Presenting the Proper Data to the Crisis Management Operator: A Relevance Labelling Strategy

Tommaso Zoppi, Andrea Ceccarelli, Paolo Lollini,
Andrea Bondavalli
Department of Mathematics and Informatics
University of Florence, Viale Morgagni 65, Florence, Italy
{tommaso.zoppi, , andrea.ceccarelli, paolo.lollini,
andrea.bondavalli}@unifi.it

Francesco Lo Piccolo, Gabriele Giunta,
Vito Morreale
Engineering Ingegneria Informatica
Viale Regione Siciliana, 7275, Palermo, Italy
{francesco.lopiccolo, gabriele.giunta,
vito.morreale}@eng.it

Abstract— **The large availability of smart portable devices and the growing interest in developing Internet of Things (IoT) oriented software components make several heterogeneous data available for analysis purposes. In the context of Crisis Management Systems, this means that people owning mobile devices when involved in natural disasters or terroristic attacks may be considered information sources as the classical ones, e.g., sensors or surveillance cameras. Including the information from the citizens in the situational analysis processes comes with two main issues that need to be addressed: i) the source could deliver wrong data (voluntarily or by mistake) that damage the integrity and the correctness of the analysis, and ii) a significant amount of heterogeneous data need to be selected, filtered and aggregated, to provide to the operator a real-time snapshot of the situation depicted using only credible and relevant information. In this paper, we define and implement a relevance labeling strategy able to process information coming from heterogeneous sources aimed at crisis situations and to provide to the human operator all the details he needs. We include provisions for detecting and removing redundancies and misleading data that can slow down or compromise the process and the a-posteriori analysis. The filtering strategy is last applied to events collected for the Secure! crisis management service-based system, showing its application to three scenarios related to real crisis situations happened in the last year.**

Keywords—crisis management system; human sensors; data filtering; relevance labeling; data fusion; Twitter;

I. Introduction

One of the most important tasks that must be performed by public authorities is to take care and ensure safety and security of infrastructures, society and citizens. The management of crisis as for example earthquakes or terroristic attacks consists of "encompass the immediate response to a disaster, recovery efforts, mitigation, and preparedness efforts to reduce the impact of possible future crises" [1]. A strong support to this activity is provided by *Crisis Management Systems* that implement functionalities to sustain and support the different parts of the management process, for example the collection of data, data filtering and visualization strategies, and presentation techniques aimed to help the human operator managing the available knowledge [16].

The interest in researching and developing crisis management systems grew significantly in the recent years, mainly due to the increasing number of information available provided by sensors, including humans (human sensors are both citizens or trained personnel which provide information, e.g., using their smartphones). The set of sensors and humans constitute a very large and heterogeneous source of data accessible through Internet or dedicated paths.

As the number of sources of information and the amount of data we can retrieve in a defined interval of time are growing more and more, the risk of being overloaded, with a dramatic slowdown of the crisis reaction process becomes a real concern. For this reason, before information can be used by a human operator, data need to be processed and filtered to avoid the delivery of wrong information and to become more readable. Nowadays, a huge amount of crisis data comes from the citizens, which can generate Volunteered Geographic Information (VGI [8]) and share them as for example through SMS, Social Media or Apps, supported by crowd-sourcing [20] and crowd-sensing [21] techniques. Crowd-sourcing is the process of obtaining needed services, ideas, or content by soliciting contributions from a large group (crowd) of people, especially from online communities, while crowd-sensing refers to the involvement of a large, diffuse group of participants in the task of retrieving reliable data from a specific field. This information sources make a lot of additional data available, but introduces quality issues that cannot be ignored [2], requiring the definition and the implementation of complex data filtering and aggregation techniques in order to reach a satisfactory credibility confidence [22].

Previous studies and frameworks such as [12], [14] tackle the management of crisis data coming from different heterogeneous sources, trying to figure out common features and merging them into a format that can be presented to the crisis management operator or to authorities. The authorities or the operators that want to i) interpret this large amount of near real-time data, or ii) analyse them for a-posteriori analysis, cannot examine each individual available information. Thus, they need support (as software modules) that implements advanced filtering techniques to show with highest priority the information *labelled as most relevant*. We identify that such *relevance labelling strategy* plays a key role in helping the crisis management operators to focus the attention on the right events.

In this paper we present a data filtering and relevance labelling solution for systems that have to deal with huge amount of data coming from heterogeneous sources that are at least both geographical and time referenced. After presenting the motivations, the structure and the details of this solution, we show the application on the 'Secure!' [13] case study. 'Secure!' is a framework for Crisis Management System that is able to collect and aggregate heterogeneous data from sources like webcams, gyroscopic sensors, social media or linked apps to provide data to the crisis coordination office. The evaluation of this strategy is performed showing three real crisis scenarios happened in Italy while a prototype of the 'Secure!' framework was running, collecting data from the above mentioned sensors.

Section II presents the motivations of our work and the state of the art on *Crisis Management Systems*, focusing the attention on those managing data coming from heterogeneous sources. Section III discusses the target and the structure of the relevance labelling strategy, while Section IV contains details about our instantiation of the strategy. Section V describes a case study, which is used in Section VI to apply the strategy on real crisis scenarios. Section VII concludes the paper.

II. RELATED WORKS AND MOTIVATIONS

There are several crisis management works that deepen the lessons learned facing emergencies and natural disasters: it is possible to find detailed reports about Tahiti earthquake [5], Katrina hurricane [6], or fire episodes happened in Russia [7] in the summer 2010. In most of these contexts, a strong help on managing the consequences of the disasters came from systems built with the aim to collect data from sensors or from citizens, which provided VGI through telephone alerts or posts on social media. As for example, regarding the earthquake that struck Port-au-Prince in January 2010 [7], a live crisis map of Haiti was launched using the Ushahidi [11] platform. Information on the impact of the disaster was initially collected from online sources, including social media channels like Facebook and Twitter, and alerts received by SMS, sent from citizens that wanted to signal their most urgent needs and location. Information coming from all sources were geolocated to build a crisis map that ten days after the earthquake was recognized from the Head of the US Federal Emergency Management Association (FEMA) as the most comprehensive and up to date map available to the humanitarian community.

Since the impact of social networks increased widely in the last years, several studies were conducted in order to understand the best way to fetch data from social network sources for crisis management purposes [9]. The focus was especially on analysing data coming from tweets , a compact source of information that can be easily indexed using the hashtags. However, such data obtained from Social Web feeds can contain "noise", misinformation and bias (which can get amplified by the viral nature of social media) and will require advanced forms of filtering and verification techniques that are not needed only considering data from owned and well-known sensors. Existing algorithms can be seen in [9], where experiments were conducted to assess the capability of such instrument to detect anomalies in the data, allowing the system to discard the untrusted ones. It is also possible to find open source frameworks such as *SwiftRiver* [10], which enables the filtering and verification of real-time data from specific channels such as SMS, Email, *Twitter* and RSS feeds.

Several works describe architectural solutions for crisis management that integrate some of these presented techniques. To the best of the authors' knowledge the most relevant contributions come from [3], [4], [15]: in [3] the authors describe a *Service Oriented Architecture* for planning and decision support for environmental information management that uses real time geospatial datasets and complex presentation tools. Another contribution comes from [4], a framework developed with the target to deliver a reliable tsunami warning message as quickly as possible, a critical activity especially in zones, such as the Asiatic south-east, in which these events happen with a higher likelihood. In [15], instead, the authors define a framework to support the authorities using the social network feeds, trying to geolocate and categorize tweets that follows specific "crisis" trends. In Table I we summarize characteristics of these existing crisis management frameworks with the aim to understand and compare how they face key issues. We splitted the Crisis Management Process in three blocks:

- *Data Collection and Integration*. The operations needed to collect, organize and control the significance and the

TABLE I. CHARACTERISTICS OF EXISTING CRISIS MANAGEMENT FRAMEWORKS

		[3]	**[15]**	**[4]**
Data Collection and Integration	***Sensor Data***	Heterogeneous	Homogeneous	Heterogeneous
	Sensor Type	Vehicle Position, Weather	Tweets	Tide Gauge, Seismology, GPS Ocean Observation, Weather
	Sensor Data Check	Temporal and spatial down-sampling selectively discard data in order to reduce the amount of data transmitted over the network	Only tweets with specific hashtags, with relevance weights	A module acts as filter reducing the data coming from the sensor streams to relevant data only
	Data Integration	Added-Value services layer	Not needed (homogeneous sensors)	Tsunami Service Bus (TSB)
Information Processing	***Event Correlation***	Adding value to real time environmental data, predicting future states, providing operational guidance, …	Finding Geospatial links for each considered tweet (if not existing in the message, got from user's location)	Sensor integration from Tsunami Service Bus (TSB), including post processing and quality checks
	Event Relevance Labeling	Missing.	Missing	Missing
Human Interface	***Presentation Techniques***	Web Interface allows for a universal access and greatly reduces learning time and thus attracts more non-GIS professionals	Browser Presentation component: uses web services to retrieve and displays information to a watch officer	Not provided: need of external applications that fetch data from Tsunami Service Bus (TSB)

integrity of data coming from sensors.

- *Information processing*. Once data is structured and collected, operations to transform raw data into actionable information are conducted.
- *Human Interface*. The set of techniques built to present the outcomes to the process to the crisis management operator.

Only few studies ([3], [4]) focus the attention on problems generated by the simultaneous usage of multiple and heterogeneous sensors. This choice leads to obtain more accurate information but needs of integration and advanced information filtering techniques applicable to the whole set of collected data, and not independently for each data flow (as for example in [10]).

Moreover, while in most of these frameworks the analysis of the data coming from sensors, the integrity check and the integration are performed very carefully by dedicated software modules, it seems difficult for a human operator to examine these huge sets of correlated data in a reasonable time. For example, in [4] the authors describe all the available data streams that can be retrieved from sensors (e.g., water height control, meteorological feeds), but the data analysis needed to understand relations amongst events coming from different streams is completely left to the human operator. We can also observe how all the works reported in Table I (see *Event Relevance Labelling* category) do not focus the attention on selecting the most promising information to be shown to the operator, assuming that data-level integrity and filtering mechanisms have already checked their validity.

In such a context, it is difficult to understand the relevance of a single value, because it depends on the scenario the operator is interested in: if at the same time several events happen in different places and the operator is interested only in observing one subset of them, he must have the opportunity to quickly choose the specific scenario without manually selecting the relevant data, saving key amounts of time. Moreover, in most of the cases the operator does not have enough information to write a query with detailed parameters (e.g., specifying a restricted area of a city or an exact time window), resulting in an output set that is polluted with undesired events. It is important to remark that this relevance labelling strategy does not aim to discard the data that are not belonging to the relevant set, but only to extract and present to the user the most significant items digging in the huge knowledge we collected using a wide variety of sensors, because the same information could be relevant in a context and less relevant in others. Moreover, this support must be provided in near real-time [28], since in most of the cases the public authorities have to react effectively.

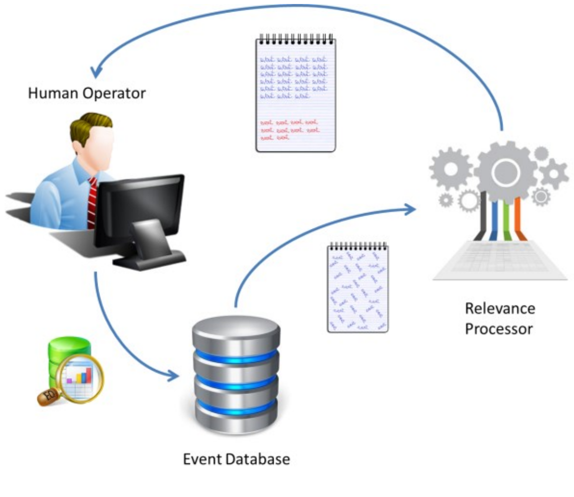

Fig. 1. Interaction between the operator and the system

III. Event Relevance Labelling

We focus the attention on an operator that wants to analyse the collected data following two targets: i) *live observation,* analysing events that happen in real-time aimed to control the current state of the environment, and ii) *historical studies*, observing the database aiming to understand past sequence of events that may repeat in the future. In both approaches, the operator scouts large set of events from a data collection system and is interested in detecting the main subset of correlated events that can point to a dangerous situation.

As shown in Figure 1, a strong support on the relevance labelling activity comes from the possibility to automatically analyse the set of retrieved data and organize them depending on the parameters of the involved events. The human operator queries the *Event Database* for stored data and the data management system answers with all the events that match the search query. These events are sent to the *Relevance Processor* (see Figure 1), which labels each event with a relevance score; its output is delivered to the operator, which now can analyse the ordered set of data following the relevance score and not the data retrieved from the database.

A. The Relevance Criteria

To build the relevance labelling strategy it is required to define a *relevance criteria*, to judge each event according to rules and label it with a relevance score resulting from the application of the rule. The definition of that rule is mandatory to tune the relevance labelling algorithm itself, and must be chosen thinking about penalizing events that are predicted less relevant for the operator of our system. Given a data collection system which retrieves data from heterogeneous sources, we assume that during a crisis the sensors will start to generate a higher number of critical data with common values for some features, for example the temporal or spatial ones. As for example, during an earthquake gyroscopic sensor alerts and tweets referred to that area can raise in number, overcoming the others that come from near regions at the same time. In this situation, the operator could be interested in focusing his attention on this group of related events, analysing them and eventually activating reaction strategies (these may range from alerting the authorities to dispatching and guiding intervention team; reaction strategies are not explored in this paper).

In other words, the operator is interested in understanding if in a large set of events it is possible to detect a smaller subset of them that are linked following some correlation rules. Formally, we are looking for a relevance labelling function *rlf* :

$$S = rlf\,(CE,\ \Omega)$$

which takes an input set *CE* of critical events and a generic set of parameters **Ω** related to the chosen implementation strategy. The function defines a set of scores that indicates what is the relevance in the context of the user query for each event in the starting set CE. This is a general formulation of the problem that can be extended depending on the characteristics of the context, and needs only an event input set in association to the **Ω** parameters. The result is a technique with a wide range of applicability, which fetches data from the framework dataset and returns a set of relevance scores that can be used to improve the quality of the already existing framework's presentation strategies.

IV. Implementation of Our Relevance Labelling Solution

We explain below our relevance labelling algorithm that implements the *rlf* function. In our implementation the general purpose function above was extended to become a function *lf*

$$[R,\ NR,\ W] = lf\,(CE,\ \alpha,\ \delta)$$

which taking an input set *CE* of critical events, an acceptability threshold *α* and a sensitivity parameter *δ*, replies with a triple that represents a partition of the input set. Each event in *CE* is labelled as relevant (*R*), non-relevant (*NR*) or wrong (*W*), depending on the chosen correlation rules. More in detail, the discarded event set *W* will contain all the events that have erroneous or incomplete values for some of their features (missing timestamp, wrong geo coordinate values, etc.) while the relevant set *R* will be the largest set of data with common characteristics, representing the predicted subset of more critical events in the starting set. All the other events will be labelled as non-relevant (*NR* group).

A. Involved Techniques

Here follows the listing of the main techniques used to build our solution. It is important to highlight that we are considering event information coming from heterogeneous sources but having common features: in particular, we are assuming that each event is *at least* geo-located and time-referenced.

Integrity Check: the target is to detect events that have incomplete set of values, wrong or missing fields or that have integrity problems. The aim is to avoid pollution of data due to events that are not valid, due to malfunctions of the sensors, of the database or damaged from adversaries during the delivery to the data center. Events with these problems are stored in the *W* set and not considered for further analysis. In addition, an alert is sent to the database administrator, who needs to fix the detected problem, updating or discarding the information.

Mean Value: we chose a very simple and fast algorithm to verify the dispersion of the event in the space calculating the mean value for each common feature (timestamp, latitude, longitude). After this computation, for each involved event, and for each of its features, we check if the value is close enough to the mean for that parameter in a range defined by a sensitivity parameter *δ*. Formally, let *CS* be the set of events with common features F and let e_f be the value for the feature *f* (in *F*) of the event *e* (in *CS*). For each event *e*, we will check the following statement,

$$\forall_{f \in F}\ \left|mean(f) - e_f\right| < \delta_f$$

where $\delta = \{\delta_f \text{ in } Real \mid f \text{ in } F\}$ contains the tolerance values for each feature *f* in *F*. It follows that an event is valid for this algorithm if and only if the values of all its features are in the mean-range of the corresponding *δ* value.

K-Medoids Clustering: considering specific event sets, it is possible to have subsets of elements with common characteristics inside the bigger group. Since the mean value step is not able to detect and classify these subsets, we chose the *K-Medoids* clustering algorithm [17] (an evolution of *K-Means* [18] one), which partitions the investigated events into *k* clusters by selecting *k* events as leaders (medoids) and assigning all the other events to the neighbourhood of the closest leader. This choice of a leader event (medoid) instead of leader coordinates (centroid) makes *K-Medoids* less sensitive than *K-Means* to outliers, keeping a similar efficiency in terms of computational time [19].

B. Building the Process

These techniques interact each other following a specific flow (depicted in Figure 2) that aims to obtain the better result in terms of the highlighted relevant set. The first performed step is the *integrity check*: we need to rule out the events that have values so different from the average of the others or incomplete to be considered wrong in such a context. These constitute the set of events *W*. On the cleaned event set, the *mean value* algorithm is executed to understand how many events are close enough to the mean for each of their features. If this number is above the chosen acceptability threshold *α,* these events build the relevant set *R*, while the others are placed in the *NR* set.

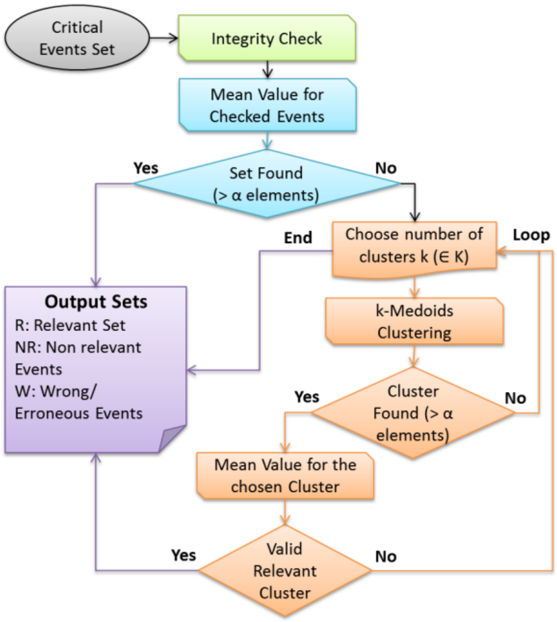

Fig. 2. Relevance labelling strategy

If the number of mean-range relevant events is under the *α* threshold, we need to perform the *k-medoid clustering* step, which tries to split the whole set (cleaned from the events that do not pass the integrity check) into *k* clusters to identify one that have at least *α* events with feature values close to the cluster mean ones. Depending on the chosen *k* value from *K* set (we consider a set of values *K* to increase the chances of success), we perform the *k-medoids clustering* and, if the process has success, we check if it is possible to find a cluster that have at least α elements. If this cluster exists, on that set we run again the *mean value algorithm* to determine if this set can be seen as the relevant set. If both the initial mean value algorithm and the clustering steps are not able to find a relevant set, the process ends without giving relevance information to the user, but only showing the events *W* judged non-valid from the integrity checker separately from the other ones. In this case, the user needs to change the query parameter's values or add other query filters.

It is important to highlight the significance of the sensitivity δ and threshold α. When the operator queries the database, he may be interested in capturing specific events that would require a smaller sensitivity value to be captured. If an operator is observing an extended area, a small value of δ could result useless for relevance labelling, because it is very difficult to detect an high percentage (defined from α) of events with very similar features in a wide area. The α parameter, instead, defines the minimum number of events in the relevant set. In our experiments we want to detect a single block of related events (the R group): consequently we set this value to 50%. It should be remarked that we could lower this value to observe more subsets of events. This is a key improvement that will be discussed as a future work at the end of this paper.

V. Application on a Crisis Management System

The Secure! system [13] is a *Decision Support System* (DSS, [14]) for crisis management that exploits information retrieved from a large quantity and several types of data sources available in a target geographical area, in order to detect critical situations and command the corresponding reactions including guiding rescue teams or delivering emergency information to the population via the Secure! app. Input data to Secure! come from: social media as for example *Twitter*, Secure! apps (apps that can be installed on smartphones and tablets and can be used by civilians or trained personnel to send data to Secure!), web sites and feeds, and sensor networks available in the infrastructures (e.g., surveillance cameras, proximity sensors).

A. Micro and Macro Events

Data are received, collected, homogenized, correlated and aggregated in order to produce a situation for the DSS system that is ultimately shown to operators in a control room which take the appropriate decisions. First, data are collected from the heterogeneous set of sensors, processed to extract basic information called *micro event* (e.g., a gun recognized in a photo)*;* then, if it is possible, these basic events are correlated depending on the values of common features (e.g., similar spatial or temporal coordinates). The resulting item is the *macro event*, or *Secure! event,* composed from several *micro events*, which represents the critical situation at a certain time and in a certain place (e.g., the status of a demonstration in a part of the city). A basic event taxonomy contains the catalogue of micro-events [23] which represents simple real events involving one category of entity (e.g., people detection, fire presence, recognition of abrupt sounds as explosions, guns firing). Hence, micro events contain the texture description of the real event, the related timestamp, the involved entity and the source that generated it. Sample sources used in Secure! are *tweets*, surveillance cameras and gyroscopic sensors.

When a critical situation happens, several micro events are generated, and afterwards correlated producing a set of macro events, which can identity a specific situation through time. On the basis of the information contained in the macro events, recovery actions can be planned. For example, when a brawl between soccer supporters happens, the presence of weapons and anomalous people behaviour, detected by sensors or humans reporting information, are identified as micro events. These micro events can be correlated using spatial and temporal data in one or more macro events that describe the current status of the brawl. These macro events can be used in a crisis management system to support intervention, directing police towards the most dangerous crowds of supporters.

B. Data collection and Storage

The Secure! system allows to find out heterogeneous data streams using the *Source Integration Framework* (SIF), a set of services that provides the management of the sources and the extraction of data and metadata (e.g., text, video, image, audio). SIF is composed by four low level modules that work on a n-modal data source; for example in *Social Media* and *Web* wrappers, *WebCrawler* and *SMCrawler* modules extract data to identify persons, companies, cities and other types of entities from HTML document, a text or in general Web content. Differently, for sensor network and mobile apps, *SWInterface* and *ADInterface* modules represent an important interface between data sources and Secure! framework. The heterogeneous data sources are managed using a Wrapper, a program that allows decoupling high and low I/O modules and normalising output using predefined schema known cartridge (e.g. RDF, XML). The data stream extracted from information sources is automatically analysed from a process whose goal is to manage analysis services on resources (e.g., filtering, validation and events domain).

C. Integration of our solution in Secure!

The Secure! *Integration Framework Architecture* is based on the SOA paradigm, with a *RestFul Web Services* approach that allows us to define a set of unambiguous identifiers that support I/O Interface (XML, JSON) and the canonical HTTP operations. The most important web service is the *getEventWithAndCondition*, which allows advanced search on *macro-events* stored on the Secure! database, offering a collection data output with JSON mediatype. The operator could query the dataset adding filters on the feature values of the events: for example, it is possible to ask for events generated in a restricted interval of time [t_1, t_2] adding the clauses *fromDate=t_1* and *toDate=t_2*.

Our solution depicted in Figure 1 is implemented in Java and integrated in Secure! as a *RESTful* service: when the operator performs a query on the Secure! database using the abovementioned service, the extracted events are sent to the *Relevance Processor*. It executes the strategy depicted in Fig. 2 and returns to the operator the three sets of events R, NR, W. All the communications are encrypted using a standard SSL certificate owned by the operator and checked before every transmission that involves the relevance processor module.

VI. EXERCISING OUR SOLUTION IN SECURE!

A. *Scenarios*

The Secure! system has been subject of an experimentation process whose goal is to endorse framework and components in a real context. In particular scenarios are:

- *Europa League Match*: clashes and vandalism between supporters and police in Rome occurred before the Roma - Feyenoord European football match. 68 macro events were collected and stored in the database regarding this happening (18th and 19th February 2015);
- *Political Manifestation*: clashes between police and violent members of Italian political factions (*Lega Nord*, *Casapound*) at a manifestation in Rome on the 27th and 28th February 2015 (52 macro events);
- *Weather Warnings*: intense atmospheric events (strong winds) raged in Tuscany on the 4th and 5th of March 2015. The sensors and the *micro event* correlation techniques in Secure! made 65 *macro events* available in the abovementioned time window.

In Table II it is possible to observe some details related to the scenarios. We highlighted the temporal clauses that the operator could use, in addition to a textual geographic description (Rome, Tuscany), to get this event set. We also reported the resulting query and the setup of the sensitivity parameters $\delta = \{\delta_{lat}, \delta_{lon}, \delta_t\}$ that we used to run the experiments reported below. We choose the same δ for each scenario to show how the same sensitivity is suitable in a scenario and needs of a different tuning in another, although a detailed sensitivity analysis of the parameters α, δ is outside of the scope of this paper and will be performed as future work. Due to the geographical dimension of the considered areas, localization uncertainty of the events, which may be up to several meters [29], is not significant for our analysis and consequently not considered in this work.

B. *Application of the Relevance Labelling Strategy*

We applied our strategy on each scenario, to understand how our solution is able to help the human operator to identify the main block of related critical events. The following evaluations are a-posteriori analysis, since during the collection phase of the data related to the three scenarios our relevance labelling solution was not implemented. However, at the current status of the work, the solution is integrated and working inside the Secure! framework.

With the support of the generated log files, which includes the macro-event and the computed relevance, we summarized these results with the graphical support of Google Maps, still keeping in mind the third dimension of analysis, which is the temporal one and cannot be graphically shown in Figure 3 and Figure 4. The red pin markers in Figure 3 and Figure 4 point to the geo-location of the events belonging to the relevant set R, while the yellow circles highlight the less relevant ones NR. In Table III it is possible to observe the results of the relevance labelling algorithm for the considered scenarios.

TABLE II. SCENARIOS AND PARAMETERS DETAILS

Scenario	**Query Clauses**		**δ**		
	fromDate	**toDate**	δ_{lat}	δ_{lon}	Δ_t
Europa League Match	18th Feb	19th Feb	0.06	0.06	4h
query:getEventWithAndCondition?Rome&fromDate=18022015&toDate=19022015					
Political Manifestation	27th Feb	28th Feb	0.06	0.06	4h
query:getEventWithAndCondition?Rome&fromDate=27022015&toDate=28022015					
Weather Warnings	4th Mar	5th Mar	0.06	0.06	4h
query:getEventWithAndCondition?Tuscany&fromDate=04032015&toDate=05032015					

C. *Europa League Match in Rome*

As it is possible to observe in Figure 3, the macro events for this scenario refer to the area of Rome nearby the stadium, which was involved by the football match, but most of them are located in other two separate areas of the city. In that case, the relevance labelling algorithm considers all the retrieved events as valid, without detecting any integrity or value problems, and then tries to execute the mean algorithm, which in this case labels the 24% of events as *R* group. Since our threshold α is set on 50% of the whole *CE* group, this partial result has to be discarded, and the clustering step becomes necessary. The clustering process reveals that is possible to consider four distinct subsets of events, where the biggest one is composed by the 63% of the events of the starting set, all referred to the area of *Campo de' Fiori*, where some collisions among policemen and Feyenoord supporters occurred [25].

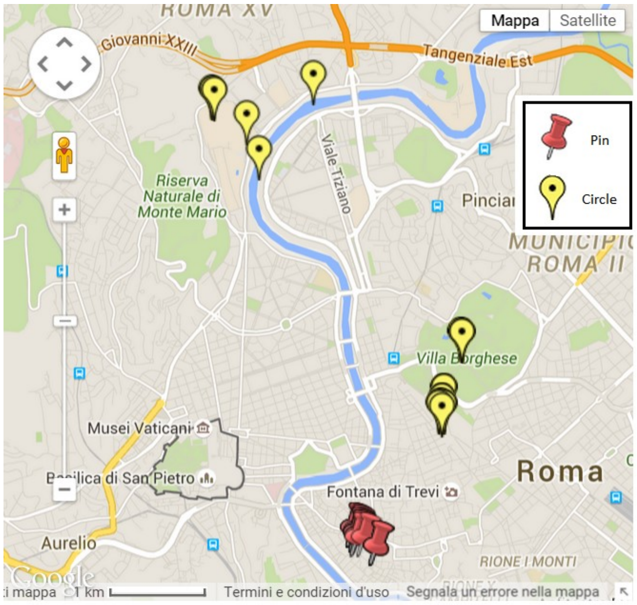

Fig. 3. Geo-location detail for events in "Europa League Match" scenario.

TABLE III. RELEVANCE LABELLING FOR THE CHOSEN SCENARIOS

Scenario	R	NR	W	Label
Europa League Match	43	25	0	Clustering Successful
Political Manifestation	32	18	5	Mean Successful
Weather Warnings	-	-	0	Unsuccessful Labelling

The others are related to events i) happened at the stadium while the match was being played, ii) related to the authorities alerts delivered in the morning that pointed to the more probable dangerous zones and iii) alerts linked to the vandalisms at the *Barcaccia* fountain [24]. Since the main cluster contains more than the 50% of events and all of them are in the range of the mean value calculation restricted to this cluster, the relevance labelling marks all the events in this cluster (the red pins in Figure 3) as R, so these are the events that go to the attention of the human operator with priority.

D. Political Manifestation in Rome

In this scenario, the outcome of the relevance labelling algorithm follows a different flow compared to the previous case study. First of all, 5 events (9% of the retrieved ones) are detected with feature values that are not valid (*W* set): an in-depth view of the raw data shows us that the latitude and longitude values are set to a *null* value, meaning that the geo-localization for that events coming from *Twitter* encountered some problems. Once these events are discarded, the mean value step find a set large enough to be considered the relevant one, composed by the 59% of the events in the starting set and labelled in Figure 4 as red pins. Looking at the image we can see that while most of the relevant events are grouped together, one seems not belonging to the same category. As abovementioned, the algorithm do not use only spatial parameters, and this event is considered relevant as well as the others even if it is located in a different position because its timestamp value is very close to the other red ones, that were created after the manifestation. The other events, marked with yellow circles in the figure, are related to the procession and the following political meeting, which did not create any problem to the authorities and happened before the clashes between demonstrators and police officers. It is important to remark that is possible to reduce the tolerance of the mean algorithm changing the sensitivity parameter δ to consider as related only events with more (or less) similar feature values.

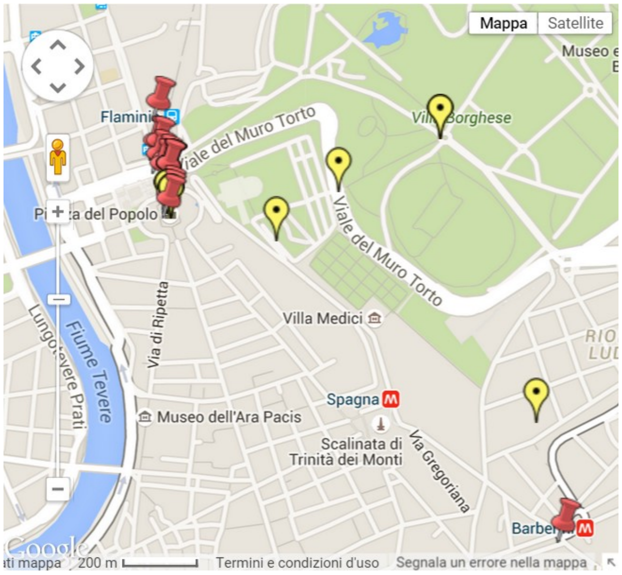

Fig. 4. Geo-location detail for events in "Political Manifestation" scenario.

E. Weather Warnings in Tuscany

The last considered scenario allows us to explore the case in which our strategy is not able to detect a relevant subset of the starting group. Since the 65 weather warnings collected in this scenario are located in different cities of the *Tuscany* region and in different time slots of the day, neither the mean algorithm nor the clustering converge to a relevant group. In these contexts, the operator has to investigate a smaller area of Tuscany, consider a restricted time window or add new clauses focusing the attention on a more specific scenario. From the point of view of the relevance labelling strategy, when the retrieved events are splitted in more than two smaller clusters, it is needed to lower α value. This requires an update of the algorithm that we planned to complete as future work.

F. Algorithm Performance

To evaluate the capabilities of the algorithm both in terms of labelling speed and effectiveness, we conducted experiments using the Secure! *macro events* collected in the first 6 months of the 2015 using *Twitter*, 3 gyroscopic sensors (placed in *Piazza dei Miracoli* in *Pisa*) and authorities' security alerts. In the tests we simulate a large group of queries that an operator can run on the database, varying the values for the query clauses *fromDate* and *ToDate* and calling the service in different dates and with different time window size. For each run of the tests, the output of the relevance labelling strategy was saved for statistical analysis purposes. We conducted the experiments on a machine with an *Intel(R) Core(TM) i7-4510U* CPU @ *2*.00GHz, 8GB RAM and each test is repeated 10 times keeping the system unaltered to optimize repeatability [26] conditions of each run.

In Table IV we can observe the average results of these tests, classified based on the outcomes of the relevance labelling process. As already described in Section IV, the outcome can be i) successful, using only the mean algorithm or with clustering support, ii) or not, when seems not possible to identify a relevant subset with at least α events. In each experiment, we measured the time spent for the labelling activity, assuming that the data are already available on the machine and provided as input to the service. This allows evaluating the performances of our strategy without considering delays due to the event fetching policy, which can vary depending on the location of the operator (same machine of the database, query trough LAN, etc.). In Table IV we can notice that our strategy is able to identify a relevant subset for about 2/3 of the possible user queries in the abovementioned

TABLE IV. RELEVANCE LABELLING PERFORMANCE

Outcome Label	Outcome Result %	Time (*ms*)		
		Mean	**Clustering**	**Total**
Mean Successful	*50.0*	*1.87*	*Not Needed*	*1.87*
Clustering Successful	*16.7*	*4.20*	*14.63*	*18.83*
Unsuccessful Labelling	33.3	3.50	22.85	26.35
All	**100.0**	**2.80**	**20.11**	**12.86**

portion of the Secure! database, and half of the cases are managed using the mean algorithm, which is fast and give responses at average in less than two milliseconds. Otherwise, the clustering step becomes mandatory and the relevance labelling time increases a lot, reaching its maximum when the process has no success: in this case we execute the mean algorithm and the clustering step, trying all the possible combinations and stopping the process only at the end.

VII. Conclusions

This paper presented the design, development and assessment of a relevance labelling strategy for crisis management systems that can retrieve and process huge amounts of data coming from heterogeneous sources. The aim is to provide a relevance labelling strategy that supports the human operator to decode and classify the alerts delivered to the decision making center of a crisis response team, assigning on each event a label that predicts its relevance for the user. Events with higher relevance scores will be presented before the others, helping the operator to quickly clarify the context to maximize the effectiveness of the response.

Our future work will include further analysis aimed to understand the influence of the threshold α and sensitivity δ parameters. Considering α values lower than 50% will allow us to get different relevant subsets of the starting group *CE*, managing crisis situations in which we have to take notice of simultaneous critical events. An effective tuning of the δ parameter, instead, will allow tailoring the relevance labelling algorithm to the needs of the operator, e.g., to narrow the area of interest. He could specify the granularity of the analysis that is suitable for his purposes and, afterwards, the value of the δ parameter. Since the development of the component in the Secure! framework is now complete, further analysis will be conducted online, executing the relevance labelling tool as an integrated module.

Acknowledgment

This work has been partially supported by the Regional Project POR-CREO 2007-2013 Secure! and the TENACE PRIN Project (n. 20103P34XC) funded by the Italian Ministry of Education, University and Research.

References

[1] Mehrotra, Sharad, Taieb Znati, and Craig W. Thompson. "Crisis management." IEEE Internet Computing 1 (2008): 14-17.

[2] Elwood, Sarah. "Volunteered geographic information: future research directions motivated by critical, participatory, and feminist GIS." GeoJournal 72.3-4 (2008): 173-183.

[3] Vescoukis, Vassilios, Doulamis, Karagiorgou. "A service oriented architecture for decision support systems in environmental crisis management." Future generation computer systems (2012) 593-604.

[4] Fleischer, J., et al. "An integration platform for heterogeneous sensor systems in GITEWS–Tsunami Service Bus." Natural Hazards and Earth System Science 10.6 (2010): 1239-1252.

[5] Zook, Matthew, et al. "Volunteered geographic information and crowdsourcing disaster relief: a case study of the Haitian earthquake." Available at SSRN 2216649 (2010).

[6] Farazmand, Ali. "Learning from the Katrina crisis: A global and international perspective with implications for future crisis management." Public Administration Review 67.s1 (2007): 149-159.

[7] Meier, Patrick. "New information technologies and their impact on the humanitarian sector." International review of the Red Cross 93.884 (2011): 1239-1263.

[8] De Longueville, Bertrand, et al. "Digital earth's nervous system for crisis events: real-time sensor web enablement of volunteered geographic information." International Journal of Digital Earth 3.3 (2010): 242-259.

[9] Wong, Mr Weng-Keen, et al. "WSARE: what's strange about recent events?" Journal of Urban Health 80.1 (2003): i66-i75.

[10] SwiftRiver Project, https://github.com/ushahidi/SwiftRiver [last accessed 28th Oct. 2015]

[11] Ushahidi Project, http://www.ushahidi.com/ [last accessed 28th Oct. 2015]

[12] Annoni, Alessandro, et al. "Orchestra: developing a unified open architecture for risk management applications." Geo-information for disaster management. Springer Berlin Heidelberg, 2005. 1-17.

[13] Secure! Project, http://secure.eng.it/ [last accessed 28th Oct. 2015]

[14] Peng, Yi, et al. "An incident information management framework based on data integration, data mining, and multi-criteria decision making." Decision Support Systems 51.2 (2011): 316-327.

[15] Cameron, Mark A., et al. "Emergency situation awareness from twitter for crisis management." Proceedings of the 21st international conference companion on World Wide Web. ACM, 2012.

[16] Jennex, Murray E., and Murali Raman. "Knowledge Management in Support of Crisis Response." International journal of information systems for crisis response and management 1.3 (2009): 69.

[17] Kaufman, L. and Rousseeuw P.J. (1987). Clustering by means of medoids, in: Y. Dodge (ed.), Statistical Data Analysis Based on the L1-norm and Related Methods. Elsevier, North-Holland, pp. 405-416.

[18] MacQueen, James. "Some methods for classification and analysis of multivariate observations." Proceedings of the fifth Berkeley symposium on mathematical statistics and probability. Vol. 1. No. 14. 1967.

[19] Park, Hae-Sang, and Chi-Hyuck Jun. "A simple and fast algorithm for K-medoids clustering." Expert Systems with Applications 36.2 (2009).

[20] C. Dietrich, and P. Pawlak, "Crowd-sourcing - crisis response in the digital age", European Union Institute for Security Studies, Issue Alert 39. Nov 2013.

[21] Raghu K. Ganti, Fan Ye, and Hui Lei, "Mobile Crowdsensing: Current State and Future Challenges", IEEE Communications Magazine, 2011.

[22] M. Bishr and K. Janowicz. "Can we trust information?-the case of volunteered geographic information." In: Towards Digital Earth Search Discover and Share Geospatial Data Workshop at Future Internet Symposium. Vol. 640. 2010.

[23] Secure! Project, Tecnic Report "Tassonomia eventi" (Events Taxonomy), 2013, http://secure.eng.it/ [last accessed 28th Oct. 2015].

[24] "Feyenoord Fans damage the Barcaccia Fountain", http://www.ansa.it/english/news/2015/02/19/feyenoord-fans-damage-barcaccia-fountain-update_ac563070-1be5-4cb7-8473-c46351759703.html (19th Feb. 2015) [last accessed 28th Oct. 2015]

[25] "Shocking pictures show Feyenoord fans clash with riot police during long-running battle through Rome", http://www.mirror.co.uk/news/world-news/shocking-pictures-show-feyenoord-fans-5193175 (19th Feb. 2015) [last accessed 28th Oct. 2015]

[26] A. Bondavalli, A. Ceccarelli, L. Falai, M. Vadursi, "A New Approach and a Related Tool for Dependability Measurements on Distributed Systems", IEEE Transactions on Instrumentation and Measurement 59(4): 820-831 (2010)

[27] Thomas, et al. "SOA with REST: Principles, Patterns & Constraints for Building Enterprise Solutions with REST". Prentice Hall Press, 2012.

[28] Ansoff, H. Igor. "Strategic issue management." Strategic Management Journal 1.2 (1980): 131-148.

[29] Bondavalli, Andrea, et al. "Experimental assessment of low-cost GPS-based localization in railway worksite-like scenarios." Measurement 46.1, (2013): 456-466.