INSIGHT
FP7-318225

D5.1:

Report on end-user requirements, test data, and on prototype definitions

TU Dortmund University

August 30, 2013

Status: Finished
Scheduled Delivery Date: 31/08/2013

Executive summary

The report at-hand presents (1) an introduction to spatio-temporal data types and analysis methods. (2) Batch data samples of the data streams described in Deliverable 2.1 are provided and described. (3) Learning tasks are derived from the two use cases (nation-wide flooding scenario and city-level scenario). (4) Data quality issues are pointed out. (5) First analyses and their results are shown. (6) Finally, the prototypes for both use cases are introduced.

Document Information

Contract Number	FP7-318225	**Acronym**	Insight
Full Title	Intelligent Synthesis and Real-Time Response using Massive Streaming of Heterogeneous Data		
Project URL	http://www.insight-ict.eu/		
EU Project Officer	Mr. Treinen		

Deliverable	**Number D5.1**	Report on end-user requirements, test data, and on prototype definitions		
Work Package	**Number**	WP5		
Date of Delivery	31/08/2013	**Actual**	30/08/2013	
Status	Finished			
Nature	Report			
Distribution Type	Public			
Authoring Partner	TU Dortmund University			
QA Partner	BBK, UoA			
Contact Person	Thomas Liebig	thomas.liebig@cs.tu-dortmund.de		
	Katharina Morik	katharina.morik@cs.uni-dortmund.de		
	Phone		**Fax**	

Project Information

This document is part of a research project funded by the IST Programme of the Commission of the European Communities as project number FP7-318225. The Beneficiaries in this project are:

No.	Name	Short Name	Country
1	National and Kapodistrian University of Athens	UoA	Greece
2	IBM Ireland Product Distribution Limited	IBM	Ireland
3	Fraunhofer-Gesellschaft Zur Foerderung Der Angewandten Forschung E.V.	Fraunhofer	Germany
4	Technische Universitaet Dortmund	TUD	Germany
5	Technion - Israel Institute of Technology	Technion	Israel
6	Dublin City Council	DCC	Ireland
7	Bundesamt für Bevölkerungsschutz und Katastrophenhilfe	BBK	Germany

Table of Contents

Index of Figures

1 Introduction

The INSIGHT application scenarios demonstrate the benefits of Big Data Analytics for public safety in the area of civil protection. As a real-world test-bed, we have chosen two complementing and challenging scenarios of high public interest: traffic and flooding monitoring in cities, here the City of Dublin, and monitoring of nation-wide disasters – here: flooding in Germany. For these scenarios, we have elicited tasks and requirements. Requirements of the end-users (public authorities in charge of civil protection, compare D6.1) have a strong impact on the architecture (D2.1) and the analysis methods (D5.1 at-hand). The real-time analysis of heterogeneous data streams poses new challenges on existing methods. Whereas existing preprocessing and analysis methods could use multiple scans, real-time analysis may not look twice and has to perform its tasks in a single-scan. Thus, besides off-line learning from batch data, analysis and prediction methods which are capable of working on data streams are demanded.

Methods will be developed for both use-case scenarios and are eventually applicable to different data sources (focus of current studies) in order to profit from synergies between the scenarios. We plan to integrate existing streaming platforms (compare D2.1 for an introduction to infosphere streams, streams-framework and storm) and already have made some progress in the integrating of event detection and streaming data analysis. A recent study of Technion and TUDo demonstrates that the streams-framework [BB12b], which already comprises multiple data analysis and mining algorithms [BB12a, Bif13], is capable of high throughput analysis [GKS+13].

In this report, we describe test data sets for the evaluation of analyses. We highlight recent analyses and work in progress to meet the requirements. This comprises the geo-coding of text messages, the visual inspection of Twitter messages, location analysis of Twitter users in Dublin, the spatial clustering of locations based on Twitter messages, analysis of traffic flow data, the imputation of traffic flow for unobserved locations and the detection of events from mobile phone usage data.

2 Spatio-Temporal Data Analysis

The common characteristic of the data streams available in INSIGHT is its spatio-temporal nature. Though not all of the data contain coordinates (e.g. some Twitter messages) they mostly contain information on locations or moving objects. In both dimensions, space and time, the data items have limited validity. For example, a message containing the weather information at a particular spatio-temporal coordinate is invalid in future or in large distance. The GPS information of a moving object (e.g. a vehicular position) loses validity immediately in future and in its close spatial neighbourhood. Thus, the models developed by INSIGHT have to incorporate latest data samples and need to perform in real-time. This does not exclude learning from historic data samples in order to compare current situations with the past and project it into future. The required architecture for these analyses is developed in D2.1. This deliverable focuses on the elicitation of end-user requirements and identification of analysis tasks for WP5 (see next sections) and we give a brief overview on possible spatio-temporal analysis tasks. For a comprehensive introduction to spatio-temporal data mining we

refer to the book [GP08], which results from the GeoPKDD project funded by the European Commission under the Sixth Framework Programme, IST-6FP-014915.

Spatio-temporal data comes in a variety of forms and representations, depending on the domain, the observed phenomenon, and the observation method. In principle, there are three types of spatio-temporal data: spatial time series, events, and trajectories.

- A spatial time series consists of tuples $(attribute, object, time, location)$.
- An event of a particular type $event_i$ is triggered from a spatial time series under certain conditions and contains the tuples verifying these conditions $(event_i, object_n, time_n, location_n)$.
- A trajectory is a spatial time series for a particular $object_i$. It contains the location per time and is a series of tuples $(object_i, time_n, location_n)$. Every timestamp $time_n$ is contained at most once.

The types may be transformed to accommodate different analysis tasks and goals; this is focus of WP3.

2.1 Frequent Patterns

The challenge of frequent pattern mining is the identification of frequently co-occurring sets of items or more complex patterns (that describe relations among space and time). Input items can be elements of spatial time series, events, or the tuples of trajectories. Output are frequent sets of these items. A common algorithm for mining these data sets for frequent item sets is the apriori algorithm that generates candidates for frequent item sets as unions of smaller frequent item sets. A common parameter for frequent item mining is the minimum support which is a threshold to distinguish among frequent and un-frequent sets of items.

As the coordinates in trajectories may be too fine-granular to identify frequently co-visited places, the T-pattern algorithm [GNPP07] extracts spatial regions from the trajectories which are frequently visited and returns frequent visit patterns among them.

2.2 Classification, Regression, Prediction

For spatio-temporal data, the group of regression and prediction tasks originates in geo-statistics. The idea is to formulate a model of the data in order to impute unknown values using this model. While classification works on discrete values, regression is for continuous ones. Prediction imputes a label (class membership, target value) using a decision or regression function that was learned from complete instances. Classification, Regression and Prediction are applicable to all three data types: spatio-temporal time series, events and trajectories.

A characteristic of spatio-temporal data (if constrained by space and time e.g. traffic flow in a street network) is the autocorrelation among the values, whereas close values are more related than distant ones [Tob70]. A commonly used regression method from geo-statistics, Kriging [Kri51], models the autocorrelation with variograms that describe the correlation among spatio-temporal values at different positions as function of their distance.

Geographically weighted regression [FBC02] is another commonly used method which models an unknown value as linear combination of observed values, the weights of the observed

values vary for different locations. Spatial k-nearest neighbour algorithm [MHK+08] imputes a data point as weighted sum of the k nearest points.

Classification of the tuples in spatio-temporal time series is important for outlier detection, possible methods are 1-class support vector machines. They describe the subspace of normal observations by a minimum enclosing ball, outliers are outside the ball. As the split decision cannot necessarily be described spherical in the attributes of the observations, they are transferred to a feature space. Instead of computing the transformation for every incoming data, an inner product in feature space is defined which can be computed directly using the observations. It maps two observations to a real number. Core vector machines compute an approximation of the minimum enclosing ball with constant space and time requirements [BC08] which contains all observations when scaled by a factor of $(1 + \epsilon)$, $\epsilon > 0$. For distributed spatio-temporal time series, recent work applied the core vector machine to outlier detection in vertically distributed data streams using core vector machines is promising [SBDM13]. Another interesting approach is that of exceptional model mining, where rare but coherent sets of exceptions indicate a event [DFK12]. This approach could possibly be extended to spatio-temporal data.

Prediction of future values in a spatio-temporal time series has to respect Tobler's law, whereas close values correlate more than distant ones [Tob70]. This autocorrelation can directly be reflected by so-called graphical models. Every observation at a location per time is assigned to a random variable. In a graphical model the conditional dependencies of the probability distributions for the random variables are denoted by edges. Recently developed method by [PLM13] uses Markov Random Fields where every random variable for a particular time slice is connected with its direct neighbours and their ancestors from previous time slice. The method becomes efficient through the regularization of the optimization step, which saves computation when the measurements remain about the same. Incorporating measurements from previous time slices, the method estimates the most likely prediction for future time slices.

2.3 Clustering and Similarity Search

Clustering focuses on the identification of groups of objects (clusters) where the elements of a group are similar and the elements of different clusters are dissimilar. For non-overlapping clusters, the result is a partitioning of the data. Clustering can be applied to all three spatio-temporal data types: events, spatio-temporal time series and trajectories.

Commonly applied spatio-temporal clustering method is the density based algorithm DBSCAN [EKSX96] that computes spatio-temporal density of the data points and extracts clusters as highly dense sets of points. DBSCAN defines similarity among points based on their spatio-temporal distance and follows Tobler's law (close objects are more related than distant ones [Tob70]). Other similarity measures e.g. among time series, among the properties of spatio-temporal events, or among trajectories can be defined. The well-known k-Means algorithm has been turned into an algorithm for streaming data, recently [FGS+13]. Another method that could be applied for cluster analysis is OPTICS [ABpKS99].

The Voronoi tessellation method [Vor08] partitions space based on a set of spatial points. Every spatial point in this set is associated with a surrounding polygon comprising all spatial locations that are not closer to any other point contained in the set.

In [ZYLG06] Zeinalipour-Yazti et al. introduced the distributed spatio-temporal similarity

search problem: given a query trajectory Q, the purpose of the proposed algorithm was to find the trajectories that follow a motion similar to Q, when each of the target trajectories is segmented across a number of distributed nodes. Two algorithms were proposed, UB-K and UBLB-K, which combine local computations of lower and upper bounds on the matching between the distributed subsequences and Q. The approach generates the desired result without pulling together all the distributed subsequences over the fundamentally expensive communication medium. The described problem finds applications in a wide array of domains, such as, e.g., cellular networks, wildlife monitoring, and video surveillance.

2.4 Geo-Coding and Map Matching

The transformation of geo-locations is subject to geo-coding and map matching. As geo-coding aims at identification of a location for a spatio-temporal event without direct reference to an identifier (e.g. a text message that mentions a street name), map matching transfers the coordinates of events or trajectories from one reference system to another one.

Map matching tasks are common for GPS trajectories which are recorded in the WGS84 [Nat00] reference system and have to be mapped to a discrete street network graph. The spatial extent of the street segments are used for distance calculations among the street network and a particular point. The algorithm in [LZZ+09] uses these distances to generate a set of closest segments for every point of a trajectory (the segment candidates). For the identification of the most likely street segment among these candidates a routing algorithm with presumptions on individual mobility is used. In result every point of the trajectory is matched to a street segment.

2.5 Spatiotemporal Burstiness Analysis

In the context of a document search engine, Lappas et alii investigated sudden high frequent queries for a particular t. "Given a term t, a burst is generally exhibited when an unusually high frequency is observed for t." While spatial and temporal burstiness have been studied individually in the past,only recently spatio-temporal term burstiness has been studied [LVGT12]. Two alternative approaches for mining spatio-temporal burstiness patterns, STComb and STLocal, are presented, providing valuable insight on spatio-temporal burstiness from different perspectives. It is shown how the mined patterns can be utilized toward an efficient document-search engine and how this engine returns documents on influential events. The efficiency of the proposed methods is demonstrated through an extensive experimental evaluation on real and synthetic datasets.

In [LVGT13], the methojd is enhanced to identifying spatio-temporal burstiness patterns in streams. Streams are spatially distributed and the system, called STEM (Spatio-TEmporal Miner), mines spatio-temporal burstiness patterns from any collection of geostamped streams, like newspapers, location-based social networks, sensor networks, urban informatics, or blogging and microblogging platforms with spatial information (e.g. Twitter). STEM implements the two alternative approaches, STComb and STLocal, on streams. Moreover, STEM implements a user-friendly interface to help end-users in exploiting the findings.

2.6 List of the Analysis Tasks

We have identified the following tasks for spatio-temporal data analysis that are relevant for the Insight applications.

1. Frequent pattern mining for finding co-occurring sets of measurements or events,
2. Kriging for describing autocorrelation within spatio-temporal series,
3. Spatial k-Nearest Neighbour for imputing a data point based on its neighbours,
4. Core or Ball vector machines for outlier detection,
5. Spatio-temporal Markov Random Fields for the prediction of future measurement values,
6. Clustering for the overview of a current state of all measurements,
7. Voronoi tessellation for structuring the space,
8. Distributed spatio-temporal search for finding similar trajectories,
9. Transformation of geo-locations for map matching,
10. Spatio-temporal local and combinatorial patterns for detecting message or query bursts.

In the next section, we characterize the use cases and indicate, where the analysis tasks are needed.

3 Use Cases, Tasks and Data

In this section, we describe the learning and prediction tasks for the use-cases together with the data sets as they will be used for the evaluation and comparison of the project's methods.

3.1 Nation-wide use case

The nation-wide application of INSIGHT is conducted in the German Joint Information and Situation Centre (GMLZ) of the Bundesamt für Bevölkerungsschutz und Katastrophenhilfe (BBK). The goals of the GMLZ are to provide information on incidents and their risks, to support responding forces with resource management, and to alarm in case of multi-national hazards. Several data channels (depicted on a large video wall) are scanned manually for warnings and alerts which are collected in daily reports. The data channels monitored so far include weather data, river stages, severe weather watch, and news channels (European Media Monitoring[1]).

First conversations with the head of the GMLZ revealed that in addition to the manual data analysis, an automatic system is desired which helps in the following.

1. The detection of incident events (e.g. flooding), and

[1] http://emm.newsexplorer.eu/NewsExplorer/home/en/latest.html

2. supporting situation understanding is required.

3. Prediction of future situations (spatio-temporal time series) and incident events is an appreciated feature for decision support in crisis management.

The INSIGHT system aims at detecting flooding situations and at analyzing additional streaming data sources (location based social networks and phone usage data) in the form of spatio-temporal time series.

The expected output of the *incident detection* are warnings, which are spatio-temporal events. Developing an ontology of these warning events is the focus of Deliverable 2.2. For each of the events, the conditions that increase its likelihood should be learned from incoming spatio-temporal time series. Traceability of the event detection decisions and transparency on which data items led to a decision are required by the BBK. The automatic event detection should detect more events than manual inspection with less detection time.

Situation understanding should enhance the knowledge on the current situation. This task is actually the prediction of a spatio-temporal time series from the input time series. In the *prediction* task the occurrence of an event and the temporal evolution of incorporated time series (incoming ones and derived ones) should be predicted to reduce the warning time in case of an incident and for support of crisis management.

The integration of the steps for the automatic event detection into a sophisticated visualization is a major requirement of the BBK. This visualization should bundle the relevant information in case of an incident and does not catch much attention otherwise.

Following quote of the BBK (already stated in D6.1) stresses the requirements on traceability, data integration and visualization.

> The display of the findings of the INSIGHT tool is one part of the information shown on the big wall display in the GMLZ (about one-third of the big wall display). Beside this information still a situation overview, incoming CESIS[2] information as well as a news ticker will be shown as well on the big wall display. During the normal monitoring and analysis phase of incoming data, the big wall display should only show one world map and one German map respectively on the Euro board[3] for the INSIGHT findings. This simplifies the traditional usage of the big wall display in the GMLZ with several maps showing different content being monitored by the staff members. In the case the INSIGHT tool detects an event, a window should immediately pop up with a symbol (the same symbols used in deNIS KM[4]) and flash at the big wall display in the situation room as well as on the individual monitors of the staff members of the GMLZ. An alarm signal should additionally be visible in the footer of the displays during such a situation with the possibility to turn it off after recognition. Further, the INSIGHT user interface should provide

[2]The Common Emergency Communication and Information System is a software system for resource management and exchange among counties of the European Union, http://ec.europa.eu/echo/policies/disaster_response/cecis_en.htm [Last accessed: 28 June, 2013]

[3]Euro Display is a manufacturer of digital billboards, http://www.eurodisplay.com [Last accessed: 28 June, 2013]

[4]deNIS is the abbreviation for the German Emergency Prevention Information System, https://www.denis.bund.de [Last accessed: 28 June, 2013]

the possibility to open and to scale windows with different content on each monitor or additional analysis tables. After recognition of an event, the sources indicating such an event should be shown. This simplifies the estimation about the reliability of the shown data; desirable would be a scale about the reliability of the data. The reliability estimation of the data is needed for upcoming decisions making. Whether e.g. only a few tweets suggest a possibility of an event or a few hundreds, which are also validated through other data sources, makes a big difference.

The focussed use case of the INSIGHT system will be a once-in-a-millennium high tide. In this scenario a large-scale evacuation of the entire flooded area and areas at risk gets initiated. Such a large-scale evacuation holds a lot of risks and uncertainties. Their automatic identification is the aim of the INSIGHT system, this should be faster than manual data inspection. The use case is of high relevance as recent events in Germany show. The data and experiences of current events will be integrated in the INSIGHT models.

3.1.1 Availability of Test Data Samples

The following data samples are available for the nation-wide flooding use case scenario: (1) data on mobile phone usage (access granted via Fraunhofer IAIS) and (2) geo-coded Twitter messages. Additionally, data for recent flooding in Germany are provided: (3) Twitter messages including 'hochwasser' and (4) a list of relevant events from the BBK.

Next chart gives an overview on the data availability for 2013 June and July, see Figure 1.

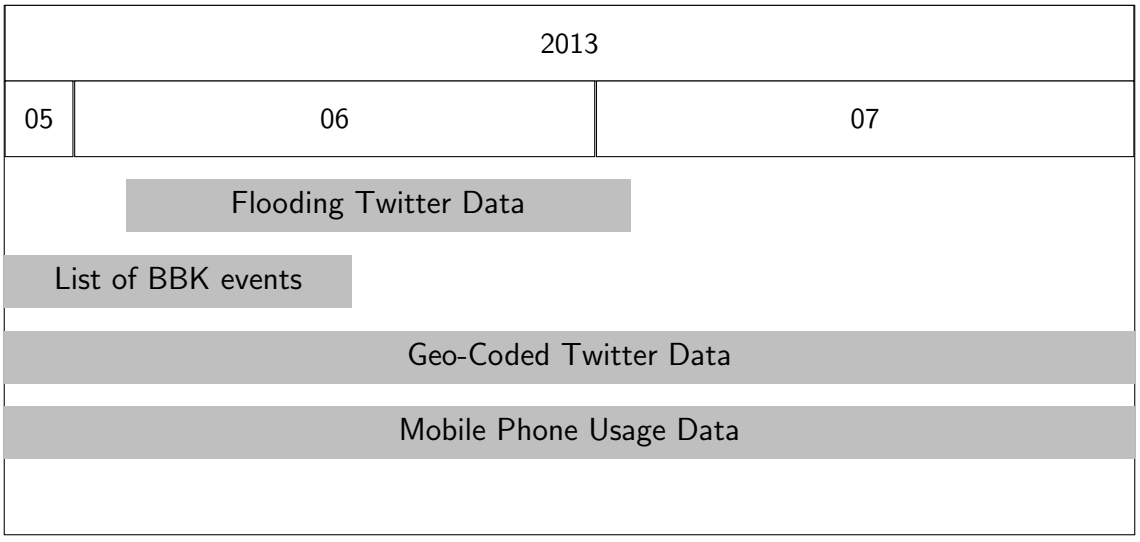

Figure 1: Data Availability (per month) for Recent Flooding Events in Germany.

The data samples are in the format described in Section 4. Utilizing the data, following analysis tasks are necessary to meet the requirements of the BBK.

3.1.2 Task I - Model Current Situation

We want to model normality in order to detect changes. Also, we want to achieve a complete picture of normality. In order to get insights on routine behaviour of the people, we extract regions with similar temporal distributions of Twitter messages i.e. spatial clusters of land use from Twitter data. Corresponding learning tasks are spatio-temporal clustering of incoming data streams, detection of frequent patterns, and Voronoi diagrams, which help to identify regular behaviour. This gives an overview of a situation and, thus, fulfils the demand for situation understanding.

For completing the picture of normality, extraction of spatio-temporal time series on people's presence from phone usage data is performed by geographically weighted regression. Detailed analyses take into account the time of the day, the season, the weather, and known events that attract many visitors. Hence, the normal behavior is also qualified by temporal aspects.

A model of the current situation supports the detection of deviations from normality and provides information on people's preferences in case of unexpected events. With the spatio-temporal time series (mobile phone usage and Twitter messages) we adjust a classification model to the normal behaviour in order to detect outliers. We exploit usage of recently developed core vector machines for distributed data streams [SBDM13].

3.1.3 Task II - Event Detection

From the incoming spatio-temporal data streams we detect automatically the incident events for the BBK. An annotated list of the interesting events during the recent flood is provided by the BBK. The goal is to identify the events more reliable than by manual inspection.

The task splits into

1. identification of the conditions in the data streams for raising an incident event and

2. extraction of its properties (event type, time, spatial extent).

With the classification model from task I that identifies normal situations in the data stream we aim at detecting anomalies. We create spatio-temporal clusters of Twitter messages (using a density based clustering e.g. DBSCAN [EKSX96]) with flooding related topics to inspect spreading of relevant messages over certain locations and to extract the spatial extent of derived events.

The identification of sequences of these anomalies which characterize an event are subject for complex event processing. Important constraints for the automatic detection are traceability and transparency of the decisions. The explanation of an event from incoming data streams is a link to WP3, focussing on complex event processing.

Validation of detected events is possible with comparison to the manually created report on important events.

3.1.4 Task III - Prediction

For early event detection and decision support, we predict future values of the incoming sensor data (traffic, population density, tides of rivers, street use, etc.) using the method we presented

in [PLM13]. Also the derived spatio-temporal time series e.g. the number of persons at a location are predicted for future time periods. The method will be run on several particular target variables. It will be compared with simpler methods, e.g. a spatio-temporal k-nearest neighbour.

3.1.5 Task IV - Generator

Since the data samples used in the project are not publicly available, generators of the data streams are required for independent tests of methods and for the publication of results. The artificial streams need to provide developers with the same characteristics as the original data streams and should contain events which can be analysed.

3.2 City Level Use Case

The city level use case is developed in cooperation with the Dublin City Council (DCC). Being situated at the sea side, the Irish capital Dublin is among the most jammed cities in Europe [Tom13]. Thus, first interviews with Dublin City Council revealed that two scenarios are of interest for the city: flooding and traffic jams.

For both scenarios the

1. early detection of critical events (flooding at a location or jam at a junction),
2. the estimation of information on traffic and flooding situation for unobserved locations, and
3. the prediction of future situations are expected.

Fulfilling these tasks supports the early warning of the citizens and the targeted control of the urban transportation system.

The data available for INSIGHT are vehicular counts derived from automatic traffic loops (SCATS), Live Drive Radio messages, Positions of the buses in Dublin NRA weather data, and Twitter messages.

The data sources will be described in Section 4 whereas here, we show its temporal availability. The formats of the data are either spatio-temporal time-series or trajectories.

In the *event detection*, the events (flooding or jam at a location) have to be derived from data streams. This comprises learning of the conditions to raise an event and detection of its spatial extent. The automatic explanation of an event (especially in case of jams) provides useful insights for the city council to improve traffic control.

The *estimation of information for unobserved locations* is crucial to the DCC. In this task, spatio-temporal time series, denoting the traffic situation and flooding situation for the city of Dublin, have to be derived from the incoming spatio-temporal time series and trajectories. The crowdsourcing application and uncertainty handling from WP3 should be incorporated to achieve reliable information on traffic situations and flooding situations in the city of Dublin.

The *prediction* of traffic or jam situations for the future is required in order to control the transportation system. Next, we describe the data sources briefly and derive more detailed tasks from the expectations of the DCC.

In addition, the INSIGHT system should provide individuals (i.e. registered and subscribed persons) with situation-dependent information

3.2.1 Availability of Test Data Samples

For the Dublin use case the following historical data samples are available: (1) NRA weather data, (2) Twitter data, (3) floating car data from the buses derived via SIRI.VM interface [SIR08], (4) Live Drive Radio [Liv13] messages and (5) vehicular counts derived from the already installed SCATS system [SCA13]. For a description of the data sources and output formats, please see Section 4. The temporal availability is shown in the Figure 2.

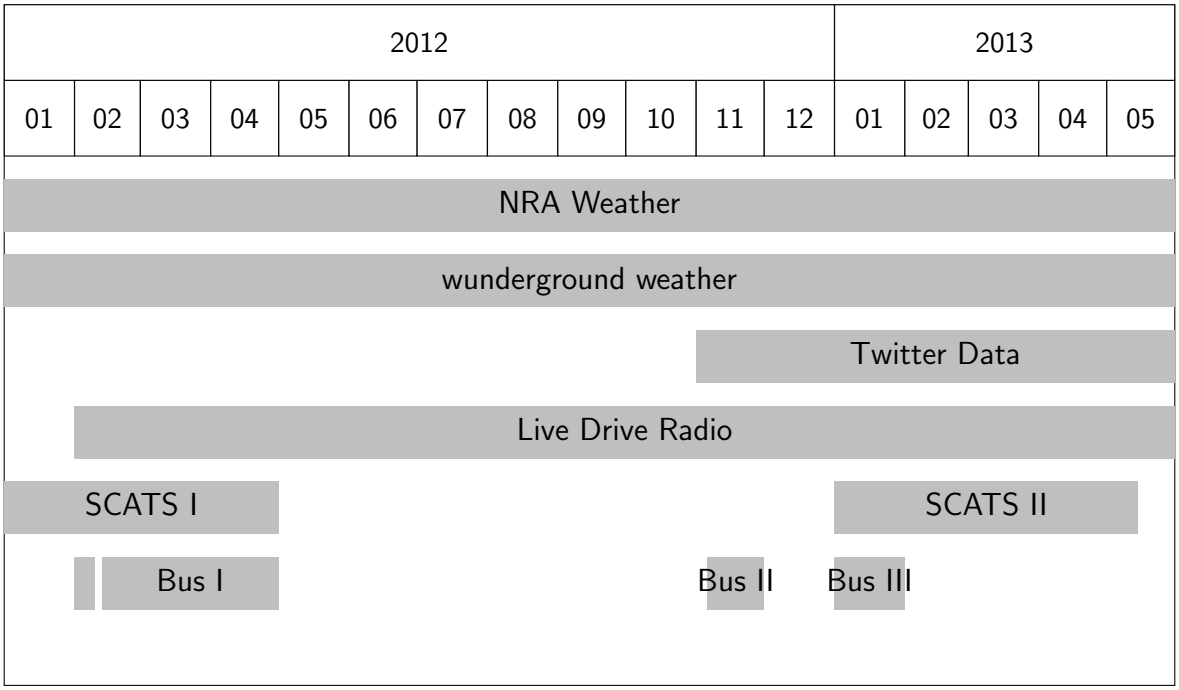

Figure 2: Data Availibility (per month) for Irish City Level Use Case Scenario, City of Dublin.

3.2.2 Data Quality Issue

A first inspection of the data identified some strange artifacts in few data samples. For example, the bus data set contains straight lines of points which may indicate interpolation of the raw data points.

The coordinates of the sensor locations in the SCATS data are sometimes assigned incorrectly. This causes errors in the intersection assignment. Also, traffic directions are missing. Therefore sensor information for one particular junction but different lanes cannot be directly joined.

The analysis and preparation of the data quality and understanding of the pre-processing steps is an important prerequisite for further method development. Jointly, the INSIGHT consortium focuses on this task. We examine filtering methods for trajectory data to remove outliers. Traffic directions can be determined in the preprocessing step that fetches the data from the SCATS system.

3.2.3 Task I - Model Current Situation

We want to model normality in order to detect changes. Also, we want to achieve a complete picture of normality.

In order to get insights on routine behaviour of the people, we extract regions with similar temporal distributions of traffic flow i.e. spatial clusters of land use from SCATS data. This supports detection of deviations from normality and provides information on people's preferences in case of unexpected events.

For completing the picture of normality, extraction of spatio-temporal time series on traffic flow for unobserved locations is important. We address the task with a method similarly to Kriging [Kri51] that models the pairwise covariances among the traffic flow at different locations. In contrast to Kriging, the model in [LXMW12] incorporates in the covariances among the traffic flow not just distances of the locations but the centrality of the streets in the street network and the probability to co-visit both locations. The uncertainties at unobserved locations are reduced by incorporation of crowd sourced data (information on traffic jams, measurement of number of cars, trajectories). The incorporation of user feedback links to WP4 and their crowdsourcing and uncertainty handling. An example of how social media users could be utilized under this crowd-sourcing setting is described in [VGBK13a].

With the spatio-temporal time series from SCATS, weather and Twitter we adjust a classification model to the normal behaviour in order to detect outliers. We exploit usage of recently developed core vector machines for distributed data streams [SBDM13].

3.2.4 Task II - Event Detection

From the incoming spatio-temporal data streams (SCATS, Live Drive Radio, Twitter, weather) and trajectories (bus data) we detect automatically the traffic jam and flooding events for the DCC. The aim is to use the information on traffic and flooding events for early warning and smart traffic control (e.g. after detection of a flooding event on a road, an alternative route on a parallel road can be proposed).

The task splits into

1. identification of the conditions in the data streams for raising an event and
2. extraction of its spatial extent.

With the classification model from previous task (core vector machine for vertically distributed data [SBDM13]) that identifies normal situations in the data stream we aim at detecting anomalies. We create spatio-temporal clusters of Twitter messages and Live Drive Radio messages (using a density based clustering e.g. DBSCAN [EKSX96]) with traffic and flooding related topics to inspect spreading of relevant messages over certain locations and to extract the spatial extent of derived events. The explanation of an event from incoming data streams is a link to WP3, focussing on complex event processing.

3.2.5 Task III - Prediction

For early event detection and decision support, we predict future values of the incoming sensor data using graphical models, presented in [PLM13]. Also the derived spatio-temporal time series, e.g., the traffic situations are predicted for future time periods.

3.2.6 Task IV - Geo-coding, Map Matching

The Live Drive Radio messages and some Twitter messages contain spatial information (e.g. the congestion state of a road) but do not provide geographic coordinates.

We identify the spatio-temporal coordinates from the spatio-temporal time series in order to incorporate the data in analyses, see Section 5.1.

4 Data Format Descriptions

NB: This section is shared by both Deliverables 2.1, 3.1, and 5.1. It elaborates upon the listing of data sources in Deliverable 6.1.

In the city-wide use cases, the inputs to the system include map data, traces of vehicle movement (e.g. buses), vehicle-count data in the Sydney Co-ordinated Adaptive Traffic System (SCATS) format, weather reports, twitter streams, and user feedback. The primary outputs are *Alerts* and *Events*; *Surveys* and *Rewards* are also output as intermediate steps in soliciting feedback from citizens. The system also produces *Routes* based on the up-to-date road-segment availability and travel-time estimates in response to requests from users, typically citizens. This section describes the data format of the testdata provided for analysis and presents the format of the output. For a comprehensive description (including intermediate data) see Deliverable 2.1.

In the nation-wide use case, the inputs to the system include map data, event descriptions and coordinates, as provided by the BBK, mobile phone usage data, twitter data, weather reports, and user feedback. The primary outputs are *Alerts* and *Events*; *Surveys* and *Rewards* are also output as intermediate steps in soliciting feedback from citizens. The system also produces *Routes* based on evacuation planning using up-to-date road-segment availability and travel-time estimates. This section describes the data format of the testdata provided for analysis and presents the format of the output. For a comprehensive description (including intermediate data) see Deliverable 2.1.

Considering the overlap of the inputs and outputs in all use cases, we present the inputs and outputs next. Notice that geo-localised parts of the input and output store the longitude and latitude in WGS84 format used by the Global Positioning System (GPS).

4.1 Input Data Sources

In this section, we detail each of the input data sources we expect the system to ingest. For details of the decentralised, generic architecture which will be employed to manage these sources, see Deliverable 2.1, Section 6.

4.1.1 Vehicle-count data (SCATS)

The Sydney Co-ordinated Adaptive Traffic System provides information on vehicular traffic at fixed sensor locations as spatio-temporal time series. The SCATS data are produced by aggregating the primary source data that are collected by the Dublin SCATS traffic sensor monitoring system.The primary data are given in the *Strategic Monitoring (SM)* format [CM03]. Each sensor sends messages with varying frequency (depending on the location, conditions and

other factors). The SM format specifies the message parameters. These messages, in addition to the information that is maintained after the aggregation to the SCATS format, includes additional system information that is not used in our analysis. This SCATS data [SD80] is a sequence of tuples (z, m, t), where z is a geographic location of the observation (the sensor position), m is a metric and t is an integer. The location is either *detector_index*, or a vector consisting of of a number of elements, including the GPS coordinates of the detector. The metric m contains:

- *aggerateCount*: aggregated vehicles volume count on the arm,
- *flow*: flow ratio calculated as the volume divided by the highest volume that has been measured in a sliding window of a week.

Integer t element is the timestamp of the 5 minute interval in POSIX time, i.e. the number of microseconds that have elapsed since 00:00:00 Coordinated Universal Time (UTC), 1 January 1970. This can be seen as a flattening of the historical data in the data model displayed in Figure 3.

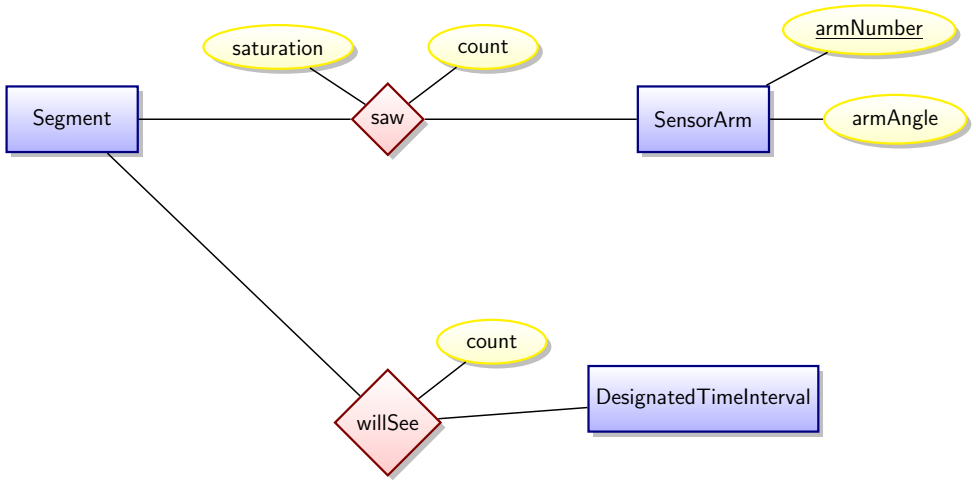

Figure 3: An idealised model of SCATS data.

In practice, data are imported into this model from two different data sources. For a period of time in 2012, the data have been recorded[5] as a sequence of following tuples:

- *streetSegId*: a unique identifier for a street segment ID,
- *armNumber*: an identifier for the arm on a street segment,
- *armAngle*: bearing of the arm,
- *gpsArm*: GPS position 20 meters into the arm,
- *gpsCentroid*: GPS position of the centroid of the intersection.

[5]available at:
http://www.dublinked.ie/datastore/server/FileServerWeb/FileChecker?metadataUUID=a5aaaf4ca2404e0ca02e21fc0bdf1882&filename=SCATS-Dublin.zip

- *aggerateCount*: aggregated vehicles volume count on the arm,
- *flow*: flow ratio calculated as the volume divided by the highest volume that has been measured in a sliding window of a week.

These samples are captured at 6-minute intervals. The more recent data from 01/01/2013 onwards are sampled every minute and are provided by DCC and IBM as a sequence of following items[6]:

- *year*, *month*, *day*, *hour*, *minute*: denoting the timestamp
- *site*: measurement location
- *strategicApproachLink*
- *isLink*
- *detector_index*: index of the detector
- *degreeOfSaturation*: $flow/capacity$
- *flow*: current flow value

These samples are used in conjunction with a file, `detectors.csv`[7], which contains the coordinates. The *detector_index* from the sequence refers to the *lane number* in the `detectors.csv` file.

4.1.2 Traces of vehicle movement (Bus GPS)

In principle, the GPS data are a sequence of vectors $y_{z,t}$, where z is a traffic object, e.g. a bus with an on-board GPS receiver, and t is an integer, e.g. the POSIX time of the acquisition.

In practice, the the data are imported from three very different data sources, even in the case of Dublin. Instead of plain coordinates, there is a more complex data model based on the General Transit Feed Specification. There, a vehicleJourney (or "route" in GTFS) is a particular instance of a journeyPattern starting at a given time. A journeyPattern is a sequence of two or more stops. In between each two stops, there are one or more blocks within a trip (or "segments" in GTFS and elsewhere)[8]. Notice that the production time table starts at 6am and ends at 3am in Dublin.

The first source of GPS traces captures the movement of buses in Dublin in the period from 01/02/2012 till 30/04/2012 (except the days 10th till 12th February 2012) and contains the following values:

[6] available at:
svn+ssh://madgik/svn/source/WCITY.zip
svn+ssh://madgik/svn/source/CCITY.zip
svn+ssh://madgik/svn/source/SCITY.zip
svn+ssh://madgik/svn/source/NCITY.zip

[7] available at:
svn+ssh://madgik/svn/source/detectors.csv

[8] Please see `https://developers.google.com/transit/gtfs/reference` for a detailed reference.

- *timestamp*: timestamp microseconds since 01/01/1970 00:00:00 GMT,
- *lineId*: bus line identifier,
- *direction*: a string identifying the direction,
- *journeyPatternId*
- *timeFrame*: the start date of the production time table (in Dublin the production time table starts at 6am and ends at 3am),
- *vehicleJourneyId*: a given run on the journey pattern,
- *operator*: bus operator, not the driver,
- *congestion*: boolean value [0=no,1=yes],
- *gpsPos*: GPS position of the vehicle,
- *delay*: seconds, negative if bus is ahead of schedule,
- *blockId*: section identifier of the journey pattern,
- *vehicleId*: vehicle identifier,
- *stopId*: stop identifier,
- *atStop*: boolean value [0=no,1=yes].

The second source of GPS traces captures the movement of buses in Dublin during a part of November 2012 (06/11/2012 till 30/11/2012) and contains tuples of the following elements:

- *timestamp*: timestamp microseconds since 01/01/1970 00:00:00 GMT,
- *lineId*: bus line identifier,
- *direction*: a string identifying the direction,
- *journeyPatternId*
- *timeFrame*: the start date of the production time table (in Dublin the production time table starts at 6am and ends at 3am),
- *vehicleJourneyId*: a given run on the journey pattern,
- *operator*: bus operator, not the driver,
- *congestion*: boolean value [0=no,1=yes],
- *gpsPos*: GPS position of the vehicle,
- *delay*: seconds, negative if bus is ahead of schedule,

- *blockId*: section identifier of the journey pattern,
- *vehicleId*: vehicle identifier,
- *stopId*: stop identifier,
- *atStop*: boolean value [0=no,1=yes].

The third source of GPS traces captures the movement of buses in Dublin during January 2013 (01/01/2013 till 31/01/2013) and contains tuples of the following elements:

- *timestamp*: timestamp microseconds since 01/01/1970 00:00:00 GMT,
- *lineId*: bus line identifier,
- *direction*: a string identifying the direction,
- *journeyPatternId*
- *timeFrame*: the start date of the production time table (in Dublin the production time table starts at 6am and ends at 3am),
- *vehicleJourneyId*: a given run on the journey pattern,
- *operator*: bus operator, not the driver,
- *congestion*: boolean value [0=no,1=yes],
- *gpsPos*: GPS position of the vehicle,
- *delay*: seconds, negative if bus is ahead of schedule,
- *blockId*: section identifier of the journey pattern,
- *vehicleId*: vehicle identifier,
- *stopId*: stop identifier,
- *atStop*: boolean value [0=no,1=yes].

4.1.3 Map Data and Transit Graph (OSM, OTP, GTFS)

The street map is represented as a graph, where vertices represent important locations in space for a given means of transport (e.g. road intersections for cars). Each edge represents a means of traversing between the vertices, which can involve actual movement (e.g. between two intersections) or waiting (e.g. at a bus-stop). The graph is illustrated in Figure 4.

The overall data model is rather complex, but closely parallels those used by OpenStreetMap, OpenTripPlanner, and the General Transit Feed Specification; we hence direct the reader to the reference documentation for those.

Our custom extensions to the standard format consist of:

- the travel-time estimates, which correspond to the weights of the edges in the graph

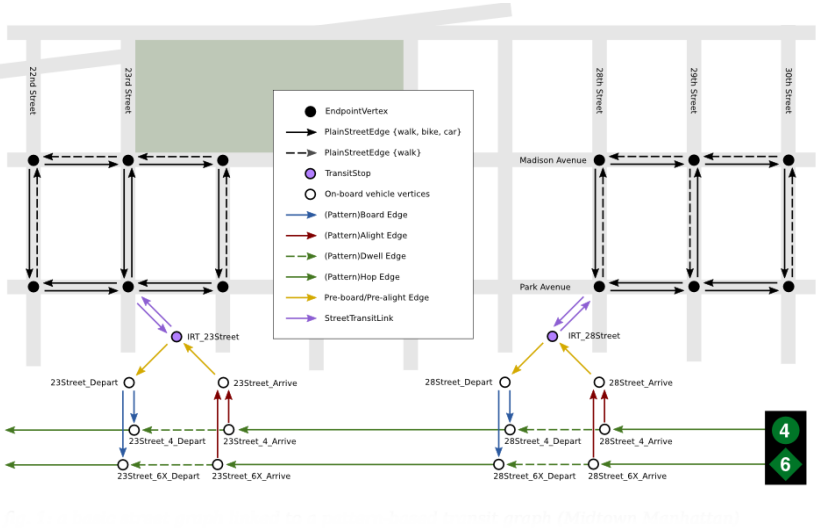

Figure 4: The vertex-based transit graph. Cited in verbatim from `https://github.com/openplans/OpenTripPlanner/wiki/GraphStructure`.

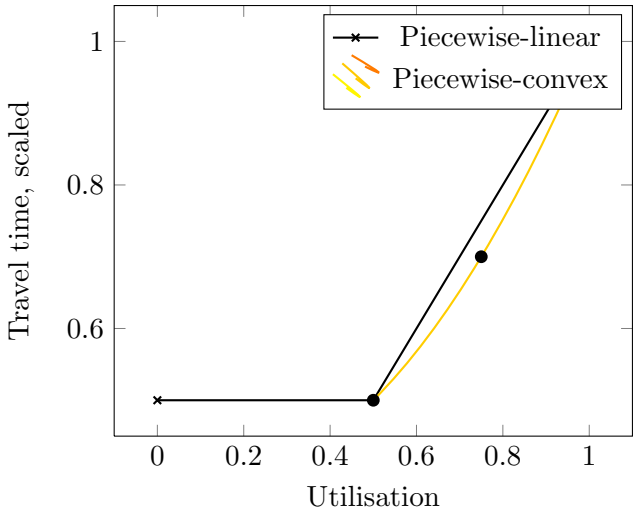

Figure 5: An illustration of a delay function, which gives the travel-time along a segment of a road as a function of its utilisation, i.e. the ratio of the number of concurrent users to the maximum thereof.

- the altitude data, which correspond to weights of the vertices in the graph

The travel-time estimates are stored as delay functions and vehicle count data. A delay function gives the travel-time along a segment of a road as a function of its utilisation, i.e. the ratio of the number of concurrent users to the maximum thereof. The delay functions are computed from the vehicle-count data (SCATS) and traces of vehicle movement (Bus GPS) described above. The vehicle-count data are stored in the format described previously, c.f. Figure 3. See Figure 6 for the entity-relationship digram.

The altitude data ("a model of terrain") improves the modelling in the flooding use case.

The model of terrain for Germany is based on the OpenTopoMap, `http://opentopomap.org/`, which in turn is based on the STS-99 Shuttle Radar Topography Mission (SRTM) data set, which has been collected in a remote sensing exercise on board Space Shuttle Endeavour in February 2000. Elsewhere, one can utilise NASA's ASTER Global dataset at `http://www.echo.nasa.gov/`. ASTER (Advanced Spaceborne Thermal Emission and Reflection Radiometer) orbits the Earth on board the Terra satellite since 1999 and collects data since February 2000. The dataset collected by ASTER is known also as the Global Digital Elevation Model. In either case, one obtains an approximate elevation for each point of interest.

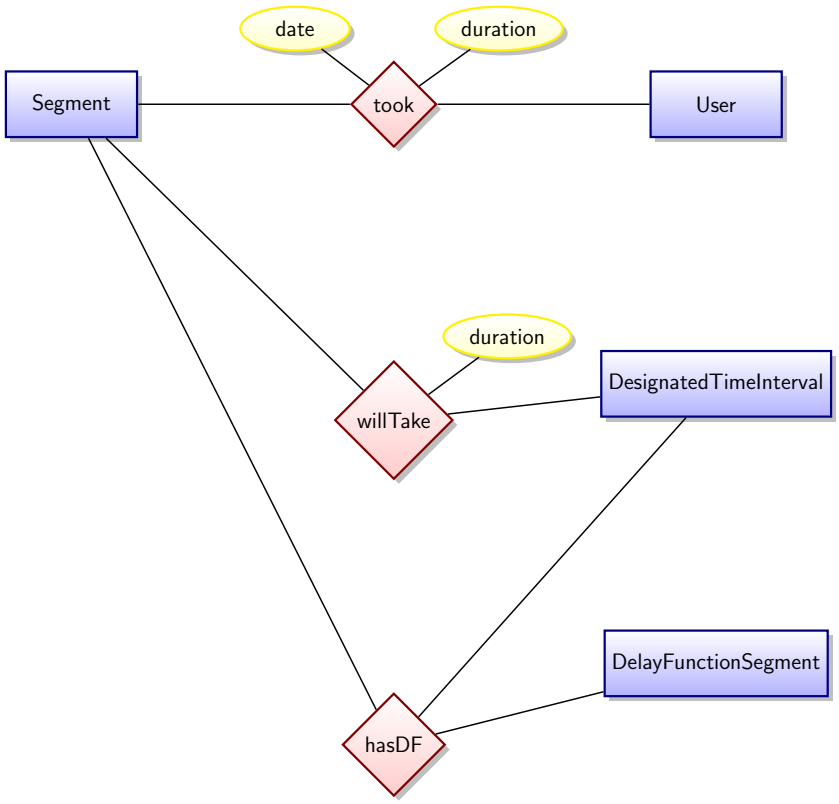

Figure 6: The travel-times data model.

4.1.4 Weather Reports (NRA, WU)

Ireland's National Roads Authority (NRA) maintains a network of sensor stations around Dublin city, each of which samples a variety of environmental factors at ten-minute intervals. As part of the initial data-collection effort, we have created a tool which pulls information from thirteen of these stations into a central database. At present, our focus is on creating a historical archive for future exploitation rather than providing the data in real-time, and as such the data is harvested only once per day; this can, of course, be changed at a later date to account for the project's evolving requirements. The database also contains meta-information about the various data points, allowing human-readable reports to be generated with ease.

The database can be queried using standard SQL. It is currently only accessible from within

IBM, but it can be easily migrated to another location as necessary. Figure 7 illustrates the structure of the database.

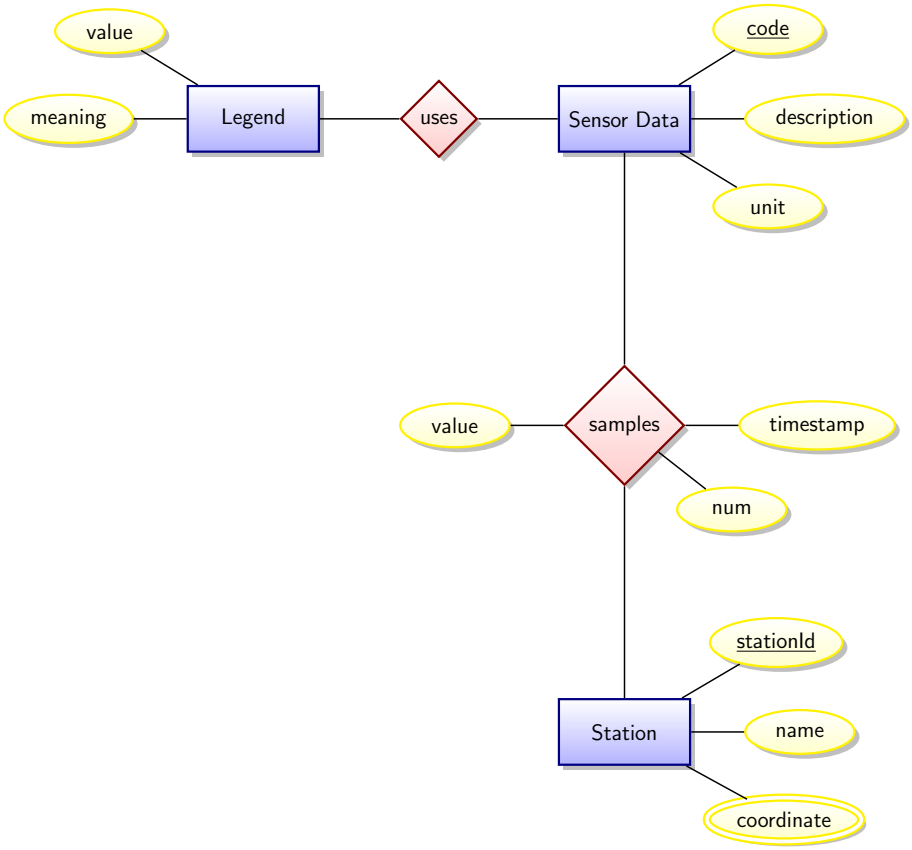

Figure 7: NRA data model. The field *CODE* indicates the type of reading. The *SENSOR-DATA* table provides a full-text description of each code, along with its associated unit of measurement. In cases where the code's unit is a status value, *LEGEND.MEANING* provides a plain-English explanation of each possible status.

The full list of stations from which these sensor data are drawn is provided in Table 1, while some of the more interesting points captured by the database are highlighted in Table 2. A visualisation of some of this data is shown in Figure 8. We envision this database as being of particular relevance to the Flooding use case - an event could be triggered if, for instance, total precipitation was observed to deviate significantly from trend over a given period - but data on the changing state of the weather would naturally be of use in explaining traffic congestion patterns as well.

Alternatively, there are weather data available at wunderground[9]. These can be queried for weather information at a particular coordinate, e.g. Dublin, posing the following request (the <key> has to be generated in advance by registering to the wunderground website):

`http://api.wunderground.com/api/<key>/hourly10day/q/Ireland/Dublin.json`

As result, a json object is returned which contains the following fields:

[9] http://www.wunderground.com

Table 1: NRA stations

Dublin Port Tunnel	M1 Drogheda Bypass
M1 Dublin Airport	M11 Bray Bypass
M4 Enfield	M50 Blanchardstown Master
M50 Blanchardstown Slave	M50 Dublin Airport
M50 Sandyford Bypass Tipping Bucket	M50 Sandyford Master
M7 Newbridge Bypass	M7 Portlaoise Bypass
N81 Tallaght	

Table 2: Illustrative NRA datapoints

Code	**Description**	**Unit**
CL	Cloud State	Status Code: Clear, Cloud, Cloud and Rain
PW	Present Weather	Status Code: 0 (unobstructed) to 99 (tornado)
WL	Water Layer	mm
SL	Snow Layer	mm
IL	Ice Layer	mm
RH	Relative Humidity	%
PR	Precipitation Total	mm
RI	Rain Intensity	mm/h
P	Pressure	hpa
T	Air Temperature	°C
TS	Surface Temperature	°C
VI	Visibility	m
WD	Wind Direction	°
WS	Wind Speed	m/s

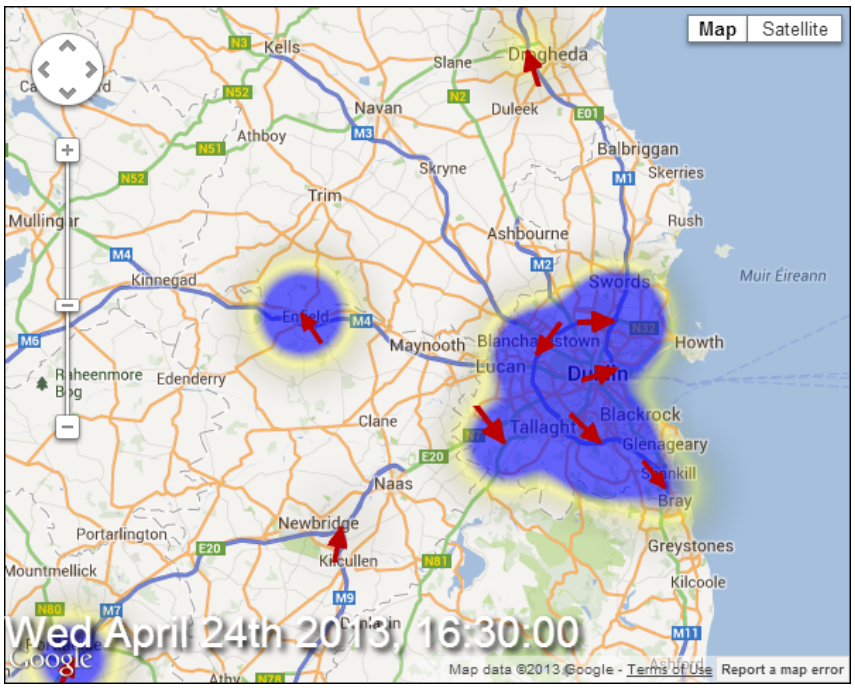

Figure 8: NRA data visualisation. Arrows indicate wind speed and direction; heatmap blobs indicate cumulative rain intensity at each station, in mm/h

- *FCTTIME*: the time of the weather forecast
- *temp*: the temperature
- *condition*: the weather condition, e.g. "Rain"
- *icon*: an icon to depict on a map, e.g. "rain"
- *icon_url*: link to an icon for graphical user interfaces
- *humidity*: humidity in percent
- *feelslike*: the perceived temperature
- and many undocumented fields.

4.1.5 Short messages (Twitter)

Further input to the system is provided by Twitter, a short messaging service. Twitter issues a stream of messages ("tweets") up to 140 characters long, optionally including one or more "hashtags" - that is, arbitrary words preceded with a hash character, used to denote topics to which the message relates (e.g., `#dublin`). Tweets may also include links to websites and other auxiliary data; see Figure 10 for some examples.

The Twitter web application and its public API allow developers to retrieve a substream of messages based on a given set of criteria; specifics hashtags, for instance, or tweets produced by a certain user, etc. The stream is a sequence of *tweets*, which primarily consist of:

- *tweetId*: a unique tweet identifier
- *date*: integer, POSIX time of the tweet publication
- *twitterUserId*: twitter user identifier
- *coordinate*: geo-localization of tweet
- *messageText*: tweet text.

The stream is indexed by hashtag and clustered according to a given set of criteria (e.g. GPS co-ordinates). See Figure 9 for the entity-relationship digram.

The Twitter substream generated within a geographical area of interest can be isolated by following relevant users (e.g., `@livedrive`) and monitoring certain hashtags (e.g., `#dublin`). Note that the input stream is not limited to users who are already known to the INSIGHT system; all tweets by Twitter users who are publicly tweeting in the area of interest are collected.

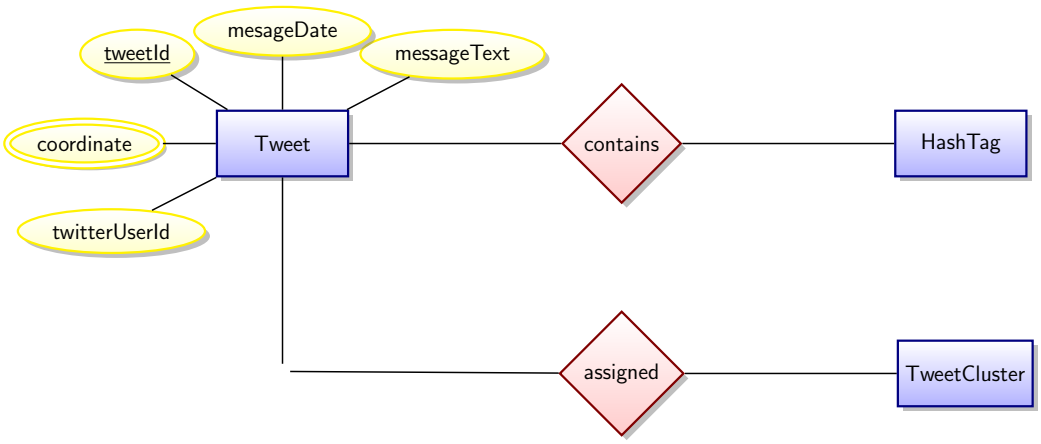

Figure 9: A data model of tweets.

In more detail, we can access the following fields in the Twitter stream:

- *tweetId*: a unique Tweet ID, assigned by Twitter
- *twitterUserId*: a twitter-ID of Tweeting user. Unique per Twitter account.
- *twitterUserScreenName*: mnemonic user name (login name)
- *latitude*: geographic latitude of sending device
- *longitude*: geographic longitude of sending device
- *messageText*: the actual tweet in raw textual form. It may include non-ASCII characters.
- *messageDate*: timestamp of message sending. Format 'YYYY-MM-DD hh:mm:ss'
- *location*: tweet location place name, for Gazetteer lookups

```
Wooow!! Das Wasser vom Rhein ist ziemlich hoch!!
```

```
#AltLostau ist jetzt nur noch mit dem Boot erreichbar
 - Schuldamm wurde ueberspuelt #Hochhwasser #lostau
http://t.co/rfM6ylcBRT
```

```
In #Lostau steht das Wasser jetzt ca. 1km hinter dem
Deich in den Kellern der Haeuser #Hochwaser #Madeburg
```

```
Bodensee-Pegel: 343,6 cm! - http://t.co/JWhYfRSk -
Hochwasser-Infos: http://t.co/dG7uuYK1
```

Figure 10: Sample Tweets mentioning floods in Germany in July 2013. Notice the misspelled words.

- *countryCode*: ISO short country code (two characters)
- *tokens*: always null
- *retweetStatusId*: referred (tweetId) to embedded retweet (original tweet). 0 if not a retweet, -1 if not set or invalid value.
- *isRetweeted*: boolean flag ('y'—'n') if tweet contained a retweet
- *replyStatusId*: referrer (tweetId) if tweet is-in-reply-to. -1 if not an answer-to tweet.
- *replyUserId*: referrer (twitterUserId) to author of original tweet being answered. -1 if not an answer-to tweet.
- *isFavorite*: Boolean flag ('y'—'n') if tweet was marked as favorite
- *followersCount*: Number of Twitter users currently following tweet author
- *followingCount*: Number of Twitter users the tweet author currently follows

For Dublin, the real-time stream follows the `#dublin` hashtag. Twitter data sets are available for the period from 05/11/2012 till 24/07/2013. Batch data samples are retrieved from the Twitter API using a spatial query.

Further for Dublin, The Live Drive Radio data set results from Twitter messages sent by people driving in Dublin that report traffic hazards to the local radio. The messages are derived from the Twitter API by following the @livedrive tag and are available for a period from 01/02/2012 till 30/04/2012 as a table[10] with following fields:

- *messageDate*: the time of the message

[10] available at:
svn+ssh://madgik/svn/source/LiveDriveRadioMessages.txt

- *messageText*: the text message

For Germany, geo-coded Twitter messages are available for the period from 22/11/2012 till 24/07/2012. The data are collected by querying the Twitter API for tweets with geographical coordinates located within a given bounding rectangle. After the collection, the tweets have been additionally filtered by checking the country code and whether the coordinates are indeed within the bounding rectangle of Germany, since the collected data also contained messages from the Netherlands and other countries as well as messages with the country code of Germany but located outside Germany, which means that the Twitter API may not provide exactly what you request. The final dataset that has been collected consists of about 6 million records.

For the summer 2012 floods in Germany, messages were retrieved by following the hashtag #hochwasser in Twitter API. This resulted in 178516 tweets. The data is available for the period from 04/06/2012 till 02/07/2012. Coordinates are available for 3975 tweets (about 2% of the data set).

4.1.6 Mobile phone data (IAIS)

The raw data is provided by Vodafone Germany. Vodafone (VF) is one of the largest mobile communications companies in Germany serving more than 32 Mio. customers across the country in 2012/13. It owns and operates a nationwide mobile network and offers a wide range of telecommunication services to its customers.

The network itself consists of more than 50.000 cells for different frequency ranges.

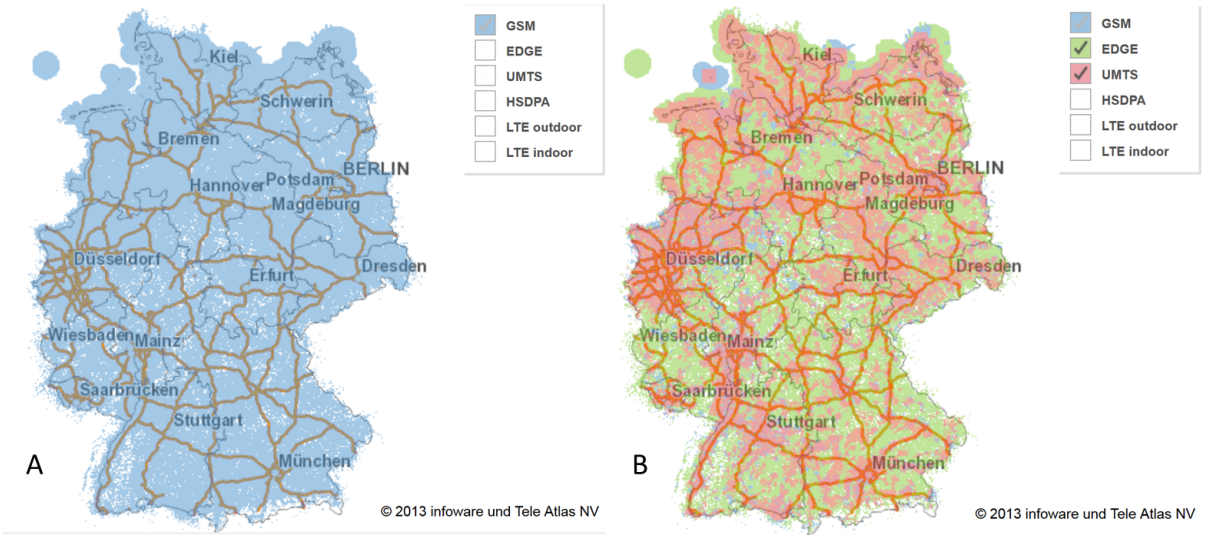

Figure 11: Network coverage of Vodafone in Germany. (a) shows that GSM services are provided in all blue colored areas, (b) visualizes the multi-layer structure of the mobile network. Source: vodafone.de, 08/2013

The network itself consists of more than 50.000 cells for different frequency ranges (see Figure 11). The coverage of the mobile network varies depending on the topology and land use of an area. In total it covers most of the populated areas. The mobile network of Vodafone operates on all three frequency ranges GSM+GPRS, UMTS, and LTE. This multi-layer structure of the network makes it fairly robust against failures and has proven to operate

through disasters (e.g. during Hurricane Sandy, earthquake in Japan and Haiti). Especially, SMS services have proven their operation during disaster situations. GSM cells, for example, can extend their range up to 40 to $50km^2$

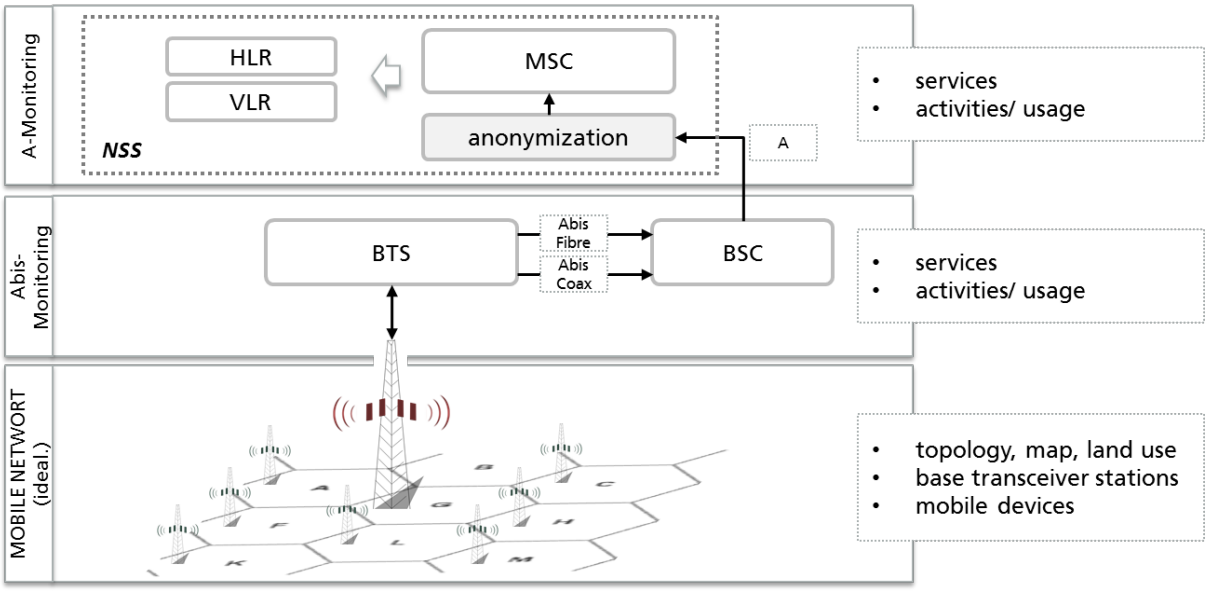

Figure 12: Data stream originating in an idealised mobile network.

Based on the location of the mobile device it will try to establish a permanent connection to the network via the Base Transceiver Station. Each time a mobile device becomes active in the network, e.g. to make or receive a phone call or send an SMS, an active connection is established. A group of antennas (BTS) is controlled by a Base Station Controller (BSC). BSC monitor network connections and are responsible for power control or initiations of Handovers (switching between two cells). If, for example, the Handover involves two BSC the Mobile Switching Station (MSC) gets involved. The MSC also keeps track of which device (customer) is logged on the system updating the Home Location Register (HLR) and the Visitor Location Register (VLR). Thus the system (MSC) always knows where a device is located in order to provide phone and data services.

The flow of data is described in Figure 12. The mobile devices usually choses the cell tower with the best reception and expected quality. While negotiating with the controllers of the antennas the current load of the antenna is taken into account. Therefore the carried load on service-providing elements such as telephone circuits or switching equipment has to be measured. This is part of the quality and network integrity measures any mobile operator has to implement.

One way of measuring the carried load of an antenna is by calculating the network performance parameter called 'Erlang' (Erl) for a given time interval l. Normalized carried traffic (load) in Erlang is related to the call arrival rate, λ, and the average call duration, h, by:

$$Erl = \frac{\lambda h}{l} \tag{1}$$

The traffic load is aggregated hourly with $l = 60$. All data are fully anonymized information

on the usage of a network. The higher the Erlang, the more network traffic is registered for one cell and hour. Note that the Erlang value is dimensionless.

The dataset consists of the Erlang values for all cells in Germany. They cover the frequency ranges 900 MHz and 1.800 MHz. The data is available since April 2010 and has an temporal resolution of one hour. Periods of vacation, holidays etc. are contained in the data as well as traffic events and emergency situations like the summer flood in Germany 2013. Each record (tuple) in the database has the following structure:

- *timeInterval*: hourly timeslice (range)
- *cellId*: unique identifier of a cell
- *erlValue*: Erlang value
- *sourceInformation*: origin of the data and information on the source

To relate this data with the street network and other spatial data (e.g. Points-of-Interests) a geometry of the coverage area of each cell is part of the dataset. Different approaches exits to represent the coverage area: approximation, probability based, tessellation. Figure 13 shows a spatial probability based calculation of a coverage area. The fragmented geometry results from combined affects of network topology and topography of the area.

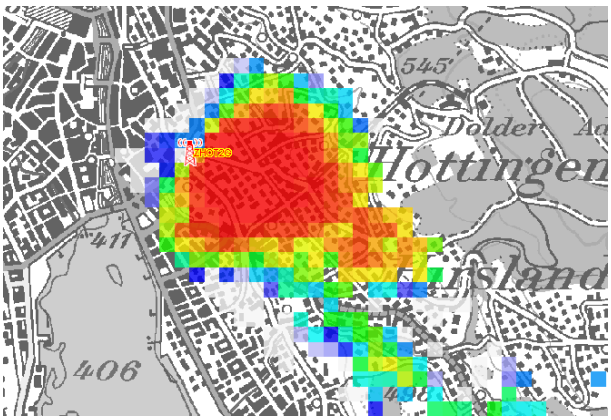

Figure 13: Result of a theoretical coverage calculation. Colors represent the probability that a mobile device is connected to the shown antenna (red=high probability, blue=low probability), Source: courtesy of Swisscom

This probability-based coverage model is extremely complex and computationally expensive. Ellipsoid approximations have been developed to reduce complexity. Also tessellation like Voronoi tessellation are commonly used by mobile operators and network vendors (see Figure 12) We operate with all three types of representation models. The structure is:

- *cellId*: unique identifier of a cell
- *geometry*: point, polygon or multi-polygon shapes of a cell as a set of vertices (X-Y-coordinates in a spatial reference system, WGS84) encoding the covering area

A description of the data model is given in Figure 14.

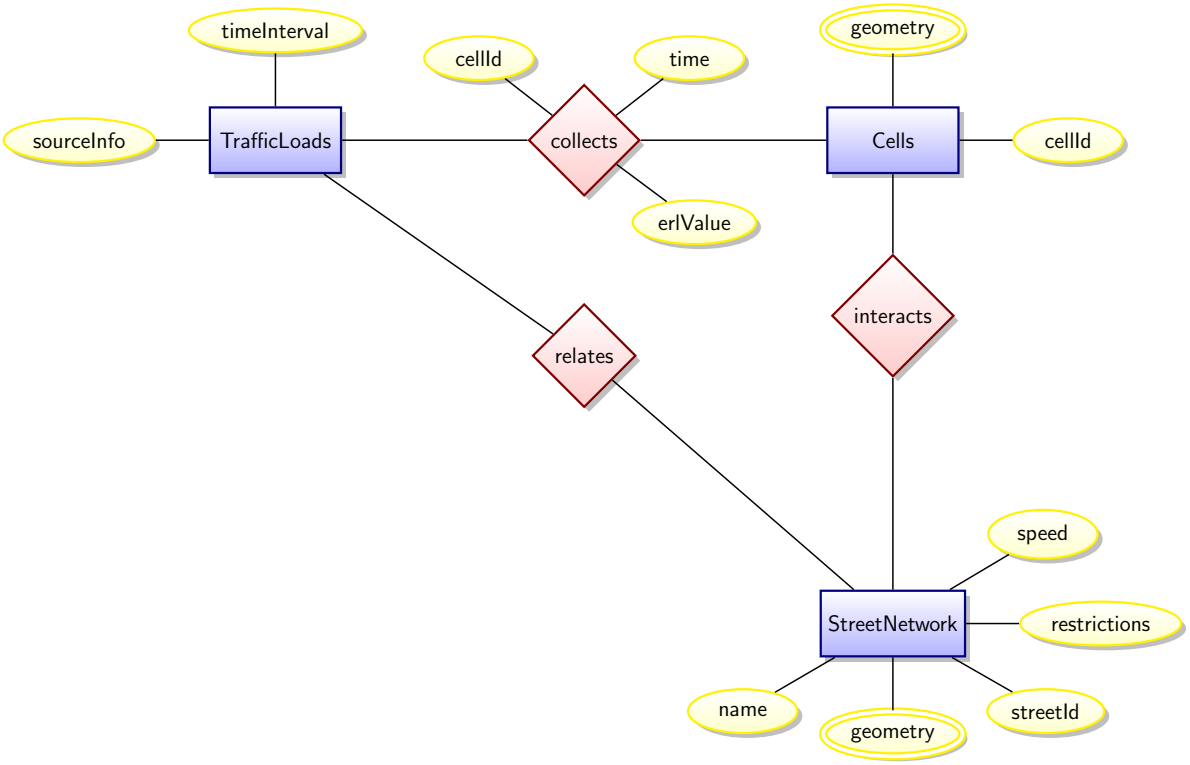

Figure 14: A simplified data model of mobile network data.

4.1.7 Traffic Frequency Data (IAIS)

Based on the Navteq street network of entire Germany with over 6.8 Mio. street segments, Fraunhofer has build a mobility model that predicts the average number of cars and pedestrians per hour. The primary objects of interest are street segments, which are parts of a street between two intersections. Each segment possesses a geometry object and has attached information about the type of street, name of street, direction, speed class, and length (see Figure 15).

The model integrates various sources of data that is linked to mobility. The multi-source approach comprises several sources of different quality and spatial resolution. For a subset of street segments frequency counts derived from video data are available, in total around 100,000 measurements for Germany. For some cities more than 2,000 measurements exists, while for others only a few dozens.

Each segment was measured at 4 different days and 4 different times between 7am and 7pm. Each measurement lasts 6 minutes. The number of cars and pedestrians were counted manually. For validation purposes we have compared video measurements with long-term traffic counts made by the federal state at number of locations where such measurements coexist. The correlation is very high (0.97), demonstrating that this kind of measurements can give accurate data for the purpose at hand.

In addition, demographic and socio-economic data about the vicinity of street segment was used, e.g. age structure, gender distribution, employment. It usually exists for official districts like post code areas. We also integrated supplementary manual traffic counts at neuralgic points in the street network.

The resulting dataset is structured:

- *streetID*: unique identifier of the street network
- *freqCar*: average frequency of cars
- *freqPed*: average frequency of pedestrians

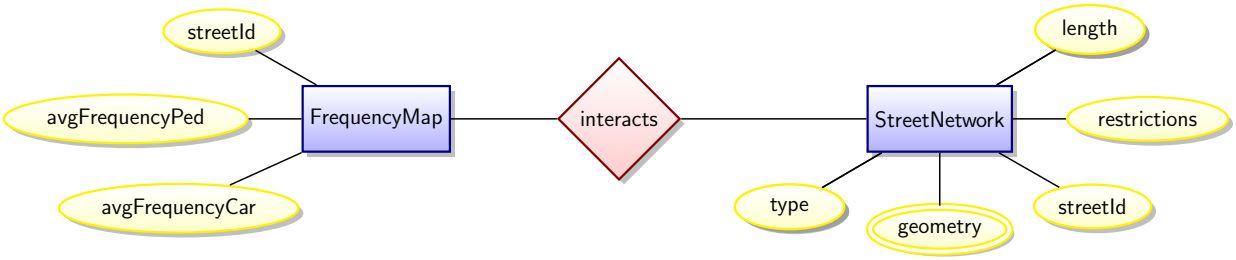

Figure 15: A simplified data model of the frequency map for Germany.

4.1.8 Event descriptions (BBK)

During the recent floods in Germany, BBK collated a list of all events mentioned in their daily reports. The resulting table comprises the events that were reported within the time period from 28/05/2013 till 16/06/2013. It contains the following fields for every event:

- *Incident*: the event that occurred
- *Location*: textual description of the location of the event
- *Time*: date and time of the BBK report that broadcasted the event

4.2 Outputs

In this section, we detail each of the various outputs we expect the system to produce. These are drawn from the use-case requirements; see D6.1 for details. We situate these outputs within our architecture in Deliverable 2.1, Section 6. N.B.: This section is shared by Deliverables 2.1, 3.5, and 5.1.

4.2.1 Events

Event objects are one of the possible outputs of the INSIGHT system. The event object is composed of:

- *eventId*: unique identifier
- *t*: integer, POSIX time of the event generation
- *coordinate*: a position of the event in terms of GPS coordinates

Each event object is associated with a set of explanation labels and corresponding weights:

- *eventId*: an identifier of the event being explained
- *explanationText*: a text of an explanation
- *explanationWeight*: a weight of an explanation
- *explanationCoordinate*: a linked position

but may be also stored as a write-once large object (BLOB) for each event. See Figure 16 for the entity-relationship diagram. Each weight represent the system's belief, based on all available measurements, that the associated explanation label is correct.

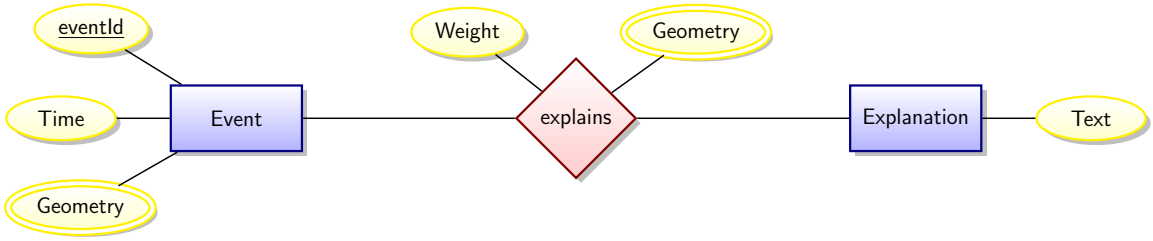

Figure 16: A data model of events.

4.2.2 Alerts

As a broadcast medium from the INSIGHT system to the people, location dependent events (so-called alerts) can be provided to every person individually. The person can subscribe to certain types of alerts by specifying the radius within which the alerts of a particular type are of interest. Alerts are issued for a certain period at a certain location. When the user enters within the pre-specified radius of the location, where an alert of the type user opted in for is active, the alert is sent to the user. See Figure 17 for the entity-relationship diagram.

4.2.3 Routes (OTP)

The system provides situation dependent routing capabilities and *routes* computed by the system can be provided. In the traffic scenario, this may consist of requested routes between a specified origin and destination, computed using real-time travel-time estimates. In the flooding scenario, a central authority may issue an evacuation plan, out of which a route is distributed to each user.

The OpenTripPlanner data model is employed to output routing information to the user, as suggested by the entity-relationship diagram in Figure 18. A trip plan consists of one or more itineraries. Each itinerary consists of one or more legs, which specify how to get from one place to another.

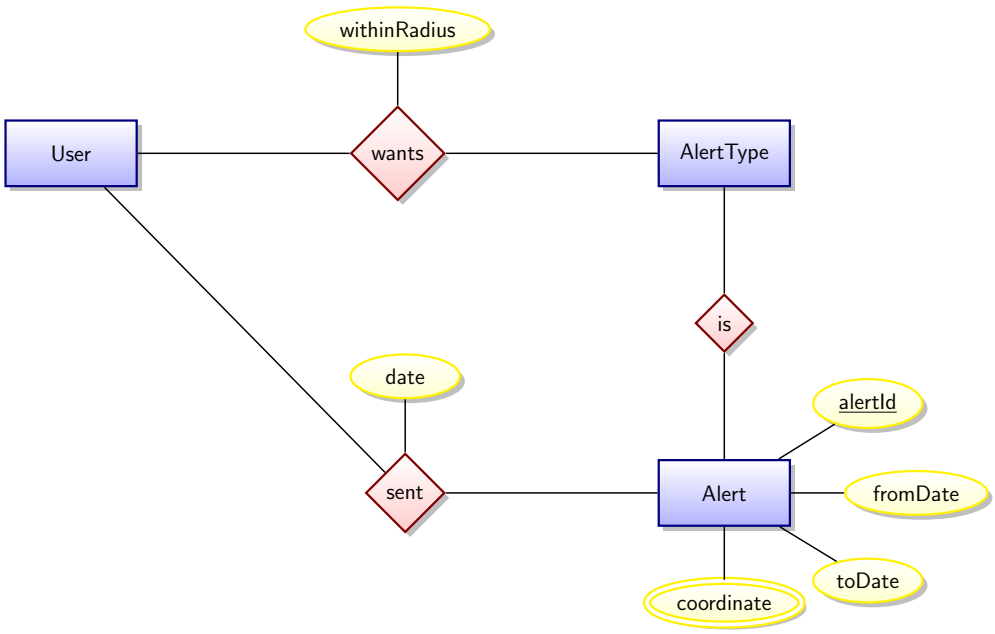

Figure 17: A data model of alerts.

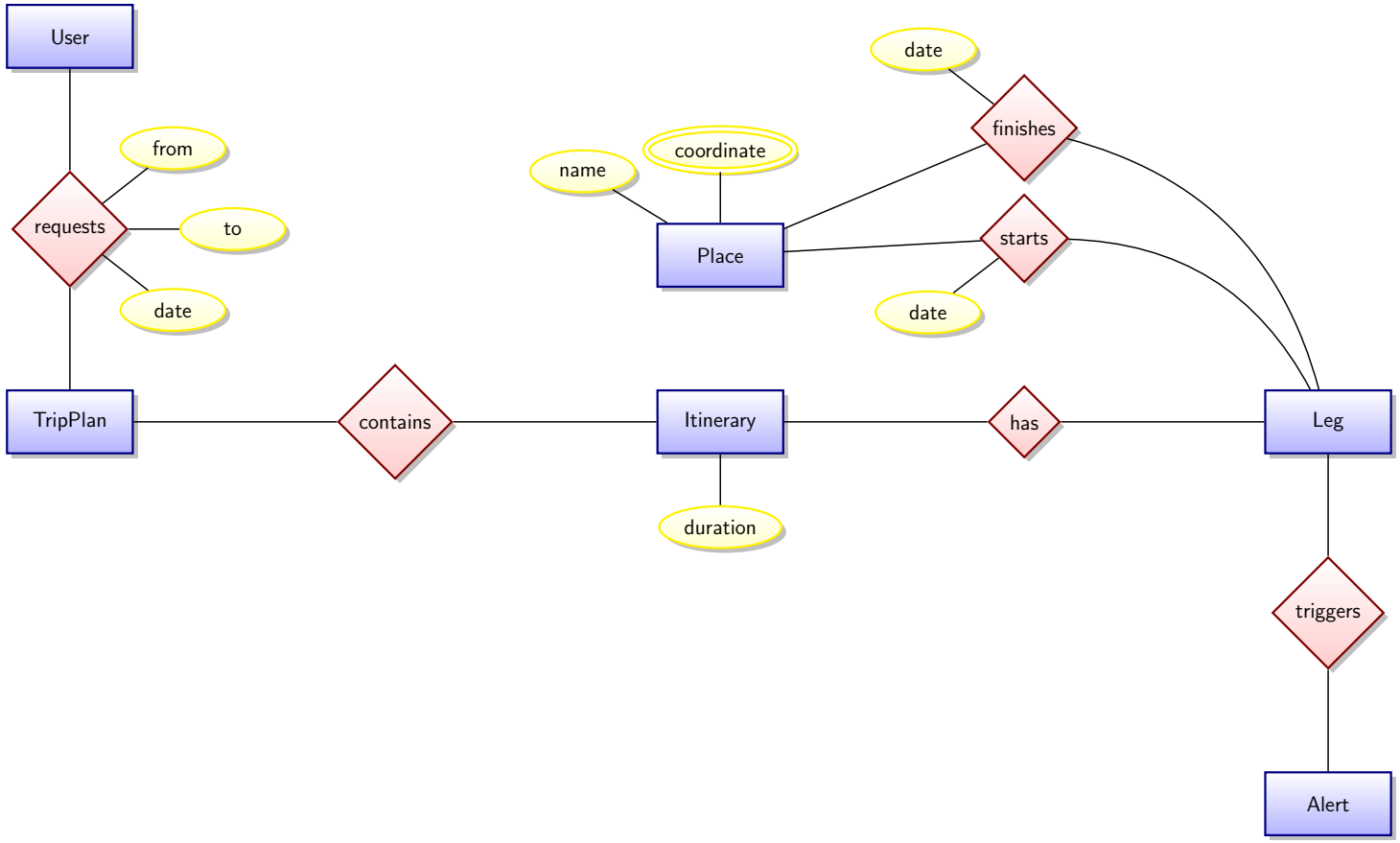

Figure 18: A data model of trip-planning.

4.2.4 Actions (SCADA)

The system may also issue *actions* to remotely controllable devices. In the city-scale traffic scenario, this might involve changing the pattern of traffic lights in order to prevent congestion bottlenecks. In other scenarios such as the nationwide emergency use case, the role of actions may be limited or absent, since the system may be intended purely for monitoring, co-ordination and information-provision purposes. Our model of actions follows the data model suggested by standards in supervisory control and data acquisition (SCADA), notably IEC 60870. Each device is assigned an address, which can be an IP address, and a port, which can be 2404, as per IEC 60870 part 5, 104. The action is embedded in an application service data units (ASDU), which has a rich structure (type identification, variable structure qualifier, cause of transmission, common address, information object address, information element, time tag, if used), as specified in IEC 60870 part 6. Notably, we aim to use single commands (SCO), such as on or off, and set point commands, which carry a floating point number. See the entity-relationship diagram in Figure 19.

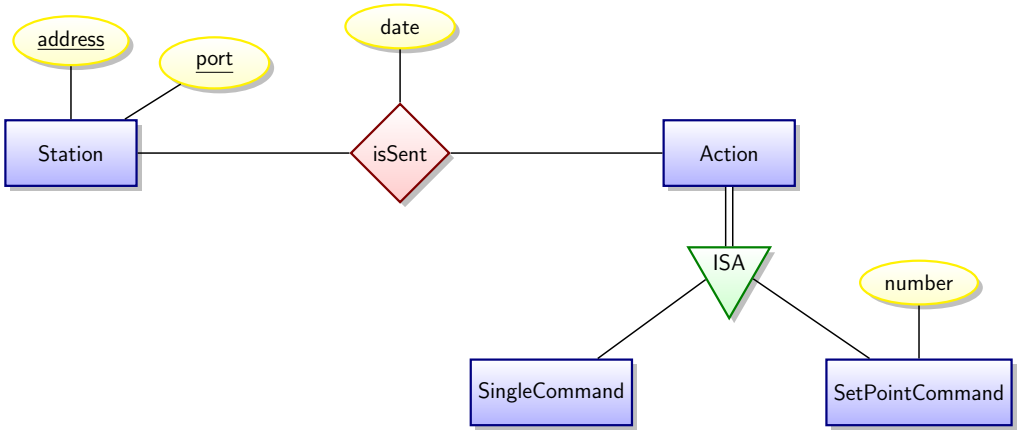

Figure 19: A data model of actions.

5 Analysis

Whereas previous sections described the use case scenarios and the derived tasks, this section focuses on first analyses and ideas to tackle the tasks. At this point we do not structure by the use case scenarios separately but by the performed tasks, as the tasks in the scenarios are similar. The data samples have been provided only recently, thus most analyses are work in progress. However, the following brief presentation highlights our progress.

5.1 Geo-coding Live Drive Radio

Resulting from a Twitter stream, the Live Drive Radio data items (compare Section 4.1.5) describe geographic information without exact WGS84 coordinates.

A method for geo-coding this data set was developed by IBM [DLB13] which utilizes word occurrences and word frequencies. After a data cleaning, i.e. removal of punctuation and

extension of abbreviations, location based words are extracted. Next, these words are used to construct N-grams (tuples of possible word combinations). These N-grams are weighted and combined to a Lucene query [HHS07]. As result, terms that identify the location are returned. The lookup of coordinates for these terms provides the coordinates for the text message [DLB13].
The results of the analysis are available to the consortium within the last three columns in the data file[11]:

- *search output*: the spatial identifier (e.g. Kilmacud Road Lower),
- *wgs84 coordinates*: the position,
- *explanation*: type of the hazard (e.g. collision).

Though this method is evaluated in [DLB13], and performs well even in difficult cases, e.g., the message referring to a jam close to a junction in Figure 20. Future work will improve the geo-coding further.

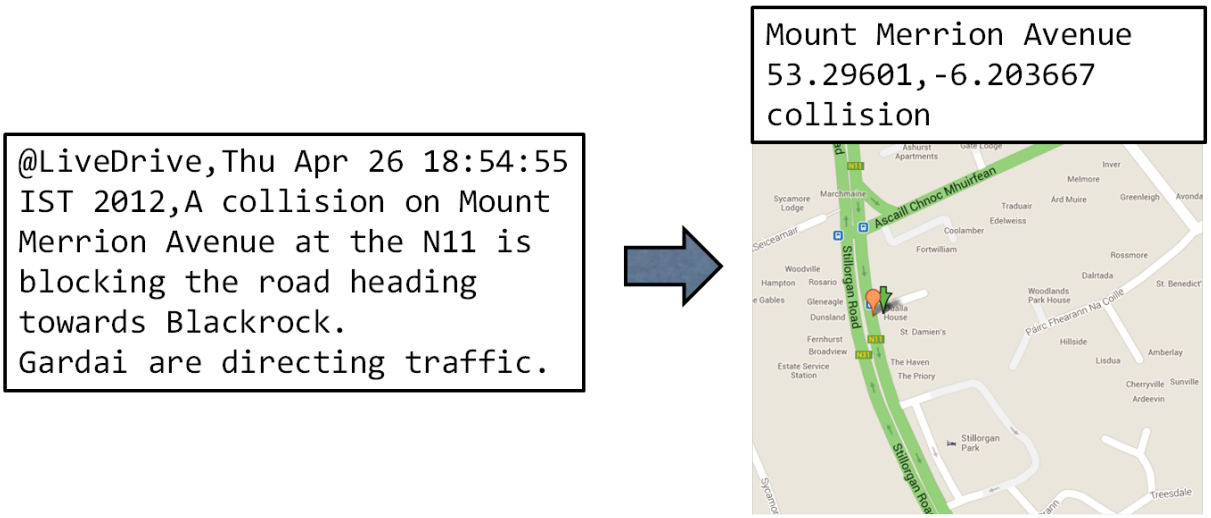

Figure 20: Exemplary Query (left) and Result (right) of Geo-Coding. The corresponding junction to the message is correctly identified.

5.2 Extraction of Normal Behaviour from Twitter Users - a Visual Analysis approach

In contrast to a visualization of analysis results, visual analytics combines interactive visualizations with automated analysis techniques. Thus, visual analytics focuses on the division of labour between humans and machines for the following reasons:

[11] available at:
svn+ssh://madgik/svn/source/LiveDriveRadioMessages.txt

- computational power amplifies human perceptual and cognitive capabilities,
- visual representations are the most effective means to convey information to a human's mind and prompt human cognition and reasoning.

The process is facilitated by the "Visual Analytics Loop" [KAF+08] that applies visualization and models as well as their entanglement to data in order to derive knowledge which, in turn, leads to novel data (see Figure 21).

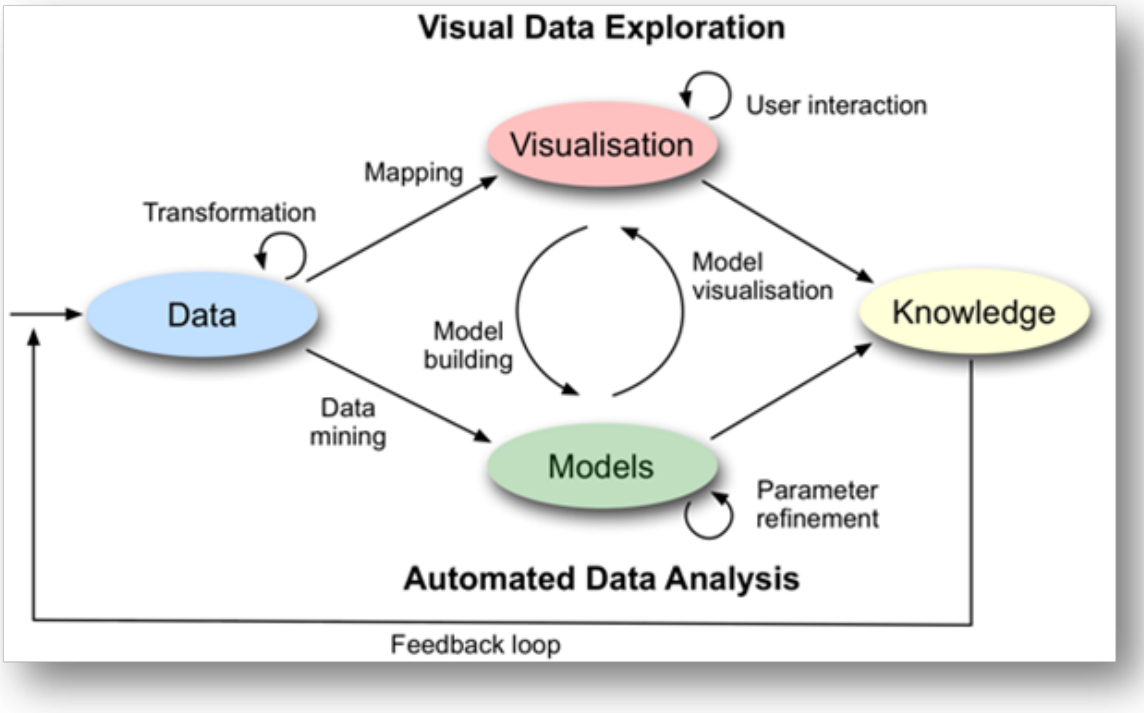

Figure 21: Visual Analytics Loop [KAF+08].

In recent work, Fraunhofer IAIS successfully applied visual analytics methods to Twitter messages in Seattle [AAB+13]. This analysis workflow has been applied to geo-coded Irish Twitter messages from 05/11/2012 till 25/04/2013 in order to detect daily routines of the Twitter users. The considered data set contains 5,213 Twitter users generating 1,637,346 position records. The trajectories of these users are plotted in Figure 22.

In a first step, personal places were extracted for each trajectory separately. In order to guarantee privacy, clusters of the messages were created with a maximal radius of $150m$, clusters with less than 5 elements were removed. For every cluster polygons were constructed as the convex hull of the points in conjunction with a small spatial buffer (since some hulls were no polygons, but points or lines). The resulting personal places are depicted in Figure 23. In the following the distribution of the number of personal places per person is inspected. As Figure 24 shows, the majority of people (1,158) has two personal places.

In a subsequent step, the messages and their spatial distributions are also considered. Thus, topics are defined as list of related words (compare Figure 25). Using these lists, each

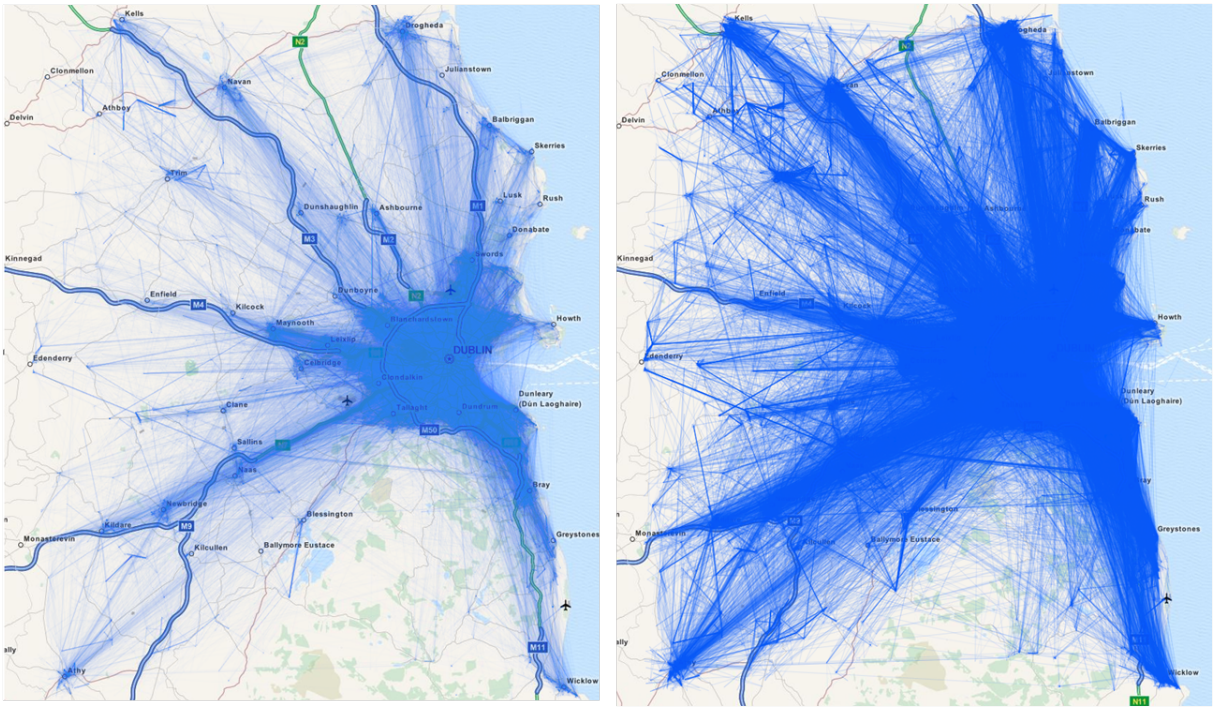

Figure 22: Trajectories of Twitter users plotted with 99% transparency (left) and 95% transparency (right).

message that belongs to a personal place (14.3% of all messages) is assigned to one or even multiple topics. Thus, message counts per topic can be derived, see Figure 26. In result every personal place is connected to a distribution of topics, these can be visually inspected, see Figure 27. However, some places did not get topic summaries (about 20%) and topics are very much mixed. Moreover, the topics are not necessarily representative for the type of the place, e.g., the topics near a supermarket: family, education, work, cafe, shopping, and many others.

After the spatial distribution of the messages, the temporal distribution has been inspected. Figure 28 depicts the number of messages aggregated per day time hour (abscissa) and location (grey connecting lines in the chart). The aggregates are computed for working days and weekend separately.

In order to extract some semantics of the locations from the temporal profile, these profiles are compared to exemplary temporal profiles of different activities using dynamic time warping [Sen08]. The exemplary profiles can be extracted from the previously generated topics, see Figure 29. In result, the places with a high similarity to a particular topic can be extracted, for example the locations identified to be similar to "breakfast" are also highly similar to the topic "transport", and "lunch". This indicates that dynamic time warping is not ideally suitable for comparison with the exemplary time profiles, in contrast it is required to limit stretching and shifting in the comparison. This is subject to future investigations.

In summary, currently available tools did not achieve good results in semantic interpretation of personal places extracted from Twitter data. Necessary improvements are a better similarity function for time series, a more comprehensive ontology for recognition of message

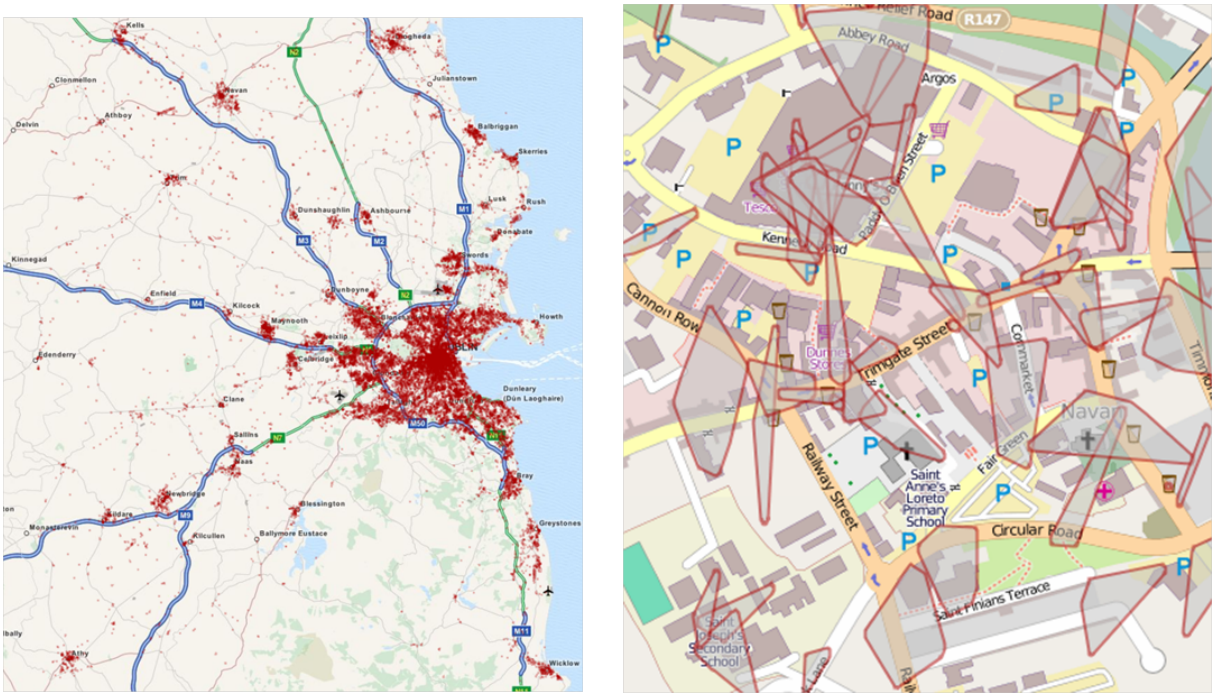

Figure 23: Extracted Personal Places from Twitter messages with different Zoom Factor.

topics and a better computational mechanism for combining time series similarity scores with the topical features. Future work may just focus on public places, since we expect that topics are easier to interpret, for example we could focus on *foursquare check-ins*. We will also take into account additional data sources, e.g. relative positions of places in daily trips or land use.

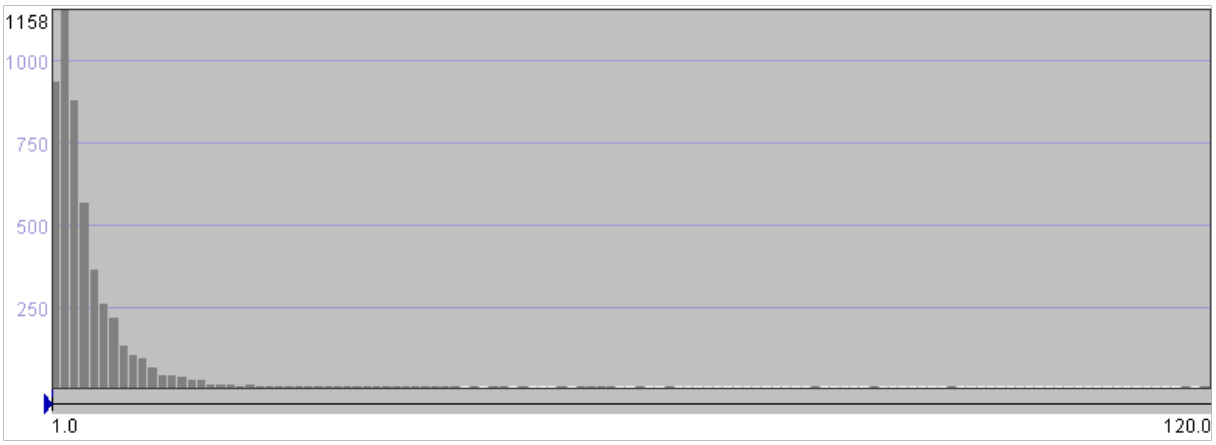

Figure 24: Number of Personal Places per Person. The ordinate axis denotes the count of persons having exactly the number of personal places denoted at the abscissa.

Topic	Related words
family	mother;mom;mommy;father;dad;daddy;kids;children;son;sons;daughter;daughters;brother_;sister_;relatives;uncle;aunt;husband;wife;spouse;boyfriend;girlfriend;
home	
education	study_;learn_;school_;college;university;teacher;class;classes;graduate;undergraduate;lecture;lesson_;exam_
work	working;workplace;"work place"
local transport	bus;metro;metrobus;tram;trolleybus;luas;"light rail";lightrail
regional transport	rail;railway;train;ferry;transit
far transport	airport;_airport;terminal;flight;boarding;airplane
eatery	cafeteria;deli;"snack bar";canteen;lunchroom;"lunch room";luncheonette;"sandwich shop";"sandwich bar";"business lunch"
restaurant	bistro;steakhouse;pizzeria;grill;grillhouse;"dine out";"dining out";cuisine
pub	brewery;winer_;bar;barroom;tavern;cabaret;nightclub;"night club";discotheque;nightery;"night spot"
cafe	coffee shop;"coffee bar";"tea room";tearoom
shopping	shop_;store
daily shopping	grocer_;supermarket;baker_;dairy;butcher;market;mart;"food shopping";"food store";confection_
other shopping	mall;outlet;boutique;"department store";"fashion store";"liquor store";"book store";"designer shop";"high street";"main street";retail;antique;jewelry
services	barber_;haircut_;hairdress_;manicur_;pedicur_;mani;pedi
health care	hospital;doctor;doc;physician;pharmac_;medic_;therap_;heal_;dentist_;hygiene;acupuncture;chiropract_
fitness	workout;biking;bike;cycling;cycle;jogging;jog;swim_;yoga;gym_;pilates;hike;hiking;exercis_;bodybuilding;aerobic_
wellness	massag_;sauna;bath;bathhouse;hammam;"aroma therapy";spa
nature	forest;woods;seaside;river;picknick_
culture	theatre;cinema;museum;gallery;art;concert;exhibition
sports	sport_;football;soccer;baseball;tennis;golf;basketball;voleyball;bball;canoe;rugby;stadium
friends	friend;schoolmate_;buddy;buddies
game	gaming
public event	parade;fest;festival;conference
private event	jubilee;wedding;party;birthday;bday
alcohol	wine;beer;cocktail;vodka;tequila;brandy;liquor;gin;whiskey;spirits;cognac;ale;cider;stout;lager;draught;guinness;guiness;kilkenny;mead;poitin;alco_
food	meal;hungry;eat;eating;lunch;breakfast;dinner;supper;bread;milk;jogurt;butter;egg;eggs;snack;steak;burger;sandwich_;soup;chowder;salad;meat;fish;seafood;chips
sweets	dessert_;cake;cakes;"ice cream";candy;candies;chocolate;shake;sundae;cookie_;pancake_;cheesecake_
coffee	espresso;capuccino;frappe;latte;mocca;mocha;cafe
tea	chai

Figure 25: Topics and Related Words.

	Message	Features	topic=family: Occurrences of topic	topic=home: Occurrences of topic	topic=education: Occurrences of topic	topic=work: Occurrences of topic	topic=local transport: Occurrences of topic	topic=regional transport: Occurrences of topic
658174	@joe_lennon I usually	education	0	0	1	0	0	0
658175	@joe_lennon together	education	0	0	1	0	0	0
658182	@jas_103 deadly; don	work	0	0	0	1	0	0
658183	Just got home and see	home	0	1	0	0	0	0
658186	So excited about my ne	sweets	0	0	0	0	0	0
658189	@iamtcdizzy I haven't b	shopping	0	0	0	0	0	0
658192	Get in from my night ou	family;home;work	1	1	0	1	0	0
658193	Home again at 6pm! N	home	0	1	0	0	0	0
658206	Bussing it home for t	Get in from my night out; my dad gets home from work two minutes later. Great timing :)		1	0	0	0	0
658210	Ah shite. It's been a g			0	0	0	0	0
658212	@ronanhutchinson be	education	0	0	1	0	0	0
658213	Dinner over and Star W	food	0	0	0	0	0	0
658221	@Sean_ODulaing drin	friends	0	0	0	0	0	0
658222	@iamtcdizzy I'm on the	local transport	0	0	0	0	1	0
658236	Simpsons on the telly,	culture	0	0	0	0	0	0
658239	@iamtcdizzy: Got a love	education	0	0	1	0	0	0
658240	@beccimoo @mimsyn	game	0	0	0	0	0	0
658244	@iamtcdizzy ah if ya ha	local transport	0	0	0	0	1	0
658248	Alarm set, back to (coll	education;work	0	0	1	1	0	0
658249	Have an exciting non-c	education;work	0	0	1	1	0	0
658250	Happy birthday @Siobl	private event	0	0	0	0	0	0
658258	@iamtcdizzy @beccim	family	1	0	0	0	0	0
658269	@niamh_leonard yeh	private event	0	0	0	0	0	0
658280	People get into 'are you	shopping;game	0	0	0	0	0	0
658283	Fair play to @craigdoyl	sports	0	0	0	0	0	0
658305	Terribly sad news on th	sports	0	0	0	0	0	0
658311	Home alone time #and	home	0	1	0	0	0	0
658315	@Cunneen92 @white	sports	0	0	0	0	0	0

Sort by: No selection Ascending TableLens condensed Attribute...

Topic	Frequency in text
food	35684
education	33186
family	31791
work	21012
home	14498
private event	12005
friends	10778
sweets	7863
sports	7646
game	7486
fitness	7187
alcohol	6749
shopping	5640
health care	5437
local transport	4164
tea	3896
pub	3642
culture	2631
coffee	2500
public event	1906
far transport	1722
wellness	1617
regional transport	1357
services	1079
daily shopping	1057
nature	1040
restaurant	613
cafe	355
other shopping	326
eatery	284

Figure 26: Message Topic Distribution.

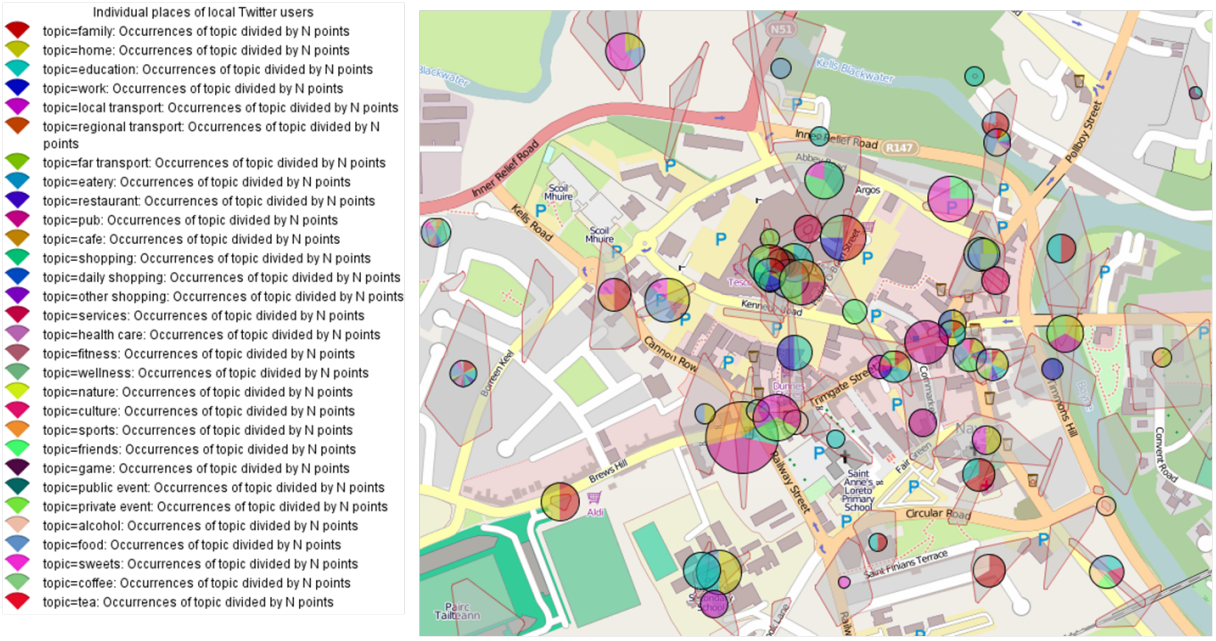

Figure 27: Topics Summarized by Personal Places.

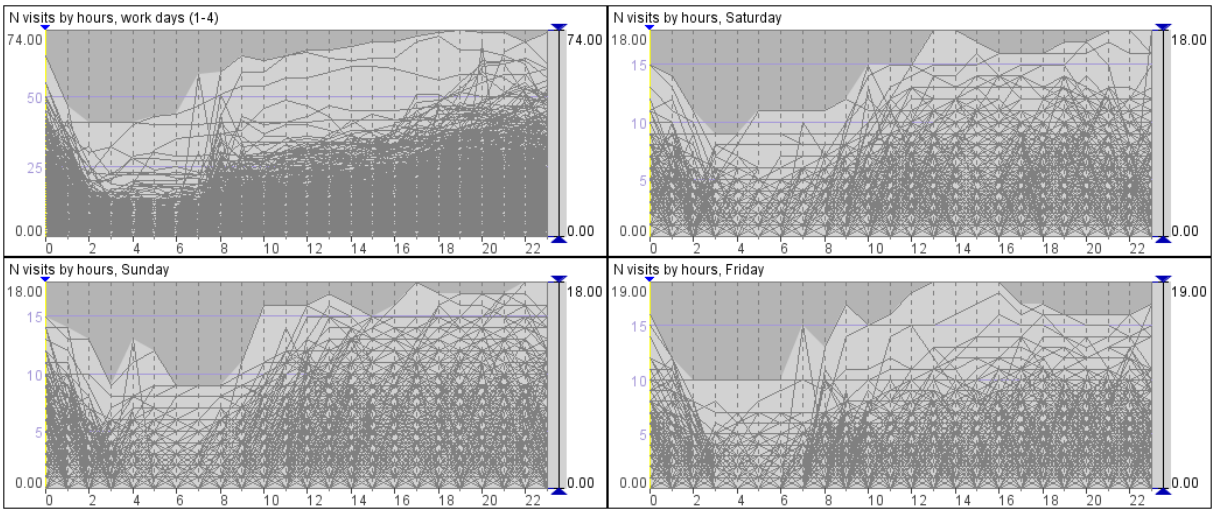

Figure 28: Temporal Message Distribution.

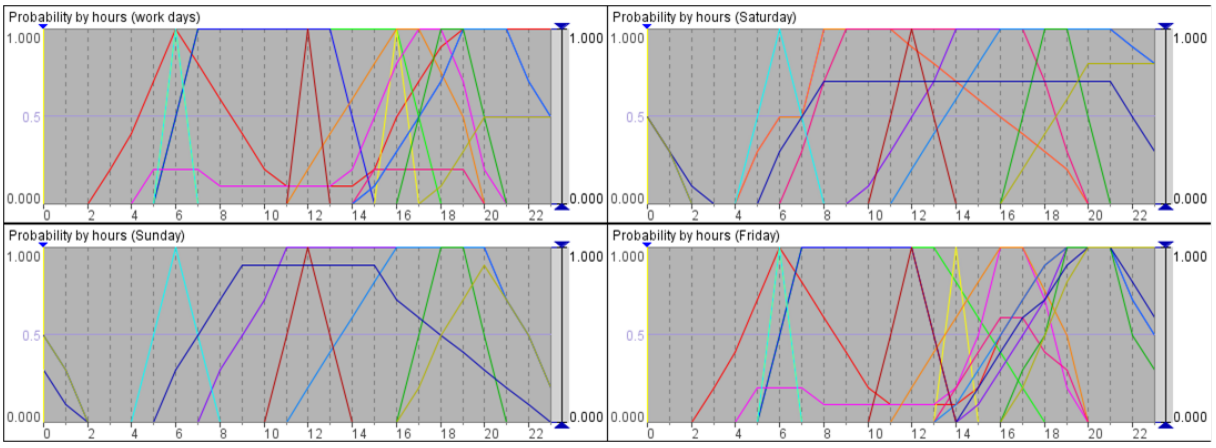

Figure 29: Exemplary Temporal Profiles of Topics.

5.3 Location Analysis of Twitter Users in Dublin

For event detection and disaster confrontation it is of significant importance to utilize the location information of Tweets. However, only a small portion of Twitter users declare their location and only a tiny fraction of tweets are geo-located. In this section we[12] present a preliminary analysis of the Twitter user behaviour in the context of location information.

We have obtained tweets posted in the geographical area of Dublin and collected all the users who tweeted. We created the "friendship graph" of these users, considering as friends users that are bi-directionally connected (both are followers of each other). We analysed the location field of users who tweeted, to see whether they report Dublin as their location (See Figure 30).

The first column represents the whole set of users, while the second one represents the largest connected component of the graph. The third column represents the users who have more than 5 edges in the graph, while the two last columns depict users who have tweeted more than N times, in a period of T hours. In Figure 30 we count as "Dublin", users who have the word "Dublin", or areas from the city in their location field. "Null" simply means that these users left their location field empty.

The next figure (Figure 31) represents the same data, but for all the users that tweeted from the Dublin area according to the approximate geo-location (bounding box). Both figures exhibit a similar behaviour, in which we can clearly observe that the percentage of users outside of Dublin clearly decreases if we implement some very simple heuristics. The most interesting observation is that the percentage of users with empty locations also decreases for the same heuristics.

For the above study, we have extracted tweets from the area seen in Figure 32. In Figure 33 we can observe the same map organized in squares where the value (height-colour) in each square represents the number of tweets. The similarity between the two maps (Virtual based on Twitter and Natural) is obvious. The sea and mountains can be clearly distinguished (absence of tweets). Squares represent an area of 250×250 meters.

The outlier in the middle (brown bar) is due to a user that is a fan of the music group "One Direction". In 90 days, out of the 14000 tweets in the area, this account posted approximately 12000 tweets, some of which were almost duplicates, mostly trying to get the band members to follow her.

Finally we have performed a preliminary lexicon-based sentiment analysis. In this case the height / colour of the bars depends of the degree of sentiment that we obtained from that particular area square (see Figure 34).

There are a few clear outliers here as well. The red one is the same fan as before, and the yellow one is a similar user. Tweeting mostly about their admiration for the band, they significantly contribute to the positive sentiment of the area. The huge negative spike (below the surface) is due to a weather service, which tweets several times a day about the weather prediction. They use the following format to report the weather:

```
07:55 BST: Temp: 11.0C, Wind: 1 mph (ave), 5 mph (gust), Humidity: 91\%,
Rain (hourly) 0.0 mm, Press: 1018 hPa #iwn
```

[12] Section provided by UoA

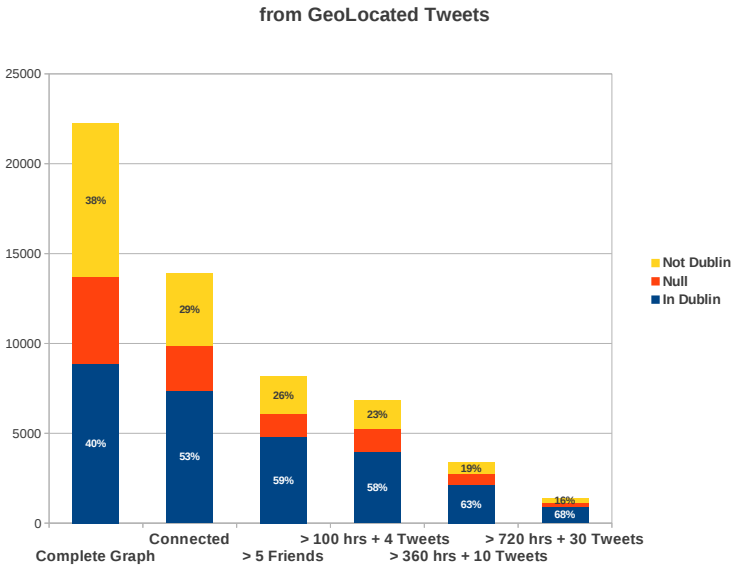

Figure 30: Location Statistics from geo-located Tweets of the Dublin Area

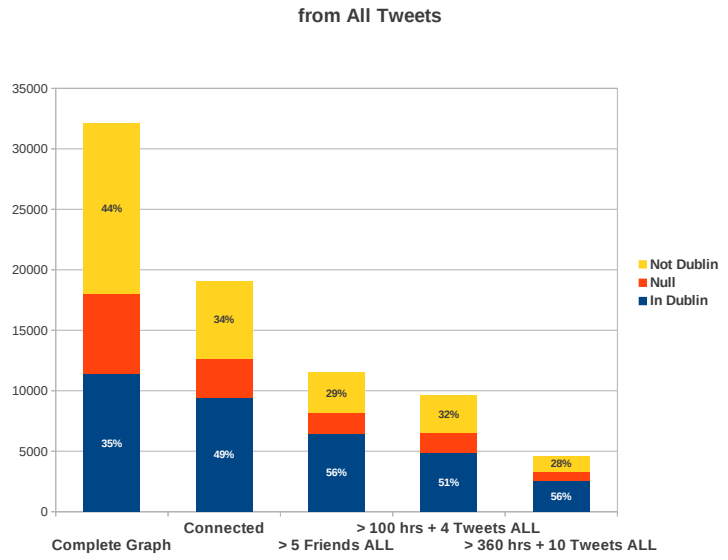

Figure 31: Location Statistics from All Tweets of the Dublin Area

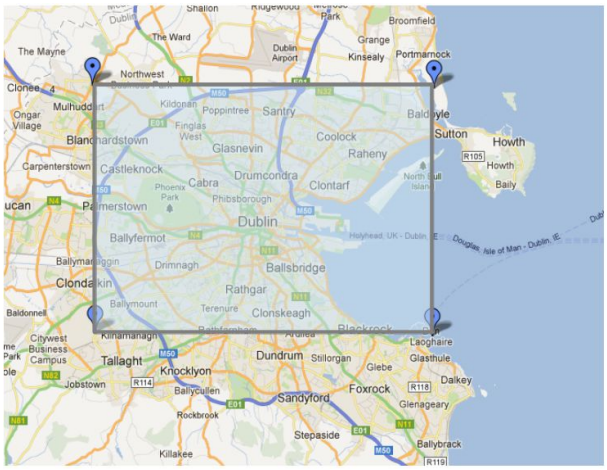

Figure 32: Bounding Rectangle for Twitter Data extraction in Dublin

The tweet presents negative sentiment since it contains the words "rain", "wind" and "humidity" that the lexicon considers as negative words.

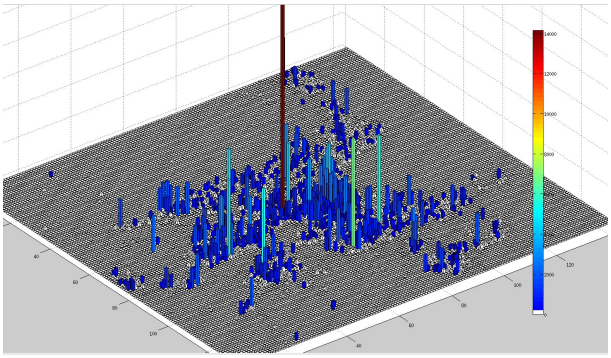

Figure 33: Twitter Map of Dublin. Histogram depicts counts of messages per 250×250 meters grid cell.

As a conclusion we could state that the noise in such data is a very important impediment since users could create a large amount of content that could negatively affect results and interpretations. Therefore, it is important to include this topic to the INSIGHT's research agenda.

5.4 Extraction of Land use from Stationary Twitter Messages

It is crucial to create a model which represents fluctuations of people's presence over time for every location in normal situations. This knowledge supports detection of deviations from normality and provides primer knowledge on people's motivations in case of traffic jams or flooding events. The information on fluctuations of people's presence for every location reflects the typical land use per location. A normal land use is for example the use of a location for working during day times or for clubbing at night.

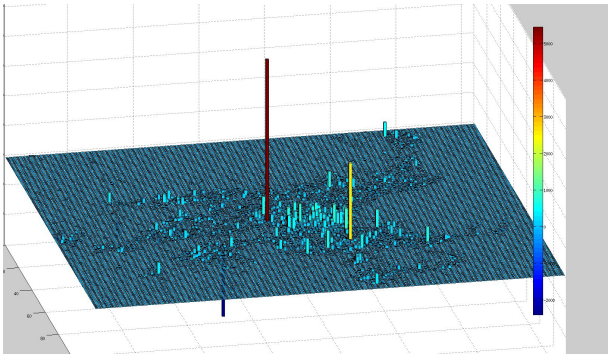

Figure 34: Twitter Map of Dublin based on Sentiment.

Recently Fraunhofer IAIS together with TUDo jointly worked on the extraction of land use patterns from geo-referenced Twitter messages [RL13]. We focussed on a subsample of the Twitter messages which were created by foursquare users. The messages are generated every time a foursquare user checks itself in at a previously defined public location (e.g. a bar, a company, a landmark). Though the messages are spatio-temporal events, the set of all messages provides a spatio-temporal time series. Temporal profiles of these check-ins are easily extractable as the locations (so-called venues) are limited to the locations of the venues and are not distributed arbitrarily in space as all Twitter messages are.

The problem is to identify and cluster similar venues based on their temporal profiles and spatial distance. We propose to utilize spectral clustering and thus deliver arbitrarily shaped clusters. Evidence accumulation clustering is applied to deduce the parameters directly from the data.

In a first step, the messages are aggregated per hour and venue resulting in a discrete temporal distributions of the check-ins per venue. These profiles are compared using an affinity measure that compares pairwise the evolution of the time profiles (all subsequent elements). With every similarity among two time series 0.5 is added to a similarity score, with simultaneous increase or decrease of the series even 1.0 is added to that score. The comparison of a venue's temporal profile with its spatial neighbours generates an affinity matrix, which contains the pairwise similarities among the locations. This is subject to spectral clustering with parameter k. Since this parameter is unknown, we use evidence accumulation clustering [FJ02] with uniformly at random drawn $k \in (k_{min}, k_{max})$.

As a result, we obtain spatially contiguous sets of venues with similar temporal profile. For visualization, we construct the convex hull of each cluster and map similar colours to similar temporal profiles and different profiles to different colours [Sam69]. The tests are performed in the city of Cologne. The data was collected from May 2012 till October 2012. All public check-ins within this period were used. The final data set for Cologne consists of 11,890 check-ins from 2,093 users over more than 1,008 venues. The result of this test is depicted in Figure 35.

Future work investigates the detection of events using spatio-temporal clustering of time series and the applicability to other data sources, e.g., the traffic counts provided in SCATS. Also the consideration of larger spatial areas is necessary for the nation-wide use case.

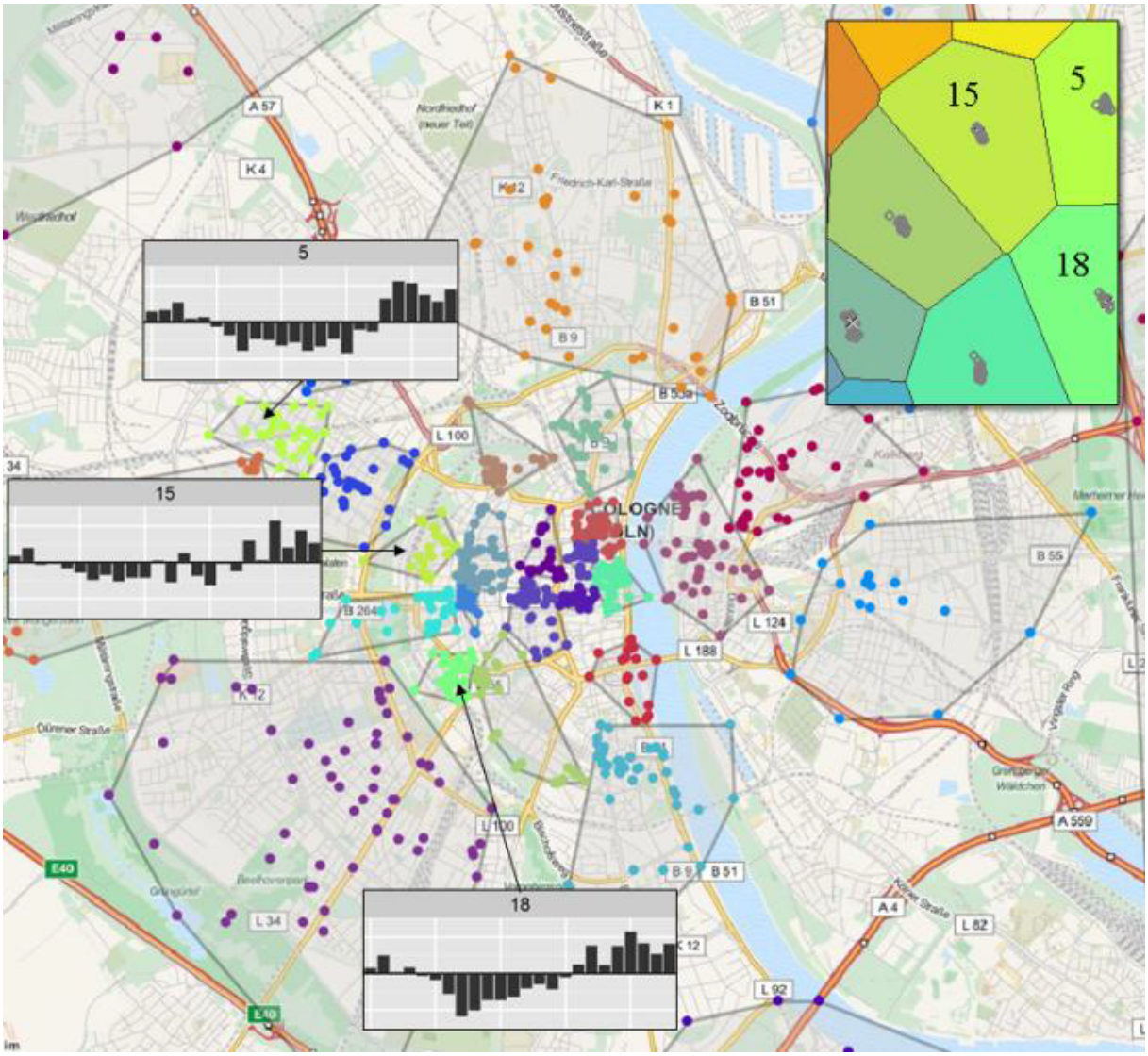

Figure 35: Land use Clusters derived from Foursquare Messages for the City of Cologne, Germany [RL13].

5.5 Traffic Quantity Estimation with Movement Patterns

In order to model normal situations, it is not only important to identify regions or time slices with similar behaviour (as previous sections focus on), but it is also important to impute values for unobserved locations.

This task is particularly crucial for the traffic loop data (SCATS), which is only available for few dedicated places within the city. Available data for investigating the traffic flow also at unmeasured locations are the measurements (SCATS) and some prior knowledge, e.g., the city's traffic network and knowledge on preferred routes by local domain experts (referring to crowdsourcing in WP3).

Recently developed approach by Fraunhofer IAIS and TUDo [LXM13, LXMW12] proposes a nonparametric Bayesian method, Gaussian Processes (GP) with a random-walk based tra-

jectory kernel. We explore not only the commonly used information known from the literature, e.g. traffic network structures and recorded counts at some measurement locations, but also the recorded movement sequences are considered.

So far, the method was successfully applied to industrial scenarios with pedestrians in closed environments. During development of the proposed prototype (Section 6.1) we will exploit usage of this method for the city level use case incorporating crowd sourced movement behaviour.

5.6 Exploration of the Geo-coded Twitter Messages from Germany

The dataset Geo-coded Twitter messages cover the time period 22 November 2012 till 24 July 2013 (compare Section3.1.1). The data are collected by querying the Twitter API for tweets with geographical coordinates located within a given bounding rectangle. After the collection, the tweets have been additionally filtered by checking the country code and whether the coordinates are indeed within the bounding rectangle of Germany, since the collected data also contained messages from the Netherlands and other countries as well as messages with the country code of Germany but located outside Germany (which means that the Twitter API may not provide exactly what you request). The final dataset that has been explored consists of about 6 million records. This dataset could be extended further using the technique provided by UoA described in [VG12]. They tackle the problem of extracting location information, usually referred to as geocoding, from additional user-provided content.They rely on software and data which are available online and publicly accessible.

The aim of the investigation at-hand[13] is to check whether disasters can be detected from the Twitter data. In particular, we investigate how a known disaster, namely, the floods of June 2013, is reflected in the Twitter data.

We conduct the analysis in two experiments: (1) we attempt to detect flooding disasters based on the number of tweets in a place (2) we explore selected Twitter messages containing relevant terms.

5.6.1 Experiment 1: Attempt to Detect Disasters based on the Number of Tweets in a Place

In the first experiment we apply following hypothesis: A disaster event, such as flood, causes increase in the overall number of tweets posted in the place where the disaster happens. This can be used for detecting places of disasters.

Data preparation

The territory of Germany has been divided into compartments (Voronoi polygons [Vor08], compare Section 2.3). For this purpose, a random 2% sample of the tweets has been taken from the database. The points have been grouped into convex spatial clusters, and the centres of the clusters have been taken as generating seeds for the Voronoi tessellation. The resulting division is shown in Figure 36.

[13] Section provided by Fraunhofer IAIS

The entire dataset has been aggregated by the spatial compartments and daily time intervals: for each compartment and day, the number of tweets has been counted. The result is a set of time series of tweet counts, each time series of the length 244 days is associated with a certain cell (compartment) of the territory division. The time series are visually represented in a time graph (line chart) in Figure 37.

Exploration of the spatial time series

The most prominent peaks (i.e., sharp increases in the number of tweets) can be easily detected by viewing the time graph. To interpret the meaning of the peaks, we have extracted the most frequent words and combinations from the tweets posted in the respective cells and time intervals, setting the minimal frequency threshold to 5. The results showed us that these peaks do not correspond to disaster events (fortunately). Thus, the highest peak, up to 2402 messages per day, occurred in Berlin from 6 to 8 May 2013. Figure 38 shows the most frequent words and combinations occurring in the messages posted in Berlin in this time interval. The frequency champion is "re publica", which occurred 511 times on May 6, 475 times on May 7, and 394 times on May 8. Using Google, we have found out that "re:publica" was a European conference on social media, blogs, and digital society that took place in Berlin in this time period. The next highest peak (1046 tweets) happened in Bochum on 24 November 2012 (and 651 tweets were posted in this place on the next day). Among the most frequent words, there is "Ruhrcongress Bochum" (which is a congress centre in Bochum). From the web, we have learnt that on November 24-25, 2012 there was a meeting of the German party Piraten (Piraten-Parteitag) in the congress centre in Bochum. The word "#piraten" occurs among the frequent words with the frequencies 32 on November 24 and 19 on November 25. Among the frequent words, there is also "Bermudadreieck" (Bermuda Triangle) with frequency 21, which is "the designation for an area with a high density of bars and restaurants" (Wikipedia). The third highest peak occurred in Hamburg on December 27-29, 2012. It was caused by the Chaos Communication Congress organized by the Chaos Computer Club the largest European organization of hackers.

To find out whether there are any peaks that can be attributed to the flood events of June 2013, we have specifically looked at the parts of the time series for the time period 25 May 01 July 2013 (Figure 39). It can be observed that there are no prominent peaks in this period. Knowing that the area of Dresden was affected by the flood on the Elbe River, we specially looked at the time series of the cells covering Dresden centre and suburbs. The time series do not show any peaks. We have noticed a small local maximum in the number of tweets in the centre of Dresden on June 3 (72 tweets compared to 32, 48, and 61 in the previous days and 57 in the next day) and looked at the most frequent words and combinations for this place and day (Figure 40). The flood-relevant word "#hochwasser" occurs among these words (it is marked with a black frame in Figure 40), but the frequency is only 9 while the highest frequencies are 61 for "#linkebpt" and 41 for "Dresden". Also frequent are the word "Neustadt" (20) and the phrase "Bunte Republik Neustadt" (19), which refer to one of the districts in Dresden.

We have also tried to find potentially interesting events by comparing the values in the time series with the mean values for the previous 14 days (Figure 41). The time graph of the differences shows very many peaks. For the time period 25 May 1 July 2013, have extracted the frequent words and combinations for the cells and days where the differences

to the 14-days means were 100 or more. There are 138 such pairs (cell, day). The respective sets of frequent words and combinations do not contain the term "Hochwasser" at all.

Conclusion

Our hypothesis was that a disaster event may cause a noticeable increase in the number of tweets posted in a disaster-affected area. Based on our experiment, we have to reject this hypothesis. We have seen that high increases in the numbers of tweets are mostly caused by public gatherings, such as conferences, but not by disaster events. (Moreover, the gatherings generating high peaks consist of quite specific public: people interested in social media, computed hackers, "pirates", etc.) We have also seen that the sets of tweets posted in places that are known to be disaster-affected may contain too few occurrences of relevant terms; hence, relevant messages can be easily lost in the bulk of posted tweets. This means that, in order to detect possible disasters, it is necessary to look specifically for tweets containing relevant terms. For this purpose, a vocabulary of relevant terms and related words needs to be created.

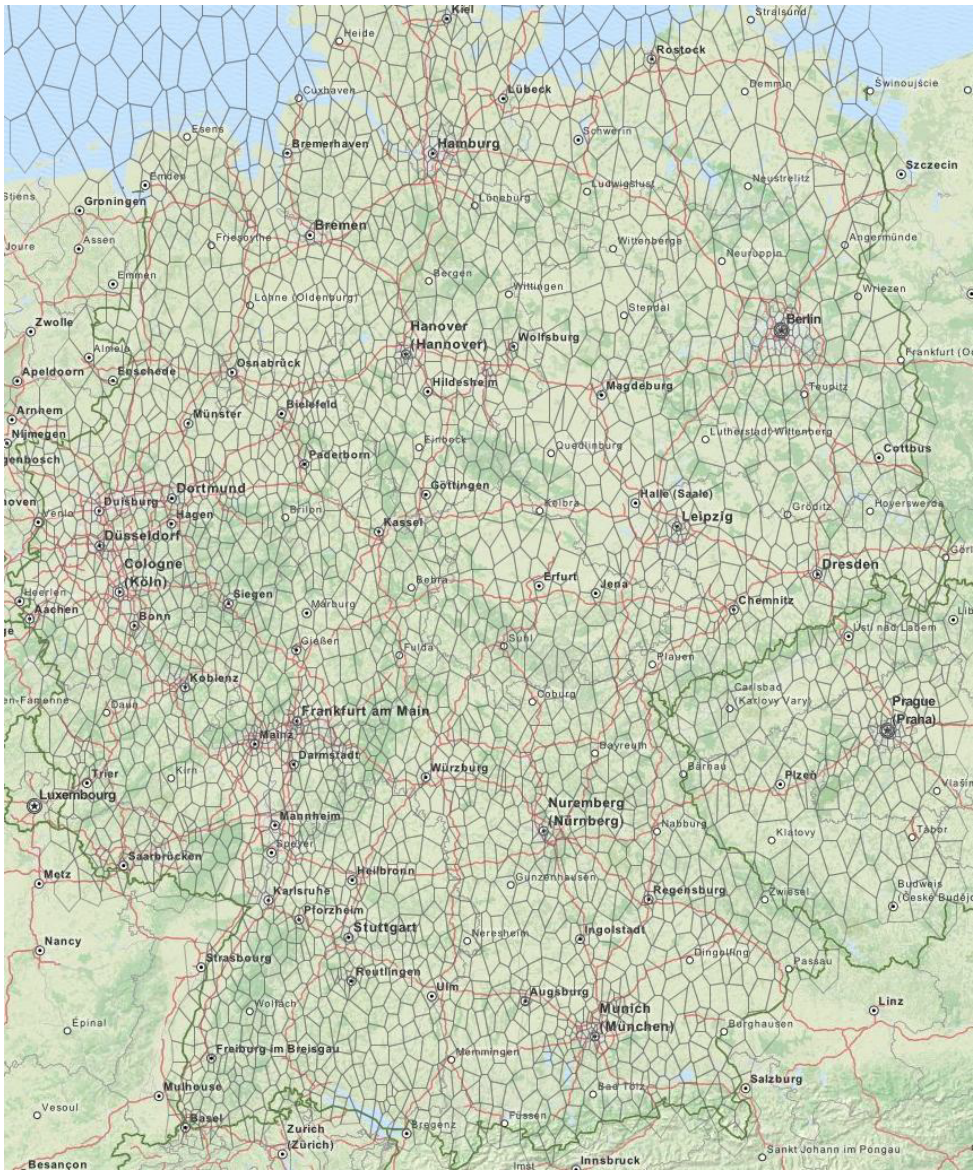

Figure 36: The Division of the Territory Covered by the Investigated Dataset into Voronoi Polygons.

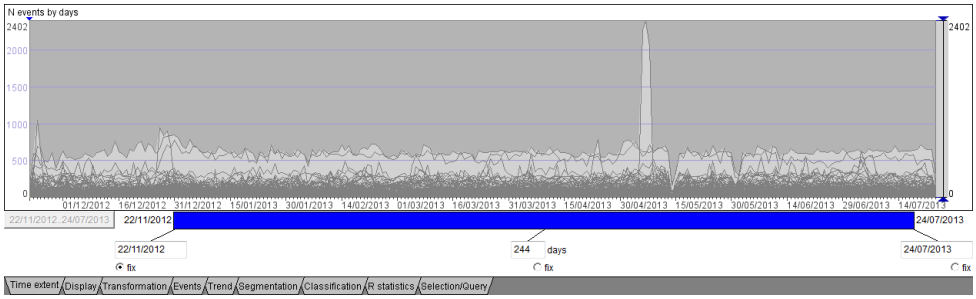

Figure 37: A Time Graph Shows the Time Series of the Tweet Counts by the Spatial Compartments.

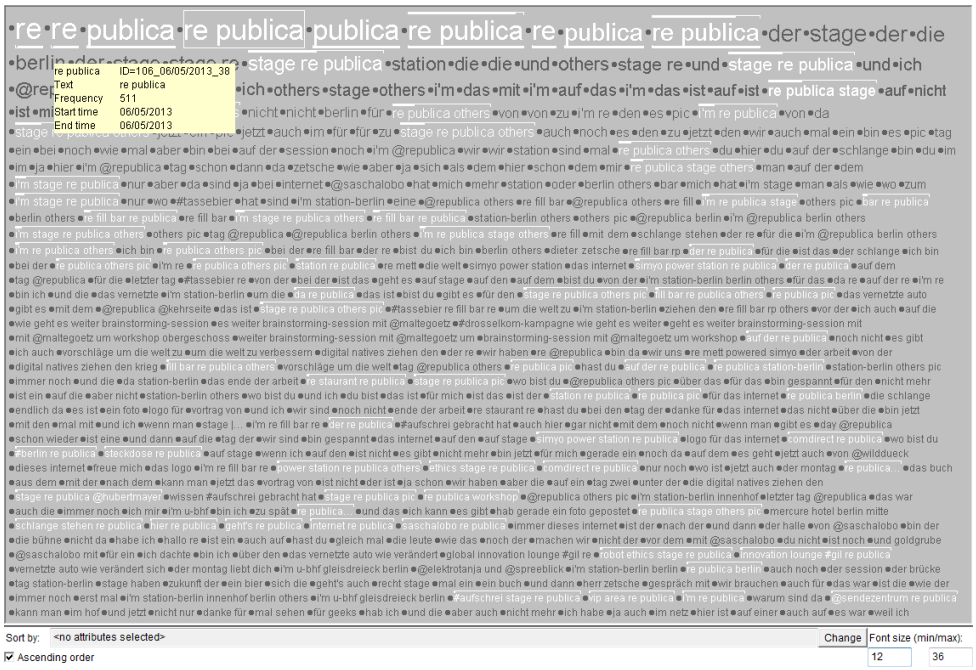

Figure 38: The Most Frequent Words and Combinations Occurring in the Tweets Posted in Berlin in May 6-8, 2013. The Font Size is Proportional to the Frequency of a Word/Phrase.

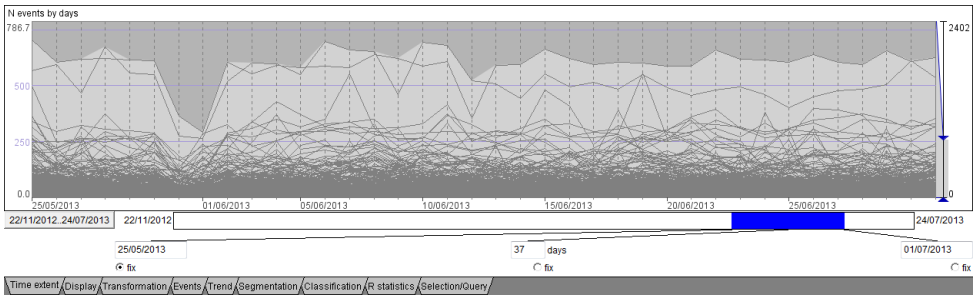

Figure 39: The Time Series Graph has been Zoomed in Time to the Interval 25 May 01 July 2013.

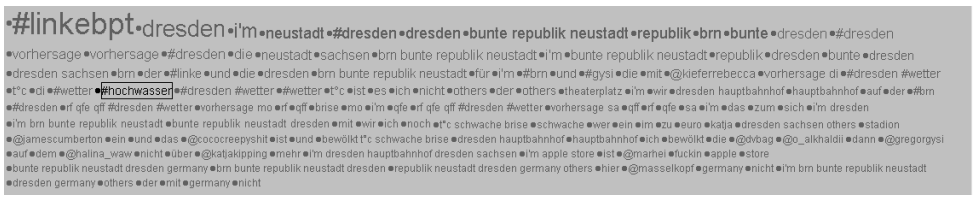

Figure 40: Frequent Words and Combinations in Dresden Centre on June 3, 2013.

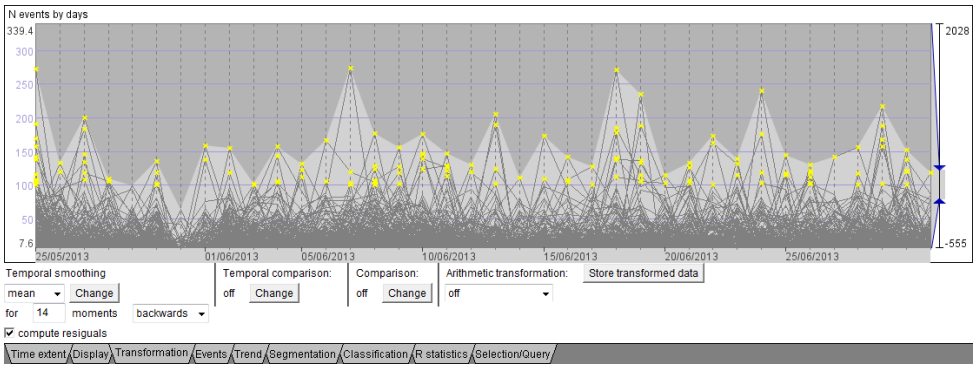

Figure 41: The Time Graph Shows the Differences (Residuals) of the Values to the Mean Values for the Previous 14 days. The yellow crosses mark the differences of 100 or more.

5.6.2 Experiment 2: Exploration of Selected Tweets Containing Relevant Terms

In the second experiment we apply following hypothesis. Spatio-temporal clusters of tweets containing relevant keywords may indicate disaster-affected places. The clusters do not need to be big (in terms of the number of messages), but several messages co-located in space and time may deserve a closer look while a single message may be not indicative.

Data preparation
We use a database query to extract from the set of German tweets only the tweets potentially relevant to flood events. Specifically, we apply a query condition that the message text must include substring "hoch" and substring "wasser", assuming that in some messages these substrings may be separated. This gives us certain amount of "false positives", such as

- "Gerade ein Foto hochgeladen Wasserschloss Haus Kemnade http://t.co/SFykqLOl";
- "Hundertwasser Ausstellung (@ Hochzeitshaus) http://t.co/JFKq0mY1".

However, there are also flood-relevant messages where "hoch" and "wasser" are separated, for example:

- "Wooow!! Das Wasser vom Rhein ist ziemlich hoch!!";
- "Für Konstanzer Verhältnisse ist der Wasserpegel auch erstaunlich hoch Hafen Konstanz http://t.co/OcgkwnCGy0".

There are also cases where the word "Hochwasser" is misspelled, for example,

- "#AltLostau ist jetzt nur noch mit dem Boot erreichbar - Schuldamm wurde überspült #Hochhwasser #lostau http://t.co/rfM6ylcBRT".

In some cases, the string "wasser" is also misspelled as "waser", but the messages were retrieved due to other occurrences:

- "In #Lostau steht das Wasser jetzt ca. 1km hinter dem Deich in den Kellern der Häuser #Hochwaser #Madeburg".

To include also texts with "wasser" misspelled as "waser", we use the query condition `instr(lower(MESSAGETEXT),'hoch')>0` and `(instr(lower(MESSAGETEXT),'wasser')>0 or instr(lower(MESSAGETEXT),'waser')>0)`. The query retrieves 2443 messages for the whole territory of Germany. After removing the messages including the substrings "hochgeladen" or "hochzeit" or "wasserkocher", 2429 messages remain. The messages span over the time period 25/11/2012 05:40:52 - 25/07/2013 11:45:16.

Exploration of the spatio-temporal distribution of the tweets
A map in Figure 42 shows the spatial distribution of the messages, which are represented by dot symbols (small circles) in violet. It is possible to notice concentrations of the symbols in the areas of Dresden, Magdeburg and some others (see Figure 43), which are known to be affected by the June floods of 2013. However, there is also a concentration in Berlin, which was not affected. By looking at the messages from Berlin, we see that they mention the flood

events in other places, actions of politicians, help to flood victims, or traffic problems caused by the floods. This shows that concentrations of disaster-related Twitter messages need to be interpreted with caution: not necessarily an event is where people tweet about it.

In Figure 44, a space-time cube shows the spatio-temporal distribution of the flood-related tweets. The vertical dimension represents the time, and the tweets are represented by balls placed in the cube according to their spatial and temporal positions. The time axis is oriented upwards; hence, the latest tweets are at the bottom and the most recent at the top of the cube. The cube has been rotated to be viewed from the southwest, i.e., the left side corresponds to the northwest and the right side to the southeast.

The cube contains a "column" made by vertically aligned balls, which means that all of the respective messages have the same location in space. The location is Konstanz (at the lake Bodensee); the messages, which are, most probably, automatically generated, inform about the water level in the lake, for example:

- "Bodensee-Pegel: 343,6 cm! - http://t.co/JWhYfRSk - Hochwasser-Infos: http://t.co/dG7uuYK1".

The highest level among the available messages is 467.5; it was reached on 03 June 2013. Besides the "column", it is possible to notice three major "layers"; we have applied different colouring for their better visual separation. The dark blue colour is used for the time period of November-December 2012, light blue for January middle March 2013, yellow for the period from mid-March to May 23, and red from May 24 till July 25. The periods include 69, 69, 39, and 2252 tweets, respectively (see Figure 45). The time histogram in Figure 46 shows the temporal distribution of the number of tweets by days. The highest number of tweets (306) was reached on 03 June 2013.

The three maps in Figure 47 show the spatial distribution of the tweets in the first two periods and in the last period. In the first period, most of the tweets are aligned along the valley of the river Rhine. Most of the messages reflect the water rise that happened around Christmas (December 20-30 of 2012). The messages of the second period (winter 2013) are more scattered over the territory. The messages of the fourth period (summer 2013) cover almost the whole territory of Germany.

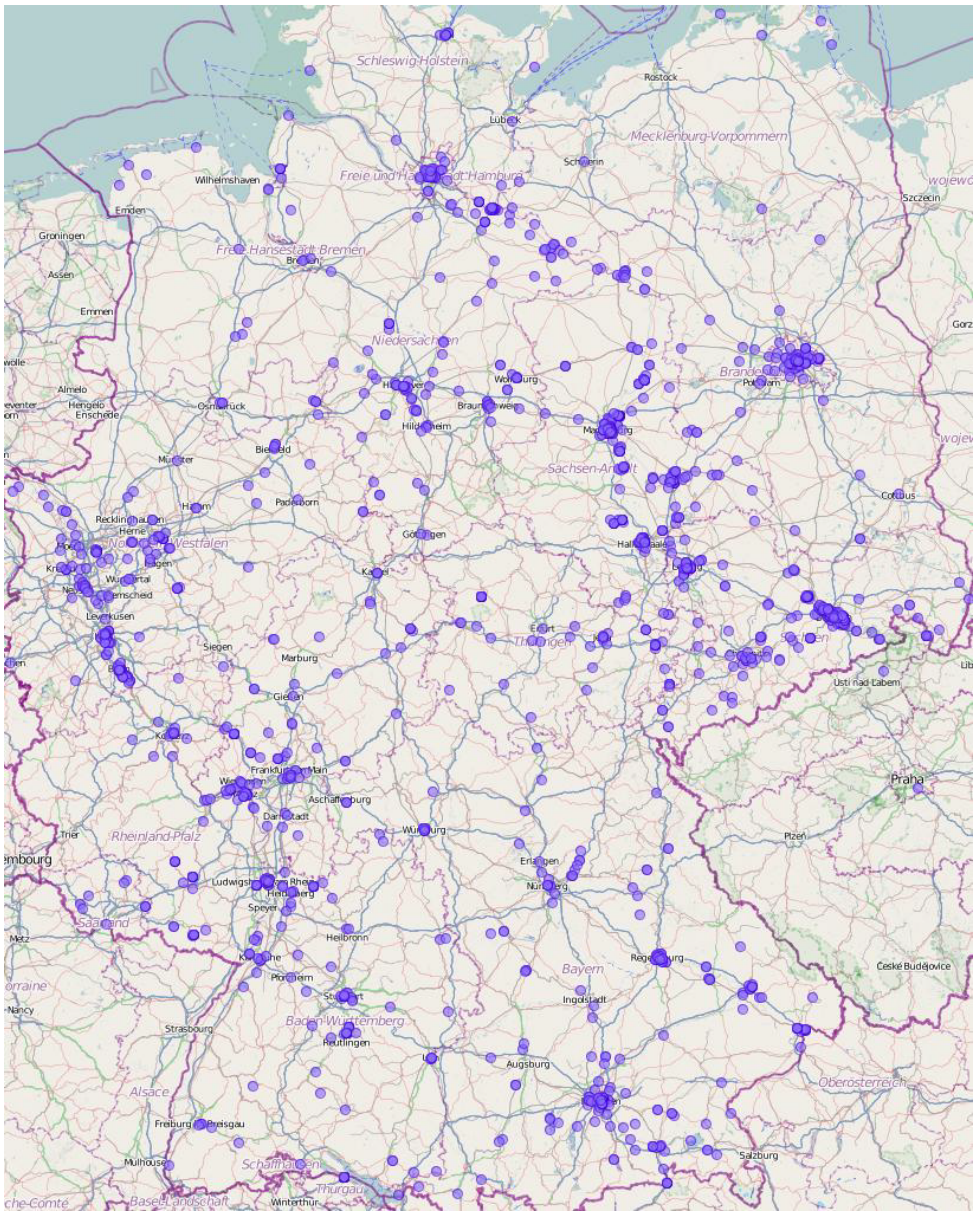

Figure 42: The Map Shows the Spatial Distribution of the Tweets Containing Flood-relevant Substrings.

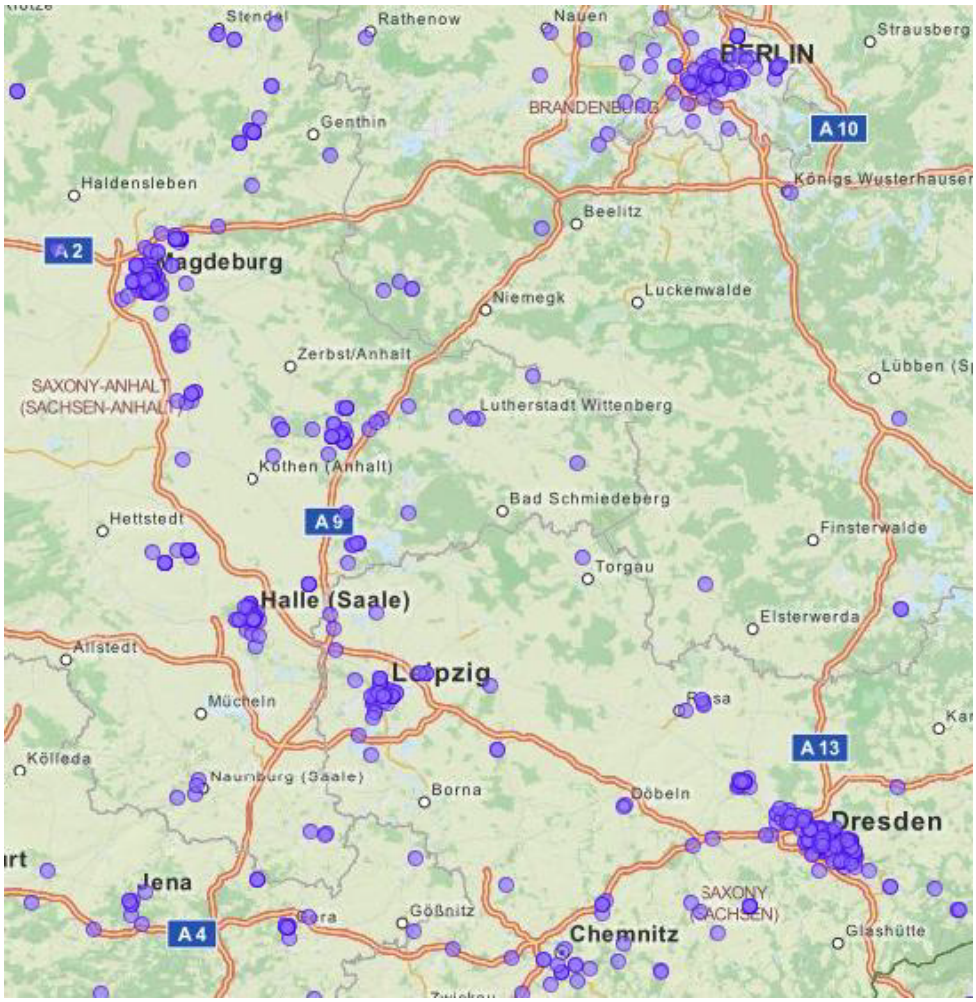

Figure 43: An Enlarged Map Fragment Shows an Area Affected by the June 2013 floods.

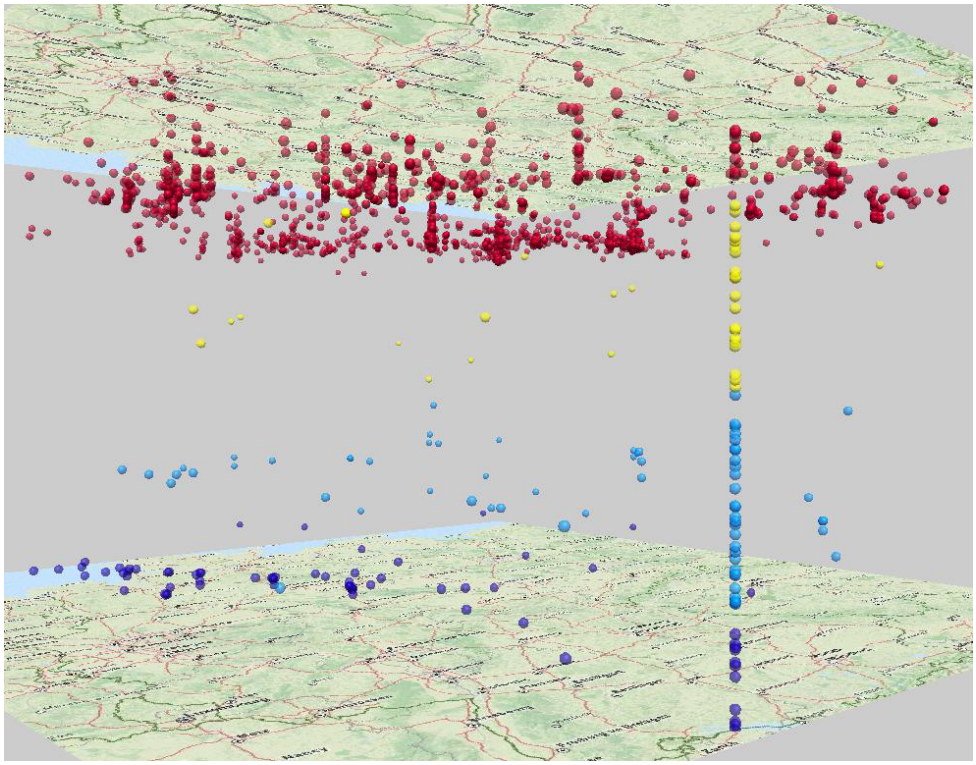

Figure 44: The Space-Time Cube Shows the Spatio-Temporal Distribution of the Flood-related Tweets.

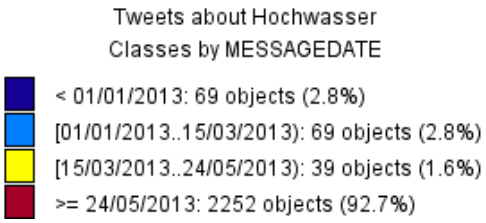

Figure 45: A legend explaining the colours in Figures 44, 46, and 47.

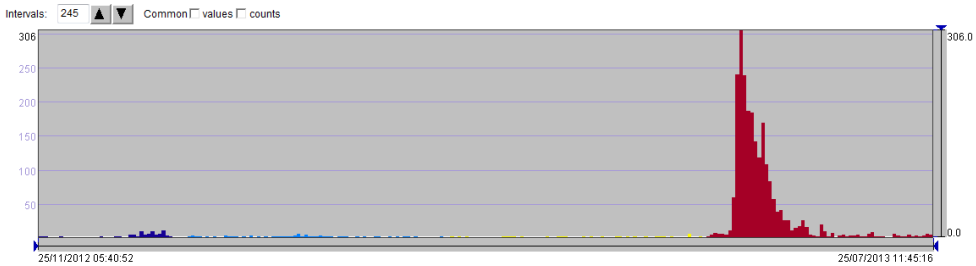

Figure 46: The Time Histogram Shows the Distribution of the Flood-related Tweets over Time.

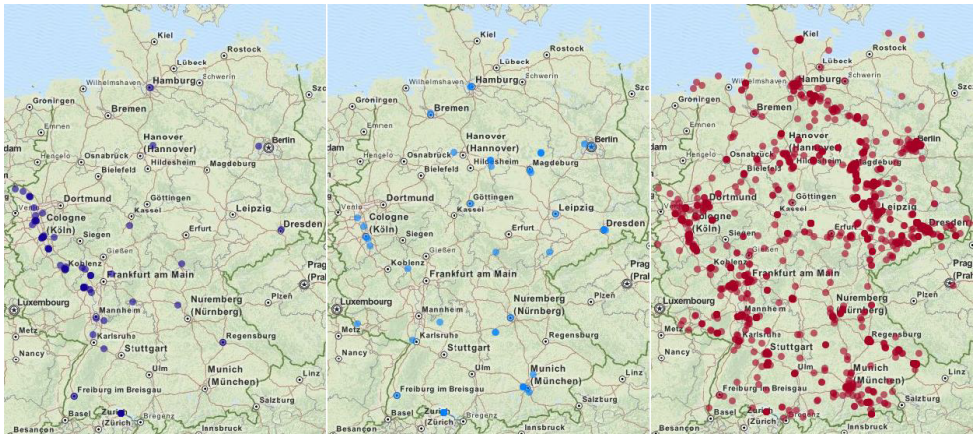

Figure 47: The Spatial Distributions of the Flood-related Tweets in 3 Time Periods: November-December 2012, January-March 2013, and End of May till End of July 2013.

Spatio-temporal clustering of the flood-related tweets

We apply density-based clustering (using OPTICS [ABKS99]) to the set of the flood-related tweets. The tweets are clustered according to their positions in space and time, i.e., as spatio-temporal points, with 30 km as the spatial distance threshold, 1 day (86400 seconds) as the temporal distance threshold, and 2 as the minimum number of neighbours within these distance thresholds. We obtain 90 clusters with the sizes from 3 to 783 including in total 1885 points (77.6% of all), and 544 points (22.4%) belong to "noise".

The map in Figure 48 and the space-time cube in Figure 49 show only the points included in the clusters, the "noise" is filtered out. Besides the points, the map contains also convex hulls built around the clusters. The space-time cube shows that only a few small clusters have been built in the first time period, none in the second and third periods, and very many in the fourth period. A fragment of a table view on the left of Figure 50 shows information about the tweets included in the clusters from the time interval 20 December 2012 till 31 May 2013. It is visible that there were no spatio-temporal clusters (with the given parameters) between 07 January and 20 May 2013.

A fragment of a table view on the right of Figure 50 shows summarized information about the clusters: number of objects (tweets), start and end times, duration (in days), and spatial extent (in metres). The largest cluster covering the area of Saxony (containing Dresden and Magdeburg) began on 01 June and ended on 17 June 2013; it includes 783 tweets and its spatial extent (bounding rectangle diagonal) is 268 km.

The spatial and temporal extents and internal structures of the clusters in the period starting 20 May till 30 June 2013 can be seen in space-time cubes in Figures 51 and 52. In Figure 52, the points making the clusters are shown together with the convex hulls of the clusters. The heights of the convex hulls represent the durations of the clusters.

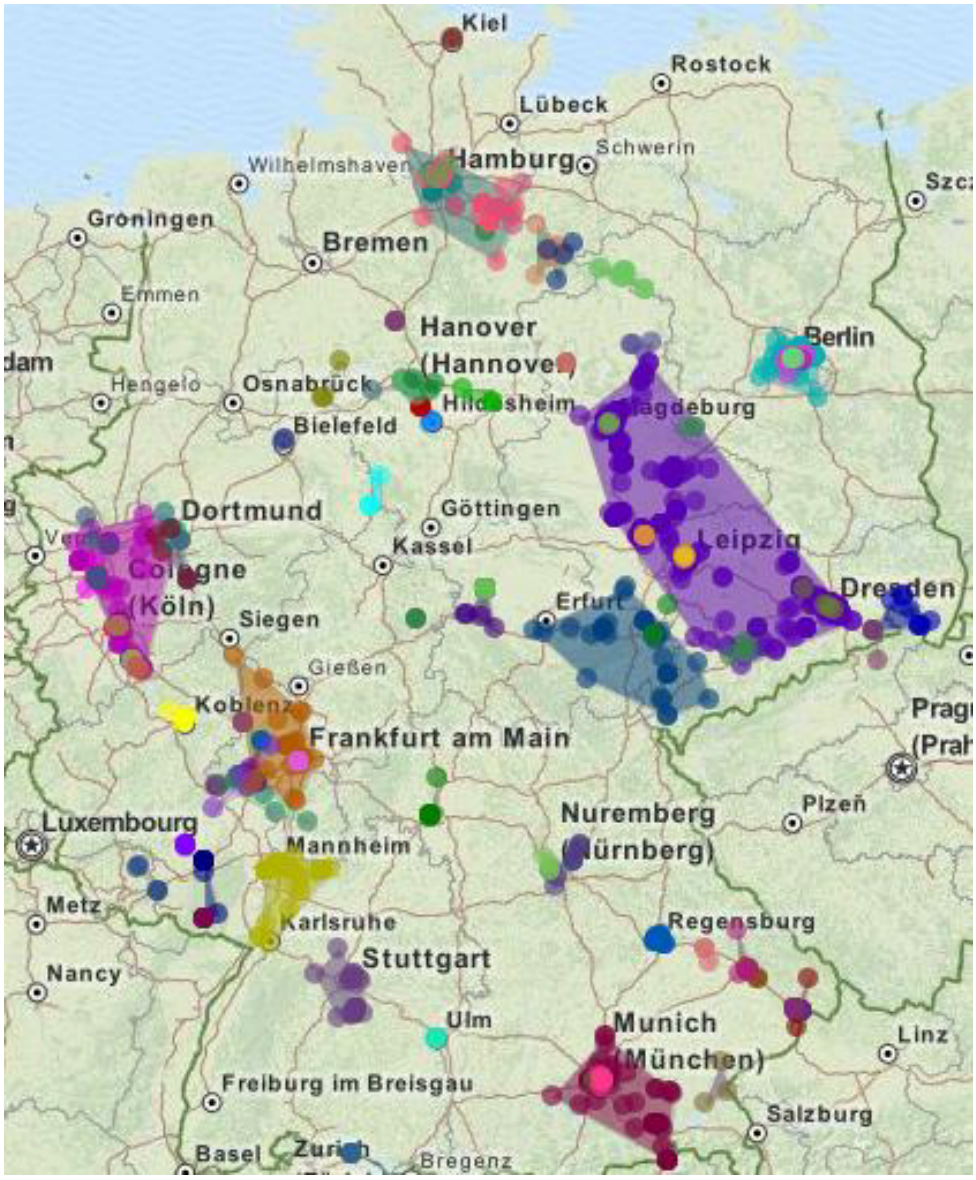

Figure 48: The Spatio-Temporal Clusters of the Flood-related Tweets are Shown on a map.

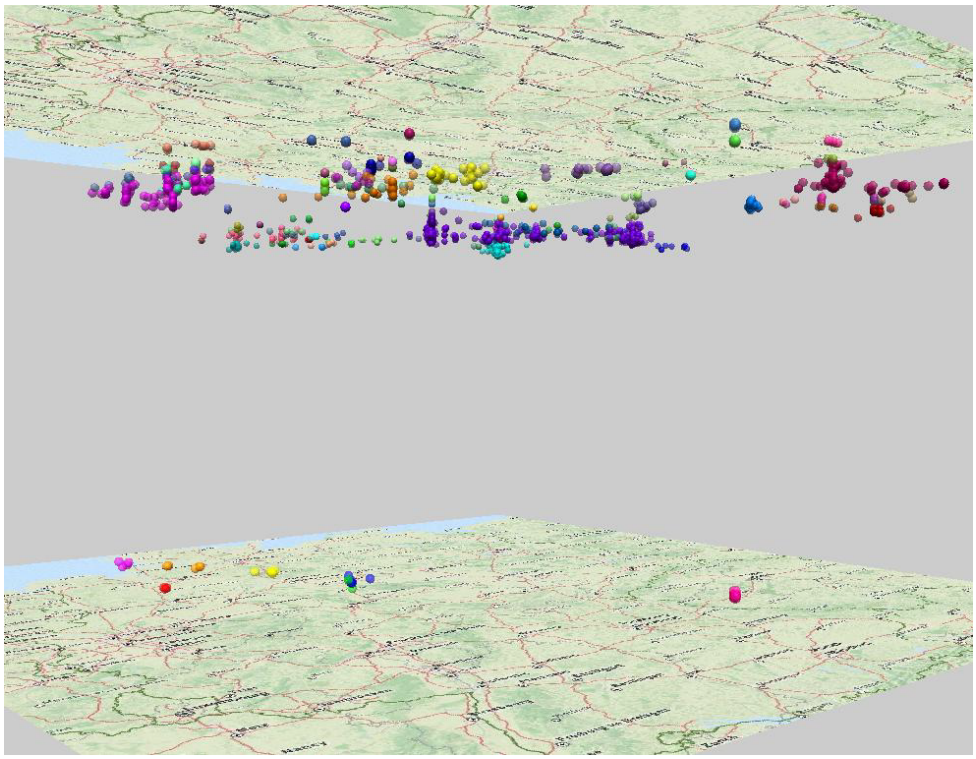

Figure 49: The Space-Time Cube Shows the Spatio-Temporal Clusters of the Flood-related Tweets.

	MESSAGETEXT	MESSAGEDATE	LOCATION
2817	Köln rüstet sich für Ho	20/12/2012 15:40:56	Köln; Köln
2818	@dancetheday awwww	20/12/2012 18:54:21	Köln; Köln
2820	@pechan58 hier auch	21/12/2012 07:11:41	Köln; Köln
2827	Nachdem ich heute ka	23/12/2012 08:40:03	Eltville am Rhein; Rhei
2828	@chkrechel was ist de	23/12/2012 12:59:11	Waldems, Rheingau-T
2829	@pyth2_0 Anstieg vom	23/12/2012 18:05:50	Waldems, Rheingau-T
2834	Gassirunde auf Umwe	25/12/2012 08:30:47	Eltville am Rhein; Rhei
2836	@Patrick_Kunkel Jupp	25/12/2012 08:43:20	Eltville am Rhein; Rhei
2838	@andi_y_ Dicker Baur	25/12/2012 10:52:45	Waldems, Rheingau-T
2839	#Hochwasser an der E	25/12/2012 16:20:43	Eltville am Rhein; Rhei
2840	Gestern konnte ich hie	26/12/2012 08:49:31	Eltville am Rhein; Rhei
2841	Hochwasser. http://t.co	26/12/2012 15:07:48	Mainz; Mainz
2846	Hochwasser @ Vallen	28/12/2012 14:14:07	Vallendar; Mayen-Kobl
2848	Hochwasserwatching.	29/12/2012 12:48:56	Koblenz; Koblenz
2850	Jemand eine Schifffah	29/12/2012 15:50:21	Neuwied; Neuwied
2850	Die wahrscheinlich ers	29/12/2012 16:15:03	Koblenz; Koblenz
2850	Wir haben hier ja gera	29/12/2012 18:27:17	Koblenz; Koblenz
2848	Wie hoch das Wasser	28/12/2012 16:07:19	Düsseldorf; Düsseldo
2847	Es ist mal wieder kurz	28/12/2012 23:41:53	Neuss; Rhein-Kreis N
2849	Hochwasser am Rhein	29/12/2012 11:09:34	Dusseldorf; Düsseldo
2850	Hochwasser. @ Rhein	29/12/2012 16:17:25	Düsseldorf; Düsseldo
2844	Achtung Hochwasser (	29/12/2012 13:17:22	Cologne; Cologne
2850	Hochwasser am Rhei	29/12/2012 16:21:18	Cologne; Cologne
2851	Die Abflussrohre mach	30/12/2012 00:06:04	Bonn; Bonn
2853	#Hochwasser #Rhein	30/12/2012 15:56:34	Bonn; Bonn
2854	Nichts geht mehr! #Ho	30/12/2012 16:04:33	Bonn; Bonn
2875	Bodensee-Pegel: 350;	05/01/2013 13:25:29	Konstanz; Konstanz
2876	Bodensee-Pegel: 351;	05/01/2013 21:36:59	Konstanz; Konstanz
2877	Bodensee-Pegel: 352;	06/01/2013 05:42:18	Konstanz; Konstanz
2878	Bodensee-Pegel: 353;	06/01/2013 13:18:34	Konstanz; Konstanz
2883	Bodensee-Pegel: 351;	07/01/2013 13:12:28	Konstanz; Konstanz
3308	#Hochwasser #Lauter	20/05/2013 15:50:54	Oberweiler-Tiefenbach
3308	#Hochwasser #Lauter	20/05/2013 15:53:12	Oberweiler-Tiefenbach
3308	#Hochwasser #Lauter	20/05/2013 16:34:14	Oberweiler-Tiefenbach
3310	Schade; dass für den r	20/05/2013 16:59:51	Oberweiler-Tiefenbach
3309	@smokediver_tech Ma	20/05/2013 18:31:11	Oberweiler-Tiefenbach
3308	3-Tage-Dauerregen ur	26/05/2013 20:10:40	Hildesheim; Hildesheim
3308	Schöne Kastanie am H	27/05/2013 00:11:12	Hildesheim; Hildesheim
3308	Nach Hochwasserum	27/05/2013 06:37:52	Hannover; Region Han
3390	#Hochwasser an der #	27/05/2013 19:07:41	Beverungen; Höxter
3390	#Weser #Hochwasser	27/05/2013 19:23:52	Beverungen; Höxter
3391	Land unter. #Weser #H	28/05/2013 16:25:10	Holzminden; Holzmind
3407	Regen in #Sarstedt. Mi	31/05/2013 04:21:02	Sarstedt; Hildesheim
3407	Lage in #Sarstedt ents	31/05/2013 04:52:07	Sarstedt; Hildesheim
3407	Trotz der #Hochwasse	31/05/2013 05:29:03	Sarstedt; Hildesheim
3407	Hoher Pegelstand der	31/05/2013 05:39:31	Sarstedt; Hildesheim
3407	gesteigertes Hochwas	30/05/2013 17:15:23	Braunschweig; Brauns
3407	Wenns weiter so #regr	31/05/2013 14:18:06	Peine; Peine
3407	#hochwasser #brauns	31/05/2013 15:18:24	Brunswick; Brunswick

	N objects	Begin time	End time	Duration, days	extent
1	3	20/12/2012 15:40:56	21/12/2012 07:11:41	0.65	2983.47
2	3	23/12/2012 08:40:03	23/12/2012 18:05:50	0.39	28732.69
3	6	25/12/2012 08:30:47	26/12/2012 15:07:48	1.28	32180.86
4	5	28/12/2012 14:14:07	29/12/2012 18:27:17	1.18	13611.63
5	4	28/12/2012 16:07:19	29/12/2012 16:17:25	1.01	19822.07
6	5	29/12/2012 13:17:22	30/12/2012 16:04:33	1.12	29862.53
7	5	05/01/2013 13:25:29	07/01/2013 13:12:28	1.99	0.00
8	5	20/05/2013 15:50:54	20/05/2013 18:31:11	0.11	59.40
9	3	26/05/2013 20:10:40	27/05/2013 06:37:52	0.44	25890.91
10	3	27/05/2013 19:07:41	28/05/2013 16:25:10	0.89	19583.75
12	3	30/05/2013 17:15:23	31/05/2013 15:18:24	0.92	20148.18
11	4	31/05/2013 04:21:02	31/05/2013 05:39:31	0.05	1602.83
16	42	01/06/2013 15:20:12	08/06/2013 18:31:08	7.13	112439.86
14	70	01/06/2013 16:17:52	07/06/2013 11:30:01	5.80	77899.30
15	102	01/06/2013 18:48:38	08/06/2013 13:07:14	6.76	124240.31
13	9	01/06/2013 19:03:53	03/06/2013 22:52:37	2.16	44707.37
19	36	01/06/2013 20:15:53	06/06/2013 20:54:58	5.03	9325.52
17	3	01/06/2013 20:47:49	02/06/2013 12:14:41	0.64	3128.62
18	783	01/06/2013 21:16:41	17/06/2013 13:23:23	15.67	268216.92
29	36	01/06/2013 23:04:55	04/06/2013 21:08:07	2.92	145495.48
21	20	01/06/2013 23:19:15	04/06/2013 21:59:04	2.94	35756.72
37	8	02/06/2013 01:17:01	02/06/2013 21:24:34	0.84	10870.79
36	86	02/06/2013 01:45:03	04/06/2013 11:54:20	2.43	5910.57
38	10	02/06/2013 08:11:30	02/06/2013 19:52:26	0.49	65904.33
27	110	02/06/2013 10:37:19	12/06/2013 12:17:40	10.08	122290.52
39	3	02/06/2013 11:07:18	02/06/2013 16:55:04	0.24	1205.35
28	24	02/06/2013 13:05:32	09/06/2013 16:10:47	7.13	33506.15
20	141	02/06/2013 14:03:46	14/06/2013 10:51:19	11.87	52922.38
30	18	02/06/2013 14:07:31	06/06/2013 18:58:56	4.20	27149.26
32	4	02/06/2013 17:05:45	03/06/2013 16:59:35	1.00	25942.72
40	3	02/06/2013 17:57:29	03/06/2013 19:57:08	1.08	33577.95
35	11	02/06/2013 18:56:30	03/06/2013 23:21:42	1.18	35802.50
23	18	02/06/2013 20:13:55	05/06/2013 13:05:31	2.70	40448.22
31	3	03/06/2013 10:15:14	03/06/2013 12:50:43	0.11	6233.03
34	3	03/06/2013 13:12:04	03/06/2013 21:12:14	0.33	214.51
33	9	03/06/2013 17:10:40	07/06/2013 14:57:55	3.91	338.90
22	10	03/06/2013 19:58:20	05/06/2013 13:22:21	1.72	26951.60
25	6	03/06/2013 20:40:56	04/06/2013 20:07:21	0.98	31072.73
24	3	03/06/2013 21:02:17	04/06/2013 19:22:38	0.93	27962.82
26	4	04/06/2013 12:49:09	05/06/2013 13:48:36	1.04	12184.11
42	5	04/06/2013 14:11:27	04/06/2013 18:10:09	0.17	13017.49
41	8	05/06/2013 12:27:20	07/06/2013 18:25:56	2.25	35603.25
53	3	05/06/2013 19:53:17	06/06/2013 16:52:59	0.87	22139.15
52	14	05/06/2013 23:31:38	06/06/2013 16:55:41	0.73	21161.71
50	4	06/06/2013 12:24:07	07/06/2013 14:25:06	1.08	10033.47
54	3	06/06/2013 17:44:35	07/06/2013 16:20:20	0.94	9350.00
51	5	06/06/2013 19:03:13	08/06/2013 00:29:08	1.23	5619.84
47	7	06/06/2013 21:31:28	10/06/2013 07:07:07	3.40	32206.08
44	4	07/06/2013 13:05:56	08/06/2013 18:30:23	1.23	24516.47

Figure 50: Fragments of two Table Views show Detailed (left) and Summarized (right) Information about the Spatio-Temporal Clusters.

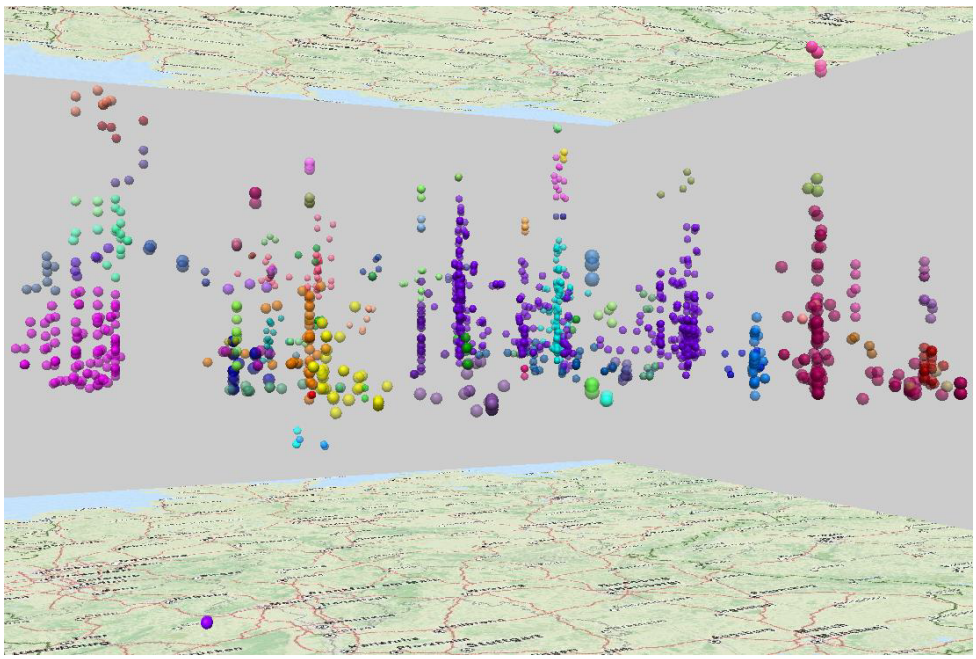

Figure 51: The Space-Time Cube shows the Spatio-Temporal Clusters of Tweets from the Time Interval 20 May till 30 June 2013.

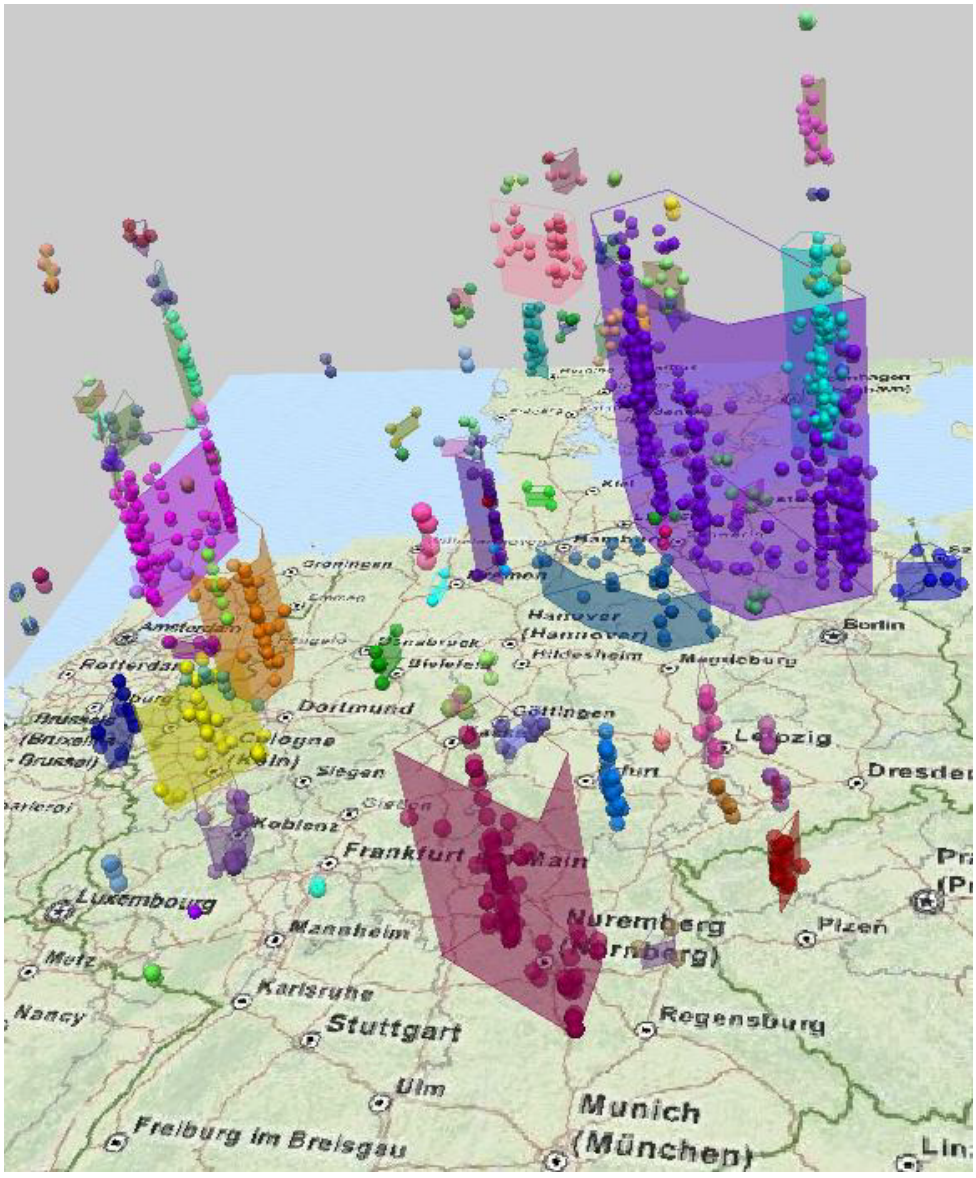

Figure 52: The Spatio-Temporal Clusters of Tweets from the Time Interval 20 May till 30 June 2013 are shown Together with their Convex Hulls.

Matching detected clusters to other sources of information
As the "ground truth" for checking the validity of our clusters, we use other information sources that can be found in the Web using Google. The first detected cluster in the period end May till June 2013 is in town Oberweiler-Tiefenbach (northwest of Kaiserslautern) in the valley of a small river Lauter. The cluster consists of 5 messages posted on 20 May between 15:53 and 18:31; i.e., it has quite short duration. We could not find information about this flood in other sources. The next cluster (3 messages) is in Hildesheim near Hannover starting on 26 May at 20:10 and ending on 27 May at 6:38. Information in other sources:

- YouTube contains 4 videos dating from 26, 27, and 28 May
- A photo from 28 May at http://fotojournalismus.tumblr.com/post/51568293826/flooded-fields-near-hildesheim-germany-on-may-28 and http://blogs.ft.com/photo-diary/tag/hildesheim/
- News articles from 30 May at http://www.thelocal.de/national/20130530-50013.html and 31 May at http://floodlist.com/europe/germany-may-2013.

Another cluster located close to Hildesheim (in Sarstedt) occurred on 31 May 04:21-05:39. The news articles from 30 and 31 May report about floods not only in Hildesheim but also in other places in Lower Saxonia, in particular, Braunschweig. We have detected a cluster at Braunschweig from 30 May 17:15 to 31 May 15:18. Other clusters in this region are 27-28 May at Höxter, 03-04 June at Minden, 02-03 June and 04 June near Hannover (2 clusters), and 11-12 June again near Hannover. The news article from 31 May mentions also floods in Bavaria, in particular, in Bamberg. We did not obtain any cluster near Bamberg and have no clusters in Bavaria beginning earlier than 1 June.

On 1 June, 9 different spatio-temporal clusters began in different regions of Germany: on rivers Main, Rhine, and Neckar on the west of the country, in Bavaria on river Danube (Passau and Regensburg) and in Saxony on rivers Spree, Elbe, Saale, and Mulde.

Other information:

- News articles from 1 June http://www.dw.de/floods-ravage-south-and-east-germany/av-16853686 and http://bigstory.ap.org/article/germany-braces-more-flooding-after-heavy-rains report rising of water levels on Rhine, Danube, and Neckar and mention cities Passau and Gera, for which we have clusters beginning on 1 June.
- News articles from 1 and 2 June also write about the flood in Passau on 1 June.
- YouTube video from 2 June shows flood on Neckar.

The discovered clusters also correspond to the Wikipedia article http://en.wikipedia.org/wiki/2013_European_floods.

Conclusion
Spatio-temporal clustering of pre-filtered disaster-related tweets may allow detection of locations and times of disaster events. However, it should be borne in mind that

- it is not guaranteed that any event is always represented by a tweet cluster;
- some tweet clusters occur in places and times where no disaster events happen since tweets may refer to events occurring elsewhere; hence, it is necessary to check the content of the messages in each cluster;
- some tweet clusters may refer not to disaster events themselves but to consequences of disaster events, such as traffic problems; this also shows a need of checking the tweet content.

5.6.3 Summary

In result of previous analysis, we conclude following points:

- To detect disasters and other "problematic" events from the stream of tweets, it is reasonable to filter the set of incoming messages based on a predefined vocabulary of relevant terms.
- It is reasonable to combine the tweets with geo- and time-referenced items from other sources (YouTube, Flickr, ...), which can be filtered based on their titles and/or tags.
- Events may be detected by spatio-temporal clustering of pre-filtered objects (tweets and, possibly, posts from other media). For this purpose, a clustering algorithm working in real time within a distributed computing architecture needs to be developed. The algorithm must be able to attach new incoming objects to appropriate existing clusters and store the history of detected clusters, i.e., how they evolve over time: move in space, expand or shrink, become denser or sparser, or keep stable.
- For each new cluster, an analyst needs to check (a) if it really refers to an event occurring in the same place and time as the cluster or to an event occurring elsewhere; (b) if it refers to locally experienced consequences of an event that occurred elsewhere.
- Besides specifically looking for predefined types of events using vocabulary-based filtering, it may be reasonable also to pay attention to unusual concentrations of tweets in space and time. This can be done by aggregating all tweets by suitable spatial compartments and time intervals into place-related time series. For each place, the current level of Twitter activities computed in real time needs to be compared with the usual level for the respective day of the week and time of the day derived from the historical data. For detected significant deviations from the usual levels, the most frequent key words and phrases may be analysed to interpret what is going on.

5.7 Geospatial Emotion Analysis for Event Detection

Fraunhofer explored the emotions expressed in Tweets in Germany during the period of the floods (June 1 - 23). The Tweets were annotated using the method described in [VG13, VGBK13b] provided by UoA. This method applies i) emotion extraction techniques on microblogs, and ii) location extraction techniques on user profiles. Combining these two, highly unstructured content is converted to thematically enriched, locational information, which is

presented to the user through a unified front-end. Tweets were annotated using the following emotions: *anger*, *disgust*, *fear*, *joy*, *sadness*, *surprise*. As a first step we have explored the emotion of *anger* for the areas with 50km radius. The resulting time series seem to have quite interesting profiles that can be related to the flooding event. We have mean-normalized each time series (for discarding differences in magnitude) and smoothed them over 5 days period. The derived time series have been clustered with kMeans with different K. Quite interesting results appear with k=5. In Figure 53 we can observe the results of clustering. Areas with the same color belong to the same cluster. Cluster 1 (red) (see Figure 54) starts with rather high counts of anger messages, but the counts monotonously decrease. This is an interesting result since these areas include flooded cities such as Dresden, Magdeburg, Ulm and Karlsruhe.

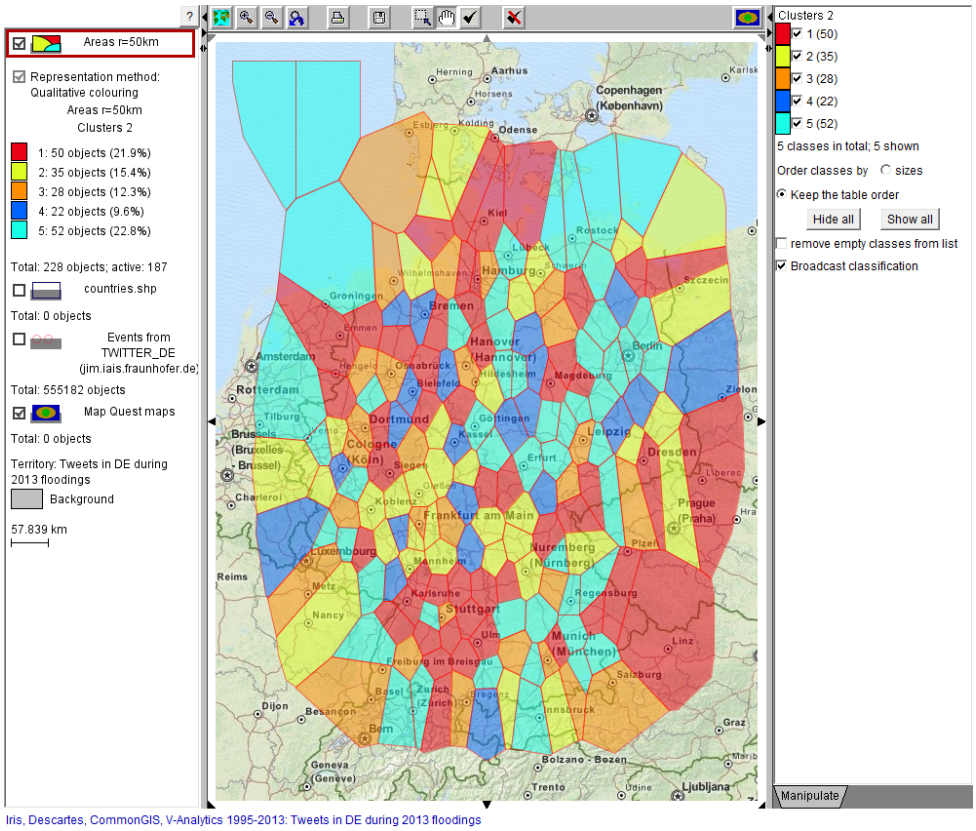

Figure 53: Results of emotion time series clustering.

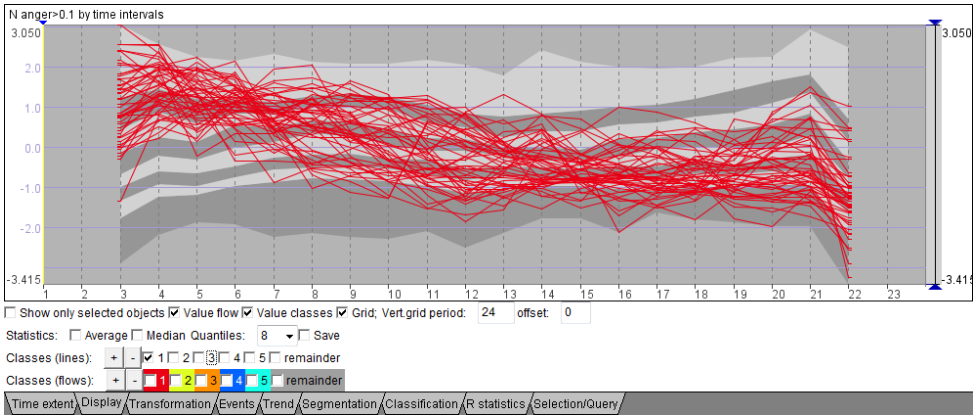

Figure 54: Emotion time series of Cluster 1 (red).

5.8 Event Detection in Mobile Phone Usage Data

The mobile phone network is a sensor network for mobility and activity information with an nearly complete coverage of Germany. We have access to network usage metrics of cells for a time period of more than 2 years. A more detailed description of this metric is provided in Section 4.1.6. In general this metrics holds information on the traffic load at a cell tower during a specific time interval. Additionally, we know the location and the coverage area for each of more than 50.000 mobile phone towers.

We are interested in discovering anomalies, identifying (relevant) events and re-constructing situations. From the overall process perspective, one important responsibility of the ISA analyzing the mobile phone data is to detect and report interesting sensor readings to the round-table.

This leaves the non trivial task of defining the term *interesting reading* to the sensor processor. As a starting point, we assume that each cell tower has some kind of normal load. Any deviation from this expected load classifies for an interesting reading. To model this state of normality we initially choose a simple model, which we can use to refer to when applying more sophisticated methods later on. We model the load of a cell at a given time as a Gaussian process. For this we assume that each cell has an expected load value at a given time and all aberrations from this value are only caused by white random noise.

Initially, we aggregate incoming sensor readings per hour of day, that is, we get a load value for every hour and for each of the 50.000 cells, for instance on Fridays at 10 o'clock and $CellID = XE3452G$. As a preliminary approach, we model each of phone cell's time series as result of a Gaussian process with $\mathcal{N}(\mu, \sigma)$. To account for the circadian and weekly cycles of human activities, each cell's model comprises of 24 hours $\times$ 7 weekdays $=$ 168 individual distributions $\mathcal{N}_i$. For the sake of confirmation of suspected events (based on the Twitter stream), we assume each cell exhibits its normal behavior during time intervals not belonging to any such detected event, which are thus used as input to the model.

Initial Experiment

To gain some practical insights, we made an initial experiment. We chose a cell near the ,,Rhein Energie Stadion" (a soccer arena in Cologne) and calculated the values for day hours

and weekdays, as described above. For this experiment we choose a simple rule based approach: a value outside the 98% confidence interval is marked unusual. If it remains off this confidence corridor for at least two hours we consider it to be an anomaly.

Applying this to a cell of interest, we find an anomaly at 22th of December, c.f. Figure 55. If we now reconsider the round-table this anomaly would be reported and discussed. After involving a human expert we find that the first peak of the signal corresponds to the first halftime and the second peak to the second half time of a soccer game.

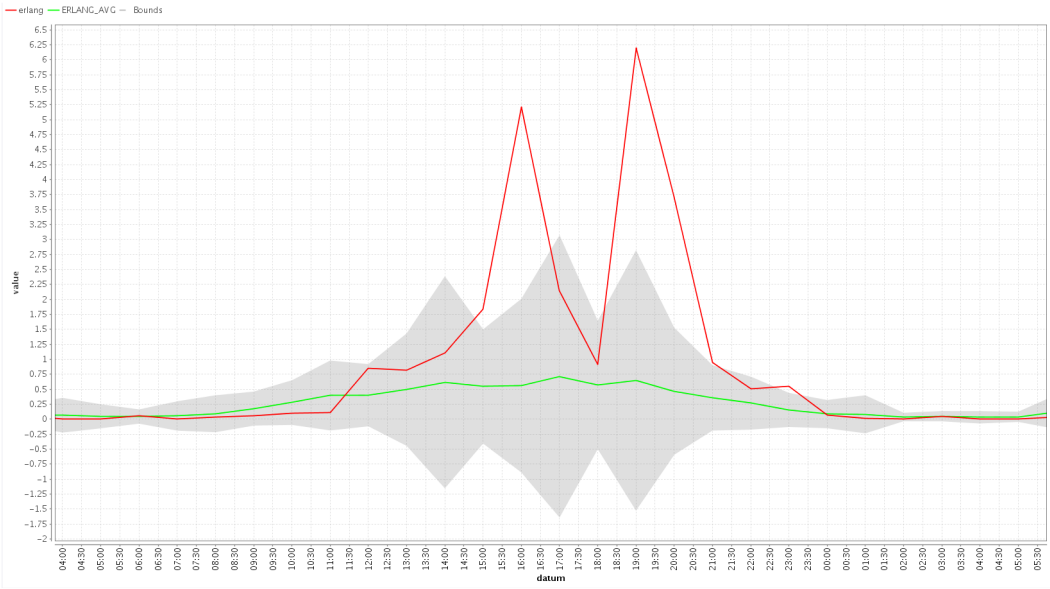

Figure 55: Anomaly detected in phone data at the time of soccer game. The green line is the average calculated from past sensor readings, the grey shade covers the confidence corridor and the red line is the actual reading.

Another interesting finding is the anomaly detected on 16th July, see Figure 56. This anomaly aligns with a rock concert that took place at the stadium. It shows that different types of anomalies can be identified applying this straight forward approach. Such events like a soccer game are, of course, highly expectable at this venue and also a concert, although it might be less expected.

This experiment shows that if we can find an anomaly we can report on the location and time quite accurately using the mobile network infrastructure. Naturally, we need additional sources of information in order to add semantics of the finding, i.e. which type of event we look at. At this stage of the project we directly involved a human expert but in the course of the project this will be automatized. We looked up an event database to identify and label both events. Concerning the INSIGHT system this aligns well with the proposed approach of the round-table, as any Intelligent Sensor Agent is expected to report it's findings to ISA and a moderator in order to verify, enrich and label events.

The experiment clearly shows that not all known events, i.e. soccer games, can be detected using the basic approach described. Noticing false-positives is not very surprising as the game schedule strongly influences the mean (expected normal value) as well as the variance. Therefore, the sensor readings at regular games naturally fall into the chosen confidence

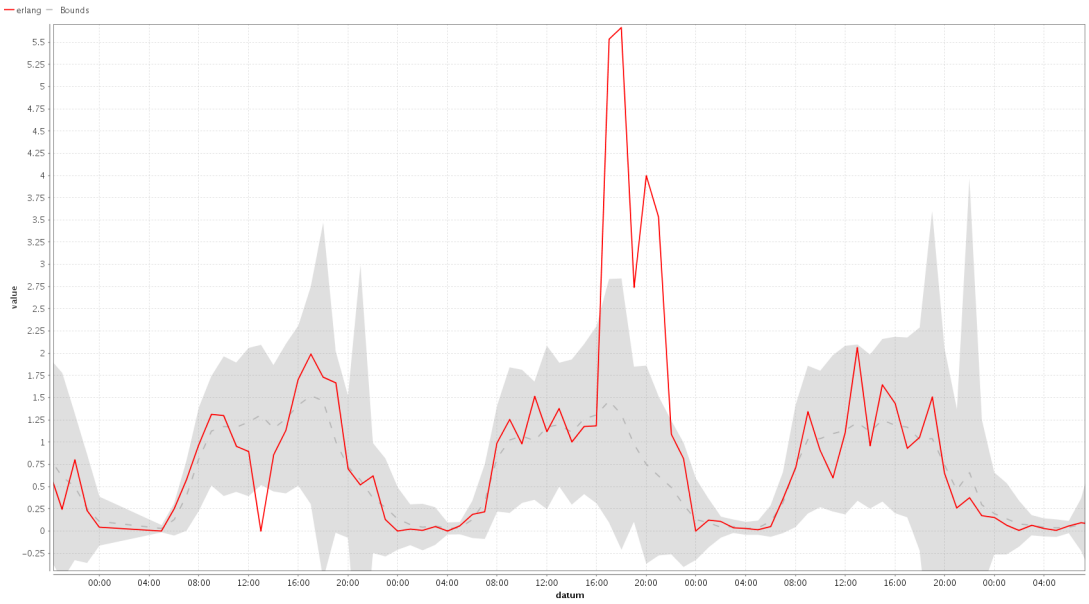

Figure 56: Anomaly in Erlang data, detected during a rock concert.

intervals. This also violates the assumption that the expected normal behavior of a cell follows a normal distribution. Reason for this violation is the weekly scheduled soccer games.

In addition to these false-negatives, we detected a couple of false-positive classifications as well. For instance an anomaly is detected for Friday, the 10th of November, starting around midnight and ending at around 6 o'clock in the morning (see Figure 57). The combination of the weekday and the time slot does not seem to be a reasonable argument for a true event. Also a web search did not uncover any hints for a sport or other event taking place in the area served by the cell. For that reasons it is very likely that this is a false finding here.

Deriving from the most obvious limitation, we currently work on methods to detect repetitive formations in the signal. Fourier transformations or the related wavelet transformation seem to be a good starting point here. It clearly can be stated, that we will not have a fully labeled data set in the application scenarios. There is an immanent need for a more capable model of events.

Currently, we are exploring ways to make use of the fact, that there is a regular structure behind the timing of many large events. One approach we are evaluating, is to decompose the signal into components that can be better "understood". This follows works previously applied to climate data set, more precisely on the analysis of CO_2 concentration in the atmosphere. The authors of [CCMT90] propose a decomposition into three parts, a (long term) trend, a repeating component (seasonal) and the remainder of the signal that is not 'explained' by one of the other two parts. This approach looks promising, for instance it would be very useful to separate the influence of a holiday break from the rest of the signal. Also it would be useful, to detect and remove a probably continuously ascending trend in the data, caused by the increasing usage of mobile phones over the years.

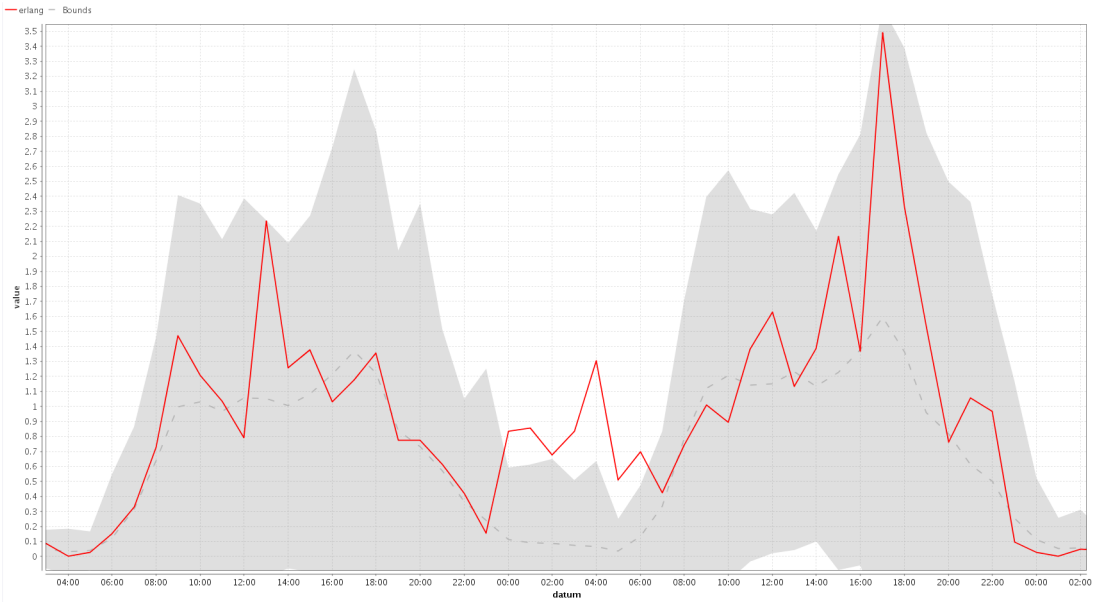

Figure 57: False finding of an anomaly.

Tracing the flood in Germany

In Mai and June 2013 a massive flood disaster hit Germany. Based on the assumption that disruptive events like flood also cause a detectable change in mobile phone utilization patterns, we wanted to test if we can detect spatio-temporal clusters of interest. In particular, we try to detect deviations from the representative daily time series of Erlang values during an event. We apply the same event detection method as describe above.

To identify the cells with significant deviations we compare the actual Erlang values with the model-predicted mean values μ. If the difference exceeds 2σ we flag the cell and time slot as an anomaly. Due to their small spatial extent flagged cells may provide more fine-grained indication of potential event locations. Figure 58 shows hotspots of significant deviations along the Elbe river, corresponding to the time interval of the rather large purple cluster in Figures 48 and 49. Areas and times correspond to major events and affected areas of the flood disaster.

It should be noted that this straight-forward approach is intended as a proof-of-concept. We were able to identify events that specifically relate to the flood situation. In combination with the analysis of Twitter messages in Section 5.6 both data source validate and extend each other.

Conclusion

Our results have shown that a disaster causes anomalies in the load distribution of a cell tower as well as an entire mobile sensor network. Those anomalies can be detected and linked to a specific area or region. We have seen that mass events (e.g. soccer game) can be identified as well as more specific and widespread events during a flood situation. Based on our experiments we could establish a connection to findings by the Twitter data analysis. This supports our

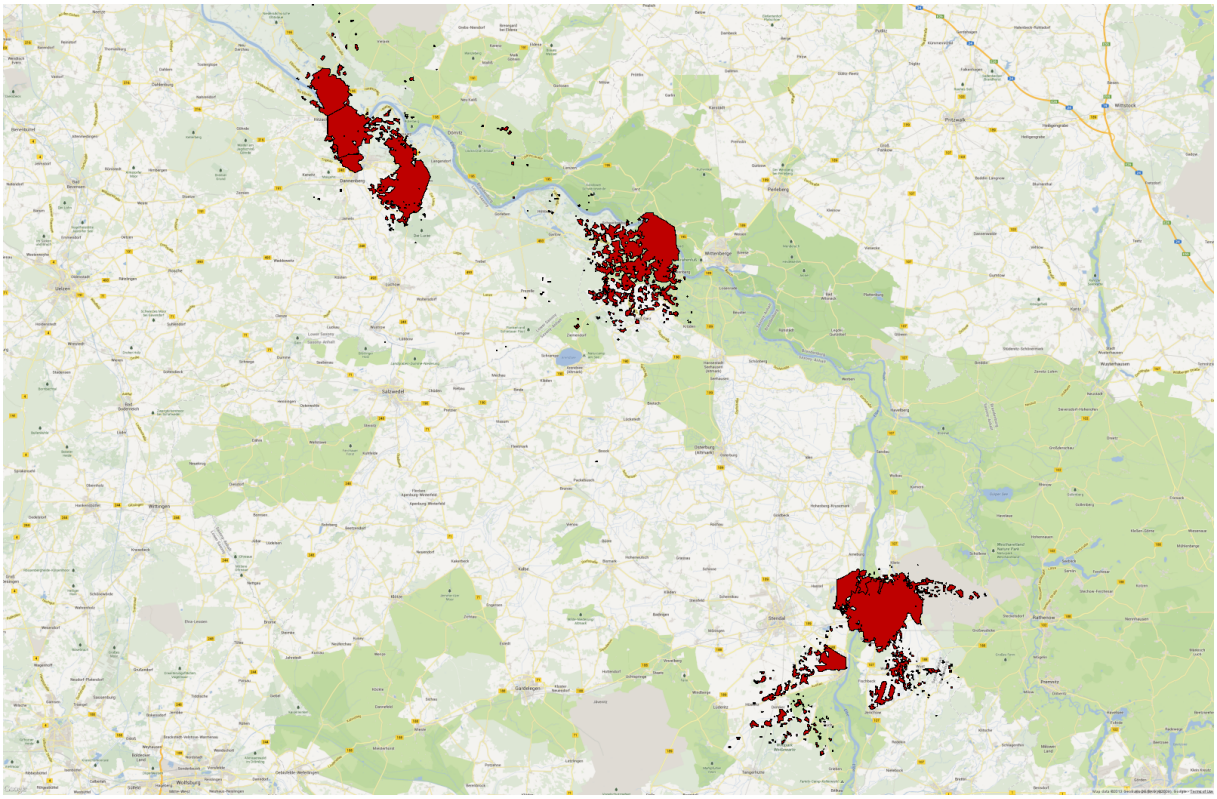

Figure 58: Affected cell towers that show anomalies.

round-table approach. Different data sources (various distributed sensors) do provide unique insights on an event. We will continue investigating the data collected for the flood situation to see the extent to which events are detectable. Furthermore, we will investigate the possibility to derive patterns or fingerprints for certain events or types of events from the data. This helps to automatically provide a label for known event.

5.9 Event Recognition Experiments

We describe our first event recognition experiments on the bus and SCATS datasets that are available from `dublinked.ie`. Section 5.9.1 presents the event processing engine. Section 5.9.2 shows a number of event patterns for intelligent transport and traffic management. Finally, Section 5.9.3 presents our preliminary experimental results.

5.9.1 A Logic-based Event Model

We present a logic-based event processing model based on the Event Calculus for Run-Time reasoning (RTEC) [ASP12]. The Event Calculus, introduced in [KS86], is a logic programming formalism for reasoning about events and their effects. Based on [ASP12], we summarise the essentials of the model. We adopt the common logic programming convention that variables start with upper-case letters (and are universally quantified, unless otherwise indicated), while predicates and constants start with lower-case letters. Our approach relies on logic programming due to the formal, declarative semantics and the rich expressiveness it offers. RTEC

Table 3: RTEC Predicates.

Predicate	Meaning
happensAt$(E,\ T)$	Event E occurs at time T
holdsAt$(F=V,\ T)$	The value of fluent F is V at time T
holdsFor$(F=V,\ I)$	I is the list of the maximal intervals for which $F=V$ holds continuously
initiatedAt$(F=V,\ T)$	At time T a period of time for which $F=V$ is initiated
terminatedAt$(F=V,\ T)$	At time T a period of time for which $F=V$ is terminated
union_all$(L,\ I)$	I is the list of maximal intervals produced by the union of the lists of maximal intervals of list L
intersect_all$(L,\ I)$	I is the list of maximal intervals produced by the intersection of the lists of maximal intervals of list L
relative_complement_all(I', L, I)	I is the list of maximal intervals produced by the relative complement of the list of maximal intervals I' wrt every list of maximal intervals of list L

supports efficient reasoning (as is evident in the empirical evaluation), and thus serves us well in illustrating the approach feasibility.

Event Representation Systems for event recognition (event pattern matching [Luc02]) accept as input a stream of time-stamped simple, derived events (SDE). An SDE (or low-level event) is the result of applying a computational derivation process to some other event, such as an event coming from a sensor [LS08]. Events that arrive at the system are not the raw events emitted by sensors. Such raw events are enriched, filtered, and aggregated by multiple mediators, whose internal functionalities may not be known to the event recognition system. Using SDE as input, event recognition systems identify composite events (CE) of interest — collections of events that satisfy some pattern. The specification of a CE (or high-level event) imposes temporal and, possibly, atemporal constraints on its deriving events, either SDEs or other CEs.

In the RTEC model, types of events are represented as n-ary predicates *event(Attribute1,...,AttributeN)*, such that the parameters define the attribute values of an event instance *event(value1,...,valueN)*. An example from the Dublin traffic management scenario is the type of SDE emitted by SCATS sensors, *traffic(StreetSegId, Flow, Count)*, which refers to the measured traffic flow and aggregate number of vehicles passing some sensor (identified by the attribute *StreetSegId*). Thus, a specific event instance is an instantiation of this predicate with constant values, *e.g.*, *traffic(s187, 4.51, 117)*.

Time is assumed to be linear and discrete, represented by integer time-points. The occurrence of an event E at time T is modelled by the two-ary predicate happensAt(E, T). To reason about the effects of events, we rely on the notion of a *fluent* F, a property that is allowed to

have different values at different points in time. Here, the term $F=V$ denotes that fluent F has value V. Informally, holdsAt$(F=V,\ T)$ represents that fluent F has value V at a particular time-point T. Interval-based semantics are obtained with the predicate holdsFor$(F=V,\ I)$, where I is a list of maximal intervals for which fluent F has value V continuously. holdsAt and holdsFor are defined in such a way that, for any fluent F, holdsAt$(F=V,\ T)$ if and only if time-point T belongs to one of the maximal intervals of I for which holdsFor$(F=V,\ I)$. Table 3 presents the main RTEC predicates.

Fluents are of two kinds: *simple* and *statically determined*. For a simple fluent F, $F=V$ holds at a particular time-point T if $F=V$ has been *initiated* by an event at some time-point earlier than T (using predicate initiatedAt), and has not been *terminated* in the meantime (using predicate terminatedAt). This is an implementation of the *law of inertia*. Statically determined fluents are defined by means of interval manipulation constructs, such as union_all, intersect_all and relative_complement_all (see Table 3).

In our model, the input SDE streams are represented by logical facts that define event instances (predicate happensAt) or the values of fluents (predicates holdsAt and holdsFor). Taking up the example of the SCATS sensor given earlier, facts of the following structure model the input stream:

$$\text{happensAt}(\mathit{traffic}(\mathit{StreetSegId}, \mathit{Flow}, \mathit{Count}),\ T)$$

CEs, in turn, are modelled as logical rules defined over event instances (happensAt), the effects of events (initiatedAt and terminatedAt), or the values of the fluents (holdsAt and holdsFor), and implement the respective temporal and atemporal constraints. For illustration, consider a CE that captures whether traffic flow as given by *traffic* SDE is decreasing. We may capture the CE as a simple fluent *flowTrend* that assumes the value *decreasing* if in two consecutive SDEs (the second one occurring six minutes, that is, 360×10^6 milliseconds, after the first one) there is a drop of more than 10% in the flow value:

$$\begin{aligned}&\text{initiatedAt}(\mathit{flowTrend}(S)=\mathit{decreasing},\ T)\leftarrow\\ &\quad\text{happensAt}(\mathit{traffic}(S,\mathit{Flow},\_),\ T),\\ &\quad\text{happensAt}(\mathit{traffic}(S,\mathit{Flow}',\_),\ T+360\times10^6),\\ &\quad\mathit{Flow}'<\ V-\mathit{Flow}\times0.1\end{aligned}\tag{2}$$

'_' denotes a 'free' variable that is not bound in a rule.

Run-Time Composite Event Recognition Based on the introduced model, run-time CE recognition is performed as follows. The RTEC engine queries, computes and stores the maximal intervals of fluents and the time-points in which events occur. CE recognition takes place at specified query times $Q_1, Q_2, \ldots$. At each query time Q_i only the SDEs that fall within a specified interval — the 'working memory' (*WM*) or 'window' — are taken into consideration: all SDEs that took place before or on Q_i-*WM* are discarded. This constraint ensures that the cost of CE recognition depends only on the size of *WM* and not on the complete SDE history. The size of *WM*, as well as the temporal distance between two consecutive query times — the 'step' (Q_i-Q_{i-1}) — are tuning parameters that can be either chosen by the user or optimized for performance.

The relationships between *WM* and Q_i-Q_{i-1} can be divided into three cases, as follows.

- *WM* $< Q_i - Q_{i-1}$, that is, *WM* is smaller than the step. In this case, the effects of the SDE that took place in $(Q_{i-1}, Q_i - WM]$ will be lost.

- *WM* $= Q_i - Q_{i-1}$. In this case, no information will be lost, *provided that* all SDEs arrive at the engine in a timely manner. If SDEs do not arrive in a timely manner, then the effects of SDEs that took place before Q_i but arrived after Q_i will be lost.

- *WM* $> Q_i - Q_{i-1}$. In the common case that SDEs arrive at the engine with delays, it is preferable to make *WM* longer than the step. This way, it becomes possible to compute, at Q_i, the effects of SDE that took place in $(Q_i - WM, Q_{i-1}]$, but arrived after Q_{i-1}.

Further details on RTEC may be found in [ASP12].

5.9.2 Composite Event Recognition

We present a set of preliminary CE definitions for transport and traffic management. To perform CE recognition with RTEC, each record of the bus dataset was converted to the following facts:

$$\begin{aligned}&\mathsf{happensAt}(move(bus, line, operator, delay),\ t)\\ &\mathsf{holdsAt}(gps(bus, lon, lat, direction, congestion) = true,\ t)\end{aligned}$$

$move(bus, line, operator, delay)$ expresses that bus is running in $line$ with a $delay$ at t, and is owned by $operator$. $delay$ is a possibly negative integer measured in seconds. t is in micro-seconds. $gps(bus, lon, lat, direction, congestion) = true$ additionally states the longitude and latitude of bus, as well as its direction (0 or 1) on the $line$. Furthermore, the gps fluent provides information about congestion (0 or 1) in the given longitude and latitude. Given this input, we developed CE definitions concerning, among others, bus, line and operator punctuality.

A bus is said to be non-punctual if it has a positive delay value attached. The durative $bpunctuality$ CE — denoting 'bus punctuality' — is represented as a simple fluent and defined in RTEC as follows:

$$\begin{aligned}&\mathsf{initiatedAt}(bpunctuality(Bus) = non\_punctual,\ T) \leftarrow\\ &\quad \mathsf{happensAt}(move(Bus, \_, \_, Delay),\ T),\\ &\quad Delay > 0\end{aligned} \tag{3}$$

$$\begin{aligned}&\mathsf{terminatedAt}(bpunctuality(Bus) = non\_punctual,\ T) \leftarrow\\ &\quad \mathsf{happensAt}(move(Bus, \_, \_, Delay),\ T),\\ &\quad Delay \leq 0\end{aligned} \tag{4}$$

A bus starts being non-punctual when a $move$ SDE with a positive $Delay$ arrives. Furthermore, a bus stops being non-punctual upon the arrival of a $move$ SDE with a non-positive $Delay$. The maximal intervals during which a bus is continuously (non-)punctual are computed from rules (3) and (4) by the RTEC built-in holdsFor predicate.

In addition to bus punctuality, we may detect 'line punctuality' — this CE may be defined

as follows:

$$\begin{aligned}&\mathsf{initiatedAt}(lpunctuality(Line) = non\_punctual,\ T) \leftarrow\\ &\quad \mathsf{happensAt}(\mathsf{end}(bpunctuality(Bus) = punctual),\ T),\\ &\quad \mathsf{holdsAt}(bus\_info(Bus) = (Line, \_),\ T),\\ &\quad [\text{there are at least } n \text{ non-punctual buses running in } Line]\end{aligned} \tag{5}$$

$$\begin{aligned}&\mathsf{terminatedAt}(lpunctuality(Line) = non\_punctual,\ T) \leftarrow\\ &\quad \mathsf{happensAt}(\mathsf{end}(bpunctuality(Bus) = non\_punctual),\ T),\\ &\quad \mathsf{holdsAt}(bus\_info(Bus) = (Line, \_),\ T),\\ &\quad [\text{there are at most } n-1 \text{ non-punctual buses running in } Line]\end{aligned} \tag{6}$$

$\mathsf{end}(F = V)$ is a RTEC built-in event that is said to take place at the last time-points of the maximal intervals for which $F = V$ holds continuously. For example, $\mathsf{end}(bpunctuality(Bus) = punctual)$ takes place when Bus stops being $punctual$. $bus\_info(Bus) = (Line, Operator)$ is a simple fluent expressing that Bus is currently running in $Line$ and operated by $Operator$. Note that a bus is not restricted to a single line and operator. According to rule (5), $Line$ starts being non-punctual when a bus running in $Line$ ends being punctual (see the first two conditions of rule (5)), and there are at least n non-punctual buses running in $Line$ (to simplify the presentation, we do not display this constraint in the RTEC syntax). According to rule (6), $Line$ stops being non-punctual when a bus running in $Line$ ends being non-punctual (see the first two conditions of this rule), and there are at most $n-1$ non-punctual buses running in $Line$.

Buses provide information about congestions. Consequently, we may compute the maximal intervals for which a congestion is reported at some location:

$$\begin{aligned}&\mathsf{initiatedAt}(reportedCongestion(Lon, Lat) = \mathsf{true},\ T) \leftarrow\\ &\quad \mathsf{happensAt}(move(Bus, \_, \_, \_),\ T),\\ &\quad \mathsf{holdsAt}(gps(Bus, Lon_B, Lat_B, \_, 1),\ T),\\ &\quad close(Lon_B, Lat_B, Lon, Lat)\end{aligned} \tag{7}$$

$$\begin{aligned}&\mathsf{terminatedAt}(reportedCongestion(Lon, Lat) = \mathsf{true},\ T) \leftarrow\\ &\quad \mathsf{happensAt}(move(Bus, \_, \_, \_),\ T),\\ &\quad \mathsf{holdsAt}(gps(Bus, Lon_B, Lat_B, \_, 0),\ T),\\ &\quad close(Lon_B, Lat_B, Lon, Lat)\end{aligned} \tag{8}$$

(Lon, Lat) are the coordinates of some area of interest, while (Lon_B, Lat_B) are the current coordinates of a Bus. The gps fluent, like the $move$ event, is given by the dataset. $close$ is an atemporal predicate computing the distance between two points and comparing them against a threshold (0.0003 in these experiments). $reportedCongestion(Lon, Lat)$ starts being true when a bus moves close to the location (Lon, Lat) for which we are interested in detecting congestions, and (the bus) reports a congestion (represented by 1 in the gps fluent). Moreover, $reportedCongestion(Lon, Lat)$ stops being true when a (possibly different) bus moves close to the location of interest and reports no congestion (represented by 0 in gps).

Combining SDE from both buses and SCATS sensors allows us increase our confidence

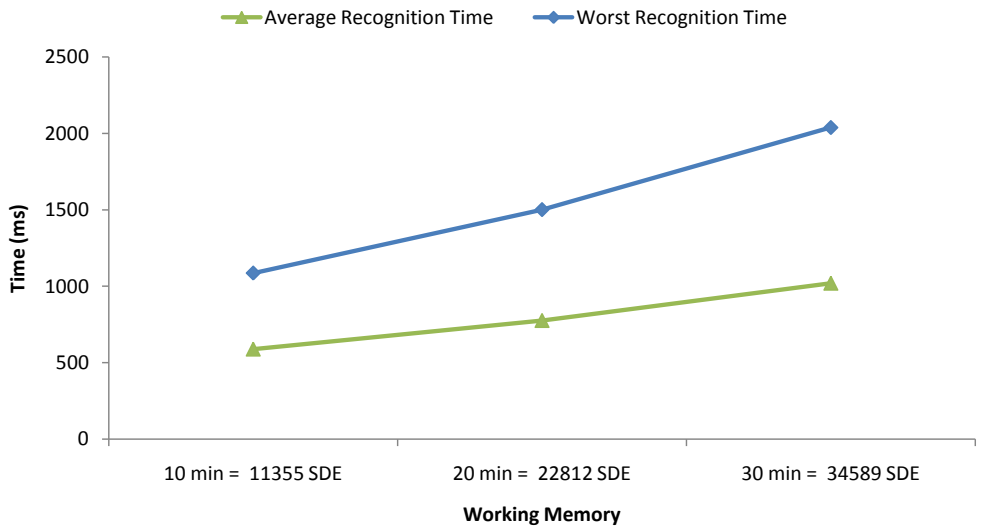

Figure 59: Average and worst CE recognition times on the bus and SCATS datasets: 943 buses, 83 lines, 8 operators and 966 SCATS sensors. Step set to 10 min = 13748 SDE.

that a congestion has actually taken place — consider the formalisation below:

$$\begin{aligned} &\text{holdsFor}(congestionEvidence(Lon_S, Lat_S) = \text{true},\ I) \leftarrow \\ &\quad \text{holdsFor}(reportedCongestion(Lon_S, Lat_S) = \text{true},\ I_1), \\ &\quad \text{holdsFor}(flow(\_, Lon_S, Lat_S) = low,\ I_2), \\ &\quad \text{intersect\_all}([I_1, I_2],\ I) \end{aligned} \tag{9}$$

$flow(S, Lon_S, Lat_S)$ expresses the value of value (low, high, etc) of traffic flow at the location (Lon_S, Lat_S) of SCATS sensor S. Whether the traffic flow is low/high/etc at some SCATS sensor depends on the type of the road in which the sensor is installed — for brevity, we omit the definition of $flow$. According to rule (9), $congestionEvidence(Lon_S, Lat_S)$ is recognised when one or more buses and SCATS S sensor provide 'consistent' information, that is, the buses report a congestion close to the location (Lon_S, Lat_S) of S, while S reports low traffic flow.

5.9.3 Experimental Results

We present initial experimental results concerning the bus and SCATS datasets. The presented experiments were performed on a computer with Intel i7 950@3.07GHz processors and 12GiB RAM, running Ubuntu Linux 12.04 and YAP Prolog 6.2.0. The bus dataset includes all Dublin buses (943), lines (83) and bus operators (8), while the SCATS dataset includes all Dublin SCATS sensors (966). Figure 59 presents the average and worst CE recognition times in CPU milliseconds (ms) for three working memory (WM) sizes: 10 min including on average 13748 SDE, 20 min including 27495 SDE and 30 min including 41437 SDE. The step is set to 10 min. At each query time, RTEC computes and stores that maximal intervals of around 45000 CE. CE recognition was performed on a single processor.

Figure 59 shows that RTEC is capable of supporting real-time CE recognition in this dataset. Moreover, we could have achieved a performance gain by running RTEC in parallel

on different processors.

5.10 Analysis of Traffic Data

In this section we[14] focus on the detection of abnormalities (i.e. not expected values) in traffic flow data obtained by SCATS and correlations with the collected Live Drive radio tweets. In this section we will adress the following:

- How to identify abnormalities.
- How to correlate abnormalities with the tweets.

5.10.1 Locations of SCATS sensors and Tweets

In total 737 tweets were found in Dublin City related with traffic congestion obtained from Live Drive, from February till April 2012. We present the sensors and the tweets locations, in Figure 60. The majority of tweets are outside the central City. At each location more than one sensor and tweet may be obtained. It is also possible to have more than one tweet referring to the same event, as their time distance could be less than half an hour and they could be placed at the same GPS location.

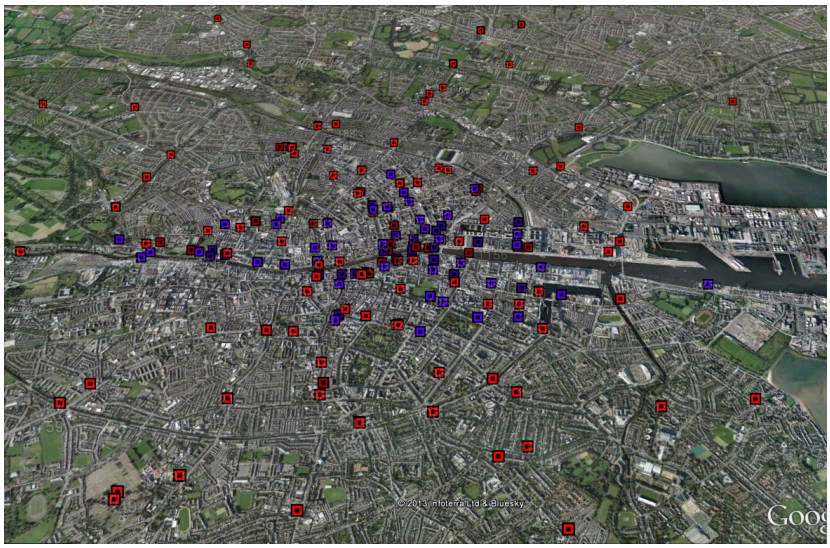

Figure 60: Locations of SCATS sensors (blue) and tweets collected from Live Drive (red) in the central area of Dublin city, from February till April 2012.

5.10.2 What is an Abnormality

Using the aggregated traffic flow value $TF_{i,d,h}$ of each sensor i, at a particular day of week d and at different hours of the day h we calculated the mean $\mu_{i,d,h}$ and the standard deviation

[14] Section provided by UoA

$\sigma_{i,d,h}$ respectively. We defined the abnormalities as the points whose distance from the mean is greater than a scaling of standard deviation:

$$TF_{i,d,h} \geq \mu_{i,d,h} \pm scale * \sigma_{i,d,h} \tag{10}$$

Figure 61 presents the distribution of traffic flow values for a specific sensor at different days and hours, it also shows the calculated mean value and scaling of standard deviation for this sensor in the relevant days and hours. Our method alerts as abnormal all the points that exceed the boundaries created from the scaling of standard deviation.

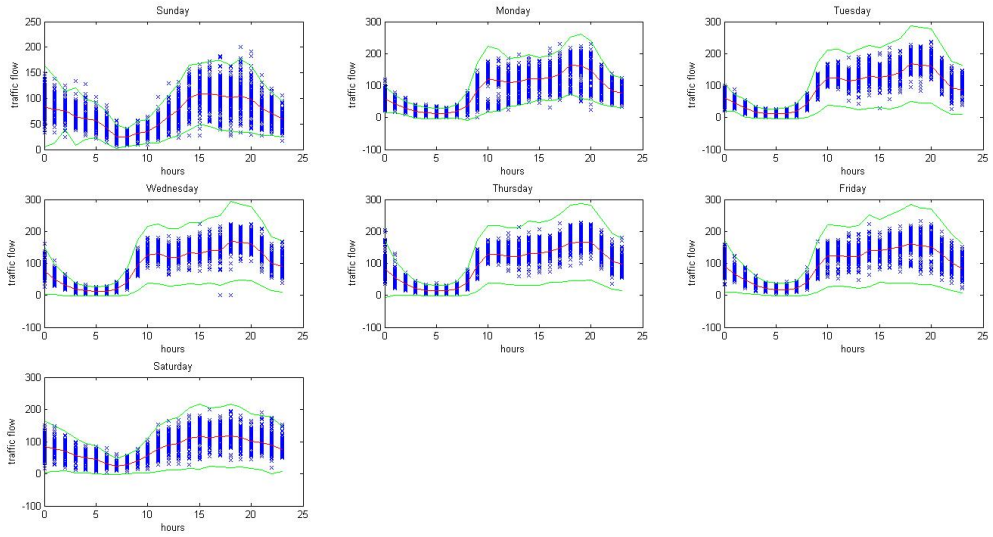

Figure 61: Traffic Flow distribution (blue), the mean value (red) and a scaling of STDV (green), for a particular sensor at different days and hours.

We observed that for several sensors traffic flow values were very noisy, this is why we apply Moving Average in order to make them smoother. Figure 62 presents the time series of traffic flow for a particular sensor at a particular time period before and after applying Moving Average. Also Figure 63 shows the number of abnormalities found from a total set of 5.075.776 values. We calculated the abnormalities for different scaling values (2, 2.5, 3 and 3.5) and Moving Average windows (0, 1, 3 and 6). Window is set to be the past and forward values used in order to calculate the average).

5.10.3 Detecting Connections Between Interesting Tweets and Spatio-temporal Abnormalities

For the set of abnormalities calculated above we identify correlations with the given tweets. We assume that a tweet captures an abnormality in traffic flow if:

1. The spatial distance between the tweet and the abnormality is less than a predefined value.

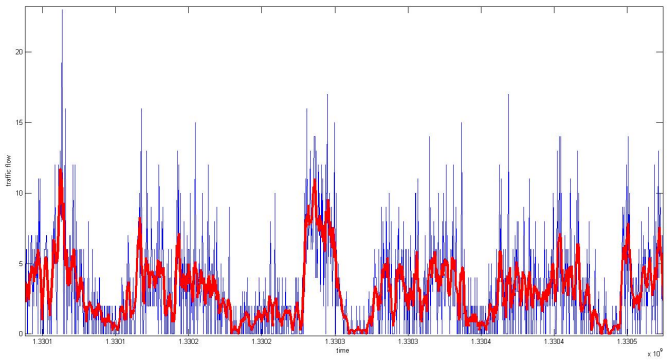

Figure 62: Traffic flow (blue) and transformed traffic flow using Moving Average (red) evolution over time.

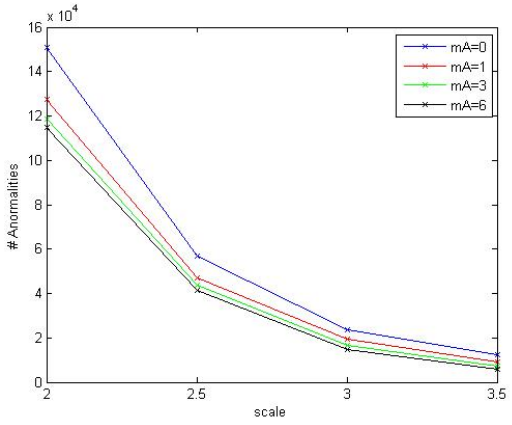

Figure 63: Number of abnormalities for different moving average windows and different scaling values.

2. The temporal distance between the tweet and the abnormality is less than a predefined value.

In order to see whether the results are reliable we add some artificially generated tweets, at the same location of the given tweets, but at random time. Then we run the experiments again. This procedure will help to validate our method's accuracy and observe whether an abnormality on traffic flow values could be explained from a tweet related with traffic congestion in that area.

5.10.4 Identifying the Abnormalities with Classification

In order to find possible relationships between tweets and traffic flow values we currently design Algorithm 1. This algorithm identifies the sensors that are close to the given tweets (as these are candidates for traffic congestion locations). When we run this Algorithm in our dataset the result is an Array with 6 columns (t_0, t_{-1}, t_{-2}, t_{-3}, t_{-4}, label) and 2438 rows (438 identified having traffic congestion and 2000 without traffic congestion). In general we keep

traffic flow values that are in the same time period with Live Drive working hours. The first step of Algorithm 1 normalizes traffic flow values for different sensors (there is high variation in traffic flow values for different sensors). We normalize traffic flow values according to the mean and the standard deviation instead of the maximum value (to avoid outliers that would shrink the normalized values). We did not do the same for the other bound (lower) as there are not such outliers in the lower bound.

Listing 1 Label traffic flow values, using the tweets

Data: Traffic Flow values, Tweets, distance d, time t, number of points to be returned N
Result: Labelled traffic flow values

1. For each sensor normalize traffic flow values, using mean and standard deviation, using the following formula:
$$TF_{i,j} = \frac{TF_{i,j} - min(TF_i)}{max(TF_i)} \tag{11}$$
, where:
$$1 \leq i \leq \text{number of sensors} \tag{12}$$
$$1 \leq j \leq \text{number of measurements for sensor i} \tag{13}$$
$$max(TF_i) = \mu(TF_i) + 3 \times \sigma(TF_i), \text{ for each sensor} \tag{14}$$
2. Find the tweets with distance to the sensor less than d.
3. For each tweet find the closest sensor.
4. For each sensor identified above find the measurement that is temporally closest to the tweet, keep the current and the four previous traffic flow values and label them as having traffic congestion.
5. Select N points and assume that there was not traffic congestion in that area for that time period. These N points will be selected randomly, ignoring the points identified above as having traffic congestion and the points that are t time away from them. Then from the identified points keep the 5 previous values and label them as not having traffic congestion.

We used Weka[15], machine learning software, for classification. Figure 64 presents the normalized traffic flow distribution, as we can see traffic flow is distributed similarly for both the points with and without traffic congestion. At that point we aim to apply several classifiers (i.e. Naive Bayes and Support Vector Machines) in order to observe if there is an underlying structure identified from the classifiers that differentiates the two classes (with and without traffic congestion).

[15]Weka: http://www.cs.waikato.ac.nz/ml/weka/

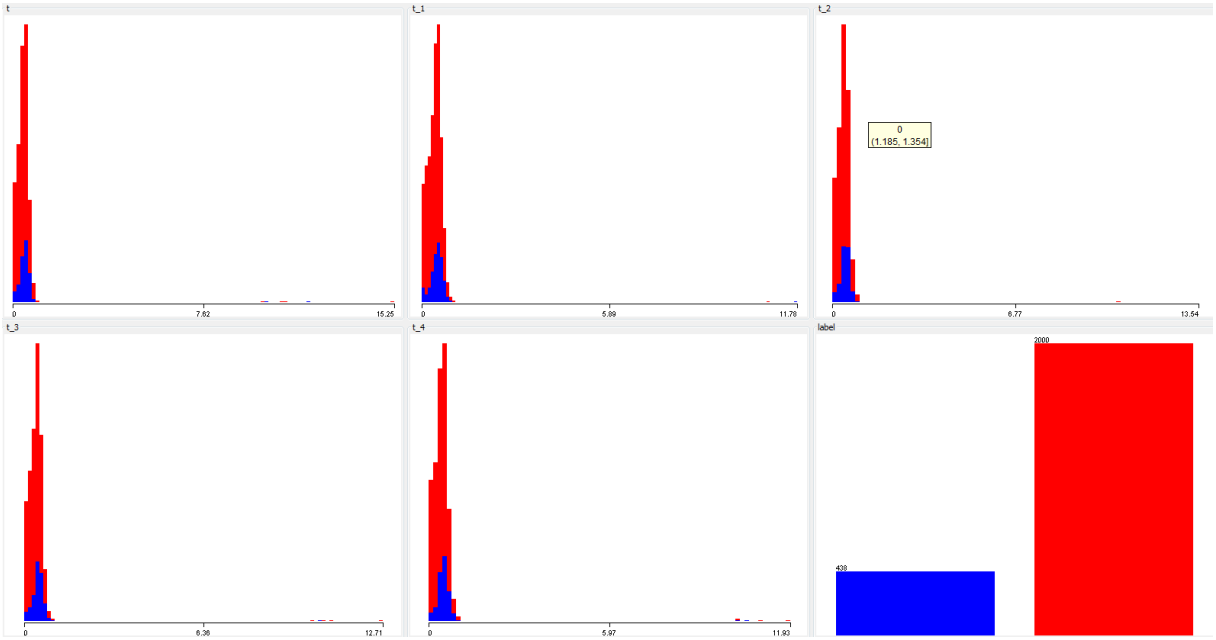

Figure 64: Traffic Flow distribution for each variable

6 Prototype Descriptions

The previous section presented the two use cases of the INSIGHT project and derived tasks for analyses. First approaches to the analyses were presented as well. Now, we describe two prototypes for the two use-case scenarios, nation-wide and city level, which will highlight the integration of the methods developed in this work package (WP5) within the components of complex event processing (WP3) and uncertainty handling (WP4).

6.1 Estimation of Information on Traffic Situations in the City of Dublin, Ireland

The estimation of information on traffic situations (time series of vehicle quantities per junction) is required by the city of Dublin in order to avoid traffic hazards with situation dependent traffic control. We focus on this task in our prototype.

The SCATS data set provides the required information for a small sample of junctions. This poses the task to derive spatio-temporal time series denoting the traffic situation for unobserved locations. To achieve reliable information on traffic situations at unobserved locations, the crowdsourcing application and uncertainty handling from WP3 is incorporated. Available trajectory data from buses is incorporated to resolve uncertainties on traffic situations. The crowdsourcing application delivers the following knowledge:

- annotation of spatio-temporal events (e.g. labelling an event as a jam),
- spatio-temporal time series of quantities of vehicles,
- trajectories of the users.

The estimation of the traffic situation for unobserved locations is a regression task. Our method bases on the assumption that the quantity of vehicles per junction is generated by a stochastic process and any finite combination of these numbers is multivariate Gaussian distributed, thus it is called Gaussian process. The Gaussian process is fully determined by the pairwise covariances among the quantities per junction and their mean. As the junctions are connected by streets of the traffic network and cars have to follow this network, the covariance among two junctions can be expressed by the number and length of jointly passing random walks in the traffic network. However, this does not reflect the anisotropic behaviour of mobility which is caused by individual motivation of mobility. Therefore, we weight preferred combinations (extracted from trajectories) higher and increase correlation among jointly visited junctions [LXMW12]. The covariances among the traffic quantities at the junctions and the measurements at observed junctions are sufficient to compute expectations for unobserved junctions. Our experiments in [LXM13, LXMW12] show that by incorporation of mobility patterns for the traffic quantity estimation the quality of the estimates improves. Additionally, the variance of the estimation can be expressed by the covariances. The locations with highest variance have the most uncertain estimates, crowdsourcing can deliver additional information (spatio-temporal time series of measurements) for these locations. Information from the crowdsourcing application on jams at a junction (spatio-temporal events) can be considered by modification of the traffic network or incorporation of additional measurements.

Consistency among the incoming spatio-temporal time series on vehicle quantities per location (SCATS) and trajectories of the buses (bus data) is checked by complex event processing methods from WP3. Inconsistencies in the data are resolved by crowdsourcing, WP4.

The architecture of the prototype is as follows, compare Figure 65. Whereas data on traffic situations (bus GPS and SCATS measurements) data streams arrive permanently at the system, the event processing component (subject of WP3) validates these streams and detects abnormal behaviour. In response, the crowdsourcing component (subject of WP4) is triggered to get additional detailed information for locations with few or inconsistent knowledge. The aggregated crowd sourced data (trajectories and quantitative measurements of vehicles) in conjunction with the SCATS data stream are combined to model the traffic situation for the city (including also unobserved locations) using the method presented in Section 5.5, compare [LXMW12]. The traffic situations denote the current number of cars for every location in the city. The architecture is integrated in the real-time thread of the lambda architecture and follows specifications of WP2.

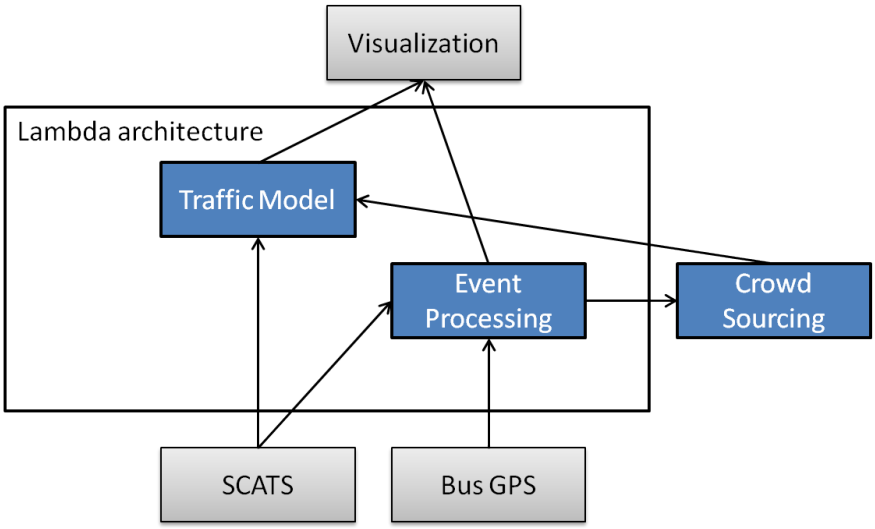

Figure 65: Prototype for Traffic Situation Estimation from Heterogeneous Spatio-Temporal Time Series and Crowd Sourced Trajectory Information.

6.2 Event Reconstruction in Recent Millennium Flood, Germany

Currently, the BBK manually scans incoming data streams for hazardous events and evaluates the risk of these events. The automatic event extraction of the INSIGHT system incorporating spatio-temporal time series (mobile phone usage data and Twitter messages) provides reliable event data and supports early detection of events.

Our prototype focuses on automatic event reconstruction in the test data sets. Considering the data on the recent flood in Germany, we focus in the prototype on the identification of relevant events (e.g. the breech of a dam, the rise of a river tide, or the blockage of a street) from the incoming data streams. The components of the prototype are illustrated in Figure 66. The incoming data streams (mobile phone usage data, Twitter messages and crowd sourced data) are scanned for anomalies using deep learning. This incorporates modeling their normality. Besides the spatio-temporal data and the distribution of the time series the Twitter messages contain useful information which we inspect for its contained information (type of the event addressed, mood of the sender, addressed location). Anomalies detected in one spatio-temporal data stream trigger the incorporation of other streams in order to evaluate whether the conditions for raising an event are fulfilled or not. A task for complex event processing is the derivation of conditions for raising an event from sequences of anomalies and the extraction of properties of an event (event type, time, location from the data streams) from the incoming spatio-temporal time series. The resulting events are reported and visualized in the user interface. The architecture for the real-time processing needs to be scalable, flexible and distributed, moreover addition of new analysis should be easy, thus we use the lambda architecture described in Deliverable 2.1.

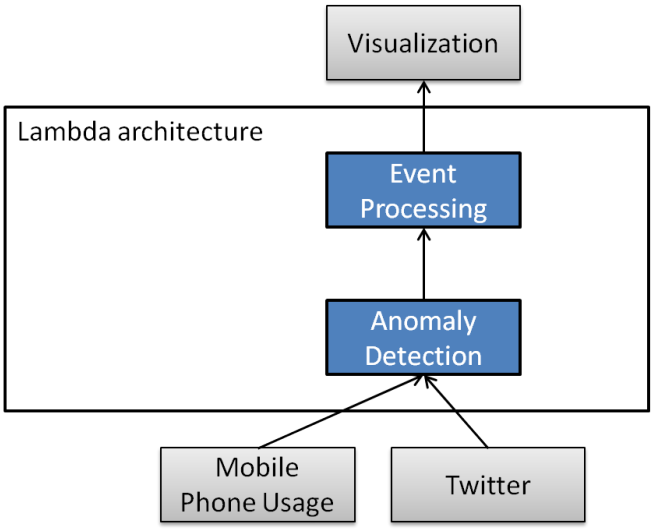

Figure 66: Prototype for Reconstruction of Flooding Events from Heterogeneous Spatio-Temporal Time Series (Mobile Phone Usage Data and Twitter Messages).

7 Summary and Conclusions

The report started with a brief introduction to spatio-temporal data analysis. We have introduced the nation-wide and city level use case scenario.

In the nation-wide use case the automatic detection of flooding events from incoming data streams is of interest. Besides this event detection step, situation understanding and prediction of future situations is important. Available data streams for the nation-wide scenario are the spatio-temporal time series on mobile phone usage and Twitter messages.

In the city level use case traffic events (jams or blockages) and flooding events should be detected automatically. For early event detection and decision support prediction of future situations is important. For the city level use case the following data are available: spatio-temporal time series on traffic flow on few junctions (SCATS), weather information, and text messages (Twitter and Live Drive Radio) as well as trajectories of buses.

These heterogeneous test data sets were described and first analyses have been conducted. The results obtained so far are promising, and besides validation and improvement of the used methods, the integration of these analysis building blocks in prototypes is the next step.

Inspection of data quality has shown that further investigations of the data sets, collection methods and pre-processing have to be done.

Permanent integration of the end-users will reveal more precise requirements, and adoption to their needs in an agile way is crucial. Therefore, a flexible software architecture needs to be defined which is a trade-off between tailoring algorithms to end-user's requirements and using synergies in data stream analysis among the use cases.

Among the streaming environments presented in D2.1: infosphere streams, storm and the streams-framework, the streams-framework of TUDo [BB12b] already comprises methods for data streams [BB12a, Bif13] which can be used for data preprocessing, and anomaly detection analysis.

The complex event processing has to be performed for fast incoming events from heteroge-

neous data sources. We tested the streams-framework for its high-throughput complex event processing capabilities in [GKS+13]. The results are very promising and we are convinced that the streams-framework which may run stand-alone or on top of storm offers a great framework for flexible and sustainable software development in INSIGHT.

References

[AAB+13] Gennady Andrienko, Natalia Andrienko, Harald Bosch, Thomas Ertl, Georg Fuchs, Piotr Jankowski, and Dennis Thom. Thematic Patterns in Georeferenced Tweets through Space-Time Visual Analytics. *Computing in Science & Engineering*, 2013.

[ABKS99] Mihael Ankerst, Markus M. Breunig, Hans-Peter Kriegel, and Jörg Sander. Optics: ordering points to identify the clustering structure. *SIGMOD Rec.*, 28(2):49–60, June 1999.

[ABpKS99] Mihael Ankerst, Markus M. Breunig, Hans peter Kriegel, and Jrg Sander. Optics: Ordering points to identify the clustering structure. pages 49–60. ACM Press, 1999.

[ASP12] A. Artikis, M. Sergot, and G. Paliouras. Run-time composite event recognition. In *Proceedings of International Conference on Distributed Event-Based Systems (DEBS)*, pages 69–80. ACM, 2012.

[BB12a] Christian Bockermann and Hendrik Blom. Processing Data Streams with the RapidMiner Streams-Plugin. In *Proceedings of the 3rd RapidMiner Community Meeting and Conference*, 2012.

[BB12b] Christian Bockermann and Hendrik Blom. The streams framework. Technical Report 5, TU Dortmund University, 12 2012.

[BC08] Mihai Badoiu and Kenneth L. Clarkson. Optimal core-sets for balls. *Comput. Geom.*, 40(1):14–22, 2008.

[Bif13] Albert Bifet. Mining Big Data in Real Time. *Informatica*, pages 15–20, 2013.

[CCMT90] Robert B. Cleveland, William S. Cleveland, Jean E. McRae, and Irma Terpenning. Stl: A seasonal-trend decomposition procedure based on loess (with discussion). *Journal of Official Statistics*, 6:3–73, 1990.

[CM03] Brian Conolly and Ken McCallum. Scats 6 strategic monitor format, 2003.

[DFK12] Wouter Duivesteijn, Ad Feelders, and Arno Knobbe. Different slopes for different folks: mining for exceptional regression models with cook's distance. In *Proceedings of the 18th ACM SIGKDD international conference on Knowledge discovery and data mining*, KDD '12, pages 868–876, New York, NY, USA, 2012. ACM.

[DLB13] Elizabeth M. Daly, Freddy Lecue, and Veli Bicer. Westland row why so slow?: fusing social media and linked data sources for understanding real-time traffic conditions. In *Proceedings of the 2013 international conference on Intelligent user interfaces*, IUI '13, pages 203–212, New York, NY, USA, 2013. ACM.

[EKSX96] Martin Ester, Hans P. Kriegel, Jorg Sander, and Xiaowei Xu. A Density-Based Algorithm for Discovering Clusters in Large Spatial Databases with Noise. In Evangelos Simoudis, Jiawei Han, and Usama Fayyad, editors, *Second International Conference on Knowledge Discovery and Data Mining*, pages 226–231. AAAI Press, 1996.

[FBC02] A. Stewart Fotheringham, Chris Brunsdon, and Martin Charlton. *Geographically Weighted Regression: The Analysis of Spatially Varying Relationships*. Wiley, 2002.

[FGS+13] Hendrik Fichtenberger, Marc Gill, Melanie Schmidt, Chris Schwiegelshohn, and Christian Sohler. Bico: Birch meets coresets for k-means. In Hans L. Bodlaender and Pino Italiano, editors, *Algorithms - ESA 2013*. Springer Berlin / Heidelberg, 2013.

[FJ02] Ana Fred and Anil K. Jain. Evidence Accumulation Clustering Based on the K-Means Algorithm. In Terry Caelli, Adnan Amin, RobertP.W. Duin, Dick Ridder, and Mohamed Kamel, editors, *Structural, Syntactic, and Statistical Pattern Recognition*, volume 2396 of *Lecture Notes in Computer Science*, pages 442–451. Springer Berlin Heidelberg, 2002.

[GKS+13] Avigdor Gal, Sarah Keren, Mor Sondak, Matthias Weidlich, Christian Bockermann, and Hendrik Blom. TechniBall: DEBS2013 Grand Challenge. In *Proceedings of the 7th ACM International Conference on Distributed Event-Based Systems*, page in press. ACM Press, 2013.

[GNPP07] Fosca Giannotti, Mirco Nanni, Fabio Pinelli, and Dino Pedreschi. Trajectory pattern mining. In *KDD*, pages 330–339. ACM, 2007.

[GP08] Fosca Giannotti and Dino Pedreschi. *Mobility, Data Mining and Privacy - Geographic Knowledge Discovery*. Springer, 2008.

[HHS07] Laurence Hirsch, Robin Hirsch, and Masoud Saeedi. Evolving lucene search queries for text classification. In Hod Lipson, editor, *GECCO*, pages 1604–1611. ACM, 2007.

[KAF+08] Daniel Keim, Gennady Andrienko, Jean-Daniel Fekete, Carsten Görg, Jörn Kohlhammer, and Guy Melançon. Visual Analytics: Definition, Process, and Challenges Information Visualization. In Andreas Kerren, John Stasko, Jean-Daniel Fekete, and Chris North, editors, *Information Visualization*, volume 4950 of *Lecture Notes in Computer Science*, chapter 7, pages 154–175. Springer Berlin / Heidelberg, 2008.

[Kri51] Daniel G. Krige. *A Statistical Approach to Some Mine Valuation and Allied Problems on the Witwatersrand*. 1951.

[KS86] R. Kowalski and M. Sergot. A logic-based calculus of events. *New Generation Computing*, 4(1):67–96, 1986.

[Liv13] Live Drive Radio. *Dublin City fm*, Available: http://dublincityfm.ie/livedrive [Last accessed: 27 June 2013], 2013.

[LS08] D. Luckham and R. Schulte. Event processing glossary — version 1.1. Event Processing Technical Society, July 2008. `http://www.ep-ts.com/`.

[Luc02] D. Luckham. *The Power of Events: An Introduction to Complex Event Processing in Distributed Enterprise Systems*. Addison-Wesley, 2002.

[LVGT12] Theodoros Lappas, Marcos R. Vieira, Dimitrios Gunopulos, and Vassilis J. Tsotras. On the spatiotemporal burstiness of terms. *PVLDB*, 5(9):836–847, 2012.

[LVGT13] Theodoros Lappas, Marcos R. Vieira, Dimitrios Gunopulos, and Vassilis J. Tsotras. Stem: a spatio-temporal miner for bursty activity. In *Proceedings of the 2013 ACM SIGMOD International Conference on Management of Data*, SIGMOD '13, pages 1021–1024, New York, NY, USA, 2013. ACM.

[LXM13] Thomas Liebig, Zhao Xu, and Michael May. Incorporating Mobility Patterns in Pedestrian Quantity Estimation and Sensor Placement. In J. Nin and D. Villatoro, editors, *Proceedings of the First International Workshop on Citizen Sensor Networks CitiSens 2012, LNAI 7685*, pages 67–80. Springer, 2013.

[LXMW12] Thomas Liebig, Zhao Xu, Michael May, and Stefan Wrobel. Pedestrian Quantity Estimation with Trajectory Patterns. In *Proceedings of the European Conference on Machine Learning and Principles and Practice of Knowledge Discovery in Databases ECML PKDD 2012, Part II, LNCS 7524*, pages 629–643. Springer, 2012.

[LZZ+09] Yin Lou, Chengyang Zhang, Yu Zheng, Xing Xie, Wei Wang, and Yan Huang. Map-matching for low-sampling-rate GPS trajectories. In *Proceedings of the 17th ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems*, GIS '09, pages 352–361, New York, NY, USA, 2009. ACM.

[MHK+08] Michael May, Dirk Hecker, Christine Körner, Simon Scheider, and Daniel Schulz. A Vector-Geometry Based Spatial kNN-Algorithm for Traffic Frequency Predictions. *Data Mining Workshops, International Conference on Data Mining*, 0:442–447, 2008.

[Nat00] National Imagery and Mapping Agency. Department of Defense World Geodetic System 1984: its definition and relationships with local geodetic systems. Technical Report TR8350.2, National Imagery and Mapping Agency, St. Louis, MO, USA, january 2000.

[PLM13] Nico Piatkowski, Sangkyun Lee, and Katharina Morik. Spatio-temporal random fields: compressible representation and distributed estimation. *Machine Learning*, pages 1–25, 2013.

[RL13] Roberto Rösler and Thomas Liebig. Using Data from Location Based Social Networks for Urban Activity Clustering. In Danny Vandenbroucke, Bndicte Bucher, and Joep Crompvoets, editors, *Geographic Information Science at the Heart of Europe*, Lecture Notes in Geoinformation and Cartography, pages 55–72. Springer International Publishing, 2013.

[Sam69] John. W. Sammon. A Nonlinear Mapping for Data Structure Analysis. *IEEE Transaction on Computers*, 18(5):401–409, may 1969.

[SBDM13] Marco Stolpe, Kanishka Bhaduri, Kamalika Das, and Katharina Morik. Anomaly Detection in Vertically Partitioned Data by Distributed Core Vector Machines. In *Proceedings of the European Conference on Machine Learning and Principles and Practice of Knowledge Discovery in Databases ECML PKDD 2013*, page in press. Springer, 2013.

[SCA13] SCATS. *Sydney Coordinated Adaptive Traffic System*, Available: http://www.scats.com.au/ [Last accessed: 27 June 2013], 2013.

[SD80] A. G. Sims and K. W. Dobinson. The sydney coordinated adaptive traffic (scat) system philosophy and benefits. *Vehicular Technology, IEEE Transactions on*, 29(2):130–137, 1980.

[Sen08] Pavel Senin. Dynamic Time Warping Algorithm Review. Technical Report CSDL-08-04, Department of Information and Computer Sciences, University of Hawaii, Honolulu, Hawaii 96822, December 2008.

[SIR08] SIRI Handbook & Functional Service Diagrams. *KIZOOM*, Available: http://www.kizoom.com/standards/siri/schema/1.3/doc/Handbook/Handbookv15.pdf [Last accessed: 27 June 2013], 2008.

[Tob70] Waldo R. Tobler. A Computer Movie Simulating Urban Growth in the Detroit Region. *Economic Geography*, 46(2):234–240, 1970.

[Tom13] TomTom European Congestion Index. *TomTom*, Available: http://www.tomtom.com/lib/doc/congestionindex/2013-0322-TomTom-CongestionIndex-2012-Annual-EUR-mi.pdf [Last accessed: 26 June 2013], 2013.

[VG12] George Valkanas and Dimitrios Gunopulos. Location extraction from social networks with commodity software and online data. In *Proceedings of the 2012 IEEE 12th International Conference on Data Mining Workshops*, ICDMW '12, pages 827–834, Washington, DC, USA, 2012. IEEE Computer Society.

[VG13] George Valkanas and Dimitrios Gunopulos. A ui prototype for emotion-based event detection in the live web. In *CHI-KDD*, pages 89–100, 2013.

[VGBK13a] George Valkanas, Dimitrios Gunopulos, Ioannis Boutsis, and Vana Kalogeraki. An architecture for detecting events in real-time using massive heterogeneous data sources. In *Proceedings of the 2nd International Workshop on Big Data, Streams and Heterogeneous Source Mining: Algorithms, Systems, Programming Models and Applications*, BigMine '13, pages 103–109, New York, NY, USA, 2013. ACM.

[VGBK13b] George Valkanas, Dimitrios Gunopulos, Ioannis Boutsis, and Vana Kalogeraki. The insight architecture: Detecting and responding to events in real-time with heterogeneous sources. BigMine '13, Chicago, Illinois, USA, 2013.

[Vor08] Georgi F. Voronoï. Nouvelles applications des paramètres continus à la théorie des formes quadratiques. Deuxième mémoire. Recherches sur les parallélloèdres primitifs. *Journal für die reine und angewandte Mathematik (Crelle's Journal)*, (134):198–287, December 1908.

[ZYLG06] Demetrios Zeinalipour-Yazti, Song Lin, and Dimitrios Gunopulos. Distributed spatio-temporal similarity search. In *Proceedings of the 15th ACM international conference on Information and knowledge management*, CIKM '06, pages 14–23, New York, NY, USA, 2006. ACM.