D3.4 – MID-TERM REPORT ON DATA COLLECTION FROM MULTIPLE SENSORS AND VISUAL CONTENT

WP3 – Enhanced sensing for the water security and safety

aqua3s project has received funding from the European Union’s Horizon 2020 research and innovation programme under grant agreement No 832876.

Document Information

GRANT AGREEMENT NUMBER	832876	ACRONYM	aqua3S	
FULL TITLE	Enhancing standardisation strategies to integrate innovative technologies for Safety and Security in existing water networks.			
START DATE	1st September 2019	DURATION	36 months	
PROJECT URL	www.aqua3s.eu			
DELIVERABLE	D3.4 – Mid-term report on data collection from multiple sensors and visual content			
WORK PACKAGE	WP3 – Enhanced sensing for the water security and safety			
DATE OF DELIVERY	CONTRACTUAL	October 2020	ACTUAL	October 2020
NATURE	Report	DISSEMINATION LEVEL	Public	
LEAD BENEFICIARY	CERTH			
RESPONSIBLE AUTHOR	Anastasia Moumtzidou (CERTH)			
CONTRIBUTIONS FROM	Anastasios Karakostas (CERTH)			
ABSTRACT	This document reports on the current status and latest advances in regards with tasks T3.2 "Area monitoring using UAVs and satellite data" and T3.3 "Social Media Monitoring". The key contributions of the deliverable are: (i) a data acquisition module for Copernicus products; (ii) a flood delineation methodology; (iii) an oil spill detection module; (iv) a framework that collects and analyses social media data from Twitter; (v) a collection of tweets relevant to the PUCs; (vi) a manually annotated training set of relevant/irrelevant tweets; and (vii) a web interface that displays the tweets and pollution maps.			

Document History

VERSION	ISSUE DATE	STAGE	DESCRIPTION	CONTRIBUTOR
0.1	22/09/2020	ToC	Table of Contents	Anastasia Moumtzidou (CERTH)
0.2	23/10/2020	Draft 1	Version for internal review	Anastasia Moumtzidou (CERTH), Anastasios Karakostas (CERTH)
0.3	29/10/2020	Final Draft	Internal review	Apostolos Apostolakis (IP-ASCR), Ioannis Lioumbas (EYATH), Caterina Christodoulou (EYATH)
1.0	30/10/2020	Final Draft	Final version	Anastasia Moumtzidou (CERTH)

Disclaimer

Any dissemination of results reflects only the author's view and the European Commission is not responsible for any use that may be made of the information it contains.

Copyright message

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

ABBREVIATIONS/ACRONYMS

AI	Artificial Intelligence
AoI	Area of Interest
API	Application Programming Interface
BoW	Bag of Words
DCNN	Deep Convolutional Neural Network
DIAS	Copernicus Data and Information Access Services
DNN	Deep Neural Network
EO	Earth Observation
HTTP	Hypertext Transfer Protocol
ID	Identification
IoU	Intersection over Union
ISO	International Organization for Standardization
JSON	JavaScript Object Notation
ML	Machine Learning
NIR	Near Infra-Red
PUC	Pilot Use Case
SAR	Synthetic Aperture Radar
SVM	Support Vector Machine
SWIR	Short-Wave Infrared
SNAP	Sentinel Application Platform
RF	Random Forests
TF	Term Frequency
TF-IDF	Term Frequency–Inverse Document Frequency
UAV	Unmanned Aerial Vehicle
URL	Uniform Resource Locator
WGS	World Geodetic System

1. Executive summary

This deliverable presents the first version of the flood extent delineation, oil spill detection, Twitter's crawling and automatic text classification of water-related tweets in aqua3S, along with a short overview of their State-of-the-Art techniques. Initially, we present the Pilot Use Cases (PUCs) for each of the targets and the corresponding areas of interest as well as the User Requirements (URs) for each use case and for the utilised/employed technologies.

As far as the flood extent delineation is concerned, it identifies the water relative changes realized on the landscape by considering a time series of EO data as input. Flood monitoring is a core part for most of the PUCs. A change detection methodology has been implemented based on outlier detection. Specific processing steps of the Sentinel-1 products were followed to account for geometric distortions and to render the images usable for analysis. For the oil spill detection task two different approaches were tested based on Artificial Intelligence. The first one discriminates each pixel of the image as clean or with oil spill using an SVM model, whereas the second approach makes the same discrimination at patch level via performing transfer learning on a pretrained with ImageNet VGG16 network, retraining with augmented data. As input datasets casual RGB images were compared against false-RGB images with the latter providing improved highlighting of the oil spills. Some PUCS include algae bloom detection using satellite data and area monitoring via drone data will be covered in other aqua3s deliverables.

Regarding the main contributions and achievements of the social media monitoring that are presented in this document, they involve continuous crawling of tweets using the Twitter Streaming API, and processing of the collected tweets in order to remove irrelevant to the use cases tweets. As far as the crawling of tweets is concerned, it depends on the filtering parameters that are inputted, i.e. the keyword-based search criteria that are defined in collaboration with the PUC leaders. After a new tweet has been collected and before it is stored in a database of forwarded into the system, various knowledge extraction techniques are applied which extend the crawled JSON by including as attributes the outcomes of the analysis. To enhance the collected social media, the following knowledge extraction methodologies have been integrated: the estimation of reliability, the detection of mentioned locations, the detection of nudity in images and the text classification to filter out irrelevant posts. As far text classification of tweets is concerned several text representation techniques and machine learning approaches are tested. Eventually, a web application has been implemented that shows the collected tweets and can also be used to collect human annotation and thus create a training dataset to be used for classifying Twitter texts as relevant or not.

2. Introduction

aqua3S combines novel technologies in water safety and security, aiming to prevent disasters relative to drinking water. Various sensor measurements are supported by videos from Unmanned Aerial Vehicles (UAVs), satellite images and social media observations from the citizens that report low-quality water in their area (e.g. by colorisation), creating also social awareness and an interactive knowledge transfer. This document presents the developed method on flood delineation that provides an illustration of flooded areas, various methodologies on oil spill detection classifying the underlying water area as clean or with the high probability of an oil spill and eventually the social media monitoring framework that extract knowledge from the tweets based on keywords denoting discomfort relative to water quality.

Chapter 3 includes the PUCs that have been arranged for each target of the project. It contains the information on all areas of interest as well as the User Requirements (URs). The main difference among PUCs is the areas that are being monitored, which can be either water sources like dams or lakes that are observed via remote sensing (i.e. satellites, drones, CCTVs and sensors) or water distribution networks that can be monitored via ground sensors. Social media data can be used in both cases.

Chapter 4 describes the process of generating various information maps, useful for the public and the authorities to prevent or tackle a hazardous situation relative to drinking water. It begins with an overview of the Area Monitoring framework that uses satellite data and then describes in detail the various steps from obtaining the necessary products to analysing them for possible threats and displaying the information maps. Copernicus data are acquired using Copernicus Open Access Hub API to fetch the appropriate products based on location, date range and satellite type. The search criteria have been defined in collaboration with the PUC leaders are listed in this document, as well as the status of the stored products and information maps for some sample past incidents. Moreover, the description of the analysis of the products is realized, with the flood delineation module that depicts the extent of the flooded areas, including a presentation of the latest related work. Then, the tested oil spill detection methodologies are presented, where the potential threat is being identified either per pixel or per patch level, with a presentation of the latest related work as well. Eventually, the required storing and indexing of Copernicus products and the produced information maps are described.

Chapter 5 begins with an overview of the Social Media Monitoring framework and then describes in detail the various steps from querying Twitter and crawling posts to analysing and displaying tweets. Social media data are acquired using Twitter Streaming API, which offers three types of search criteria, but only keyword-based search has been selected. The keywords that have been defined in collaboration with the PUC leaders are listed in this document, as well as the current status of the collection based on them. The required processing, storing and indexing of tweets are also described, with a focus on the date information and range queries (obtaining tweets from a specific period). Furthermore, this chapter includes the description of four methodologies that aim to extract knowledge from the tweets, i.e. the estimation of reliability, the detection of mentioned locations, the detection of nudity in images and the text classification to filter out irrelevant posts. Several text representation and machine-based classification approaches are evaluated by considering the manually annotated provided by the end users. Finally, the chapter concludes with the visualisation of collected and analysed tweets, demonstrating two user interfaces; one that displays tweets in a list and one that places them on an interactive map

3. Areas of interest per use case

In the scope of aqua3S project seven PUCs have been scheduled that are covering the areas of Trieste (Italy), Thessaloniki (Greece), Paris (France), Limassol (Cyprus), Brussels (Belgium), Sofia (Bulgaria) and Botevgrad (Bulgaria). Based on the nature of each PUC, which involves the particular are of interest, and also on the user requirements as defined in D2.1 “Use cases requirements v1”, we are able to define which of the techniques discussed in the current deliverable (i.e. UAVs, satellite data, social media) are of interest for the end users. Thus, monitoring the water supply system in the city of Brussels can use only social media. The Paris use case focuses on a different level of the platform which involves modules at the business layer and not the interoperability layer (see D7.2 “aqua3S System architecture definition”) as the ones described in the current deliverable. In Italy, the focus is on the area near the city of Trieste as well as near Isonzo and Timavo rivers and the user interests revolve around involve floods, oil spills and algae blooms events by using UAVs, satellite images and the social media. At Thessaloniki, the Aliakmonas river and Thessaloniki water treatment plant is monitored via remote sensing for oil spills. Drones or CCTVs are also considered for identifying objects. Social media could be used but they aren’t considered a priority for the end user partners of this PUC. At Limassol, we monitor the desalinated and treated (surface) water via satellite data to identify oil spills and algae bloom and monitor the social media to check water quality. Finally, in Bulgaria, the water reservoir of Iskar and Bebresh are monitored for oil spills and algae bloom via satellite data and social media, with extra monitoring of the water levels of Bebresh dam due to a recent drought incident. Also the use of a drone is discussed in the case of the Sofia PUC. Table 1 presents the aforementioned information in a concise manner.

PUC ID	Area	Information regarding the area of interest	Techniques tackled					
			Satellite data			Drones	CCTVs	Social Media
			Oil spill	Algae bloom	Floods/ droughts			
1	Trieste (Italy)	Areas near Isonzo and Timavo rivers	✓	✓	✓	✗	✗	✓
2	Thessaloniki (Greece)	Aliakmonas river and Thessaloniki water treatment plant	✓	✗	✗	✓	✓	✗
3	Paris (France)	Region in Paris	✗	✗	✗	✗	✗	✗
4	Limassol (Cyprus)	Desalinated and treated (surface) water	✓	✓	✗	✗	✗	✓
5	Brussels (Belgium)	Water distribution network of Brussels	✗	✗	✗	✗	✗	✓
6	Sofia (Bulgaria)	Iskar dam and water	✓	✓	✗	✓	✗	✓

		distribution network						
7	Botevgrad (Bulgaria)	Bebresh dam	✓	✓	✓	✗	✗	✓

Table 1. PUC information on areas targeted and visual methods.

As far as social media are concerned, the collection of tweets although is language-based, it follows the same procedure in all interested PUCs and thus it was initiated in all of them from the start of the project. However, the analysis of satellite data in order to identify oil spills, algae bloom and floods/ droughts and the analysis of the images from drones and CCTVs differ significantly. Thus, in the current deliverable we are focusing only on techniques identifying oil spills and floods/ droughts, while the algae bloom and data from drones and CCTVs will be considered in the next deliverable D3.6. In order to be able to create models that identify oil spills and floods/ drought, it is necessary to have satellite data capturing such events and also have human annotation of this data on pixel level. Thus, we have developed and tested our methodologies based on the existence of such data. Specifically, for the case of floods/ drought we considered data from the Italian PUC and for the case of oil spills we considered data from the Greek PUC. However, the methods developed can be and will be evaluated to the other regions as well.

Thus, as far the Italian PUC is concerned, we study the Trieste region (Figure 1). Specifically, on 12 November 2019, the Mediterranean area was affected by a deep cyclonic circulation resulting in an intense phase of severe weather over Italian peninsula. The severe weather conditions caused a storm surge of many areas near the city of Trieste. Thus, we are investigating the extent of the flood incident be exploiting a flood delineation technique using SAR data.

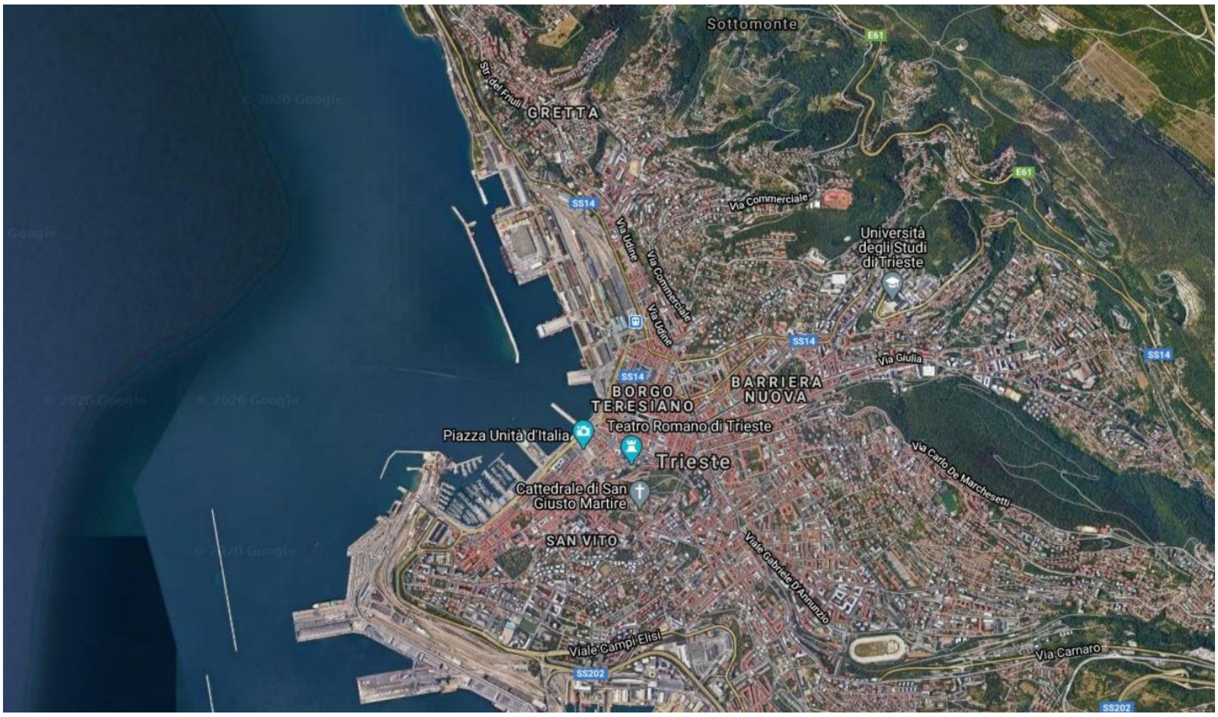


Figure 1. Google maps image with the city of Trieste, Italy.

As far as the Greek PUC is concerned, we study Polyfytos lake at Thessaloniki – Greece (Figure 2), where small oil spills are detected during the 25th Dec (1 oil spill) and 30th Dec 2017 (5 oil spills) by domain experts. The oil spills are visible on 2 bands (band 4 and 8) at 10m resolution, and 6 bands at 20m resolution (bands 5, 6, 7, 8A, 11 and 12). During the 25th of Dec, a small number of additional oil spills look-alike areas have also been detected in the shallower part (SW) of the lake but were not used due to their small size.

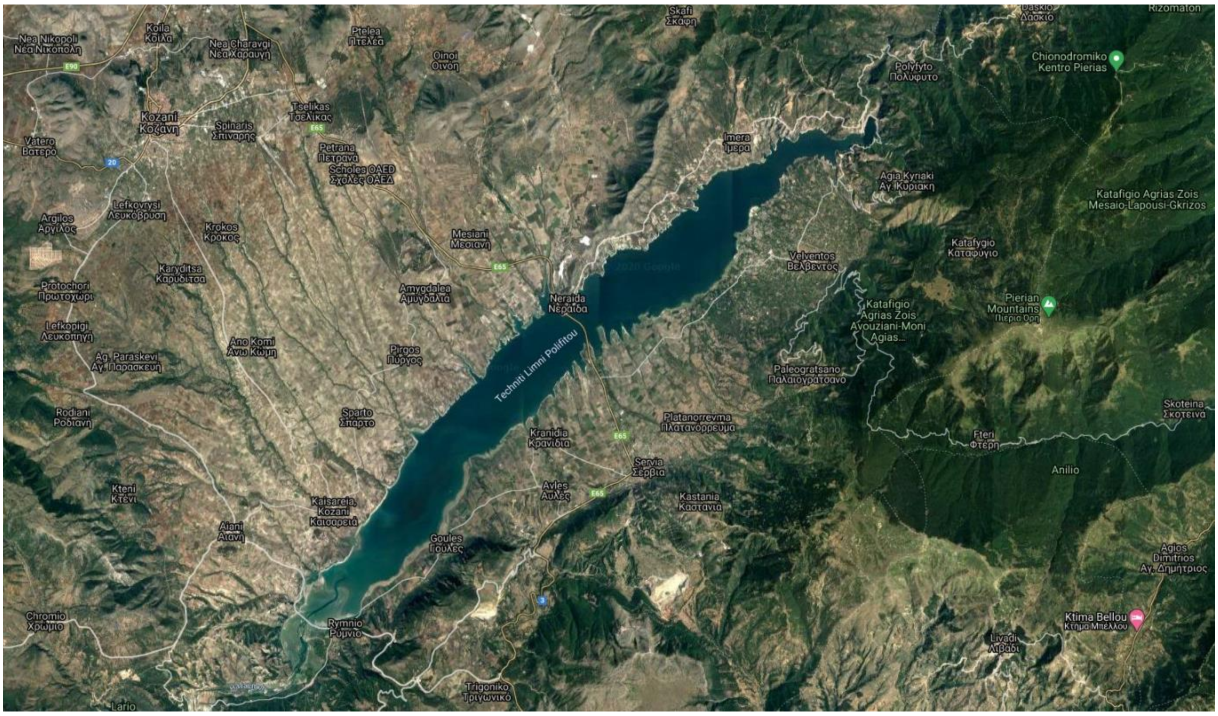


Figure 2. Google maps image with the Poloyfytos lake of Thessaloniki, Greece.

4. Area monitoring using satellite data

The Copernicus Programme[1] offers a consistent and reliable source of information to the aqua3S system and facilitates the generation of information maps on a steady interval for any area around the globe targeting to support the creation of awareness of water quality that associates with surface water.

This chapter is dedicated on the work that has been done so far in the Area Monitoring using Satellite Data task, which will be supplemented in deliverable D3.6 (M27). First, an overview of the framework will be presented (4.1) in order to provide the reader with the overall picture of the implementation. The acquisition of the satellite media data will follow, involving the official Copernicus API (4.2.1), the definition of the search criteria that downloading of products will be based on (4.2.2), the storing and indexing of downloaded products and processed maps (4.2.3) and the description of the current collection of products based on past incidents (4.2.4). The next subchapter refers to the analysis of Copernicus data in order produce water masks (4.3.1) and the oil detection methodologies (4.3.2). Finally, the Web interface that visualises the output flood delineation and the oil spill map, are demonstrated (4.4).

4.1 Overview of the framework

The Area Monitoring framework aims to collect, analyze and push into the aqua3S system satellite data and specifically Sentinel products from Copernicus Open Access Hub. The core components of the framework inside the analysis APIs are the Flood Delineation (4.3.1) and Oil Spill Detection (4.3.2) modules. The complete workflow of this framework is illustrated in Figure 3 and described here.

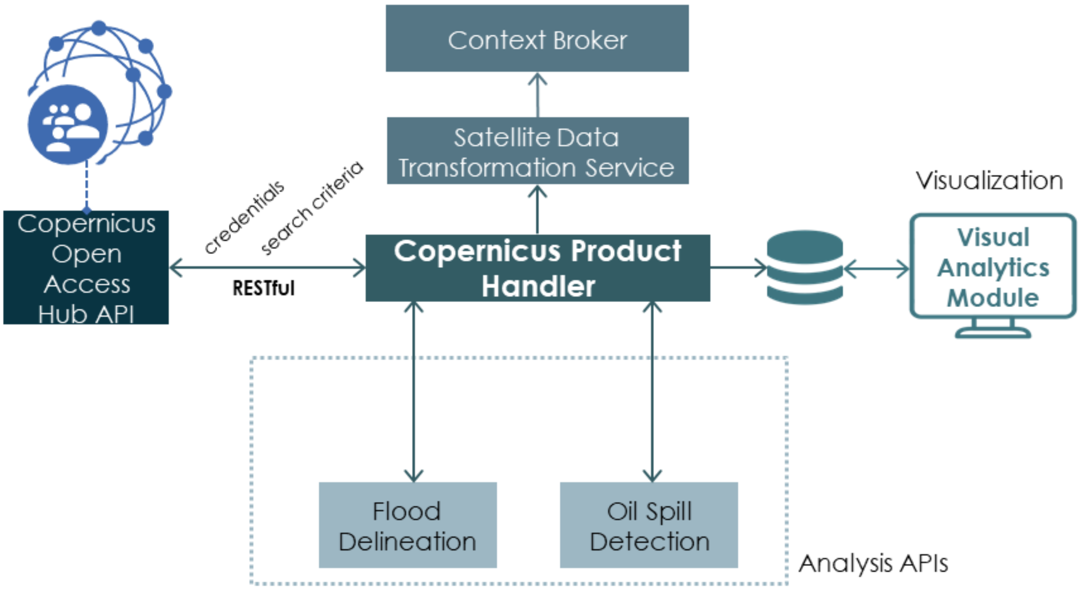


Figure 3. The complete workflow of the Area Monitoring using Satellite Data framework

The orchestration of the framework is performed by the Copernicus Product Handler, displayed in the middle of the figure, which handles every step of the workflow. Initially, the Product Handler uses the

[1] https://www.copernicus.eu/en

necessary credentials to send a request using the Copernicus Open Access Hub API forming a complex query in order to receive periodically a list of the latest Copernicus products that satisfy this query and download them if not already exist in the local system.

For every newly received product, a satellite data analysis procedure is performed by calling the respective APIs. The analysis involves: (i) the delineation of the flooded areas, (ii) the classification of underlying water areas as clear water or oil spills.

After the analysis of a single product, the results are added to its existing metadata and the complete information is stored to a MongoDB database. The web interface namely Visual Analytics Module (a local implementation is described at 4.4) connects to this database and visualises the information maps.

After storing the analyzed products in the database, the Product Handler also produces a minimised version of each information map (keeping only some fundamental attributes) and sends it to the Satellite Data Transformation Service (more details will follow in D4.3 (M24)), which subsequently converts the raster information into an appropriate format and inserts it into the Context Broker, thus pushing it into the aqua3S system.

4.2 Satellite data acquisition

4.2.1 Fetch satellite data from Copernicus Open Access Hub

For the task of the area monitoring using satellite data we will use ESA's Sentinel data using the Copernicus Open Access Hub (previously known as Sentinels Scientific Data Hub) that provides complete, free and open access to a variety of Sentinel satellite products. Sentinel Data are also available via the Copernicus Data and Information Access Services (DIAS) through several platforms. We will focus on the Sentinel-1 radar data and the Sentinel-2 optical data. For the acquisition of data, the Copernicus Open Access Hub is used. An account is necessary to access the API in order to download products. In the API hub portal [2]details can be found on exposing the two dedicated APIs (OData & Open Search), the available search filters and scripts with examples.

4.2.2 Defining search criteria

As explained earlier, the satellite data that is download from the Copernicus open access hub API depends on the filtering parameters that are inputted to the API, i.e. the search criteria. The definition of these criteria in the aqua3S project are mainly the coordinates of the AOIs and the desired case type to be monitored and have been achieved in close collaboration with the PUC leaders, since the searching parameters should reflect the needs of each use case and lead to information maps that are valuable to the end users. The search options vary for each of the available information map. Each PUC is described by its own areas in the longitude / latitude geographic coordinate system and information type (e.g. flood monitoring, algae bloom, oil spill).

4.2.3 Storing and indexing

After a new Copernicus product has been downloaded the metadata of the file are extracted in order to be reused at the processing phase. Table 2 describes the entry that is inserted into MongoDB each time a new product is downloaded.

[2] https://scihub.copernicus.eu/twiki/do/view/SciHubWebPortal/APIHubDescription

Attribute	Value description
_id	the unique naming convention that MongoDB uses across all of its content
product_type	S2MSI2A for Sentinel 2, GRD for Sentinel-1 products
source	The url of the hub: “https://scihub.copernicus.eu”
url	The unique download url: https://scihub.copernicus.eu/dhus/odata/v1/Products('f5005654-4a3c-4b92-be03-4a0b15aa8ccb')/$value"
timestamp	The date and time the product was download locally in timestamp format
productName	The unique product name, e.g. “S2A_MSIL2A_20190713T101031_N0213_R022_T32TQR_20190713T135651”
platform	The satellite type, S1A/S1B or S2A/S2B
queryGeo(type, coordinates)	The coordinates of the search query in polygon WKT format, e.g. “POLYGON((11.4868 45.3816,11.8231 45.3816, 11.8231 45.6236,11.4868 45.6236,11.4868 45.3816))”
datetime_sensing	The datetime that the satellite captured the image
datetime_ingestion	The datetime that the Copernicus product was ingested in Copernicus hub
location_name	The PUC name, e.g. PUC 1 – Trieste
geo	The coordinates of the downloaded product
sensor_mode	Thee sensor mode e.g. IW for Sentinel-1
zippedData	The local directory of the downloaded Copernicus product, e.g. "/data/public/S1_products/ S2A_MSIL2A_20190713T101031_N0213_R022_T32TQR_20190713T135651.zip "

Table 2. The basic information that is inserted into MongoDB after a product is downloaded

After the product is downloaded the analysis is triggered. Depending on the PUC the corresponding analysis API is called, that includes the oil spill detection or the flood delineation module. Once the algorithm finishes execution the aforementioned entry gets updated with the type of pollution and the file path of the generated information map.

Eventually a version of the previous JSON with limited fields is sent to the Satellite data Transformation Service, in order to be converted into an appropriate format for insertion to the Context Broker (more details will follow in D4.3 (M24)), while the complete JSON is stored in the MongoDB database. See Appendix I for examples of a descriptive entry, as well as an oil spill and a flood map entry.

4.2.4 Current status of the retrieved products

For the two used cases we have downloaded the necessary Copernicus products.

For PUC1 – Trieste we are monitoring the flood that occurred lately and we downloaded a series of Sentinel-1 GRD-IW products. The peak of the flood event is estimated to be on 15 November of 2019. Thus, we used this date as the target date (Figure 4) for the flood monitoring module and we downloaded a timeseries of 30 products three months up to three months prior the target date.

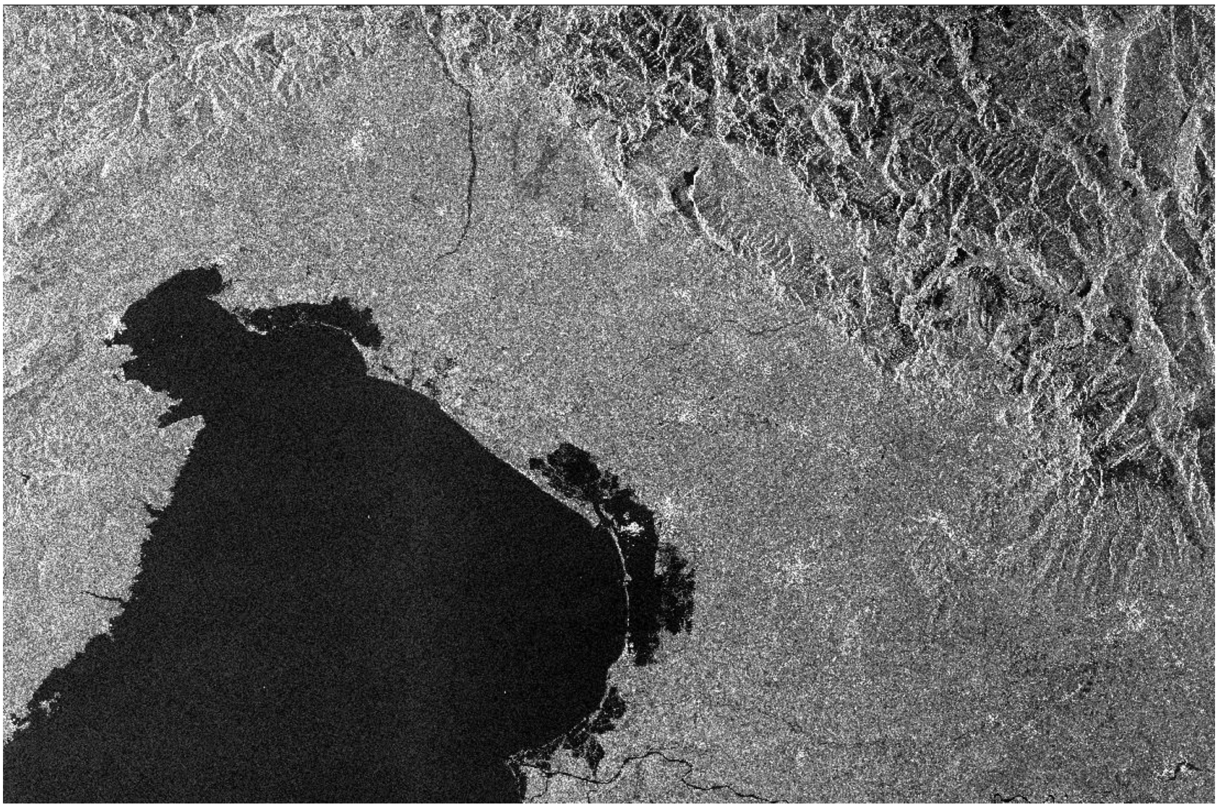


Figure 4. Sentinel-1 GRD-IW product of day 15/11/ 2019 near Trieste
S1A_IW_GRDH_1SDV_20191115T051904_20191115T051929_029917_0369DD_B3EC

For PUC2 – Thessaloniki and Polyfytos lake we used optical data to perform the oil spill detection. The required Sentinel-2 level 2A products provide depictions of the Polyfytos lake on 20, 25 and 30 December of 2017 (Figure 5).

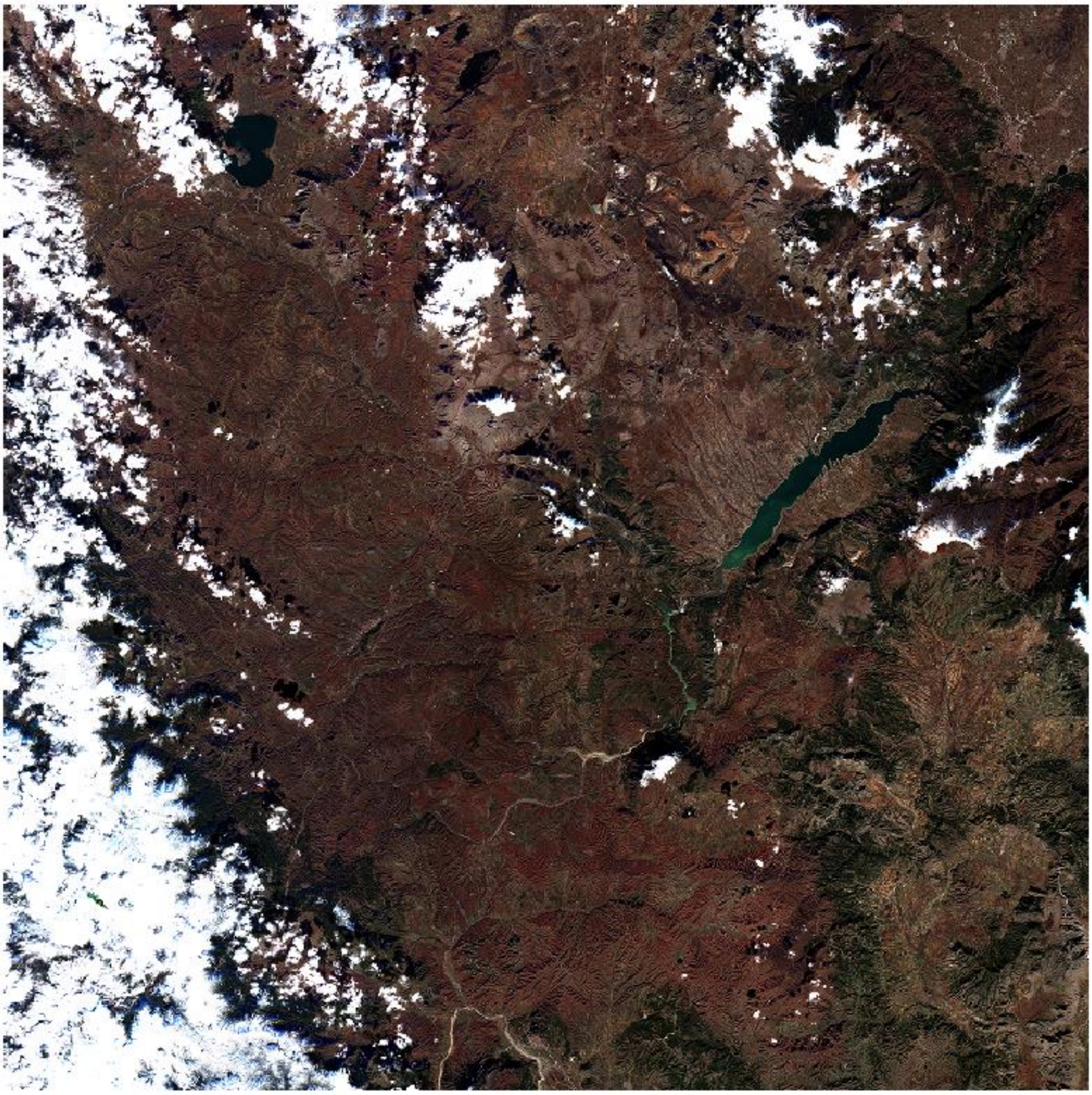


Figure 5. The natural color (TCI) 10m resolution band of whole product on 30/12/2017.

4.3 Classification of areas using satellite data

In this section, we present the related work and the proposed visual approaches used for identifying water pollution and water areas via satellite images. As far as water pollution is concerned, we aim at identifying oil spills or algae concentrated on water surface. However, in the current deliverable we will focus only on identifying oil spills as algae bloom identification will be part of the next deliverable D3.6 (M27). Regarding the detection of water areas, our aim is to identify whether a flood or a drought has happened or is on progress.

4.3.1 Flood delineation

4.3.1.1 Related work

Due to the penetration capacity of synthetic aperture radar (SAR) data through clouds and bad weather conditions, it is ideal for flood monitoring. Many approaches have been investigated using radar data to directly delineate water areas or by using change detection techniques to generate flood maps. For automatic extraction of flooded areas in multi-temporal satellite imagery acquired by Sentinel-1 Synthetic Aperture Radar (SAR), two neural network algorithms are presented (Nallapareddy et al. 2020): Feed-Forward Neural Network and Cascade-forward back-propagation neural network. The models are first trained using a variety of input data until the percentage of error with respect to water body detection is within an acceptable error limit. These models are then used to extract the water features effectively and to detect the flooded regions. Finally, flood area is calculated in sq. km in during flood and post-flood imagery using these algorithms. The results thus obtained are compared with that from the binary thresholding method from previous studies. The chosen study area corresponds to few districts of Uttar Pradesh, India where River Rapti merges into River Ghaghara (SAR-1 data).

An automatic SAR image change detection and classification system is presented (Pandeeswari et al., 2020) that utilises a Radial Basis Function-based Deep Convolutional Neural Network (RBF-DCNN). The methodology comprises of 6 phases: pre-processing, obtaining difference image, pixel-level image fusion, Feature Extraction (FE), Feature Selection (FS), and also change detection (CD) utilizing the classifier. Initially, the noise is eliminated as of the input, SAR image 1 and SAR image 2, utilizing the NLMSTAF approach. Subsequently, the difference image is attained by utilizing a Log-ratio operator (LRO) and Gauss-LRO, and the attained difference image is then fused. Next, the LTrP, WST, edge, and MSER features are extracted from the fused image at pixel level. As of those features that were extracted, the necessary features are selected utilizing the Hybrid GWO-GA algorithm in order to avoid time complexity. The features (selected) are finally inputted to the RBF-DCNN classifier for detecting the changes in an image.

A practical method of change detection is suggested (Kim et al., 2020) that simply computes flood extent and water volume in rapidly analysis. At first, a -stable distribution was fitted to intensity histogram for removing the non-water-affected pixels. This fitting differs from other typical histogram fitting methods, which is applicable to histograms with two peaks, as it can be applied to histograms with not only two peaks but also one peak. Next, another type of threshold based on digital elevation model (DEM) data was used to correct for residual noise, such as speckle noise. in order to detect the extent of flooding with high accuracy, VV images were used to detect water-based pixels that changed over time due to the dam collapse and to determine the change in the volume of water using DEM data.

4.3.1.2 Flood delineation methodology

For the flood detection task, we tuned the baseline method proposed in H2020-776019 EOPEN by performing thermal noise removal in the preprocessing of the Sentinel-1 GRD-IW products and apply the orbit file that provides an accurate position of SAR image and the update of the original metadata of the SAR product.

The baseline technique:

To tackle changes in terms of flood/drought, we applied a change detection technique on a time series of processed Sentinel-1 products. The processing steps of the initial products are described in 4.3.1.3. Change points are abrupt variations in time series data. Such abrupt changes may represent transitions that occur between states. Detection of change points is useful in modelling and prediction of time series and is found in the application area of flood monitoring. To detect the changes relative to flood, we take into account a time series of the previous 30 satellite images, in order to detect fluctuations when comparing to a normal state of an area. Outlier detection was applied comparing the target image against the time series (more details on outlier detection are given in Section 3). In our test case we investigated the flood event near Trieste region in Italy on 15/11/2019 with the time series of the 30 satellite images falling between 17/08/2019 and 10/11/2019. For the outlier detection (i.e. the flood areas) we used the formula (Iglewicz et al, 1993):

$$\frac{X - TS_{mean}}{TS_{std}} > alpha$$

where X is the target image of dimensions $width \times height \times 1$, TS_{mean} the mean average of the 30 images time series per pixel of dimension $width \times height \times 1$, and TS_{std} the standard deviation of the 30 images time series per pixel of dimension $width \times height \times 1$. As for alpha the value of 5.0 was selected by manual inspection that allows us to significantly reduce the number of false positives.

4.3.1.3 Dataset description

For the flood detection we processed the Sentinel-1 GRD-IW products of the flooded day and the timeseries images using ESA's Sentinel Application Platform[3] (SNAP). Following processing steps were applied (Filipponi, 2019), with the steps of 'Apply Orbit File' and 'Thermal Noise Removal' being introduced to the current implementation in comparison with the baseline implementation:

[3] https://step.esa.int/main/toolboxes/snap/

- Apply Orbit File: The operation of applying a precise orbit available in SNAP allows the automatic download and update of the orbit state vectors for each SAR scene in its product metadata, providing an accurate satellite position and velocity information.
- Thermal Noise Removal: Reduces noise effects in the inter-sub-swath texture, in particular, normalizing the backscatter signal within the entire Sentinel-1 scene and resulting in reduced discontinuities between sub-swaths for scenes in multi-swath acquisition modes.
- Subset: the initial product is cropped so it contains only the lake we want to observe and its close surrounding areas. Some balance between the inundated and non-inundated areas is desired. Radiometric calibration: Fixes the uncertainty in the radiometric resolution of satellite sensor.
- Speckle noise removal: Helps removing the pepper and salt like pattern noise that is caused by the interference of electromagnetic waves. “Lee Sigma” filter of Lee (1981) with a 5×5 filter size is used to filter the intensity data. As noted by Jong-Sen Lee et al. (2009), this step is essential in almost any analysis of radar images, due to the speckle noise aggravation of the interpretation process. The term noise itself is not strictly correct, because the effect appears due to the coherence of the transmitted pulse, where all of the waves emitted at the same time have the same frequency and phase, and does not reduce the quality of the image.
- Terrain correction: Projects the pixels onto a map system (WGS84 was selected) and re-sampled to a 10m spatial resolution. Also, topographic corrections with a Shuttle Radar Topography Mission (SRTM) digital elevation model (DEM) is performed. Corrects the distortions over the areas of the terrain.
- Linear to Decibel (dB): The dynamic range of the backscatter intensity of the transmitted radar signal values is usually a few orders of magnitudes. Thus, these values are converted from linear scale to logarithmic scale leading to an easier to manipulate histogram, also making water and dry areas more distinctive.

4.3.1.4 Evaluation of the proposed methodology

A qualitative analysis between the improved version and the baseline has been performed. Here you can see the two flood masks with thermal removal and the one without the extra preprocessing steps (Figure 6).

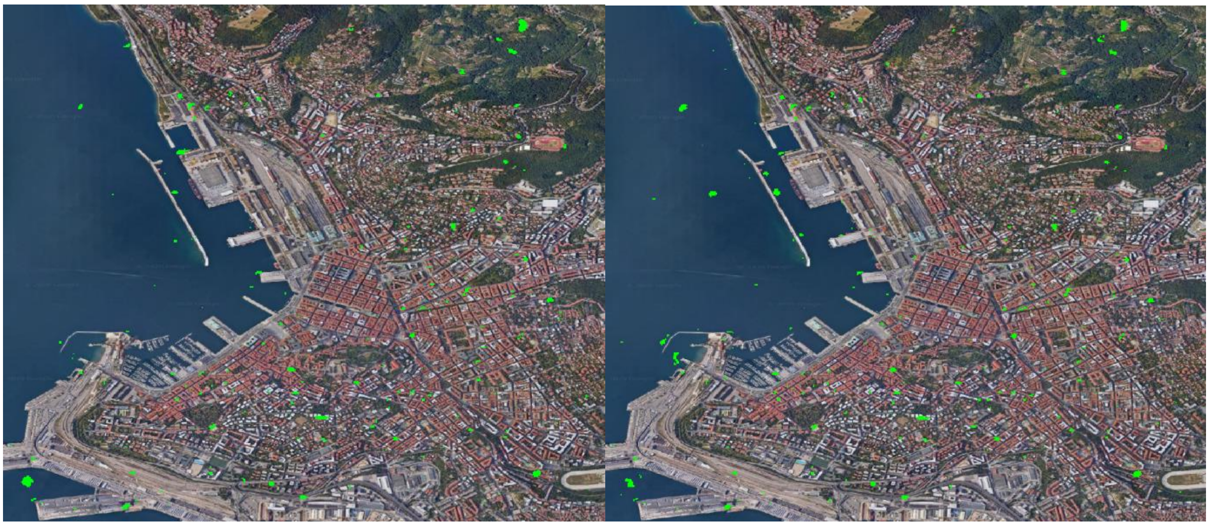


Figure 6. Depiction of the flooded areas with the baseline method (left) and improved method (right) of Trieste flood incident on 15/11/2018

Some minor differences can be observed between the two images, with the thermal noise removal tending to remove some minor artifacts, i.e. sparse water areas.

4.3.2 Oil spill detection

4.3.2.1 Related work

Modern satellites and sensors are capturing huge amounts of data providing the opportunity to monitor any area around the globe for a variety of natural phenomena or changes caused by human intervention, including oil spill detection on aquatic resources. For this, a number of remote sensing techniques have been developed, some of which include synthetic aperture radar (SAR) imagery (Minchew et al. 2012, Krestenitis et al. 2019), and techniques that make use of different portions of the electromagnetic spectrum, e.g. the Infrared (IR), the Near Infrared (NIR), visible and the ultraviolet (Sun et al. 2016; Fingas and Brown, 2014; Fingas and Brown 2017). In cases like oil spill detection in Polyfytos lake at Thessaloniki, Greece we have to deal with an even more challenging situation, because in lakes there are no amplitude waves. Therefore, the usual SAR analysis that apply in oceans where oil spill concentrations can easily be discriminated from the clean wavy water needs to be modified accordingly. To that end, we exploit solutions based on optical data to overcome the aforementioned limitations.

Monitoring our oceans is a huge challenge, from an operational perspective, given that very often oil spills take place in the open ocean and can be the result of deliberate human actions. Additionally, the vast amount of data collected by satellites, require a rapid and efficient analysis technique that will prevent oil spills from spreading to other areas. During the last decade the implementation of Machine Learning algorithms that include Support Vector Machines (SVMs) (Konstantinidou et al. 2019), decision tree forests (Topouzelis and Psyllos 2012; Singha et al. 2013), and the widely used Deep Neural Networks (Guo et al. 2017) provide the automated process for analyzing efficiently the vast remote sensing data collected. By far, the most common technique is the classification of dark patches in SAR images. Spaceborn SAR sensors measure sea surface roughness. Capillary and short gravity waves, that contribute to the surface roughness, are dampened over water surfaces that are covered by oil film, which in turn appear dark in SAR images (Topouzelis 2008). Given that, besides anthropogenic oil spills, the dark patches in SAR images can result from other natural causes (called lookalikes), the challenge is their optimal classification, and for this several methods have been published.

Liu et al. (2010) demonstrated an analysis of variance to extract features based on the geometry grey level and texture of dark areas in SAR images, and used a fuzzy logic algorithm to separate oil spills from lookalikes. Huang et al. (2014) has shown that the combination of Grey-Level Co-occurrence Matrix, used to extract information from SAR images, and the Deep Belief Network does increase the classification performance. Liu et al. (2017) proposed a coarse-to-fine approach to discover and detect suspected oil spills in multitemporal images from global to local scales by following a binary-to-multiple change detection procedure that mainly consists of three steps: 1) multitemporal image preprocessing; 2) coarse oil spill change analysis; and 3) fine oil spill change analysis. Gallego et al. (2018) implemented deep selectional autoencoders and very deep Residual Encoder-Decoder Networks (RED-net) to detect oil spills in Side-Looking Airborne Radar (SLAR) imagery with great accuracy. A recently developed version of the VGG16 Deep Convolutional NN, named OSCNet, for the detection of oil spills based on SAR dark patches has shown promising results by improving classification rate by 2-5% compared to more traditional Machine Learning classifiers (Zeng and Wang 2020). On the other hand, Kolokoussis and Karathanassi (2018) used multi-spectral Sentinel 2 bands for oil spill detection by examining a number of features of the oil spills through Object Based Image Analysis. By using various band ratio they concluded that such an aproach produces increased classification rates.

4.3.2.2 Oil spill detection methodologies

We decided to investigate two different machine learning techniques, SVM[4] and Deep Neural Networks[5] (DNNs) for the detection of oil spills since they have shown high performance in literature. With the SVM we analyse 4 different pixel-based methodologies experimenting with different annotation and bands, whereas with the neural network approach we work with the identification of oil spills either per pixel or per patch level.

There are different ways to run an SVM to detect oil spills, one is pixel-based (pixel-by-pixel), and the other is object based. We are going to focus on the first method. For this we need to create our training and testing set. Both of these will be a group of pixels (some depicting oil spills and the other clean water.

Given that we have seven oil spills (1 during the 25th December and 5 during the 30th December), we run the SVM model 7 times. Each time six oil spills are chosen for training purposes and the 7th is chosen to test the algorithm, to perform a k-fold validation for k=7. Three methods are implemented, and the accuracy of the SVM model is estimated by computing the IoU (Intersection over Union) performance metric.

Method 1 [SVM classification]. We create our training set by grouping the pixels that have been annotated as oil spills via visual inspection (the pixels surrounding the oil spills are not used given the limitations of the visual inspection) together with 400 clear water pixels from an area with no visible oil spills. From this training set the information that is fed in the SVM are the radiation from the Red and the NIR bands along with the binary annotation (0 or 1 for clear surface water and oil spill respectively).

The test set is created in a similar way using the patch from the original Sentinel 2 image that contains the 7th oil spill. In this case all pixels are included, the ones annotated as “1” and the rest annotated as “0”. The annotation of this oil spill represents the ground truth and is compared to SVM prediction once the test set is fed to the SVM model.

Method 2 [RED-NIR threshold identification]. Instead of using the visual inspection to create our training set, we use the information from the Red and NIR bands to annotate the pixels as oil spill or clear water. This is done based on certain thresholds that have to be met for oil spill and clear water pixels. For this we first inspect a Red vs NIR scatter plot (Figure 7) of the areas containing the oil spills, which show us how these bands behave in an oil spill and help us choose the proper thresholds.

[4] https://towardsdatascience.com/support-vector-machine-introduction-to-machine-learning-algorithms-934a444fca47

[5] https://towardsdatascience.com/learning-process-of-a-deep-neural-network-5a9768d7a651

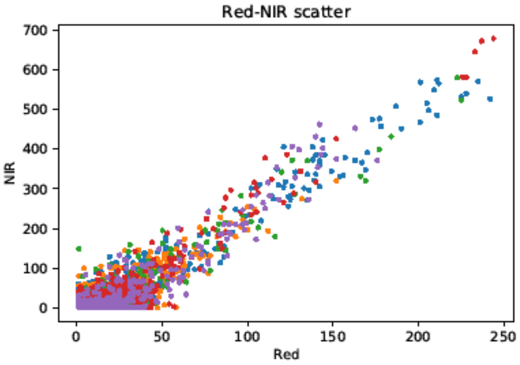


Figure 7. Red-NIR scatterplot. Pixels from patches containing all oil spills have been included. Different colors represent different oil spills.

We annotate the pixels for the training set as oil spills (value 1) or clear water (value 0) by choosing a pair of thresholds for the Red and NIR bands for both classifications (oil and clear water). The NIR-Red scatterplot shows that these measurements follow a linear regression, where NIR increases 2.5 times faster than Red as the oil spill becomes denser. This would suggest that the threshold for NIR would have to be 2.5 times the one for Red, at least for the oil spills. However, this would be counterproductive given that NIR has a high sensitivity for oil spills and can even capture the small ones with very little noise. On the other hand, the Red band is less sensitive and contains a lot more noise. Furthermore, clear water pixels in shallow areas exhibit high brightness in the Red band, which is not the case for the NIR channel, which appear to have zero brightness. It should be noted, that the latter is probably also the result of low wind conditions, and that under higher winds this might not be the case. Overall, we had to choose a relatively low threshold for NIR (15) and a higher for Red (50). All pixels that had brightness bigger than these values were annotated as 1 (oil spill). For the clear water pixels, the thresholds are 8 and 30 for the Red and NIR bands, respectively, and all pixels that had a brightness smaller than these values were annotated as 0. In contrast, the threshold method is not used to create the annotation for the test set. It is very important to note that the annotation must be done via an independent method, hence leading to the visual inspection of our TCI image.

Method 3 [Index-based threshold identification]

An additional attempt was made by using the other two Sentinel-2 bands[6] from the visible spectrum (B2 - Blue and B3 – Green with wavelengths of ~493nm and ~560nm respectively) despite the fact that they do not detect the oil spills. This is done to augment the oil-clear water contrast of the Red and NIR bands, which is achieved by using the ratios

$$\frac{B2-B4}{B2+B4} \quad \text{and} \quad \frac{B2-B8}{B2+B8}$$

, in case we use the B2 band. So instead feeding into the model the Red and NIR brightness, we feed in the same way the above ratios, where B2 is the Blue band (493nm) and B4 is the Red band (665nm). Everything else is similar.

Deep Neural Network (DNN)

Pixel classification

[6] https://sentinel.esa.int/web/sentinel/user-guides/sentinel-2-msi/resolutions/radiometric

On the same Oil Spill dataset as in the SVM approach, we implement a Deep Neural Network (DNN) with two hidden layers and an output layer. The hidden layers have 12 and 8 units, and they use a Relu activation. The output layer uses a sigmoid activation and has only two output classes. The DNN is run with different learning rates (0.0001, 0.001, 0.01 and 0.1), a batch size of 10 and with 100 epochs. Given the fact that every time we run the model the weights of the Neural Network are randomly initialised, leading to a different result, we run the model 10 times with the same the combination of hyperparameters. At the end we estimate the average of the metrics used.

Patch classification

We attempt to tackle the oil spill detection task as a binary classification problem by predicting the presence of oil spill within smaller patches of the initial image. Different well known convolutional neural network architectures were tested (VGG16, ResNet50, Inception v3) with the VGG16 providing the best results. We perform fine-tuning on ImageNet dataset by retraining the four top layers of the VGG16 network. We also modified the last layer that uses a SoftMax activation in order to have only two output classes (oil and water) instead of the 1000 ImageNet classes. For the input we test with either RGB or composite RGB images that consist of more appropriate for the task Sentinel-2 bands.

4.3.2.3 Dataset description

The first step in the creation of the train and test dataset for the pixel classification approach is to annotate our oil spills via visual inspection of the Sentinel 2 RGB (TCI) images, where oil spills appear as orange-brown (dataset A). For this we choose small patches of the original Sentinel 2 image that each contains an oil spill, and we set 1 for pixels clearly depicting oil spills and 0 for the rest. Because of this, a considerable percentage of pixels clearly depict oil spills and are easily annotated. On the other hand, pixels corresponding to much thinner oil spills, usually on the vicinity of the visually detected oil spills, exhibit a much smaller color contrast with clear water pixels, and therefore the efficiency of the visual annotation is limited. However, these pixels are annotated as 0s along with the pixels depicting clear water. In total 332 pixels are annotated as oil spills. The training set is comprised from the pixels annotated as oil spills and a roughly equal number of pixels that are chosen from a different area (away from the oil spills) that has no visible oil spills and therefore are annotated as clear water.

For the DNN per pixel approach the same dataset as the SVM method was used.

For the DNN per patch methodology we create another dataset that consist of PNG images based on specific Sentinel-2 bands that appear more appropriate for the task since the casual RGB images make it difficult to discriminate oil spills from water (dataset B).

For the dataset, the composite images consist of a combination of Nir, Green & Swir Sentinel-2 bands (i.e. Bands 8, 3 & 11). With this combination water appears transparent in contrary to oil and other objects. In here, we scale the radiance values of the Sentinel-2 bands from 0-1024 to 0-255. With this scale the oil spills, shores and other objects that are present in the lake are highlighted in a pink color.

The RGB and false composite images provide different visualizations of the underlying areas that contain oil spills (Figure 8 and Figure 9). In here the oil spill / clean water discrimination ability of the composite images against the RGB representation is evident, where at many cases the oil spills are difficult to spot in the RGB images.

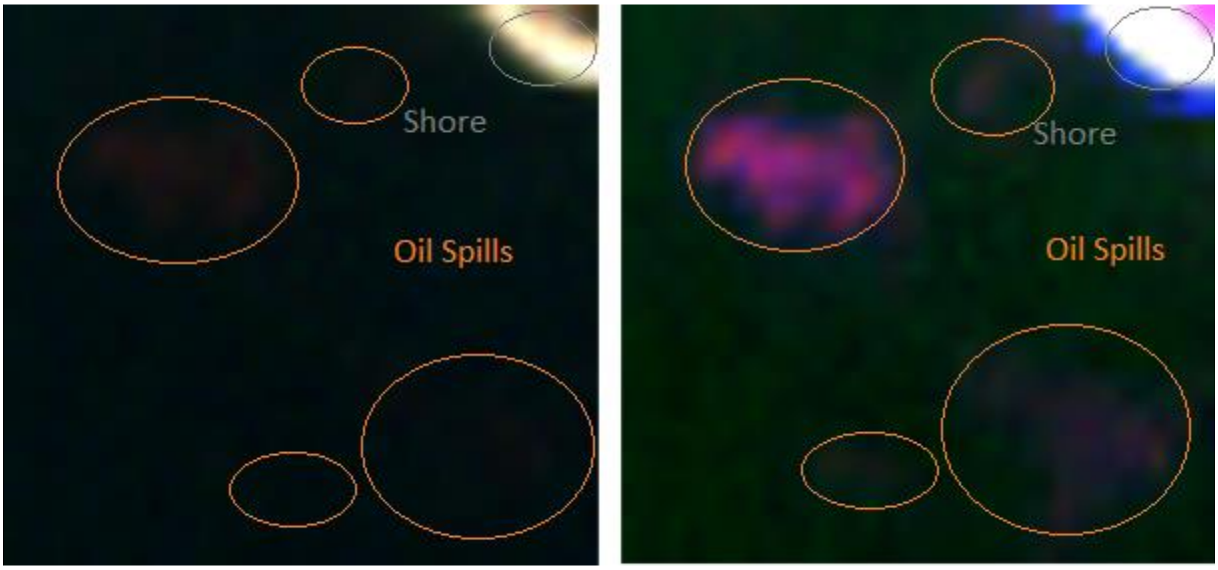


Figure 8. Oil spills visualization in RGB (left) and false composite (right). On the top right of each image some permanent shore can be seen.

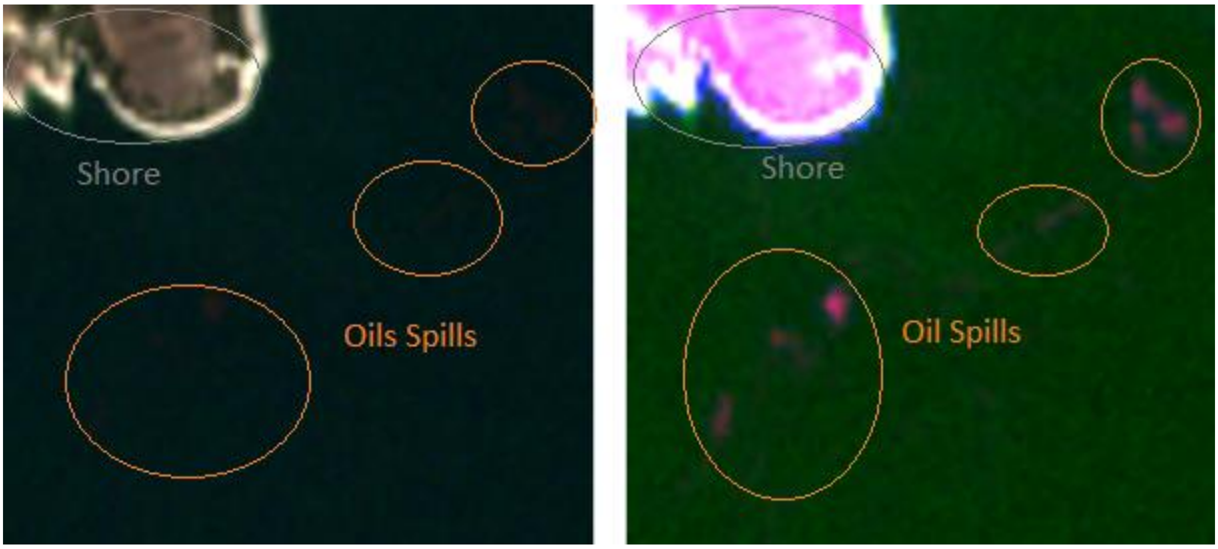


Figure 9. Another oil spills visualization in RGB (left) and false composite (right). On the top left of each image some permanent shore can be seen.

To create the PNG images, we split the whole Sentinel-2 image in a grid with square size of 56x56 pixels. Since we are interested only in the areas in the lake, we generated a shapefile that depicts the contour of the lake and used it to crop and keep only the main lake water bodies (Figure 10) and then discard the square-patches that fall outside of it, keeping 249 patches of the Sentinel-2 image.

For the training set we used samples of the oil spills that appeared on 30 December 2017. Since the oil spills are seen on only 9 patches of that day, we manually created a dataset based on these samples. We use the image on 20/12/2017 to produce 249 patches with clean water. Then these patches are used as a template, then we manually insert randomly shaped oil spills that resemble the ones on 30/12/2017, resulting to a balanced dataset of 249 clean patches and 249 patches with oil spills, that right after we resize to 224x224 pixels which is the input size of the pretrained with ImageNet VGG16.

Additionally, we apply data augmentation to increase the dataset by rotating, flipping and changing the brightness and contrast of the patches resulting to an increase of the aforementioned dataset by a factor of 24, generating a total sum of 11952 patches.

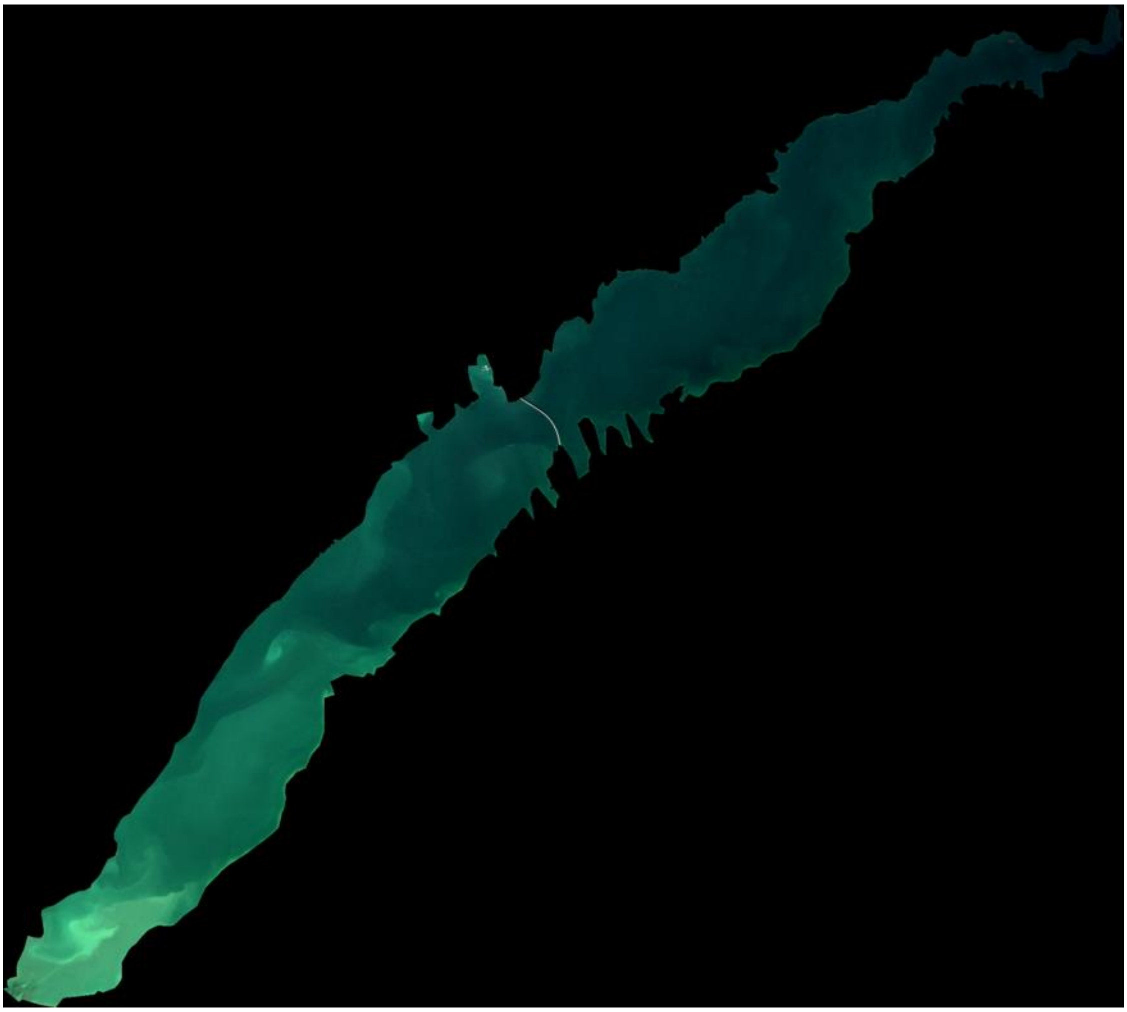


Figure 10. Polyfytos lake on 30/12/2017 after trimming the shore areas.

4.3.2.4 Evaluation of the proposed methodologies

SVM method

From now on, OS1 represents the experiment where oil spill 1 is used for testing purposes and the rest are used for training (the same for OS2, OS3 ctr). Our results show that the most accurate SVM configuration is the 5th order Polynomial and the least is the RBF. Regarding the polynomial option, the accuracy increases as the order of the polynomial becomes higher. Methods 1 and 2 (Table 3 and Table 4) have comparable results in general, but Method 1 has a slightly increased accuracy compared to Method 2 when the order of the polynomial is small (n=1, linear) and exactly the opposite happens when the order of the polynomial is high (n=5). Note that in computing the averages, the results regarding OS7 were not included. Figure 11 shows Oil Spills 1 and Figure 12 shows oil spill 7 as seen through our analysis.

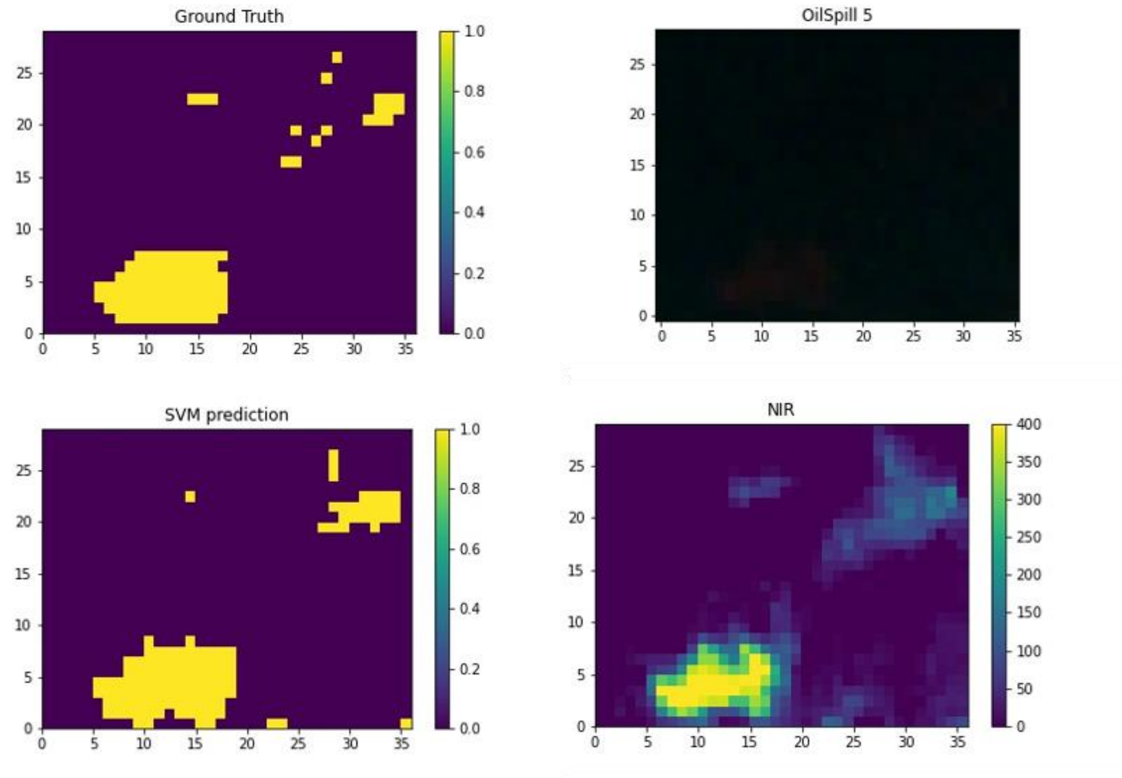


Figure 11. Oil Spill OS1 (30th Dec) method 2. Oil mask from visual annotation (upper left) and SVM prediction (lower left). NIR brightness for (lower right) and TCI (upper right).

Also, the SVM model fails to capture the 2nd Oil spill of Dec 25 (OS7) in the linear and RBF configuration by predicting that all pixels are oil spills. This specific oil spill is different compared to the other ones because it is most likely thinner and smaller, it is located in shallower waters, and the visual annotation revealed only a small number of pixels as oil spills. However, when the 5th polynomial is employed it seems that the SVM model captures only a part of the oil spill. Inspecting the predicted oil mask, it seems that the SVM model overpredicts the oil spill (it detects more pixels as oil spill compared to the ground truth). However, when comparing these to the TCI and the NIR image, it seems that the SVM model captures the oil spill correctly, and that the ground truth is the one that underrepresents the oil spill. We observe that the visual annotation is not successful in capturing the oil spill margins, most likely due to the fact that it is a thin oil spill exhibiting only a small brightness on the RGB bands.

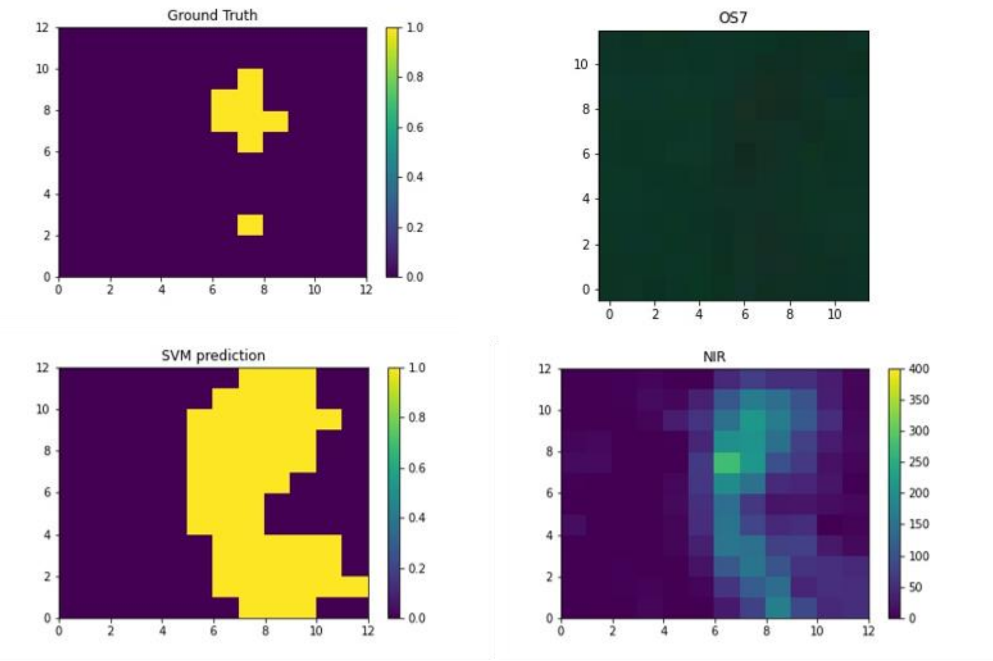


Figure 12. Similar to figure 1 but for Oil Spill OS7 (25th Dec).

In general, the SVM model overpredicts the number of oil spill pixels for all oil spills. This is not only due to the limitations of the visual annotation, but also because these pixels are picked up by the model because the model is trained based on the NIR band, which is highly sensitive.

Applying the SVM model using Method 3 did show an improvement when applying the linear option compared to the previous methods (Table 5). However, when the higher order polynomial options were chosen the model would either predict all pixels as clear water or oil spill (clearly showing that the model is not successful), and therefore the resulting IoUs are not shown.

	OS1	OS2	OS3	OS4	OS5	OS6	OS7	Average
Poly-1 (linear)	78.61	96.32	87.79	83.24	90.21	79.64	-	85.97
Poly-3	88.95	98.52	94.55	94.08	95.68	84.61	34.57	92.73
Poly-5	92.61	99.08	96.46	95.71	97.26	88.08	53.19	94.86
RBF	50.97	87.56	65.56	52.17	76.38	60.85	-	65.58

Table 3. IoU results of method 1

	OS1	OS2	OS3	OS4	OS5	OS6	OS7	Average
Poly-1 (linear)	79.53	96.50	88.85	85.51	91.70	81.26	-	87.22
Poly-3	88.27	98.45	94.36	94.08	95.84	84.04	16.13	92.51
Poly-5	90.85	98.85	95.88	95.53	96.47	86.33	43.28	93.98
RBF	50.97	87.56	65.56	52.17	76.38	60.85	-	65.58

Table 4. IoU results of method 2

	OS1	OS2	OS3	OS4	OS5	OS6	OS7	Average
Poly-1 (linear)	80.94	98.13	91.75	92.30	95.37	81.27	71.43	87.31
Poly-3	-	-	-	-	-	-	-	
Poly-5	-	-	-	-	-	-	-	
RBF	41.56	58.85	33.33	24.63	41.78	44.93	-	40.85

Table 5. IoU results of method 3

During training, the DNN achieves scores (measured with several metrics like accuracy, precision, recall, and f-score) above 99%. However, the same thing cannot be said about the testing, where some of these metrics are reduced considerably.

One interesting thing about the prediction of the DNN, compared to the one of the SVM, is that even when we implement the IoU as a metric, it is considerably reduced. The IoU metric is defined based on the ground truth, and as we mentioned earlier, is subject to the limitations of the visual inspection. However, comparing the DNN prediction to the NIR band, we can say that the network captures most of the oil spill, even though this is not fully captured in the visual annotation. This means that the network learns from the annotated but limited pixels, and transfers this while performing the prediction. Eventually, this translates in a weaker prediction but only because a large portion of the oil spill can't be identified visually. This is perfectly captured in the figure below (Figure 13), where oil spill OS1 as seen in the NIR band matches perfectly with the predicted one by the DNN but is only partially captured by the SVM prediction. The same holds for the rest of the oil spills, except OS7 for the same reasons mentioned in the SVM section.

Additional experiments were performed using a different optimizer (SGD) and learning rates and the results are summarized in (Table 6, Table 7and Figure 14).

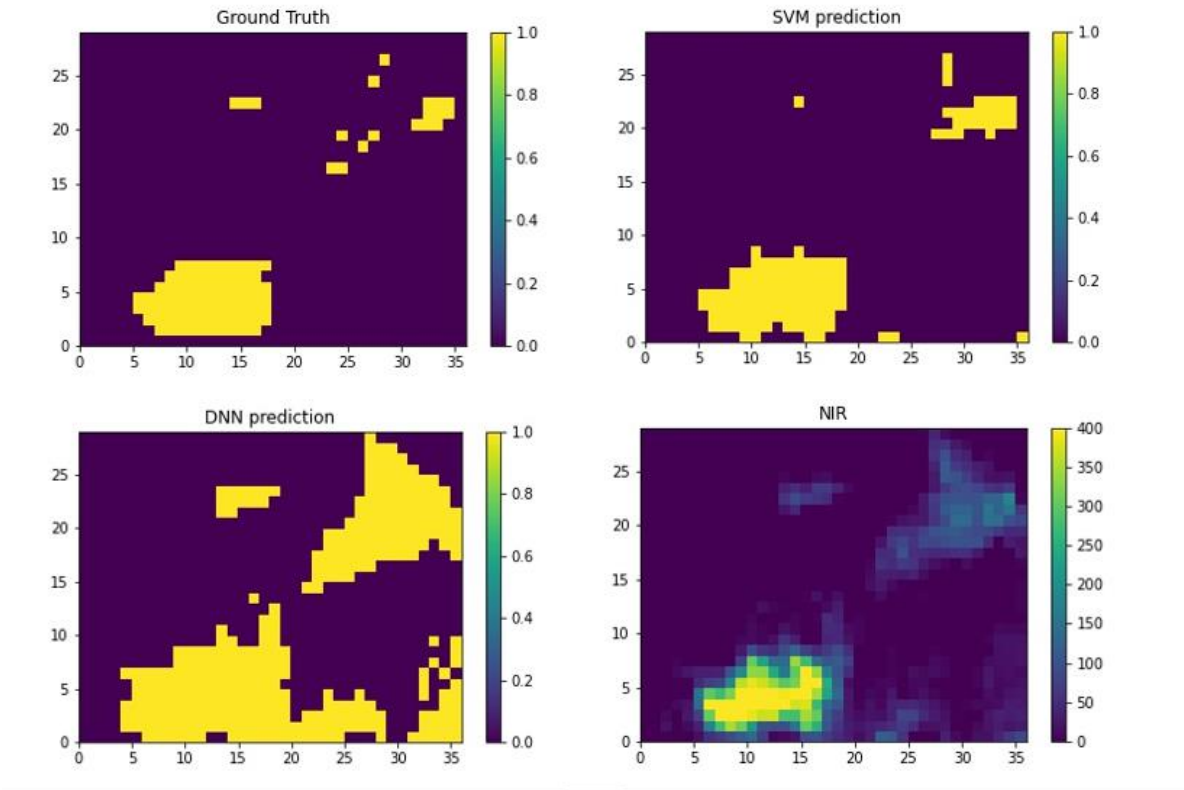


Figure 13. Oil Spill OS1 (30th Dec) method 2. Oil mask from visual annotation (upper left) SVM prediction (upper right) and DNN prediction (lower left). NIR brightness (lower right).

OPT	LR	OS1	OS2	OS3	OS4	OS5	OS6	OS7	Average
Adam	0.1	59.51 (2.8)	89.89 (9.31)	77.10 (3.48)	65.89 (4.52)	82.68 (2.38)	71.19 (2.06)	43.96 (21.50)	70.03 (6.58)
	0.01	69.69 (4.64)	91.84 (1.31)	78.03 (1.17)	67.39 (1.52)	82.24 (1.96)	73.32 (1.72)	-	77.09 (2.03)
	0.001	68.56 (3.64)	93.15 (0.95)	78.91 (1.27)	69.59 (2.44)	84.65 (1.66)	74.03 (1.62)	-	78.15 (1.93)
	0.0001	60.43 (4.07)	91.02 (0.71)	78.00 (2.46)	66.03 (4.59)	82.68 (1.71)	71.47 (2.65)	64.55 (11.56)	73.45 (3.96)
SGD	0.1	66.33 (7.19)	93.61 (2.77)	83.19 (7.08)	63.77 (2.38)	75.19 (20.11)	72.72 (1.37)	-	75.80 (6.81)
	0.01	66.57 (4.97)	92.07 (1.00)	76.31 (2.00)	65.45 (4.56)	83.54 (1.27)	72.59 (2.46)	33.14 (1.26)	69.95 (2.50)
	0.001	56.93 (2.19)	90.15 (0.45)	77.34 (4.67)	57.59 (13.06)	80.84 (0.95)	70.56 (0.67)	57.93 (16.78)	70.19 (5.53)
	0.0001	32.18 (19.17)	71.56 (32.75)	69.03 (13.25)	51.30 (16.37)	48.74 (26.91)	51.09 (19.70)	69.41 (13.59)	56.18 (20.25)

Table 6. Results after comparing the ground truth with the DNN prediction, for the 7 oil spills, using the IoU as a metric. The results correspond to a batch size of 10.

OPT	LR	OS1	OS2	OS3
Adam	0.0006	69.51	93.63	78.98
	0.0008	68.83	93.02	78.56
	0.001	68.56	93.15	78.91
	0.002	69.07	93.15	78.51
	0.004	69.33	92.53	78.37
	0.006	71.17	92.51	77.58
	0.008	64.07	92.59	76.39
	0.01	69.69	91.84	78.03
	0.02	66.53	92.00	76.66
	0.04	62.65	91.12	75.51

Table 7. Same as Table 1, but with more learning rate options.

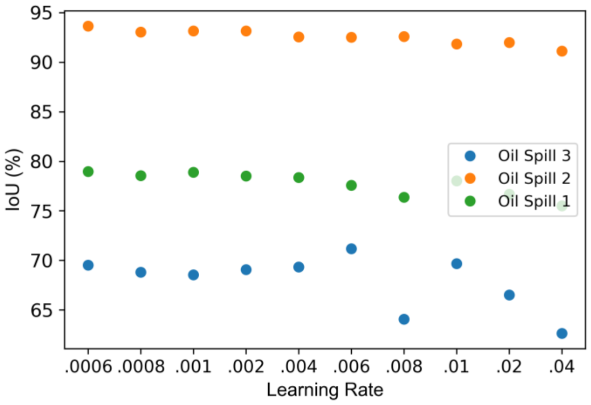


Figure 14. Plotting the results from Table 5.

Deep Neural Network (DNN) per patch method

Hyper parameter tuning performed on the network with the dataset B and provided the optimal training / validation scores with Adam optimiser, learning rate of 0.00005, batch size of 10 and 30 epochs (Figure 15). Combining with the balanced and adequate in size training dataset would allow us to proceed with the prediction phase.

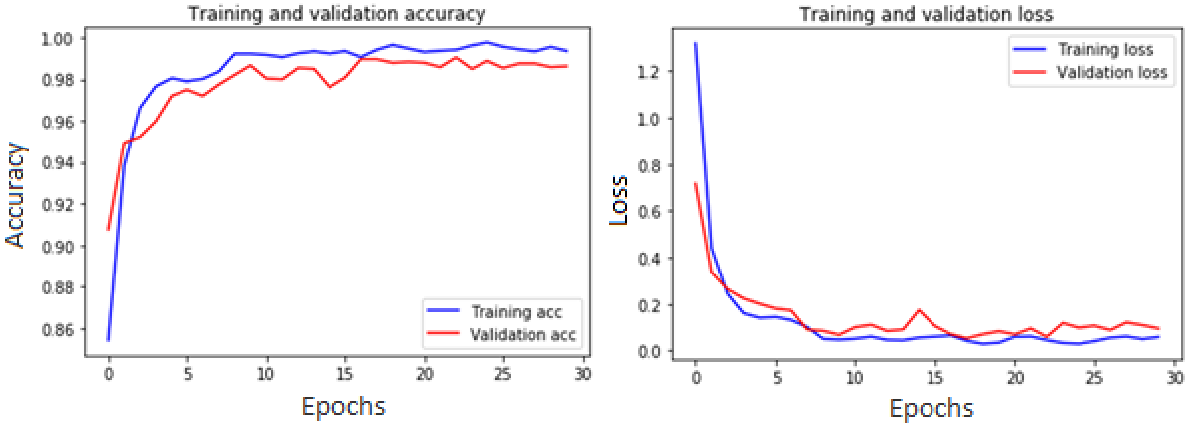


Figure 15. Training and validation scores for the VGG16.

Since we only have data for one incident, we tested the model on the Sentinel-2 image (S2B_MSIL2A_20171230T092359_N0206_R093_T34TEK_20171230T115949) of the 30/12/2017. The image was split into patches with similar way as descripted at 4.3.1.3 and then was fed to the trained model for prediction. The model managed to identify 7 of the 9 oil spill patches, whereas it gave three false positives, close to the shores in the bottom side of the middle of the lake (Figure 1). The confusion matrix and some basic metrics of the testing can be seen at Table 8 and respectively.

	Predicted Water	Predicted Oil
Actual Water	237	3
Actual Oil	2	7

Table 8. Confusion matrix with the oil and clean water predictions for the 30/12/2017 incident at Polyfytos lake with the augmented balanced composite patches model

Accuracy	Misclassification	Precision	Recall	F-Score
0,98	0,02	0,7	0,777	0,737

Table 9. Various metrics on the predictions for the 30/12/2017 incident at Polyfytos lake with the augmented balanced composite patches model

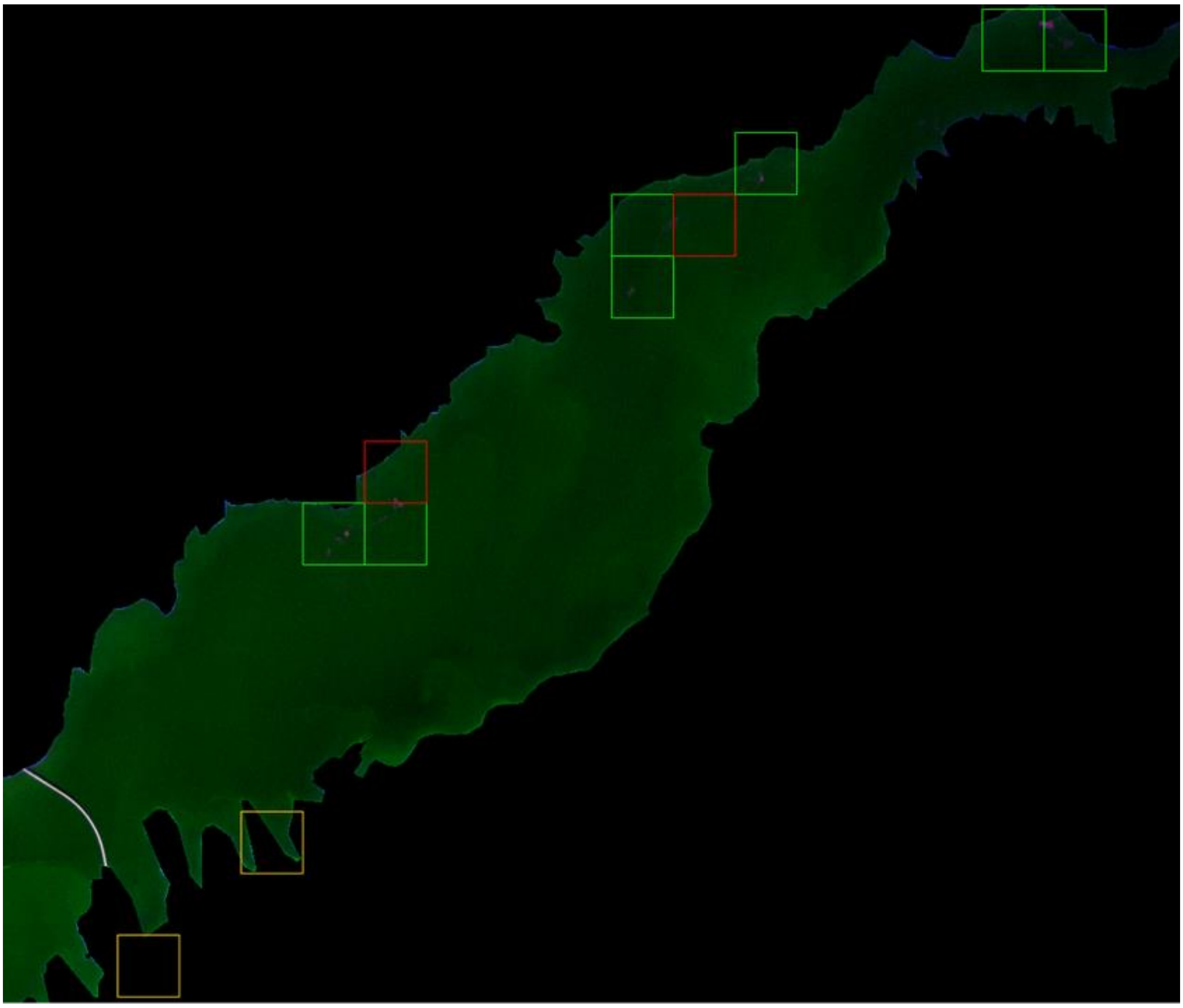
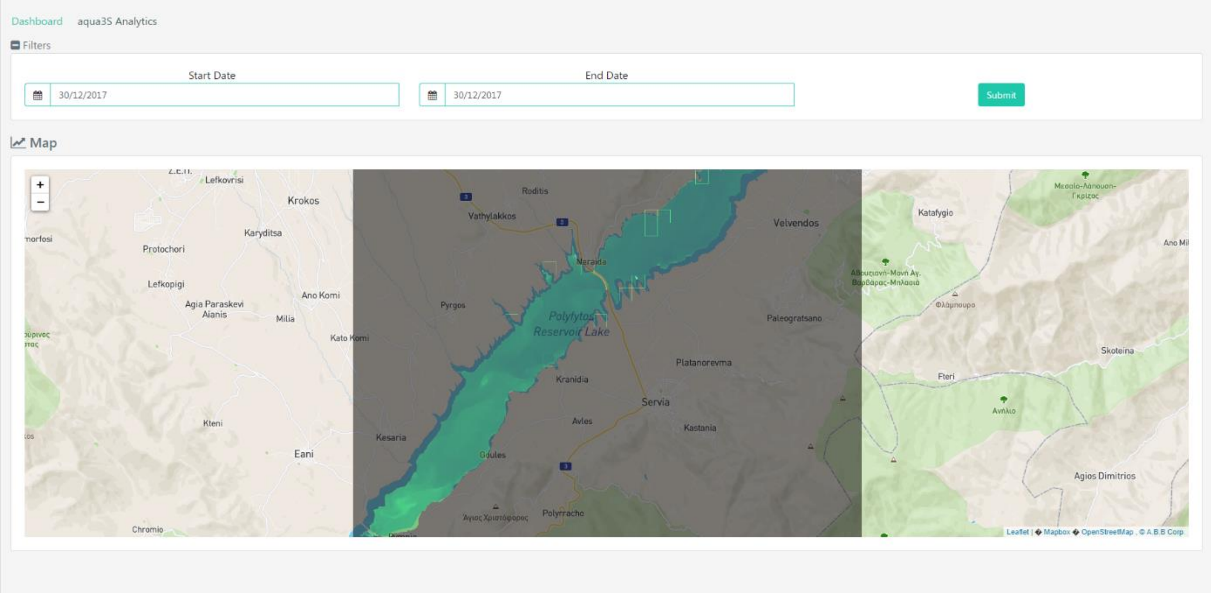


Figure 16. Output of the oil spill prediction module on product S2B_MSIL2A_20171230T092359_N0206_R093_T34TEK_20171230T115949. With green the true positives. With red the false negatives. With yellow the false positives.

4.4 Visualisation of generated pollution maps

The satellite data that are collected and analyzed inside the aqua3S project can be currently visualized in Visual Analytics Module. This module provides a clearer picture of the geographical extent of potential issues, e.g. an incident of water pollution or a flooding event. A screenshot can be seen in Figure 17, where the user selects the date or the period of interest, clicks on the "Submit" button and then the map is updated with the processed image of the selected date of Polyfytos lake, where green rectangles indicate the possible presence of an oil spill. Zoom and free navigation features are supported.

The Visual Analytics Module will be described in more detail in D5.1 (M18).



Dashboard aqua3S Analytics

Filters

Start Date: 30/12/2017

End Date: 30/12/2017

Submit

Map

Figure 17. Screenshot of the Visual Analytics Module depicting oil spill incident at Polyfytos lake on 30/12/2017

5. Social media monitoring

With the wide adoption of social media in the daily life of billions of people worldwide, it can be argued that published content by users is able to reflect timely the general sentiment of the crowd and what is happening anywhere in the world. As nowadays any topic can be covered by social media posts, water safety and security are another subject that is expected to concern online users. The Social Media Monitoring task is responsible for the acquisition of such social media data, offering an alternative source of information to the aqua3S system and targeting to support the creation of social awareness in a water distribution network.

This chapter is dedicated on the work that has been done so far in the Social Media Monitoring task, which will be supplemented in deliverable D3.6 (M27). First, an overview of the framework will be presented (5.1) in order to provide the reader with the overall picture of the implementation. The acquisition of the social media data will follow, involving the means to crawl Twitter (5.2.1), the definition of the search criteria that crawling will be based on (5.2.2), the processing, storing and indexing of collected posts (5.2.3) and the description of the current collection (5.2.4). The next subchapter refers to the analysis of social media data in order to extract knowledge and includes the estimation of the posts' reliability (5.3.1), the geotagging module (5.3.2), the nudity detection approach (5.3.3), and the text classification (5.3.4) to determine the relevance of a post. The latter is presented to a greater degree, because there has been significant effort to adapt the methodology for the project's purposes. Finally, two different Web interfaces that visualise the collected and analyzed social media data are demonstrated (5.4).

5.1 Overview of the framework

The Social Media Monitoring framework aims to collect, analyze and push into the aqua3S system social media data and specifically posts from Twitter, the so-called tweets. The complete workflow of this framework is illustrated in Figure 18 and described here.

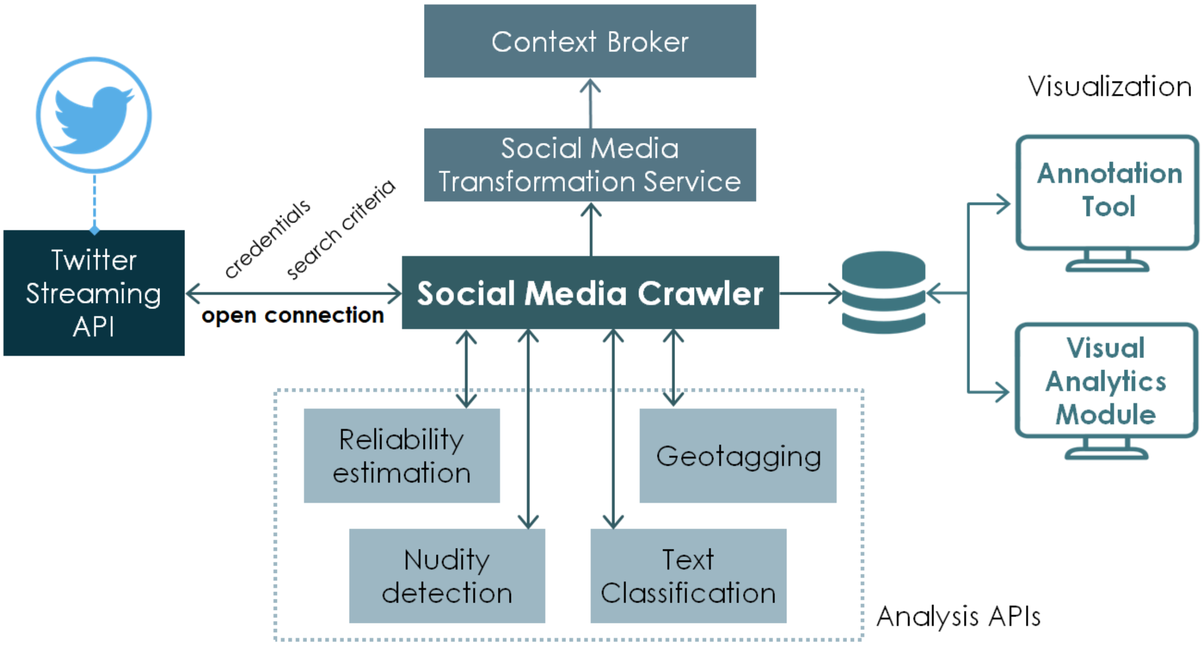


Figure 18. The complete workflow of the Social Media Monitoring framework

The core component of the framework is the Social Media Crawler, displayed in the middle of the figure, which handles every step of the workflow. Initially, the Crawler uses the necessary credentials to establish an open connection to the Twitter Streaming API and formulates a complex query in order to receive in a real-time manner tweets that satisfy this query.

For every single tweet received, a four-step analysis procedure is performed by calling the respective APIs. The analysis involves: (i) the estimation of the reliability of a tweet so as to identify fake news, (ii) the detection of nudity in the tweet's image (if existing) to avoid users getting frustrated because of pornographic material, (iii) the automatic geotagging of a tweet based on locations mentioned in its text, and (iv) the classification of a tweet as relevant or not to the aqua3S use cases.

After the analysis of a single tweet is concluded, the results are added to its existing metadata and the complete information is stored to a MongoDB database. Two Web interfaces that serve different scopes connect to this database and visualise the tweets: the first one is the Annotation Tool, which is used for collecting human annotation (read more in 5.3.4.3) as well as demonstration reasons, and the second one is the Visual Analytics Module, which displays visual information from heterogeneous sources on an interactive map, including social media images.

Apart from storing the analyzed tweets in the database, the Crawler also produces a minimised version of each tweet (keeping only some fundamental attributes) and sends it to the Social Media Transformation Service, which subsequently converts the tweet information into an appropriate format and inserts it into the Context Broker, thus pushing it into the aqua3S system.

5.2 Social media data acquisition

5.2.1 Crawling from Twitter

For the Social Media Monitoring task, we have decided to focus on the popular social media platform of Twitter. By the second quarter of 2020, Twitter counted 186 million daily active users worldwide[7]; a clear indication of the platform's high popularity, which promises plentiful and up-to-the-minute crowdsourcing information.

Our preference for Twitter is further reinforced by the fact that it offers a great variety of API endpoints for retrieving tweets. Amongst these available APIs, the most suitable for our objectives is the free of charge Twitter Streaming API, which permits access to Twitter's public stream of data and retrieves tweets nearly at the moment they are published. In order to use the Streaming API, it is necessary to create a Twitter account and then to apply for a developer account, providing a description of the projected usage. Once the request is approved, the following credentials are created: *Consumer Key*, *Consumer Secret*, *Token*, and *Token Secret*. A successful connection to the API requires all these four access tokens.

To determine what kind of tweets should be retrieved from the data stream, the Streaming API provides three filtering options that can be used separately or combined. The "Follow" option retrieves tweets that are posted by specific user accounts, by defining a list of user IDs. The "Track" option brings tweets that contain one or more keywords inside their text, while the "Locations" option retrieves tweets that are posted from a specific area, by providing a set of bounding boxes (a bounding box can be defined as two pairs of coordinates - one for the southwest corner of the box and one for the northeast corner). It

[7] https://www.statista.com/statistics/970920/monetizable-daily-active-twitter-users-worldwide/

should be noticed that the free access allows up to 5,000 user IDs, 400 keywords, and 25 bounding boxes.

In order to start consuming from the Twitter Streaming API, the Social Media Crawler uses the Hosebird Client[8], an open-source Java HTTP client that is able to establish an open connection to the endpoint and receive new messages every time a newly published tweet matches the predefined filtering options. As input the Hosebird Client gets the aforementioned access tokens and the filtering parameters (lists of keywords, user IDs, bounding boxes), while the output for each retrieved tweet is a JSON string. We choose to keep the proposed JSON format, given that it is very flexible to add more attributes, such as the results of the knowledge extraction, and it is also indicated for MongoDB databases, which we prefer.

5.2.2 Defining the search criteria

As explained in the previous subchapter, the social media content that is consumed from the Twitter Streaming API totally depends on the filtering parameters that are fed as input to the API, i.e. the search criteria. The definition of these criteria in the aqua3S project has been achieved in close collaboration with the PUC leaders, since the searching parameters should reflect the needs of each use case and lead to posts that are valuable to the end users.

The main topic to be monitored on social media is water safety and security. Out of the three filtering options only the "Track" option is used, so a keyword-based search is realised. Since this option is language-dependant, we have selected the following languages, which relate to the areas of interest of each PUC: Italian for PUC1, Greek for PUC4, French for PUC5, Bulgarian for PUC6 and English for all of them (the PUCs that are not mentioned will not exploit social media data). For every language a different set of keywords (a keyword can comprise more than one words) has been suggested by the PUC leaders and can be seen in Table 10. In case that a keyword exists in several languages, it appears in the same line. Strikethrough text indicates keywords that had to be removed after an examination of the first search results, e.g. the word "Trieste" that brought a very large number of retrieved tweets that mention the city but in a context totally irrelevant to water quality.

Apart from the topic of water safety and security, the leaders of PUC1 and PUC6 suggested to monitor also the topic of floods and droughts. Thus, two more sets of keywords have been created and can be seen in Table 11. Again, keywords that exist in both languages appear in the same line and strikethrough keywords have been removed, e.g. the word "Corno" that eventually was not related only to the city but also to adult content.

English (all PUCs)	Italian (PUC1)	Greek (PUC4)	French (PUC5)	Bulgarian (PUC6)
aggressive water	aggressività acque		eau agressive	
aquifer	falda aquifera			
chlorine water	cloro acqua		chlore eau	
corrosive water	corrosione acque		eau corrosive	
deposit water	deposito acqua		dépôt eau	
	deposito calcare			

[8] https://github.com/twitter/hbc

drinking water	acqua potabile			вода за пиене
	pozzi idropotabili			
groundwater	acque sotterranee			
irritant water	irritante acqua		eau irritante	
Isonzo river	fiume Isonzo			
Karst	carso			
not drinkable water	non potabile acqua			
Moschenizze	Moschenizze			
muddy water	acqua fangosa			мътна вода
particle water	particella acqua		particules eau	
	falda pressione			
Sablici	Sablici			
sand water	sabbia acqua		sable eau	
Sardos	Sardos			
smell water	puzza acqua			мирис на водата
spill water	sversamento acqua			
tap water	acqua rubinetto			чешмяна вода
Timavo	Timavo			
~~Trieste~~	~~Trieste~~			
undrinkable water	non bevibile acqua		imbuvable eau	
~~water bottle~~	bottiglia acqua			
water color	colore acqua	νερό χρώμα	eau colorée	
water contamination	contaminazione acqua			
water coloured	acqua colorata			
water not clean	acqua non pulita	νερό δεν είναι καθαρό νερό βρώμικο		
water not clear	acqua non chiara	νερό θολό		
water pollution	inquinamento acqua			
water pumping	emungimento acqua			
water odor water odour	odore acqua	μυρωδιά νερό μυρωδιά νερού	odeur eau	
water quality	qualità acqua		eau qualité	качество на водата
water supply	fornitura acqua			
	condotta alimentazione acqua tubazione acquedotto			
	sistema approvvigionamento acqua			
water table	superficie falda acquifera			
water taste	sapore acqua	γεύση νερού γεύση νερό	goût eau	вкус на водата
water trouble	problemi acqua		eau trouble	
Sofian water				Софийска вода
water fish smell				вода мирише риба

Iskar dam pollution				замърсяване яз.Искър
discharge Iskar				зауставане Искър
sewerage Iskar				канални води Искър
stone water			pierres eau	
water illness			maladie eau	
cramp water			crampes eau	
diarrhea water			diarrhée eau	
vomiting water			vomissements eau	
gastrointestinal disorder water			troubles gastrointestinales eau	
skin irritation water			irritation peau eau	

Table 10. Search criteria for the topic of water safety and security

Italian (PUC1)	Bulgarian (PUC6)
alluvione	Наводнение
	Преливане
siccità	Суша
	Разлив
alluvione Isonzo	
alluvione Trieste	
sotto acqua	
sott'acqua	
allagamento	
allagato	
allagata	
Acqua alta	
fiume in piena	
fiume alto	
fiume grosso	
fiume ingrossato	
esondazione	
breccia arginale	
tracimazione	
tracimazione spondale	
sottopasso allagato	
livello fiume	
~~Trieste~~	
Fiume Isonzo	
falda	
falde prosciugate	
prosciugamento falda	
livello falda basso	
siccità Trieste	
allerta meteo	
allerta meteo Trieste	
pozzo prosciugato	
pozzo vuoto	

carenza idrica	
crisi idrica	
scarsità di risorse idriche	
scarsità d'acqua	
serbatoio vuoto	
livello serbatoio basso	
livello pozzo basso	
sorgente prosciugata	
Salcano	
Gorizia	
Madonnina del Fante	
Doberdò	
Doberdò del Lago	
San Dorligo della Valle	
Dolina	
San Dorlingo della Valle - Dolina	
Monrupino	
Sgonico	
Noghere	
Barcola	
Fogliano	
Redipuglia	
Fogliano Redipuglia	
Monfalcone	
Miramare	
Ronchi dei Legionari	
San Giovanni di Duino	
Duino	
Aurisina	
Duino-Aurisina	
Sistiana	
Muggia	
Rupa	
Gabria	
Savogna d’Isonzo	
Sagrado	
Gradisca d’Isonzo	
Saletto	
Poggio Terza Armata	
Sardos	
Farra d’Isonzo	
San Canzian d’Isonzo	
Turriaco	
Fiumicello	
Ruda	
Sablici	
Pietrarossa	
~~Corno~~	
Lisert	

Polosko	
Moschenizze	
Moschenizza	
Randaccio	
San Pier d'Isonzo	
Campagnuzza	
Staranzano	
Timavo	
Settefontane	
torrente Chiave	
rio Orsenigo	
Marchesetti	
Rosandra	
Ospo	
rio Chiarbola	
rio Baiamonti	
rio Primario	
rio Storto	
Zaule	
rio Spinoleto	
rio Mercese	
torrente S.Antonio	
torrente Sant'Antonio	
Fugnan	
Pisciolon	
Farnei	
Rabuiese	
Menariolo	
Elleri	
Vipacco	
Groina	
Piumizza	
canale dei Dottori	
canale Principale Dottori	
canale Dottori	
canale secondario I di Bonifica	
canale Principale III di Bonifica	
canale dei Grigi	
roggia della Risaia	
roggia dei Boschi	
roggia Fogliano	
sorgenti di Turriaco	
canale Chiarodici	
canale San Pietro	
canale S.Pietro	
Brancolo	
canale Fiumicino	
roggia Fiumincino	
canale Macorina	

roggia di San Canziano	
roggia dei Clici	
roggia del Molino	
Quarantia	
canale della Quarantia	
Isola della Cona	
Golfo di Panzano	
Punta Barene	
Consorzio di Bonifica Pianura Isontina	
Irisacqua	
AcegasApsAmga	
Autorità di bacino distrettuale delle Alpi Orientali	
Seng	

Table 11. Search criteria for the topic of floods and droughts.

5.2.3 Processing, storing and indexing

After a new tweet has been collected, a two-part processing is applied before the tweet is stored or forwarded into the system. The first part involves the extension of the JSON, which is provided by Twitter per each tweet, with additional attributes based on original Twitter attributes, in order to have later simpler and faster queries to the database.

Before presenting these attributes, it is necessary to note that three types of tweets can be identified: *General tweets*, *Extended tweets*, and *Retweets*. General tweets are the regular messages posted to Twitter containing text and/or images, extended are the tweets that encapsulate messages longer that the original 140-character limit and retweets are re-postings of other tweets. An example of each type's JSON is given in Appendix II, since they contain a very large number of attributes.

The first appended attribute is named *is_retweeted_status* and its value is true when the original JSON contains the field *retweeted_status*. Having a Boolean attribute, it is much easier to retrieve the retweets (true) or the general tweets (false).

The next attribute is named *full_text* and contains the complete message of the tweet, which is possible to find in the fields *text*, *extended_tweet.full_text*[9], *retweeted_status.text*, and *retweeted status.extended_tweet. full_text*. Having detected the complete text a priori, there is no need to check all these fields when querying.

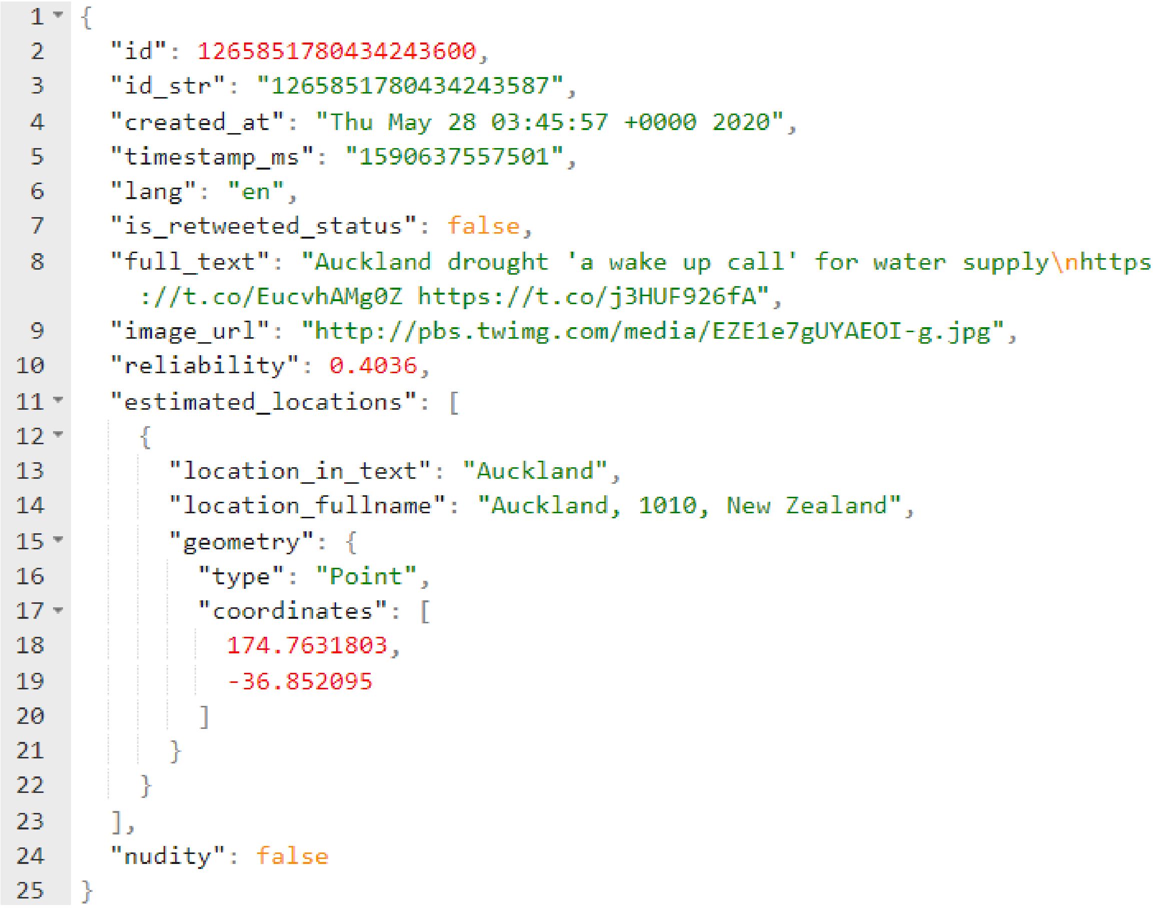
Similarly, the attribute *image_url* contains the link to an image attached to the tweet, which can be found in *entities.media.media_url*, *extended_tweet.entities.media.media_url*, *retweeted_status. entities.media.media_url*, *retweeted_status.extended_tweet.entities.media.media_url*.

The second part of extending the original JSON involves the addition of attribute-value pairs that refer to the outcomes of the various knowledge extraction analyses. The Double attribute *reliability* contains the score of the reliability estimation (5.3.1), the attribute *estimated_locations* is an array of detected locations (5.3.2), and the Boolean attribute *nudity* is true when there is possibility of nudity depicted in the tweet's image (5.3.3).

[9] Dot in JSON fields should be perceived as nesting, e.g. *extended_tweet.full_text* means that the field *extended_tweet* is an object that contains the field *full_text*.

To sum up, **Error! Reference source not found.** displays the abovementioned additional attributes (lines 7-24) as well as some fundamental original attributes (lines 2-6). The depicted JSON with limited fields is sent to the Social Media Transformation Service, in order to be converted into an appropriate format for insertion to the Context Broker (more details will follow in D4.3 (M24)), while the complete JSON is stored in the MongoDB database.

There are two separate MongoDB collections, one for the use case of water safety and security and one for the floods and droughts. Thus, a reverse check is performed after the crawling of a tweet, in order to identify for which of the two use cases the tweet has been crawled for, by checking which one of the specified keywords is contained in the collected text.



```
 1 {
 2   "id": 1265851780434243600,
 3   "id_str": "1265851780434243587",
 4   "created_at": "Thu May 28 03:45:57 +0000 2020",
 5   "timestamp_ms": "1590637557501",
 6   "lang": "en",
 7   "is_retweeted_status": false,
 8   "full_text": "Auckland drought 'a wake up call' for water supply\nhttps
       ://t.co/EucvhAMg0Z https://t.co/j3HUF926fA",
 9   "image_url": "http://pbs.twimg.com/media/EZE1e7gUYAEOI-g.jpg",
10   "reliability": 0.4036,
11   "estimated_locations": [
12     {
13       "location_in_text": "Auckland",
14       "location_fullname": "Auckland, 1010, New Zealand",
15       "geometry": {
16         "type": "Point",
17         "coordinates": [
18           174.7631803,
19           -36.852095
20         ]
21       }
22     }
23   ],
24   "nudity": false
25 }
```

Figure 19. Fundamental and additional attributes of a tweet's JSON

Apart from collecting and analyzing tweets in near real-time manner, we are also interested in retrieving subsets of the collected tweets fast; for example, tweets that have posted in a specific time period (range queries) or tweets that contain images. For this reason, we have investigated indexing techniques in MongoDB in order to support faster retrieval. While for some fields it is quite straight-forward (e.g. by adding an index for the Boolean field *is_retweeted_status*, retweets can be retrieved more quickly), more investigation was needed for the date-related field of *timestamp_ms*.

In particular, three parameters have been investigated: (i) the type of the date (i.e. string, long number, or ISODate), (ii) the inclusion or not of indexing and projection (to be explained below), and (iii) the quantity of items to be retrieved (many or few). The results can be seen in Figure 20, while the table with the complete results is included in Appendix III.

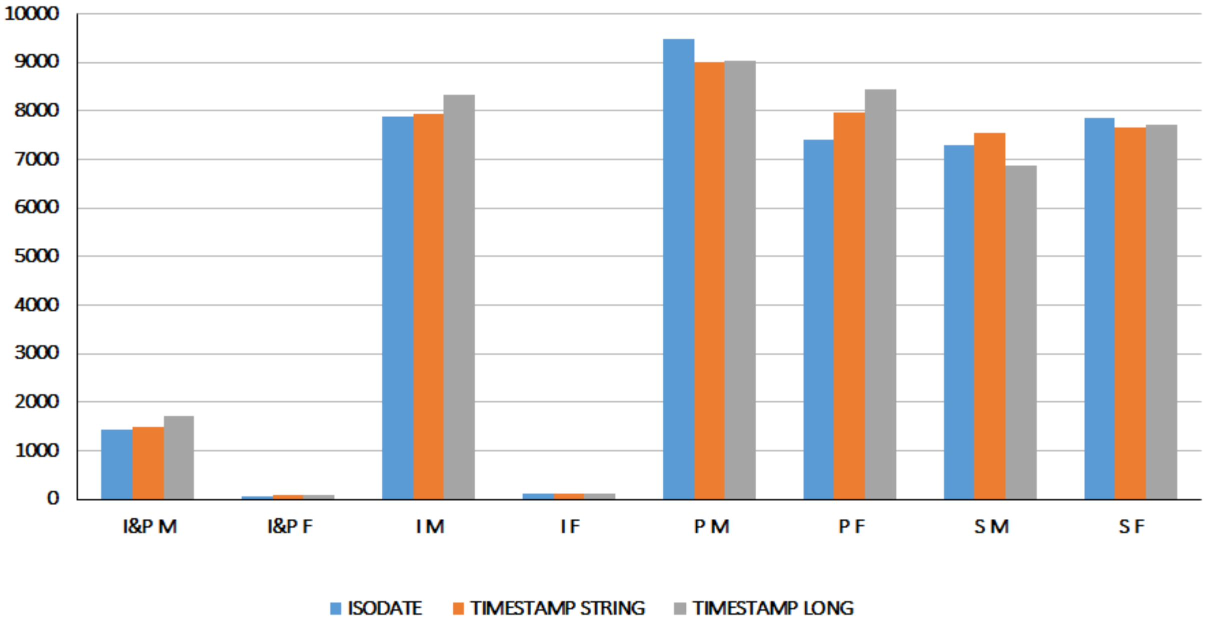


Figure 20. Response time (in milliseconds) of MongoDB queries in relation to date type, indexing (Indexing, Projection, Simple queries), and quantity of retrieved items (Many, Few)

Regarding the first parameter, dates in MongoDB can have a few different representations, so we have examined how each type can change the response time of range queries. Twitter API provides the string attribute *timestamp_ms* (e.g. “1602487793000”), so we have included in our examination its conversion to long number (1602487793000) and to ISODate ("2020-10-12T03:29:53.000Z"), which is automatically produced by MongoDB when you try to store a Java Date object. As seen in Figure 20, the differences in response time between date types are not significant, but apparently the ISODate achieves the lowest time.

The second parameter was the involvement of indexing and/or projection. If an appropriate index (I) exists for a query, MongoDB can use the index to limit the number of documents it must inspect. Querying only the index can be much faster than querying documents outside of the index, since index keys are typically smaller than the documents they catalog, and indexes are typically available in RAM or located sequentially on disk. Furthermore, projection (P) is to specify or restrict fields to return in a query, in order to limit the amount of data that MongoDB sends, because by default MongoDB queries return all fields in matching documents. When the query criteria and the projection of a query include only the indexed fields (I&P), MongoDB returns results directly from the index without scanning any documents or bringing documents into memory, and these covered queries can be very efficient. Finally, in simple queries (S), without indexes or projection, MongoDB has to perform a collection scan, i.e. scan every document in a collection, to find the documents that match the query statement. The results show that using solely projection does not help, whereas indexes are valuable only when few documents match the criteria. However, the combination of indexes and projection achieves a notable decrease in response time.

The third parameter was the quantity of documents to be retrieved, in the form of two distinct categories “many” (M) (a range query that matched 2,499,930 documents) and “few” (F) (a range query that matched 120,795 documents). In simple queries and in queries using projection, the number of retrieved documents does not seem to affect. In case of using indexes, few items can be returned speedily, but many items take as much time as without indexing, which highlights the necessity of projection. Indeed, covered queries respond significantly faster in both cases, but still the quantity appears to affect time.

All in all, our experiment has shown that the most suitable approach for having fast range query responses is the combination of indexing and projection, while the type of date does not play an important role.

5.2.4 Current status of the collection

Before continuing to the knowledge extraction and visualisation of the collected social media data, this subchapter is dedicated on how many tweets have been collected by the time of writing and some initial statistics on them. The collection of tweets has started on 19/12/2019 for the use case of water safety and security and some months later, on 28/04/2020, for the use case of floods and droughts. The current status of both collections can be seen in Table 12.

As expected, tweets in English, which is an international language, are much more than tweets in other languages (2 millions), but still more than 100,000 tweets have been collected in Italian and French. Unfortunately, both use cases are not discussed topics in Greece and Bulgaria, resulting to a very low number of crawled posts.

A large part of the collected tweets are retweets (44-73% in most cases), indicating that reposting is quite often on Twitter, but especially for English it has been decided to disregard retweets, because it led to a massive number of collected posts, making it harder to handle the incoming information. 13-37% of tweets contain an image, which is a low percentage but still significant, proving that visual information can also derive from social media. It also has to be noted that less than 1% of tweets come with geo-information provided by Twitter and this highlights the need for automatically detecting locations, which can geotag up to 26% of all collected tweets in English and Italian.

The status of the collection will be updated in deliverable D3.6 (M27).

Time period	Use Case	Language	Collected	Retweets	With image	With location by Twitter	With detected location
19/12/2019 – 06/10/2020	Water safety and security	English	2,278,662	N/A	309,693 (13.6%)	6,436 (0.3%)	113,479 (5%)
		Italian	133,351	75,445 (56.6%)	49,532 (37.1%)	530 (0.4%)	4,433 (3.3%)
		Greek	993	721 (72.6%)	137 (13.8%)	1 (0.1%)	N/A
		French	124,605	91,261 (73.2%)	21,555 (17.3%)	58 (0.05%)	N/A
		Bulgarian	115	17 (14.8%)	16 (13.9%)	1 (0.9%)	N/A
28/04/2020 – 06/10/2020	Floods and droughts	Italian	85,936	38,193 (44.4%)	26,635 (31%)	307 (0.4%)	23,058 (26.8%)
		Bulgarian	182	27 (14.8%)	45 (24.7%)	1 (0.5%)	N/A

Table 12. The current status of the social media collection

5.3 Knowledge extraction from social media data

As mentioned in 5.2.3, after a new tweet has been crawled and before it is stored to the database or forwarded to the Context Broker, a set of analysis techniques are performed to extract further knowledge from the social media data and the original JSON provided by Twitter for each collected tweet is enhanced with the analysis outcomes as additional pairs of attribute-value, appended to the JSON. The different techniques are described in this chapter, however the text classification is presented in much more detail, because a lot of effort has been put into this task and the contribution is novel.

5.3.1 Reliability estimation

The growing issue of fake news shared online naturally affects the content published on Twitter and consequently the crawled information that comes into the aqua3S system. For this reason, a quality check is highly essential and a mature solution (Boididou et al., 2017; Boididou et al., 2018), implemented in the past by CERTH, is utilised to estimate how reliable a tweet is.

The adopted module is an automatic verification technique that is able to classify a given tweet as real or fake, along with a confidence value. It relies on two independent classification models that are built on the same training data but using different sets of features: the tweet-based features and the user-based features. Since it is interesting to know which characteristics indicate a fake tweet, the features are given here in detail:

1. Tweet-based
    1.1. Text-based: characteristics of the text (e.g. length), stylistic attributes (e.g. number of exclamation marks, uppercase characters), existence of emoticons
    1.2. Language-specific: positive and negative words from sentiment lexicons in English
    1.3. Twitter-specific: number of retweets, hashtags, mentions, and URLs
    1.4. Link-based: reliability of the URLs shared through the tweet
2. User-based:
    2.1. User-specific: whether the user is verified by Twitter, number of followers, existence of a profile image, posting ratio, etc.
    2.2. Link-based: reliability of the URL shared in the profile description, if existing

After feature extraction is completed, model bagging is applied in order to obtain more reliable results, based on the predictions of the two classification models (one from each feature set). The classification algorithm is Logistic Regression, while an agreement-based retraining strategy is performed at prediction time to combine in a semi-supervised learning manner the two bags of models.

The methodology has been implemented as a standalone API that receives as input a tweet, in the original JSON format provided by Twitter, and replies with a Boolean label, i.e. true for "real" and false for "fake", and a percentage of confidence for the prediction. In order to have a range of reliability rather than a strict binary decision, we convert the prediction into a score as follows:

- "real" label with X confidence gives a reliability score of $X\%$
- "false" label with X confidence gives a reliability score of $(100 - X)\%$

5.3.2 Geotagging

The limited geographical information that Twitter provides (see Chapter 5.2.4) motivates the integration of an automatic geotagging methodology that is able to turn tweets into geo-referenced data based on their textual content. In particular, a module, which detects any locations mentioned in a tweet's text and links them to coordinates, is adopted and used for the supported languages, i.e. English and Italian.

Each tweet text is preprocessed properly and is fed to a Bidirectional Long Short-Term Memory model (Lample, et al. 2016) that assigns Named Entity Recognition labels to every qualified word of the text. Single words (e.g. "Genova") or sets of words (e.g. "via Mannara Pagani") that are recognised as locations are given as query to the OpenStreetMap API[10], which connects them with open geo-data and responds with the exact WGS84 coordinates.

The geotagging methodology has been implemented as a standalone API that gets a string input, i.e. the tweet text, and replies with the list of the detected locations, where each location is expressed with a pair of coordinates and the complete name as it appears on OpenStreetMap. For example, in the English tweet "*Residents get brooms out to tackle Welwyn Garden City flooding after Storm Alex*", the API detects the location:

- Welwyn Garden City, Hertfordshire, East of England, England, AL8 6TP, United Kingdom [51.8031083, -0.2068872]

while in the Italian tweet "*Allerta meteo: domani scuole chiuse a Pozzuoli, Bacoli e Monte di Procida #allertameteo #pozzuoli #bacoli #montediprocida*", the API detects the following locations:

- Pozzuoli, Napoli, Campania, Italy [40.822643, 14.1219109]
- Bacoli, Napoli, Campania, Italy [40.7966129, 14.0777834]
- Monte di Procida, Napoli, Campania, Italy [40.800907, 14.051827]

5.3.3 Nudity detection

Pornography and other forms of consensually produced adult content are allowed on Twitter, so this can lead to crawling inappropriate (visual) information. In order to prevent the aqua3S end users from viewing adult material and to avoid frustrating demonstrations, a nudity detection methodology has been applied to estimate whether a collected Twitter image contains nudity or not.

The used methodology is a procedure with two steps. In the first step, a DCNN produces the feature vector representation of the given image. Specifically, a 22-layer GoogleNet network (Szegedy, 2015) trained on 5,055 ImageNet concepts (Pittaras, 2017) is used, so the output of the trained network is a fully connected layer and has a dimension equal to 5,055. In the second step, a Linear Regression model takes as input the DCNN-based feature vector and proceeds with the binary classification of the image to one of the two classes "nudity" or "non-nudity", together with a probability of the prediction.

Similar to the previous knowledge extraction methods, the nudity detection module has been implemented as a standalone API that takes the URL of an image and returns the classification prediction as a Boolean, where true means there is nudity depicted.

5.3.4 Text classification

Text classification is becoming an increasingly important as it allows to easily get insights from data, and automate business processes. Specifically, text classification assigns a document/ text to one or more predefined categories according to their content. As far as aqua3S is concerned, the categories that are used are directly drawn from the aqua3S use cases which involve floods/ droughts and water safety and

[10] https://wiki.openstreetmap.org/wiki/API

security. Thus, the aim is to recognize whether the text retrieved from Twitter belongs to either of the two categories or it does not have relevant content although it might share common vocabulary. The latter is rather common in text classification task given that the meaning of words vary according to the context. For example, although the word "flood" is one of the keywords (Table 11 and Table 10 contain all the keywords in different languages used for retrieving tweets potentially relevant to aqua3S) used for retrieving tweets related to the floods/ droughts use case, the word), it can have a different meaning to the desired one (e.g. "my timeline is flooded with photos"), which consequently results in obtaining content from irrelevant tweets. Therefore, by adding a task classification step after tweet retrieval, we aim at removing as many as irrelevant tweets to the aqua3S use cases.

In this section, we begin with an overview of state-of-the-art methods for text classification, then we present the proposed framework and an evaluation of different methods, and finally we draw some conclusions.

It should be noted that the methods presented in this deliverable are also part of the EOPEN EU [11] project deliverable D3.3 (Andreadis, et al. 2020). However, the data used and eventually the models developed, are different since they are based on the data collected within aqua3S for its use cases.

5.3.4.1 Related work

Text classification task involves the following steps (see Figure 21):

1. Data gathering that includes collecting data stored in a variety of formats such as doc, html, or simple text.
2. Text Preprocessing, which involves cleaning converting the original text data in a data-mining-ready structure, and where the most significant text-features used for differentiating between text-categories are identified. It involves several step including: a) cleaning the data (e.g. emoji removal, punctuation removal, tag removal); c) tokenizing the text, that involves partitioning the text into a list of tokens; d) removing stop word, which involves the removal of frequently occurring words (e.g. "and", "the"); e) word stemming, which reduces words to their root form.
3. Text representation (Yan, 2009) that models documents and transforms them into numeric vectors. Several text representation methods have been proposed. For example, Vector Space Model (VSM) methods represent documents as vectors of words. A common VSM is the Bag of Words model (BOW) that uses all words appearing in a document as the index of the document vectors. Furthermore, it supports different term weighting schemas, including a) the Boolean model, where documents are represented by binary vectors; b) the Term Frequency model (TF) that uses the frequency of the terms; c) the Term Frequency Inversed Document Frequency (TFIDF) model, which also considers the term distribution among documents to weight terms in each document vector. A major drawback of the aforementioned representations is that they can't capture polysemy and synonymity and the surrounding context of the document. In order to tackle the issue of polysemy and synonymity, a method was proposed that included the N-gram statistical language models that attempt to capture the term correlation within document. However, this approach main problem that limits its application, is the exponentially increase of the data dimension. Another approach that considers the surrounding text and not just the terms is the word2vec approach (Mikolov et al. 2013). Word2vec involves building a model for producing word embeddings (i.e. representation of words from a given vocabulary as vectors in a low-dimensional space) based on deep neural networks (NN). Two types of models were proposed; the Continuous Bag-of-Words (CBOW) and the Skip-gram models. The major difference between the CBOW and Skip-gram models is that the first one tries to predict a word given the context of the word, while the second tries to predict the

[11] https://eopen-project.eu/

context of a word given the word. However, a significant prerequisite for both models is to be trained on large corpuses. Another approach similar to word2vec is GloVe (Pennington, et al. 2014) that obtains vector representations for words and is trained on aggregated global word-word co-occurrence statistics from a corpus. Contrary to the word2vec, GloVe doesn't use NN, and it has by default embedded in it explicit global information. One of the latest proposed approaches is the Bidirectional Encoder Representation from Transformers (BERT) algorithm (Devlin, et al. 2018), that includes an attention mechanism that learns contextual relations between words in a text. BERT aims at generating a language model, and the used mechanism reads the entire sequence of words at once, contrary to aforementioned approaches that read the text input sequentially. Therefore, it is considered bidirectional or non-directional.

4. Feature selection (Aggarwal, 2012; Chandrashekar, 2014), which reduces the dimensionality of the dataset by removing features that are considered irrelevant for the classification. The main two categories that are recognized are the filtering and the wrapper methods. Filtering techniques rank the features, then keep the highly ranked features and eventually apply on them the predictor, whereas in wrapper techniques the predictor is wrapped on a search algorithm which will find a subset that gives the best performance.
5. Classification Algorithms that are used for modeling classes and labeling text. There are several methods used to classify text such as Support Vector Machine, Naive Bayes Classifier, Logistic Regression and Decision Trees. The parameters of algorithms are tuned by considering the dataset in order to achieve the best performance possible.

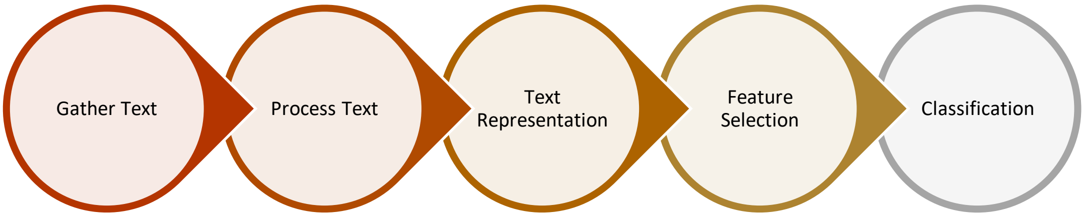


Figure 21. Text classification steps.

Apart from the aforementioned text classification methods, which are applicable to documents of various length, there exist also methods developed for handling text with special features including text from Twitter. The main characteristics of Twitter text is that it usually noisier, less topic-focused, shorter, and also it may contain non-standard terms, misspellings, "emojis", slang and abbreviations. The tweet's characteristics combined with the Twitter's increasing popularity, lead to the proposal of approaches that target specific its content. For example, Selvaperumal (2014) proposes considering the emoticons found inside a tweet and the use of a network algorithm that classifies tweets based on the use of tweet features like URL's, the retweeted tweets and the influential users tweet.

5.3.4.2 Text classification methodology

Within the context of aqua3S and as part of the first version of the data collection related deliverable, we have evaluated some traditional text classification techniques. These methods were evaluated on specific PUCs by considering both the number of collected tweets and the annotated ones. The approach that was followed involves the following steps:

1. Collection of short text messages from Twitter, as already described in Section 5.2.1.

2. Preprocessing of the tweets, which involves: a) removing URLs; b) removing emojis; c) mentions '@'; d) removing punctuation and all non-characters; e) splitting camel case words; and e) removing stop words. Moreover, word stemming is applied by using Porter language specific stemmers. However, we should note that stemming isn't applied in all cases as its outcome isn't satisfactory for certain languages.
3. Text representation by applying widely known methods including BoW using Term Frequency (TF), BoW using TF-IDF, and word2vec. Various experiments are realized for different feature length and n-gram values (i.e. n-gram = 1 or 2) for the BOW representation methods, and different vector dimensions and words window for the word2vec method.
4. Feature extraction step is omitted.
5. Creating classifier, which involves serving the text feature vector to the classifier (i.e. SVM, Naïve Bayes or Random Forests) and tuning the parameters of each one in order to reach the best possible performance. Thus, for SVM the parameter that is tuned is the penalty parameter (i.e. 0.01, 0.1, 1.0, 2.0, 3.0, 4.0, 5.0) and the type of the kernel (i.e. rbf or polynomial). For the Naïve Bayes, the additive smoothing parameter is tuned (i.e. 0.01, 0.1, 1.0). And finally, for the Random Forests the number of trees in the forest is tuned (i.e. 10, 50, 100, 200, 500, 1000). For the remaining parameters, default values are used. The classifiers are trained with the training set and validating against the validation set. It should be noted that cross validation is realized and thus the dataset is split into 5 datasets.

The techniques presented and validated in the current deliverable are traditional text classification techniques and can be considered as the baseline version of the text classification for aqua3S. However, in the next deliverable (D3.6), we will validate also more recent techniques such as BERT and also Twitter-oriented methods for text classification.

5.3.4.3 Training set creation with human annotation

The automatic text classification of tweets as relevant or not is a supervised ML technique that requires to be trained with annotated data. Due to the fact that all PUCs concern very specific topics and they are language-dependent, there is a total lack of already annotated datasets. To overcome this issue, we rely on creating a new training set with manual annotation. Manual annotation requires human effort, so as to mark a set of tweets with the labels "relevant" or "irrelevant", and this task has been assigned to the PUC leaders, who share a great knowledge on the use cases and speak the respective languages. In order to support their effort, an online Annotation Tool has been implemented (read more in 5.4) and the status of the human annotation at the time of writing can be seen in Table 13.

More than 6,000 English tweets about water safety and security have been annotated so far, but the limited number of relevant examples is not enough for a proper training set, so the effort should continue. On the other hand, the annotation of Italian tweets about both use cases is much richer (approximately 9,000 about water quality and 15,000 about floods and droughts) and more balanced, making it suitable for training a classification model. In Greek and Bulgarian the number of collected tweets is extremely inadequate; nevertheless, they have been all annotated. Finally, the task in French has not started yet.

It should be also noticed here that the high percentage of irrelevant tweets confirms the necessity of a text classification algorithm to further filter the tweets that satisfy the search criteria.

To have a better understanding of the logic behind the annotation, regarding the topic of water safety and security, the PUC leaders considered as relevant any water pollution indications and images, reports on drinking water, indications of aqueduct management, ban on the use of tap water, non-potable water, broken water pipes, water tariffs, and water purifiers. For the topic of floods and droughts, they considered as relevant tweets about recent floods, recent droughts, water supply issues, water crisis, weather forecasts, climate changes, water bombs, recent flood data and instruments that may be useful to predict rains. Moreover, tweets about historical floods and droughts were considered irrelevant.

Use Case	Language	Relevant	Irrelevant
Water safety and security	English	335 (5.3%)	5990 (94.7%)
	Italian	2121 (23%)	7086 (77%)
	Greek	10 (1.2%)	820 (98.8%)
	French	0	0
	Bulgarian	61 (54.5%)	51 (45.5%)
Floods and droughts	Italian	2228 (15.2%)	12418 (84.8%)
	Bulgarian	47 (37.6%)	78 (62.4%)

Table 13. Results of human annotation

5.3.4.4 Evaluation of the proposed methodology

The evaluation of the text classifiers is realized using the following metrics: precision, recall, and F-score that are commonly used in classification problems. Precision captures the proportion of positive identifications that were actually correct. Recall captures the proportion of actual positives that were identified correctly and, F-score (or F-measure) is a combination of precision and recall and is used to facilitate the comparison of models performance.

The aforementioned metrics are calculated in every run in order to decide the best performing classification method. By considering the numbers of the available (see Table 12) and the annotated tweets (see ***Table 13***) for each use case, in the current deliverable we develop solely the text classifiers for the Italian language for both use cases since in order to train efficient classifiers a significant number of records are required. Unfortunately for the other languages (i.e. Greek and Bulgarian), both the number of annotated tweets and the number of collected tweets are very few and can't be used for building a classifier. Finally, as far as the English data are concerned, although the number of collected tweets is significant, the number of tweets annotated as relevant is very low and it is impossible to build a good classifier based on them. In order to solve the issue of having too many irrelevant tweets collected, it is expected that the end users will validate again the provided list of keywords and improve it by removing terms that return irrelevant content.

Table 14 and Table 15 include the best performing models when different text processing options (i.e. without stop words, and without stop words and with stemming), and different text representations options (i.e. n-gram equals to 1, 2 or 3 for TF and TF-IDF representation methods, window size equals to 2 or 2 for word2vec representation method, and different feature length for all representation techniques) are considered for the two aforementioned use cases. After a careful observation, we can deduce that for the "Floods and droughts" PUC the performance of the different methods is comparable. However, the simple TF representation method with n-gram value equal to 2 and when stemming isn't applied performs slightly better and reaches a 86.4% F-score. Also, as far as the "Water safety and security" PUC is concerned, again the performance of the different methods is comparable. However, the simple TF representation method with n-gram value equal to 2 and when stemming is applied during text processing, performs slightly better and reaches a 66.8% F-score. Finally, an interesting observation one can make after studying carefully the best performing models in both PUCs, are that in both cases the best performing text representation method in both PUCs is the TF , which is also the simplest one, also that while the size of the annotated datasets is comparable between the two PUCs, their performances differ significantly. This is probably due to the fact that the Water safety and security" PUC is a far more complex use case, which keywords can vary significantly based on the context. We expect that when the end users improve the keywords used for collecting related social media tweets, an increase in the classifiers performance will be realized as well.

Parameter	Stop word/ stemming and corpus	Feature Length	Classifier	Precision	Recall	F-score
TF representation method						
n-gram = 1	Without stop words	601	Random Forest { num of trees: 1000}	0,8867	0,8303	0,8576
	Without stop words & with stemming	1969	Naïve Bayes { smoothing param: 0.1}	0,9175	0,8091	0,8599
n-gram = 2	Without stop words	1149	Random Forest { num of trees: 1000}	0,9013	0,83030	0,8644
	Without stop words & with stemming	7312	Naïve Bayes {smoothing param: 0.1}	0,9193	0,79394	0,8520
n-gram = 3	Without stop words	1546	Random Forest { num of trees: 1000}	0,9007	0,8242	0,8608
	Without stop words & with stemming	13589	Naïve Bayes { smoothing param: 0.1}	0,9158	0,7909	0,8488
TF-IDF representation method						
n-gram = 1	Without stop words	2548	Naïve Bayes { smoothing param: 0.1}	0,9003	0,8212	0,8589
	Without stop words & with stemming	1971	Naïve Bayes { smoothing param: 0.1}	0,9051	0,8091	0,8544
n-gram = 2	Without stop words	1149	Random Forest { num of trees: 1000}	0,8918	0,8242	0,8567
	Without stop words & with stemming	7312	Naïve Bayes { smoothing param: 0.1}	0,9147	0,8121	0,8604
n-gram = 3	Without stop words	1546	Random Forest { num of trees: 500}	0,8951	0,8273	0,8598
	Without stop words & with stemming	1482	Random Forest { num of trees: 200}	0,8849	0,8152	0,8486
word2vec representation method						
Words_window = 2	Without stop words, full tweets for Corpus	200	SVM {penalty param: 5.0}	0,9333	0,7636	0,8400
	Without stop words, only positive tweets for Corpus	400	SVM {penalty param: 5.0}	0,8139	0,5303	0,6422
Words_window = 3	Without stop words, full tweets for Corpus	100	SVM { penalty param: 4.0}	0,9379	0,7788	0,851
	Without stop words, only positive tweets for Corpus	100	SVM { penalty param: 0.01}	nan	0,0000	0,0000

Table 14. Evaluation of different representation and classification methods for the Italian Floods and droughts use case.

Parameter	Stop word/ stemming	Feature Length	Classifier	Precision	Recall	F-score
TF representation method						
n-gram = 1	Without stop words	1301	Naïve Bayes { smoothing param: 0.1}	0,7345	0,5991	0,65990

	Without stop words & with stemming	1291	Naïve Bayes { smoothing param: 0.1}	0,7639	0,5668	0,65079
n-gram = 2	Without stop words	1990	Naïve Bayes { smoothing param: 0.1}	0,7381	0,5714	0,64416
	Without stop words & with stemming	10516	Naïve Bayes { smoothing param: 0.01}	0,7875	0,5807	0,66844
n-gram = 3	Without stop words	18950	Naïve Bayes { smoothing param: 0.01}	0,7399	0,5899	0,65641
	Without stop words & with stemming	18398	Naïve Bayes { smoothing param: 0.01}	0,7785	0,5668	0,65600
TF-IDF representation method						
n-gram = 1	Without stop words	1301	Naïve Bayes { smoothing param: 0.1}	0,7174	0,6083	0,6584
	Without stop words & with stemming	1280	Naïve Bayes { smoothing param: 0.1}	0,7807	0,5576	0,6505
n-gram = 2	Without stop words	1990	Naïve Bayes { smoothing param: 0.1}	0,7222	0,5991	0,6549
	Without stop words & with stemming	636	Naïve Bayes { smoothing param: 0.1}	0,7727	0,5484	0,6415
n-gram = 3	Without stop words	2463	Naïve Bayes { smoothing param: 0.1}	0,7414	0,5945	0,6599
	Without stop words & with stemming	18260	Naïve Bayes { smoothing param: 0.01}	0,7455	0,5668	0,6439
word2vec representation method						
Words_window = 2	Without stop words, full tweets for Corpus	100	SVM { penalty param: 5.0}	0,6742	0,4101	0,510
	Without stop words, only positive tweets for Corpus	100	SVM { penalty param: 0.01}	nan	0,0000	0,000
Words_window = 3	Without stop words, full tweets for Corpus	100	SVM { penalty param: 5.0}	0,7317	0,5530	0,6299
	Without stop words, only positive tweets for Corpus	100	SVM { penalty param: 0.01}	nan	0,0000	0,0000

Table 15. Evaluation of different representation and classification methods for the Italian Water Safety and Security use case.

5.4 Visualisation of collected social media data

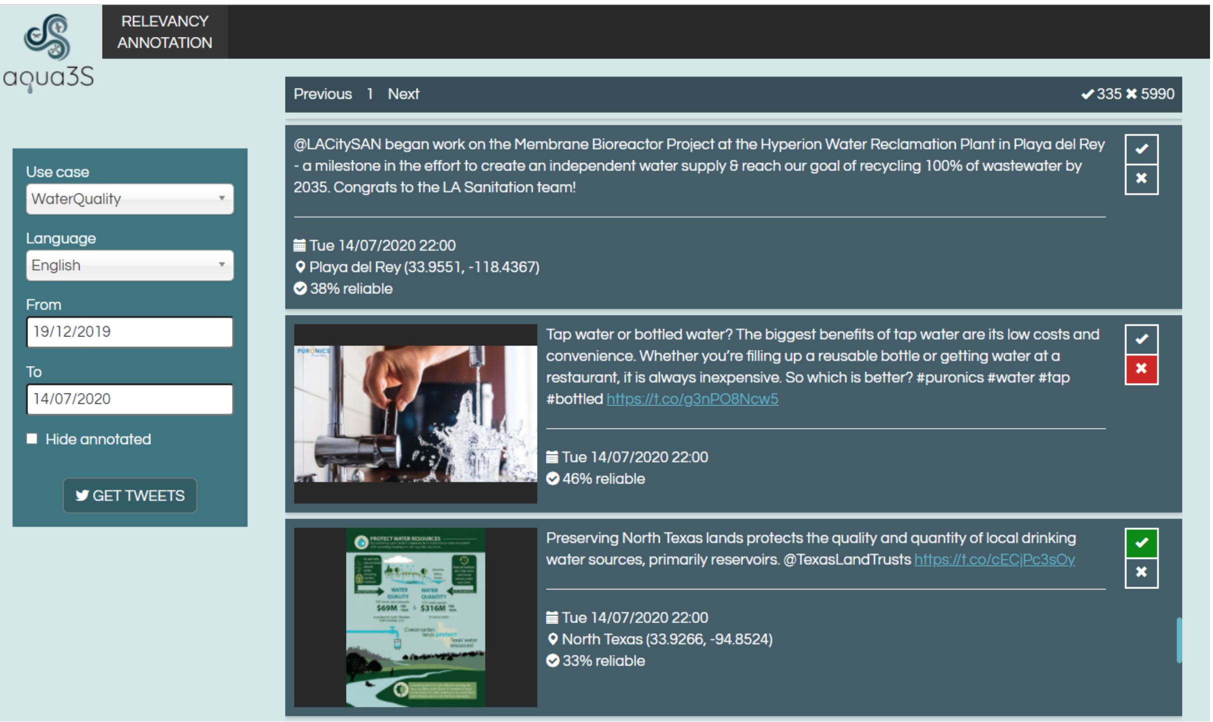
The social media data that are collected and analyzed inside the aqua3S project can be currently visualised in two Web interfaces: the Annotation Tool and the Visual Analytics Module, each of which serves a different scope.

As far as it concerns the Annotation Tool, its main purpose is to assist the end users in the task of creating a training set for the text classification (5.3.4.3) by providing an easy and straightforward way to annotate tweets. Additionally, the interface can be used as a demonstration tool, in order to show all the tweets that are being collected for the examined use cases as well as the outcomes of the various analyses, including the irrelevant/fake tweets, since it is not part of the core aqua3S system and information doesn't have to be filtered.

A screenshot of the Annotation Tool can be seen in Figure 22. The user interface consists of two components: a menu on the left and a results panel on the right.

The menu allows the user to select the use case of interest (i.e. water safety and security or floods and droughts), the language of interest (i.e. English, Italian, Greek, French and Bulgarian for the first; Italian and Bulgarian for the latter), and the time period of interest, by defining a starting date and an ending date. After all the options are set, the user can click on the "GET TWEETS" button and receive the respective results in the right panel, by querying the MongoDB where the tweets are stored.

The tweets appear in a scrollable list, paginated in sets of 50, while pages can be visited by clicking on "Previous" and "Next". Each tweet is visualised in a separate box and contains the following information: (i) the text of the tweet, with links being clickable, (ii) the image of the tweet, if existing, which can also be viewed in full screen when clicked and is blurred in case nudity is detected (5.3.3), (iii) the date and time that the tweet was posted on Twitter, converted in the time zone of the user's browser, (iv) the locations that have been detected inside the text by the geotagging technique (5.3.2), along with their coordinates, and (v) the reliability score (5.3.1). Moreover, for each tweet there are two buttons for annotation: one (tick symbol) to annotate as relevant and one (x symbol) to annotate as irrelevant. To support the end user with information about the progress of the annotation task, the number of already annotated tweets shows on the upper right part of the tweets list and is continuously updated.



Use case: WaterQuality

Language: English

From: 19/12/2019

To: 14/07/2020

□ Hide annotated

@LACitySAN began work on the Membrane Bioreactor Project at the Hyperion Water Reclamation Plant in Playa del Rey - a milestone in the effort to create an independent water supply & reach our goal of recycling 100% of wastewater by 2035. Congrats to the LA Sanitation team!

Tue 14/07/2020 22:00
Playa del Rey (33.9551, -118.4367)
38% reliable

Tap water or bottled water? The biggest benefits of tap water are its low costs and convenience. Whether you're filling up a reusable bottle or getting water at a restaurant, it is always inexpensive. So which is better? #puronics #water #tap #bottled https://t.co/g3nPO8Ncw5

Tue 14/07/2020 22:00
46% reliable

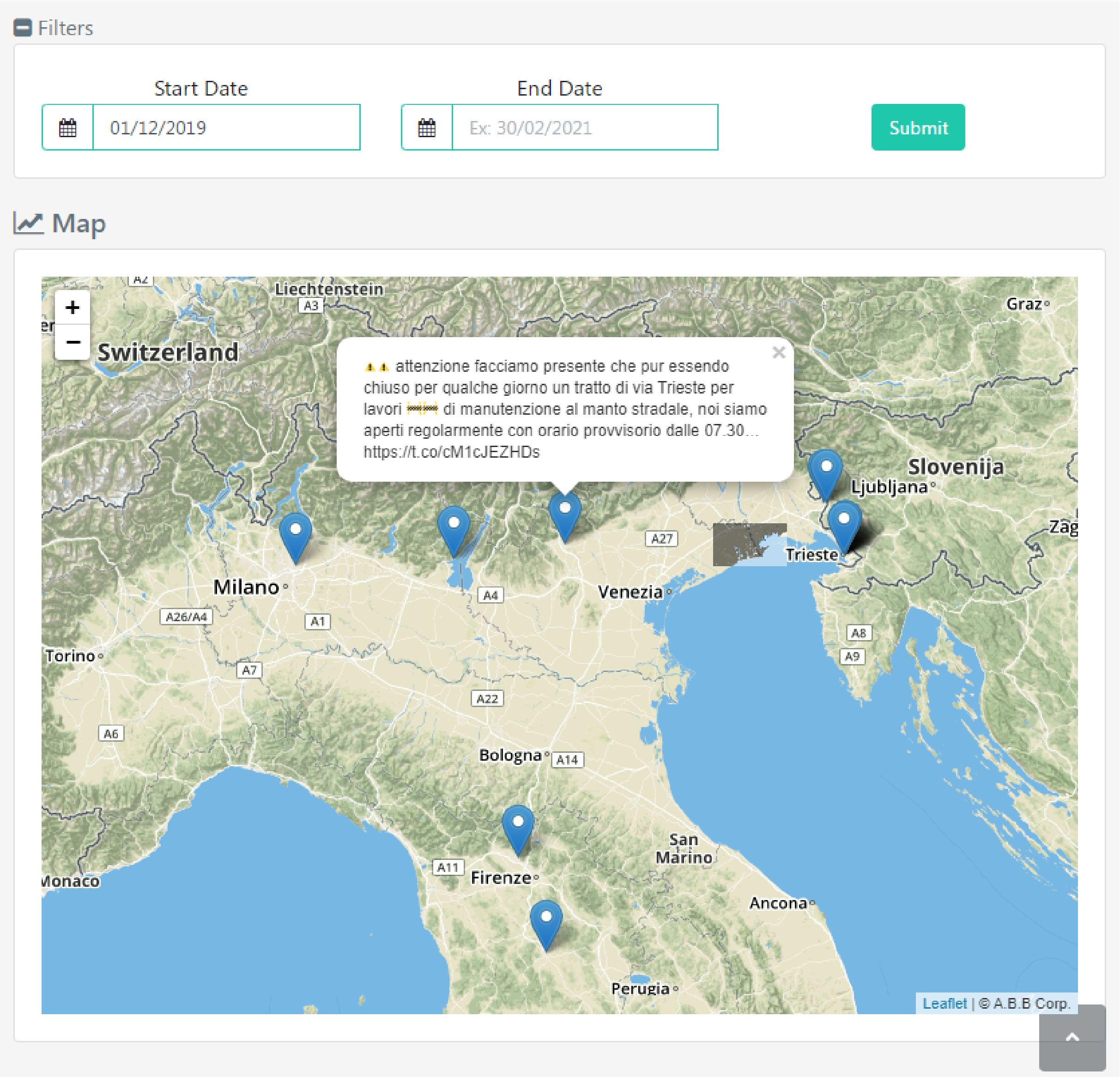
Preserving North Texas lands protects the quality and quantity of local drinking water sources, primarily reservoirs. @TexasLandTrusts https://t.co/cECjPc3sOy

Tue 14/07/2020 22:00
North Texas (33.9266, -94.8524)
33% reliable

Figure 22. Screenshot of the Annotation Tool

The second implementation that visualises the social media data is the Visual Analytics Module. One of the aims of this Web interface is to display information obtained from alternative sources, such as UAVs, satellites, and crowdsourcing, on an interactive map, so the module includes texts and images from tweets that are geotagged. A screenshot can be seen in Figure 23, where the user selects the date or the period of interest, clicks on the "Submit" button and then the map is updated with pins that represent tweets, positioned on the points of the locations detected in their text. Clicking on a pin shows a pop-up bubble that contains the message and the image (if existing) of the respective tweet. In contrast to the aforementioned visualization, this module provides a clearer picture of the geographical extent of potential issues, e.g. an incident of water pollution or a flooding event.

The Visual Analytics Module will be described in more detail in D5.1 (M18).



Start Date 01/12/2019

End Date Ex: 30/02/2021

⚠⚠ attenzione facciamo presente che pur essendo chiuso per qualche giorno un tratto di via Trieste per lavori 🚧🚧 di manutenzione al manto stradale, noi siamo aperti regolarmente con orario provvisorio dalle 07.30... https://t.co/cM1cJEZHDs

Figure 23. Screenshot of the Visual Analytics Module.

6. Conclusions

In this deliverable, we have presented the techniques for identifying oil spills and floods/ droughts from Earth Observation (EO) data and specifically from data retrieved from the Copernicus Hub. Moreover, we have initiated the retrieval and processing of data coming from social media, i.e. Twitter, in order to be used for capturing the societies response to water related events, such as poor water quality or floods/ droughts..

Starting with the monitoring of areas using satellite data, we defined the problem and presented the problems that it attempts to tackle, followed by an overview of the framework that depicts the workflow of the designed approach. For the downloading of necessary satellite data an automatic process has been implemented that uses the Copernicus Open Access Hub API, based on different criteria for each PUC, that is able to fetch new data and trigger the processing algorithms in a daily basis. Also, the way that the data is physically stored and then indexed in MongoDB making it available for later retrieval is described. Afterwards, the two main subtasks were presented that analyses the requested data. The first one involved the delineation of flooded areas, where a baseline method was extended to improve accuracy of the results. In the second subtask, various oil spill detection methodologies based on AI were evaluated that classify the underlying area as clean water or oil spill, with the selected detection methods being testing on either pixel or patch level. We concluded that moving away from the classic RGB channels of the image by using some specific bands that Sentinel-2 provides and by performing data augmentation to increase the size of the training dataset can significantly improve the prediction rate of oil spills. Eventually, we visualize the produced re information maps on a WEB UI.

As far as the Social Media Monitoring task is concerned, we presented the problem and provided an overview of the framework to introduce the reader to the implemented approach. The core component, i.e. the developed crawler, was presented in detail, including the usage of the Twitter Streaming API and the necessary processing after the tweets are collected. An examination of how tweets should be stored and queried showed that the combination of indexing and projection is necessary for efficient retrieval, while the type in which date is stored does not affect significantly range queries. Furthermore, the search criteria that were defined by the end users and dictate the acquisition of social media data and the status of the collections by the time of writing were both reported in tables in this document. A large number of English, French, and Italian tweets have been collected, but very few posts have been found in Greek and Bulgarian, showing that water-related issues are not popular on Twitter in these countries. Next, the reliability estimation, the geotagging, and the nudity detection algorithms were described, which add further knowledge to the collected posts. A text classification methodology was also presented and several text representation methods and machine learning techniques were investigated. The aim of this module was to remove irrelevant tweets before applying event detection on the collected tweets. The need for a training dataset led to a task of human annotation and end users were asked to manually annotate tweets as relevant and irrelevant; the results were reported here and Italian annotation was found adequate to be used for training. We concluded with the visualisation of tweets and two different user interfaces were demonstrated.

This deliverable is the first version of the visual content social media crawlers report, thus several updates will be realized in the second and final version of the deliverable that dues on M27. These updates involve developing a bias removal technique within the context of the flood scenario in order to decrease fluctuations caused by different satellite orbits of the Sentinel-1 products. Moreover, we plan on running more extensive experiments regarding the oil spill scenario that will text the proposed methodology on other dates and regions. Furthermore, we plan on developing and evaluating a technique for algae detection, and also apply State of the Art techniques of object detection for analysing the data received from drones and CCTVs. Finally, as far as social media data are concerned, event detection techniques will be investigated in order discover incidents that can be interesting or alerting regarding to the explored use cases, or they can produce respective notifications in the aqua3S system or used by other systems like the Crisis Classification module.

7. References

Andreadis, S., Moumtzidou, A., Gialampoukidis, I., Vrochidis, S., Karsisto, P. (2020). D3.3 - EOPEN Social Media Crawlers. Report of EOPEN ("opEn interOperable Platform for unified access and analysis of Earth observatioN data") EU Horizon 2020 project.

Boididou, C., Papadopoulos, S., Apostolidis, L., & Kompatsiaris, Y. (2017, June). Learning to detect misleading content on twitter. In Proceedings of the 2017 ACM on International Conference on Multimedia Retrieval (pp. 278-286).

Boididou, C., Papadopoulos, S., Zampoglou, M., Apostolidis, L., Papadopoulou, O., & Kompatsiaris, Y. (2018). Detection and visualization of misleading content on Twitter. International Journal of Multimedia Information Retrieval, 7(1), 71-86.

Filipponi, F. (2019). Sentinel-1 GRD Preprocessing Workflow. In Multidisciplinary Digital Publishing Institute Proceedings (Vol. 18, No. 1, p. 11).

Iglewicz, B., & Hoaglin, D. C. (1993). How to detect and handle outliers (Vol. 16). Asq Press.

Kim, Y., & Lee, M. J. (2020). Rapid Change Detection of Flood Affected Area after Collapse of the Laos Xe-Pian Xe-Namnoy Dam Using Sentinel-1 GRD Data. Remote Sensing, 12(12), 1978.

Kolokoussis, P., & Karathanassi, V. (2018). Oil spill detection and mapping using sentinel 2 imagery. Journal of Marine Science and Engineering, 6(1), 4.

Krestenitis, M., Orfanidis, G., Ioannidis, K., Avgerinakis, K., Vrochidis, S., & Kompatsiaris, I. (2019, January). Early Identification of Oil Spills in Satellite Images Using Deep CNNs. In International Conference on Multimedia Modeling (pp. 424-435). Springer, Cham.

Lample, G., Ballesteros, M., Subramanian, S., Kawakami, K., Dyer, C. (2016). Neural Architectures for Named Entity Recognition. Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (pp. 260-270), San Diego, California.

Liu, S., Chi, M., Zou, Y., Samat, A., Benediktsson, J. A., & Plaza, A. (2017). Oil spill detection via multitemporal optical remote sensing images: A change detection perspective. IEEE Geoscience and Remote Sensing Letters, 14(3), 324-328.

Nallapareddy, A., & Balakrishnan, B. (2020). Automatic Flood Detection in Multi-Temporal Sentinel-1 Synthetic Aperture Radar Imagery Using ANN Algorithms. International Journal of Computers, Communications & Control, 15(3).

Pandeeswari, B., Sutha, J., & Parvathy, M. (2020). A novel synthetic aperture radar image change detection system using radial basis function-based deep convolutional neural network. Journal of Ambient Intelligence and Humanized Computing, 1-14.

Pittaras, N., Markatopoulou, F., Mezaris, V., & Patras, I. (2017, January). Comparison of fine-tuning and extension strategies for deep convolutional neural networks. In International Conference on Multimedia Modeling (pp. 102-114). Springer, Cham.

Szegedy, C., Liu, W., Jia, Y., Sermanet, P., Reed, S., Anguelov, D., ... & Rabinovich, A. (2015). Going deeper with convolutions. In Proceedings of the IEEE conference on computer vision and pattern recognition (pp. 1-9).

Zeng, K., & Wang, Y. (2020). A Deep Convolutional Neural Network for Oil Spill Detection from Spaceborne SAR Images. Remote Sensing, 12(6), 1015.

8. Appendix I

JSON Template used for storing pollution map information:

```
{  "_id" : ObjectId("5f9aa89bf00e76400a78610a"),
   "uuid" : "some random id",
   "productName" : "the used coperncus product name",
   "url" : "the unique download URL",
   "product_type" : "the product type",
   "sensor_mode" : "the sensor mode",
   "datetime" : "the datetime of the item insertion in mongo",
   "timestamp" : "the timestamp of the item insertion in mongo",
   "datetime_sensing" : "the date and time in ISODate format that the area was captured by the
satellite"
   "platform" : "the satellite platform",
   "location_name" : "the puc unique name",
   "source" : "the copernicus hub url",
   "queryGeo" : { "type" : "Polygon",
      "coordinates" : [  "the bounding box of the area of the PUC in wkt format" ]},
   "geo" : {
      "type" : "Polygon",
      "coordinates" : [
         [ 5 points like   ["lon lat point 1 of initial product]
      ]
   },
   "generated_maps" : {
      "oil_spill_map" : "path of the output generated oil spill map, if calculated",
      "flood_map" : "path of the output generated flood map, if calculated"
   } }
```

Flood map:

```
{
   "_id" : ObjectId("5f9aed77f00e76400a78610b"),
   "uuid" : "4c18958d-a885-4b8e-a61a-054dcbbb31e3",
   "productName" :
"S1A_IW_GRDH_1SDV_20191115T051904_20191115T051929_029917_0369DD_B3EC"
   "url" : "https://scihub.copernicus.eu/dhus/odata/v1/Products('bb5a3764-4638-4423-9a24-
08eeff70b365')/$value",
   "product_type" : "S2MSI2A",
   "sensor_mode" : null,
   "datetime" : ISODate("2020-05-27T16:58:02.000Z"),
   "timestamp" : NumberLong(1590587882000),
    "datetime_sensing" : ISODate("2019-11-15T05:19:04.000Z")
   "platform" : "S2A",
   "location_name" : "PUC 1 - Trieste",
   "source" : "https://scihub.copernicus.eu",
   "queryGeo" : {
      "type" : "Polygon",
      "coordinates" : [
         "POLYGON((21.82819250155612 40.105607552915984,22.108395761417437
40.105607552915984,22.108395761417437 40.30529609284443,21.82819250155612
40.30529609284443,21.82819250155612 40.105607552915984))"
      ]
   },
   "geo" : {
      "type" : "Polygon",
      "coordinates" : [[ "initial copernicus product coordinates" ]]
   },
   "generated_maps" : {
      "oil_spill_map" : null,
      "flood_map" : "
S1A_IW_GRDH_1SDV_20191115T051904_20191115T051929_029917_0369DD_B3EC
_flood_map.png"
   }
}
```

9. Appendix II

General Tweet:

```
{
   "_id" : ObjectId("5ec47e7360b2b2f3b4167067"),
   "created_at" : "Wed May 20 00:48:20 +0000 2020",
   "id" : NumberLong(1262907980066041857),
   "id_str" : "1262907980066041857",
   "text" : "Take note, Peterborough! Rusty-coloured tap water for the next several weeks! https://t.co/UK6kympEmv",
   "display_text_range" : [
      0,
      77
   ],
   "source" : "<a href=\"https://mobile.twitter.com\" rel=\"nofollow\">Twitter Web App</a>",
   "truncated" : false,
   "in_reply_to_status_id" : null,
   "in_reply_to_status_id_str" : null,
   "in_reply_to_user_id" : null,
   "in_reply_to_user_id_str" : null,
   "in_reply_to_screen_name" : null,
   "user" : {
      "id" : 355726884,
      "id_str" : "355726884",
      "name" : "TransitionTownPtbo",
      "screen_name" : "TransitionPtbo1",
      "location" : "Peterborough ON",
      "url" : "http://www.new.transitiontownpeterborough.ca/ttp/",
      "description" : "Transition Town Peterborough",
      "translator_type" : "none",
      "protected" : false,
      "verified" : false,
      "followers_count" : 1159,
      "friends_count" : 673,
        "listed_count" : 36,
        "favourites_count" : 781,
        "statuses_count" : 1160,
        "created_at" : "Mon Aug 15 20:03:14 +0000 2011",
        "utc_offset" : null,
        "time_zone" : null,
        "geo_enabled" : true,
        "lang" : null,
        "contributors_enabled" : false,
        "is_translator" : false,
        "profile_background_color" : "642D8B",
        "profile_background_image_url" : "http://abs.twimg.com/images/themes/theme10/bg.gif",
        "profile_background_image_url_https"                                              :
"https://abs.twimg.com/images/themes/theme10/bg.gif",
        "profile_background_tile" : true,
        "profile_link_color" : "19CF86",
        "profile_sidebar_border_color" : "65AFDA",
        "profile_sidebar_fill_color" : "7AC3EE",
        "profile_text_color" : "242126",
        "profile_use_background_image" : true,
        "profile_image_url"                                                               :
"http://pbs.twimg.com/profile_images/1257046560015343616/StLhxvVm_normal.jpg",
        "profile_image_url_https"                                                         :
"https://pbs.twimg.com/profile_images/1257046560015343616/StLhxvVm_normal.jpg",
        "profile_banner_url" : "https://pbs.twimg.com/profile_banners/355726884/1588538230",
        "default_profile" : false,
        "default_profile_image" : false,
        "following" : null,
        "follow_request_sent" : null,
        "notifications" : null
    },
    "geo" : null,
    "coordinates" : null,
    "place" : null,
  "contributors" : null,
  "is_quote_status" : false,
  "quote_count" : 0,
  "reply_count" : 0,
  "retweet_count" : 0,
  "favorite_count" : 0,
  "entities" : {
    "hashtags" : [],
    "urls" : [],
    "user_mentions" : [],
    "symbols" : [],
    "media" : [
      {
        "id" : NumberLong(1262907799522234374),
        "id_str" : "1262907799522234374",
        "indices" : [
          78,
          101
        ],
        "media_url" : "http://pbs.twimg.com/media/EYa_89aX0AYQuRn.jpg",
        "media_url_https" : "https://pbs.twimg.com/media/EYa_89aX0AYQuRn.jpg",
        "url" : "https://t.co/UK6kympEmv",
        "display_url" : "pic.twitter.com/UK6kympEmv",
        "expanded_url" : "https://twitter.com/TransitionPtbo1/status/1262907980066041857/photo/1",
        "type" : "photo",
        "sizes" : {
          "thumb" : {
            "w" : 150,
            "h" : 150,
            "resize" : "crop"
          },
          "small" : {
                  "w" : 283,
                  "h" : 680,
                  "resize" : "fit"
               },
               "medium" : {
                  "w" : 400,
                  "h" : 960,
                  "resize" : "fit"
               },
               "large" : {
                  "w" : 400,
                  "h" : 960,
                  "resize" : "fit"
               }
            }
         }
      ]
   },
   "extended_entities" : {
      "media" : [
         {
            "id" : NumberLong(1262907799522234374),
            "id_str" : "1262907799522234374",
            "indices" : [
               78,
               101
            ],
            "media_url" : "http://pbs.twimg.com/media/EYa_89aX0AYQuRn.jpg",
            "media_url_https" : "https://pbs.twimg.com/media/EYa_89aX0AYQuRn.jpg",
            "url" : "https://t.co/UK6kympEmv",
            "display_url" : "pic.twitter.com/UK6kympEmv",
            "expanded_url"                                                        :
"https://twitter.com/TransitionPtbo1/status/1262907980066041857/photo/1",
        "type" : "photo",
        "sizes" : {
          "thumb" : {
            "w" : 150,
            "h" : 150,
            "resize" : "crop"
          },
          "small" : {
            "w" : 283,
            "h" : 680,
            "resize" : "fit"
          },
          "medium" : {
            "w" : 400,
            "h" : 960,
            "resize" : "fit"
          },
          "large" : {
            "w" : 400,
            "h" : 960,
            "resize" : "fit"
          }
        }
      }
    ]
  },
  "favorited" : false,
  "retweeted" : false,
  "possibly_sensitive" : false,
  "filter_level" : "low",
  "lang" : "en",
  "timestamp_ms" : "1589935700800"
}
```

Extended Tweet:

```
{
  "_id" : ObjectId("5ec2aaff60b2b2f3b416633e"),
  "created_at" : "Mon May 18 15:33:51 +0000 2020",
  "id" : NumberLong(1262406048725577730),
  "id_str" : "1262406048725577730",
  "text" : "@NYCMayor The mayor is LYING. Public beaches are paid for with public dollars; he cannot stop the public from using… https://t.co/aMBUz0xJwg",
  "display_text_range" : [
    10,
    140
  ],
  "source" : "<a href=\"https://mobile.twitter.com\" rel=\"nofollow\">Twitter Web App</a>",
  "truncated" : true,
  "in_reply_to_status_id" : NumberLong(1262116317412409344),
  "in_reply_to_status_id_str" : "1262116317412409344",
  "in_reply_to_user_id" : 19834403,
  "in_reply_to_user_id_str" : "19834403",
  "in_reply_to_screen_name" : "NYCMayor",
  "user" : {
    "id" : NumberLong(738060708714156032),
    "id_str" : "738060708714156032",
    "name" : "Dirty Pirate",
    "screen_name" : "DirtyPirate2016",
    "location" : "Austin, TX",
    "url" : null,
    "description" : "\"If we don't change the path we're on, we may just get to where we're going.\" ~ Chinese Proverb",
    "translator_type" : "none",
    "protected" : false,
    "verified" : false,
    "followers_count" : 21,
    "friends_count" : 34,
      "listed_count" : 0,
      "favourites_count" : 24,
      "statuses_count" : 610,
      "created_at" : "Wed Jun 01 17:32:40 +0000 2016",
      "utc_offset" : null,
      "time_zone" : null,
      "geo_enabled" : false,
      "lang" : null,
      "contributors_enabled" : false,
      "is_translator" : false,
      "profile_background_color" : "F5F8FA",
      "profile_background_image_url" : "",
      "profile_background_image_url_https" : "",
      "profile_background_tile" : false,
      "profile_link_color" : "1DA1F2",
      "profile_sidebar_border_color" : "C0DEED",
      "profile_sidebar_fill_color" : "DDEEF6",
      "profile_text_color" : "333333",
      "profile_use_background_image" : true,
      "profile_image_url"                                                        :
"http://pbs.twimg.com/profile_images/1257707697702608896/od4Lgg-L_normal.jpg",
      "profile_image_url_https"                                                  :
"https://pbs.twimg.com/profile_images/1257707697702608896/od4Lgg-L_normal.jpg",
      "profile_banner_url"                                                       :
"https://pbs.twimg.com/profile_banners/738060708714156032/1476205119",
      "default_profile" : true,
      "default_profile_image" : false,
      "following" : null,
      "follow_request_sent" : null,
      "notifications" : null
   },
   "geo" : null,
   "coordinates" : null,
   "place" : null,
  "contributors" : null,
  "is_quote_status" : false,
  "extended_tweet" : {
    "full_text" : "@NYCMayor The mayor is LYING. Public beaches are paid for with public dollars; he cannot stop the public from using them. The water &amp; sand have bacteria New Yorkers need. Get out &amp; get reacquainted with nature. Your bodies need the interaction. Educate yourselves. https://t.co/2tQpP7cejG",
    "display_text_range" : [
      10,
      296
    ],
    "entities" : {
      "hashtags" : [],
      "urls" : [
        {
          "url" : "https://t.co/2tQpP7cejG",
          "expanded_url" : "http://livinginconsciousness.blogspot.com/",
          "display_url" : "livinginconsciousness.blogspot.com",
          "indices" : [
            273,
            296
          ]
        }
      ],
      "user_mentions" : [
        {
          "screen_name" : "NYCMayor",
          "name" : "Mayor Bill de Blasio",
          "id" : 19834403,
          "id_str" : "19834403",
          "indices" : [
            0,
            9
          ]
```

```
            }
         ],
         "symbols" : []
      }
   },
   "quote_count" : 0,
   "reply_count" : 0,
   "retweet_count" : 0,
   "favorite_count" : 0,
   "entities" : {
      "hashtags" : [],
      "urls" : [
         {
            "url" : "https://t.co/aMBUz0xJwg",
            "expanded_url" : "https://twitter.com/i/web/status/1262406048725577730",
            "display_url" : "twitter.com/i/web/status/1…",
            "indices" : [
               117,
               140
            ]
         }
      ],
      "user_mentions" : [
         {
            "screen_name" : "NYCMayor",
            "name" : "Mayor Bill de Blasio",
            "id" : 19834403,
            "id_str" : "19834403",
            "indices" : [
               0,
               9
            ]
         }
```

```
        ],
        "symbols" : []
    },
    "favorited" : false,
    "retweeted" : false,
    "possibly_sensitive" : false,
    "filter_level" : "low",
    "lang" : "en",
    "timestamp_ms" : "1589816031043"
}
```

Retweet:

```
{
    "_id" : ObjectId("5ebc0c7b60b2b2f3b41622ee"),
    "created_at" : "Wed May 13 15:04:16 +0000 2020",
    "id" : NumberLong(1260586665543897090),
    "id_str" : "1260586665543897090",
    "text" : "RT @dibellagf: In USA continua la grande mattanza dopo: i senzatetto abituali,afroamericani,ispano americani, zingari e apolidi,adesso è i…",
    "source" : "<a href=\"http://twitter.com/download/android\" rel=\"nofollow\">Twitter for Android</a>",
    "truncated" : false,
    "in_reply_to_status_id" : null,
    "in_reply_to_status_id_str" : null,
    "in_reply_to_user_id" : null,
    "in_reply_to_user_id_str" : null,
    "in_reply_to_screen_name" : null,
    "user" : {
        "id" : NumberLong(701445550705606657),
        "id_str" : "701445550705606657",
        "name" : "urania",
        "screen_name" : "urania2906",
        "location" : null,
        "url" : null,
        "description" : "Storico ed economista. Esperto  problematiche sociali e tributarie.Scrive
per hobby romanzi e qualche verso. Tecnico Figc. Ama la letteratura e la filosofia.",

        "translator_type" : "none",

        "protected" : false,

        "verified" : false,

        "followers_count" : 1339,

        "friends_count" : 2742,

        "listed_count" : 2,

        "favourites_count" : 373,

        "statuses_count" : 1361,

        "created_at" : "Fri Jan 04 09:07:35 +0000 2013",

        "utc_offset" : null,

        "time_zone" : null,

        "geo_enabled" : false,

        "lang" : null,

        "contributors_enabled" : false,

        "is_translator" : false,

        "profile_background_color" : "C0DEED",

        "profile_background_image_url"                                                  :
"http://abs.twimg.com/images/themes/theme1/bg.png",

        "profile_background_image_url_https"                                            :
"https://abs.twimg.com/images/themes/theme1/bg.png",

        "profile_background_tile" : false,

        "profile_link_color" : "1DA1F2",

        "profile_sidebar_border_color" : "C0DEED",

        "profile_sidebar_fill_color" : "DDEEF6",

        "profile_text_color" : "333333",

        "profile_use_background_image" : true,

        "profile_image_url"                                                             :
"http://pbs.twimg.com/profile_images/3239785106/b13a193aa2a52c767c5d5573f426a9c7_nor
mal.jpeg",

        "profile_image_url_https"                                                       :
"https://pbs.twimg.com/profile_images/3239785106/b13a193aa2a52c767c5d5573f426a9c7_no
rmal.jpeg",

        "default_profile" : true,
      "default_profile_image" : false,
      "following" : null,
      "follow_request_sent" : null,
      "notifications" : null
    },
    "geo" : null,
    "coordinates" : null,
    "place" : null,
    "contributors" : null,
    "is_quote_status" : false,
    "extended_tweet" : {
      "full_text" : "In USA continua la grande mattanza dopo: i senzatetto abituali,afroamericani,ispano americani, zingari e  apolidi,adesso è in corso una vera e propria strage dei nativi americani. La Nazione Navajo(quella di Tex Willer)rischia di scomparire, senza acqua, mortalita 15 volte+alta https://t.co/Id925F12ZS",
      "display_text_range" : [
        0,
        278
      ],
      "entities" : {
        "hashtags" : [],
        "urls" : [],
        "user_mentions" : [],
        "symbols" : [],
        "media" : [
          {
            "id" : NumberLong(1260528448826748928),
            "id_str" : "1260528448826748928",
            "indices" : [
              279,
              302
            ],
            "media_url" : "http://pbs.twimg.com/media/EX5L8hWXsAAwgZ0.png",
            "media_url_https" : "https://pbs.twimg.com/media/EX5L8hWXsAAwgZ0.png",
              "url" : "https://t.co/Id925F12ZS",
              "display_url" : "pic.twitter.com/Id925F12ZS",
              "expanded_url"                                                       :
"https://twitter.com/dibellagf/status/1260528636660195329/photo/1",
              "type" : "photo",
              "sizes" : {
                "small" : {
                  "w" : 647,
                  "h" : 458,
                  "resize" : "fit"
                },
                "large" : {
                  "w" : 647,
                  "h" : 458,
                  "resize" : "fit"
                },
                "medium" : {
                  "w" : 647,
                  "h" : 458,
                  "resize" : "fit"
                },
                "thumb" : {
                  "w" : 150,
                  "h" : 150,
                  "resize" : "crop"
                }
              }
            }
          ]
        },
        "extended_entities" : {
          "media" : [
            {
```

```
            "id" : NumberLong(1260528448826748928),
            "id_str" : "1260528448826748928",
            "indices" : [
              279,
              302
            ],
            "media_url" : "http://pbs.twimg.com/media/EX5L8hWXsAAwgZ0.png",
            "media_url_https" : "https://pbs.twimg.com/media/EX5L8hWXsAAwgZ0.png",
            "url" : "https://t.co/Id925F12ZS",
            "display_url" : "pic.twitter.com/Id925F12ZS",
            "expanded_url"                                                              :
"https://twitter.com/dibellagf/status/1260528636660195329/photo/1",
            "type" : "photo",
            "sizes" : {
              "small" : {
                "w" : 647,
                "h" : 458,
                "resize" : "fit"
              },
              "large" : {
                "w" : 647,
                "h" : 458,
                "resize" : "fit"
              },
              "medium" : {
                "w" : 647,
                "h" : 458,
                "resize" : "fit"
              },
              "thumb" : {
                "w" : 150,
                "h" : 150,
                "resize" : "crop"
```

```
                     }
                  }
               }
            ]
         }
      },
      "quote_count" : 0,
      "reply_count" : 0,
      "retweet_count" : 4,
      "favorite_count" : 4,
      "entities" : {
         "hashtags" : [],
         "urls" : [
            {
               "url" : "https://t.co/OyjS6GtLyZ",
               "expanded_url" : "https://twitter.com/i/web/status/1260528636660195329",
               "display_url" : "twitter.com/i/web/status/1…",
               "indices" : [
                  117,
                  140
               ]
            }
         ],
         "user_mentions" : [],
         "symbols" : []
      },
      "favorited" : false,
      "retweeted" : false,
      "possibly_sensitive" : false,
      "filter_level" : "low",
      "lang" : "it"
   },
   "is_quote_status" : false,
  "quote_count" : 0,
  "reply_count" : 0,
  "retweet_count" : 0,
  "favorite_count" : 0,
  "entities" : {
    "hashtags" : [],
    "urls" : [],
    "user_mentions" : [
      {
        "screen_name" : "dibellagf",
        "name" : "Giovanni Di Bella",
        "id" : 1059968365,
        "id_str" : "1059968365",
        "indices" : [
          3,
          13
        ]
      }
    ],
    "symbols" : []
  },
  "favorited" : false,
  "retweeted" : false,
  "filter_level" : "low",
  "lang" : "it",
  "timestamp_ms" : "1589382256290"
}
```

10. Appendix III

	Many items to be retrieved				Few items to be retrieved			
	Simple	Projection	Index	Projection & Index	Simple	Projection	Index	Projection & Index
ISODate	8,125	8,475	6,170	1,555	9,146	7,899	106	71
	8,485	9,148	7,341	1,422	6,555	6,844	105	69
	7,179	9,364	9,355	1,397	6,954	7,550	108	71
	7,541	10,564	6,455	1,420	9,301	6,790	106	70
	6,610	8,749	7,226	1,403	6,717	8,250	105	71
	6,834	8,718	9,723	1,442	9,469	7,855	105	72
	6,730	10,454	8,532	1,457	7,866	7,116	104	72
	6,542	9,945	8,276	1,467	6,647	7,158	105	71
	7,315	9,051	7,563	1,422	8,620	7,031	105	69
	7,535	10,205	8,244	1,381	7,298	7,676	105	71
Average	**7,289.6**	**9,467.3**	**7,888.5**	**1,436.6**	**7,857.3**	**7,416.9**	**105.4**	**70.7**
Timestamp string	7,427	8,966	8,392	1,597	9,188	7,805	107	72
	6,183	9,356	8,209	1,430	5,613	7,343	109	71
	8,469	9,535	9,493	1,535	7,395	7,150	111	73
	8,162	8,509	8,022	1,484	9,838	6,954	107	73
	8,040	8,787	6,957	1,444	7,237	9,086	106	74
	7,744	8,764	6,486	1,472	7,027	10,907	107	76
	6,689	8,539	7,278	1,465	7,368	9,525	107	72
	7,807	9,114	8,674	1,537	6,745	7,525	110	73
	6,955	9,181	6,920	1,580	9,818	6,013	108	73
	7,907	9,213	8,954	1,458	6,407	7,453	108	71
Average	**7,538.3**	**8,996.4**	**7,938.5**	**1,500.2**	**7,663.6**	**7,976.1**	**108.0**	**72.8**
Timestamp long	5,894	8,427	8,560	1,823	8,187	10,236	119	81
	7,273	10,234	7,739	1,674	6,524	8,119	117	81
	6,914	8,305	8,762	1,645	9,718	8,182	117	82
	8,060	10,294	8,283	1,674	6,687	9,687	118	81
	7,763	8,428	8,285	1,663	7,596	7,124	119	81

	5,943	8,383	8,483	1,635	9,173	9,891	118	83
	7,847	9,111	8,337	1,648	6,980	6,954	116	81
	6,064	9,078	9,370	1,832	6,864	7,825	116	96
	6,199	8,467	7,732	1,784	5,649	9,548	118	82
	6,835	9,595	7,707	1,806	9,747	6,928	137	83
Average	**6,879.2**	**9,032.2**	**8,325.8**	**1,718.4**	**7,712.5**	**8,449.4**	**119.5**	**83.1**

Table 16. Response time (in milliseconds) of MongoDB queries in relation to date type, indexing, and quantity of retrieved items