EOPEN

opEn interOperable Platform for unified access and analysis of Earth observatioN data

H2020-776019

D4.4

Change, event and community detection techniques

Dissemination level:	Public
Contractual date of delivery:	Month 33, 31/07/2020
Actual date of delivery:	Month 33, 30/07/2020
Workpackage:	WP4 Knowledge discovery and content extraction
Task:	T4.1 Change Detection in EO data T4.2 Concept and Event Detection in non EO data T4.5 Community detection in Social Media
Type:	Demonstrator
Approval Status:	Approved

Version:	1.0
Number of pages:	66
Filename:	D4.4-Change, event and community detection techniques_2020-07-30_v1.0.docx

Abstract

This deliverable reports on the achievements regarding Work Package 4 "Knowledge discovery and content extraction" and specifically tasks T4.1 about change detection in EO (Earth Observation) data, T4.2 about concept and event detection in non-EO data, and T4.5 about community detection in social media. The key contributions of the deliverable are: (1) a water bodies delineation technique with Deep Convolutional Neural Network (DCNN); (2) two change detection techniques based on optical and Synthetic Aperture Radar (SAR) data, respectively; (3) an event detection methodology based on outliers in the collected social media data; (4) a community detection and key-players identification technique; (5) the integration of a new DCNN for concept extraction from non-EO images; and (6) a DCNN model to perform multi-label concept similarity on EO data.

The information in this document reflects only the author's views and the European Community is not liable for any use that may be made of the information contained therein. The information in this document is provided as is and no guarantee or warranty is given that the information is fit for any particular purpose. The user thereof uses the information at its sole risk and liability.

This project has received funding from the European Union's Horizon 2020 research and innovation program under grant agreement 776019

History

Version	Date	Reason	Revised by	Approved By
0.1	19/06/2020	Initial Draft - TOC	Stelios Andreadis (CERTH)	Ilias Gialampoukidis (CERTH)
0.2	17/07/2020	Contributions (Sections 3 & 4)	Stelios Andreadis (CERTH) Ioannis-Omiros Kouloglou (CERTH)	Anastasia Moumtzidou (CERTH)
0.3	19/07/2020	Contributions (Sections 2 & 5)	Anastasia Moumtzidou (CERTH) Marios Bakratsas (CERTH)	Stelios Andreadis (CERTH)
0.4	20/07/2020	Final contributions	Stelios Andreadis (CERTH)	Ilias Gialampoukidis (CERTH)
0.5	24/07/2020	Internal review	Dennis Hoppe (USTUTT)	Dennis Hoppe (USTUTT)
1.0	30/07/2020	Updated document after review Ready for submission	Stelios Andreadis (CERTH)	Ilias Gialampoukidis (CERTH) Stefanos Vrochidis (CERTH)

Author list

Organization	Name	Contact Information
CERTH	Stelios Andreadis	andreadisst@iti.gr
CERTH	Anastasia Moumtzidou	moumtzid@iti.gr
CERTH	Marios Bakratsas	mbakratsas@iti.gr
CERTH	Ioannis-Omiros Kouloglou	kouloglou@iti.gr
CERTH	Ilias Gialampoukidis	heliasgj@iti.gr
CERTH	Stefanos Vrochidis	stefanos@iti.gr

Executive Summary

In this deliverable we report the work that has been carried out for WP4 regarding knowledge discovery and content extraction. The document focuses on the task of change detection in EO data (T4.1), the task of concept and event detection in non-EO data (T4.2), and the task of community detection in social media (T4.5).

The key contributions and accomplishments of the above tasks that are described in this deliverable are:

1. A water bodies delineation technique with a DCNN that considers the polarisation bands of a Sentinel-1 product, paired with the corresponding elevation information.
2. Two alternative change detection techniques; a DCNN model based on optical data and an outlier detection method based on SAR data.
3. An event detection methodology based on outlier detection on the number of collected tweets.
4. A comparison of community detection algorithms to find the most efficient and fast approach.
5. A technique to identify the most influential users in communities, namely key-players.
6. The integration of a new DCNN called EfficientNet, used for extracting high-level concepts from Twitter images.
7. A DCNN model to perform multi-label concept similarity on EO images.

Abbreviations and Acronyms

API	Application Programming Interface
ATSED	Automatic Targeted-domain Spatiotemporal Event Detection
BC	Betweenness Centrality
dB	Decibel
DCNN	Deep Convolutional Neural Network
DEM	Digital Elevation Model
EO	Earth Observation
GRD-IW	Ground Range Detected - Interferometric Wide
JAR	Java Archive
JSON	JavaScript Object Notation
LMS	Learning Management System
LTT	Location-Time constrained Topic
MEB	Mapping Entropy Betweenness
ML	Machine Learning
MNDWI	Modified Normalized Difference Water Index
NASA	National Aeronautics and Space Administration
NGA	National Geospatial-Intelligence Agency
PCA	Principal Component Analysis
PUC	Pilot Use Case
SAR	Synthetic Aperture Radar
SRTM	Shuttle Radar Topography Mission
STSS	Space-Time Scan Statistics
SVM	Support Vector Machine
VGG	Visual Geometry Group
VH	Vertical Transmit-Horizontal Receive Polarisation
VV	Vertical Transmit-Vertical Receive Polarisation
WGS	World Geodetic System

Table of Contents

1 INTRODUCTION

This document relates to WP4 and presents the work that has been done for the tasks T4.1 “Change Detection in EO data”, T4.2 “Concept and Event Detection in non EO data” and T4.5 “Community detection in Social Media”, whose common objective is to discover knowledge and extract content for EO and non-EO (social media) data. Deliverable D4.1 (“Change detection techniques in Earth Observation”) included a thorough description of our work in water mask generation (T4.1), which is now extended and complemented with our proposed change detection techniques (T4.1). Also, D4.1 contained a brief presentation of event (T4.2) and community (T4.5) detection, which are presented here with much more detail.

In Section 2 we discuss the techniques on water detection. We investigate two different scopes of the subject. The first deals with the delineation of the water bodies of an area, taking into account the morphology of the ground and reducing the misclassified areas that appear near steep areas. The second scope focuses exclusively on the flooded regions. To achieve this, we test change detection techniques that compare the current image against other images of past dates. An outlier detection technique is applied on a time series of previous Sentinel-1 images that depict the casual state of the area to find the changed (i.e. flooded) areas. In this direction, we also present two more change detection methodologies that use Sentinel-2 data, one based on DCNNs (Deep Convolutional Neural Networks) and the other one on the remote sensing MNDWI (Modified Normalized Difference Water Index) index, that subtract images of consecutive days in order to detect if the changed areas are enough to characterize this transition as a flood incident.

Following, Section 3 focuses on the problem of event detection and proposes a methodology that considers outliers in the fluctuation of the daily number of collected tweets as events. The section starts with a study of related work (3.1) and continues with the suggested methodology (3.2) including the steps to discover further insights on the detected events. The implementation of the methodology as a process in the EOPEN platform and the connection to other processes are described in 3.3. The section concludes with two experiments that evaluate the proposed method: a quantitative evaluation (3.4.2) for PUC1 and a qualitative evaluation (3.4.1) for PUC2 and PUC3.

Section 4 concerns the discovery of communities in networks of social media accounts that are interlinked through their online behaviour and the identification of hubs (key-players) in these communities. Related work is presented in 4.1 and is followed by the description of the methodology in 4.2. The implementation of the approach as a service deployed in the EOPEN platform and the development of a dashboard in the EOPEN User Portal to visualise the results of the service are described in 4.3. A set of experiments follows in 4.4 that aim to evaluate different community detection techniques, compare the communities formed for each PUC and investigate the type of key-players identified during an event.

Next, in Section 5 we present the techniques on concept detection that extract the concepts from EO and non-EO images. As far as non-EO data are concerned, a new DCNN is used for extracting the high-level concepts from an image and its performance is compared against the DCNN presented in D4.1. Regarding concept detection in EO Data, it involves an analysis of the multi-label concept similarity task, followed by the suggested solutions based on deep learning techniques, and closes with a depiction of the proposed DCNN model. Three well-known pre-trained DCNNs and a trained one that resembles Visual Geometry Group (VGG) CNN architecture are presented to tackle the task, followed by an overview of the architecture of the trained network. A detailed presentation of this task can be found in D4.3 (“Multimodal fusion for information retrieval”).

Finally, Section 6 concludes the deliverable.

2 CHANGE DETECTION

This section describes the task of change detection that is focused on floods that comes in agreement with PUC1 about flood risk assessment and prevention. In general, change detection is a technique that can identify abrupt changes of any type. In our scope we investigate techniques that are focused on detecting changes caused by water presence. Initially, we present a water delineation technique based on artificial intelligence that exploits the elevation information of the area when using Synthetic Aperture Radar (SAR) data and then we present the developed change detection approaches that handle Sentinel-1 or Sentinel-2 products.

Regarding implementation, the Machine Learning (ML) water delineation method has been developed as a process and it has been integrated to the EOPEN platform, while the services responsible for the "change detection" are currently hosted on CERTH's premises.

2.1 Water body detection

In D4.1 we described flood monitoring approaches that could be applied on detecting changes of the water level that are attributed to floods or draughts. We presented two frameworks developed for identifying flooded areas using satellite images. Both frameworks were considered as baseline methods. The first framework was a thresholding algorithm that can be applied with slight modification both to Sentinel-1 and Sentinel-2 data. Specifically, for Sentinel-2 data thresholding of the MNDWI (Modified Normalized Difference Water Index) index (Xu, 2006) was used for discriminating water from non-water areas. As far as Sentinel-1 data are concerned, a pre-processing step of the initial product is required, followed by splitting the processed VV (Vertical Transmit-Vertical Receive Polarisation) or VH (Vertical Transmit-Horizontal Receive Polarisation) band into patches and calculating their average threshold that separates inundated from non-inundated areas. The second framework was exclusively oriented to handling optical images and thus it can be applied only to Sentinel-2 products and is characterized as a discriminant analysis method. This method is a combination of Mahalanobis distance-based classification for flood mask creation and morphological post-processing for flood mask correction which aimed at removing erroneous areas (Michail et al., 2018).

In the current deliverable, we present a new method for water body detection that involves the development of a DNN model that generates water-bodies masks for Sentinel-1 satellite data by fusing the SAR backscatter coefficients and the Digital Elevation Model (DEM) data. Hence, disregarding the steep sloped areas can eradicate these false positives, in the cost of removing at the same time some actually inundated areas. Our proposed method uses as input the different polarisation bands of Sentinel-1 images and combines them with the corresponding DEM information to estimate the output class, filtering out the highly sloped areas, leading to improved delineation accuracy. The model involves a training stage using ML algorithms including neural networks. The algorithm decides upon the effectiveness of slope removal for the specific pixel or block of pixels. In order to validate these findings and to quantify the impact of high slope removal, we run experiments on three major Italian lakes and their surrounding territories.

2.1.1 **Method**

The method proposed is based on ML techniques that involve the study of algorithms and statistical models that computer systems use in order to perform a specific task effectively without using explicit instructions, relying on patterns and inference instead. ML algorithms build mathematical models based on sample data, known as “training data”, in order to make predictions or decisions without being explicitly programmed to perform the task.

A set of pre-processing steps is required in order to transform a Sentinel-1 GRD-IW (Ground Range Detected - Interferometric Wide) product to a format that is suitable for further analysis. The following five major steps are executed on both the VH and VV bands:

- Subset: the initial product is cropped so that 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 55 filter size is used to filter the intensity data. As noted by (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 (World Geodetic System) 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.

In order to classify the various areas of lake images to the classes “inundated” and “non-inundated”, we built two DNN models from scratch, since there is no existing pretrained network similar to our task that could be used for transfer learning. The selected DNN architecture allowed us to evaluate the impact of the elevation information for water delineation in complex ground morphology conditions. The first model takes as input two features, the VV and VH values in decibel per pixel, whereas the second model considers as well the DEM values. The proposed model is a 3 layers network and it is fully connected since all neurons connect to all neurons in the next layer. The tuples of pixel values for each investigation area are inserted to the first (input) layer, bringing the initial data into the system for further processing. Then, two hidden layers follow; the first one consists of 12 neurons, whereas the second one of 8 neurons. Eventually, each of the initial tuples of values is classified at the output as “water” or “non-water”.

The input of the model is produced by generating tuples of various characteristics for each pixel. Thus, given that the input images are satellite images that consist of several polarisation bands, we evaluate triples of VV and VH band paired with the elevation values. Regarding the DEM file, the SRTM data at 30m Global 1 arc second V003 elaborated by NASA (National Aeronautics and Space Administration) and NGA (National Geospatial-Intelligence Agency) was used.

The following combinations were used to create a representation for each of the two models:

- VV-VH: The first NN model only uses the backscatter coefficients of the two polarisation bands.
- VV-VH-Dem: The second NN model uses the backscatter coefficients of the two polarisation bands and the corresponding DEM information.

Figure 1 depicts the framework of our Deep Neural Network that takes as input the processed VV and VH bands as well as the corresponding DEM values and outputs the predicted class for each pixel.

For the classification of areas as “inundated” or “non-inundated” a mix of values describing each pixel is required. Thus, triplets of the VV, VH and DEM values are fed as input to the model. Before this step, training the model is performed with a subset of triplets of values of the three lakes, paired with the annotated flood condition for the corresponding pixel. As an additional parameter, the slope values that derive directly from DEM could be added to the model’s architecture.

To evaluate the performance of the described DCNN approach, an examination towards the baseline automatic thresholding technique (D4.1) was performed on both VV and VH band polarisations, which distinguishes water areas from land in SAR imagery, e.g. (Townsend & Walsh, 1998). Pixels with values less than the threshold value are marked as water pixels, whereas pixels with values greater or equal to the threshold are marked as dry. Since the histogram thresholding technique is performed on different satellite images and areas, the outcome thresholds vary. For each lake the best result between VV and VH is kept.

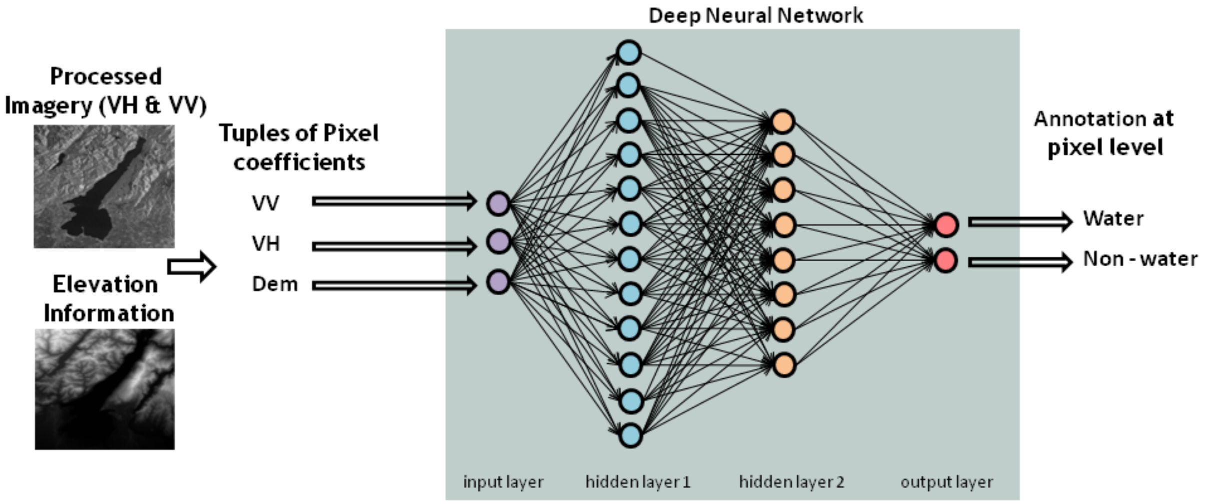


Figure 1: Our Deep Neural Network framework

2.1.2 **Experiments**

Dataset

The dataset is based on three satellite images consisting of millions of pixels, with each image depicting a different lake in Italy and the surrounding area. The dataset additionally includes the corresponding elevation information. The annotation file delineating lakes and water reservoirs is provided by Alto Adriatico Water Authority (AAWA). To align the annotation file with the under-investigation satellite images, Bing Maps high resolution images were used to make corrections to the annotated data. The processing steps were applied to the three images, and then the baseline method and the deep leaning approach were applied and compared. For the baseline method, both VV and polarisation bands of the three depictions of the lakes were used as input to estimate the threshold and calculate the water mask. Then, the best result for each lake is kept for reference. For the deep learning approach two different cases occur:

- Only Polarisation Bands: In the first case the input for our model are the values of VV and VH bands for each pixel.
- Polarisation bands paired with DEM data: In the second case the input for our model are the values of VV, VH bands and the DEM values for each pixel.

For the deep learning dataset, to speed up the training, a random fraction of the initial data was used. The amount of the selected inundated pixels equals the non-inundated pixels creating a balanced dataset, in order to avoid favouring any of the two classes, as most learners will exhibit bias towards the majority class as quoted by (Johnson & Khoshgoftaar, 2019).

Results

The results of our analysis and the comparison with the aforementioned baseline method are shown in Table 1. It should be noted that the metrics provided for the baseline refer to the polarisations that demonstrated the best performance. Regarding the automatically detected thresholds they all fell within the optimum ranges for the classification of flood water as specified by (Manjusree et al., 2012), where the backscattering coefficient of water, using Sentinel-1A data with VV polarisations, varies from -6 to -15 dB, and for VH polarisations, it varies from -15 to -24 dB.

Table 1: Results (F-score) for three major Italian lakes, comparing the three methodologies

Lake	Baseline	VV + VH	VV + VH + Dem
Maggiore	0,7533	0,9112	**0,9277**
Garda	0,8110	0,8738	**0,9410**
Trasimeno	0,7554	0,8757	**0,8891**

After careful observation of the table, we can see the direct impact of DEM to our neural network model. For the Maggiore lake, the F-Score increased from 91.12% to 92.77%, when moving from VV-VH to VV-VH-DEM, comparing to 75.33% of the baseline. For the Garda lake, the F-Score increased from 87.38% to 94.10%, comparing to 81.10% of the baseline. Finally, for the Trasimeno lake the F-Score increased from 87.57% to 88.91%, comparing to 75.54% of the baseline.

Apart from the F-score, several images are produced that depict the waterbodies using different methods. Specifically, Figure 2 depicts the generated water masks for the three lakes using histogram thresholding, where we can see some qualitative results of the baseline approach. Figure 3 depicts the generated water masks for the three lakes using our DNN trained with VV band, VH band and DEM file, where we can see some qualitative results of our approach.

To sum up, SAR imagery is characterized of high noise. The thresholding approaches are usually applied on a single polarisation band and are able to provide satisfactory results but only for smoother water surfaces. In areas with a more complex morphology, filtering the steep sloped areas can improve the delineation of the detected water masks. However, there are cases that it worsens the results, because it may remove actually flooded areas lying on steeper river banks.

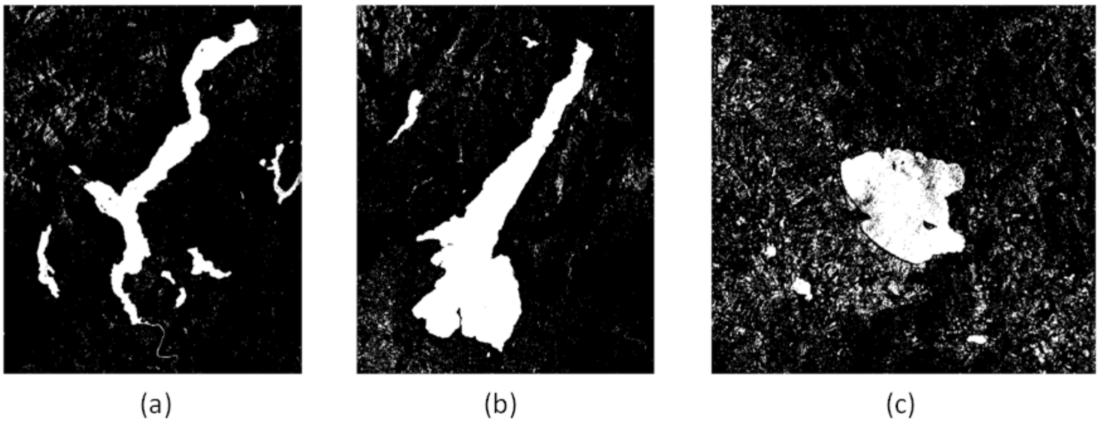


Figure 2: Water masks of lakes (a) Maggiore, (b) Garda, (c) Trasimeno using histogram thresholding

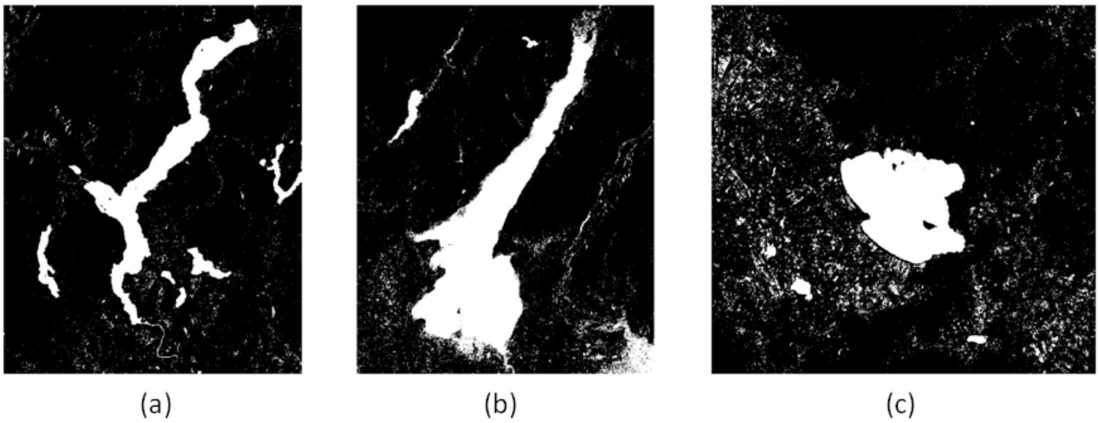


Figure 3: Water masks of lakes (a) Maggiore, (b) Garda, (c) Trasimeno using our DNN with VV, VH and DEM

2.1.3 **Implementation and integration to the EOPEN platform**

The ML based water detection module has been implemented and imported on the EOPEN platform (Figure 4). The necessary processes are chained, forming a workflow that is scheduled to be executed daily. The workflow begins with the initiation of the discovery process, searching for available Sentinel-1 GRD-IW products of the predefined area of interest of the last day. Then the downloading of the product(s) follows. Then, for each of the downloaded products, the generation of the S-1 watermask process is triggered. Multiple containers will spawn in case of multiple products and will be processed in parallel. The output of the process is the waterbodies mask in the file formats of shapefile (.shp) and raster (.tiff). Eventually the output file is being published to the vSphere for visualisation reasons and a notification email is sent to inform the owner of the workflow whether the execution has completed successfully.

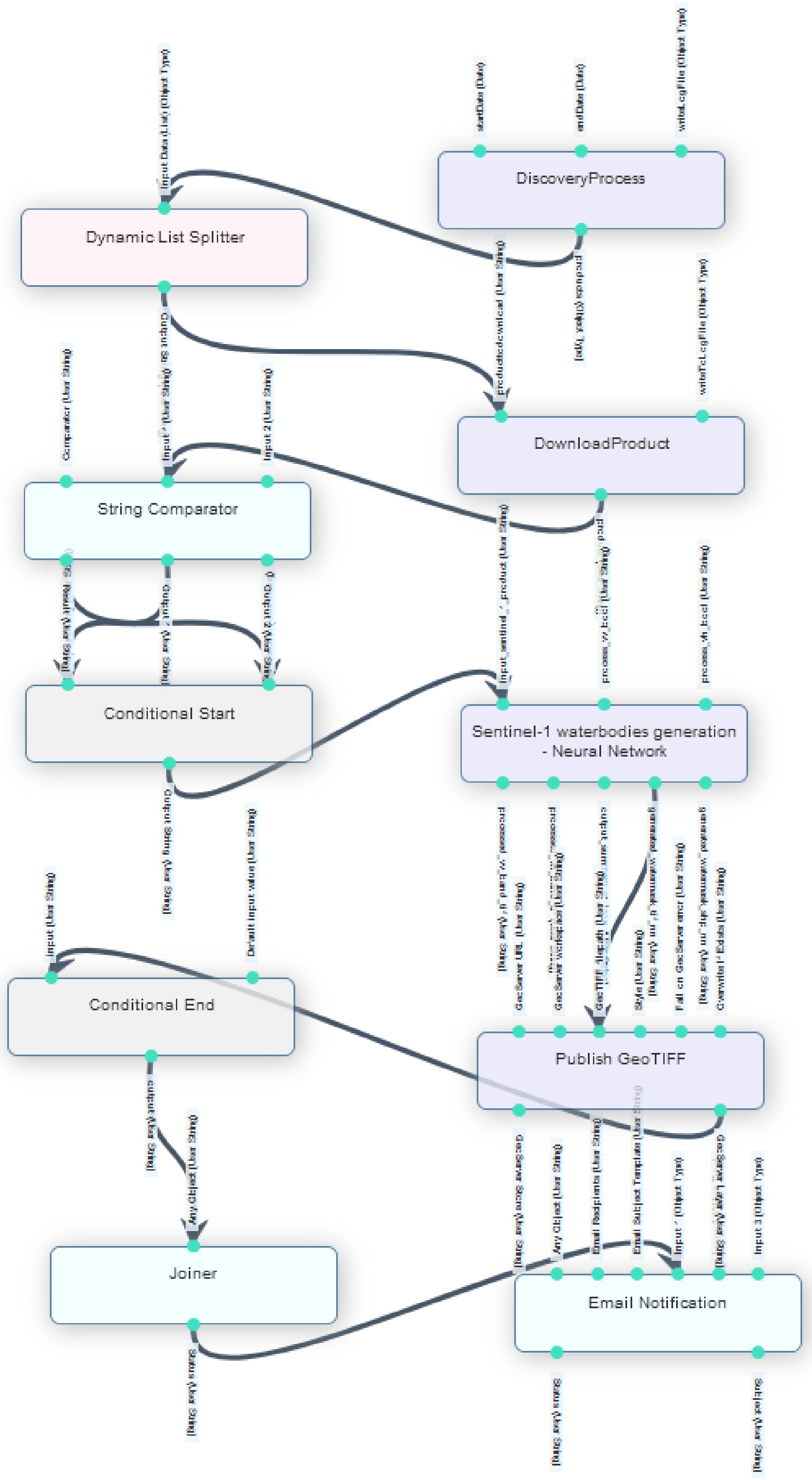


Figure 4: Workflow of Sentinel-1 Watermask generation using DCNN

2.2 Change detection techniques

Change detection captures the spatial changes from multi-temporal satellite images due to manmade or natural phenomena. In remote sensing, monitoring environmental changes and more specifically floods is of great importance. Remote sensing satellites acquire satellite images at varying resolutions that can be used to detect changes. This section presents some of the more recent advancements and obtained results of our proposed change detection methods that can handle both SAR and optical data.

2.2.1 State of the Art for Change Detection

For the task of change detection, several approaches have been developed. However, interest has also grown recently at the examination of a time series of satellite images, facilitating the creation of reference images that depict the normal state of an area, whereas it allows us to track the time that a change has happened. Thus, (Clement et al., 2018) used a sequence of multiple SAR images to generate a reference image based on the median value of each sequence of pixels. Then, bi-temporal analysis was applied between the under-investigation image and the reference image, to estimate the flooded areas. Contrary to these approaches, we have used changed detection techniques on several spectral indices that are sensitive to water bodies estimation and compare them with Machine/Deep Learning approaches.

Machine learning techniques have been used for detecting changes in satellite images. (Huang et al., 2008) and (Bovolo et al., 2008) treat change and no-change as a binary classification problem using Support Vector Machine (SVM), which is a well-known supervised non-parametric statistical learning technique. The (Bovolo et al., 2008) approach aims at extracting the change information by jointly analysing the spectral channels of multi-temporal images in the original feature space without any training data. Some other machine learning algorithms used for classification and change monitoring include random forest (Pal, 2005; Smith, 2010) and decision trees (Long et al., 2011). In the work of (Chaouch et al., 2012), the combination of SAR data and optical images, when coupled with a high-resolution digital elevation model (DEM), was shown to be useful for inundation mapping. DEM is used to delineate the high contour line where combined with the low contour line of the optical satellite image provides the flood-prone area. Using SAR data, change detection is performed in the aforementioned area to generate the flooded/non flooded area map.

2.2.2 Change detection in Sentinel-1 time series

Sentinel-1 images are able to provide surface information regardless the weather condition, day or night, rendering them ideal at monitoring flood incidents. As change detection is defined how the attributes of some area have changed between two different periods of time. To make the approach more robust and decrease statistical error for this measurement, we use a stack of 30 images prior to a flood incident in order to create the reference average normal-state image.

Methodology

To tackle changes in terms of flood/drought, we applied a change detection technique on a time series of Sentinel-1 products. 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 Lemene region in Italy on 17/11/2019 with the time series of the 30 satellite images falling between 17/08/2019 and 15/11/2019. We also calculated the flood map for a dry state on 15/11/2019, two days prior the flood event. Here, the satellite images between 13/08/2019 and 09/11/2019 were taken into account for the time series. In Figure 5 there is a visualisation of the two flood maps of Lemene region. For the outlier detection (i.e. the flood areas) we used the formula:

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

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 reduce significantly the number of false positives.

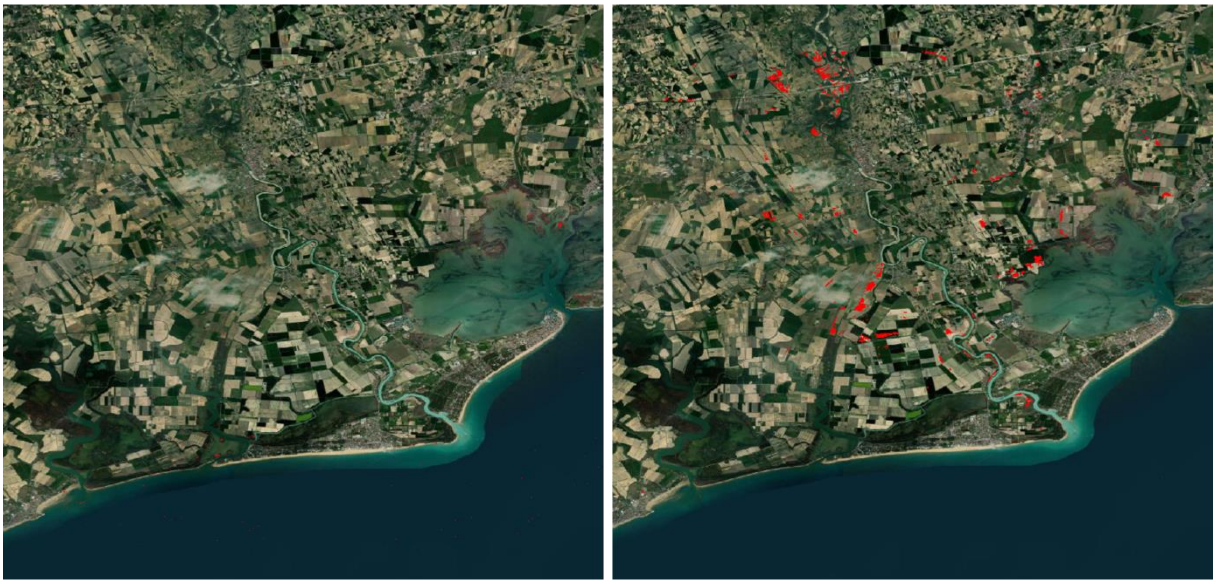


Figure 5: Flood map of Lemene river region on 15/11/2019, at a dry state (left) and on 17/11/2019, during a flood event (right). Flooded areas appear in red colour.

2.2.3 **Change detection in Sentinel-2 time series**

Various methods were investigated and presented in MME 2019[1] and ISCRAM 2020[2] that detect flood events. The first approach presented the discriminative ability of a classic remote sensing method that delineates changed (i.e. flooded) areas by subtracting two satellite images, characterizing this difference as a transition from a dry to a flooded state or

[1] http://www.multimediaeval.org/mediaeval2019/

[2] https://www.drrm.fralinlifesci.vt.edu/iscram2020/index.php

not. In a second approach, we trained a DCNN model and used it to discriminate a flooded from a non-flooded image, as a whole, without working on pixel level and eventually detect the presence of a flooded event when examining a time series of the optical data.

Methodology

MNDWI spectral index

A change detection approach based on the remote sensing spectral index of MNDWI was implemented. Within each event, the MNDWI differences of the consecutive days were calculated. Then, the outliers were estimated as follows: pixel's values that fall within $[m - \gamma\sigma, m + \gamma\sigma]$ (Lu et al., 2005) denote no change, i.e. no flood. A minimum water ratio needs to be set to characterize a difference image as flooded and is defined as the sum of the changed pixels in the difference image divided by the sum of all pixels in the image. Figure 6 depicts an example of the technique, where two MNDWI images of consecutive days are being subtracted to each other and then the outlier technique highlights the changed pixels.

DCNN approach

The second approach to detect flood events using satellite sequences involved the use of a deep learning model which was trained with two different datasets consisting of three-channel images with the differences of two days within an event. The first dataset was created by combining the Red-Green-Blue (B02-B03-B04) bands and the second by combining the Red-Swir-Nir (B02-B03-B04) that are sensitive to water. Then, the three bands were stacked and re-scaled to JPEG. Eventually, within each event, all the unique differences between its days were calculated. In Figure 7 the two input RGB images and their subtraction is presented. The difference is more obvious on the top part of the area highlighting the flooded area.

In the sequel, pre-trained networks on full ImageNet dataset were fine-tuned in order to learn the new features of our described dataset. The last pooling layer was replaced with a densely-connected NN layer with a softmax function with 2 outputs. Figure 8 depicts the outlier detection framework.

The following parameters were considered: (i) evaluation of the Adam and SGD optimizers, and (ii) evaluation of learning rates 0.1, 0.01, 0.001. The batch size was set to 32.

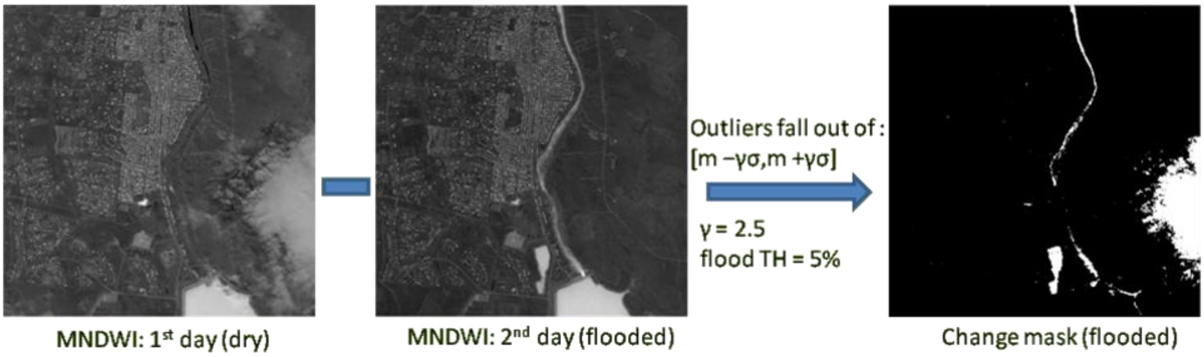


Figure 6: MNDWI differences of two consecutive days

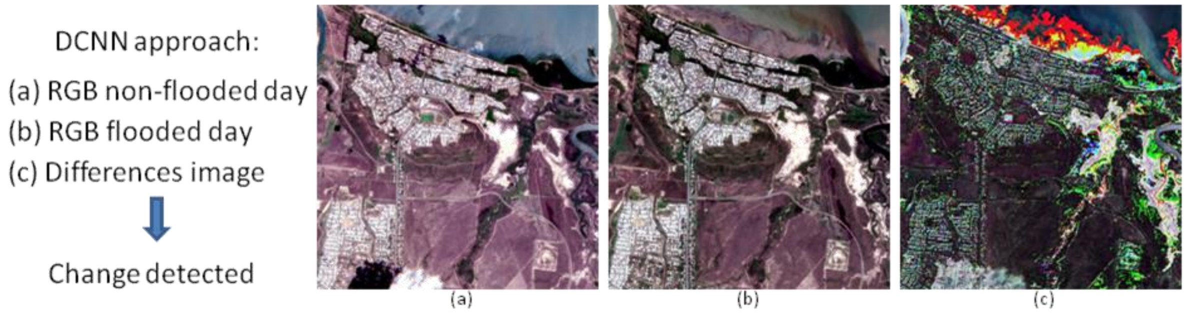


Figure 7: Two consecutive RGB images and their difference image (DCNN approach)

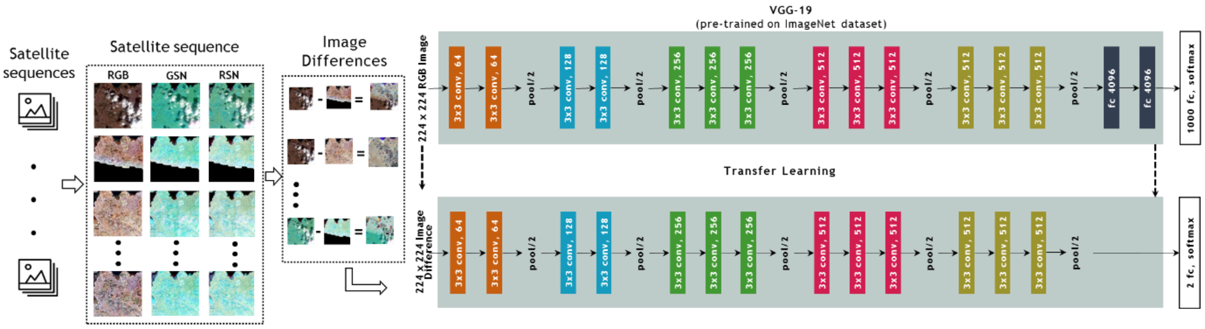


Figure 8: DCNN-based method framework for change detection

Experiments

Dataset

The dataset consists of a set of sequences of satellite images that depict a certain city over a certain length of time and are provided for the MediaEval 2019 Satellite Task[3], and specifically for the "City-centered satellite sequences" subtask (Bischke et al., 2018). In total, 335 events, each consisting of varied number of images, are provided, 267 of which are considered as the training set whereas the remaining 68 are the test set. Each event has a number of layers that depict different acquisition dates of the satellite images. Therefore, each sequence can be represented as a 3D array with size equal to image width x image height x number layers. Layers are of size 512x512 pixels for the 10m bands (B03, B04, B08). The 20m band (B11) band was sharpened to 10m thus re-sized to the same dimensions as the previous bands.

For the DCNN-based approach, data augmentation was applied in order to increase the training set. Specifically, for each image sequential rotations of 90 degrees were applied, combined with three different modifications of the contrast and brightness ratio, also flipping horizontally, resulting to a dataset increased 24 times. In order to further augment the dataset we considered as input not only the difference between two consecutive layers (dates) but between any two images within the same event. This led eventually to a dataset of 180,000 image differences as training set and 58,000 image differences as test set. However, given the limitations imposed by GPU, we have narrowed down the dataset to 30,000 records as train set and 9,000 records as test set.

[3] http://www.multimediaeval.org/mediaeval2019/multimediasatellite/

The selection of the images that would form the two datasets was random. Moreover, it should be noted that both the training and the test dataset were balanced, in respect to the number of difference images that denote change and no change. Furthermore, for the creation of the false-colour PNG images the following triplets of bands were considered: Red - Green - Blue, Red - SWIR1 - NIR and Green - SWIR1 - NIR. Finally, an extra parameter that was considered for creating the dataset was whether the satellite image has any missing part due to the partially covered tiles that correspond to those at the edge of the swath path at a satellite pass. To tackle this issue, we produced two different training and test datasets. In the first one, we considered as input images those that had 100% of the pixels appearing correctly, while in the second one, we considered as input images those that had at least 50% of the pixels appearing correctly. In both cases images that had less of 50% of the pixels appearing were omitted.

Results

The complete results in the training set and the test set as presented in the Multimedia Satellite Task of MediaEval can be seen in Table 2. In detail, detecting the outliers on the differences of MNDWI consecutive images achieved a 76.47% F-score. The image differencing technique proved adequate to detect changes relative to flood events, using the σ and minimum $water\_ratio$ values that were calculated on the annotated training set. Using DCNN provided decent results (70.58%), showing its ability to learn to detect flood patterns even with a small training set. The three more layers of VGG19 when compared to VGG16 do not result to a performance increase.

Table 2: Results (F-score) for all approaches

Approaches	Training set	Test set
	F-Score (%)	F-Score (%)
MNDWI (γ =2.1 ratio=0.05)	83.54	76.47
VGG16 Red-Green-Blue	60.74	70.58
VGG16 Red-Swir-Nir	60.74	70.58
VGG19 Red-Swir-Nir	60.74	70.58

The specification of the flooded areas is a crucial issue for the disaster management authorities. With this information, they are able to correlate the flooded areas with their relevant characteristics (resources, infrastructures, etc.) in order to get prepared for future events. Additionally, the authorities can estimate the impact of the flood for a specific area and the results for the measures that have been taken.

The results demonstrate the ability of the combined method of image differencing and water relative index of MNDWI to detect flood events, showing better robustness with balanced FP and FN rates, compared to the DCNN approach that follows at close range.

3 EVENT DETECTION

Event detection is a subtask of WP4's task T4.2 "Concept and Event Detection in non EO data" and aims to identify real-world events by discovering anomalies in streams of social media data.

A preliminary investigation has been reported in D4.1, which proved that peaks in the fluctuation of the number of tweets that have been collected per day for a specific use case can be associated with real incidents. Thus, our work has been focused on discovering these peaks.

After studying the related work in event detection (3.1), we have developed a methodology that is based on outlier detection and specifically the method of z-score (3.2). When an event is detected, additional information extraction is applied in order to extract further information about the event, i.e. the location where it has occurred and the most mentioned words so as to gain insights on what the incident really is about.

A process has been deployed in the EOPEN platform that implements the above methodology and the final outcome of the module is "translated" from a JSON file to a natural language alert in order to notify the end users of the platform (3.3).

The continuous crawling of Twitter posts during the project's lifetime has resulted in a very large collection of more than 11 million tweets, which now allows experiments towards a large-scale evaluation. We have performed a quantitative evaluation for PUC1 (3.4.2), where the success of identifying real floods through the number of tweets is calculated with respective metrics. Furthermore, we have performed a qualitative evaluation for PUC2 and PUC3 (3.4.1), where detected events during the year 2019 are presented, along with the extracted details, and are linked to real incidents.

3.1 Related work

Twitter is currently the most popular social media platform, with millions of active users per month and millions of tweets being posted every day. This vast amount of crowdsourcing information –and real time in fact– is able to express the actual world in the form of opinions or facts/news/events. Discovering real-world incidents by automatically identifying events in Twitter streams has attracted the interest of the scientific community and a multitude of techniques has been proposed towards achieving this goal.

(Atefeh & Khreich, 2015) provide a detailed survey of different techniques for event detection from Twitter streams. The techniques are categorised by (i) targeting a specified or unspecified type of event, (ii) applying a supervised or unsupervised detection method, (iii) referring to retrospective or new event detection, and (iv) the field of application. In addition, the features (either general or Twitter-specific) used, the datasets and the evaluation metrics are all summarised in this work.

More recently, (Hasan & Schwitter, 2018) prepared another survey on event detection methods that are applied to Twitter data streams. This survey classifies the various techniques into four groups: (i) term-interestingness-based approaches, (ii) topic-modelling-based approaches, (iii) incremental-based approaches, and (iv) miscellaneous approaches, e.g. hybrid techniques.

Some interesting and highly cited works are described in more detail next. (Abdelhaq et al., 2013) aim to detect localised events in real time from a Twitter stream and also to track the evolution of the events over time. Localised information is extracted by clustering keywords based on their spatial similarity, while events are identified with a scoring scheme.

(Hua et al., 2016) propose a semi-supervised method for automatic targeted-domain spatiotemporal event detection (ATSED) in Twitter. A customised classifier, based on the features of tweets, learns from historical data to detect ongoing events, while a multinomial spatial-scan model identifies the geographical locations of the events.

(Cheng & Wicks, 2014) suggest a space-time scan statistics (STSS) methodology to identify disaster events, which searches for clusters of tweets by taking into consideration the location and time, while ignoring their content.

The work in (Cordeiro, 2012) analyses hashtag occurrences in the Twitter stream and uses Continuous Wavelet Transformation to identify abrupt increases, i.e. peaks, on the mentions of hashtags. This method is combined with a Latent Dirichlet Allocation topic inference model based on Gibbs Sampling to describe the detected events.

Focusing on a natural disaster, (Sakaki et al., 2010) propose an algorithm to detect earthquake incidents through tweets. A classifier is built based on features such as the keywords in the Twitter text, the number of words, and their context, and a probabilistic spatiotemporal model is produced to discover the centre and the trajectory of the event location. Kalman filtering and particle filtering are applied for location estimation.

On the other hand, (Alsaedi et al., 2017) present an end-to-end integrated framework that combines the classification and online clustering of collected tweets, in order to detect smaller-scale events, which can disrupt social safety and security, such as riots.

Regarding improvements, (Ozdikis et al., 2012) propose an enhancement to an event detection technique by lexico-semantically expand the tweet contents. Based on syntagmatic and paradigmatic relationships between words, texts are enriched with similar words and then document similarity and agglomerative clustering algorithms are applied.

In addition, the work of (Zhou & Chen, 2014) is not limited to Twitter and presents a graphical model named location-time constrained topic (LTT) to capture the content, time, and location of social media messages and represent them as a probability distribution over a set of topics by inference. Events are detected by conducting efficient similarity joins over social media streams.

Finally, (McMinn et al., 2013) do not propose another event detection technique, but introduce a large-scale corpus of more than 120 million tweets, covering a period of 4 weeks, in order to be used for evaluating event detection approaches. Over 150,000 tweets of the corpus have been annotated as relevant or not and relate to more than 500 events.

Our scope is to propose a straight-forward approach, alternative to the aforementioned works, which employs outlier detection to achieve event detection.

3.2 Description of the methodology

Outliers are data points that are notably different from other data points. In other words, they're unusual values in a dataset. Even though outliers can be problematic for many statistical procedures, because they can be errors and distort real results, they may also indicate variability in a measurement or a novelty.

Some of the most popular methods for outlier detection, as presented in (Santoyo, 2017), are:

- Z-Score or Extreme Value Analysis (parametric)
- Probabilistic and Statistical Modelling (parametric)
- Linear Regression Models (PCA, LMS)
- Proximity Based Models (non-parametric)
- Information Theory Models
- High Dimensional Outlier Detection Methods (high dimensional sparse data)

We have selected z-score, since it is a very effective method when the values in the feature space can be described with a Gaussian distribution, which is the case in the number of collected tweets per day. This can be proved, for example, in the frequency histogram illustrated in Figure 9 for the number of collected Italian tweets about floods in years 2017-2018. It should be noted that maximum 100 tweets are collected normally per day, so values larger than 100, which occur only during incidents, are considered extreme values and are excluded from the histogram.

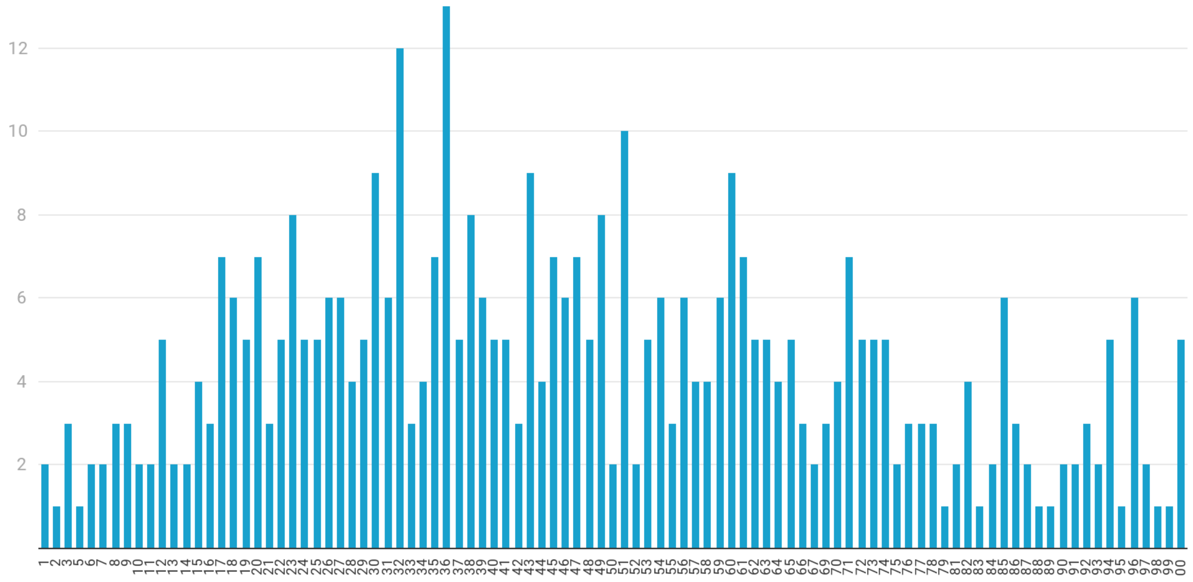


Figure 9: Frequency histogram of collected Italian tweets about floods in 2017-2018

(Shiffler, 1988) defines the z-score of an observation as a metric that indicates how many standard deviations a data point is from the sample's mean, assuming always a Gaussian distribution. This makes z-score a parametric method.

Mathematically, the formula to calculate the z-score is the following:

$$z = \frac{x - \mu}{\sigma}$$

where x is the value of the examined point, μ is the mean of a sample $x_1, x_2, \dots, x_n$, i.e.

$$\mu = \frac{1}{n}\left(\sum_{i=1}^{n} x_i\right)$$

and σ is the standard deviation of the sample, i.e.

$$\sigma = \sqrt{\frac{1}{n-1}\sum_{i=1}^{n}(x_i - \mu)^2}$$

To characterize a data point as an outlier, its z-score needs to be above a specified threshold:

$$if\ z > t\ then\ x\ is\ outlier$$

For the threshold bibliography suggests a value between 2.5 and 3.5 (see also Figure 10).

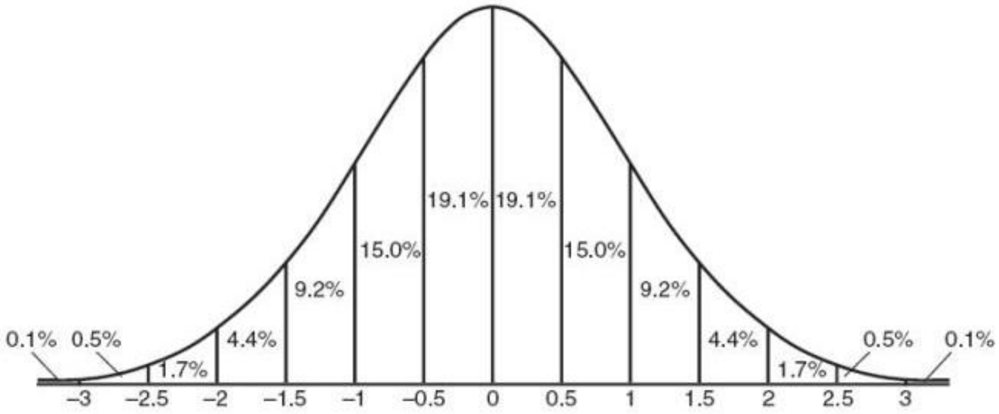


Figure 10: Normal curve of standard deviation

In order to apply the outlier detection methodology to our problem, i.e. the detection of events through social media data, we calculate the z-score of each date. The data points refer to the number of collected tweets per date and each point can be compared to the other points of the year, of the month or of the last thirty days. If the z-score exceeds a constant threshold, then we consider there is an event in that date.

In the event detection module, which has been implemented in the EOPEN platform and will be described in the next subsection, we have set a sample of the last thirty days and a threshold of 3. The first has been selected because we wish to check for events every day and we cannot wait for the whole month or year to be completed, while the latter has been selected after the evaluation described in Section 3.4.1 (best F-score compared to the other examined thresholds).

When an outlier, namely an event, is detected, we proceed to a two-step analysis so as to obtain additional information on the event. The first step is to estimate the location of the incident. Since some of the tweets are geotagged, as a result from the localisation technique reported in D5.1, we calculate which location is mentioned the most in the tweets of the date that has been marked as an event.

The second step of this analysis is to extract the ten most mentioned words, which will give an insight on what the incident refers to. A text pre-processing is necessary here, so that only meaningful words are included in the final top-ten list. Pre-processing involves:

1. Removing common Twitter terms, e.g. “RT” that means “retweet”
2. Removing URLs
3. Removing punctuation
4. Removing numbers (steps 1-4 use regular expressions)
5. Converting to lowercase
6. Removing stop words (different per language)

After the text of every tweet is cleaned, we calculate the frequency of each word appearing across all tweets and we select the ten most frequent words.

Finally, the results of the event detection are stored in JSON format. “timestamp” refers to the date and time the procedure runs, but also indicates the date when the event happened. “usecase” and “language” specify the collection whose tweets have been examined, e.g. about snow coverage in Finnish. “score” is the value of the z-score, while “change” refers to the rise of the number of tweets compared to number of the previous days, in percentage. The most mentioned location is given with the attributes “location” for the name and “point” for the exact coordinates. Lastly, the most mentioned words are shown as an array in the attribute “keywords”.

```
{
     "timestamp" : long,
     "usecase" : string,
     "language" : string,
     "score" : double,
     "change" : double,
     "location" : string,
     "point" :
     {
            "x" : double,
            "y" : double
     },
     "keywords" : [string]
}
```

3.3 Implementation and integration to the EOPEN platform

The implemented workflow regarding event detection can be seen in Figure 11. The methodology described above has been implemented as a JAVA application and has been integrated into the EOPEN platform with a process that runs the respective JAR file. The process is currently scheduled to run every day (but could run more frequently according to the desires of the users) and event detection is performed for all the tweet collections that are stored in the MongoDB (refer to D3.3 for information on the collections).

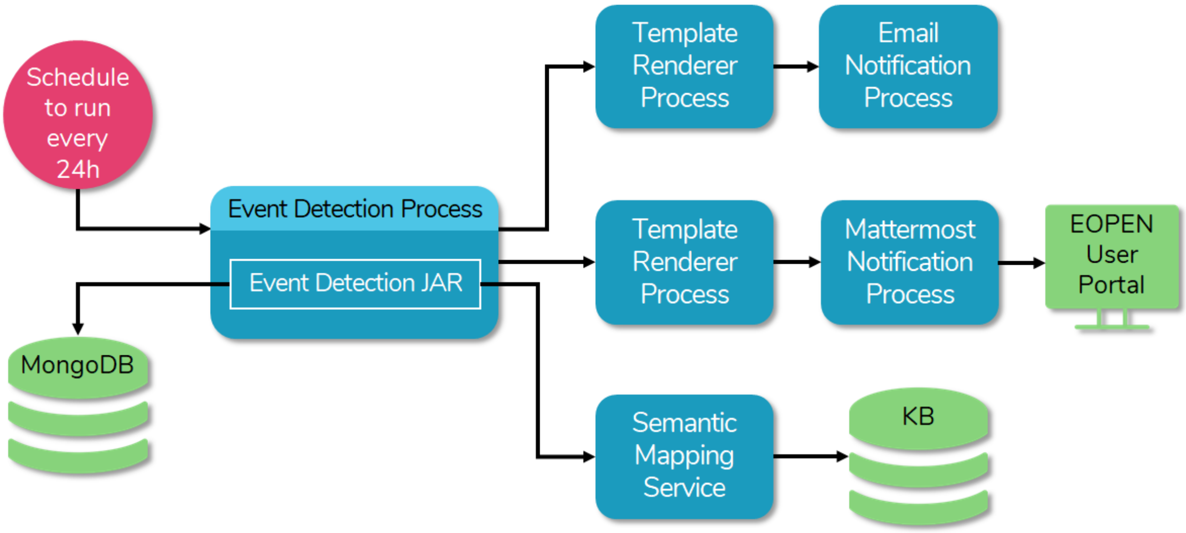

Schedule to run every 24h
Event Detection Process
Event Detection JAR
MongoDB
Template Renderer Process
Email Notification Process
Template Renderer Process
Mattermost Notification Process
EOPEN User Portal
Semantic Mapping Service
KB

Figure 11: The complete workflow of the event detection implementation

If an event is discovered, the Semantic Mapping service (reported in D5.2) is called with the aforementioned JSON outcome given as input. The JSON is then mapped to RDF triplets and results are stored to the Knowledge Base, which allows semantic queries, e.g. a GeoSPARQL query to retrieve events in a bounding box.

Furthermore, the Event Detection process is connected with another process that utilises the produced JSON file with the results. The Template Renderer process takes the values of the JSON attributes and uses them to synthesize a notification in natural language. An example is given below.

This is a detected event in JSON format:

```
{
    "timestamp" : 1588608753523,
    "usecase" : "Snow",
    "language" : "English",
    "score" : 3.5,
    "change" : 2.56,
    "location" : "Kemijoki",
    "point" :
    {
          "x" : 24.502557,
          "y" : 65.780364
    },
    "keywords" : ["rivers", "snow", "melt", "spring", "rains", "rapid",
"flow", "flooding", "levels", "damages"]
}
```

And this is the transformation to a notification:

256% raise in English tweets about snow on Mon, 04 May 2020 16:12. Possible event in Kemijoki!

rivers · snow · melt · spring · rains · rapid · flow · flooding · levels · damages

The Template Renderer process is then connected separately to two processes that are able to publish the notification in two alternative ways. The first one, i.e. the Email Notification process, sends an email with the alert to specified recipients. The second one, i.e. the Mattermost Notification process, pushes an alert to the relevant Mattermost channels and the alert can be visualised as a new message in the Notifications Dashboard of the EOPEN User Portal. For more details about the Template Renderer, the Email Notification and the Mattermost Notification processes, the reader is referred to D6.4.

The described workflow can be extended with more processes that will be triggered when an event is detected. For example, if there is an event in the tweets about floods, the estimated location can be used to retrieve satellite imagery from that area and perform change detection, in order to investigate the possibility of a flooding incident.

3.4 Evaluation

3.4.1 Quantitative evaluation

In order to be able to evaluate the proposed event detection methodology by measuring the success of prediction with typical metrics, such as the F-score, ground truth is necessary. In the examined problem, ground truth should refer to all the real incidents that happened, but it can be a very hard task to specify them.

Nevertheless, we have designed an experiment that concerns PUC1 and is able to provide a quantitative evaluation of the approach. We have collected all the flooding incidents that occurred in Italy in the years 2017 and 2018 (Table 3) to use as ground truth and applied the event detection method on the Italian tweets that were collected in this time period and include keywords related to floods.

Table 3: Real flooding incidents in Italy in 2017-2018 (ground truth)

Date	Incident	Reference
04/08/2017	Flood in Veneto region	http://floodlist.com/europe/italy-flash-floods-and-landslides-in-veneto-region-leave-1-dead
09-10/09/2017	Flood in Livorno	http://floodlist.com/europe/italy-floods-livorno-september-2017
12/12/2017	Flood in Emilia-Romagna region	https://watchers.news/2017/12/12/major-flooding-after-rivers-overflow-in-emilia-romagna-italy/
04/10/2018	Flood in Calabria	http://floodlist.com/europe/italy-floods-calabria-sicily-october-2018
10/10/2018	Flood in Cagliari, Sardinia	https://www.ansa.it/english/news/general_news/2018/10/10/bad-weather-hits-sardinia_f18ae195-1275-4edd-a1be-61e81e7960e8.html

26-30/10/2018	Flood in Veneto	https://www.bbc.com/news/world-europe-46029302
03/11/2018	Flood in Sicily	http://floodlist.com/europe/italy-flash-floods-sicily-novermber-2018

Having these dates as ground truth, we examine each date of years 2017 and 2018 and define as:

- True Positive (TP), when the date of a detected event matches the date of a real incident (+/- one day);
- False Positive (FP), when the date of a detected event does not match the date of a real incident;
- False Negative (FN), when a real incident is not detected overall.

The metrics we use for the results are precision, recall, and F-score.

As mentioned in Section 3.2, the proposed approach involves two parameters, i.e. (i) the sample to compare each point and (ii) the threshold of z-score above which the point is considered an event. In this evaluation, we examine different values of these parameters: the sample can be the whole year, the total month, or the previous thirty days, while the threshold can be 2.5, 3, or 3.5.

The results are displayed in Table 4. The best precision (1.00) is achieved for the year sample and the 3.5 threshold, followed by thresholds 3 (0.67) and 2.5 (0.67). This means that all or most outliers in the annual fluctuation of tweets are linked to real events. On the other hand, the best recall (0.86) is achieved for the sample of the last thirty days for all three thresholds, which means that more real events were able to be discovered, but the lower precision indicates that many false alerts were detected. It can be seen in the results that precision and recall are inversely proportional in this problem, so parameterization depends on the target: comparison for the whole year and higher thresholds to get accurate detection or comparison of the recent days to get as many detected events as possible. However, F-score expresses the balance between precision and recall and the best score (0.73) has been achieved for the same parameters as those of the best precision.

It should be noted here that out of the seven real incidents (Table 3), only the flood in Veneto region in the August of 2017 was not be detected with any parameters, possibly because it was not covered enough on social media.

Table 4: Evaluation of the event detection methodology

Sample to compare	Threshold	TP	FP	FN	Precision	Recall	F-score
Year	3.5	4	0	3	**1.00**	0.57	**0.73**
Month	3.5	3	12	4	0.20	0.43	0.27
Last 30	3.5	6	17	1	0.26	**0.86**	0.40
Year	3	4	2	3	0.67	0.57	0.62

Month	3	5	14	2	0.26	0.71	0.38
Last 30	3	6	14	1	0.30	**0.86**	0.44
Year	2.5	4	2	3	0.67	0.57	0.62
Month	2.5	5	17	2	0.23	0.71	0.34
Last 30	2.5	6	18	1	0.25	**0.86**	0.39

It is also interesting to mention that some of the False Positives might not refer to occurring flooding incidents, but they can still be considered as events on social media, in the sense that a large number of tweets is discussing about them. Indicative cases can be floods from the past, floods in other countries and news that indirectly relate to floods. Some examples of such detected events are given in Table 5.

Table 5: Detected events that are not linked

Date	Incident	Reference
08/04/2017	Pastificio Rummo: the averted bankruptcy of the pasta factory that suffered extensive damage in the Sannio flood	https://napoli.fanpage.it/pastificio-rummo-evitato-il-fallimento-l-azienda-aiutata-dalla-campagna-social-dopo-l-alluvione/
19/07/2017	Flood in Valtellina in 1987	https://en.wikipedia.org/wiki/Val_Pola_landslide
04/11/2017	Flood in Florence in 1966	https://en.wikipedia.org/wiki/1966_flood_of_the_Arno
05/05/2018	Flood in Sarno in 1998	https://www.italyonthisday.com/2016/05/mudslides-campania-sarno-1998.html
29/06/2018	Flood in India	http://floodlist.com/asia/india-jammu-kashmir-floods-june-2018
09/07/2018	Flood in Japan	https://en.wikipedia.org/wiki/2018_Japan_floods
31/08/2018	“Binderemo per l’alluvione”: a sentence captured during the interceptions on a procurement investigation regarding the deaths during the floods on September 10, 2017	https://www.lanazione.it/cronaca/brinderemo-alluvione-intercettazioni-1.4112986

Finally, in order to observe the apparent relation between the fluctuation of the collected tweets per day and the events detected (both True and False Positives) by our approach, Figure 12 is presented.

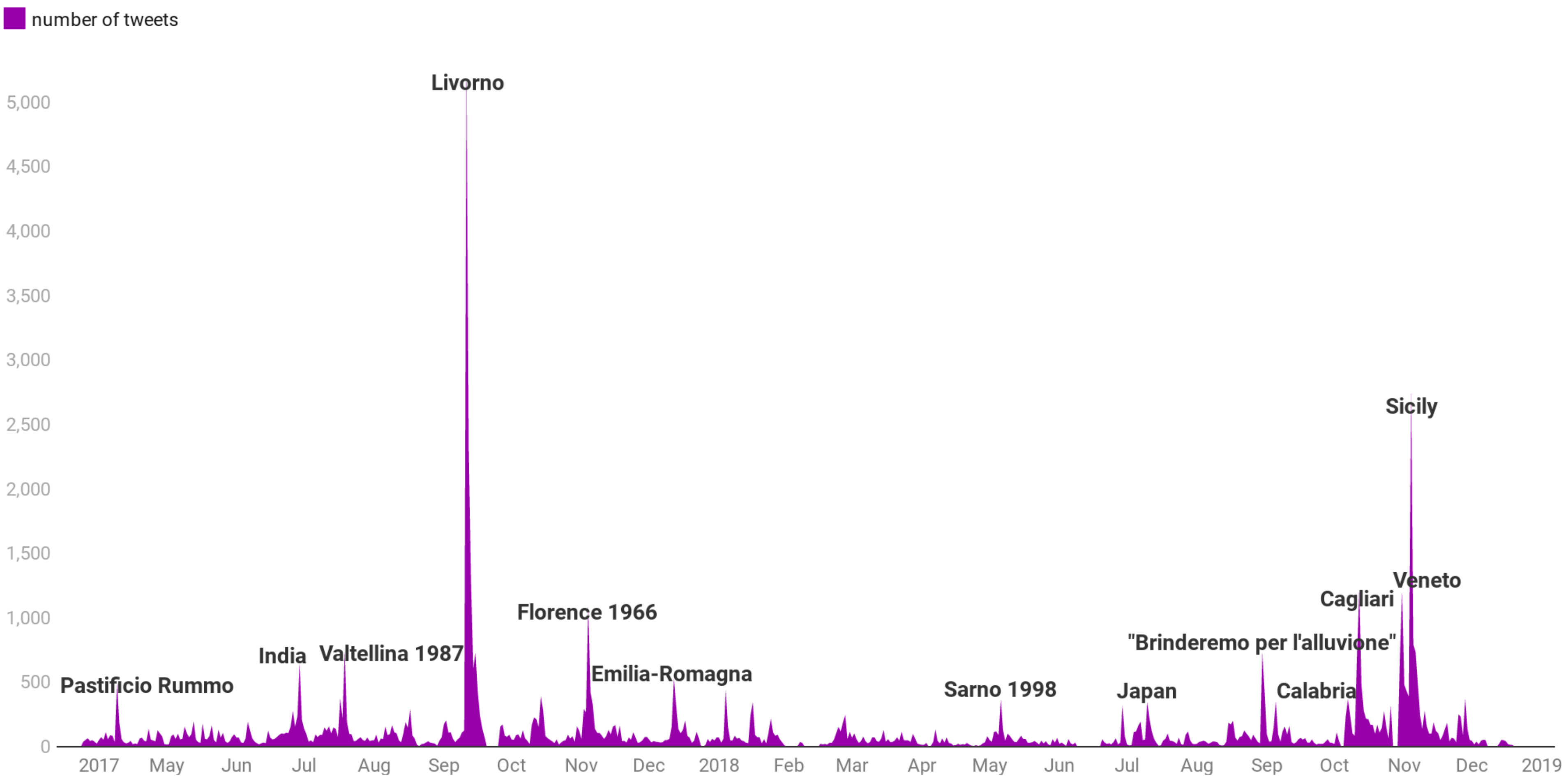


Figure 12: Detected events based on the fluctuation of the number of Italian, flood-related tweets collected per day in years 2017 & 2018

3.4.2 **Qualitative evaluation**

Despite the lack of ground truth, we would also like to show what the proposed event detection method is able to provide to PUC2 and PUC3 with two exercises. As parameters we have selected the pair that achieved the best F-score in the above evaluation, i.e. an annual sample and a threshold of 3.5.

The first exercise focuses on Korean tweets about food security that have been collected during the year of 2019. The detected events are listed in Table 6 and include (i) the date, (ii) the measured z-score, (iii) the most meaningful keywords out of the top ten most-mentioned words, (iv) their translation, (v) a short description of the real incident they refer to, and (vi) a relevant Web link. The detected events are also visualised in Figure 13.

Table 6: Detected events in Korean tweets about food security in 2019

Date	z-score	Top keywords (original)	Top keywords (translation)	Incident	Reference
01/08/2019	8.34	외교장관회의	Foreign Ministers' Meeting	The 9th East Asia Summit (EAS) Foreign Ministers' Meeting in Bangkok, Thailand on 2 August 2019	https://asean.org/chairmans-statement-9th-east-asia-summit-foreign-ministers-meeting/
		동아시아정상회의	East Asia Summit		
		방콕으로	To Bangkok		
		아세안+	ASEAN+		
		아세안지역안보포럼	ASEAN Regional Security Forum		
14/09/2019	9.78	환경부	Ministry of Environment	Korean Thanksgiving on 12-14 September 2019	https://en.wikipedia.org/wiki/Chuseok
		한가위를	Thanksgiving		
		보내시길	Send		
		mevpr	(The Twitter account of Ministry of Environment)		
		송편과	Songpyeon (traditional rice cake)		
14/11/2019	4.32	콩순이	Kongsoon (Character)	A Korean environmental animation video	https://www.youtube.com/watch?v=wDGvjz
		mevpr	(The Twitter		

			account of Ministry of Environment)	published on 13 November 2019	hbjbU
		환경부	Ministry of Environment		
		구독하고	Subscribe		
		이야기를	Tell stories		
25/11/2019	7.73	억원의	100 million won	ASEAN-ROK Commemorative Summit in Busan on 25-26 November 2019	https://www.asean2019.go.th/en/meeting/asean-rok-commemorative-summit/
		아세안	ASEAN		
		생산유발	Production inducement		
		효과와	With effect		
		부산시는	Busan city		

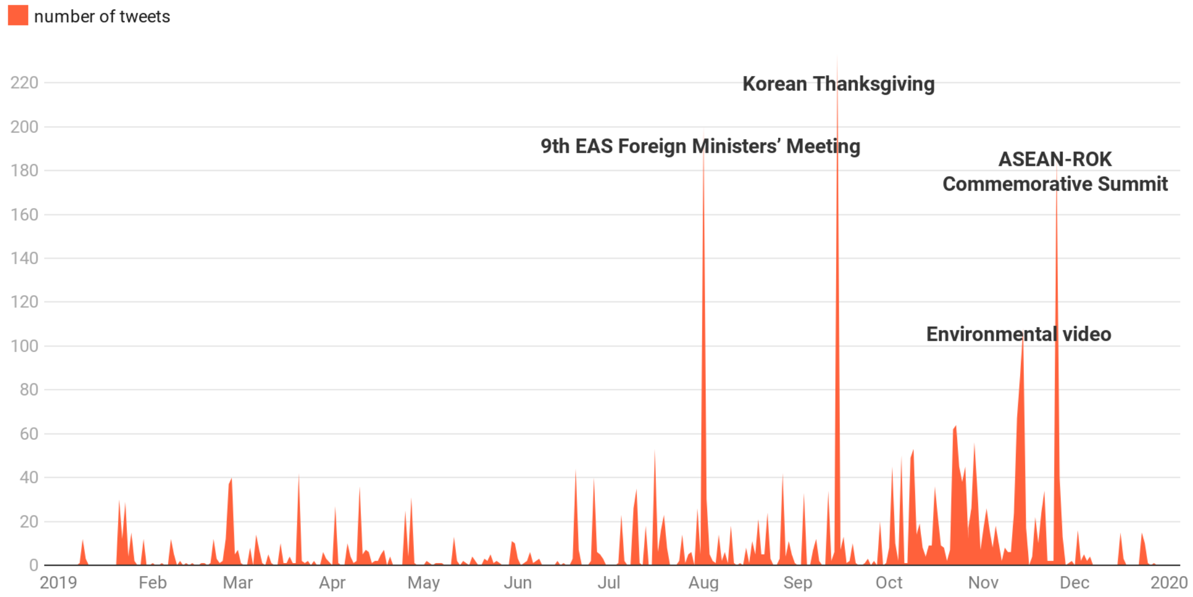


Figure 13: Detected events in Korean tweets about food security in 2019

The second exercise focuses on Finnish tweets about snow that have been collected during the year of 2019. The detected events are listed in Table 7, again with the same information as before, and also visualised in Figure 14.

Table 7: Detected events in Finnish tweets about snow in 2019

Date	z-score	Top keywords (original)	Top keywords (translation)	Incident	Reference
16/01/2019	3.51	lunta	Snow	Blizzard on southern	https://www.helsinkitimes.fi/finland/fi
		lumi	Snow		

		lumimyräkkä	Snowstorm	parts of Finland	nland-news/domestic/16122-blizzard-to-dump-up-to-20cm-of-snow-on-southern-finland.html
		etelään	Southward		
		ajokeli	Driving conditions		
17/01/2019	6.83	lunta	Snow	Blizzard on southern parts of Finland	https://www.helsinkitimes.fi/finland/finland-news/domestic/16122-blizzard-to-dump-up-to-20cm-of-snow-on-southern-finland.html
		lumi	Snow		
		aamulla	In the morning		
		lumipyry	Snowstorm		
		suomessa	Finland		
30/01/2019	3.76	lunta	Snow	Snowfall episode (29–30 January)	https://www.eumetsat.int/website/home/Images/ImageLibrary/DAT_4250628.html
		lumi	Snow		
		helsinki	Helsinki		
		hiekottaa	Hiekotin (snow spreader)		
		lumenauraajat	Snow plow times		
03/02/2019	5.58	lunta	Snow	Snow melting	https://www.mtvuutiset.fi/artikkeli/loskaa-pukkaa-jos-haluat-hiihtaa-etela-suomessa-tee-se-nyt-helmikuu-sulattaa-lumet-pian-pois-olikohan-talvipakkaset-tassa/7269700#gs.a9rw7y
		lumi	Snow		
		talvi	Winter		
		sulattaa	Melt		
		helsinki	Helsinki		
07/02/2019	3.99	lunta	Snow	Snowstorm in Helsinki	
		lumi	Snow		
		helsinki	Helsinki		
		illalla	In the evening		
		vastaanottopaikoille	Reception sites		
07/11/2019	3.96	lunta	Snow	First snow of snow	https://newsnowfinland.fi/domestic/m
		lumi	Snow		

		talvi	Winter	season in Helsinki	[orning-headlines-thursday-7th-november-2019]()
		helsinki	Helsinki		
		ensilumi	First snow		

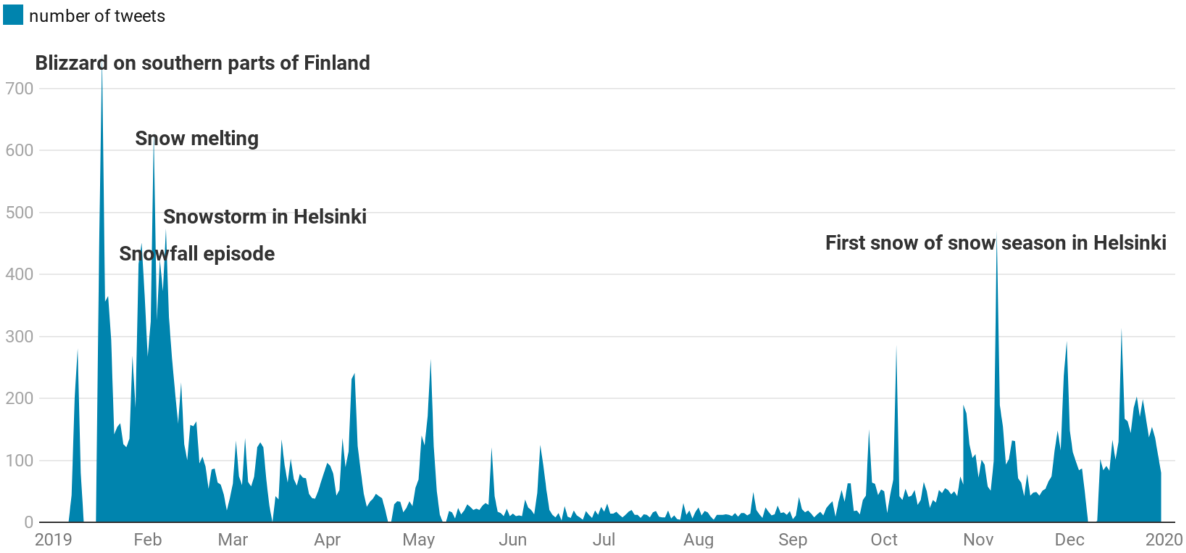


Figure 14: Detected events in Finnish tweets about snow in 2019

4 COMMUNITY DETECTION

Community detection is the scope of WP4's task 4.5 "Community detection in Social Media" and aims to discover communities of social media users, i.e. Twitter accounts that are interlinked through their online behaviour. Additionally, the identification of the most influential users, namely key-players, is also considered significant part of this task.

D4.1 included a brief description of the community detection methodology we have developed as well as a section with the state of the art. In this document, related work is extended with more recent publications (4.1) and the methodology is presented with more details (4.2). After experimentation, the Louvain algorithm is adopted for detecting communities and a novel entropy-based centrality measure is proposed to identify key-players.

Regarding implementation, the community detection method has been developed as a service and it has been integrated to the EOPEN platform as a Docker image (4.3.1). Moreover, a respective dashboard has been created (4.3.2) that utilizes the community detection service and enables end users to visualize Twitter communities and key-players.

This section concludes with a series of experiments. The first experiment (4.4.1) compares different community detection approaches in terms of various metrics and also investigates how these metrics change in regards to time (days). The second experiment (4.4.2) compares the communities detected in the different use cases of EOPEN, while the last experiment (4.4.3) examines the key-players identification during a real-world incident.

4.1 Related work

As mentioned in the introduction of the section, state of the art in the community detection field has already been described in Section 6.1 of D4.1. However, we extend it here with some related work that has been published more recently.

(Bedi & Sharma, 2016) present a survey of the existing algorithms and approaches for the detection of communities in social networks. In their work they place the various community detection algorithms in the following categories: (i) graph partitioning-based approaches, (ii) clustering-based, (iii) genetic algorithms-based, (iv) label propagation-based, (v) semantics-based, (vi) clique-based for overlapping communities, (vii) non-clique-based, and (viii) methods for dynamic networks. In addition, they include some standard datasets for community detection.

The survey of (Javed et al., 2018) aimed to serve as an up-to-date report on the evolution of community detection. Apart from providing a taxonomy of the related algorithms and highlighting the strengths and weaknesses of each approach, they include a comparison based on the measurement of important quality metrics regarding the performance and the computational cost.

(Yang et al., 2016) also report a comparative analysis of community detection algorithms, but focus on eight state-of-the-art techniques to measure their accuracy and computing time. Moreover, they provide techniques that can determine which algorithm is the most suited based on the properties of the network.

Aside from surveys, (Wang et al., 2018) propose a community detection algorithm that combines unsupervised extreme learning machine, which maps the adjacency matrix of the nodes distance to low-dimensional space, with weighted k-means to label the groups. (Veldt et al., 2018) introduce a new community detection framework that is called LamdaCC and is based on a specially weighted version of correlation clustering. A key component in this methodology is the clustering resolution parameter λ, which controls the size and structure of the formed clusters.

Focusing on identifying communities in social networks, (Ahajjam et al., 2018) present a scalable and deterministic approach that consists of two steps: first, retrieving the leader nodes, i.e. individuals that are influential and drive trends, and then detecting communities based on the similarity between nodes. On the other hand, (Whang et al., 2016) target the common problem that individuals might belong to multiple communities and suggest an overlapping community detection algorithm using a seed expansion approach, where centroid vertexes are considered good seeds and their neighbourhood a seed region. Finally, (He et al., 2018) propose a meta-approach, namely HICODE, which tries to discover both dominant and hidden community structure by applying structure weakening methods.

4.2 Description of the methodology

The scope of the proposed methodology is twofold: on one hand, the detection of communities in networks derived from social media (Gialampoukidis et al., 2017) and, on the other hand, the identification of key-players, i.e. most influential user accounts (Gialampoukidis et al., 2016).

The first step is to denote by $G(N, L)$ the social network, with N nodes that each represents a Twitter user account and L links, where a link between two users (i, k) exists if user n_i mentions or is mentioned by user n_k. This relationship has been preferred over other, e.g. following, because it expresses a more temporary connection between users and user communities shift continuously based on trending topics and events. Next, community detection algorithms can be applied to divide the network into groups of users that are more densely connected to each other within the group than to the rest of the network outside of the group. The bibliography suggests several community detection algorithms, such as Edge Betweenness (Newman & Girvan, 2004), Fast Greedy (Clauset et al., 2004), Label Propagation (Raghavan et al., 2007), Louvain (Blondel et al., 2008), Walktrap (Pons & Catapy, 2005), and Infomap (Rosval et al., 2009; Bohlin et al., 2014). After comparing the above approaches in terms of different metrics (reported in Section 4.4), the fast and scalable Louvain algorithm has been selected. The outcome of the algorithm is the number of detected communities and the set of nodes that belongs to each community.

In order to identify key-players, an entropy-based centrality measure is used, namely the Mapping Entropy Betweenness (MEB) centrality, which takes into consideration the betweenness centrality of nodes (Gialampoukidis et al., 2016).

Specifically, the betweenness centrality (BC) of node n_k is based on the number of shortest paths $g_{ij}(n_k)$ from node n_i to node n_j that pass through node n_k to the number of all shortest paths g_{ij} from node n_i to node n_j, summed over all pairs of nodes (n_i, n_j) and normalised by its maximum value, i.e. $(N^2 - 3N + 2)/2$:

$$BC_k = \frac{2\sum_{i<j}^{N} \frac{g_{ij}(n_k)}{g_{ij}}}{N^2 - 3N + 2}$$

When a node acts as a bridge between many pairs of nodes, then its betweenness centrality is relatively high. For that reason, MEB centrality focuses on the betweenness centrality of a node, but is also further elaborated, by taking into account the betweenness centrality of its first neighbours. MEB is defined as follows:

$$MEB_k = -BC_k \sum_{n_i \in N(n_k)} logBC_i$$

where the weight assigned to BC_k is the sum of all $-logBC_i$ over the neighbourhood of node n_k. The nodes with the maximum MEB centrality are considered key-players.

4.3 Implementation and integration to the EOPEN platform

4.3.1 Community detection service

The community detection and key-players identification methodology described in 4.2 has been implemented as an API that developers/users can call in order to retrieve the communities and the key-players inside a tweets collection in a given time period. The API is implemented in JAVA and utilises a script in R language for the algorithms. The input parameters to the service are the language and use case of interest (e.g. “Finnish” and “Snow” for PUC3) and the examined time period, i.e. a start date and an end date.

An example of calling the API can be seen here:

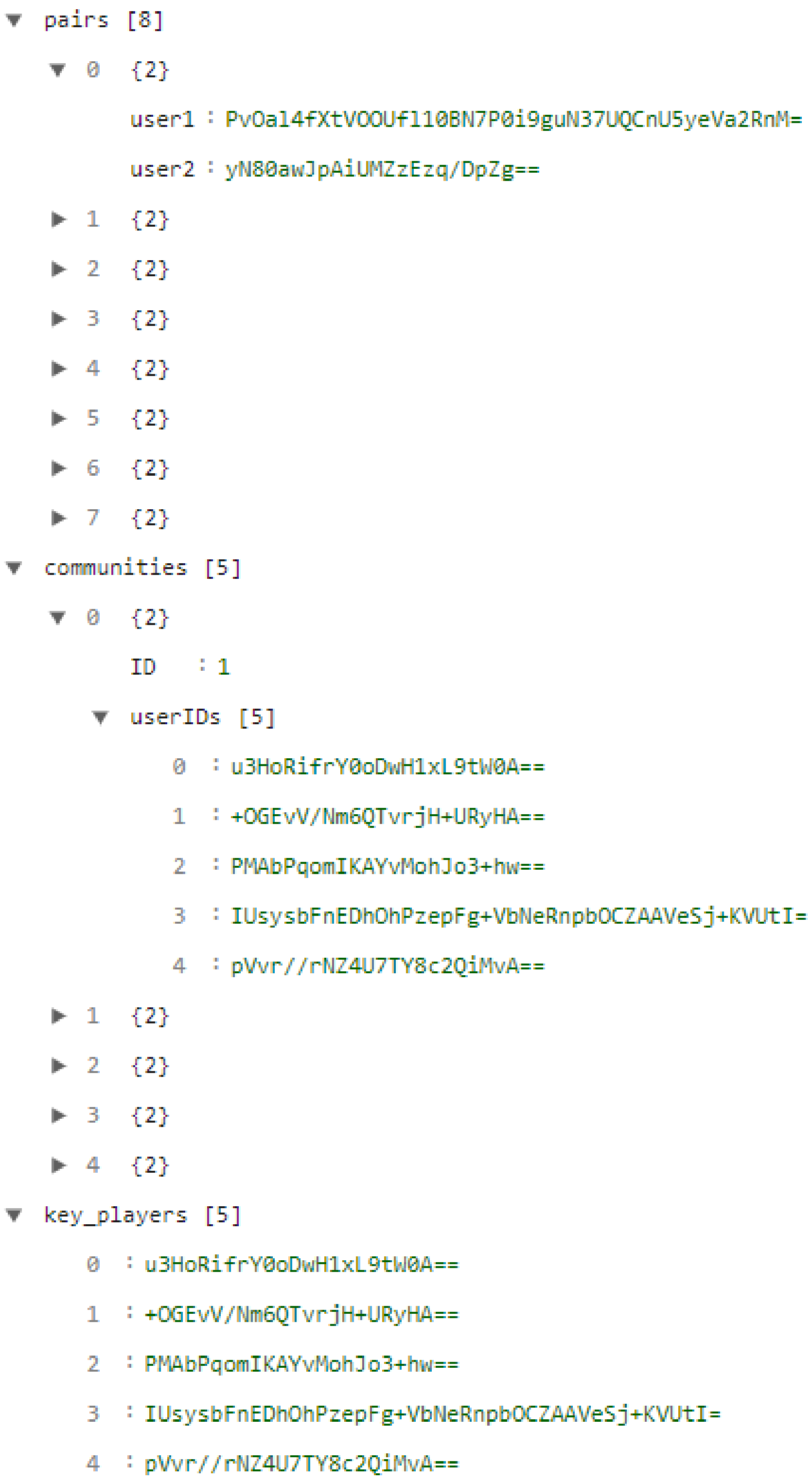
```
http://<IP>:<port>/api/communityDetection/usingDays?language=<language>&
pilot=<pilot>&from=<YYYY-MM-DD>&to=<YYYY-MM-DD>

e.g.

http://<IP>:<port>/api/communityDetection/usingDays?language=Finnish&pil
ot=Snow&from=2020-05-01&to=2020-05-07
```

When the service is called, it communicates with the MongoDB database where the collected social media data are stored and retrieves the tweets of the requested collection that have been posted in the requested time frame. Then, every tweet is examined and if the user (author) mentions another user account, a pair of these two is created. After all tweets are examined, the list of pairs is printed to a text file and this file is fed to the R script, which performs community detection and key-players identification. The outcome of the script is converted to JSON format and is returned as the service’s response. An example of the JSON response can be seen in Figure 15. It includes the pairs of users who are connected though the “mentioning” interaction, the detected communities (their ID and the set of users it comprises) and the top key-players. It should be noticed here that all user IDs are pseudonymized, so as to protect the privacy of the individuals behind these Twitter accounts.

Furthermore, the service has been dockerised and deployed in the EOPEN platform, thus it can be accessed by processes or the User Portal.



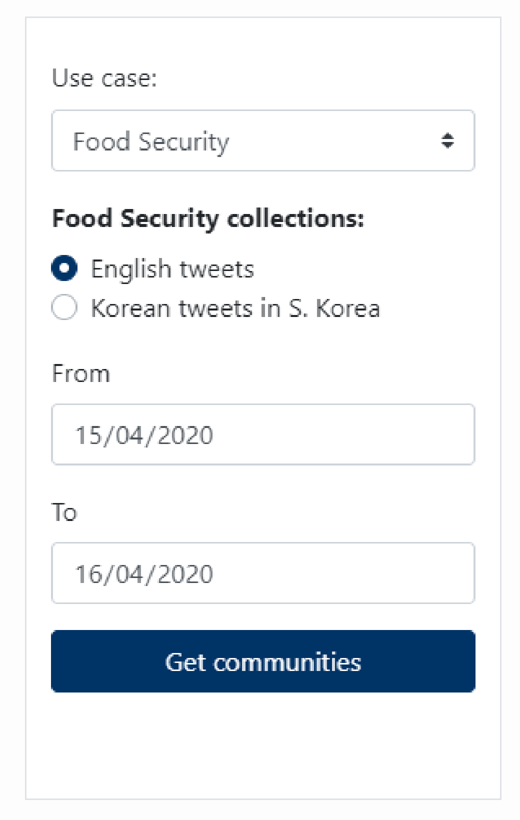
```
▼ pairs [8]
    ▼ 0  {2}
          user1 : PvOal4fXtVOOUf110BN7P0i9guN37UQCnU5yeVa2RnM=
          user2 : yN80awJpAiUMZzEzq/DpZg==
    ▶ 1  {2}
    ▶ 2  {2}
    ▶ 3  {2}
    ▶ 4  {2}
    ▶ 5  {2}
    ▶ 6  {2}
    ▶ 7  {2}
▼ communities [5]
    ▼ 0  {2}
          ID     : 1
        ▼ userIDs [5]
               0  : u3HoRifrY0oDwH1xL9tW0A==
               1  : +OGEvV/Nm6QTvrjH+URyHA==
               2  : PMAbPqomIKAYvMohJo3+hw==
               3  : IUsysbFnEDhOhPzepFg+VbNeRnpbOCZAAVeSj+KVUtI=
               4  : pVvr//rNZ4U7TY8c2QiMvA==
    ▶ 1  {2}
    ▶ 2  {2}
    ▶ 3  {2}
    ▶ 4  {2}
▼ key_players [5]
      0  : u3HoRifrY0oDwH1xL9tW0A==
      1  : +OGEvV/Nm6QTvrjH+URyHA==
      2  : PMAbPqomIKAYvMohJo3+hw==
      3  : IUsysbFnEDhOhPzepFg+VbNeRnpbOCZAAVeSj+KVUtI=
      4  : pVvr//rNZ4U7TY8c2QiMvA==
```

Figure 15: An example output of the community detection service

4.3.2 Visualisation of communities in the EOPEN User Portal

The EOPEN User Portal allows the users to implement their own components with the Vue.js[4] framework and then utilise them to synthesise a customised dashboard. Exploiting this functionality, we have developed four new components and created a new dashboard that offers end users the ability to use the aforementioned community detection service in a user-friendly way and visualize the results.

The first component (Figure 16), "Communities Filters", is a form where the user is able to select on which collection of tweets and in which time period the community detection algorithm will run. The collection can be defined by selecting the use case from a dropdown box and the language from radio buttons, while the time period can be defined by selecting a "from" date and a "to" date. By clicking the "Get communities" button, the community detection service is called with the selected options as input parameters.



Use case: Food Security

Food Security collections:

● English tweets
○ Korean tweets in S. Korea

From 15/04/2020

To 16/04/2020

Figure 16: Communities Filters component

The results of the service are used to visualize in the "Communities Graph" component (Figure 17) the social network of Twitter accounts that are connected through mentioning each other. The unique users found in the pairs are displayed as nodes, pairing is displayed as edges between the two users/nodes, and communities are expressed as different colors of the nodes and edges. As mentioned before, the IDs of the users are pseudonymized for privacy reasons. The graph can also be zoomed in and out, to take a closer look to the detected communities.

[4] https://vuejs.org/

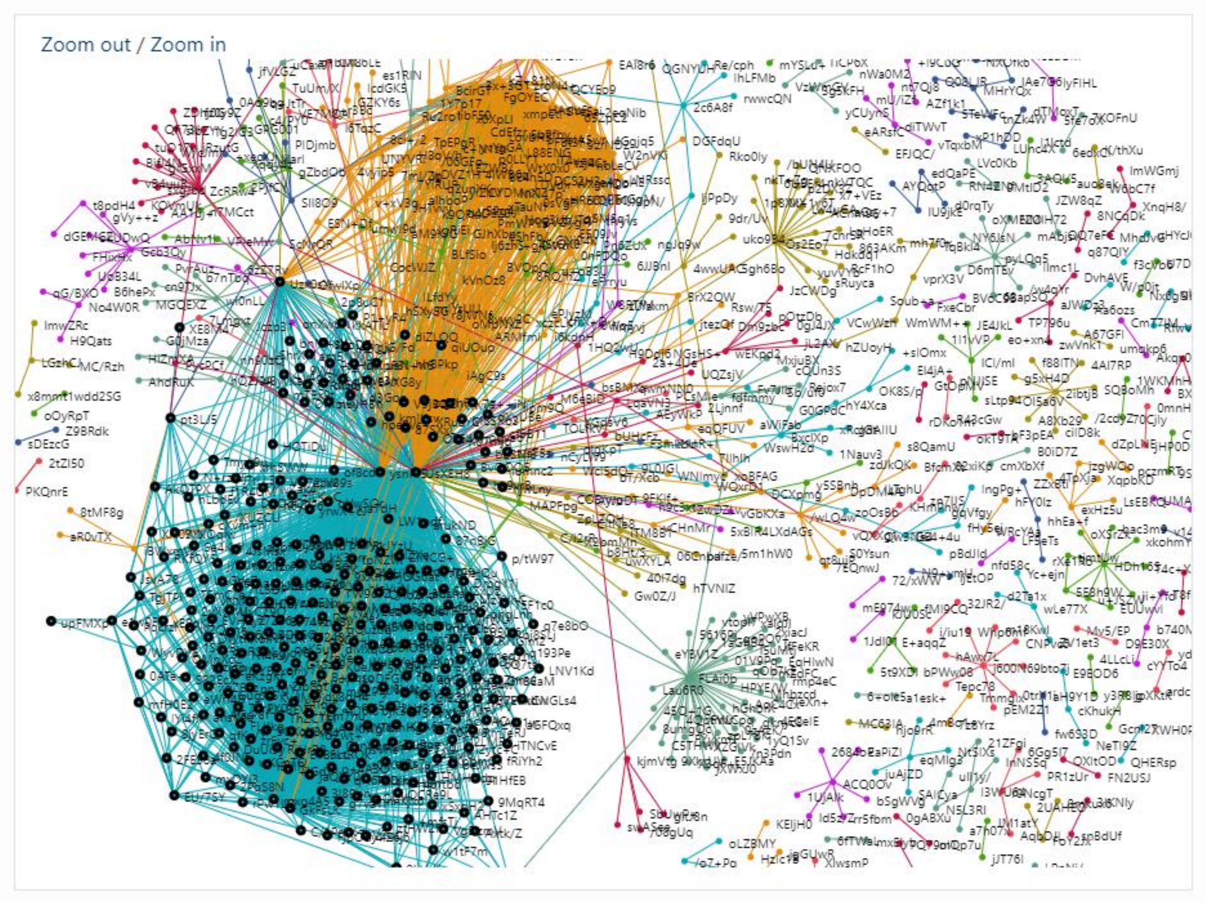
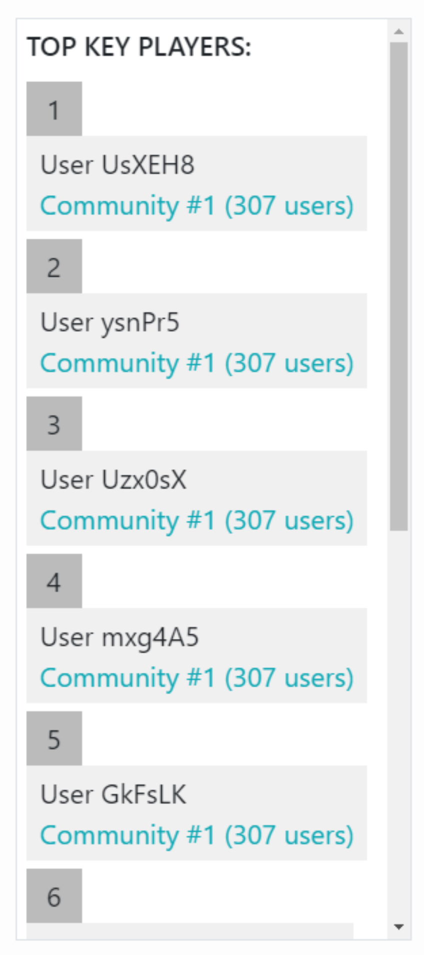


Figure 17: Communities Graph component

The service also returns the top (up to ten) key-players that are hubs in the graph and seem to influence the other users on Twitter. The sorted list is displayed in the “Top Key-Player List” component (Figure 18), where the pseudonymized ID of each key-player and the community they belong to are shown. The community is mentioned by its ID, along with the number of users it comprises, and is also coloured with the same colour of the related nodes. When clicking on the coloured text, the nodes of the community are highlighted in the graph (see the bottom left community in Figure 17) to indicate the position of the community.



TOP KEY PLAYERS:

1

User UsXEH8

Community #1 (307 users)

2

User ysnPr5

Community #1 (307 users)

3

User Uzx0sX

Community #1 (307 users)

4

User mxg4A5

Community #1 (307 users)

5

User GkFsLK

Community #1 (307 users)

6

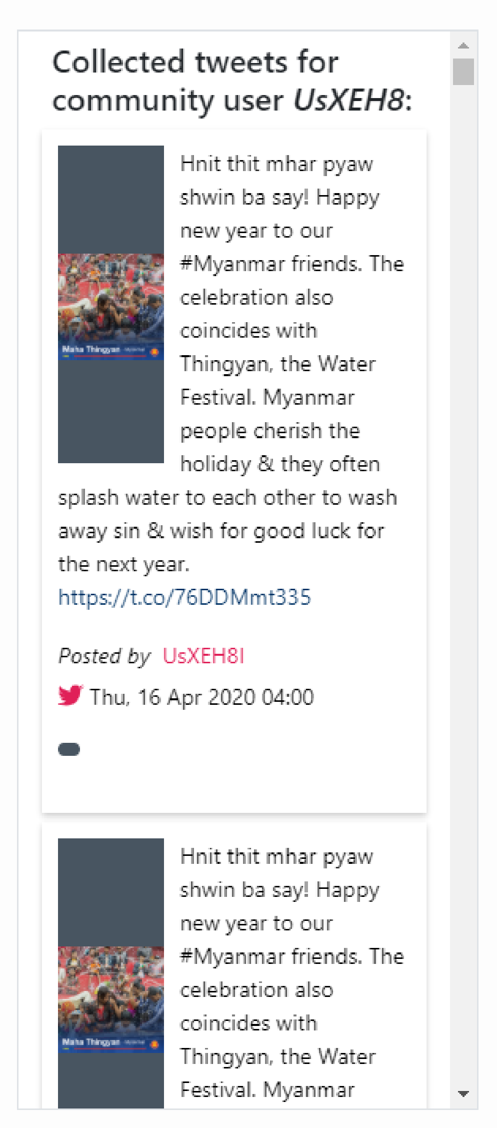
Figure 18: Top Key-Player List component

Finally, when a node in the graph is clicked, then the tweets that have been posted by the respective user are displayed in the "User Tweets List" component (Figure 19). The "Tweet Card" template, already implemented from other social media-related components, is reused to show each single tweet in the list.

Having all the above components, a custom dashboard has been created and can be seen in Figure 20. The user defines the filters in the leftmost component and clicks the button to get communities. The network is shown in the second component and the top key-players in the third component. When a user is clicked in the network, their tweets are listed in the rightmost component.

The visualisation of detected communities and identified key-players in a user interface is also demonstrated in a video that presents the various implementations that have been developed in the frame of EOPEN and are related to the collection and analysis of social media. The video is available here to watch:

https://eopen-project.eu/wp-content/uploads/2020/07/social-media-crawler-v3.mp4.



Collected tweets for community user *UsXEH8*:

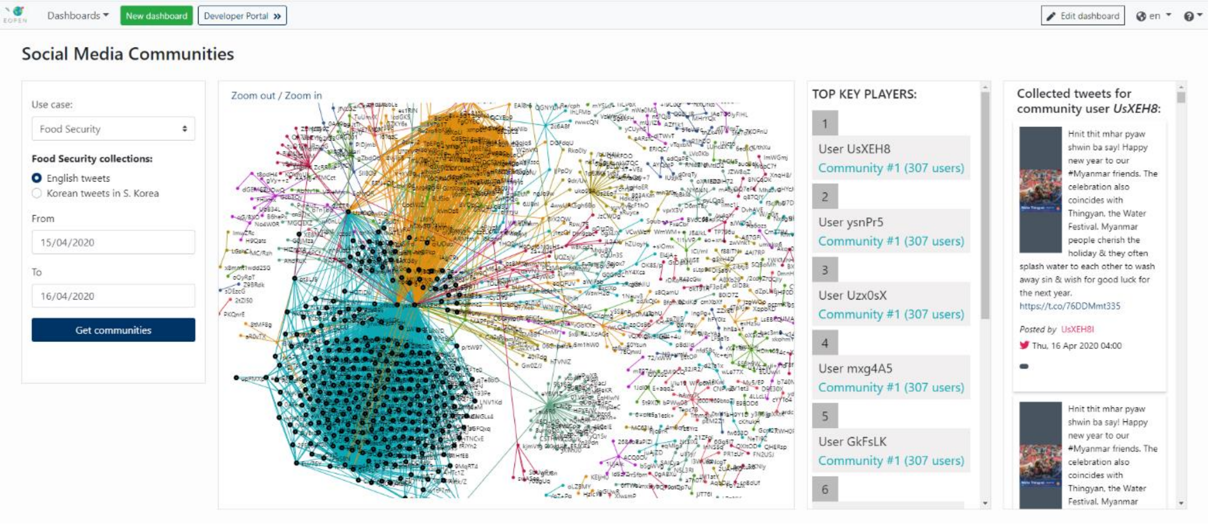
Hnit thit mhar pyaw shwin ba say! Happy new year to our #Myanmar friends. The celebration also coincides with Thingyan, the Water Festival. Myanmar people cherish the holiday & they often splash water to each other to wash away sin & wish for good luck for the next year. https://t.co/76DDMmt335

Posted by UsXEH8I

Thu, 16 Apr 2020 04:00

Hnit thit mhar pyaw shwin ba say! Happy new year to our #Myanmar friends. The celebration also coincides with Thingyan, the Water Festival. Myanmar

Figure 19: User Tweets component



Social Media Communities

Use case: Food Security

Food Security collections:

● English tweets

○ Korean tweets in S. Korea

From 15/04/2020

To 16/04/2020

TOP KEY PLAYERS:

1 User UsXEH8 Community #1 (307 users)

2 User ysnPr5 Community #1 (307 users)

3 User Uzx0sX Community #1 (307 users)

4 User mxg4A5 Community #1 (307 users)

5 User GkFsLK Community #1 (307 users)

6

Collected tweets for community user *UsXEH8*:

Hnit thit mhar pyaw shwin ba say! Happy new year to our #Myanmar friends. The celebration also coincides with Thingyan, the Water Festival. Myanmar people cherish the holiday & they often splash water to each other to wash away sin & wish for good luck for the next year. https://t.co/76DDMmt335

Posted by UsXEH8I

Thu, 16 Apr 2020 04:00

Hnit thit mhar pyaw shwin ba say! Happy new year to our #Myanmar friends. The celebration also coincides with Thingyan, the Water Festival. Myanmar

Figure 20: The “Social Media Communities” dashboard

4.4 **Experiments**

In this section we describe three experiments with different objectives. The first experiment is a comparison of community detection approaches in regards to various metrics, in order to demonstrate the superiority of the selected Louvain algorithm, but it also shows how these metrics change in relation to time. The second experiment examines the results of the proposed community detection methodology, as they differentiate from PUC to PUC. The third and final experiment shows the top key-players that are identified during a real incident for PUC1.

4.4.1 **Comparison of different community detection approaches**

The community detection algorithms to be evaluated are, as mentioned before, the Edge Betweenness, the Fast Greedy, the Label Propagation, the Louvain, the Walktrap, and the Infomap algorithm, all implemented in R. The dataset selected for this experiment is Italian tweets about floods that have been collected between 01/09/2017 and 15/09/2017; this time frame includes the event detected in Section 3.4.1 that refers to the Livorno floods on the 9th and 10th of September 2017. This information is important, because we are able to see how the examined metrics are affected when an incident occurs, apart from the comparison of the approaches.

The first metric to be investigated is the number of detected communities in relation to how many days are taken into consideration (Figure 21). As days increase, the number of collected tweets rises, leading to an increase of pairs of Twitter accounts mentioning each other, which then leads to more communities. There is a notable increment from 9 to 10 days, which shows that the occurred event has a direct effect to the formation of communities. Regarding the different methods, Infomap detects the largest number of communities, while Louvain the smallest, indicating that the latter manages to incorporate new user pairs into the existing communities. The complete results can be found in Table 9 in the Appendix.

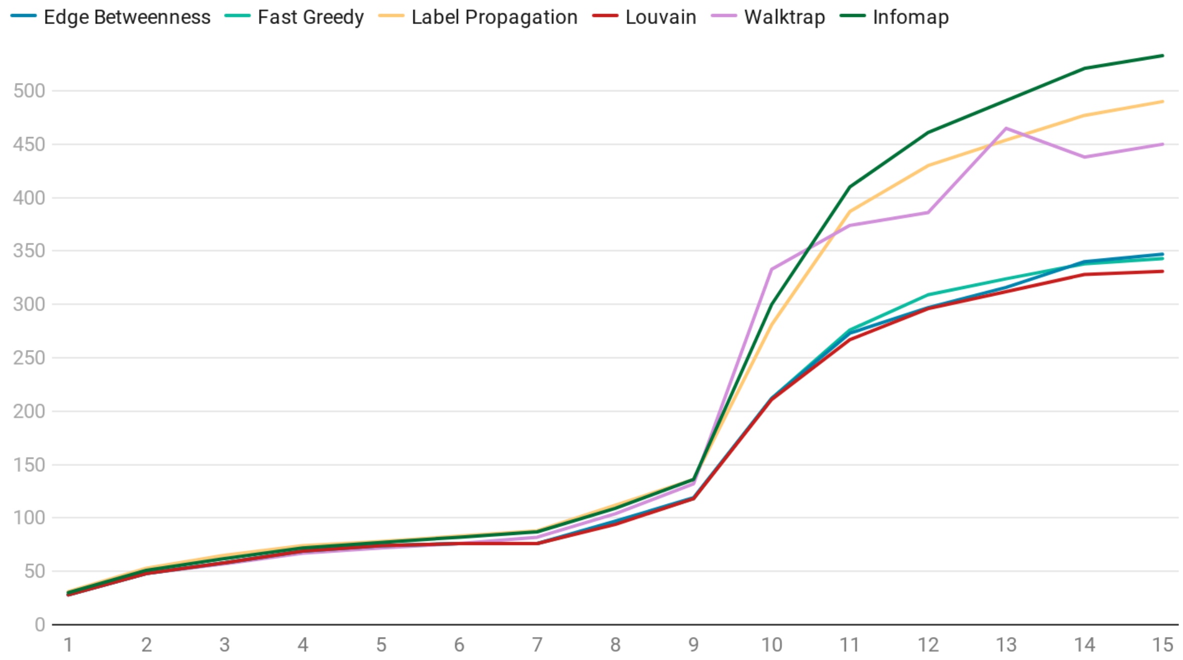


Figure 21: Number of communities detected by each algorithm as days increase

The next metric is the maximum community size (Figure 22; Table 10), i.e. the largest number of users that are assigned in the same community. Again, there is a big increase at 10 days, but this time the increase is much less smooth, highlighting how much larger communities are formed during an event. It is not clear in the figure, but Walktrap shares the same results as Infomap, while the other four algorithms perform the same.

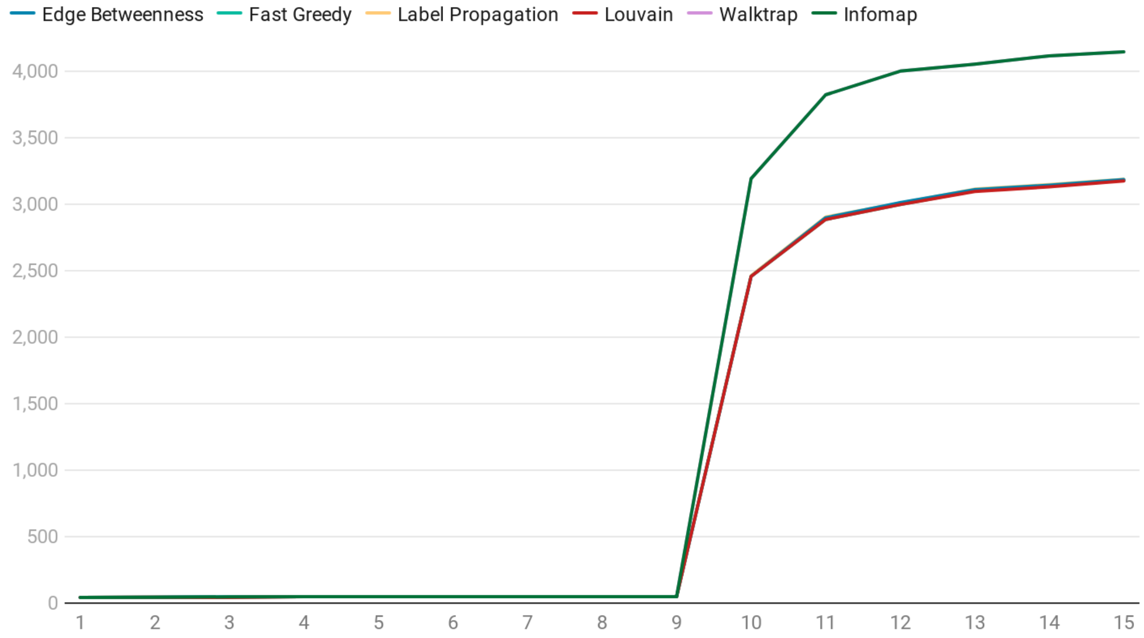


Figure 22: Maximum community size detected by each algorithm as days increase

Modularity is a measure of the structure of networks. It was designed to measure the strength of division of a network into modules, i.e. communities. Networks with high modularity have dense connections between the nodes within modules, but sparse connections between nodes in different modules. The calculation of modularity follows:

$$Q = \frac{1}{2m}\sum_{i=1}^{c}(e_{ii} - a_i^2)$$

where e_{ij} is the fraction of links between a node in community i and a node in community j, a_i is the fraction of links between two members of the community $i, m = \sum_k deg(n_k)$.

Figure 23 and Table 11 show the results of modularity. What is interesting in relation to time is that communities grow denser as days pass (from 1 to 9 days and from 10 to 15), but an event has such a strong effect in the relationships of users that communities are re-formed after the incident (decrease of modularity at 10 days). The Walktrap algorithm detects the least dense communities, while Louvain outperforms all the other methods.

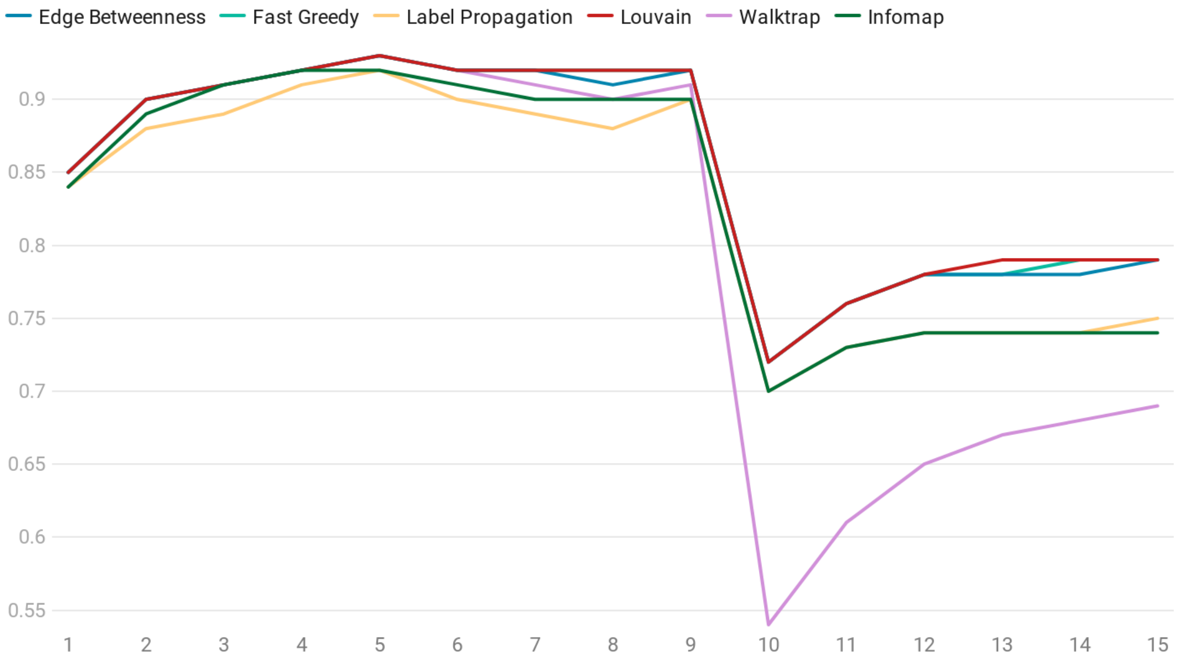


Figure 23: Modularity of communities detected by each algorithm as days increase

In contrast with modularity, where maximizing is desired, code length is a measure that describes random walks on the network and needs to be minimized. Unfortunately, code length can be calculated only for Infomap (Figure 24; Table 12). The constant increment through the days is expected, because the increasing complexity of the graph makes it harder to "travel" from one node to another.

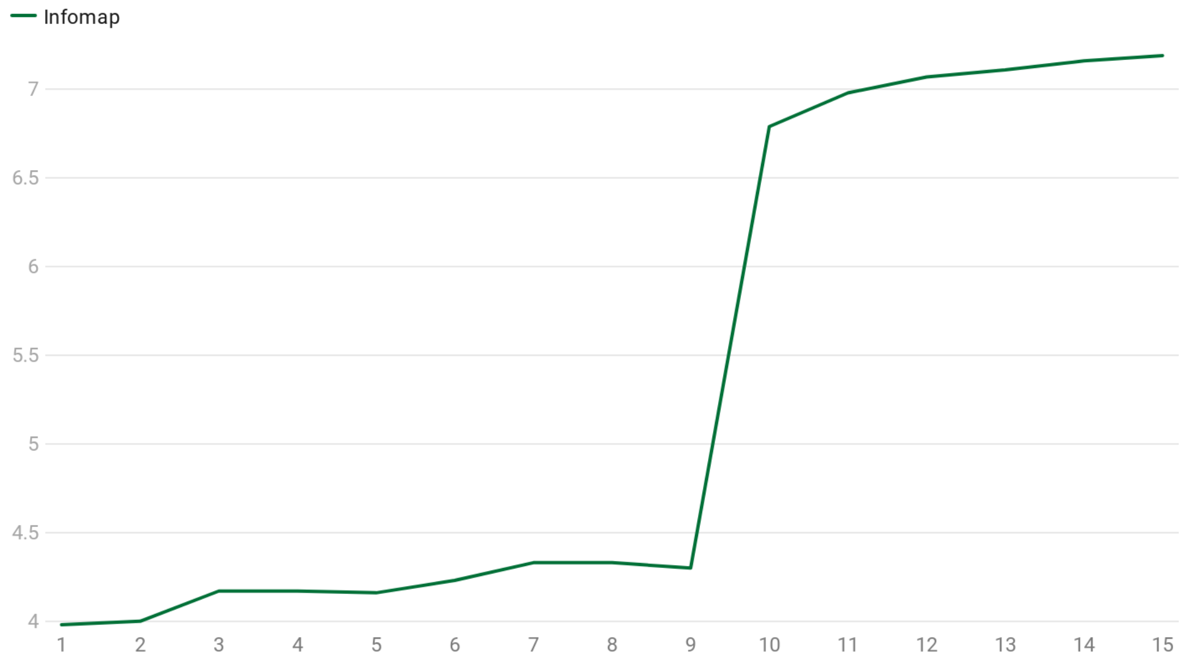


Figure 24: Code length of communities detected by Infomap as days increase

The last metric examined is the execution time of each algorithm (Table 13). In Figure 25 it is evident that Edge Betweenness is by far the slowest approach; even though it needs some milliseconds to run on circa 600 pairs (9 days), it requires more than 20 minutes to run on 10,000 pairs (15 days), contrary to the other algorithms that maintain their fast response and thus are considered much more scalable.

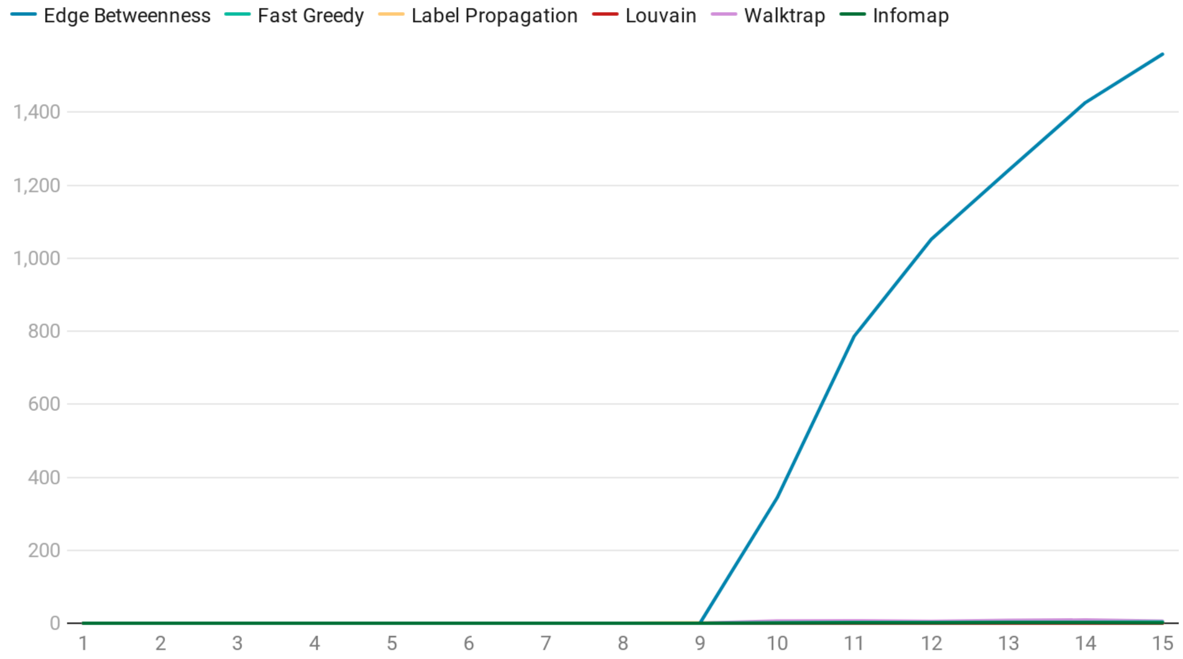


Figure 25: Execution time of each algorithm as days increase

Since in Figure 25 the comparison of the other algorithms is not clear, Figure 26 shows their execution times, excluding the results for Edge Betweenness. Louvain achieves the best average execution time and is also proven to be the most scalable. This, in combination with the highest achieved modularity, signifies the superiority of the Louvain algorithm compared to other methods and justifies our selection to select it for the EOPEN implementations.

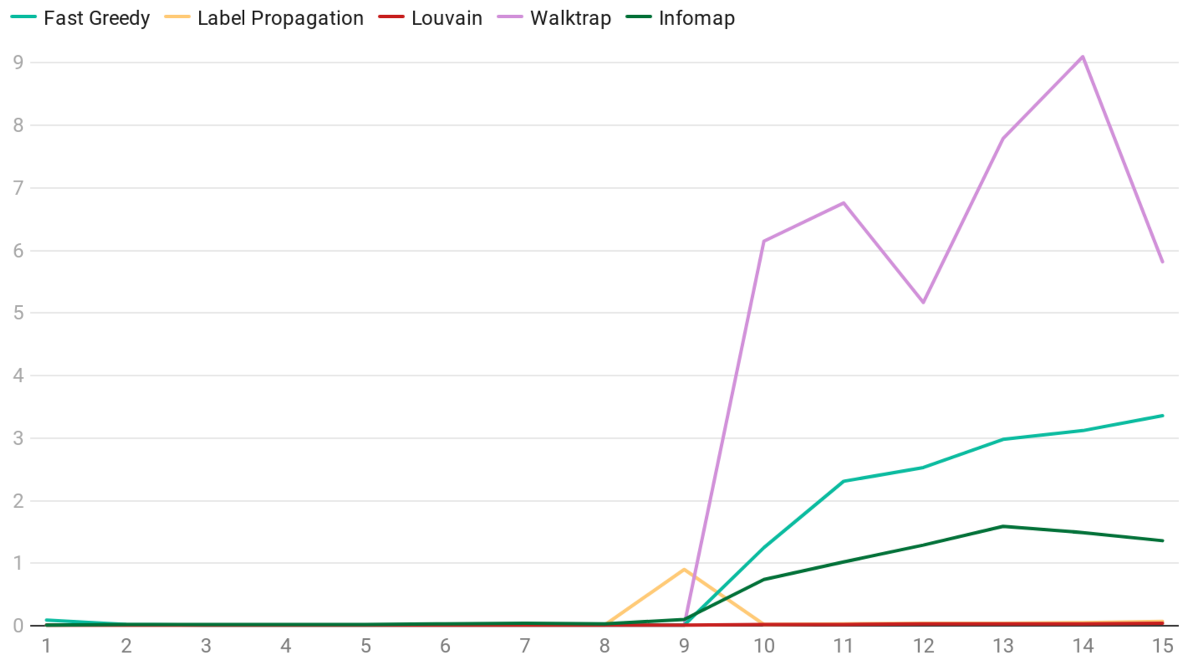


Figure 26: Execution time of each algorithm as days increase (excluding Edge Betweenness)

4.4.2 **Examination of detected communities in different use cases**

The scope of the second experiment is to demonstrate the differences between communities that are detected in the diverse use cases of EOPEN. The algorithm that is applied is Louvain and the dataset is tweets from all the collections that have been crawled between 13/12/2019 and 19/12/2019 (a week). This time frame is selected because no events have been detected in any of the collections (by the Event Detection methodology of Section 3), thus the outcome is not affected by an incident. The complete results can be found in Table 14.

Figure 27 shows the number of pairs (Twitter accounts where one mentions the other) that have been found for each collection in the period of one week. English tweets about floods and food security are the most collected in this time frame, leading to the most pairs. Subsequently, in Figure 28 we can see that the most communities are formed in these cases, but it has to be noticed that English tweets about food security form relatively less communities than the tweets about floods. On the other hand, they include the largest community (Figure 29). This could mean that in the domain of food security few but large communities are formed, while flood-related communities are more but smaller.

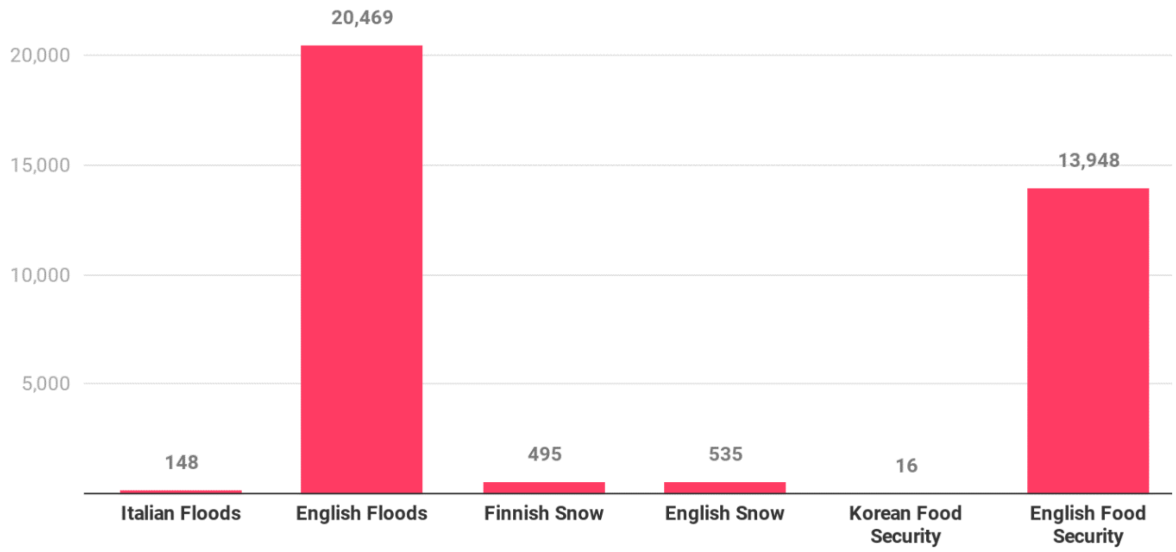


Figure 27: Number of pairs found per each collection

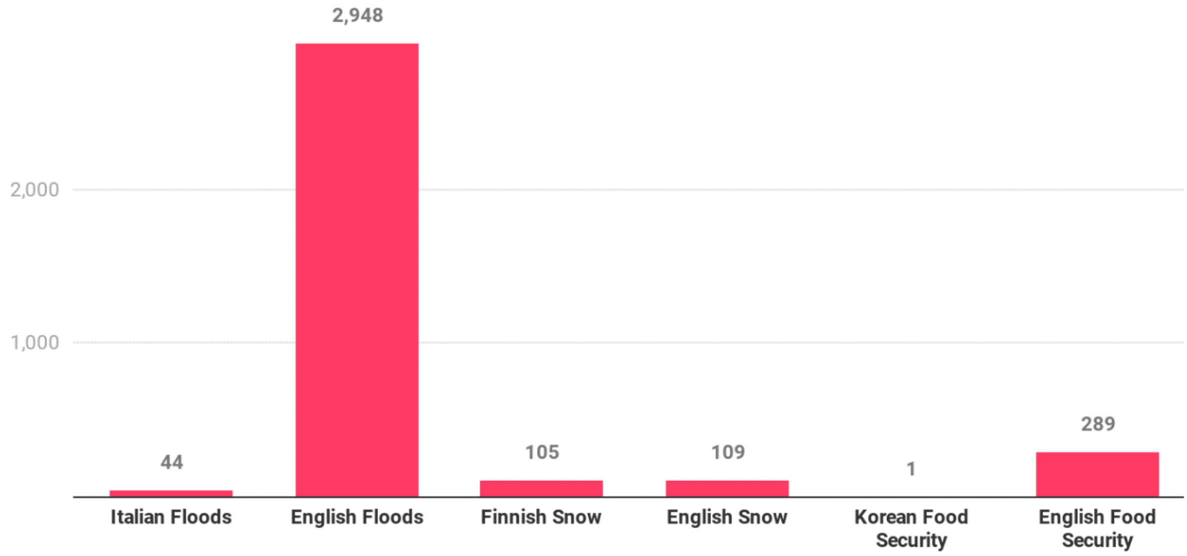


Figure 28: Number of communities detected per each collection

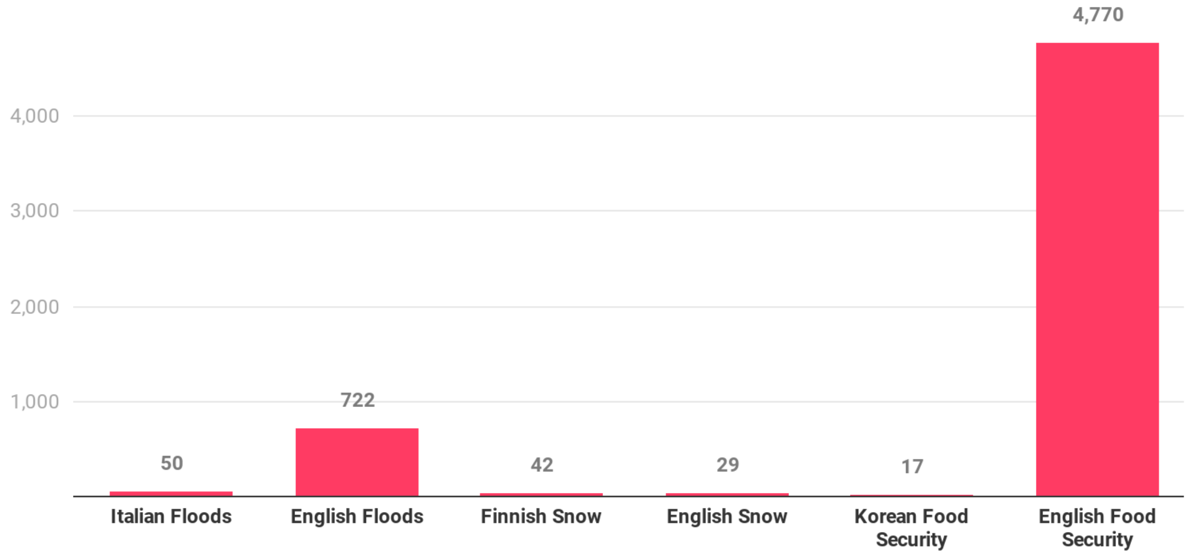


Figure 29: Maximum community size detected per each collection

Finally, the modularity of the detected communities in each collection (Figure 30) provides the most useful insights. Communities in weather-related collections are very dense, indicating that Twitter users tend to mention other users in cases of flooding/snow incidents probably to inform and alert. However, that is not the case in the food security domain, where the emergency element scarcely exists and users mostly get informed rather than influence each other. The Korean tweets collected in a week are very few, so the modularity cannot be calculated.

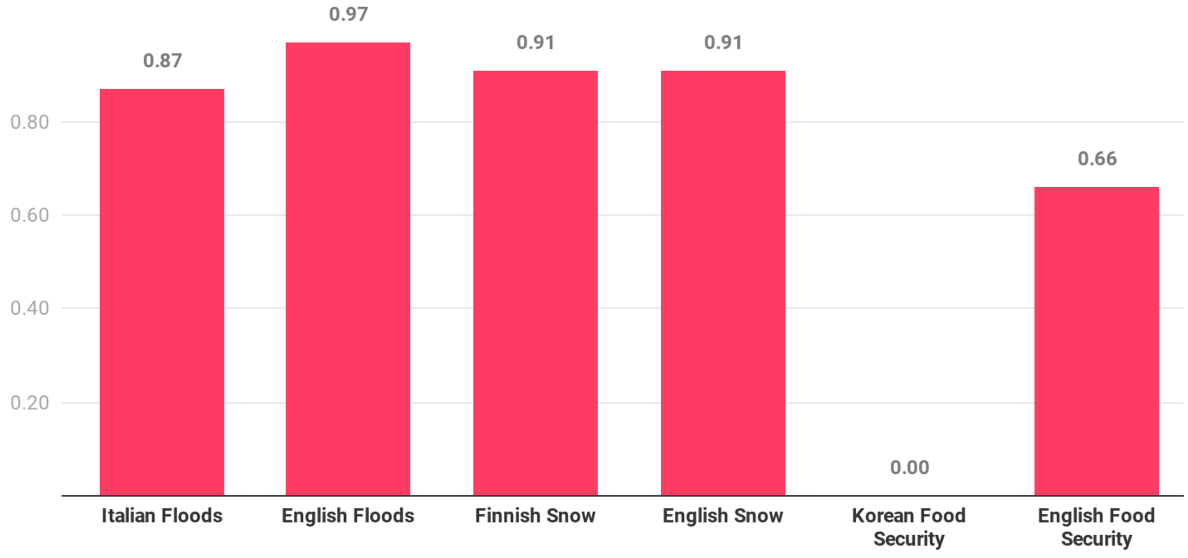


Figure 30: Modularity of communities detected per each collection

4.4.3 Investigation of identified key-players during an event

The third and final experiment is to examine what types of key-players are detected by our methodology when an incident is happening. The selected dataset is Italian tweets that were posted on 10/09/2017, when the Livorno floods occurred.

The top ten identified key-players are listed in Table 8. It is very common in Twitter for public figures (#1, #2, #6, #10) to be influential and this remains the case also during an event. In addition, news accounts (#3, #5, #9) and crisis-related organisations (#7, #8) appear to be mentioned by many users either for providing or requiring information. Finally, it is possible for individuals (#4), e.g. citizens, to obtain a more authoritative role amid an incident, because they share valuable information that might not be yet publicly known. We highlight that a citizen (#4) can be influential and very central during a flood event, even more than a central emergency management Twitter account (#7).

It should be reminded here that the actual implementation involves pseudonymized Twitter usernames to respect their privacy; real account names have been displayed solely in the frame of this experiment.

Table 8: Top key-players on 10/09/2017 (Livorno floods)

#	Twitter user name	Twitter user screen name	Description
1	Benji & Fede	@BenjieFede	Music band
2	Massimiliano Allegri	@OfficialAllegri	Italian football manager
3	Alessandro Barabino	@ale_barabino	Journalist
4	**Act@rus**	**@actarus1070**	**Citizen**
5	Allarme24	@allarme24	News
6	Eugenio Cardi	@EugenioCardi	Writer
7	**Vigili del Fuoco**	**@emergenzavvf**	**Central emergency management**
8	Io non rischio	@iononrischio	Communication campaign
9	Sky tg24	@SkyTG24	News
10	Enrico Letta	@EnricoLetta	Former prime minister of Italy

5 CONCEPT EXTRACTION

Concept extraction is part of task T4.2 "Concept and Event Detection in non EO data", where more focus has been given on the subtask of event detection. Nevertheless, there has been some progress regarding to concept detection and it is described in this section.

In general, concept detection involves extracting high-level content (i.e. concepts) from textual or visual low-level information in order to be able to retrieve relevant content and to mark multimodal content as relevant or not to set of categories (i.e. classes). Within EOPEN, concept detection is used for extracting concepts from visual information from images coming from social media data (i.e. data retrieved from Twitter) and Earth Observation (EO) imagery. However, given that the images (i.e. EO and non-EO data) considered in these cases differ significantly, the set of concepts used differs as well.

In the following, we provide a short description of the usual procedure followed in concept detection task. Concept detection is a two-step process that involves first the construction of a model by using a training set of the target category; and secondly the application of the model for classifying unseen data (testing set). The algorithm that implements the classification is the classifier, i.e. a function that maps one observation to a pre-defined class.

5.1 Visual concepts from non-EO images

In this deliverable a new version of the concept detection framework used for extracting high-level concepts from non-EO data (i.e. Twitter images) is proposed. In the previous framework, which was described in D4.1, a 22-layer GoogleNet network was trained on 5055 ImageNet concepts (Pittaras et al., 2017) and then fine-tuning was performed by replacing the classification layer with dimensionality 5055 with a classification layer with dimension equal to 345 which equals to the number of SIN TRECVID concepts[5].

Although the initial method achieved very good performance, when validated with the TRECVID dataset, it also introduced several issues including significant higher inference time, and a fixed overhead of some seconds due to calling an external executable that resulted in making it inefficient for repeated use, which is the case for EOPEN since new tweets are collected every second.

The new version involves the use of a newly proposed DCNN model, the EfficientNet (Tan & Le, 2019) and specifically the use of EfficientNetB1 and EfficientNetB3. EfficientNets are a family of neural network architectures released by Google in 2019 that were using an optimization procedure that aimed at maximizing the accuracy for a given computational cost. EfficientNets are recommended for classification tasks, since they beat many other networks (e.g. DenseNet (Huang et al., 2017), Inception (Szegedy et al., 2015), ResNet (He et al., 2016)) on the ImageNet benchmark, while running significantly faster (see Figure 31). Specifically, the new version of concept detection involves averaging the concept probabilities from EfficientNet-B1 and EfficientNet-B3 DCNN models (Apostolidis et al., 2020). This approach achieves similar performance to the initial approach but it is much

[5] http://www-nlpir.nist.gov/projects/tv2012/tv11.sin.500.concepts_ann_v2.xls

faster, which is critical due to the rate of incoming images. Extensive analysis and evaluation results can be found in (Apostolidis et al., 2020).

The use of concept detection in EOPEN is twofold. It is used as one of the modalities used in similarity retrieval (described in "D4.3 - Multimodal fusion for information retrieval") in order to retrieve content similar to a given query (tweet) and also it is presented in tweet visualization in order to provide some semantics to the image. The latter is achieved by assigning to each image concepts taken from the TRECVID concept pool that describe its content. Figure 32 depicts part of the 345 TRECVID concepts and the highlighted ones are those that are most closely connected to the EOPEN Use cases flood and snow.

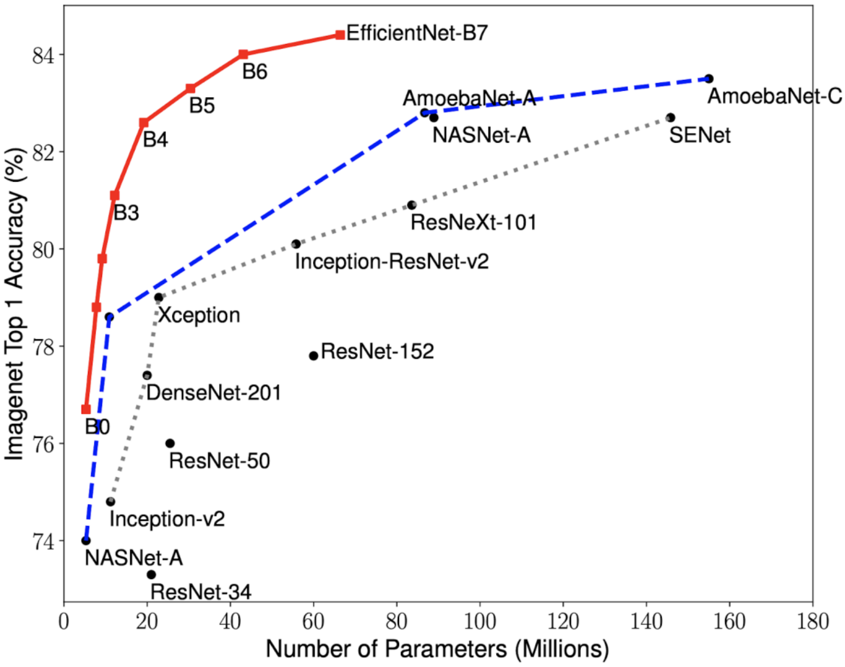
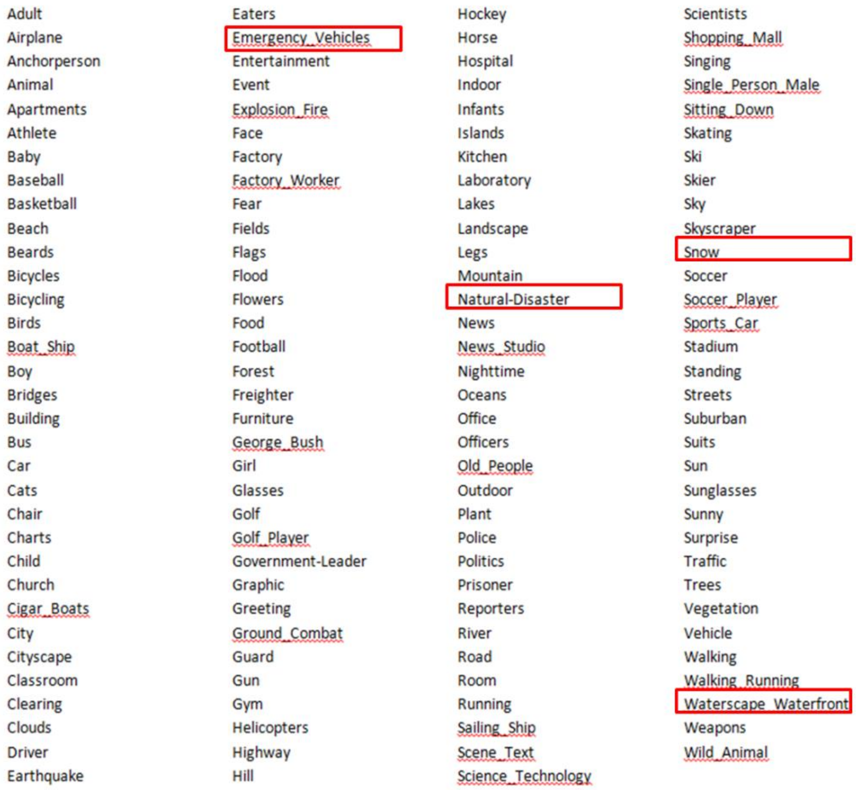


Figure 31: Comparing accuracy of EfficientNets vs other networks (from Google AI Blog)



Adult
Airplane
Anchorperson
Animal
Apartments
Athlete
Baby
Baseball
Basketball
Beach
Beards
Bicycles
Bicycling
Birds
Boat_Ship
Boy
Bridges
Building
Bus
Car
Cats
Chair
Charts
Child
Church
Cigar_Boats
City
Cityscape
Classroom
Clearing
Clouds
Driver
Earthquake

Eaters
Emergency_Vehicles
Entertainment
Event
Explosion_Fire
Face
Factory
Factory_Worker
Fear
Fields
Flags
Flood
Flowers
Food
Football
Forest
Freighter
Furniture
George_Bush
Girl
Glasses
Golf
Golf_Player
Government-Leader
Graphic
Greeting
Ground_Combat
Guard
Gun
Gym
Helicopters
Highway
Hill

Hockey
Horse
Hospital
Indoor
Infants
Islands
Kitchen
Laboratory
Lakes
Landscape
Legs
Mountain
Natural-Disaster
News
News_Studio
Nighttime
Oceans
Office
Officers
Old_People
Outdoor
Plant
Police
Politics
Prisoner
Reporters
River
Road
Room
Running
Sailing_Ship
Scene_Text
Science_Technology

Scientists
Shopping_Mall
Singing
Single_Person_Male
Sitting_Down
Skating
Ski
Skier
Sky
Skyscraper
Snow
Soccer
Soccer_Player
Sports_Car
Stadium
Standing
Streets
Suburban
Suits
Sun
Sunglasses
Sunny
Surprise
Traffic
Trees
Vegetation
Vehicle
Walking
Walking_Running
Waterscape_Waterfront
Weapons
Wild_Animal

Figure 32: Examples of TRECVID SIN concepts related to EOPEN Use cases

5.2 Concepts from EO imagery

The aim of this module is to generate a model that extracts high-level concepts from EO images that characterize them based on a set of predefined classes. The extraction of features from satellite patches was evaluated using two different approaches. The first involved three pre-trained ImageNet DCNNs: VGG-19 (Simonyan & Zisserman, 2014), ResNet-50 (He et al., 2016), and Inception-ResNet-v2 (Szegedy et al., 2017), where the feature vectors from the last layers were extracted. In the second approach, a custom DNN with a structure that resembles VGG was trained (Figure 33). It contains blocks of convolutional layers with 3x3 filters followed by a max pooling layer. Dropout is used to prevent the model from overfitting. This pattern is repeated with a doubling in the number of filters with each block added. The model will produce a 7-element vector with a prediction between 0 and 1 for each output class. Since it is a multi-label problem, the sigmoid activation function was used in the output layer with the binary cross entropy loss function. For training and testing purposes, we experimented with both 3 channel images (as done with the pre-trained networks) and also images that consisted of 5 bands of Sentinel-2 images. The dataset used in both cases was the BigEarthNet[6] dataset, where each image patch is annotated by the multiple land-cover classes (i.e., multi-labels) that were extracted from the CORINE Land Cover inventory of the year 2018 (CLC 2018). Seven major classes were formed (rice, urban rock, vineyards, forest, water, snow), covering several of the PUCs

[6] http://bigearth.net/

subjects. The 5-channel custom DNN presented the best results at mean average precision metric.

More details can be found in D4.3 (“Multimodal fusion for information retrieval”).

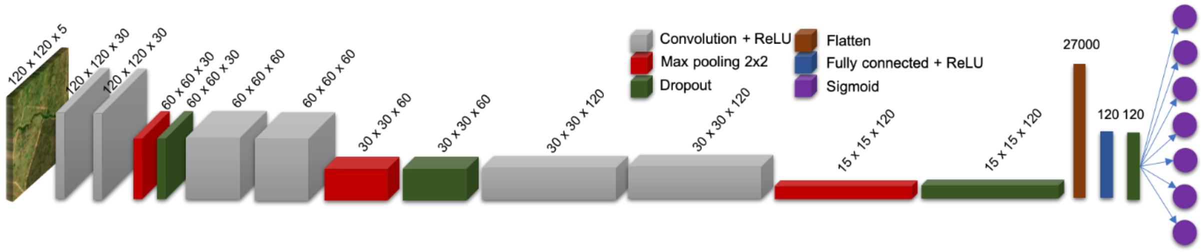


Figure 33: Architecture of the DCNN framework

6 CONCLUSIONS

This deliverable focused on the tasks of detecting changes in EO data, detecting events and communities in social media data, and extracting concepts from both types of data.

Section 2 included a presentation of a water delineation methodology that combines an artificial neural network with backscatter Sentinel-1 values paired with elevation information that reduces the misclassified as water areas due to high slope. The effectiveness of the approach was demonstrated when compared to the same DCNN model without the usage of the elevation information and against a baseline thresholding technique. Then we demonstrated a couple of pure change detection approaches that involved subtracting consecutive images of a time series of Sentinel-2 images. At the task of the confirmation of a flood incident within a time series, using MNDWI outperformed the DCNN approach with the RGB images. Also, we demonstrated change (i.e. flooded) maps that are the product of the subtraction of the images.

Moreover, Section 3 presented the proposed event detection methodology that is based on applying outlier detection on the number of daily collected tweets and is extended to discover the most frequent keywords and the mentioned location, so as to gain further knowledge on the detected events. The method was evaluated on detecting real floods in Italy for years 2017 and 2018 and achieved high recall and high precision depending on the parameters. A qualitative evaluation followed, presenting the events that have been detected for food security in Korea and snow coverage in Finland.

Next, Section 4 described the methodologies for discovering user communities based on the "mentioning" relationship between Twitter accounts and identifying the key-players of these communities. A comparison of the most popular community detection approaches showed the superiority of the Louvain algorithm, both in terms of performance and execution time. A comparison of the selected algorithm running on all EOPEN use cases showed that a larger number of denser communities tend to be formed for weather-related topics, such as floods and snow. Moreover, the key-players of the communities were detected during a flooding incident, proving that individuals (i.e. citizens) can evolve to influential users when they share valuable information about an occurring event.

Finally, Section 5 concerned concept detection, applied in EO and non-EO data. In both cases we followed a DCNN-based approach. However, given that the images and the concepts recognized differ significantly, we used different networks. Specifically, for non-EO data we evaluated a new DCNN network called EfficientNet which performs similarly in terms of accuracy with the initial approach but much better in terms of time. Regarding concept extraction in EO data, we provided a brief presentation of the dataset, the selected classes and the evaluation results.

7 REFERENCES

Abdelhaq, H., Sengstock, C. and Gertz, M., 2013. "Eventweet: Online localized event detection from twitter", Proceedings of the VLDB Endowment, 6(12), pp.1326-1329.

Ahajjam, S., El Haddad, M. and Badir, H., 2018. "A new scalable leader-community detection approach for community detection in social networks", Social Networks, 54, pp.41-49.

Alsaedi, N., Burnap, P. and Rana, O., 2017. "Can we predict a riot? Disruptive event detection using Twitter", ACM Transactions on Internet Technology (TOIT), 17(2), pp.1-26.

Apostolidis, K. Gkalelis, N., Apostolidis, E., Mezaris, V., et al. 2020. "Enhancing and Re-Purposing TV Content for Trans-Vector Engagement", Enhancing and Re-Purposing TV Content for Trans-Vector Engagement (ReTV), H2020 Research and Innovation Action - Grant Agreement No. 780656, pp. 1-54.

Atefeh, F. and Khreich, W., 2015. "A survey of techniques for event detection in twitter", Computational Intelligence, 31(1), pp.132-164.

Bedi, P. and Sharma, C., 2016. "Community detection in social networks", Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery, 6(3), pp.115-135.

Bischke, B., Helber, P., Zhao, Z., Bruijn, J. de, and Borth, D., 2018. "The Multimedia Satellite Task at MediaEval 2018", In: MediaEval.

Blondel, V.D., Guillaume, J.L., Lambiotte, R. and Lefebvre, E., 2008. "Fast unfolding of communities in large networks", Journal of statistical mechanics: theory and experiment, 2008(10), p.P10008.

Bohlin, L., Edler, D., Lancichinetti, A. and Rosvall, M., 2014. "Community detection and visualization of networks with the map equation framework. In Measuring scholarly impact (pp. 3-34)", Springer, Cham.

Bovolo, F., Bruzzone, L., and Marconcini, M., 2008. "A novel approach to unsupervised change detection based on a semi supervised SVM and a similarity measure", In: IEEE Transactions on Geoscience and Remote Sensing46.7, pp. 2070–2082.

Chaouch, N., Temimi, M., Hagen, S., Weishampel, J., Medeiros, S., and Khanbilvardi, R., 2012. "A synergetic use of satellite imagery from SAR and optical sensors to improve coastal flood mapping in the Gulf of Mexico", Hydrological processes, 26(11), 1617-1628

Cheng, T. and Wicks, T., 2014. "Event detection using Twitter: a spatio-temporal approach", PloS one, 9(6), p.e97807.

Clauset, A., Newman, M.E. and Moore, C., 2004. "Finding community structure in very large networks", Physical review E, 70(6), p.066111.

Clement, M., Kilsby, C., and Moore, P.,2018. "Multi-temporal synthetic aperture radar flood mapping using change detection", In: Journal of Flood Risk Management 11.2, pp. 152–168.

Cordeiro, M., 2012, January. "Twitter event detection: combining wavelet analysis and topic inference summarization", In Doctoral symposium on informatics engineering (pp. 11-16).

Gialampoukidis, I., Kalpakis, G., Tsikrika, T., Papadopoulos, S., Vrochidis, S. and Kompatsiaris, I., 2017, June. "Detection of terrorism-related twitter communities using centrality scores", In Proceedings of the 2nd International Workshop on Multimedia Forensics and Security (pp. 21-25).

Gialampoukidis, I., Kalpakis, G., Tsikrika, T., Vrochidis, S. and Kompatsiaris, I., 2016, August. “Key player identification in terrorism-related social media networks using centrality measures”, In 2016 European Intelligence and Security Informatics Conference (EISIC) (pp. 112-115). IEEE.

Hasan, M., Orgun, M.A. and Schwitter, R., 2018. “A survey on real-time event detection from the twitter data stream”, Journal of Information Science, 44(4), pp.443-463.

He, K., Li, Y., Soundarajan, S. and Hopcroft, J.E., 2018. “Hidden community detection in social networks”, Information Sciences, 425, pp.92-106.

He, K., Zhang, X., Ren, S., and Sun, J., 2016. “Deep residual learning for image recognition”, In: Proceedings of the IEEE conference on computer vision and pattern recognition, pp. 770–778.

Hua, T., Chen, F., Zhao, L., Lu, C.T. and Ramakrishnan, N., 2016. “Automatic targeted-domain spatiotemporal event detection in twitter”, GeoInformatica, 20(4), pp.765-795.

Huang, C., Song, K., Kim, S., Townshend, J. R., Davis, P., Masek, J. G., and Goward, S. N., 2008. “Use of a dark object concept and support vector machines to automate forest cover change analysis”, In: Remote Sensing of Environment 112.3, pp. 970–985.

Huang, G., Liu, Z., Van Der Maaten, L., & Weinberger, K. Q. 2017. “Densely connected convolutional networks”, In Proceedings of the IEEE conference on computer vision and pattern recognition (pp. 4700-4708).

Javed, M.A., Younis, M.S., Latif, S., Qadir, J. and Baig, A., 2018. “Community detection in networks: A multidisciplinary review”, Journal of Network and Computer Applications, 108, pp.87-111.

Johnson, J. M. and Khoshgoftaar, T. M., 2019. “Survey on deep learning with class imbalance”, Journal of Big Data, 6(1):27.

Lee, J.S., 1981. “Refined filtering of image noise using local statistics”, Computer graphics and image processing, 15(4), pp.380-389.

Lee, J.S., Wen, J.H., Ainsworth, T.L., Chen, K.S. and Chen, A.J., 2008. “Improved sigma filter for speckle filtering of SAR imagery”, IEEE Transactions on Geoscience and Remote Sensing, 47(1), pp.202-213.

Long, S.A., Fatoyinbo, T.E. and Policelli, F., 2011. “Simplified change detection method for flood extent mapping using SAR”, AGUFM, 2011, pp.H14C-05.

Lu, D., Mausel, P., Batistella, M., and Moran, E., 2005. “Land-cover binary change detection methods for use in the moist tropical region of the Amazon: a comparative study”, In: International Journal of Remote Sensing 26.1,pp. 101–114.

Manjusree, P., Kumar, L. P., Bhatt, C. M., Rao, G. S., and Bhanumurthy, V., 2012. “Optimization of threshold ranges for rapid flood inundation mapping by evaluating backscatter profiles of high incidence angle sar images”, International Journal of Disaster Risk Science, 3(2):113–122.

McMinn, A.J., Moshfeghi, Y. and Jose, J.M., 2013, October. “Building a large-scale corpus for evaluating event detection on twitter”, In Proceedings of the 22nd ACM international conference on Information & Knowledge Management (pp. 409-418).

Michail, E., Moumtzidou, A., Gialampoukidis, I., Avgerinakis, K., Scarpino, M. G., Vrochidis, S., Vingione, G., Kompatsiaris, I., Labbassi, K., Menenti, M., Elghandour, F.-E., 2018. “Testing a Flood Mask Correction Method Of Optical Satellite Imagery Over Irrigated Agricultural Areas”, 2nd Mapping Water Bodies from Space Conference (MWBS2018), ESA-ESRIN Frascati, Italy.

Newman, M.E. and Girvan, M., 2004. “Finding and evaluating community structure in networks”, Physical review E, 69(2), p.026113.

Ozdikis, O., Senkul, P. and Oguztuzun, H., 2012, August. “Semantic expansion of tweet contents for enhanced event detection in twitter”, In 2012 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining (pp. 20-24). IEEE.

Pal, M., 2005. “Random forest classifier for remote sensing classification”, In: International Journal of Remote Sensing 26.1, pp. 217–222.

Pittaras, N., Markatopoulou, F., Mezaris, V., and Patras, I. 2017. “Comparison of Fine-tuning and Extension Strategies for Deep Convolutional Neural Networks”, Proc. 23rd Int. Conf. on MultiMedia Modeling (MMM'17), Reykjavik, Iceland, Springer LNCS vol. 10132, pp. 102-114.

Pons, P. and Latapy, M., 2005, October. “Computing communities in large networks using random walks”, In International symposium on computer and information sciences (pp. 284-293). Springer, Berlin, Heidelberg.

Raghavan, U.N., Albert, R. and Kumara, S., 2007. “Near linear time algorithm to detect community structures in large-scale networks”, Physical review E, 76(3), p.036106.

Rosvall, M., Axelsson, D. and Bergstrom, C.T., 2009. “The map equation”, The European Physical Journal Special Topics, 178(1), pp.13-23.

Sakaki, T., Okazaki, M. and Matsuo, Y., 2010, April. “Earthquake shakes Twitter users: real-time event detection by social sensors”, In Proceedings of the 19th international conference on World wide web (pp. 851-860).

Santoyo, S., 2017. “A Brief Overview of Outlier Detection Techniques”, Towards Data Science, September, 12.

Shiffler, R.E., 1988. “Maximum Z scores and outliers”, The American Statistician, 42(1), pp.79-80.

Simonyan, K. and Zisserman, A., 2014. “Very Deep Convolutional Networks for Large-Scale Image Recognition”, In: CoRR abs/1409.1556. arXiv: 1409.1556.

Smith, A., 2010. “Image segmentation scale parameter optimization and land cover classification using the Random Forest algorithm”, In: Journal of Spatial Science 55.1, pp. 69–79.

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).

Szegedy, C., Ioffe, S., Vanhoucke, V., & Alemi, A. A., 2017, February. “Inception-v4, inception-resnet and the impact of residual connections on learning”, in Thirty-first AAAI conference on artificial intelligence.

Tan, M., and Le, Q. V., 2019. “Efficientnet: Rethinking model scaling for convolutional neural networks”, Proceedings of the36thInternational Conference on MachineLearning, Long Beach, California, PMLR 97, 2019.

Townsend, P. A. and Walsh, S. J., 1998. “Modeling floodplain inundation using an integrated gis with radar and optical remote sensing”, Geomorphology, 21(3):295 – 312. Application of remote sensing and GIS in geomorphology.

Veldt, N., Gleich, D.F. and Wirth, A., 2018, April. “A correlation clustering framework for community detection”, In Proceedings of the 2018 World Wide Web Conference (pp. 439-448).

Wang, F., Zhang, B., Chai, S. and Xia, Y., 2018. “An extreme learning machine-based community detection algorithm in complex networks”, Complexity, 2018.

Whang, J.J., Gleich, D.F. and Dhillon, I.S., 2016. “Overlapping community detection using neighborhood-inflated seed expansion”, IEEE Transactions on Knowledge and Data Engineering, 28(5), pp.1272-1284.

Xu, H., 2006. “Modification of normalised difference water index (NDWI) to enhance open water features in remotely sensed imagery”, In: International journal of remote sensing 27.14, pp. 3025–3033.

Yang, Z., Algesheimer, R. and Tessone, C.J., 2016. “A comparative analysis of community detection algorithms on artificial networks”, Scientific reports, 6, p.30750.

Zhou, X. and Chen, L., 2014. “Event detection over twitter social media streams”, The VLDB journal, 23(3), pp.381-400.

APPENDIX

Table 9: Number of communities detected by each algorithm as days increase

Days	Edge Betweenness	Fast Greedy	Label Propagation	Louvain	Walktrap	Infomap
1	28	28	31	28	28	30
2	48	48	53	48	48	51
3	58	58	65	58	57	62
4	69	69	74	69	67	72
5	74	74	78	74	72	77
6	76	76	83	76	76	82
7	76	76	88	76	82	87
8	97	95	112	94	104	109
9	119	118	136	118	132	136
10	212	212	281	211	333	300
11	273	276	387	267	374	410
12	297	309	430	296	386	461
13	316	324	454	312	465	491
14	340	338	477	328	438	521
15	347	343	490	331	450	533

Table 10: Maximum community size detected by each algorithm as days increase

Days	Edge Betweenness	Fast Greedy	Label Propagation	Louvain	Walktrap	Infomap
1	42	42	42	42	42	42
2	44	44	46	44	44	44
3	47	47	47	42	47	47
4	48	48	48	48	48	48
5	48	48	48	48	48	48
6	48	48	49	48	48	48
7	48	48	48	48	48	48
8	48	48	49	48	48	48
9	48	48	48	48	48	48
10	2456	2456	2462	2456	3191	3191
11	2897	2884	2902	2884	3821	3821
12	3010	2996	3007	2996	3999	3999
13	3108	3099	3112	3094	4051	4051
14	3142	3134	3146	3129	4114	4114
15	3184	3178	3186	3173	4144	4144

Table 11: Modularity of communities detected by each algorithm as days increase

Days	Edge Betweenness	Fast Greedy	Label Propagation	Louvain	Walktrap	Infomap
1	0.85	0.85	0.84	0.85	0.85	0.84
2	0.9	0.9	0.88	0.9	0.9	0.89
3	0.91	0.91	0.89	0.91	0.91	0.91
4	0.92	0.92	0.91	0.92	0.92	0.92
5	0.93	0.93	0.92	0.93	0.93	0.92
6	0.92	0.92	0.9	0.92	0.92	0.91
7	0.92	0.92	0.89	0.92	0.91	0.9
8	0.91	0.92	0.88	0.92	0.9	0.9
9	0.92	0.92	0.9	0.92	0.91	0.9
10	0.72	0.72	0.7	0.72	0.54	0.7
11	0.76	0.76	0.73	0.76	0.61	0.73
12	0.78	0.78	0.74	0.78	0.65	0.74
13	0.78	0.78	0.74	0.79	0.67	0.74
14	0.78	0.79	0.74	0.79	0.68	0.74
15	0.79	0.79	0.75	0.79	0.69	0.74

Table 12: Code length of communities detected by Infomap as days increase

Days	Infomap
1	3.98
2	4
3	4.17
4	4.17
5	4.16
6	4.23
7	4.33
8	4.33
9	4.3
10	6.79
11	6.98
12	7.07
13	7.11
14	7.16
15	7.19

Table 13: Execution time of each algorithm as days increase

Days	Edge Betweenness	Fast Greedy	Label Propagation	Louvain	Walktrap	Infomap
1	0.02	0.09	0.01	0.01	0.01	0.01
2	0.03	0.02	0.01	0.01	0.01	0.02
3	0.06	0.01	0.01	0.01	0.01	0.02
4	0.08	0.01	0.01	0.01	0.01	0.02
5	0.08	0.01	0.01	0.01	0.01	0.02
6	0.09	0.01	0.01	0.01	0.01	0.03
7	0.11	0.01	0.01	0.01	0.01	0.04
8	0.16	0.01	0.01	0.01	0.01	0.03
9	0.23	0.01	0.9	0.01	0.01	0.1
10	343.2	1.25	0.02	0.02	6.15	0.74
11	786	2.31	0.03	0.02	6.76	1.02
12	1051.8	2.53	0.04	0.03	5.17	1.29
13	1240.2	2.98	0.04	0.03	7.79	1.59
14	1426.2	3.12	0.05	0.03	9.1	1.49
15	1558.8	3.36	0.07	0.04	5.82	1.36

Table 14: Metrics for the Louvain algorithm on different collections

Use Case	Number of pairs	Number of communities	Max community size	Modularity
Italian Floods	148	44	50	0.87
English Floods	20469	2948	722	0.97
Finnish Snow	495	105	42	0.91
English Snow	535	109	29	0.91
Korean Food Security	16	1	17	0
English Food Security	13948	289	4770	0.66