



# Detecting prominent microblog users over crisis events phases

Imen Bizid, Nibal Nayef, Patrice Boursier, Antoine Doucet

## ► To cite this version:

Imen Bizid, Nibal Nayef, Patrice Boursier, Antoine Doucet. Detecting prominent microblog users over crisis events phases. *Information Systems*, 2018, 78, pp.173 - 188. 10.1016/j.is.2017.12.004 . hal-03025566

**HAL Id: hal-03025566**

**<https://hal.science/hal-03025566>**

Submitted on 15 Dec 2020

**HAL** is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

# Detecting Prominent Microblog Users over Crisis Events Phases

Imen Bizid<sup>a,1</sup>, Nibal Nayef<sup>a</sup>, Patrice Boursier<sup>a</sup>, Antoine Doucet<sup>a</sup>

<sup>a</sup>*L3i, University of La Rochelle*

---

## Abstract

During crisis events such as disasters, the need for real-time information retrieval (IR) from microblogs becomes essential. However, the huge amount and the variety of the shared information in real time during such events over-complicates this task. Unlike existing IR approaches based on content analysis, we propose to tackle this problem by using user-centric IR approaches with identifying and tracking prominent microblog users who are susceptible to share relevant and exclusive information at an early stage of each analyzed event phase. This approach ensures real-time access to the valuable microblogs information required by the emergency teams. In this approach, we propose a phase-aware probabilistic model for predicting and ranking prominent microblog users over time according to their behavior using Mixture of Gaussians Hidden Markov Models (MoG-HMM). The model utilizes a new user representation which takes into account both the user and the event specificities over time. This user representation comprises the following new aspects (1) Modeling microblog users behavior evolution by considering the different event phases (2) Characterizing users activity over time through a temporal sequence representation (3) Time-series-based selection of the most discriminative features (4) prominent users prediction using probabilistic phase-aware models learned *a priori*. We have conducted experiments during flooding events: we trained our identification models using a dataset relative to the “Alpes-Maritimes floods” and we tested its identification performance using a new dataset relative to another flooding disaster “Herault floods”. The achieved results show that our model significantly outperforms phase-unaware models and identifies most of the prominent users at an early stage of each event phase.

**Keywords:** Information Retrieval from Microblogs, Prominent Users Prediction, User Behavior Modeling, Phase-Temporal Representation

---

---

*Email addresses:* `imen.bizid@univ-lr.fr` (Imen Bizid), `nibal.nayef@univ-lr.fr` (Nibal Nayef), `patrice.boursier@univ-lr.fr` (Patrice Boursier), `antoine.doucet@univ-lr.fr` (Antoine Doucet)

## 1. Introduction

The effectiveness and ease-of-use of supported microblogging platforms – such as Twitter – have revolutionized the communication habits in our society. Any user can quickly and conveniently post and get information on the latest news. During crisis events, the amount of communicated information in such platforms increases significantly. This makes information retrieval more challenging. Most of the shared tweets during these events are non-valuable, redundant, outdated or incredible. Moreover, this shared data is generally expressed in several languages and various formats (i.e. texts, images, links and videos). Thus, content-based retrieval approaches are not well suited for this task as they are time consuming.

This information retrieval problem has been addressed in the literature by associating the quality of tweets with the prominence of their authors in a specific topic or event [1, 2]. In the context of this article, we define *prominent users* as microblog users who are susceptible to share relevant and exclusive information during crisis events regardless of their popularity and their domain of expertise in the platform. To the best of our knowledge such users category has never been targeted in the literature. However, there have been several works targeting other categories of important authors known as domain experts or topical authorities.

The detection of these categories has gained a wide interest in the literature. However, the detection techniques proposed for these categories are not suitable to identify prominent users targeted in this paper. Prominent users in the context of crisis events cannot be systematically considered as domain experts or topical authorities. Most of prominent users refer to *ordinary* users who may provide their testimony based on what they experience in the region of a crisis.

Targeted key users in the literature are generally identified using ranking techniques based on either a graph-based user modeling approach or a vector-based user modeling approach. Graph-based approaches are sensitive to popular microblog users who have a large number of connections, such as celebrities and news outlet channels. Vector-based user modeling approaches have thus been proposed to deal with this problem. Such approaches describe users by a vector of features reflecting the overall tweeting activity on each user based on textual, microblogging and social network structure features. However, such vector-based user modeling approach does neither realistically nor accurately represent the evolution of user behavior over time. This yields weaker performance of detection and ranking algorithms which learn to distinguish behavioral differences among different users.

Characterizing users without considering the temporal distribution of their activities over event phases would not reveal the real user behavior. This is due to the following: (1) Quantitative characterization of users: Practically, such characterization would promote users sharing much information about an event even if this information is irrelevant or outdated. (2) Uniform user characterization over the event duration (from the beginning of an event until its end): Realistically, the behavior of users may differ according to the evolution of the

event. (3) Overall user prominence evaluation over the event duration: Such strategy would fail to discover true prominent users who were active in only one – however important – event phase, because their activity statistics are lower compared to other users who were active in prior phases.

Moreover, the problem of key users prediction has never been tackled in the literature. Most of the proposed identification models have been modeled and experimented to classify or/and rank such users by the end of an event and not over time. The challenge behind our proposed model is to predict such prominent users at an early stage of each event phase in order to track these users and get access to the relevant information they are sharing.

This work alleviates these shortcomings by proposing a new user modeling and prediction approach considering: -Event evolution over time, and -User behavioral change over event phases and over time of each phase. Crisis events – specially natural disasters – are usually described in terms of “*phases*” having their specific goals, characteristics and experts. Each phase influences users’ behavior differently according to their interest and involvement in that phase. This proposed user modeling approach is implemented within prominent users prediction and ranking model learned using data from prior crisis events. This model overcomes the problem of time-consuming information retrieval techniques by considering features which can be computed in real time and learning *a priori* the identification models adapted to each category of crisis events. Through our experiments, we have trained a model adapted for prominent microblog users identification during flooding events.

The rest of this paper describes the integration of these ideas for prominent users identification during crisis events. In Section 3, we describe our phase-aware user behavior modeling approach. We list the different extracted user features used by the feature selection process to characterize user behavior at each event phase in Section 4. Our temporal phase-aware probabilistic model for the classification and ranking of microblog user’s behavior is detailed in Section 5. The evaluation set-up is described in Section 6. Experimental evaluation is presented in Section 7. Finally, we present the discussion and conclusions along with directions for future work in Sections 8 and 9.

## 2. Related Work

To the best of our knowledge, the issue of prominent users identification has not been explored in depth in the context of crisis events. However, there have been several works which proposed models to identify other categories of key users such as microblog influential users, topical authorities and domain experts in a more general context [3, 2]. Such models have mainly focused on proposing a user modeling approach that is able to highlight the differences between key and non-key users on specific topics. Through that user characterization, machine learning or ranking algorithms are generally explored to learn or identify similar users behaviors. These state-of-the-art user modeling approaches fall into two categories -A graph-based user modeling approach describing user interaction in

the network [4, 5, 6] or a vector-based user modeling based on a list of descriptive user behavior features [1, 2, 7].

The graph-based-user modeling approach represents users behavior by a graph composed of nodes and edges denoting respectively users and any nature of relation that may link them. Such representation is generally adopted for both influencers [3, 8] and domain experts detection [5]. The IP-influence model [3] – which identifies influencers – defined edges as pairwise influence and passivity according to the retweeting activity of users. TwitterRank [5] identified domain experts using the PageRank algorithm ranking users according to their position on Twitter graph constructed according to the tweeting activity of users. Such user representation has been criticized as it makes the identification process sensitive to popular users who are not necessarily prominent [1].

The vector-based user characterization has been proposed as a new alternative to address this sensitivity. This user characterization approach was firstly introduced by Pal and Counts [1] in the context of domain experts identification. They represented by a single vector composed of 15 features describing the user tweeting activity in order to cluster and rank each user according to his/her expertise. Similarly, Xianlei et al. [2] employed this same user characterization by referring to linguistic, user activity and profile features in order to classify them using a machine learning algorithm. Ghosh et al. [7] represented users by a topic vector composed of different weighted terms extracted from the Twitter lists. Through this representation, users are ranked by computing the topical similarity scores between the different vectors.

While most of those vector-based models which identify domain experts have been applied in topics referring to events such as “The world cup”, these models remain unsuitable for the context of crisis events. Firstly, prominent users – in crisis events – are not necessarily domain experts, they may be ordinary users who are implicated involuntarily in a particular disaster which has occurred in their region. Thus, such users cannot be detected *a priori* using Twitter lists [7]. Second, characterizing users uniformly and quantitatively during the whole event using such representation would not reflect the real user behavior [9]. The user behavior and interest change over time according to the evolution of the event. Finally, the user’s prominence may not be associated with the whole event, users may be prominent only in one particular phase.

The present contribution addresses these limitations. In a previous work [9], we have presented a new user characterization approach consisting of representing users by a sequence of feature vectors extracted over time independently of the event characteristics. In this paper, we propose a complete user characterization considering both the user behavior and the event evolution over time in order to predict prominent users in real time. We also tackle the problem of prominent users identification in terms of prediction, not classification. In other words, we focus on learning a model which is able to predict prominent users over time and not by the end of the event. To the best of our knowledge, such problem has never been tackled in the literature.

### 3. User Behavior Representation in the Context of Crisis Events

In order to consistently model microblog users with their realistic behavior during events, we propose a user behavior characterization approach that alleviates the shortcomings stated in Sections 1 and 2. An analyzed event is divided into different phases according to its nature/context. This section firstly describes how we have considered event phases while representing the user behavior. Then, in the second subsection, we detail our proposed per-phase temporal characterization approach for modeling the behavior change of users over time.

#### 3.1. Crisis Events Evolution and its Impact on the Behavior and Prominence of Microblog Users

Like users have their own specificities, events or even topics have their own criteria that have to be considered while modeling user behavior. In the context of crisis events, the event time-line is generally represented as a sequence of “*phases*” referring to the evolution states of the event over time. Crisis events characteristics and level of importance change according to each event phase. The user interest and behavior regarding a particular event differ from one phase to another. Modeling microblog users uniformly during the whole event would give a misleading image of the real behavior of the user over time. This subsection describes how our proposed user behavior characterization approach takes into account the impact of the event evolution on both the user behavior and prominence.

This approach characterizes any crisis event by a sequence of  $d$  different successive phases  $E = (P_1, P_2, \dots, P_d)$ . These phases are defined *a priori* by the domain experts according to the crisis event context. Similarly, each microblog user is represented by a sequence of  $d$  representations reflecting his/her behavior at each phase.

$$R(u) = (R_{P_1}, R_{P_2}, \dots, R_{P_d}) \quad (1)$$

In this paper, we consider the standard crisis events phases categorization consisting of characterizing crisis events evolution – specially natural disasters – into three main phases [10]. These phases boundaries can be automatically defined in real time by referring to the announcement of expert official organizations (e.g. meteorological organizations in the case of flooding). In the following, we detail the specificities of these three different phases:

**Phase 1** Preparedness: is the phase announcing a possible risk that may arise on the next hours or minutes.

**Phase 2** Response: is the most delicate phase during the disaster as it covers the period of the disaster occurrence.

**Phase 3** Recovery: refers to the actions made following a disaster in order to inventory the damages and regain the usual level of functioning before the disaster.

In these following subsections, we detail the specificities of these phases and their effect on microblog users behavior over time.

### 3.1.1. Event Phases Impact on Users' Prominence

As each crisis event phase has its particularities, we propose to associate microblog users' prominence to each phase rather than to the whole event. Prominent users differ according to the disaster phase. During the first phase while the risk is not yet confirmed, expert meteorologists are involved to analyze and communicate any news. Once the risk is confirmed and the red alert is raised, the response phase has to be managed. Emergency first responders such as police officers, fire-fighters, paramedics and emergency medical technicians intervene in order to address the immediate threats. When the situation becomes under control, emergency first responders retire in order to give way to experts who are in charge of recovering the disaster consequences. Similarly, in microblogs, not all users are interested in a disaster from its beginning to its end. For example, prominent users in the first phase may not necessarily remain prominent in the second or the third one.

We thus propose to characterize user's behavior per phase in order to ensure a fair evaluation among microblog users at each phase and a relevant per-phase prominent users detection. The high or low activity of a particular user in a prior phase is not considered in the next phases. All the features characterizing the activity of users are reset to zero at the beginning of each new phase. Only users' activity registered at the analyzed phase is considered. Then, detected prominent users in a particular phase would be tracked only during that phase unless they prove their prominence in other phases. In this way, we avoid to track users who were prominent just in a particular phase during the whole event.

To summarize, we model the impact of event phases on users' prominence by associating the user's prominence with his/her activeness at each event phase rather than the whole event. Each microblog user is characterized by a per-phase representation  $\{R_{P_1}\}, \{R_{P_2}\}, \dots, \{R_{P_d}\}$ . Based on this representation, each user is classified in one of these corresponding classes  $\{C_1^{P_1}, C_2^{P_1}\}, \{C_1^{P_2}, C_2^{P_2}\}, \dots, \{C_1^{P_d}, C_2^{P_d}\}$ .  $C_1^{P_j}$  and  $C_2^{P_j}$  refer respectively to prominent and non-prominent microblog users during the phase  $j$ . The classification and prediction model appropriated for each phase will be described in Section 5.

### 3.1.2. Event Phases Impact on Users' Behavior

In the context of crisis events, user behavior differs in the first and third phase from the second phase. Prominent microblog users in the second phase are generally in panic and would mainly concentrate on expressing what they are seeing and experiencing regarding the event. However, in other phases, they will act somehow like ordinary days. Representing users differently at each particular phase highlights users behavior specificities per phase and makes true prominent users more discoverable.

To cover these users' behaviors changes according to each event phase, we model each user differently at each phase by using different features. We select

the best  $k$  representative features reflecting users' behavior at that phase  $j$ .

$$F^{P_j} = (F_1^{P_j}, F_2^{P_j}, \dots, F_k^{P_j}) \quad (2)$$

The sequence of feature vectors characterizing each user  $R_{P_j}$  is composed only by the selected features  $F^{P_j}$  characterizing users behavior during that phase  $P_j$ . These features are selected from a large set of raw and engineered features characterizing user activity in microblogs using a multi-variate feature selection algorithm Corona [11] (See Section 4). Using this strategy, we represent users behavior differently according to the analyzed phase by using appropriate features selected *a priori*. This selection is conducted by learning the behavior of users during the different phases of similar events. The extraction and selection processes of these features will be described in depth in the next section.

### 3.2. User Behavior Modeling as Temporal Sequences

In order to track the real user behavior over time, we represent each user activity during each phase by a temporal sequence of feature vectors. These feature vectors are computed based on the selected features reflecting the user behavior at that specific phase. The time-line of each event phase is divided into equispaced intervals at  $m$  time-stamps  $t_1, t_2, t_3, \dots, t_m$  from the beginning of the phase  $P_j$  until its end. At each time-stamp  $t_i$ , we represent each microblog user  $u$  by a feature vector  $V_{P_j}^{t_i}$  characterizing his/her behavior from the time-stamp  $t_{i-1}$  to  $t_i$ .

$$V_{P_j}^{t_i}(u) = (F_1^{P_j, (t_i)}, F_2^{P_j, (t_i)}, \dots, F_k^{P_j, (t_i)}) \quad (3)$$

Then, the resulted vector is added in the sequel of the temporal sequence  $R_{P_j}^{t_{i-1}}$  composed of the prior calculated vectors from the beginning of that phase.

$$R_{P_j}^{t_i}(u) = (V_{P_j}^{t_1}(u), V_{P_j}^{t_2}(u), \dots, V_{P_j}^{t_{i-1}}(u), V_{P_j}^{t_i}(u)) \quad (4)$$

Segmenting the sequence of user activity at each phase into time-series feature vectors offers a rich and personalized user representation. Users sharing the same quantity of information would not be represented similarly. By highlighting the temporal activity of each user, it becomes easier to differentiate between users interacting and sharing exclusive news at the beginning of each phase and users sharing the same information at its end. Our user modeling approach offers a full vision of users behavior by taking into account the evolution of both users and events over time. It provides a detailed user representation closer to his/her real behavior in microblogs. Such representation eases the identification of users behavior regularities, similarities and dissimilarities at each phase.

## 4. Extraction and Selection of Microblog Users' Features

In order to efficiently model the user behavior particularities at each phase, we evaluate the effectiveness of a large set of features  $X$  composed of state-of-the-art and our new proposed raw and engineered features. Based on this set,

we select a subset of features  $X_s$  that could best reflect the real user’s behavior according to each phase. Both the features extraction and features selection steps are processed off-line using prior disaster datasets. Trough these processes, the best representative features of users behavior per phase can be selected. The following subsections describe these two *off-line* processes in detail.

#### 4.1. Features Extraction

At each phase, we extract and compute the following set of raw and engineered features for each microblog user  $u$  and each time-stamp  $t_i$  during a particular event phase  $P_j$ . These features are computed by considering our on- and off-topic user interactions categorization that was introduced in our prior work [12]. *On-topic* refers to any user activity containing a subset of a list of keywords and hashtags which are defined to describe the event under consideration and not including any keywords reflecting non-serious or non-valuable contents (i.e. lol, mdr, rent...). *Off-topic* refers to any activity that was not recorded as an on-topic one.

##### 4.1.1. Raw Features

We define *raw features* as statistics collected to quantitatively measure the data characterizing different natures of the user interactions in a microblog. The feature values are computed by analyzing raw users activities.

We compute these features by considering mainly the three natures of user interactions extractable from the user time-line: *original tweets* ( $T$ ) are tweets originally expressed by the user, *retweets* ( $RT$ ) recognizable with “RT@” prefix are tweets already shared by another microblog user and forwarded later by the user, and *mentions* ( $M$ ) are tweets addressed to one or several users mentioned using the “@” symbol. Table 1 summarizes the extracted features from both our proposed and state-of-the-art metrics.

As it has been shown in our prior work [12], characterizing microblog users using only simple raw metrics, hides many interesting characteristics of users’ behavior. Such features have many correlations between them that have to be exploited. For example, features extracted from tweets having the same nature (e.g.  $RT_1$  and  $RT_2$ ;  $R1_{on}$  and  $R1_{off}$ ) can be combined in order to construct more representative engineered features.

##### 4.1.2. Engineered Features

*Engineered features* are defined as higher level hand-designed features which are usually constructed by combining and enhancing simple raw features and studying the relationships among them. The goal is typically to get additional relevant and more discriminative features characterizing user behavior in order to increase the predictive power of learning algorithms.

By exploring the possible combinations between the raw extracted features, we propose a set of engineered features (i.e. adjusted and not adjusted) detailed in Table 2. The rationale behind the adjustment of on-topic features with the off-topic ones is to penalize users – such as news outlets – who toggle among

Table 1: List of raw features describing on- and off-topic user’s behavior during each time-stamp of each event phase. **(new)** denotes our proposed features. Note that these features are computed twice for on- and off-topic user activity.

Original tweets	
$T1^{P_j, (t_i)}$	: #original tweets [1, 2]
$T2^{P_j, (t_i)}$	: #links shared [13]
$T3^{P_j, (t_i)}$	: #keyword and hashtags [1]
$T4^{P_j, (t_i)}$	: #favorites of original tweets (new)
$T5^{P_j, (t_i)}$	: #tweets geo-located in the event area (new)
Retweets	
$R1^{P_j, (t_i)}$	: #retweets of other’s tweets [14, 2]
$R2^{P_j, (t_i)}$	: #unique users retweeted by the evaluated user (new)
$R3^{P_j, (t_i)}$	: #retweets of the tweets of the evaluated user (new)
$R4^{P_j, (t_i)}$	: #unique users who retweeted the tweets of the evaluated user [14]
Mentions	
$M1^{P_j, (t_i)}$	: #mentions of other users by the evaluated user [15, 1]
$M2^{P_j, (t_i)}$	: #unique users mentioned by the evaluated user [15, 1]
$M3^{P_j, (t_i)}$	: #mentions by others of the evaluated user [15, 1]
$M4^{P_j, (t_i)}$	: #unique users mentioning the evaluated user [15, 1]
Graph	
$G1^{P_j, (t_i)}$	: #active followers [4]
$G2^{P_j, (t_i)}$	: #active followees [4]

several topics, and who may share outdated information. In the following, we describe these engineered features:

*EF1* and *EF2* refer respectively to the influence of user’s original tweets in the network, and the user’s productivity and involvement regarding the event.

*EF3* and *EF4* measure respectively the impact of the event-related tweets on the user retweeting activity and the other users reaction regarding the user’s own tweets.

*EF4* and *EF5* analyze respectively the user received mentions and the sent mentions addressed to other users.

*EF6* refers to the centrality degree of each user regarding the event.

Once both raw and engineered features are computed at each time-stamp during each event phase, we represent each user  $u$  by an initial feature vector  $\tilde{V}_{P_j}$  characterizing his/her activity at each time-stamp  $t_i$  during each phase  $P_j$ . Each feature vector is composed of the complete features set  $X$  (i.e. 30 raw features and the 14 engineered features).

$$\tilde{V}_{P_j}^{t_i}(u) = (T1_{on}^{P_j, (t_i)}, T1_{off}^{P_j, (t_i)}, ..., EF6_{NAdj}^{P_j, (t_i)}) \quad (5)$$

By assembling the different feature vectors computed at each time-stamp  $t_i$  during  $P_j$ , we associate each user with an initial temporal sequence of vectors  $\tilde{R}_{P_j}$  describing the user behavior at that phase.

$$\tilde{R}_{P_j}^{t_i}(u) = (\tilde{V}_{P_j}^{t_1}(u), \tilde{V}_{P_j}^{t_2}(u), \tilde{V}_{P_j}^{t_3}(u), ..., \tilde{V}_{P_j}^{t_i}(u)) \quad (6)$$

Table 2: List of our proposed engineered features derived from the raw features. Not-adjusted features describe on-topic user interactions, and the adjusted ones refer to both on- and off-topic interactions.

	Not-Adjusted Features	Adjusted Features
$EF1^{P_{j_i}(t_i)}$	$T2_{on}^{(t_i)} * \log(T2_{on}^{(t_i)} + R3_{on}^{(t_i)} + 1)$	$\frac{T2_{on}^{(t_i)} * \log(T2_{on}^{(t_i)} + R3_{on}^{(t_i)} + 1)}{T2_{off}^{(t_i)} * \log(T2_{off}^{(t_i)} + R3_{off}^{(t_i)} + 1) + 1}$
$EF2^{P_{j_i}(t_i)}$	$T1_{on}^{(t_i)} + T2_{on}^{(t_i)}$	$\frac{T1_{on}^{(t_i)} + T2_{on}^{(t_i)}}{T1_{off}^{(t_i)} + T2_{off}^{(t_i)} + 1}$
$EF3^{P_{j_i}(t_i)}$	$R1_{on}^{(t_i)} * \log(R2_{on}^{(t_i)} + 1)$	$R1_{on}^{(t_i)} * \log(R2_{on}^{(t_i)} + 1) - R1_{off}^{(t_i)} * \log(R2_{off}^{(t_i)} + 1)$
$EF4^{P_{j_i}(t_i)}$	$R3_{on}^{(t_i)} * \log(R4_{on}^{(t_i)} + 1)$	$R3_{on}^{(t_i)} * \log(R4_{on}^{(t_i)} + 1) - R3_{off}^{(t_i)} * \log(R4_{off}^{(t_i)} + 1)$
$EF5^{P_{j_i}(t_i)}$	$M3_{on}^{(t_i)} * \log(M4_{on}^{(t_i)} + 1)$	$M3_{on}^{(t_i)} * \log(M4_{on}^{(t_i)} + 1) - M3_{off}^{(t_i)} * \log(M4_{off}^{(t_i)} + 1)$
$EF6^{P_{j_i}(t_i)}$	$M1_{on}^{(t_i)} * \log(M2_{on}^{(t_i)} + 1)$	$M1_{on}^{(t_i)} * \log(M2_{on}^{(t_i)} + 1) - M1_{off}^{(t_i)} * \log(M2_{off}^{(t_i)} + 1)$
$EF7^{P_{j_i}(t_i)}$	$\log(G1_{on}^{(t_i)} + 1) - \log(G2_{on}^{(t_i)} + 1)$	$\log(\frac{G1_{on}^{(t_i)} + 1}{G1_{off}^{(t_i)} + 2}) - \log(\frac{G2_{on}^{(t_i)} + 1}{G2_{off}^{(t_i)} + 2})$

#### 4.2. Features Selection

Once all features characterizing microblog users behavior are extracted, we select the best representative features set  $X_s^{P_j}$  for each phase  $P_j$ . Through this process, we can reduce the dimensionality of each feature vector  $\tilde{V}_{P_j}^{t_i}(u)$  and obtain an optimal user characterization  $R_{P_j}(u) = \tilde{R}_{P_j}^*(u)$  at each phase by eliminating redundant and irrelevant features.

As we modeled users by a Temporal Sequence of Feature Vectors (TSFV)  $\tilde{R}_{P_j}(u)$  to ensure the characterization of each user behavior over time, we use Corona [11], a supervised feature subset selection technique for TSFV. Using Corona, we select the top relevant features at each event phase. This process occurs off-line during training the model on data from previous events. Corona was selected for this task as it maintains the correlation between the different feature vectors  $\tilde{V}_{P_j}^{t_i}(u)$  computed at different time-stamps  $t_i$  corresponding to the same event phase  $P_j$ .

Corona computes at first the correlation coefficient matrix of each TSFV using Equation 7. This correlation matrix represents the relationship between each two feature vectors included in the TSFV at each phase according to the used training data. Assume that  $a$  and  $b$  refer respectively to the feature vector  $\tilde{V}_{P_j}^{t_i}(u)$  characterizing the user behavior at time-stamp  $t_i$  and the feature vector  $\tilde{V}_{P_j}^{t_{i+1}}(u)$  of the same user at time-stamp  $t_{i+1}$ . The dimension of those vectors is  $l = 44$ . This number corresponds to the initial number of features.

$$corr(a, b) = \frac{\sum_{k=1}^l (a_k - (\bar{a}))(b_k - (\bar{b}))}{(l-1)\sigma_a\sigma_b} \quad (7)$$

Where  $(\bar{a})$  and  $(\bar{b})$  are respectively the averages of the feature vectors computed at time-stamp  $t_i$  and time-stamp  $t_{i+1}$ ;  $\sigma_a$  and  $\sigma_b$  are the standard deviations of  $a$  and  $b$ .

Each resulted correlation coefficient matrix is then vectorized. Using these vectors, we subsequently train a SVM model to obtain the weights relative to each feature included in the training stage. We then aggregate the resulted weights in order to have one weight value relative to each feature. Based on these aggregated values, we select the worst feature using a greedy approach consisting of identifying the feature whose maximum weight is the minimum compared to all the other features weight. Subsequently, we remove the selected worst feature.

This whole process is then repeated until the  $k$  best features that reflect users behavior at each phase  $P_j$  are obtained. The selected features are then used to represent each microblog user at that phase.

### 5. Learning to Predict and Rank Phase-aware Users' Prominence

In this Section, we describe our phase-aware probabilistic model for prominent microblog users prediction during crisis events. Figure 1 describes how this model is learned *off-line* and how it works *on-line* during crisis events. During

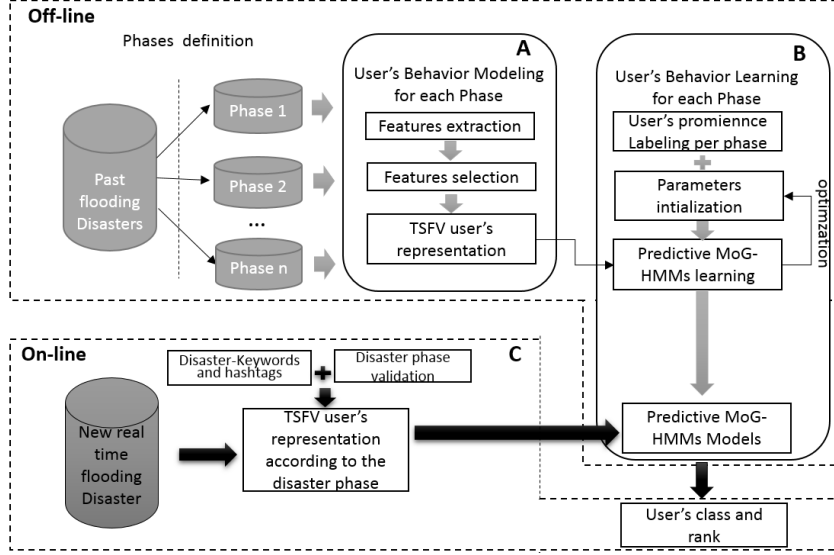


Figure 1: Block diagram of real-time prominent microblog users identification model during crisis events.

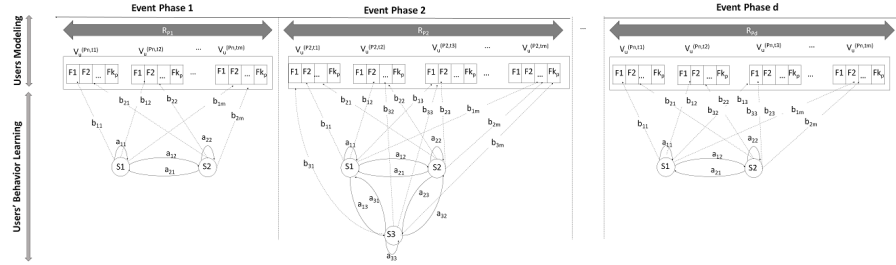


Figure 2: The different ergodic MoG-HMM models trained for prominent users detection at each event phase.

the off-line stage, the model learns (selects) which features to compute for each event phase as shown in *Box A*. Then, the phase-aware model is built by learning the different behaviors of prominent and non-prominent microblog users during each phase of similar previous disasters. Once the model has learned to differentiate between prominent and non-prominent users behavior over time, it can be applied in real time during similar disasters. In the on-line stage, each microblog user behavior has to be represented by the TSFV user's representation corresponding to each phase. Hence, the features are automatically extracted according to the analyzed disaster phase. In the following, we detail further the learning step described in the *Box B* of Figure 1 and the prediction model process in real-time as represented in *Box C*.

In order to evaluate the prominence of each new microblog user interacting during the analyzed phase, we aim to learn *a priori* the phase-aware prediction models for each crisis event category and test the resulted model in similar events. As described in Figure 1, these learned models have to classify over time each microblog user behavior characterized by the TSFV  $R_{P_j}^{t_i}$  in either class  $C_1^{P_j}$  or  $C_2^{P_j}$  referring respectively to whether the user is classified as prominent or not at the analyzed event phase  $P_j$ . The TSFV representation step is processed once the keywords and hashtags describing the analyzed event are defined and the current event phase is identified.

Learning such binary classification models is especially critical in crisis events context, where training data from the positive class  $C_1^{P_j}$  are inherently rare, and are costly to analyze. In fact, although there is a huge amount of disaster-related-information shared in microblogs during the different disasters phases, the number of real prominent users who provide valuable information is small. Thus, collecting samples describing prominent microblog users' behavior during crisis events for the model learning remains difficult.

Taking into account the stated training data limitations, we address prominent user's behavior identification problem using generative classification methods. Indeed, both theoretical and empirical studies pointed out that while discriminative models achieve lower asymptotic classification error, generative methods tend to be superior when training data are limited [16].

The generative approach of MoG-HMMs for classification and ranking can be suited to our problem. Thus, we train separate ergodic MoG-HMM models for each class at each time-stamp during each event phase as described in Figure 2. Each MoG-HMM model  $H_{P_j}$  is represented by 4 parameters  $H_{P_j} = \{S_{P_j}, \pi_{P_j}, A_{P_j}, B_{P_j}\}$  described in Equation 8. We optimize these parameters using the Baum-Welch algorithm [17] based on the EM algorithm selecting the maximum probability that fits better the observed user behavior in the training data from the beginning of the analyzed event phase at each timestamp  $t_i$ .

$$H_{P_j}(t_i) = \underset{H_{P_j}(t_i)}{\operatorname{argmax}} P(R_{P_j}^{training}(t_i) | H_{P_j}(t_i)) \quad (8)$$

Where:

$S_{P_j} = S_1, S_2, S_3, \dots, S_f$  refers to the set of  $f$  hidden states describing the levels of users activities at each timestamp of the phase  $P_j$ . The state of a user at time  $t$  can be expressed by  $(X_t \in S)_{t_i} 1 \leq i \leq m$  refers to the user's behavior state at a particular time-stamp  $t_i$ .

$\pi_{P_j}$  denotes the initial probability of the different states.  $A_{P_j} = a_{ij}$  is the state transition probability matrix to change from state  $S_e$  to  $S_l$  where  $a_{ij} = P(X_{t+1} = S_j | X_t = S_i)_{1 \leq i, j \leq f}$ .

$B$  refers to the continuous output probability matrix where the probability  $B_{P_j} = b_i(V^t)$  represents the probability of observing a feature vector  $V^t$  from a state  $S_i$ , where  $b_i(t) = P(V^t | X_t) = (S_i)_{1 \leq i \leq k}$ .

In order to transform the sequence of feature vectors into a sequence of dis-

crete states, we generate a continuous observation probability density function (PDF) matrix  $B$  according to the training data using equation 9.

$$b_e(V^t) = \sum_{k=1}^M c_{ek} \mathcal{N}[V^t, \mu_{ek}, W_{ek}] \quad (9)$$

where  $c_{ek}$  is the mixture weight,  $\mathcal{N}$  is the normal density,  $\mu_{ek}$  is the mean vector and  $W_{ek}$  is the covariance matrix for the  $k^{th}$  mixture component in state  $S_e$ .

Once the HMM-MoG models  $H_{P_j}^{C_1}(t_i)$  and  $H_{P_j}^{C_2}(t_i)$  parameters corresponding to each phase are optimized using the training dataset, each microblog user can be classified into one of the analyzed event phase classes by computing the following probabilities  $P(R_{P_j}^{t_i}(u)|H_{P_j}^{C_1}(t_i))$  and  $P(R_{P_j}^{t_i}(u)|H_{P_j}^{C_2}(t_i))$ . These probabilities are computed given the two learned models and the user behavior from the beginning of that phase until  $t_i$  using the forward-backward algorithm [18]. If the returned probability by the model  $H_{P_j}^{C_1}(t_i)$  is greater than  $P(R_{P_j}^{t_i}(u)|H_{P_j}^{C_2}(t_i))$ , then this user is classified as prominent and has to be tracked until the end of the phase  $P_j$ .

In order to rank the selected prominent users, we sort the likelihood  $P(R_{P_j}^{t_i}(u)|H_{P_j}^{C_2}(t_i))$  of the different microblog users sequences regarding the model  $H_{P_j}^{C_2}(t_i)$ . The smaller this probability, the bigger the prominence of that user. Our rationale behind ranking users by referring to their likelihood regarding  $MoG - HMM_{C_2}$  rather than  $MoG - HMM_{C_1}$  consists of targeting the model which tends to be the most precise.  $MoG - HMM_{C_2}$  is generally learned using a consistent number of samples covering most of the non-prominent users behaviors, thus, its returned likelihood regarding a new evaluated microblog user behavior tends to be more precise than the one returned by  $MoG - HMM_{C_1}$  trained using limited data.

## 6. Performance Evaluation

### 6.1. Datasets Description

There are various publicly available datasets for information retrieval community for dealing with information retrieval challenges during crisis events. The most popular ones are those published in CrisisLex [19]. However, to the best of our knowledge, such datasets are not adapted to test user-centered information retrieval approaches. To test these approaches, in particular the one proposed in this paper, both on- and off-topic user activities during particular disasters are needed.

Moreover, motivated by the access to domain experts able to label new data, we have implemented our own data collection system adapted to this purpose. This system named MASIR is based on a multi-agent architecture enabling an extensive tweets extraction process. The specificity of MASIR is its ability to boost the number of tracked microblog users and extracted tweets using multiple hosts and Twitter developers accounts to cope with the extraction limits of Twitter APIs. More details about the architecture of MASIR can be

found in our previous work [20]. To perform data collection for the following experiments, MASIR extraction module was executed using 5 hosts and 30 developer accounts. This module proceeds in two steps:

The first step consists of extracting any shared tweet containing at least one hashtag or keyword describing the analyzed disaster. Table 3 lists the different defined keywords for the collection of these tweets during the the Herault and Alpes-Maritimes flooding events. These keywords were defined by selecting the main frequent key terms used by Twitter users to identify these analyzed events. The key terms identification process could also be automated using existing keywords extraction techniques [21, 22]. Once these tweets are extracted, MASIR identifies each microblog user who has shared them.

The second step consists of crawling the profiles of these identified users. Any tweet shared by the identified users has to be extracted even if it is not related to the disaster event. The idea behind storing all users’ shared information is to have a complete view of their behavior from the beginning of the analyzed disaster until its end.

Table 3: Selected keywords for the extraction of tweets describing the Herault and Alpes-Maritimes floodings.

	AlpesMarDB	HeraultDB
Keywords	AlpesMaritimes, Orage, Alpes-Maritimes, Intempries, Orages, Antibes, Nice, Nice06, Cannes, Inondations, ...	Herault, Hrault, Crue, Crues, Orage, Orages, Intempries, Flooding, Montpellier, Alert, RedAlert, ...

Using MASIR, we have collected two disaster datasets relative to two different flooding events: “*Herault floods*” and “*Alpes-Maritimes floods*”. These two events have occurred respectively in the south-east and south of France in September 2014 and October 2015. To evaluate the extraction performance of MASIR, we compared its extracted tweets for a sample of 20 crawled users profiles from each dataset to those displayed in the web interface of the users’ profiles. We found that between 80% to 100% of these users’ tweets were fully and correctly extracted. To recover the missing tweets, a recovering process was launched by MASIR to extract any detected missing tweet.

As the targeted events fall in the same category of natural disasters, we used the first dataset to train our model and the second one to test the learned model performance for prominent users identification during similar flooding cases.

During the training and testing of our models, a disaster is considered as a sequence of three phases: *P1*, *P2* and *P3* referring to the standard disaster phases *Preparedness*, *Response* and *Recovery* respectively. The phases boundaries were set by referring to the official meteorological organizations of the regions threatened and affected by the disaster. Such organizations determine and announce the alert level and the state of the disaster evolution during natural disasters. Table 4 shows statistics of the collected tweets at each phase in both datasets.

The first dataset “*Alpes-MaritimesDB*” is used to build our user behavior

Table 4: Number of the different natures of tweets recorded in the two datasets **AlpesMarDB** and **HeraultDB** at each phase.

Event Phases	#OnT	#OnRT	#OnM	#OffT	#OffRT	#OffM	
AlpesMarDB	P1	155	91	32	1506	788	434
	P2	6692	4046	300	5840	3547	1064
	P3	22343	13579	1960	51596	28736	9693
HeraultDB	P1	513	329	36	9102	4333	2165
	P2	3357	2480	202	5823	2904	1427
	P3	2229	1260	208	4586	2293	1083

characterization and prediction model. This dataset refers to the floods that have occurred in the Alpes-Maritimes area between the 3<sup>rd</sup> and 7<sup>th</sup> October 2015. 152,402 tweets shared by 21,364 users were collected during this event. The different disaster phases *P1*, *P2* and *P3* have lasted 3.5, 18.5 and 72 hours respectively according to the information provided by the meteorological vigilance center of Provence-Alpes-Cote d’Azur.

The second dataset “*HeraultDB*” is used in order to test the model learned using the first dataset. “*HeraultDB*” refers to the floods that have occurred from 29<sup>th</sup> to 30<sup>th</sup> September 2014 in the Herault area. This dataset consists of 44,330 on- and off-topic tweets shared by 3,338 users during the whole event. The different disaster phases *P1*, *P2* and *P3* of this described event have lasted 15, 17, 15 hours respectively according to the information provided by the meteorological center of Aix-en-Provence.

## 6.2. Ground-truth Description

To create the ground-truth of our two collected datasets, we conducted a subjective user study for manually labeling each user at each phase  $P_j$  as  $C1^{P_j}$  “prominent” or  $C2^{P_j}$  “non-prominent”. As “*Alpes-MaritimesDB*” includes a huge number of microblog users that requires a long time and a great effort to label it. We filter this dataset by retaining only microblog users who have shared at least one event-related tweet during the evaluated phase  $j$  to be manually labeled. The non-retained ones are automatically labeled in  $C2^{P_j}$ . The labeling of the retained users in our two datasets is conducted by three participants having known the two flooding disasters’ areas and having followed these two disasters. These participants were also required to be familiar with the concept of tweets and fluent with the languages (i.e. french, english) used by microblog users interested in the analyzed disasters. Two of these participants were separately asked to manually label all the microblog users according to the relevance and exclusiveness of their shared disaster tweets at each phase. To check the exclusivity of user tweets, we have provided these participants a report listing in a chronological order most of the important disaster news with their time of first announcement. Once, all the users were labeled by the first two participants, the third participant is asked to break the labels’ disagreement between these two participants. The final ground-truth results of the two datasets are described in Tables 5 and 6.

A second study is then conducted to rank the users labeled as prominent. The same participants have been asked to attribute a score on a scale from 4 to 10 to each user labeled as prominent. Each score has to reflect the relevance and freshness of each user tweets shared during the analyzed phase. The average of scores set for each user is then calculated and retained.

Table 5: Results of the subjective user study for the ground-truth construction of the two datasets according to each phase.

Event Phases	#Prominent users	#Non-prominent users
AlpesMarDB	P1	20
	P2	99
	P3	157
HerauldDB	P1	35
	P2	87
	P3	67

Table 6: Common ( $\cap$ ) and distinct ( $\cup$ ) prominent users in the different phases of each dataset.

Prominent users sets	AlpesMarDB	HerauldDB
$\{C_1^{P_1} \cap C_1^{P_2}\}$	12	21
$\{C_1^{P_2} \cap C_1^{P_3}\}$	31	20
$\{C_1^{P_1} \cup C_1^{P_2} \cup C_1^{P_3}\}$	233	148

In order to evaluate the quality of the constructed ground-truth, we conduct an evaluation study by calculating the inter-annotator agreement (IAA) scores and the Pearson rank correlation coefficient. IAA scores are calculated in order to measure the level of disagreement between the first two annotators for the microblog users classification task. The Pearson rank correlation coefficient scores are measured to evaluate the scores given separately by each annotator to rank the users.

According to this study, the Herauld floodings constructed ground-truth has an IAA of 0.89 and a Pearson score of 0.75 and Alpes-Maritimes flooding one has an IAA score of 0.85 a Pearson score of 0.7. These results show that there has not been much disagreement between the first two annotators. The high registered IAA and Pearson scores show the high level of agreement of the different annotators regarding the same data.

We conclude that judging the prominence of users is a tricky task even for human annotators. The annotators evaluating users regarding the exclusiveness and relevance of their tweets have been in doubt in some cases regarding the prominence of such users. Therefore, the identification of the prominent users would be trickier for our model. Such model has to judge each user only regarding his/her behavior independently of the relevance and exclusiveness of his/her tweets. Using our proposed identification model, we aim to reach the same level of results as those provided by the two first annotators.

### 6.3. Evaluation Metrics

To evaluate our learned models and set its parameters, we use standard evaluation metrics: recall and ranking measures such as Recall@10 and Precision@K where  $K = \# \text{number of ground truth prominent users}$ .

- Recall@10 =  $\frac{\# \text{detected top10 ground-truth prominent users}}{10}$
- Precision@K =  $\frac{\# \text{detected true prominent users ranked in the top K}}{K}$

### 6.4. Experimental Set-up

In order to set a proper value to the time interval  $m$  for user behavior modeling, we tune  $m$  from 15 minutes to 9 hours while extracting time series feature vectors representing each user in our training dataset. This parameter  $m$  does not vary across different phases. For example, after 4 hours from the beginning of a particular phase  $P_j$ , each microblog user would be represented by a sequence of 8 vectors of features if  $m = 30$  minutes and a sequence of 2 feature vectors if  $m = 2$  hours.

Figure 3 shows the prediction results in terms of  $Recall_{C1}$  and  $Precision@k$  at the one-third, half, two-third and the end of each *Alpes-Maritimes* event phase, while tuning the temporal sequences' interval  $m$ . According to the obtained results, we note that representing microblog users behavior into short sequences of vectors erodes the model ranking and classification results. However, representing users using very long sequences is not recommended either (as the case with  $m = 15$  minutes). Recognizing relevant patterns in very long input sequences can turn out to be difficult to analyze [23]. Thus, we set  $m$  to 30 minutes while representing users temporal sequences of vectors both in the training and test phase.

To learn the different models  $H_{c1}^{t(i)}$  and  $H_{c2}^{t(i)}$  for predicting user prominence over time at each phase, we use "*Alpes-MaritimesDB*" dataset as our training dataset. Once the sequence interval is set up, we represent each user in this dataset by a sequence of features vector characterizing his/her behavior during each phase. Thus each user who has interacted during the *Alpes-Maritimes* flooding event would be represented by a sequence of 7 feature vectors at  $P1$ , a sequence of 37 feature vectors at  $P2$  and 144 feature vectors at  $P3$ . The length of these sequences was defined according to the duration of each phase. For example, as phase  $P1$  has lasted 3.5 hours, each user has to be represented by a sequence of 7 feature vectors at the end of  $P1$ . However, the duration of each phase would not be common to the same phase of all crisis events. Such duration varies from a disaster to another. To adjust this duration, we extend the duration of short phases of the *Alpes-Maritimes* flooding event by padding the sequences with zeroes so that all sequences extracted during the event considered for training and test have the same length and can be evaluated correctly. This padding technique is a common technique to deal with the problem of classifying sequences of variable length[24]. It has been already used in the literature for the classification of sentences of variable length, the translation of sentences, etc.

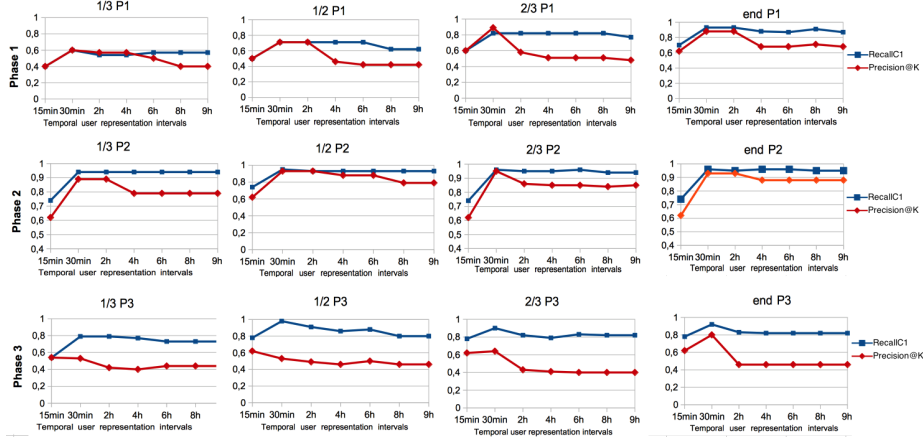


Figure 3: Prediction results comparison at **one-third, half, two-third** and the **end** of each disaster phase using different temporal sequence intervals ( $m$ ) for user representation.  $m = 30min$  is selected herein as the best interval value for temporal user behavior representation.

In other words, the fixed length of sequences used for padding the extracted sequences of vectors at each phase is defined in our experiments by selecting the longest duration among the phases characterizing the two analyzed flooding events. In order to make the trained model suitable for any flooding independently of the duration of its phases, sequence length could be defined a priori by domain experts by setting the longest duration that a phase could last during flooding disasters. This is also the case for sentences analysis or/and translation.

After representing each user by a sequence of vector characterizing his/her behavior at each phase, we proceed to the training of the different models  $H_{c1}^{t(i)}$  and  $H_{c2}^{t(i)}$ . Starting from the  $1^{st}$  hour of each phase of the Alpes-Maritimes flooding event, a new model is learned at each time-stamp  $m$  until the end of each phase. We thus learn 29  $H_{c1}$  models instead of 6  $H_{c1}$  (where  $m = 30$  minutes and P1 duration 3.5 hours) as we have extended the length of sequences at P1 to 30 in order to have the same sequence length in the test dataset for phase P1.

The parameters of the different  $H_{c1}^{P_j}$  and  $H_{c2}^{P_j}$  models are chosen as follows. We evaluate the models performance with tuning the parameters values relative to the number of states (1 – 4) and the number of multivariate Gaussian (1 – 4) with the training dataset. For each phase model, we select the parameters that yield the best Precision@K relative to the class  $C1_{P_j}$ .

## 7. Experimental Results

To experimentally validate our prominent microblog user prediction model during specific events, we compared its performance with several baselines as described below:

**Ours:** This refers to our proposed model which represents each user by a sequence of feature vectors characterizing the user behavior from the beginning of each analyzed phase independently of the other ones as described in Sections 3 and 4. It uses an additional Boolean feature  $Bf$  indicating if the user was detected as prominent in the previous phase or not. This feature is automatically extracted in the beginning of each new phase and set to 1 if the user was detected as prominent in the previous phase, or set to 0 if he/she was not.

**Pal:** This refers to the system built by Pal and Counts [1]. This system represents microblog users uniformly during the whole event by a single feature vector composed of 15 features. It classifies and ranks users according to their behavior from the beginning of the event without considering event phases. Through this state-of-the-art model, we aim to prove that considering the user activity during the previous event phases would erode the results.

**Pal\*:** This baseline uses the same specificities of Pal model presented above, however, it considers the different event phases while representing user’s activities. Through this baseline, we aim to prove that our phase-aware modeling approach can improve the prediction results of *Pal*.

**b1:** This baseline uses the same specificities of our model, but, it does not consider the Boolean feature  $Bf$ . Through this baseline, we want to evaluate the contribution of the Boolean feature  $Bf$  on enhancing the prediction results over time.

**b2:** This baseline follows the same user representation and classification principles used in our model. However, it is learned at each phase by referring to all the prominent microblog users  $\{C_1^{P_1} \cup C_1^{P_2} \cup C_1^{P_3}\}$  independently of their phase of prominence. Through this baseline, we aim to validate our assumption considering that user prominence has to be associated to each phase rather than the whole event.

**b3:** This model has the same specificities as our model. However, it characterizes users uniformly during the whole event. The model uses Corona to select the relevant features that better reflect users’ behavior during the whole event and not during each particular phase. Through this baseline, we evaluate the efficiency of our user behavior modeling consisting of characterizing user behavior differently at each phase.

### 7.1. Efficacy of the Real-time Prominence Prediction Model

Through the conducted experiments in this subsection, we evaluate the efficiency of our phase-aware model to predict prominent users. By comparing *ourmodel* performance with *Pal*, we evaluate the impact of considering the  $Bf$  feature, as the only indication of the user activity in the previous phase. We also evaluate the importance of characterizing and evaluating users per phase by comparing *Pal* and *Pal\** prediction results.

Figure 4 shows the prediction results obtained by the *ours*, *b1*, *Pal* and *Pal\** baselines at each time-stamp relative to each phase in terms of *Recall* and ranking (*Precision@K*). Additionally, Table 7 reports more detailed results of these baselines in terms of *Recall@10*, *Recall<sub>C2</sub><sup>Pj</sup>*, *CommonP1P2* and *CommonP2P3*

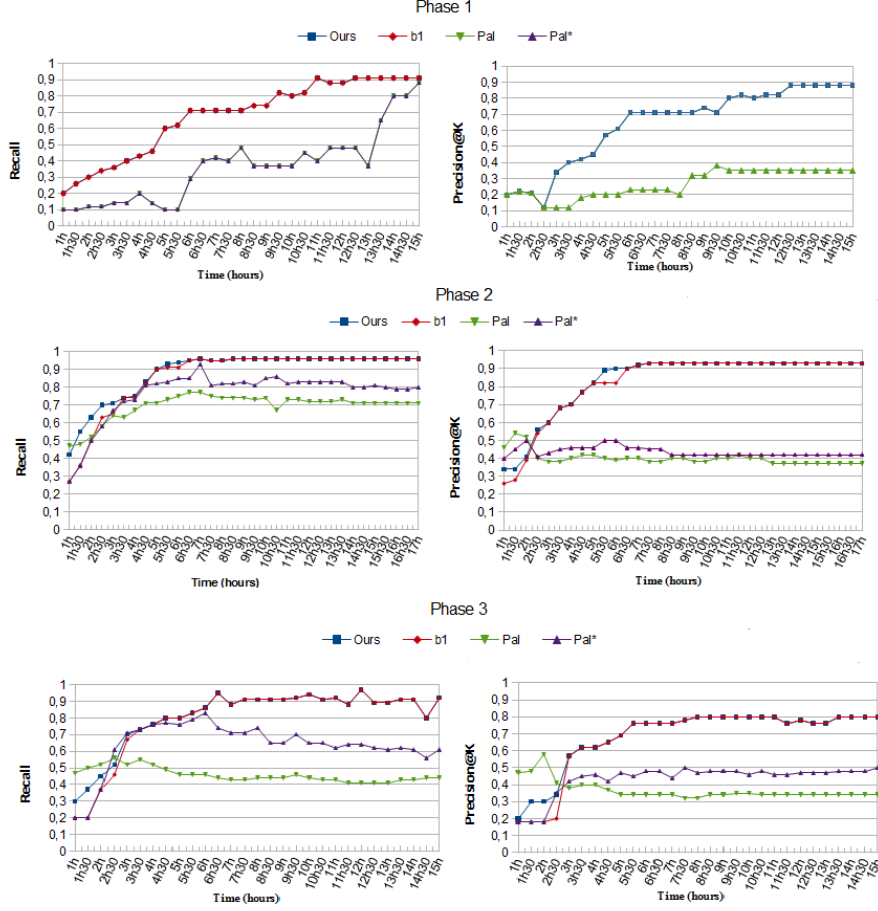


Figure 4: Comparing the prediction results of **ours** model with **Pal\***, **Pal** and **b1** baselines in terms of  $Recall_{C1Pj}$  and  $Precision@K$  during each phase. *Alpes-MaritimesDB* is used for training and *HeraultDB* for testing. At the first phase, *b1* and *Ours* are identical. Similarly, **Pal\*** and **Pal** are similar in P1 as there are no prior phases. The different results were registered while testing the model at different timestamps during the Herault floods.

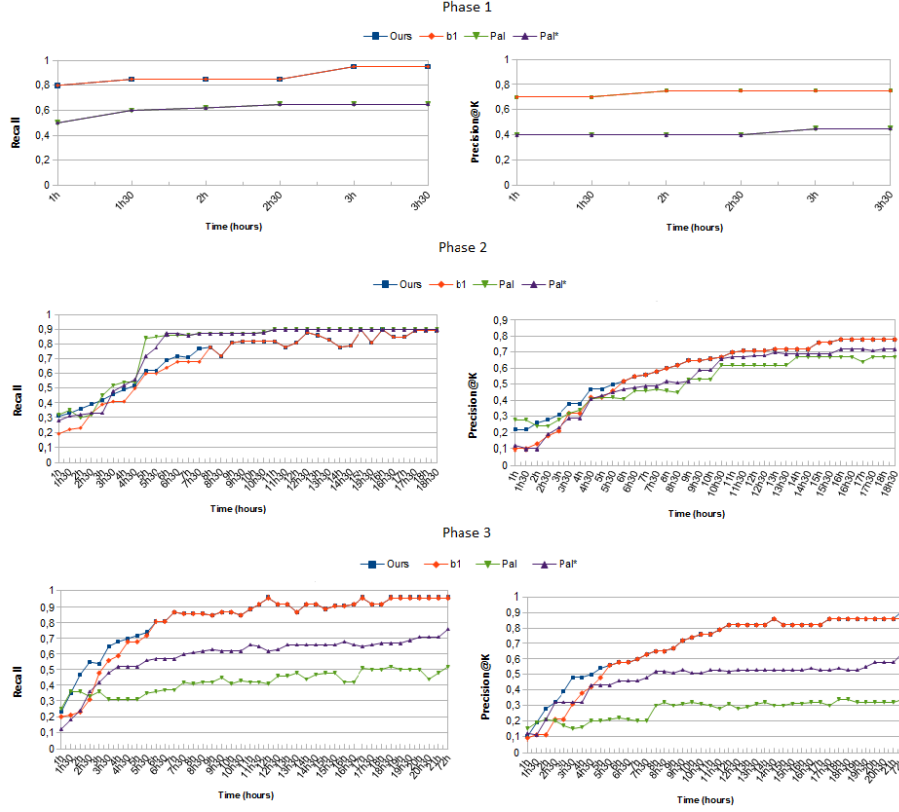


Figure 5: Prediction results comparison of **ours**, **Pal\***, **Pal** and **b1** baselines in terms of  $Recall_{C1Pj}$  and  $Precision@K$  during each phase. *HeraultDB* is used for training and *Alpes-MaritimesDB* for testing. At the first phase, *b1* and *Ours* are identical. Similarly, **Pal\*** and **Pal** are also identical in P1 as there are no prior phases. The different results were registered while testing the model at different timestamps during the Herault floods.

at the beginning, one-third, half and the end of each phase. *CommonP1P2* and *CommonP2P3* refer to the detection rate of the common prominent users defined in the  $\{C_1^{P1} \cap C_1^{P2}\}$  and  $\{C_1^{P2} \cap C_1^{P3}\}$  ground-truth prominent users sets respectively.

According to the prediction results based on the  $Recall_{C1}^{Pj}$  and  $Recall_{C2}^{Pj}$  measures, our model detects most of the prominent users at an early stage of each phase and discards a large number of the non-prominent ones. By evaluating the ranking results indicated by the *Precision@K* curves described in Figure 4, we observe that most of prominent users were detected and top ranked by our model at an early stage of the different Herault phases.

We also note that our model has detected all the top 10 prominent users (i.e. 100% Recall@10) after a few hours of each phase. Comparing these results with *Pal*, *Pal\** and *b1* baselines, our model performs the best in terms of prediction and ranking. Using the *Bf* feature, we succeed to identify more prominent users at the beginning of each phase compared to the *b1* model. This feature helps to identify common prominent users between the current and the previous phase as shown through *CommonP1P2* and *CommonP2P3* measures results. We also observe that *Pal* slightly outperforms our model and *Pal\** at the beginning of *P2* and *P3* as it detects most of the prominent users that were already detected in the previous phases by considering their tweeting activity from the beginning of the event. However, the performance of the baseline *Pal* erodes further with time as it is not able to detect the new prominent users relative to the current phase. Moreover, we note that *Pal\** which does not consider any information about user activity in prior phases outperformed *Pal* results after few hours. This validates our assumption. Using the phase-unaware model *Pal*, the new prominent users will not be favored with respect to the prior ones. To deal with our model “cold start” and obtain results similar or better results compared to those obtained by *Pal* at the beginning of each new phase, it would be more rational to keep tracking prominent users identified in the prior phase during the first 2 hours of the next phase in order to ensure the tracking of the common prominent users.

In order to prove the efficacy of our model for prominent users prediction independently of the duration of the crisis events phases, we train our model this time using *HeraultDB* and we test it using *Aples-MaritimesDB*. The prediction results of the obtained models are illustrated in Figure 5. According to these results, we observe that our model has detected most of prominent users even during the first phase which is characterized by a short duration of 3.5 hours. The obtained experimental results also confirm the comparison findings pointed through comparing the different models learned using *Aples-MaritimesDB* and tested using *HeraultDB*.

We conclude that the phase-unaware baseline *Pal* considering all the users’ activities in the previous phases leads to better results in the first hours of each phase compared to our phase-aware model. However such recorded prior activities would erode the model performance after few hours (*Pal* vs. *Pal\**). The obtained results also demonstrate the positive impact of the *Bf* feature

which improves the prediction results during the first hours of each new phase (*Ours* vs. *b1*). Such feature promotes users who were previously detected as prominent without biasing the real user activity and prominence during the analyzed phase.

## 7.2. Phase-aware vs Phase-unaware Models

Through the conducted experiments in this section, we aim to validate our assumption considering that the user prominence and behavior have to be associated with each event phase rather than the whole event. Thus, we compare our model with the phase-unaware baseline *b2*, and the phase-unaware-model *Pal* with *Pal\**. Both *Pal* and *b2* consider that user prominence has to be evaluated according to their prominence during the whole event and not per event phase. In this experiment, we evaluate the different baselines' performance to identify prominent users at the end of each phase. Figure 6 reports the prominent users' classification results of each baseline by the end of each phase of the Herault event.

According to the obtained results, we observe that phase-aware-models (*Pal* vs. *Pal\**) (*Ours* vs. *b2*) perform better than phase-unaware-models. The classification results of *Pal* and *Pal\** models are the same at P1, as it is the first phase and all the users' features are already set to zero for the two models. However, *Pal\** performs better than *Pal* in the next phases. The phase-unaware-users' representation of *Pal* promotes users who were prominent in the prior phases over the new prominent ones.

Comparing our phase-aware model with the phase-unaware model *b2*, we observe that *b2* registers low results at P1 and good results close to ours at P2 and P3 in terms of  $Recall_{C1}^{Pj}$ . These results can be explained by the fact that learning identification models by referring to all prominent users independently of their phases of prominence tends to bias the learning of the classification and ranking model.

To determine whether our model yields a statistically significantly higher precision than other models, we compare the aggregate Precision@K registered over time by our model using one-sided paired t-tests with the null hypothesis ( $H_0$ ) that the average Precision@K of the two models are the same and hypothesis  $H_a$ , that the mean Precision@K of our model is higher, within a 99% confidence interval. We have chosen to apply the one-sided t-test for our experiments as we have to test the statistical significance in one direction. We need to prove that  $H_a$  is valid (i.e. the mean of precision@K results obtained by our model at each phase are greater than those obtained by the other models).

Table 8 shows t-test results of our model versus the other baselines. We reject  $H_0$  in all the cases except for *b1* at the different phases. Overall, we establish that Precision@K scores registered by our model are higher than the other baselines. As  $H_0$  is rejected for *b1*, we note here that the *bf* feature added in our model did not significantly help to improve the detection of prominent users over time. We establish that the other baseline models except *b1* performed worse, and we thus conclude that our model outperformed all four baseline models (i.e. *b2*, *b3*, *Pal* and *Pal\**) at the different phases.

Table 7: Prediction performance comparison of the models: **ours**, **Pal** and **b1** at the **beginning**, **one-third**, **half** and at the **end** of each event phase.

	<b>Ours</b>			<b>b1</b>			<b>Pal</b>			<b>Pal*</b>		
				Phase 1								
Metrics	2h	6h	8h	15h	2h	6h	8h	15h	2h	6h	8h	15h
<i>Recall<sub>C1</sub></i>	30%	71%	71%	91%	30%	71%	71%	91%	12%	12%	20%	88%
<i>Precision<sub>C1</sub></i>	10%	9%	8%	9%	10%	9%	8%	9%	5%	5%	6%	11%
<i>Recall<sub>C2</sub></i>	93%	92%	92%	92%	93%	92%	92%	92%	97%	97%	96%	92%
<i>Precision<sub>C2</sub></i>	97%	97%	97%	99%	97%	97%	97%	99%	97%	97%	98%	98%
<i>Recall@10</i>	40%	100%	100%	100%	40%	100%	100%	100%	30%	30%	50%	100%
<i>Precision@K</i>	40%	82%	82%	89%	40%	82%	82%	89%	20%	25%	57%	29%
Phase 2												
	2h	7h	9h	17h	2h	7h	9h	17h	2h	7h	9h	17h
<i>Recall<sub>C1</sub></i>	63%	96%	96%	95%	50%	96%	95%	94%	52%	77%	74%	71%
<i>Precision<sub>C1</sub></i>	11%	12%	10%	12%	20%	15%	10%	15%	17%	16%	17%	16%
<i>Recall<sub>C2</sub></i>	88%	81%	78%	92%	92%	85%	85%	85%	93%	89%	90%	90%
<i>Precision<sub>C2</sub></i>	98%	99%	99%	99%	98%	99%	99%	99%	95%	96%	96%	96%
<i>Recall@10</i>	90%	100%	100%	100%	60%	100%	100%	100%	90%	100%	100%	100%
<i>Precision@K</i>	40%	92%	95%	95%	55%	95%	95%	95%	52%	41%	41%	37%
<i>CommonP1P2</i>	90%	100%	100%	100%	68%	100%	100%	100%	86%	95%	95%	90%
Phase 3												
	2h	6h	8h	15h	2h	6h	8h	15h	2h	6h	8h	15h
<i>Recall<sub>C1</sub></i>	45%	86%	97%	95%	37%	86%	91%	91%	52%	46%	44%	44%
<i>Precision<sub>C1</sub></i>	12%	16%	14%	12%	14%	16%	14%	14%	13%	11%	11%	10%
<i>Recall<sub>C2</sub></i>	88%	91%	87%	92%	93%	91%	87%	88%	92%	92%	92%	92%
<i>Precision<sub>C2</sub></i>	97%	99%	99%	99%	98%	99%	99%	99%	96%	96%	96%	96%
<i>Recall@10</i>	50%	90%	100%	100%	40%	90%	100%	100%	60%	60%	50%	50%
<i>Precision@K</i>	55%	53%	55%	59%	50%	53%	59%	58%	60%	58%	53%	53%
<i>CommonP2P3</i>	66%	90%	80%	100%	57%	90%	95%	90%	95%	90%	90%	90%

Table 8: Our model vs the other baselines models. One sided t-test of registered Precision@K per model during each phase. H0 is rejected in all cases except for b1 at the different phases for a significance level of 0.01.

Baselines	Phase 1	Phase 2	Phase 3
b1		0.011	0.011
b2	0.006	0.001	0.001
b3	0.005	0.001	0.001
Pal	0.001	0.001	0.001
Pal*	0.002	0.004	0.001

Overall, we conclude that the consideration of event phases for representing user activity during the event leads to a better prominent users detection. Evaluating and representing microblog users according to their prominence at each phase would guarantee the construction of a more efficient prediction model (As demonstrated by the comparison of *Ours* vs. *b2*) and ensure a fair evaluation for all users at any time of the event (As demonstrated by the comparison of *Pal\** vs. *Pal*).

### 7.3. Phase-based User Characterization

Through the conducted experiments in this subsection, we aim to prove the importance of modeling users behavior differently according to the particularities of each phase. Our model is compared with the *b3* model which characterizes users uniformly using the same features during the different phases. Figure 6 reports the results of this experiment.

By referring to the different evaluation metrics, our approach performs better than *b3* for both the classification and ranking of prominent users in the different phases. *b3* failed to identify the prominent users in P1. It registered a Precision@K of 40% and 50% respectively at P1 and P3 (e.g. only 40% of prominent users were ranked on top K at P1). We also observe that modeling users uniformly during the whole event leads to good results only for phases characterized by high activity of prominent users such as P2. *b3* registered a high Precision@K score of 90% at P2 which is a phase characterized by a high activity of users. However, it has failed to identify the prominent ones during phases recording a low activity regarding the event topic such as P1 and P3.

Characterizing users' behavior differently at each phase would highlight the relevant users behavior features relative to each phase. As demonstrated in these experiments, such characterization improves the identification results.

In order to better understand users behavior differences at different phases, we analyzed in Table 9 the nature of features selected by Corona at each phase in the pre-processing step. According to the obtained results, we observe that the number of selected on-topic features is close to the number of off-topic ones in P2 which is not the case in P1 and P3. This can be explained by the fact that the behavior of prominent microblog users during P1 and P3 would be similar to their behavior in regular days as the danger is either not yet confirmed or discarded. In such situations, users would share relevant information regarding

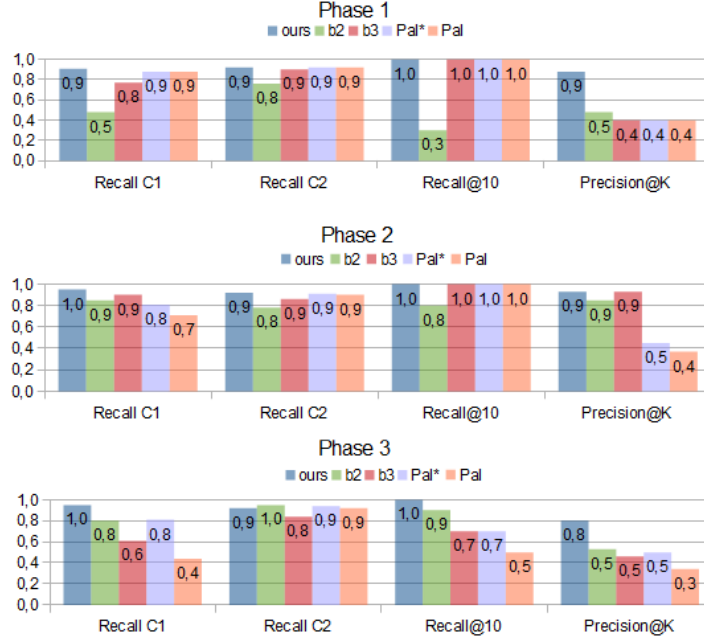


Figure 6: Comparison of the classification and ranking results of our model (**ours**), and other phase-unaware models *Pal*, *b2* and *b3* at the end of the different phases **P1**, **P2** and **P3**.

the disaster but keep also tweeting about other topics. Thus, there is no need to penalize them regarding their off-topic behavior.

However, during P2, prominent microblog users who are generally concerned by the disaster would be in panic and would frequently share updates describing what they are seeing, hearing and experiencing. They would focus mainly on sharing the disaster event news. Thus, it is more rational to consider off-topic features (i.e. on-engineered features adjusted by the off-topic ones or off-topic raw-features) in order to penalize users toggling among different topics and who are not necessarily concerned by the disaster. Using this strategy, the identification model would rank users active only regarding the disaster higher than those who are extremely active in several topics (e.g. news outlet users).

Through these experiments, we have shown the importance of selecting the most appropriate features for each event phase. This phase-based feature selection highlights the behavioral differences between prominent and non-prominent users, and hence improves the precision and the efficiency of the prediction model.

#### 7.4. Adequacy of the Feature Selection Algorithm

Through the previous experiments, we have shown the importance of the feature selection process per phase. In these experiments, we evaluate the ap-

Table 9: Statistics about the selected feature categories using the different feature selection algorithms. **On** and **Off** refer respectively to on- and off-topic raw and engineered (**Eng**) features.

Algorithm	Phase 1				Phase 2				Phase 3			
	Raw vs Eng		On vs Off		Raw vs Eng		On vs Off		Raw vs Eng		On vs Off	
	Raw	Eng	On	Off	Raw	Eng	On	Off	Raw	Eng	On	Off
<b>Corona</b>	<b>6</b>	<b>3</b>	<b>6</b>	<b>3</b>	<b>5</b>	<b>4</b>	<b>5</b>	<b>4</b>	<b>6</b>	<b>3</b>	<b>7</b>	<b>2</b>
Clever	4	5	5	4	4	6	3	6	6	3	7	2
ReliefF	4	5	8	1	4	6	5	4	4	5	7	2
Average	0,52	0,48	0,7	0,3	0,48	0,59	0,48	0,52	0,59	0,41	0,78	0,22

Table 10: Performance comparison of different feature selection algorithms for the detection of prominent users at each phase in terms of  $Recall_{C1}^{Pj}$  and  $Precision@K$ .

Algorithm	Phase 1		Phase 2		Phase 3	
	Recall C1	Precision@K	Recall C1	Precision@K	Recall C1	Precision@K
<b>Corona</b>	<b>0.91</b>	<b>0.89</b>	<b>0.95</b>	<b>0.95</b>	<b>0.95</b>	<b>0.59</b>
Clever	0.42	0.7	0.94	0.43	0.74	0.51
ReliefF	0.31	0.6	0.82	0.5	0.62	0.48

appropriateness of the chosen feature selection algorithm. Thus, we compare our adopted algorithm Corona with the following two feature selection algorithms:

**Clever** [25] belongs to the family of unsupervised feature subset selection methods for multivariate time-series based on principal component analysis.

**ReliefF** [26] is a supervised feature selection algorithm which selects relevant features which works only on vectorized data. To apply this technique, we vectorized each time-series sequence representing each user by summing the values of the same features recorded at each time-stamp.

Table 9 describes the statistics of the different categories of the selected features by each algorithm. According to these statistics, we observe that the number of selected raw, engineered, on- and off-topic (except P1 for on and off) by the different algorithms is nearly the same for the different phases. For the phases P1 and P2, we note that there is a low number of off-topic features considered compared to the number of on-topic ones.

As the selected features by the different algorithms are not necessarily the same even if they belong to the same category, we trained our model using the selected features by each feature selection algorithm in order to evaluate their effectiveness. Table 10 describes the obtained results by the different models for the selected prominent users class  $C1$  at the end of each phase in terms of  $Recall_{C1}^{Pj}$  and  $Precision@K$ . We observe that the features selected by Corona give the best results.

Through these experiments, we have shown that the vectorization of the time-series representation without taking into account the different correlations of data hides the real importance of each feature. Thus, the choice of an appropriate feature selection algorithm has to take into consideration the temporal distribution of user behavior over time.

## 8. Discussion

The presented phase-aware probabilistic model performs prominent microblog users prediction during crisis events. The level of importance of these events evolves over time. This evolution has to be considered while modeling users behavior and evaluating users prominence over time. Figure 7 shows a comparison of the ROC curves of our model considering the impact of this evolution and all the other considered baselines in our experiments. These curves confirm the importance of considering the user behavior change over event phases when evaluating their prominence over time.

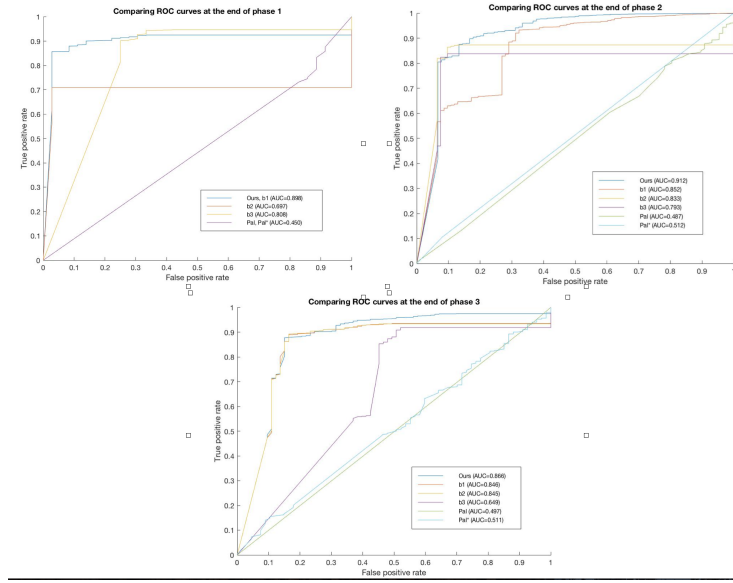


Figure 7: Comparison of the ROC curve of our model (**ours**) with the other baselines *Pal\**, *Pal*, *b1*, *b2* and *b3* at the end of each different phases **P1**, **P2** and **P3**.

Prominent users change at each new phase. As demonstrated by our ground-truth study, only few prominent users have been prominent from the first phase until the last one. We have also noted that only 2% of all the users who have interacted regarding the disaster were labeled as prominent. Such statistics were expected. During crisis events, many microblog users share or/and report event-related-information. However, few of them would share the exclusive and relevant information required by emergency teams. Through our experiments, we have also shown that neglecting users prominence phase in the user behavior learning step would lead to overfitting. A learned model in a such way would not be able to differentiate between the true prominent users over phases (*b2* vs. *Ours*).

By comparing *Pal* vs. *Pal\** prediction results described in Figure 6, we have observed that a per-phase user behavior modeling approach improves significantly the identification results. In phases two and three, *Pal\** has detected

most of the prominent users relative to each phase. However, *Pal* has not succeeded to detect most of prominent users from phase to phase. A phase-unaware modeling approach cannot ensure a fair evaluation for all users at any time of the event.

Our strategy to model users behavior change according to the event evolution has also proved its effectiveness. We have shown that representing users uniformly using the same features (*b3* vs *ours*) would not reflect the real behavior of users at each particular phase. Users behavior changes during the event according to the event evolution. As reported in Table 9 which details the statistics of the selected features at each phase using different algorithms, around 48% of the selected features were on-topical and 52% were off-topical in the second phase. This can be argued by the fact that real prominent users during this phase are generally in panic, so, they tend to focus their attention only on what is happening during the disaster by sharing only on-topical information. Thus, by considering fairly both on- and off-topical metrics in such phase, the identification model will be able to distinguish between microblog users who are toggling between several topics and those active only regarding the disaster. We also note that during the first and last phase, the off-topical features were not extensively considered by the selection algorithms. This can be explained by the fact that in such phases there is no potential danger thus even prominent users tend to be active regarding on- and off-topics. In such cases, the off-topical features can not make prominent and non-prominent users more distinguishable.

We have also shown that our temporal sequence representation approach characterizing the user activity details at different timestamps of each event-phase has proved its importance. In Figure 3, we have shown that our model performance tends to decrease if we consider longer intervals between the different timestamps. The more we detail the user activities differences by considering several timestamps, the more the identification results are better. Highlighting the temporal distribution of user activity can point out the hidden patterns reflecting the prominence of each user according to his behavior over time during each phase.

Lastly but most importantly, we have demonstrated that our model can identify prominent users in real time at an early stage of each event phase. For example, 63% of prominent users were detected after two hours from the beginning of the most important phase which is the second one. Even with learning our classification models *a priori* using similar events data, as described in Figure 4, our model outperforms the state-of-the-art phase-aware and unaware models (*Pal* and *Pal\**) which are using unsupervised algorithms for classifying and ranking microblog users. We have also shown that with considering the user prominence in prior phases –using the *Bf* feature–, we can detect more prominent users at the first hours of each event phase. As reported in Table 7 by referring to the *CommonProm* measure results of *b1* and *Ours*, we succeed to detect the common prominent users relative to the prior and the current phases from the first hours. However, we note that the *Pal* models outperform our model on the first hours of each phase.

## 9. Conclusion and Future Work

This paper has proposed a phase-aware prediction model for detecting prominent microblog users during unexpected events. It is based on a new user modeling approach taking into account both the user behavior and the event evolution over time. Using this approach, microblog users are characterized differently in the beginning of each event phase using the best relevant features that can characterize their behavior according to the analyzed event phase particularities. Users are evaluated according to their prominence per phase in order to ensure a fair evaluation.

Through the conducted experiments, we showed that our prediction model significantly outperforms state-of-the-art models by detecting most of the prominent users at an early stage of each phase. We also proved that associating user’s prominence with event phases ensures a fair evaluation for all users at each phase. We also demonstrated that our proposed user modeling approach characterizing users using different features at each event phase improves the detection results and helps to highlight the user behavior differences according to each phase specificities. We showed that the choice of the feature selection algorithm has to be in harmony with the selected user characterization.

Overall, we conclude that the different aspects considered in our model have each contributed to reach these results. This model is not designed only for flooding events. It could also work for other types of crisis events by using other datasets reflecting user behavior during similar events. This would not require changing the features characterizing user behavior, since the features proposed in this paper can also be used to describe microblog users in the context of any event type. At the same time, new features can easily be integrated into our model.

For future work, we aim to automatically extract the keywords related to each disaster phase. We thus can study if the extraction of phase-aware keywords would improve the model performance or not. We also wish to propose a more dynamic user behavior modeling approach by automatically detecting the user behavior state change over time. Users would not be necessarily characterized by temporal sequences having the same length at a specific time-stamp. We also aim to reduce the length of sequences, with a new modeling approach based on sessions series rather than time series. We also wish to detect the boundaries of the different phases by automatically analyzing information provided by meteorological departments on Twitter.

## 10. Acknowledgements

We thank our colleagues from the L3i laboratory who have greatly participated in the construction of the ground-truth of our databases.

Funding: This work was supported by the Poitou Charentes Department.

## 11. References

- [1] A. Pal, S. Counts, Identifying Topical Authorities in Microblogs, in: WSDM, ISBN 978-1-4503-0493-1, 45–54, doi:10.1145/1935826.1935843, 2011.
- [2] S. Xianlei, Z. Chunhong, J. Yang, Finding Domain Experts in Microblogs, in: WEBIST, 2014.
- [3] D. M. Romero, W. Galuba, S. Asur, B. A. Huberman, Influence and Passivity in Social Media, in: WWW, ISBN 978-1-4503-0637-9, 113–114, 2011.
- [4] H. Kwak, C. Lee, H. Park, S. Moon, What is Twitter, a Social Network or a News Media?, in: WWW, ISBN 978-1-60558-799-8, 591–600, 2010.
- [5] J. Weng, E.-P. Lim, J. Jiang, Q. He, TwitterRank: Finding Topic-sensitive Influential Twitterers, in: WSDM, ISBN 978-1-60558-889-6, 261–270, 2010.
- [6] M. Zhang, C. Sun, W. Liu, Identifying Influential Users Of Micro-Blogging Services: A Dynamic Action-Based Network Approach., in: PACIS, 223, 2011.
- [7] S. Ghosh, N. Sharma, F. Benevenuto, N. Ganguly, K. Gummadi, Cognos: Crowdsourcing Search for Topic Experts in Microblogs, in: SIGIR, ISBN 978-1-4503-1472-5, 575–590, 2012.
- [8] W. Chen, Y. Wang, S. Yang, Efficient Influence Maximization in Social Networks, in: KDD, ISBN 978-1-60558-495-9, 199–208, doi:10.1145/1557019.1557047, 2009.
- [9] I. Bizid, N. Nayef, P. Boursier, S. Faiz, A. Doucet, Identification of Microblogs Prominent Users during Events by Learning Temporal Sequences of Features, in: CIKM, 1715–1718, 2015.
- [10] M. Perez-Lugo, Media Uses in Disaster Situations: A New Focus on the Impact Phase, *Sociological Inquiry* 74 (2) (2004) 210–225, ISSN 1475-682X.
- [11] K. Yang, H. Yoon, C. Shahabi, A Supervised Feature Subset Selection Technique for Multivariate Time Series, in: FSDM, 2005.
- [12] I. Bizid, N. Nayef, P. Boursier, S. Faiz, J. Morcos, Prominent Users Detection during Specific Events by Learning On- and Off-topic Features of User Activities, in: ASONAM, 500–503, 2015.
- [13] A. Java, Pranam, T. Finin, T. Oates, Modeling the Spread of Influence on the Blogosphere, in: WWW, 2006.
- [14] D. Boyd, S. Golder, G. Lotan, Tweet, Tweet, Retweet: Conversational Aspects of Retweeting on Twitter, in: HICSS, ISSN 1530-1605, 1–10, 2010.

- [15] C. Honey, S. Herring, Beyond Microblogging: Conversation and Collaboration via Twitter, in: HICSS, ISSN 1530-1605, 1–10, 2009.
- [16] L. Deng, N. Jaitly, Deep discriminative and generative models for pattern recognition, Tech. Rep. MSR-TR-2015-59, URL <http://research.microsoft.com/apps/pubs/default.aspx?id=251677>, 2015.
- [17] A. P. Dempster, N. M. Laird, D. B. Rubin, Maximum likelihood from incomplete data via the EM algorithm, *Journal of the Royal Statistical Society* 39 (1) (1977) 1–38.
- [18] L. E. Baum, J. A. Eagon, An inequality with applications to statistical estimation for probabilistic functions of Markov processes and to a model for ecology, *Bull. Amer. Math. Soc.* 73 (3) (1967) 360–363.
- [19] A. Olteanu, C. Castillo, F. Diaz, S. Vieweg, CrisisLex: A Lexicon for Collecting and Filtering Microblogged Communications in Crises, in: *Proc. of the 8th International Conference on Weblogs and Social Media*, 376–385, URL <http://crisislex.org/>, 2014.
- [20] I. Bizid, P. Boursier, J. Morcos, S. Faiz, MASIR : A Multi-agent System for Real-time Information Retrieval from Microblogs during Unexpected Events, in: *KES-AMSTA-15*, 1–8, 2015.
- [21] D. Spina, J. Gonzalo, E. Amigó, Discovering Filter Keywords for Company Name Disambiguation in Twitter, *Expert Systems with Applications* ISSN 09574174, doi:10.1016/j.eswa.2013.03.001, URL <http://dx.doi.org/10.1016/j.eswa.2013.03.001>.
- [22] P. Karisani, F. Oroumchian, M. Rahgozar, Tweet Expansion Method for Filtering Task in Twitter, *Springer International Publishing, Cham*, 55–64, 2015.
- [23] Y. Bengio, P. Simard, P. Frasconi, Learning Long-term Dependencies with Gradient Descent is Difficult, *Trans. Neur. Netw.* 5 (2) (1994) 157–166, ISSN 1045-9227, doi:10.1109/72.279181.
- [24] K. Cho, B. van Merriënboer, Ç. Gülçehre, F. Bougares, H. Schwenk, Y. Bengio, Learning Phrase Representations using RNN Encoder-Decoder for Statistical Machine Translation, *CoRR* abs/1406.1078.
- [25] K. Yang, H. Yoon, C. Shahabi, CLeVer: A Feature Subset Selection Technique for Multivariate Time Series, in: *PAKDD*, 516–522, 2005.
- [26] M. Robnik-Šikonja, I. Kononenko, Theoretical and Empirical Analysis of ReliefF and RReliefF, *Mach. Learn.* 53 (1-2) (2003) 23–69, ISSN 0885-6125.