speed} \quad (km/h), \quad \text{occupancy rate} \quad (\%)\}$, measurement at time t, for a ramp loop detector with ID $r \in \{1, 2\}$

We extract the flow, speed and occupancy rate data for the lane-loop detectors, then we sum up the flow data of each lanes of a specific loop detector, while we compute the mean for the speed and the occupancy rate.

3.1.2 Data preparation

Since the distance between two consecutive loop detectors can be greater than 2 km, we do not consider the data from the adjacent loop detector in the imputation procedure.

The completion of missing data (only if they are missing for less than 3 hours) is equivalent for loops s and ramps r:

$$Y_k^{(c)}(t) = \frac{Y_k^{(c)}(t-1) + Y_k^{(c)}(t+1)}{2},$$

where c indicates a ramp loop detector with ID r or a loop detector station with ID d.

We used data from year 2019 as train set (excluding days with more than 3 hours of consecutive missing data). Besides, we select 4 weeks (28 days) in 2020 as test set:

- Mon, 20.01. – Sun, 26.01;
- Mon, 24.02. – Sun, 01.03 (winter school break);
- Mon, 23.03. – Sun, 29.03 (first lockdown);
- Mon, 20.07. – Sun, 26.07 (summer).

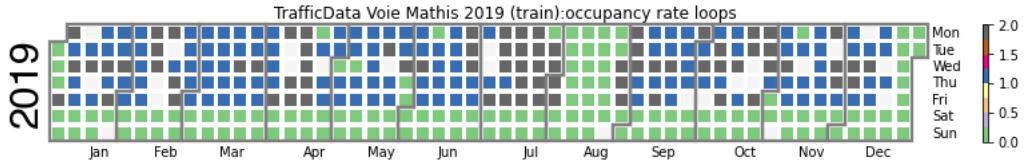
For each day, we select only observations from 6:00 am to 10:54 pm, i.e. 170 observations with 6 minute granularity.

We fix the number of cluster to 3 for every procedure applied. Increasing the number of cluster to 4 makes one cluster disappear in the test set, or produces a cluster with a number of days lower than the fixed bound.

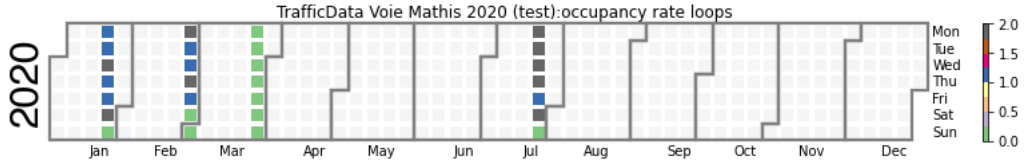
3.1.3 Clustering on both directions

In this section, the clustering procedure takes into account both direction of the Voie Mathis to have an overview of the traffic in the area. We consider the occupancy rate of 6 loops detectors: three in the North direction (Cimiez Nord, Philippe Nord, Magnan) and three in the South direction (Gloria, Philippe Sud and Cimiez Sud) (see Figure 3). For 2019 data, we use

340 daily time series as input of K-means seeded with softDTW. The resulting classification is showed in Figure 4. The parameter γ is set to 11 using (2.5),((2.6). The first cluster $k = 0$ identifies no working days (Saturdays, Sundays, public holidays[3] and the month of August). The second cluster $k = 1$ represents "normal" working days. The third cluster $k = 2$ selects working days during school holidays[4] and majority of Wednesdays especially in the period September - December and January.



(a) Train set (loops)



(b) Test set (loops)

Figure 4: Occupancy rate (%) clustering, both directions (6 loops)

For 2020 data, we can see classified in the cluster $k = 2$ Wednesday 22/1, Monday 24/2 and Friday 28/02, which belong to the week of school break[5], and the majority of the days in the July week, where the school are closed as well. All days in the week of March (week during first lockdown) are classified in the cluster $k = 0$, as the majority of Saturdays and Sundays in the other weeks.
In Figure 11, a random sample of time series for every detector is represented with the corresponding centroid of the cluster. For all detectors, we can recognize the same tendency that distinguishes the three clusters: the first cluster $k = 0$ does not present morning and afternoon peaks, the second cluster $k = 1$ represents most trafficated days while the third one $k = 2$ identifies working days without "school effect" with morning and afternoon peaks lower than the ones in the second cluster $k = 1$. As expected, the level of traffic between detectors is not the same, Cimiez Nord and Cimiez Sud detectors seem to have in general a lower level of traffic than the other ones.

[3] 22/4 Easter Monday, 1/5 Labour day, 8/5 Victory in Europe Day, 30/5- 31/5 Ascension Day, 10/6 Whit Monday, 14/7 Bastille Day, 1/11 All Saints' Day, 11/11 Armistice Day (2019)
[4] Winter break from 9/2 to 25/2, Easter break from 6/4 to 23/4, Summer break from 5/7 to 2/9, All Saints' break from 19/10 to 4/11 (2019)
[5] from 15/2 to 2/3 (2020)

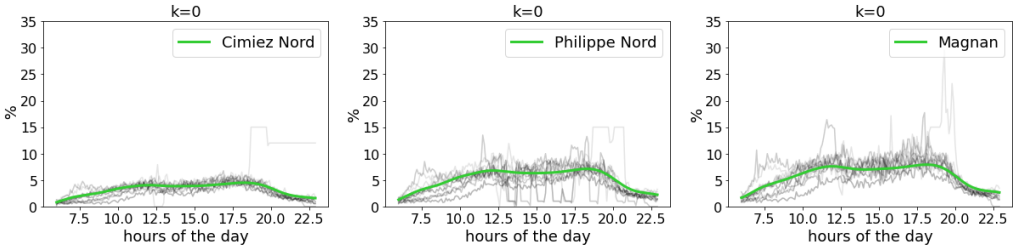


Figure 5: No working days, north direction

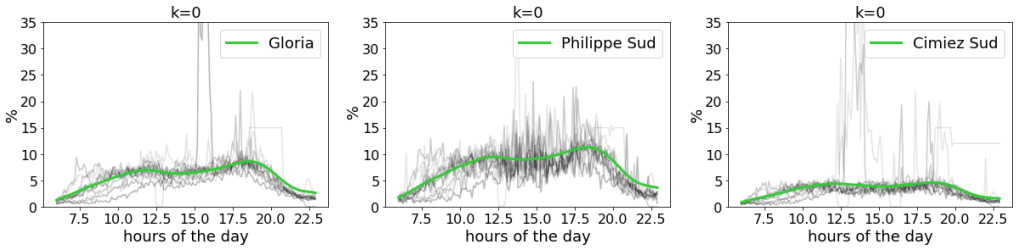


Figure 6: No working days, south direction

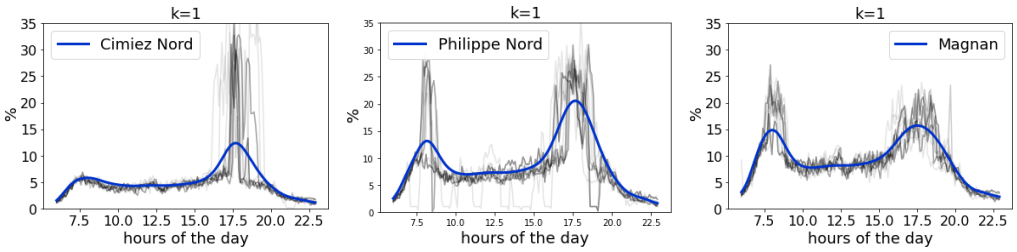


Figure 7: Working days, north direction

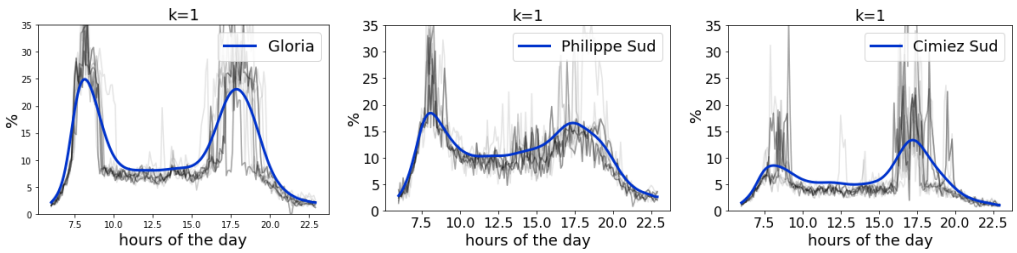


Figure 8: Working days, south direction

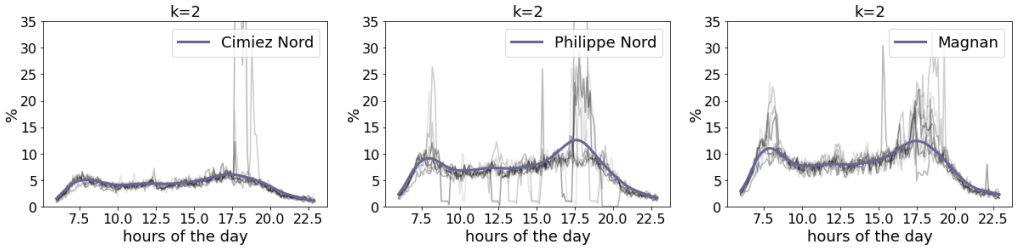


Figure 9: Working days without school, north direction

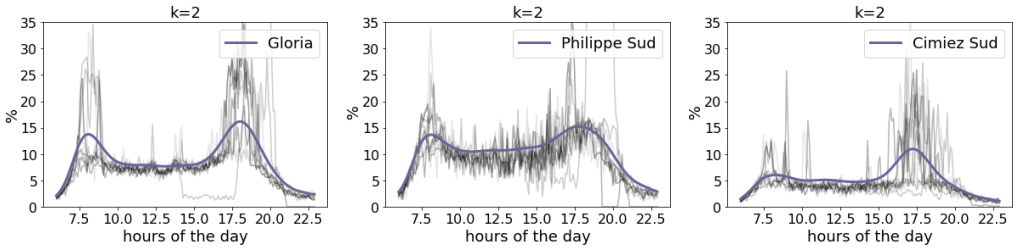


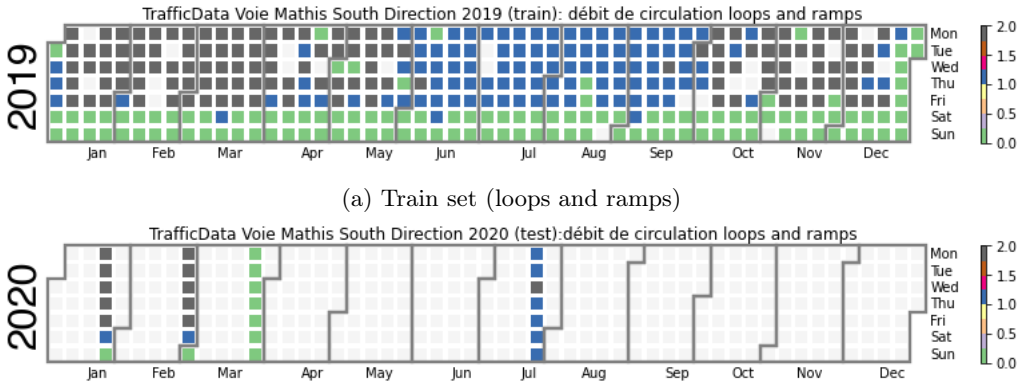
Figure 10: Working days without school, south direction

Occupancy rate (%) centroids: both directions (6 loops).

3.1.4 Clustering on south direction

For the south direction, we try to study the impact of considering the ramps in the clustering procedure. We use the flow measurements of 3 loops (Gloria, Philippe Sud and Cimiez Sud)[6] and 2 ramps (Carras sortie and Carras entrèe) (see Figure 3). For 2019 data, we use 343 daily time series as input of K-means seeded with softDTW. As we can see in Figures 12 and 13, the classification of the days both in the train and test sets give similar results. The parameter γ is set to 19 in Figure 12 and to 16 in Figure 13. In the train set, the cluster $k = 0$ represents no working days (Saturdays, Sundays and Public holidays), the cluster $k = 1$ identifies working days during the summer months (June, July, August and September), while the cluster $k = 2$ portrays working days during the winter months (from January to May and from October to December). In the test set, the cluster $k = 0$ contains all the days in the week of March, when restrictive measurement were in place due to Covid-19. The cluster $k = 1$ identifies the week in July and Saturdays in the weeks of January and February, while the cluster $k = 2$ represents the others working days. From the classification of the days in the test set, we can say that the level of traffic in 2020 during the weekend (especially Saturdays) is greater than in 2019. Saturdays and Sundays belong mostly to cluster $k = 0$ in the train set, but in the test set only Sundays (January and February weeks) are identify by the cluster $k = 0$.

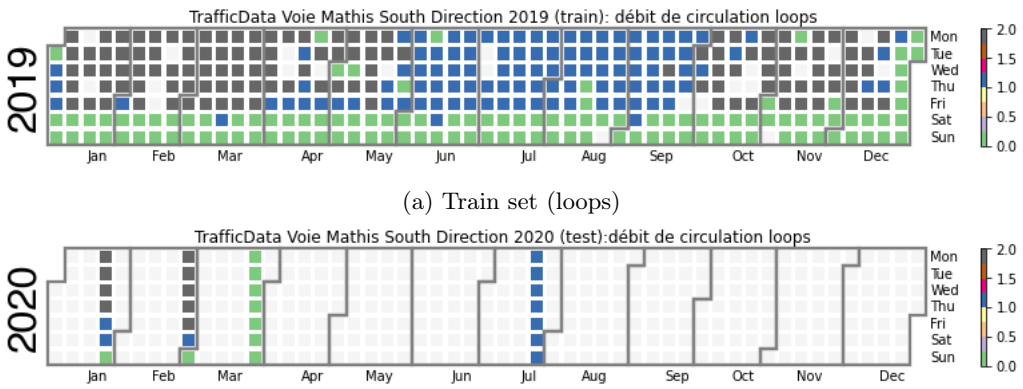
[6] Augustin loop presents aberrant measurements for one lane



(a) Train set (loops and ramps)

(b) Test set (loops and ramps)

Figure 12: Flow clustering, south direction (3 loops and 2 ramps)



(a) Train set (loops)

(b) Test set (loops)

Figure 13: Flow (*veh*) clustering, south direction (3 loops and 2 ramps)

In Figure 17, a random sample of time series for every detector is represented with the corresponding centroid of the cluster. For all detectors we can recognize the same tendency that distinguishes the three clusters: the first cluster $k = 0$ does not present the morning peak. The second cluster $k = 1$ represents traffic days during the summer where the peak in the morning is smoother than the one in the third cluster $k = 2$. In addition the traffic levels outside the morning and afternoon peaks is higher than the ones present in cluster $k = 2$ for Gloria and Philippe Sud detectors, probably due to tourist traffic in the city centre. The last cluster $k = 2$ identifies working days in winter with highest peaks of traffic especially in the morning. As expected the level of traffic in the ramps is not the same as in the loops detectors. The off-ramp Carras register the lowest level of traffic, while the paths of the centroids in the on-ramps Carras follow the paths of the centroids of the detectors with different level of traffic. Gloria and Philippe Sud are the most trafficated locations, probably due to their proximity to the station and the city centre.

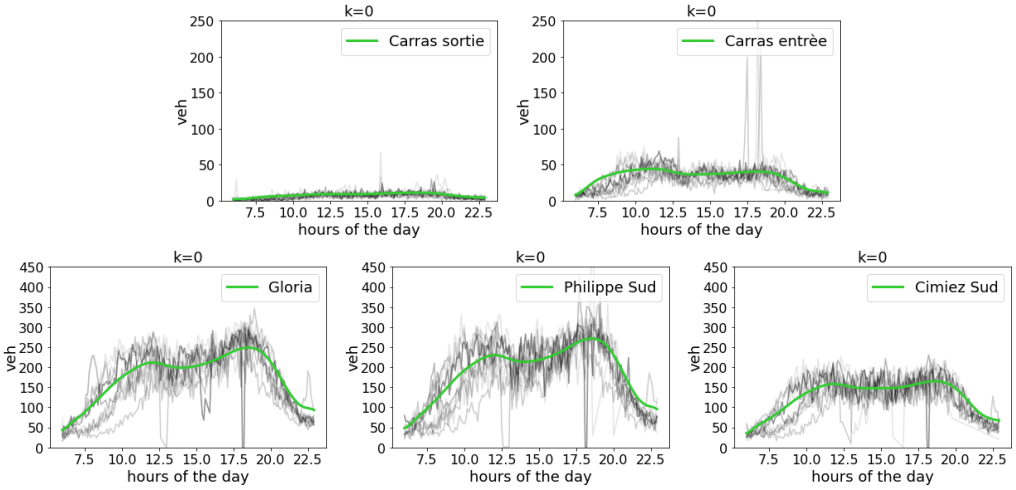


Figure 14: No working days

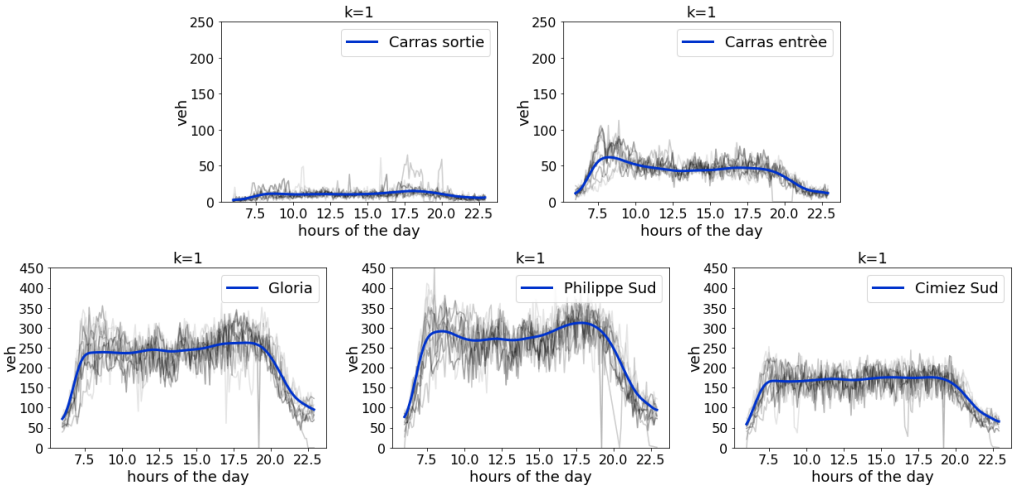


Figure 15: Working days in summer

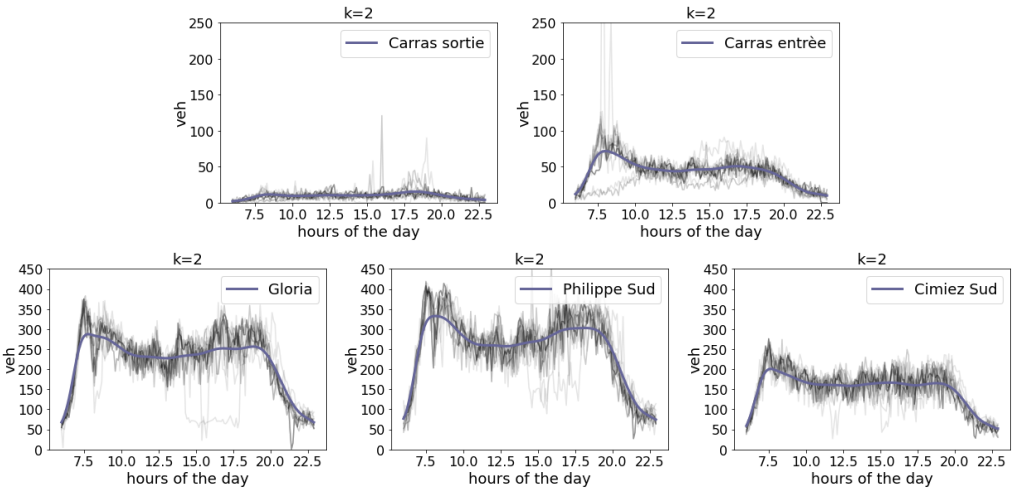


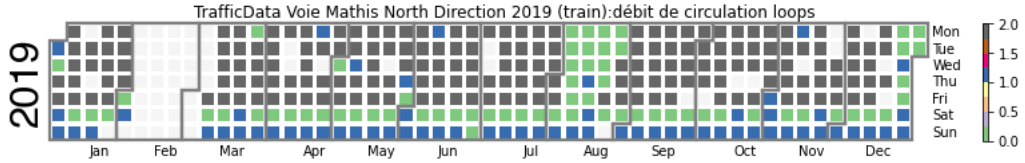
Figure 16: Working days in winter

Flow (*veh*) centroids: south direction (3 loops and 2 ramps).

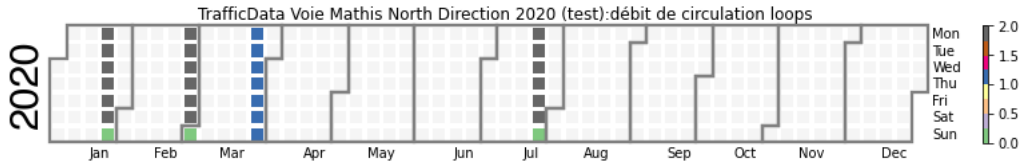
3.1.5 Clustering on the north direction

For the north direction we run the clustering procedure for two different variables: flow and speed. We use 4 loops (Cimiez Nord, Philippe Nord , Magnan, Grinda). For 2019 data, we use 312 daily time series as input of K-means seeded with softDTW. Figure 18 shows the classification of the days with respect to the flow variables. The parameter γ is set to 29. In the train set, the cluster $k = 0$ identifies the Saturdays and the month of August. The cluster $k = 1$ selects Sundays and public holidays, while the cluster $k = 2$ portrays the other working days. In the test set, the week of March belongs to cluster $k = 1$ (less trafficated cluster contains the entire week in lockdown phase). In the remaining weeks, the cluster $k = 0$ represents Sundays and not Saturdays as in the train set and the cluster $k = 2$ captures the days from Monday to Saturday. As in the south direction, also in the north direction we can say that the level of traffic in 2020 during the weekend (Saturdays and Sundays) is greater than in 2019.

In Figure 22 a random sample of time series for every detector is represented with the corresponding centroid of the cluster. Cluster $k = 0$ does not present traffic peaks in the morning and in the afternoon. Higher number of cars is registered in the time slot 17:00-20:00. Cluster $k = 1$ represents days with less traffic. An inflection of the traffic in the time slot 12:00-16:00 distinguishes it from cluster $k = 0$. Cluster $k = 2$ has peaks in the morning and in the afternoon, for all loops detectors the peak in the morning is higher than the one in the afternoon. Philippe Nord and Magnan register a higher level of traffic in all clusters probably due to their proximity to the city centre and the station.



(a) Train set (loops)



(b) Test set (loops)

Figure 18: Flow (*veh*) clustering, north direction (4 loops)

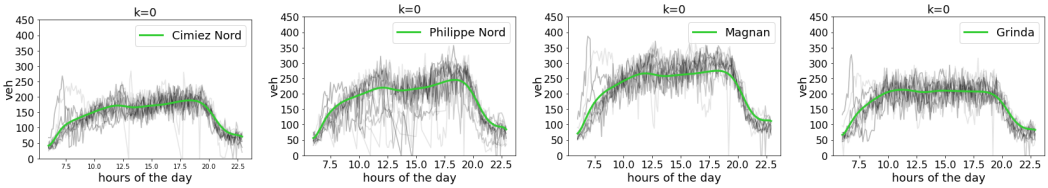


Figure 19: Saturdays, August

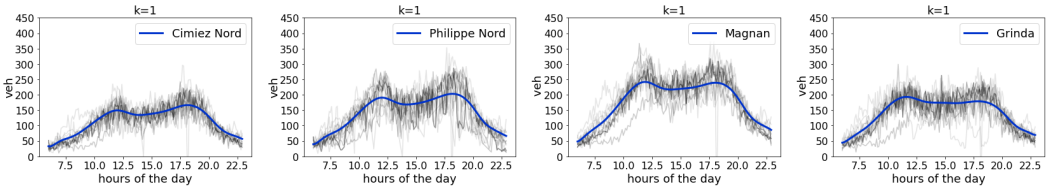


Figure 20: Sundays

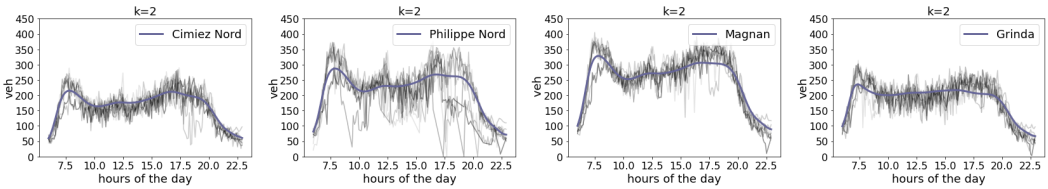


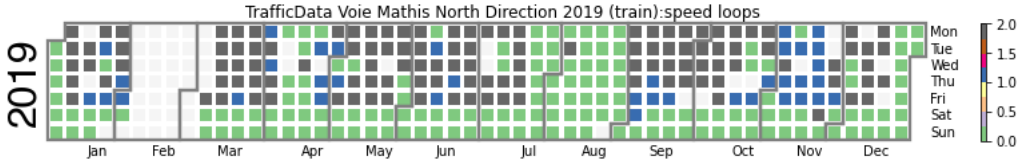
Figure 21: Working days

Flow (*veh*) centroids: north direction: 4 loops.

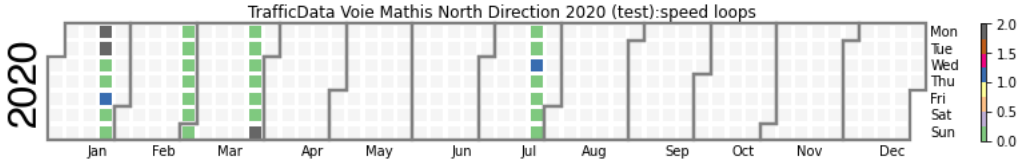
Figure 23 shows the classification of the days with respect to the speed variables. The parameter γ is set to 15. In the train set, the cluster $k = 0$ maps Saturdays, Sundays, public holidays, the last week of July, August and the two weeks of school break for Easter holidays. Clusters $k = 1$ and $k = 2$ identify the remaining working days. The first difference of the speed clustering with respect to the flow clustering is that in the speed one we have 2 clusters for working days, instead in the flow clustering we have 2 clusters for no-working days and August. The second

difference is in the test set: we do not have a proper distinction between working-days and no-working days for the speed clustering.

In Figure 27 a random sample of time series for every detector is represented with the corre-



(a) Train (loops)



(b) Test (loops)

Figure 23: Speed (km/h) clustering, north direction (4 loops)

sponding centroid of the cluster. The cluster $k = 0$ identifies the free flow situation, where the centroid of each loop remains around the speed limit of 70 km/h all day long. The cluster $k = 1$ contains days in which accidents, stationary vehicles, obstacles or heavy rain have been recorded[7]. This cluster registers a heavy reduction in the speed especially in the time slot 17:00-19:00 and even in the time slot 7:00-9:00 for Grinda loop. The cluster $k = 0$ represents the remaining working days, where the reduction of the velocity in the traffic peaks is less pronounced than in cluster $k = 1$. During working days, the greatest reduction in the speed, in the time slot 7:00-9:00, is at Grinda loop (see Figure 3), at the end of the Voie Mathis, in the direction of the airport and motorways to Sophia-Antipolis/Antibes/Grasse.

3.1.6 Walk forward predictions on the north direction

As mentioned earlier we run the Walk forward prediction in three different ways: only considering data from the loop detector that we want to predict ("Walk Forward SVR single loop"), considering data from all others loops of the road segment with and without previous clustering (respectively "Walk Forward SVR all loops" and "Walk Forward SVR all loops with clustering"). We select 5 days from the test set in Figure 23, that belong to the three clusters, and we select a time slot of 2 hours during the morning or the afternoon peaks. In the "Walk forward SVR all loops" at every iteration, $X_{train}(t)$ is a 3-D array of size $[ts, s, d] = [312, 5, 4]$ while $Y_{train}(t)$, the loop that we would like to predict, becomes a 1-D array $[ts] = [312]$ since the last two dimensions are equal to 1 (1 step in the future, 1 loop). For the "Walk forward SVR all loops with

[7] evenements-2019.xlsx (Events having impacted traffic on Voie Mathis)
Days that belong to cluster $k = 1$ in the train set (2019) and for which have been recorded impact events: 18/1 accidents, 25/1 stationary vehicles, 1/2 accident, 22/3 stationary vehicles, 1/4 accident, 3/4 stationary vehicles, 23/4 stationary vehicle, 9/10 stationary vehicle, 12/10 accident, 18/10 obstacle, 6/9 obstacle, 7/9 accident, 12/9 stationary vehicle, 13/9 obstacle, 20/9 stationary vehicle, 25/10 stationary vehicle, 31/10 stationary vehicle, 5/11 accident, 7/11 obstacle, 8/11 stationary vehicle, 12/11 stationary vehicle, 14/11 stationary vehicle, 17/11 stationary vehicle, 18/11 accident, 19/11 stationary vehicle, 20/11 accident, 25/11 obstacle

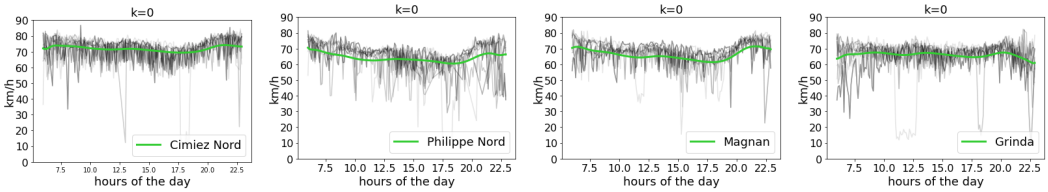


Figure 24: No working days, August and Easter

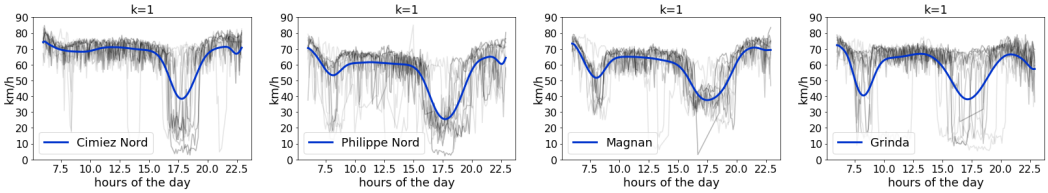


Figure 25: Congested Working days

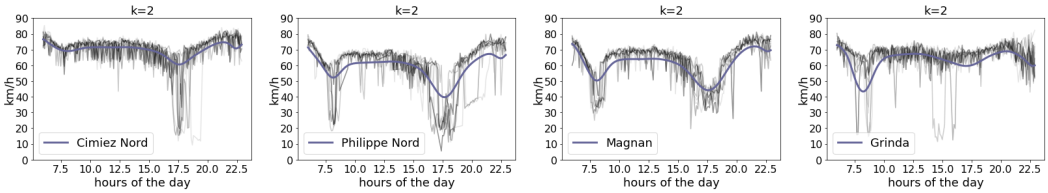


Figure 26: Working days

Centroids speed (km/h) north direction: 4 loops.

clustering", we fixed the number of clusters to 3 [8]: we train the K-means algorithm with soft Dynamic Time Warping on $X_{train}(t) + Y_{train}(t)$ (3-D array of size $[ts, s+1, d] = [312, 6, 4]$) multivariate time series, we see in which cluster belongs $X_{test}(t)$ (3-D array of size $[1, s, d] = [1, 5, 4]$), then we select only $X_{train}(t) + Y_{train}(t)$ that belong to the same cluster of X_{test}[9] to train the model. As the prediction, also the clustering procedure is done iteratively as the target become available and the prediction origin t steps forward. At every iteration, the first dimension of the input-output 3-D arrays $(X_{train}(t), Y_{train}(t) : [z, s, d], [z, 1, d]$ respectively) is reduced by the clustering procedure before the prediction with SVR. The number of days considered for the SVR prediction is $z \leq ts$ as we only select days which belongs to the same cluster of the target day $X_{test} : [1, s, d]$.
To decide the number of observations s, we try s=5 and s=10 namely 30 minute and 1 hour before the prediction origin t and we select the window size that give us lower errors in the selected days using (2.23).

Remark 1. *The train set for the predictions is the same as the clustering, a 3-D array of size* $[312, 170, 4]$ *of 2019 data for "Walk Forward SVR all loops" options while for "Walk Forward SVR single loop" is a 2-D array of size* $[312, 170]$ *since we use data only from the loop detector that we want to predict.*

The results of the predictions are showed in Table 3, where the selected window size s in

[8] we assume that the 3 clusters are separable at any time of day, regardless of the length of the array considered during the train phase

[9] soft-DTW as well as DTW can classify arrays of different lengths

the past is equal to 30 minutes. For "Walk Forward SVR all loops" options (with and without previous clustering) we can see that reducing the number of training days with the clustering procedure does not improve the forecasting accurancy. Only for one day the speed prediction is slightly better with the clustering procedure (Friday 24.02.2020). Although the clustering procedure can be useful to reduce the speed computation when the training data set is large, the hyperplane constructed by SVR depends only on a portion of the training samples called support vectors. Removing training days by the clustering procedure does not guarantee that all these samples are not relevant as support vectors. Some days might have no effect on building the proper decision function but other can affect the construction of the hyperplane [9]. Thus, reducing the number of days of the training set by the clustering approach can lead to poorer forecasting performance.
Moreover "Walk Forward SVR single loop" reaches lower errors than "Walk Forward SVR all loops" options. Including data from other loops does not improve the performance of the model.

day	time	WF SVR single loop $\hat{E}^{\text{speed}}$	WF SVR all loops with clustering $\hat{E}^{\text{speed}}$	WF SVR all loops $\hat{E}^{\text{speed}}$
Mon, 20.01.2020	7:00-8:54 am	**6.5444**	8.8612	6.7655
Wed, 22.01.2020	17:00-18:54 pm	**6.1125**	7.0318	6.2553
Fr, 24.01.2020	17:00-18:54 pm	**8.5847**	9.3787	9.4734
Thu, 27.02.2020	7:00-8:54 am	**3.3779**	4.2912	3.5198
Wed, 22.07.2020	17:00-18:54 pm	**6.9624**	9.2092	7.2292

Table 3: 2 hour prediction errors: one step speed predictions with Walk Forward SVR approaches.

Figures 28 and 29 represent speed predictions for the north direction on 22.07.2020 and 20.01.2020. The "Walk forward SVR all loops with clustering" prediction fluctuates more around the ground truth values, probably due to a lower number of samples used to build the model. This behaviour is more noticeable in Figure 28 for Cimiez Nord, Philippe Nord and Grinda loops.

For "Walk forward SVR single loop" and "Walk forward SVR all loops" predictions we observe the same tendency: there is a delay in the prediction at the beginning and at the end of more congested phases. This is observable in Figure 29 for Magnan and Grinda loops. To reduce the impact of this delay, one can try to reduce the aggregation time (less than 6 minutes), keeping in mind that lower aggregation time can lead to higher errors in the prediction.

3.2 Minnesota data: I-35 north bound direction to Minneapolis

With Minnesota data, we used a resolution of 6 minutes, as for the Voie Mathis data. We study the impact of considering the ramps in the clustering procedure of the flow variable. We observe that the ramps do not change significantly the classification of the days in the train and test sets. For the SVR predictions we implement both one step ("Walk forward SVR single loop" and "Walk forward SVR all loops") and multistep ("SVR single loop" and "SVR all loops") predictions for flow and speed variables. We study the impact of considering all loop detectors in the predictions; the prediction errors do not improve when we consider all loop detectors of the road segment in the prediction of a single loop for both prediction horizons (one step and multistep) and for both variables. For one step and multistep SVR predictions we use the same window size in the past s, equal to 30 minutes, to predict one hour in the future (h equal to 10

Figure 28: Speed predictions on 22/7/2020, "Walk Forward SVR single loop" and "Walk Forward SVR all loops with clustering".

observations). For both one step and multistep SVR the window size s of 30 minutes in the past gives lower errors (lower values of (2.23)) than a window size of 1 hour. We compare also SVR multistep predictions ("SVR single loop" and "SVR all loops") results with the "Classification" approach. For this approach we use a window size in the past s, equal to 2 hours, after having compared its error with a window size s of 1 hour.

3.2.1 Data description

We considered the following RTMC data [14]:

- I-35, north bound direction:
  - 4.85 km long road stretch
  - 5 lanes, $l \in \{1, 2, \ldots, 5\}$
  - 8 aggregate loop detector stations (IDs: S54, S1706, S56, S57, S1707, S59, S60, S61), $d \in \{1, 2, \ldots, 8\}$
  - 2 on-ramps (ID: 129, 130)
  - 3 off-ramps (ID: 169, 170, 171)

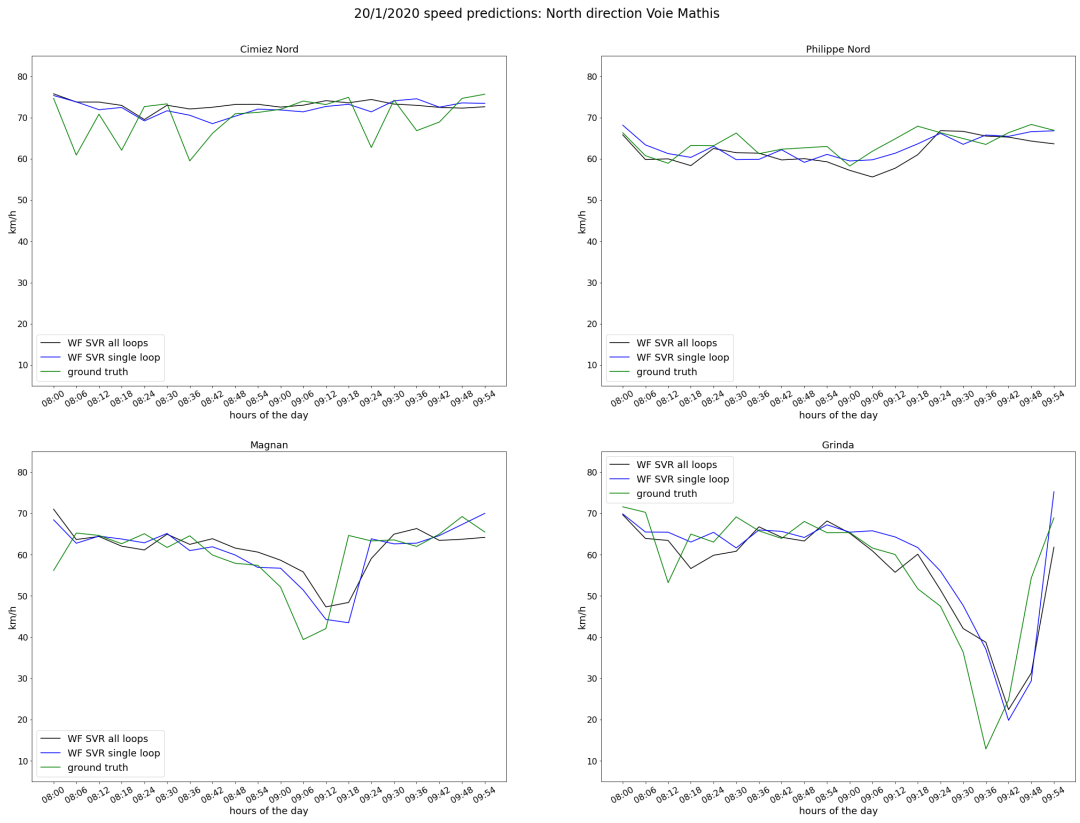


Figure 29: One step speed predictions on 20/1/2020, "Walk Forward SVR single loop" and "Walk Forward SVR all loops".

ID	S54	S1706	169	S56	129	S57	170	1707	S59	130	171	S60	S61
location in km	0	0.75	0.76	1.25	1.85	2.05	2.85	3.15	3.45	3.75	3.95	4.25	4.85

Table 4: Location and arrangement of loop detectors on I-35 north bound direction to Minneapolis: main road loop detectors (black), off-ramps (red), on-ramps (green).

We use the following notations:

- d_l: lane loop detector corresponding to the aggregate loop detector d on lane l,
- $Y_{\mathrm{k}}(x,t) : k \in \{\text{flow } (veh/h), \text{ speed } (km/h), \text{ density } (veh/km)\}$, measurement at time t, for an aggregate loop detector at location x,
- $Y_{\mathrm{k}}^{d_l}(t) : k \in \{\text{flow } (veh/h), \text{ speed } (km/h), \text{ density } (veh/km)\}$, measurement at time t, for a lane loop detector d_l
- $Y_{\mathrm{k}}^{r}(t) : k \in \{\text{flow } (veh/h), \text{ speed } (km/h), \text{ density } (veh/km)\}$, measurement at time t, for a ramp loop detector with ID r

We extract the flow and density data for the lane-loop detectors (in total: 5×8 (main loops) + 5 (ramps) = 45), i.e. $Y_{\text{flow}}^{d_l}(t)$ and $Y_{\text{density}}^{d_l}(t)$.

We apply the following formulas for the computation of the remaining traffic quantities:

$$Y^{d_l}_{\text{speed}}(t) = \frac{Y^{d_l}_{\text{flow}}(t)}{Y^{d_l}_{\text{dens}}(t)},$$

$$Y^{d}_{\text{flow}}(x,t) = \sum_{l=1}^{5} Y^{d_l}_{\text{flow}}(t), \quad Y^{d}_{\text{density}}(x,t) = \sum_{l=1}^{5} Y^{d_l}_{\text{density}}(t)$$

and

$$Y^{d}_{\text{speed}}(x,t) = \frac{1}{Y^{d}_{\text{flow}}(x,t)} \sum_{l=1}^{5} \left(Y^{d_l}_{\text{flow}}(t) \times Y^{d_l}_{\text{speed}}(t) \right)$$

Remark 2. *The computation of the speeds for the aggregate detectors are based on the California Department of Transportation*[10] *computations.*

3.2.2 Data preparation

Following [10], we need to clean our raw-data before using them for clustering. This cleaning-process is done as follow:

For the completion of missing we take in account not only observations in the loop in which we want to have the imputation but also the loops before and after in the road segment. For I-35, the distances between two consecutive loops is lower than the ones for Voie Mathis (see Tables 1 and 2), for this reason we choose to perform the imputation considering the time and the spatial locations.
for a lane loop detector d_l:

$$Y^{d_l}_{k}(t) = \frac{Y^{d_l}_{k}(t-1) + Y^{d_l}_{k}(t+1) + Y^{(d-1)_l}_{k}(t) + Y^{(d+1)_l}_{k}(t)}{4}$$

for a ramp loop detector with ID r, the imputation remains the same as for Voie Mathis data:

$$Y^{r}_{k}(t) = \frac{Y^{r}_{k}(t-1) + Y^{r}_{k}(t+1)}{2}$$

Remark 3. *If the data are missing for more than 2 hours, for any traffic quantity (flow, speed, density) the imputation procedure is not performed.*

For the train set we use data from year 2013 (357 days): 8 days of the 2013 are not used in the training due to continuous missing data for more than 2 hours[11]. For each day, we select only observations from 5:00 am to 10:54 pm, 180 observations with a resolution of 6 minute. For the validation (test set) we use data from 5 weeks of the year 2014 (35 days):

3.2.3 Clustering

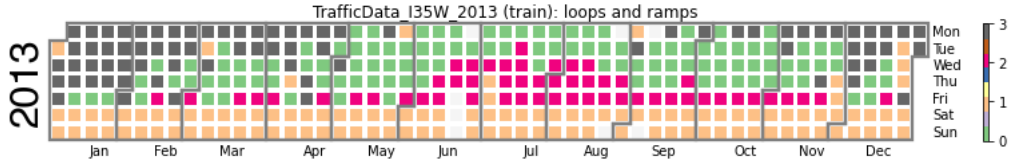
For the clustering procedure we consider two options:

- Option A : flow values for all loops and ramps, the train set is a 3-D array of size $[357, 180, 13]$

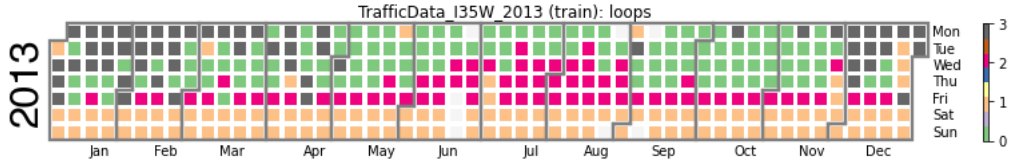
[10] https://pems.dot.ca.gov/
[11] Days with missing values for more than 2 hours: 21/06, 22/06, 23/06, 24/06, 25/08, 26/08, 08/09, 09/09.

- Option B: flow values for all loops (no ramps), the train set is a 3-D array of size $[357, 180, 8]$

Based on the clustering results we decide for 4 clusters (increasing to five clusters makes a cluster disappear in the test set for both options).
Figure 30 shows the classification of the days in the train set with respect to the flow variable for options A and B. The parameter γ is set to 164 for option A and to 89 for option B. The classification of the train days for both options is almost the same. In general, the ramps do not change the allocation of the days inside each cluster and the path of the centroids of the loops. The cluster $k = 1$ represents Saturdays, Sundays and Public Holidays[12]. The cluster $k = 2$ identifies Fridays throughout the year and also Wednesdays and Thursdays during summer months (June, July and August). The cluster $k = 3$ portrays Mondays and some others days (mainly Tuesdays and Wednesdays) during winter months (From January to March and December). Cluster $k = 0$ locates mainly the first part of the week in May, September, October and November and the remaining days not covered by the other clusters.



(a) Option A



(b) Option B

Figure 30: Clustering results for the train set for options A and B.

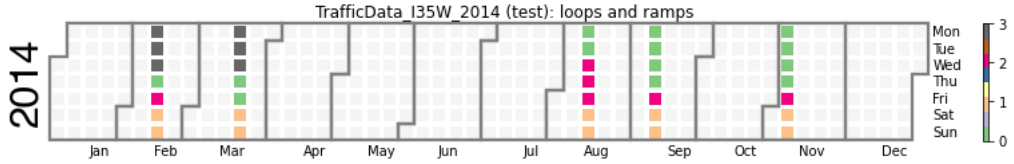
For the test set, we used the following weeks:

- Mon, 10.02. – Sun, 16.02.
- Mon, 17.03. – Sun, 23.03.
- Mon, 11.08. – Sun, 17.08.
- Mon, 08.09. – Sun, 14.09.
- Mon, 03.11. – Sun, 09.11.

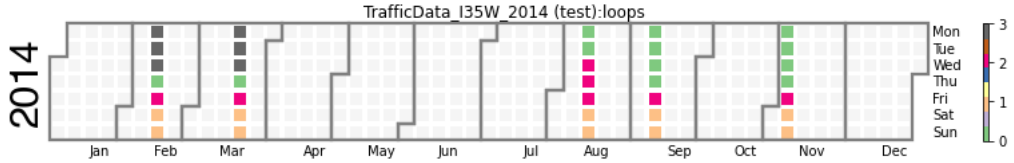
Figure 31 shows the classification of the days in the test set for options A and B. The classification of the days in option A and option B differ only for one day Friday 21.03. For both options, the classification of the test days is coherent with the one in the train set. Cluster $k = 1$ identifies the weekends. Cluster $k = 2$ represents Fridays for all weeks and Wednesday and Thursday in the August week. Cluster $k = 3$ portrays Mondays, Tuesdays, Wednesdays in the winter weeks of February, March and cluster $k = 0$ detects the remaining days. We can conclude that the

[12]01.01.2013 First of the year, 27.05.2013 Memorial Day, 04.07.2013 Independence Day, 02.09.2013 Labor Day, 28.11.2013 Thanksgiving Day, From 23-12-2013 to 31-12-2013 Christmas Holidays.

ramps do not change the classification of the days in the test set, they do not add information in describing the traffic on the road segment with respect to the flow variable.



(a) Option A



(b) Option B

Figure 31: Clustering results for the test set for options A and B.

3.2.4 Prediction days and time:

The days selected for the predictions present different level of congestion regimes in the time slot selected, but none of them presents a free flow situation for every loop of the road segment. For all predictions approaches we compare the errors for 1 hour in the future (10 observations).

- Wed, 10.09.2014: 8:00am - 8:54am ;
- Sat, 22.03.2014: 4:30pm - 5:24pm;
- Fr, 15.08.2014: 2:30pm - 3:24pm;
- Wed, 12.02.2014: 8:00am - 8:54am.

The selected test days belong to different clusters as we can see in Figure 31.

Remark 4. *The train set for the predictions is the same of the clustering. For the prediction using all loops a 3-D array of size* $[357, 180, 13]$ *and for prediction using only the loop that we want to predict a 2-D array of size* $[357, 180]$ *since the last dimension is equal to 1, just one single loop.*

3.2.5 Speed predictions

Table 5 shows the errors of one step speed predictions: "Walk forward SVR all loops" and "Walk forward SVR single loop". The prediction errors of "Walk Forward SVR single loop" are lower than the errors of "Walk Forward SVR all loops". By looking at these results, we can claim that, for one step speed predictions, including also other loops of the road segment in the prediction of a single loop does not improve the prediction errors. We reach lower errors only using data from the loop that we want to predict.

day	time	WF SVR all loops $\hat{E}^{\text{speed}}$	WF SVR single loop $\hat{E}^{\text{speed}}$
Wed, 10.09.2014	8:00-8:54 am	11.3115	**7.91**
Sat, 22.03.2014	16:30-16:24 pm	12.0235	**6.8839**
Fr, 15.08.2014	14:30-15:24 pm	13.5737	**11.4872**
Wed, 12.02.2014	8:00-8:54 am	8.6001	**5.6885**

Table 5: 1 hour prediction errors : one step predictions ("Walk Forward SVR all loops" and "Walk Forward SVR single loop").

Figure 32 shows the one step speed predictions on each loop of the road segment for the 12.02 time slot. As showed in Table 5, "Walk Forward SVR single loop" predictions are better than "Walk Forward SVR all loops" predictions. In particular "Walk Forward SVR all loops prediction" lines for loops S1707, S59, S61 are not always close to the ground truth lines. Even if "Walk Forward SVR single loop" predictions have lowest errors there is a delay in the predictions with respect to the ground truth values especially when the speed starts to increase or decrease . This behaviour was also observed with Voie Mathis Data in Figure 29, where the predicted values are shifted ahead with respect to the ground truth values.

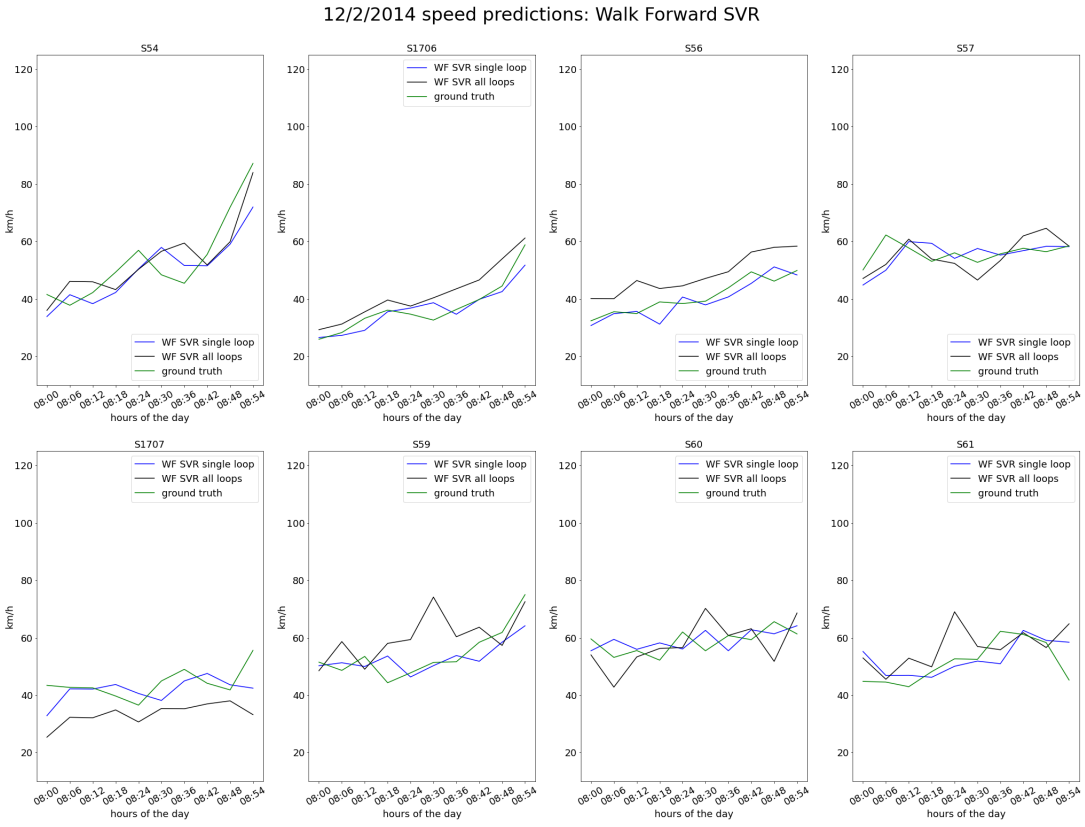


Figure 32: Speed predictions on 12/2/2014, one step Walk Forward SVR.

Table 6 shows the errors of the multistep speed predictions. "SVR single loop" errors are

smaller than "SVR all loops"; except for one day, Wednesday 15.08. Also the "Classification" approach reaches lower errors than "SVR all loops"; except for Wednesday 15.08. In general, we can claim that, for SVR multistep speed predictions, including data from other loops in the road segment does not improve the errors. We have only one day that performs better when also other loops are included. "Classification" approach gives promising results for the days selected. We remark that this is an empirical method based on the selection, from the train set, of the closest day to the target. Thus, if the type of congestion that we want to predict is similar to an old one included in the train set, the prediction errors for the "Classification" approach can be lower than the predictions errors of SVR (see Table 6: Saturday 22.03).

day	time	SVR single loop $\hat{E}^{\text{speed}}$	SVR all loops $\hat{E}^{\text{speed}}$	Classification $\hat{E}^{\text{speed}}$
Wed, 10.09.2014	8:00-8:54 am	**9.887**	13.4041	11.0031
Sat, 22.03.2014	16:30-16:24 pm	13.9687	18.663	**8.8517**
Fr, 15.08.2014	14:30-15:24 pm	27.8764	**23.3976**	26.7894
Wed, 12.02.2014	8:00-8:54 am	**8.7275**	10.5186	9.4164

Table 6: 1 hour speed prediction errors: multistep predictions ("SVR single loop", "SVR all loops", "Classification" approach.

Figure 33 shows the multistep speed predictions on each loop of the road segment for the 12.02 time slot. In general, we can observe that the "multistep SVR" prediction lines ("SVR single loop" and "SVR all loops") are more flat than "Walk Forward SVR" prediction lines presented in Figure 32. Since for the "Walk Forward" approaches the train set is iteretivaly uploaded, we are able to reach higher accuracy than with "multistep SVR", where instead the train set remains fixed and it is used multiple times for each point that we would like to predict in the future. For Wednesday 12.02 the error of the "Classification" approach is lower than "SVR all loops". The "Classification" approach selects the closest day to the target by considering all the loops in the road segment, thus speed predictions can be better for some loops than others. In Figure 33, the "Classification" approach works better for S54,S170 loops predictions than for S59 loop prediction.

3.2.6 Flow predictions

Table 7 shows the errors of one step flow predictions: "Walk forward SVR all loops" and "Walk forward SVR single loop". Both options register similar errors for the days selected except for Wednesday 12.02 for which the error of "Walk Forward SVR single loop" is lower than "Walk Forward SVR all loop". Thus, including other loops of the road segment does not improve the one step flow predictions.

Figure 34 shows the one step flow predictions on each loop of the road segment for the Wednesday 10.09 time slot. "Walk Forward SVR single loop" and "Walk Forward all loop" prediction lines appear similar. For both prediction lines we can recognize the delay in the prediction more visible for loops S1707 and S59 where the flow peaks at 8:24 are detected at 8:30.

Table 8 shows the errors of multistep flow predictions. "SVR all loops" and "SVR single loop" present similar errors, except for Friday 15.08 when "SVR all loops" performs better than "SVR single loop". The "Classification" approach gives lower errors than both SVR approach for two days: Wednesday 10.09 and Wednesday 12.02. Instead for Friday 15.08 the predictions errors

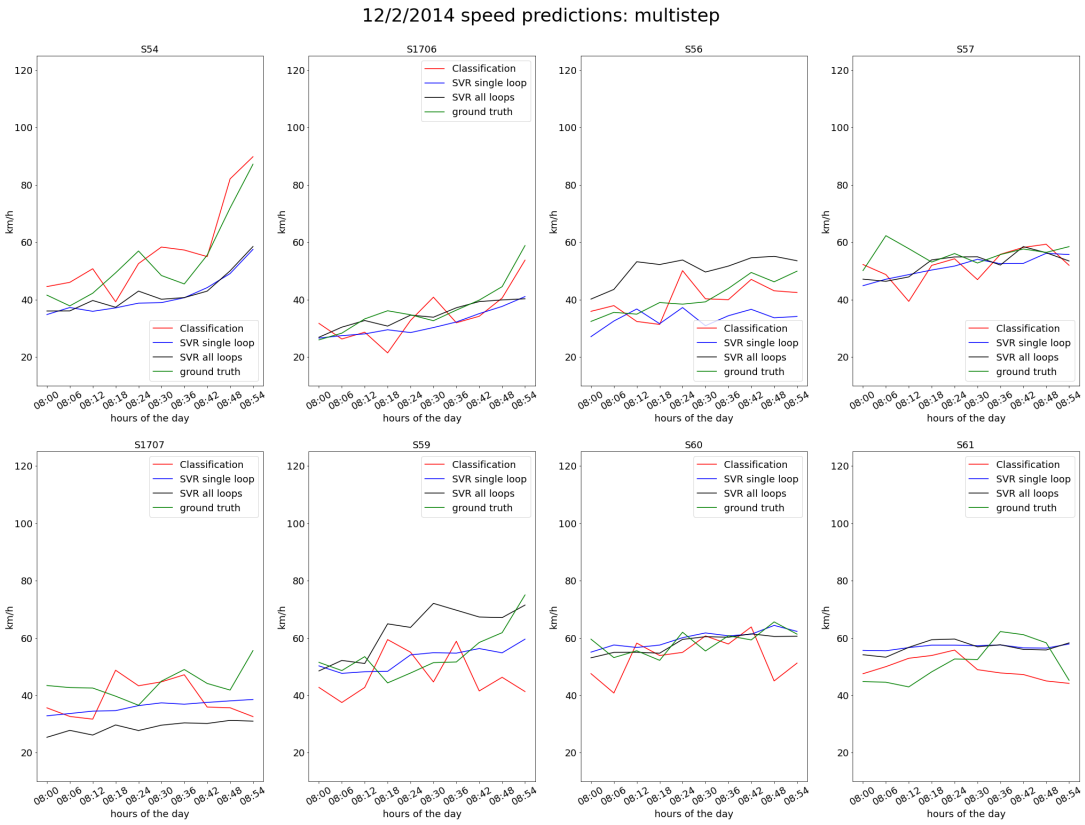


Figure 33: Multistep speed predictions on 12/2/2014.

is the highest one. In general, we can say that "SVR all loops" does not improve significantly the predictions errors with respect to "SVR single loop". The "Classification" approach reaches promising results but, if the type of congestion that we want to predict is never been detected in the train set, the prediction errors for the "Classification" approach can be higher than the predictions errors of SVR (see Table 8: Friday 15.08).

day	time	SVR single loop $\hat{E}^{\text{flow}}$	SVR all loops $\hat{E}^{\text{flow}}$	Classification $\hat{E}^{\text{flow}}$
Wed, 10.09.2014	8:00-8:54 am	513.722	521.8898	**438.7639**
Sat, 22.03.2014	16:30-16:24 pm	**319.7686**	320.8227	400.7524
Fr, 15.08.2014	14:30-15:24 pm	530.0597	**470.0556**	602.0372
Wed, 12.02.2014	8:00-8:54 am	450.3737	426.837	**361.7751**

Table 8: Prediction error for flow 1 hour: multistep prediction, SVR single loop, SVR all loops, Classification.

Remark 5. *-For speed and flow variables-*
Despite "SVR all loops" and "Walk Forward SVR all loops" (equivalently "SVR single loop" and "Walk Forward SVR single loop") have the same input data for the first step, when the prediction

day	time	WF SVR all loops $\hat{E}^{\text{flow}}$	WF SVR single loop $\hat{E}^{\text{flow}}$
Wed, 10.09.2014	8:00-8:54 am	**401.754**	409.9089
Sat, 22.03.2014	16:30-16:24 pm	369.161	**354.3346**
Fr, 15.08.2014	14:30-15:24 pm	**471.0098**	474.672
Wed, 12.02.2014	8:00-8:54 am	332.2375	**299.7833**

Table 7: 1 hour flow prediction errors: one step predictions ("Walk Forward SVR all loops" and "Walk Forward SVR single loop").

origin t is the same, the predictions can be different. This happen because the modality used to choose the hyperparameters ϵ and C is not the same. For multistep SVR we used a pipeline, thus the choice of the hyperparameters, based on the Gridsearch, takes in account not only one observation in the future as Walk forward SVR, but h observations in the future.

4 Conclusions and outlooks

In this project we have introduced machine learning tools in order to study and infer road traffic time series.
Both Dynamic Time Warping and soft-Dynamic Time Warping are proposed as alternative to the Euclidean distance respectively for multistep predictions and clustering of multivariate time series. We show that soft-DTW regularization is particularly well suited to average and cluster time series. The adopted clustering procedures identify different seasonal and weekly traffic paths according to the variable used. For instance, flow and occupancy rate variables classify coherently test days even if the train and test sets belong to different years. The speed variable shows a different behaviour, we do not have a proper systematic relation between the classification of the days in the train and test sets. This may be due to the variability of the speed measurements, the granularity of the data or a change in the driving style which might differ from year to year (i.e. Covid-19). For the flow variable, we show that including the ramps in the clustering does not change the description of the traffic in the road segment.

For future work, it will be interesting to study the relation between the number of clusters and the number of loops considered. By increasing the number of loops in the clustering procedure, treating a longer road segment, deleting days with exogenous events (i.e. accidents, stationary vehicles), we can separate each day of the week in its own cluster.

In this study, Radial Basis Function (RBF) is used as a kernel function for Support Vector Regression (SVR) models. (RBF) kernel function is efficient for traffic speed and flow predictions. The proposed SVR approaches is found to be suitable and useful in real-world operations for both one step and multistep predictions. They fit congested situations, which are more challenging than free flow situations. We try to look at time series models that incorporate all loops in the road segment, instead of only the data from the single loop that we want to predict. We find out that, for both prediction horizons, the speed and flow errors are not significantly improved when we use data from all loop detectors of the road segment. We prove that, for "Walk forward SVR all loops" speed predictions, reducing the number of days considered in the training phase with the clustering procedure, before support vector regression, does not increase the accuracy of the predictions. In general, one step predictions are more accurate than multistep predictions. For the multistep predictions also the "Classification" approach, with Dynamic Time Warping, reaches promising results only using historical data from the train set.

Speed and flow predictions can be used for travel time predictions and for the calibration

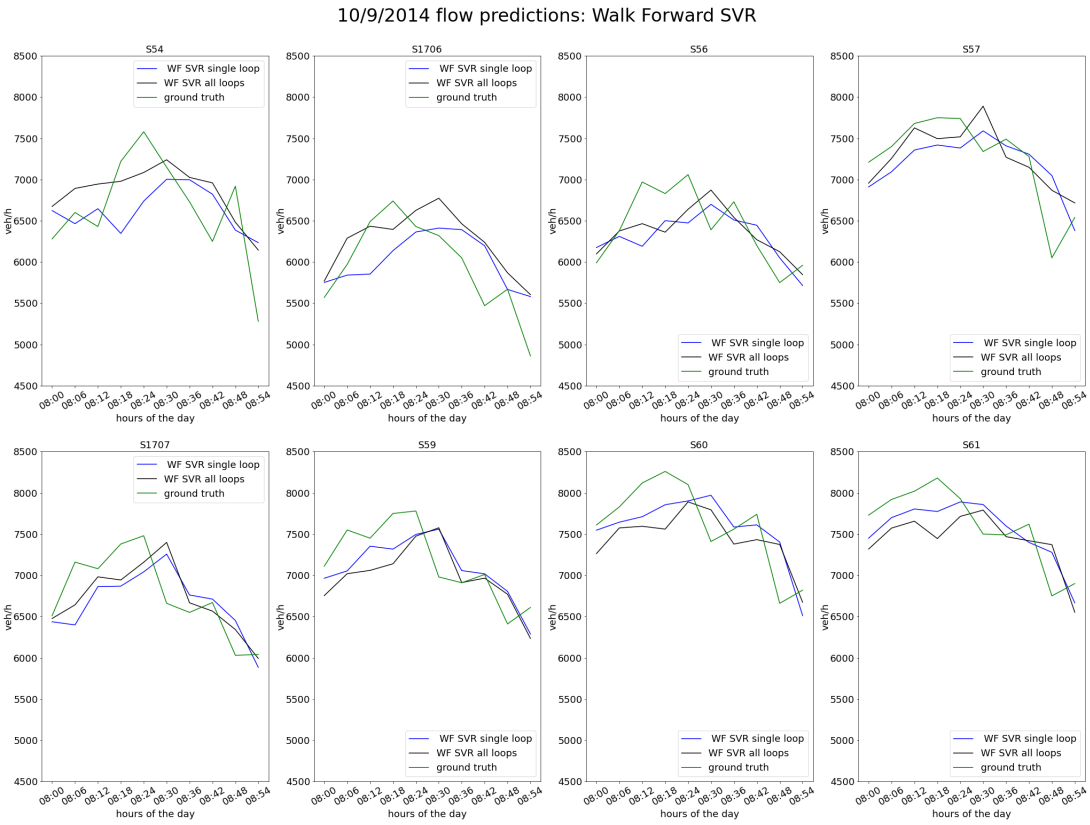


Figure 34: Flow predictions on 12/2/2014, one step Walk Forward SVR.

of macroscopic models. In order to increase the accuracy, two further strategies can be investigated. First, reducing the granularity of data (now fixed at 6 minutes) we add more information and probably we are able to reach lower prediction errors and mitigate the delay in one step predictions. Second, the train and test sets belong to different years, with a significant temporal jump between the two sets. By reducing the temporal jump between train and test, including all days before the target day we want to predict, we could be able to improve the predictions.

References

[1] M. Blondel, A. Mensch, and J.-P. Vert. Differentiable divergences between time series. In International Conference on Artificial Intelligence and Statistics, pages 3853–3861. PMLR, 2021.

[2] M. Castro-Neto, Y.-S. Jeong, M.-K. Jeong, and L. D. Han. Online- SVR for short-term traffic flow prediction under typical and atypical traffic conditions. Expert systems with applications, 36(3):6164–6173, 2009.

[3] C.-C. Chang and C.-J. Lin. LIBSVM: a library for support vector machines. ACM transactions on intelligent systems and technology (TIST), 2(3):1–27, 2011.

[4] C. Cortes and V. Vapnik. Support-vector networks. Machine learning, 20(3):273–297, 1995.

[5] M. Cuturi. Positive definite kernels in machine learning. arXiv preprint arXiv:0911.5367, 2009.

[6] M. Cuturi. Fast global alignment kernels. In Proceedings of the 28th international conference on machine learning (ICML-11), pages 929–936, 2011.

[7] M. Cuturi and M. Blondel. Soft-dtw: a differentiable loss function for time-series. In International Conference on Machine Learning, pages 894–903. PMLR, 2017.

[8] G. Gordon and R. Tibshirani. Karush-kuhn-tucker conditions. Optimization, 10(725/36):725, 2012.

[9] R. Koggalage and S. Halgamuge. Reducing the number of training samples for fast support vector machine classification. Neural Information Processing-Letters and Reviews, 2(3):57–65, 2004.

[10] J.-M. Loubes, E. Maza, M. Lavielle, and L. Rodriguez. Road trafficking description and short term travel time forecasting, with a classification method. Canadian Journal of Statistics, 34(3):475–491, 2006.

[11] J.-M. Loubesi, É. Maza, M. Lavielle, and L. Rodriguez. Road trafficking description and short term travel time forecasting, with a classification method. Canadian Journal of Statistics, 34(3):475–491, 2006.

[12] Métropole Nice Côte D'Azur. Open data de Nice Côte d'Azur. Website: `http://opendata.nicecotedazur.org/site/`.

[13] Z. Mingheng, Z. Yaobao, H. Ganglong, and C. Gang. Accurate multisteps traffic flow prediction based on SVM. Mathematical Problems in Engineering, 2013, 2013.

[14] Minnesota Departement of Transportation. Mn/Dot Traffic Data. Website: `http://data.dot.state.mn.us/datatools/`.

[15] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, et al. Scikit-learn: Machine learning in python. the Journal of machine Learning research, 12:2825–2830, 2011.

[16] P. J. Rousseeuw. Silhouettes: a graphical aid to the interpretation and validation of cluster analysis. Journal of computational and applied mathematics, 20:53–65, 1987.

[17] H. Sakoe and S. Chiba. Dynamic programming algorithm optimization for spoken word recognition. IEEE transactions on acoustics, speech, and signal processing, 26(1):43–49, 1978.

[18] A. Sardá-Espinosa. Comparing time-series clustering algorithms in r using the dtwclust package. `https://cran.r-project.org/web/packages/dtwclust/`, 2017.

[19] A. J. Smola and B. Schölkopf. A tutorial on support vector regression. Statistics and computing, 14(3):199–222, 2004.

[20] I. Syarif, A. Prugel-Bennett, and G. Wills. SVM parameter optimization using grid search and genetic algorithm to improve classification performance. Telkomnika, 14(4):1502, 2016.

[21] R. Tavenard, J. Faouzi, G. Vandewiele, F. Divo, G. Androz, C. Holtz, M. Payne, R. Yurchak, M. Rußwurm, K. Kolar, et al. Tslearn, a machine learning toolkit for time series data. Journal of Machine Learning Research, 21(118):1–6, 2020.

[22] U. Thissen, R. Van Brakel, A. De Weijer, W. Melssen, and L. Buydens. Using support vector machines for time series prediction. Chemometrics and intelligent laboratory systems, 69(1-2):35–49, 2003.

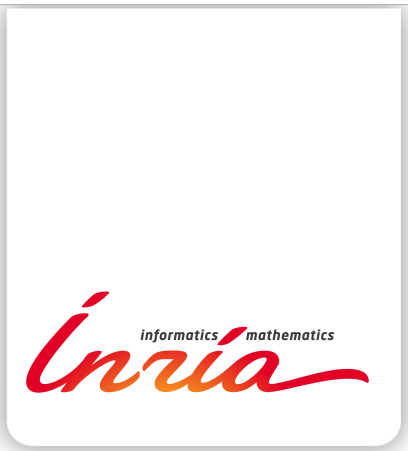
[23] M. Treiber and A. Kesting. Traffic flow dynamics: data, models and simulation. Physics Today, 67(3):54, 2014.

[24] I. W. Tsang, J. T. Kwok, and P.-M. Cheung. Core vector machines: Fast SVM training on very large data sets. Journal of Machine Learning Research, 6(13):363–392, 2005.

[25] V. Tyagi, S. Kalyanaraman, and R. Krishnapuram. Vehicular traffic density state estimation based on cumulative road acoustics. IEEE Transactions on Intelligent Transportation Systems, 13(3):1156–1166, 2012.

[26] C.-H. Wu, J.-M. Ho, and D.-T. Lee. Travel-time prediction with support vector regression. IEEE transactions on intelligent transportation systems, 5(4):276–281, 2004.

[27] Y. Zhang, T. Bouadi, and A. Martin. An empirical study to determine the optimal k in EkNNclus method. In International Conference on Belief Functions, pages 260–268. Springer, 2018.



RESEARCH CENTRE
SOPHIA ANTIPOLIS – MÉDITERRANÉE

2004 route des Lucioles - BP 93
06902 Sophia Antipolis Cedex

Publisher
Inria
Domaine de Voluceau - Rocquencourt
BP 105 - 78153 Le Chesnay Cedex
inria.fr

ISSN 0249-6399