MASTER

Multiple ASpect TrajEctoRy management and analysis

Project Acronym	***MASTER***
Project Full Name	***M**ultiple **AS**pects **T**raj**E**cto**R**y management and analysis*
Project Number	***777695***
Deliverable Title	*Preliminary software prototypes*
Deliverable No.	*D5.3*
Contract Delivery Date	*28/02/2023 (M60)*
Actual Delivery Date	*28/02/2023 (M60)*
Responsible Authors	*Chiara Renso (CNR)*
	This project has received funding from the European Union's Horizon 2020 research and innovation programme under the Marie-Sklodowska Curie grant No 777695

DOCUMENT INFORMATION

GRANT AGREEMENT N.	777695
PROJECT ACRONYM	MASTER
PROJECT FULL NAME	Multiple Aspects Trajectory Management and Analysis
STARTING DATE (DUR.)	01/03/2018 (70 months)
ENDING DATE	31/12/2023
PROJECT WEBSITE	http://www.master-project-h2020.eu
COORDINATOR	Chiara Renso (CNR)
WORKPACKAGE N. | TITLE | START – END MONTH	WP5 | Application Scenarios | M1 – M70
WORKPACKAGE LEADER	UPRC
DELIVERABLE N. | TITLE	D5.3 | ***Preliminary software prototypes***
RESPONSIBLE AUTHOR	Chiara Renso (Unit Manager of CNR)
DATE OF DELIVERY (CONTRACTUAL)	28 February 2023 (M60)
DATE OF DELIVERY (SUBMITTED)	28 February 2023 (M60)
VERSION | STATUS	1.0
NATURE	REPORT
DISSEMINATION LEVEL	PUBLIC
AUTHORS (PARTNER)	UPRC, CNR, UNIVE, HUA, UVSQ

CONTRIBUTORS	Alessandra Raffaetà (UNIVE), Giulia Rovinelli (UNIVE), Chiara Pugliese (CNR) Chiara Renso (CNR), Raffaele Perego (CNR), Lorenzo Gabrielli (CNR), Yannis Kountopolous (HUA), Konstantinos Tserpes (HUA), Michela Rial (CNR), Iulian Sandu Popa (UVSQ), Karine Zeitouni (UVSQ), Marta Simeoni (UNIVE), Andrea Marin (UNIVE)

ACRONYM LIST

MASTER	Multiple Aspects Trajectory Management and Analysis
ICT	Information and Communication Technologies
ISTI	Istituto di Scienza e Tecnologie dell'Informazione "A. Faedo"
CNR	Consiglio Nazionale delle Ricerche
UNIVE	Ca' Foscari University of Venice
UVSQ	University of Versailles Saint-Quentin
UPRC	University of Piraeus Research Center
UFC	Federal University of Ceará
HUA	Harokopio University of Athens
PUC	Pontificial University of Rio de Janeiro
DAL	Dalhousie University
THIRA	Municipality of Thira
ER	Experienced Researcher
ESR	Early Stage Researcher
ACTV	Azienda del Consorzio Trasporti Veneziano (Company responsible for public transportation in province of Venice)
AIS	Automatic Identification System (AIS)
ARIMA	Autoregressive integrated moving average

TABLE OF CONTENTS

1. INTRODUCTION

The objective of this deliverable, that is linked to WP5, is "to *revise the requirements, the application questions and the information about the data available for the three scenarios*" as stated in the MASTER Grant Agreement Annex 1 Part A pages 18-19. This activity has been carried out during secondments linked to WP5 and parallel activities of project partners. We have executed a total from the start of the project of 19,73 person months during which secondees have refined requirements and information about the datasets available at the partners relative to the three application scenarios: tourism, sea monitoring and transportation.

WP5 has the following four objectives:

1. **to understand the application needs and perform the requirements analysis driving the design of the methods developed in WP3 and WP4.** Activities to reach this objective have been mainly performed during the second and third year, by secondments to PUC, DAL and THIRA that confirmed the requirements exposed during the first and second year and focused on methods closer related to application prototypes. All planned tasks related to the period for the fulfilment of this objective, have been fully achieved.

2. **to host the interaction between academic and non-academic partners to create awareness and best practices on the non-academic world needs, thus increasing the potential in career developing especially in ESRs.** Activities to reach this objective have been carried out during the first, second and third year, with secondments to Thira. All planned tasks related to the period for the fulfilment of this objective, have been fully achieved with remote interactions with Thira for the application questions;

3. **to test the developed techniques in real-world scenarios possibly on data owned by the non-academic partner and on the basis of their actual daily problems.** Activities to reach this objective have been performed during secondments to DAL, THIRA and PUC. All planned tasks related to the period for the fulfilment of this objective, have been fully achieved.

4. **to develop software prototypes to facilitate the interaction and transfer of knowledge among academic and non-academic partners.** Activities for the fulfilment of this objective are still ongoing and concern development of research prototypes to be reported in this deliverable (D5.3) and in D5.4 due at M60 and M70, respectively.

Deliverables D5.1 and D5.2, already submitted, fulfill the first, second and third objectives w.r.t. to the progress towards the goals of **WP5**. Deliverables D5.3 and D5.4 will provide the preliminary and the final reports on application scenarios and software prototypes, respectively.

The current deliverable presents three software prototypes developed in the three different application scenarios.

From M24 to M60 the number of secondments, or splits, linked to WP5 is 10, as listed below in Table 1. According to the Researcher Declarations submitted to SyGMa system, the total number of person months for this period is 6,07.

The outcome of this deliverable is to feed Deliverable D5.4 "Final Report on application scenarios and Software prototypes" due at M70.

Table 1: Secondments executed from M24 to M60 linked to WP5 Application scenarios from the SyGMa system

RD N.	Second. N	Secondee Name	Fellow ID	Sending Institution	Hosting Institution	From	To	PM	Task
41	61	Zaineb Chelly Dagdia	29	UVSQ	THIRA	15/05/2022	29/05/2022	0.5	T5.1
10	26	Marta Simeoni	12	UNIVE	DAL	08/08/2022	02/09/2022	0.87	T5.2
28	33	Andrea Marin	10	UNIVE	PUC	29/07/2022	29/08/2022	1.03	T5.3
3	3	Raffaele Perego	3	CNR	THIRA	01/10/2022	08/10/2022	0.27	T5.1
4	4	Chiara Renso	4	CNR	THIRA	01/10/2022	08/10/2022	0.27	T5.1
50	115	Iulian Sandu Popa	28	UVSQ	THIRA	29/04/2022	28/05/2022	1	T5.1
51	27	Alessandra Raffaetà	11	UNIVE	DAL	29/07/2022	29/08/2022	1.03	T5.2
6	23	Andrea Marin	10	UNIVE	THIRA	25/09/2022	06/10/2022	0.40	T5.1
5	52	Karine Zeitouni	5	UVSQ	THIRA	02/10/2022	08/10/2022	0.23	T5.1
12	54	Karine Zeitouni	5	UVSQ	PUC	17/12/2022	30/12/2022	0.47	T5.3

2. TOURISM SCENARIO

This chapter of the deliverable is related to the activity of Task 5.1. about Tourism application scenario, whose responsible is CNR. We report here about the software developed based on the D5.1, D.52 and research and development activities carried out by partners. The secondments related to this task are reported in Table 1 and have a total of 2,67 PMs.

APPLICATION QUESTIONS

As reported in the Deliverable D5.1, the main question that Thira Municipality wants to answer is: *How can we monitor the tourism flow and support the decision-making process*?
We studied the proposed questions after having explored the available datasets and they have been used to plan the prototype.

The first version of the prototype will include questions **a)**, **b)** and **d)**, while we leave the remaining **c)** and **e)** to the final version in Deliverable D5.4 due at M70.

a) Which is the origin Country of travelers?

b) How long do the tourists stay in Santorini?

c) Are the first-time visitors and returning visitors behaving differently?

d) **Can we predict the tourist arrival flows**?

e) Which is the qualitative level of satisfaction / dissatisfaction of the user?

THE SOFTWARE PROTOTYPE

The software prototype has been developed using Python[1] to analyse the following datasets: Twitter posts, Flights arrivals and departures, OpenStreetMap all described in previous deliverable D5.1 and Airbnb dataset described here.

The software allows the user to answer question **a)** analysing the languages of Twitter posts and the home countries of users; question **b)** analysing the distribution of the minimum number of nights people have to stay for each room registered on Airbnb and the mean of duration of OpenStreetMap trajectories; question **d)** analysing the flow of flight arrivals and departures to and from Santorini Island (showing a comparison with Greek islands that are similar to Santorini).

To answer research question **a**), we selected from the dataset the tweets that were geolocated in the Santorini area or contained the word "Santorini". We conducted two types of analysis to identify the country of origin of the users who authored the selected tweets. The first analysis concerns the frequency distribution of the languages used to write the tweets (when stated). With the second analysis, we observed the frequency distribution of the stated country of origin in the profiles of the users who authored the previously selected tweets. In addition, we also analyzed the most popular types of places where users registered (e.g., point of interest, city, or country).

To answer research question **b**), we analysed the frequency of the minimum number of nights required to book a room or a flat in Santorini. We confirmed our results by analyzing how OSM trajectories are distributed over time by grouping them by users.

To answer research question **d**), we studied the arrival and departure trends of domestic and international flights. Already from a first reading of the data, we can see that the number of passengers has been increasing in recent years and that it follows a seasonal pattern (peaks always occur during the summer months). To predict the number of arrivals, we used Seasonal Autoregressive Integrated Moving Average (SARIMA) that is an extension of ARIMA (Autoregressive integrated moving average) that explicitly supports the forecasting of univariate time series data with a seasonal component.

The software prototype can be found at the Github: https://github.com/chiarap2/MASTER

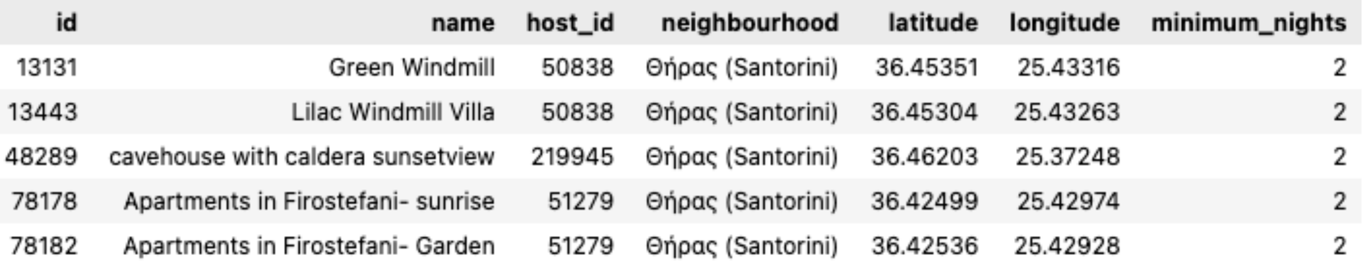
DATASET

The datasets we analyzed to answer questions **a)**, **b)** and **d)** are as follows:

- dataset of tweets containing the word "Santorini" and geolocated in Santorini (dataset specifications are described in Deliverable 5.1);
- dataset on the number of passengers arriving and departing to and from Santorini (dataset specifications are described in Deliverable 5.1);
- dataset of room and/or apartment listings on Airbnb, described here below.

[1] https://www.python.org/

The Airbnb dataset contains information regarding all rooms (and apartments) registered on the platform on 30/09/2019. An example of this dataset is shown in Fig. 1.



id	name	host_id	neighbourhood	latitude	longitude	minimum_nights
13131	Green Windmill	50838	Θήρας (Santorini)	36.45351	25.43316	2
13443	Lilac Windmill Villa	50838	Θήρας (Santorini)	36.45304	25.43263	2
48289	cavehouse with caldera sunsetview	219945	Θήρας (Santorini)	36.46203	25.37248	2
78178	Apartments in Firostefani- sunrise	51279	Θήρας (Santorini)	36.42499	25.42974	2
78182	Apartments in Firostefani- Garden	51279	Θήρας (Santorini)	36.42536	25.42928	2

Figure 1. A screenshot of the Airbnb dataset.

In this dataset we have information about the identifier and name of the room or apartment listing, the identifier of the host, the name of the location, the geographical coordinates (latitude and longitude) of the location and the minimum number of nights required to book the room or apartment.

These datasets are not made available in the public repository due to privacy and confidentiality constraints as already specified in D5.1 and D5.2. However, we show below examples of the results of running the prototype un such datasets.

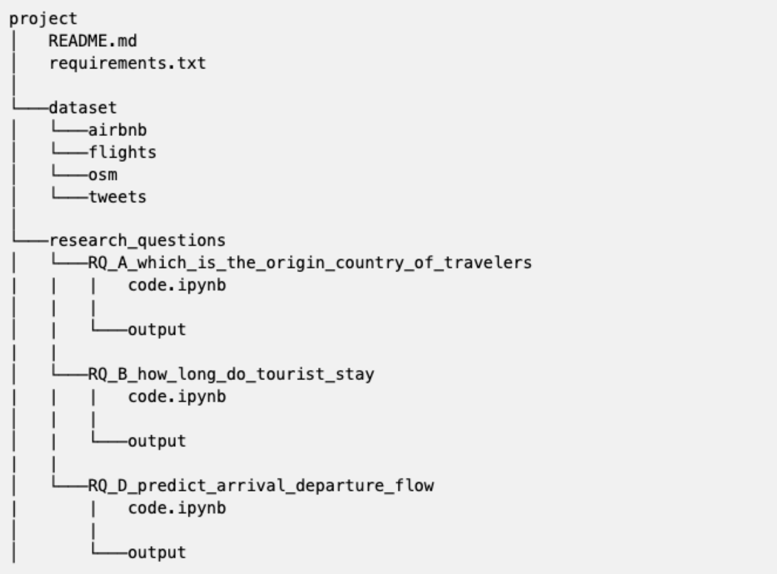
INSTALLATION INSTRUCTIONS

To visualize the code entirely and interact with plots, you can browse the repository following this link: https://nbviewer.org/github/chiarap2/MASTER/tree/master/.
To install the prototype and run it locally, you can download the repository and install the requirements with the following commands.

```
$ pip install -r requirements.txt
```

HOW TO USE THE PROTOTYPE

In order to visualize the results of our analysis, it is possible to browse the Github repository (https://github.com/chiarap2/MASTER). The repository is organized as in Fig. 2.



```
project
│   README.md
│   requirements.txt
│
└───dataset
│   └───airbnb
│   └───flights
│   └───osm
│   └───tweets
│
└───research_questions
│   └───RQ_A_which_is_the_origin_country_of_travelers
│   │   │   code.ipynb
│   │   │
│   │   └───output
│   │
│   └───RQ_B_how_long_do_tourist_stay
│   │   │   code.ipynb
│   │   │
│   │   └───output
│   │
│   └───RQ_D_predict_arrival_departure_flow
│       │   code.ipynb
│       │
│       └───output
│
```

Figure 2. The folder structure of MASTER repository on Github.

In the `dataset` folder there is one subfolder for each dataset needed for the analysis. In the `twitter` folder, you can directly insert files from the Twitter streaming API (we reall that the tweets cannot be made public due to the Terms of Use). In the `airbnb` folder, you can insert an airbnb.csv. In the `osm` folder, you can insert the `santorini.gpx` file containing the trajectories from OpenStreetMap API (you can download it from JOSM application - https://josm.openstreetmap.de/, selecting the bounding box of Santorini). In the `flights` folder, you can insert files containing the information about the number of domestic and international passengers per month and year.

In the `research_questions` folder, you can find a folder for each of the five questions described above. In each of these folders, you find a file with the `.ipynb` extension (a Python notebook) in which we analyse the datasets described above and an `output` subfolder with the results (represented with both plots and description files).

For example, for question **d)**, we can visualize the plot of the comparison between arrivals and departures of Santorini and the other selected islands (Fig. 3) by downloading the image in the output folder of the Github repository[2] or by visualizing it on the nbviewer website[3].

[2] https://github.com/chiarap2/MASTER/blob/master/research_questions/RQ_D_predict_arrival_departure_flow/output/domestic_flights_arr_dep.png

[3] https://nbviewer.org/github/chiarap2/MASTER/blob/master/research_questions/RQ_D_predict_arrival_departure_flow/Predict_Flights_Departure_Arrival_Flows.ipynb

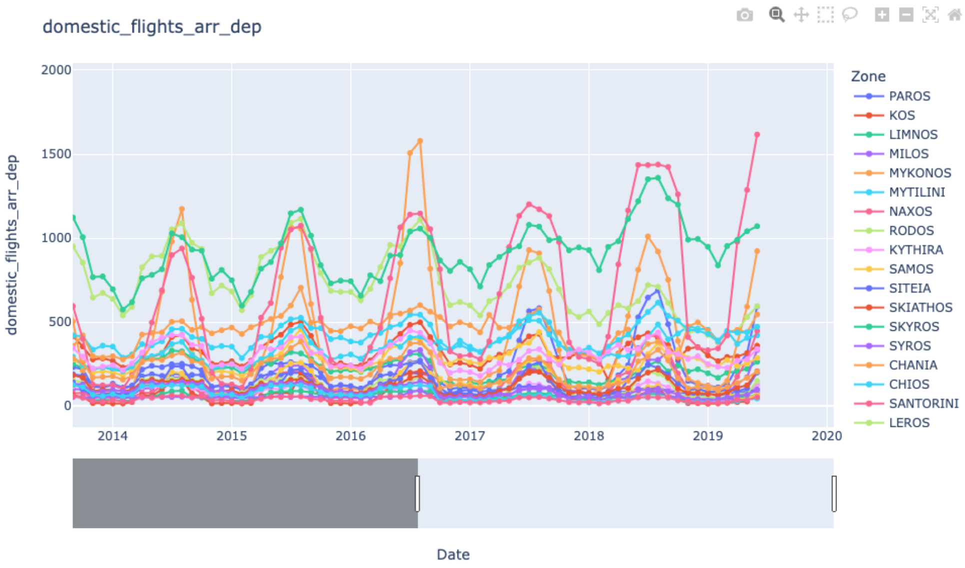


Figure 3. Comparison between the number of domestic passengers of Santorini and similar islands.

3. SEA MONITORING SCENARIO

This section of the deliverable is dedicated to the activity of Task 5.2, namely the Sea Monitoring application Scenario, whose responsibility is UPRC. The material of this part of the deliverable is mainly prepared during secondment to DAL, and integrated during parallel research activities of project partners. The activities have focused on processing and analyzing the datasets and developing a software prototype that addresses the application questions based on the datasets.

The Secondments for the period M24-M60 linked to the Tasks 5.2 are reported in Table 1 at the Introduction section of this document and have a total effort of 1,9 PMs.

APPLICATION QUESTIONS

The activities of this task were focused on three categories of application requirements, namely:

- Data-driven extraction and classification of maritime patterns of life
- Data-driven extraction and classification of maritime anomalies
- Improving the knowledge of the fishing activities in the Northern and Central Adriatic Sea

that were already highlighted and recognized in deliverable D5.1. Despite the fact that these three categories can differentiate, proper spatio-temporal analysis techniques can address all of them.

In spatiotemporal analysis, data mining includes trajectory clustering, classification, anomaly detection, and event prediction. Specifically, trajectory classification is a widely used technique with which normality and behavioral models are created able to identify anomalous patterns or events of interest. On the other hand, trajectory clustering approaches are often employed to form groups of vessel positions with similar spatiotemporal behaviors, uncovering behaviors that are harder to predefine. Although there is an abundance of studies in the literature regarding offline trajectory classification and clustering, fewer works have focused on stream processing of events in the maritime domain. Event processing methodologies are faced with significant challenges when employed on streaming data where the requirements for such applications demand low memory consumption and decreased latencies.

The context of the analysis in most studies is typically the physical world and the geography. Latitude and longitude are the basic features in a multi-dimensional space (speed, direction, etc.). However, experts rely heavily on the visualization of trajectories to manually identify parts of the trajectory that are of some importance. This provides the intuition to move the analysis into a different domain, leveraging computer vision techniques in classification. In computer vision, the most commonly used techniques include Convolutional Neural Networks (CNNs). Each layer of a CNN identifies a different feature of the image, including but not limited to shape and color. In order to increase the performance of CNNs, researchers have employed deep learning, which increases the complexity of the networks in terms of number of hidden layers and nodes. One of the most common goals of such networks is to classify a set of images into a predefined set of labels that are of interest. For these reasons, we have focused our efforts into developing a methodology by transforming trajectories into image representations and then classifying them by employing CNNs.

Trajectory classification can act as the basis for the identification of maritime patterns of life, anomalies or even fishing activities. For the first scenario, a suitable classification methodology is able to train on already annotated patterns of life and identify these pre-defined patterns on streams of vessel tracking data in real-time. Anomaly detection can also be performed through trajectory classification and that is why in the literature authors build normality models based on patterns of usual behavior. The classifier is trained on trajectories of normal behavior and is able to detect possible moving patterns that deviate from the normal ones and are therefore considered anomalous. Regarding the third scenario, when vessels are engaged in fishing activities, they tend to perform specific movement patterns that might correspond to retracting the fishing net or moving towards the fishing area. The movement patterns might differ based on the fishing technique and the fishing gear used. Therefore, a classification methodology is able to train on specific patterns and recognize them on new datasets. The results of the classification can be obtained by the maritime authorities that will further inspect and analyze them in order to draw the final conclusions for the vessels' behaviors.

Efforts in this task have resulted in the development of a software prototype that can provide online summarized representations and classifications of large volumes of trajectories. The research conducted during the activities of this task has led to a research publication [1].

Regarding the datasets, during the current year we did not have variations in the datasets descriptions already reported in D5.1. In the current deliverable, we report only the datasets which we used for answering the application questions, that will feed the prototype.

- **AIS Surveillance data from Eastern Mediterranean**

Efforts towards answering the application questions have shown that we are able to provide promising solutions without the usage of sea state datasets. Nevertheless, we will continue to investigate whether the other listed datasets in Deliverable D5.1 can improve our models.

THE SOFTWARE PROTOTYPE

The software prototype, called TraClets, is implemented in Python 3 using the Numpy and Keras frameworks, that provide efficient numerical routines for multi-dimensional data and neural network implementations, respectively. A TraClet is an image representation of a trajectory. This representation is indicative of the mobility patterns of the moving objects. TraClets needs to efficiently visualize and capture two key features that characterize the trajectory patterns of moving objects: i) the shape of the trajectory which indicates the way the object moves in space, and ii) the speed that indicates how fast the object moves in space.

The prototype provides support for the classification of trajectories. Most trajectory classification approaches found in the literature, require a pre-processing step that involves the understanding and analysis of data and the selection of features suitable only for the moving objects' trajectories to be classified. This means that features selected for a certain trajectory (e.g. cars) cannot be applied to other patterns as well (e.g. vessels). The proposed software skips entirely the aforementioned pre-processing step; the same technique for classifying an image (e.g. CNNs) can be applied for the classification of other trajectories since they are transformed into images. Therefore, TraClets yield a promising universal approach for the classification of moving objects' trajectories and have been used for the classification of vessel mobility patterns at sea.

Figure 1 summarizes the pipeline of the proposed software. Initially, historical trajectories are collected and are annotated into specific mobility patterns, each pattern representing a different classification label. Then, trajectories of each label are transformed into images. Finally, these images are used to train a deep learning model.

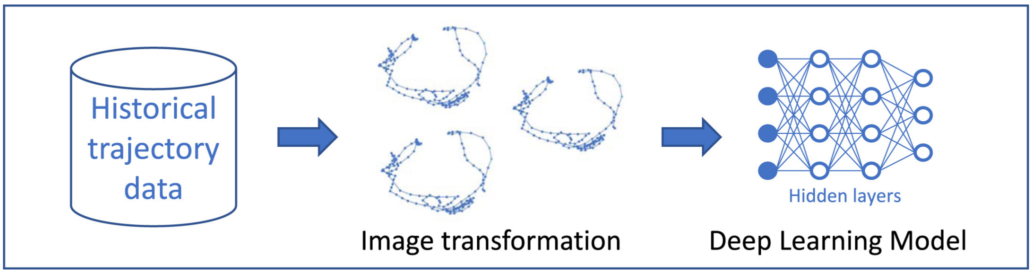


Figure 1: Pipeline of the proposed software.

The software prototype can be found at the Github: https://github.com/kontopoulos/TraClets

DATASET

The dataset used for this example contains trajectories collected from a terrestrial receiver of vessel tracking data that covers the Saronic Gulf (Greece) including the port of Piraeus – a subset of the AIS Surveillance data from Eastern Mediterranean. The dataset provides information for 1229 unique vessels and contains 11,769,237 positional records in total. The vessels have been monitored for almost one and a half month period starting at February 18th, 2020 and ending at March 31th, 2020. Finally, the dataset contains three classes, one for each of the mobility patterns present in the trajectories, namely Anchored, Moored and Underway.

INSTALLATION INSTRUCTIONS

To install the prototype, one can simply clone the repository using the following command in a UNIX Bash shell:

git clone https://github.com/kontopoulos/TraClets.git

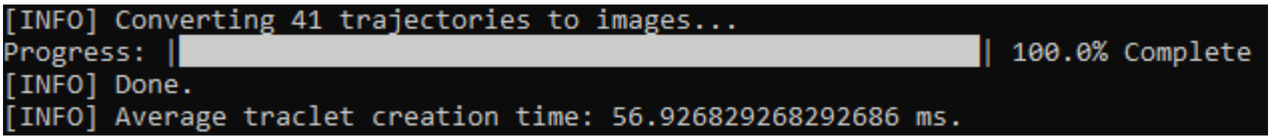
It is worth noting that "git" needs to be installed first. Then, the Python 3 programming language needs to be installed along with the dependencies mentioned in the GitHub repository.

HOW TO USE THE PROTOTYPE

To use the software, the user can simply run the following command:

```
python traclet.py --d [dataset_path] --s [size of the resulting images]
```

where "traclet.py" is the python file that converts a trajectory dataset into a set of images. The trajectory dataset contains positional data of vessels along with the annotations of each trajectory, e.g., anchored or fishing. Moreover, the --d parameter denotes the location of the dataset file and the --s parameter denotes the size of the resulting images, e.g., 224 means that the images will have a resolution of 224 by 224 pixels. After running the above command a progress bar will appear indicating the progress of the overall procedure (see Figure 2). Furthermore, the total number of trajectories and the average time it took to convert the trajectories into images will also be indicated.



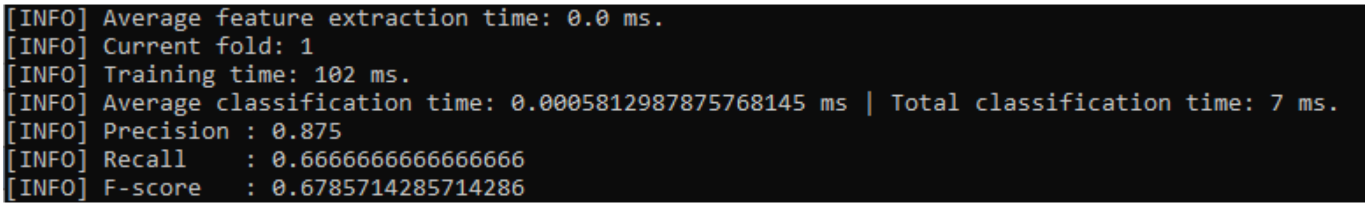
```
[INFO] Converting 41 trajectories to images...
Progress: |██████████████████████████████████████████| 100.0% Complete
[INFO] Done.
[INFO] Average traclet creation time: 56.926829268292686 ms.
```

Figure 2: Example of trajectory transformation into images.

Following the creation of the images is the actual trajectory classification. To perform the classification, the following command needs to run:

```
python classifer.py --d [dataset_path] --f [number of folds]
```

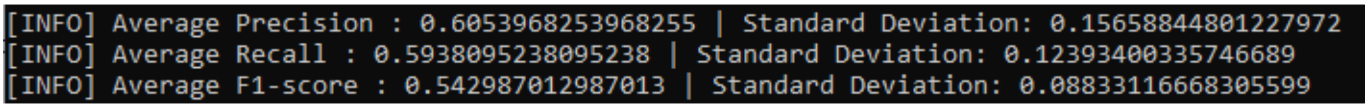
where "classifier.py" is the python file that classifies the newly created image dataset into pre-defined labels. Furthermore, the --d parameter refers to the dataset folder created by the previous python file that contains the trajectory images and the --f parameter refers to number of folds the cross-validation methodology will use when training and evaluating the performance of the classifier. The classifier used in this example is the Random Forests, but Neural Networks can also be used to classify the images. The GitHub repository https://github.com/AntonisMakris/d-LOOK contains instructions on how to use pre-trained Convolutional Neural Networks (CNNs) to classify sets of images. Figure 3 visualizes the results that will be obtained when the classifier is run. Specifically, for each fold in the cross-validation, the training time, the average feature extraction time and the classification time will be illustrated. Moreover, the classification performance of each fold will also appear.



```
[INFO] Average feature extraction time: 0.0 ms.
[INFO] Current fold: 1
[INFO] Training time: 102 ms.
[INFO] Average classification time: 0.0005812987875768145 ms | Total classification time: 7 ms.
[INFO] Precision : 0.875
[INFO] Recall    : 0.6666666666666666
[INFO] F-score   : 0.6785714285714286
```

Figure 3: Illustration of the results obtained when running the classifier.

Finally, when the cross-validation finishes, the average classification results will be showed as seen in Figure 4.



```
[INFO] Average Precision : 0.6053968253968255 | Standard Deviation: 0.15658844801227972
[INFO] Average Recall : 0.5938095238095238 | Standard Deviation: 0.12393400335746689
[INFO] Average F1-score : 0.542987012987013 | Standard Deviation: 0.08833116668305599
```

Figure 4: Average classification resutls.

4. TRANSPORTATION SCENARIO

This Chapter of the deliverable is related to Task 5.3, namely the Transportation Scenario, whose responsible is UNIVE. From GA Annex 1 Part A: Task 5.3: This task will study methods for: creating a public transportation observatory for buses data; improving traffic prediction; improving ride sharing methods to reduce private vehicle usage. We will exploit the data from PUC and UNIVE and datasets collected from social media to develop a prototype application.

The material of this part of the deliverable has been prepared during secondments to PUC, but also integrated during parallel activities of the project partner UNIVE. The activities have focused on processing and analyzing the datasets and developing a software prototype that addresses the application questions based on the datasets.

The secondment of the period M24-M60 linked to Task 5.3 is reported in Table 1 in the Introduction section of this document for a total of 1,5 PMs.

During secondments to PUC, secondees have looked in depth and discussed with PUC researchers about the datasets presented in D5.1 namely BIKE-RIO and BUSES-RIO and how these datasets can be useful for reaching task objectives and building the prototype planned at M60. It has been observed that the available datasets are quite obsolete with respect to the changes in the transport infrastructure due to Olympic Games in 2016. For this reason, the prototype has been implemented by using only the dataset available at UNIVE.

PUBLIC TRANSPORTATION IN VENICE (ITALY)

UNIVE is collaborating with ACTV, the main public transport company, in order to study the flows and to understand the behavior of its users.

As already described in Deliverable D5.2, ACTV provided us two different GTFS (**General Transit Feed Specification)** data: one about the navigation service and the other one about the bus transport. The number of stops is large and this makes it difficult to find meaningful behavior patterns of users. Hence, we reduced this number by merging stops. We obtained five big areas for the bus transport, i.e., Lido, Airport, Piazzale Roma, Mestre Train Station and Mestre (urban area) and 69 merged stops for the navigation service aggregating stops located on the same landing stage (see Deliverable D5.2 for details). After this aggregation process, from 2500 stops, we obtained a set containing only 74 stops.

Moreover, ACTV gave us several validation datasets. These datasets appear like big CSV files where each line represents a single validation of a specific user. Each validation includes the following pieces of information:

- Date and hour of the stamp
- Serial number of the user
- Code_Profile
- Code of the stop in which the ticket has been stamped
- Name of the stop.

As already highlighted in D5.1, for privacy reason, the Serial number of the user is anonymized, in a way we cannot disclosure the identity of the user.

As described in D5.2, from these data we built trajectories, by grouping the stamps associated with the same user. Hence a trajectory is a sequence of couples of this form: (*stop, time*), where *stop* is the identifier of the stop in which the user gets on the waterbus/bus and *time* is the moment when this action takes place. Notice that we replaced the original stops with the ones we created according to the aggregation process we have just described. The sequence of couples is ordered with respect to the time component. Some cleaning operations are done to avoid useless validations. For instance, when for a user there are several validations having close timestamps, few minutes of difference, we keep only the last validation.

Moreover, we use the code Profile of the ticket in order to select different typologies of users. In particular, we consider:

1. One-day ticket
2. Two days ticket
3. Three days ticket
4. Seven days ticket
5. Monthly ticket
6. Yearly ticket

Based on the knowledge of ACTV we assume that tickets belonging to categories 1 to 4 (called *time limited tickets*) are typically bought by tourists whereas categories 5 and 6 are purchased by people living, working or studying in Venice.

APPLICATION QUESTIONS

In Deliverable D5.2 we reported the questions of interest for ACTV, and how we intend to find a solution. For the sake of completeness, we recall the questions:

a) How do the stamps of the users vary in space and time?
b) Is it possible to classify the users into categories, like workers, students, tourists?
c) Are there typical patterns in the movements of the users?

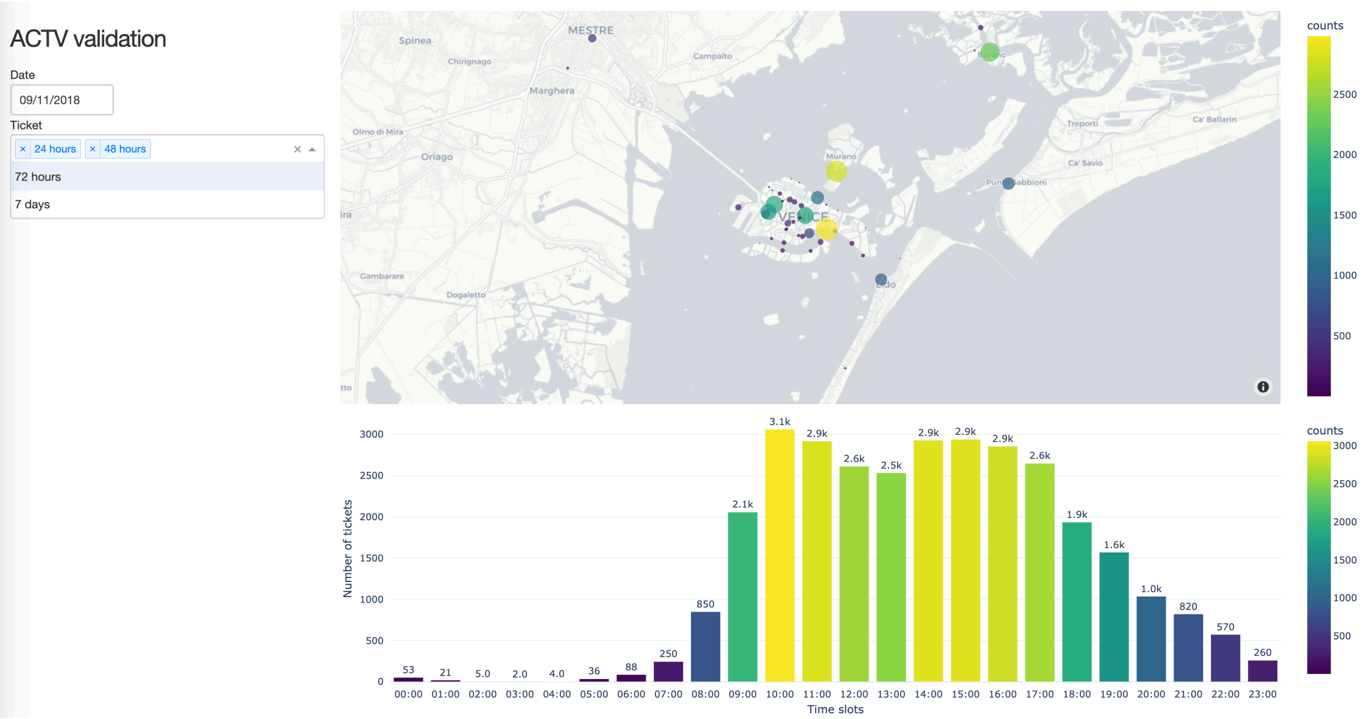
d) Are there different behaviors during the weekday and at the weekend?
e) Which are the common itineraries for tourists?
f) Detect the behavior of tourists during the different days of their stay.

The first version of the prototype will cope with questions **a)**, **c)** and **d)** and in the next section we will provide details on the solutions. We leave the remaining questions **b)**, **e)** and **f)** to the final version of the prototype.

THE SOFTWARE PROTOTYPE

The software prototype has been developed using Python and it is built by using the datasets previously described, i.e., Validations, Stops, Aggregated stops and Reconstructed trajectories.

To answer question **a)** the prototype allows for the visualization of the locations (i.e., the stops) of the validations along the different time periods. The user can choose a day and a typology of the ticket, even more than one typology. The tool returns an image composed by two parts: in the upper part the spatial distribution of the validations is illustrated and the color and the dimension of the balls express and are proportional to the number of validations. In the bottom part a histogram shows the temporal distribution of the validations along the different hours of the day. Figure 5 is an example of the result for question a).



ACTV validation

Date
09/11/2018

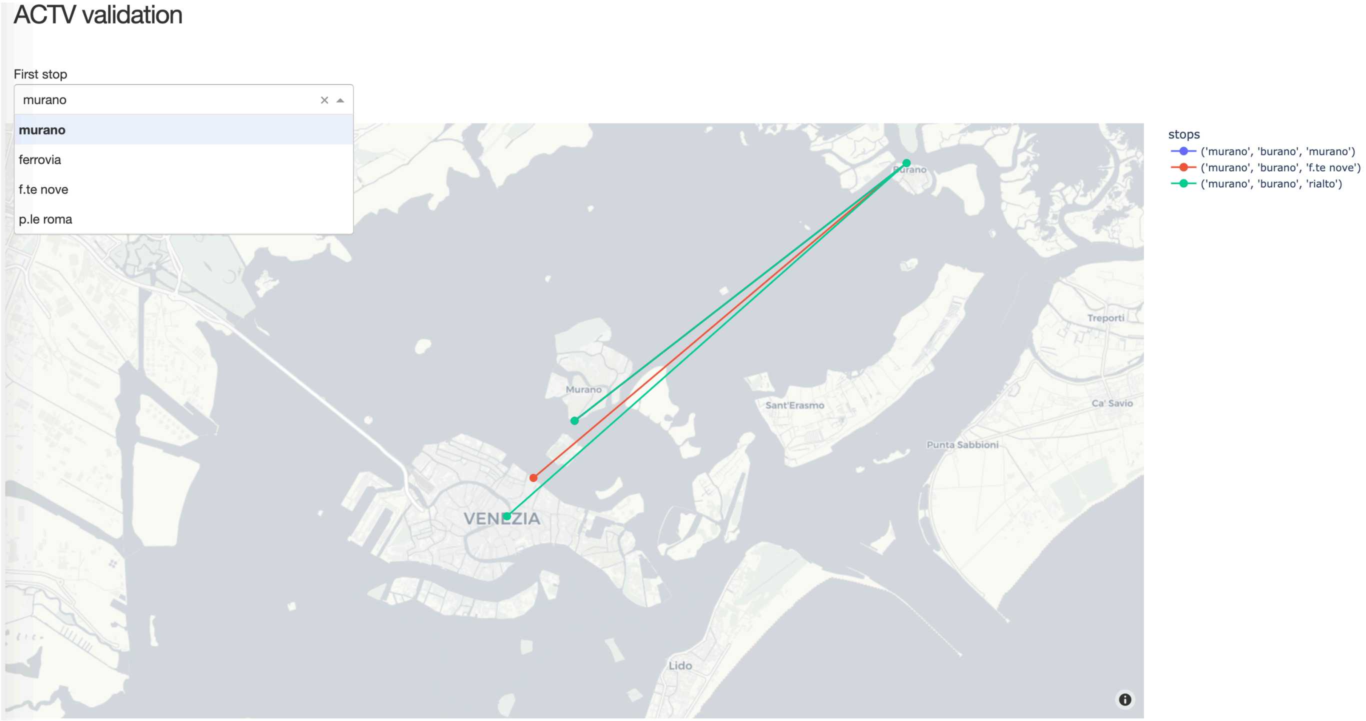
Ticket
× 24 hours × 48 hours
72 hours
7 days

Figure 5. Stamps of the users in a single day with tickets of 24 and 48 hours. In the map it is possible to see how the stamps are distributed in space in all the ACTV stops, while the bar chart shows the distribution over time (for all hours of the day).

Another functionality provided by the prototype useful to browse through the users' validations in space and time allows for building animations showing the heatmaps of the validations every three hours.

To answer question **c)** we focused on one-day tickets since this is the largest dataset and we considered users having trajectories with 3 or 4 or 5 stops. We applied different clustering techniques, such as K-Means, DBscan, Hierarchical cluster, considering only the spatial component of the trajectory. For each algorithm we compared the results obtained by using several distance metrics, such as Jaccard, Edit, LCSS (longest common subsequence). The best result we obtained was by using hierarchical cluster and, as similarity measure, we defined a combination of the Longest Common Subsequence (LCSS) with the longest common substring, called LCSS balance. Recall that a subsequence of a sequence is obtained by removing some elements, hence it can have "holes", while a substring must include consecutive elements of the original sequence. Thus, LCSS balance gives more emphasis to continuous subsequences. For a wider description of the approach, refer to the work [2].

Among the mined patterns, the prototype visualizes the top patterns followed by more than 2000 users. Figure 6 shows patterns starting from Murano stop.



ACTV validation

First stop: murano

Figure 6. Three common patterns of the movement of the tourists.

To answer question **d)** the user can select periods of consecutive days. In particular, she/he can consider only weekdays or weekends and compare the different results.

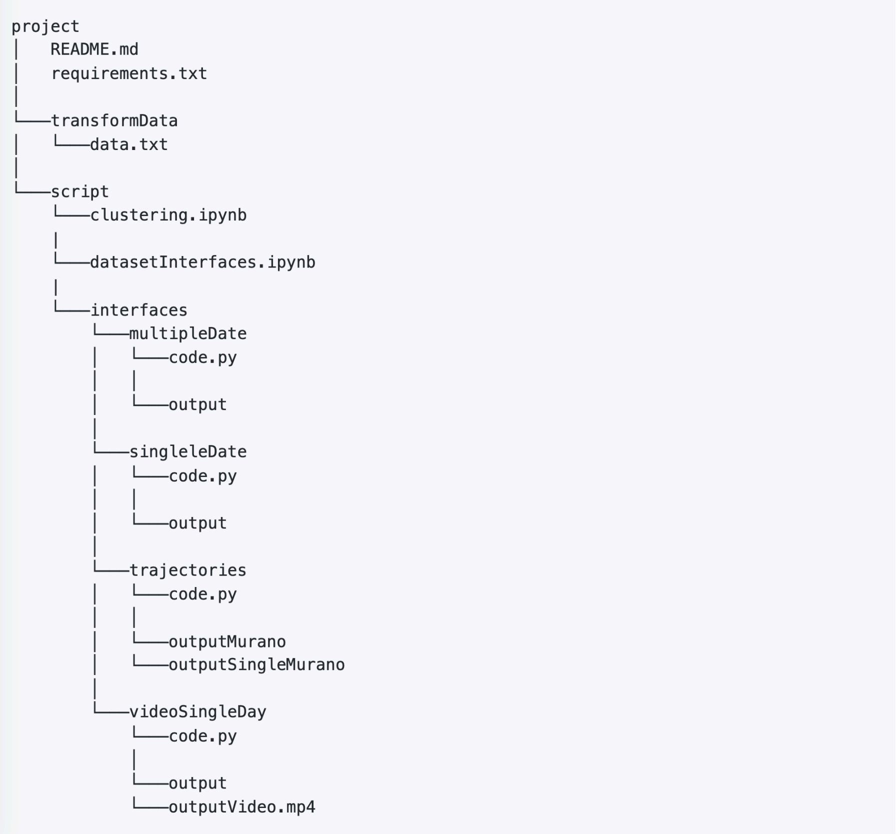
INSTALLATION INSTRUCTIONS

To run the code entirely and visualize the results of our analysis, it is possible to browse the Github repository (https://github.com/giuRov/MASTER.git).
To install the prototype and run it locally, you can download the repository and install the requirements with the following command.

```
pip install -r requirements.txt
```

HOW TO USE THE PROTOTYPE

The GitHub repository is organized as in Figure 7.



```
project
│   README.md
│   requirements.txt
│
└───transformData
│   └───data.txt
│
└───script
    └───clustering.ipynb
    │
    └───datasetInterfaces.ipynb
    │
    └───interfaces
        └───multipleDate
        │   └───code.py
        │   │
        │   └───output
        │
        └───singleDate
        │   └───code.py
        │   │
        │   └───output
        │
        └───trajectories
        │   └───code.py
        │   │
        │   └───outputMurano
        │   └───outputSingleMurano
        │
        └───videoSingleDay
            └───code.py
            │
            └───output
            └───outputVideo.mp4
```

Figure 7. The folder structure of MASTER repository on Github.

In the `transformData` folder there is a `.txt` file with the list of the datasets needed for the analysis. In the `script` folder there are two files with the `.ipynb` extension in which we analysed the datasets in the folder described above, and we built the `.csv` files used for displaying the data in the `interfaces` folder. In the `interfaces` folder, you can find a subfolder for each of the interfaces. In each of these subfolders, you find a file with the `.py` extension which contains the code used to build the interfaces and an `output` file with the results.

For example, we can see how the stamps of the users vary in space and time for a single day choosing different types of tickets (Figure 5) by downloading the image in the `singleDate` folder of the Github repository[4].

5. CONCLUSIONS

This deliverable reports "Preliminary Software prototypes" for application scenarios. As reported in the MASTER Grant Agreement Annex 1 Part A and B we have identified three application scenarios linked to three relative tasks: T5.1 tourism, T5.2 sea monitoring and T5.3 transportation.

[4] https://github.com/giuRov/MASTER/blob/main/script/interfaces/singleDate/output.png

The content of the deliverable has been produced during secondments linked to WP5 and parallel activities of project partners. The total effort in PMs for secondments linked to WP5 from M1 is 19,73 PMs. The effort related to the period of this deliverable, i.e. from M24 to M60, is 6,07

In this deliverable we report about the three software prototypes: (1) tourism scenario based on data and application questions from Thira partner; (2) Sea monitoring scenario based on Data and question from DAL partner and (3) transportation scenario based on the data and application questions from UNIVE and PUC partners.

The current deliverable therefore takes as input D5.1 and D5.2 and it will be an input to D5.4 at M70.

6. PUBLICATIONS

[1] Ioannis Kontopoulos, Antonios Makris, Konstantinos Tserpes. TraClets: A trajectory representation and classification library. SoftwareX, vol. 21, February 2023. This is publication N. 45 in SyGMA and it is gold open access. Link to repository https://www.sciencedirect.com/science/article/pii/S235271102300002X

[2] Héctor Cogollos Adrián, Santiago Porras Alfonso, Bruno Baruque Zanon, Alessandra Raffaetà, Filippo Zanatta. This is publication N. 46 in SyGMA Discovery of tourists' movement patterns in Venice from public transport data. Proceedings of SAC 2022: 564-568.