A Metadata-based Event Detection Method using Temporal Herding Factor and Social Synchrony on Twitter Data

Nirmal Kumar Sivaraman, Vibhor Agarwal, Yash Vekaria, Sakthi Balan Muthiah
The LNM Institute of Information Technology, Jaipur, India
{nirmal.sivaraman, vibhor.agarwal.y16, yash.vekaria.y16, sakthi.balan}
@lnmiit.ac.in

Abstract. Detecting events from social media data is an important problem. In this paper, we propose a novel method to detect events by detecting traces of herding in the Twitter data. We analyze only the metadata for this and not the content of the tweets. This makes our method computationally less intensive as compared to the existing methods in the literature. We evaluate our method on a dataset of 3.3 million tweets that was collected by us. We then compared the results obtained from our method with a state of the art method called Twitinfo on the above mentioned 3.3 million dataset. Our method showed better results. To check the generality of our method, we tested it on a publicly available dataset of 1.28 million tweets and the results convey that our method can be generalised.

Keywords: Event detection · Temporal herding factor · Social network analysis

1 Introduction

In this modern digital connected era, online societal engagement is in abundance. Twitter is a platform that is used by people to broadcast text, pictures, videos, etc. regarding different topics of their liking. In this work, we study the online engagements in Twitter and examine if we can find a definite behavioral trait for tweets concerning events without looking at the content of the tweets. This is a novel approach to event detection. Our hypothesis here is that the content that is posted in Twitter about events will generate herd behavior. Hence, we propose a method to quantify this herd behavior, not by looking at the content but just by looking at the metadata (including hashtags) of the tweets, and use this to detect events.

The working definition of an *event* is as follows – something that happens and captures the attention of many people. In case of online social media like Twitter, *measuring the attention* is equivalent to measuring whether they are putting any tweet about what has happened. Time period (or duration) is one of the main characteristics of an event. For example, the time period for an event

like a football match will usually be for a few hours and that for an event like Olympics will be a few weeks time. Any event that has a substantial impact on the society will be a talking point in the social media for at least a few days. In this work, we use one day as the granularity of time to analyze the events that span over at least a few days. Our hypothesis here is that the events that are important or significant to people will be in focus in the social media at least for a few days.

There are a lot of works on event detection in the literature. According to [12], the event detection methods can be broadly classified into four,

- Term-interestingness-based approaches,
- Topic-modeling-based approaches,
- Incremental-clustering-based approaches, and
- Miscellaneous approaches.

Term-interestingness-based approaches rely on tracking the terms (from the Twitter data stream) that are likely to be related to an event [15] [10]. Topic-modeling-based approaches depend on the probabilistic topic models to detect real-world events by identifying latent topics from the Twitter data stream [21] [5]. In this approach, each tweet is associated with a probability distribution over various latent topics to find the hidden semantic structures from a collection of tweets used to guide the event detection task. These methods rely on sophisticated models to infer latent topics. As traditional clustering algorithms usually require the total number of clusters to be fixed, it is difficult to predict the total number of expected event clusters in advance for high-volume, real-time Twitter data where a wide variety of topics are discussed. Incremental-clustering-based approaches follow, at their very core, a clustering strategy, which is incremental in nature, in order to avoid having a fixed number of clusters [11] [24]. Miscellaneous approaches are the ones that adopt hybrid techniques, which do not directly fall under the three categories discussed above [1] [8] [23].

In this work, we propose a novel method for event detection using a novel measure called *Temporal Herding Factor (THF)*. Our approach to event detection is a term interestingness approach [12], where we consider hashtags as the terms at a granularity of time of one day. We use the idea of social synchrony [22] to detect events using THF to quantify the traces of herding in the Twitter data. When there is herding, we consider that there is corresponding event. Importantly, we use only metadata to detect events. This makes the model simple and easy to process, especially when compared to the clustering approaches.

For evaluation of our work, we collected a dataset that contains 3.3 million tweets. We call this dataset as 3.3M dataset[1]. We got precision, recall and $F1$ score of 0.76, 0.89 and 0.82 respectively.

To check the generality of our method, we considered the generic dataset that has no geotags and is from a different time period. We tested this dataset using the same thresholds that were calculated for the 3.3M dataset. We got a precision, recall and $F1$ score of 0.70, 0.97 and 0.81 respectively. The dataset

[1] We will make the data and code publicly available once the paper is accepted.

that is available publicly contains 1.28 million tweets (We call this dataset as generic dataset). Also we compared our results with a state of the art method called Twitinfo that is closest to our approach. We observed that our results are better.

This paper is structured as follows. In the next section, we describe the related works. In Section 3, we introduce our model to detect events from Twitter data. In Section 4, we evaluate our results and compare them with the state of the art model. In the section following that, we discuss the generality of our method. Paper concludes with the Section 7.

2 Related Works

In this section, we describe the event detection methods that are relevant to our method. In [13], Top-k bursty word segments are identified within a specific time window and then Jarvis-Patrick clustering algorithm is used. Events are ranked using newsworthiness score, calculated using Wikipedia. Work presented in [16] detects high-frequency terms within a specific time window. Tweets are grouped based on term co-occurrence, following a greedy strategy. In [2], shift in correlation values of tag pairs are identified. Grouping is performed based on co-occurrence of tag pairs in a minimum number of documents. Events are ranked based on an average burstiness of a topic and an average number of documents containing all tags of a topic. A specialized term scoring measure is presented in [10] which is utilized to retrieve top-K terms from a dynamic temporal window. A modified SFPM algorithm [19] [9] is used to group related terms in a topic. Ozdikis et al. in [18] proposed four event detection methods in Twitter. These methods differ only in the way the tweet vector is created. They construct the tweet vectors in four ways – using Words in Tweets without Semantic Expansion, using Words with Semantic Expansion, using Hashtags without Semantic Expansion and using Hashtags with Semantic Expansion. After that they cluster the tweets to find the events.

Our approach to event detection falls under the term interestingness approach where we consider hashtags as the terms. In [12] it is mentioned that Twitinfo method [15] has the best F1 score among the term interestingness approaches. Hence, we discuss the work on Twitinfo [15] briefly here. The Twitinfo model [15] makes a frequency plot of the time line of tweets. Then they calculate a historically weighted running average of tweet rate and find the rate that are prominently higher than the mean tweet rate. Algorithm starts a window at i^{th} time slice, if the bin count at time slice i is more than threshold (T) mean deviations from the current mean. This update requires another parameter $\alpha < 1$ to capture the historical information. Twitinfo uses the parameters: $p = 5$, $T = 2$ and $\alpha = 0.125$. This model is the closest one to our model.

To summarize their algorithm, when the algorithm encounters a prominent increase in bin count relative to the historical mean, it starts a new window and follows the rise to its surge. The surge's window terminates once either the bin

count returns to the same level it started at or if there occurs another significant increase.

There are two key parameters in this model – $Mean$ and *Mean deviation*. $Mean$ is initialized to the first bin count in the beginning and *Mean deviation* to the variance of the first p bins.

We compare the results of our model with the results obtained from Twitinfo in Section 4.2.

3 A Model to Detect Online Events

We detect events by using the ideas of herding that is calculated as THF and social synchrony. In this section, we discuss Herding, our formulation of THF, social synchrony and our methodology.

3.1 Social Synchrony

According to [22], surge and social synchrony are defined as follows:
Surge: *a social phenomenon where many agents perform some action at the same time and the number of such agents first increases and then decreases.*
Social Synchrony: *a surge where the agents perform the same action.*

To characterize a phenomenon as a social synchrony, the following steps are important.

- Defining the criteria for agents to be considered for observation
- Detecting the surge in the number of agents under observation
- Defining the criteria to measure the *sameness* of the agents' actions

The problem of detecting the presence of events may be described as detecting social synchronies in Twitter with the following criteria:

- The criteria for agents to be considered for observation - all the users tweeting with the same hashtag
- The surge in the number of agents may be found using the Algorithm 1.
- The criteria to measure the *sameness* of the agents' actions - tweeting with the same hashtag and the parameter that we introduce in this paper called *Temporal Herding Factor* of the surge being above a threshold value. This is discussed in detail in section 3.3.

3.2 Herding

One of the early paper on this subject is [4] that studies herding behaviour in a population of hypothetical agents who make choices between assets based on their own information and also on the observed behaviour of other agents. It visualizes herding as a phenomenon in which people tend to converge on similar behaviour giving rise to a situation where *everyone is doing what everyone else*

Algorithm 1 Algorithm to detect the surge in the number of agents performing an action [22]

1. Observe the activity in a continuous time period.
2. Divide observation into equal time slices.
3. Count the number of unique agents who carried out the action in each time slice and plot it.
4. Detect the maxima (surges) of the observation and label the time slices at which it occurs as $p_1, p_2, p_3, ...p_n$.
5. Do the steps 6, 7 and 8 for each surge in $\{p_1, p_2, p_3, ...p_n\}$.
6. Fix a threshold value L (as a % of x_{p_i}). The surge x_p corresponds to a surge only if the activity goes below L% of x_{p_i} between p_{i-1} and p_i and between p_i and p_{i+1}, where x_{p_i} is the count of unique agents at time slice p_i.
7. Detect two minimas a and b that lie before and after the surge respectively such that
   $x_a < x_{p_i} \frac{L}{100}$
   $x_b < x_{p_i} \frac{L}{100}$
8. **If** such points do not exist
   then There is no surge corresponding to p_i.
   else The activity between time slices a and b is a surge.

is doing. In the cognitive science literature, we find the definition of herding as – *A form of convergent social behaviour that can be broadly defined as the alignment of the thoughts or behaviours of individuals in a group (herd) through local interaction and without centralized coordination* [20]. At a behavioural level, the most popular form of herding behaviour is the tendency to imitate results [4]. Herding behaviour usually occurs when individuals alter their private beliefs to correspond more closely with the publicly expressed opinions of others [6] [3].

Traces of herding could be found in the online social networking websites such as Twitter. A study on how the re-tweet count, or the number of others who have already forwarded a message, influences people's spreading of disaster-related tweets is present in [14]. This says that retweets can be taken as markers of *the tendency to imitate results*. Here, we assume that most of the users that retweet agree with the contents of the original tweet. Hence, we consider retweets as a marker of herding. Based on the retweeting behaviour, we formulate a parameter to detect herding in Twitter. This is discussed in the following section.

3.3 Temporal Herding Factor (THF)

To detect herding behavior in the Twitter users that tweet regarding a hashtag, we observe their tweeting activity. At each time slice, we consider the new users with respect to the previous time slice and find out the fraction of them who retweets. We call this parameter *Temporal Herding Factor (THF)*.

We consider all the hashtags that are present in the dataset. We first take the list of all the hashtags and consider the set of hashtags as $H = \{h_1, h_2, h_3, ..., h_n\}$. The set of users who tweet regarding a topic h_j are represented as:

$$U^{h_j} = \{u_{j1}, u_{j2}, u_{j3}, ..., u_{jm}\}.$$

A surge is the distribution of tweets regarding a hashtag where the number of tweets increases first and then decreases. Let a surge with respect to hashtag h be represented as S_h, S_h be divided into N time slices of equal-length and t_i denote the i^{th} time slice of the surge where $i \in \{1, 2, ..., N\}$.

Let $U^h_T(t_i)$ denote the set of all the unique users who posted *tweet(s)* that are not *retweets* related to the hashtag h in the time slice t_i, $U^h_{RT}(t_i)$ denotes the set of all the unique users who retweeted related to the hashtag h in the time slice t_i and $U^h_{all}(t_i)$ denotes the set of all the unique users involved in the tweeting or retweeting activity related to the hashtag h in the time slice t_i.

$$U^h_{all}(t_i) = U^h_T(t_i) \cup U^h_{RT}(t_i)$$

THF at the i^{th} time slice (t_i) of S_h is defined as follows:

$$THF(t_i) = \begin{cases} 0 & : \text{if } |U^h_{all}(t_i) - U^h_{all}(t_{i-1})| = 0 \\ \frac{|U^h_{RT}(t_i)|}{|U^h_{all}(t_i)|} & : \text{if } t_i = t_1 \\ \frac{|U^h_{RT}(t_i) - U^h_{all}(t_{i-1})|}{|U^h_{all}(t_i) - U^h_{all}(t_{i-1})|} & : \text{otherwise} \end{cases}$$

Here, $|U^h_{RT}(t_i) - U^h_{all}(t_{i-1})|$ represents the number of all the unique users who have retweeted with the hashtag h in the time slice t_i but have not tweeted or retweeted in t_{i-1}. When the combined set of all the unique users tweeting or retweeting with the hashtag h are same for two consecutive time slices t_i and t_{i-1} (i.e., $|U^h_{all}(t_i) - U^h_{all}(t_{i-1})| = 0$), then $THF(t_i)$ is considered as 0. The above formula is used for computing THF values for every time slice $t_i \in \{t_2, t_3, ..., t_N\}$.

At time slice $t_i = t_1$, we're assuming that all the users are new users (i.e. they are involved in the surge for the first time) and hence,

$$THF(t_1) = \frac{|U^h_{RT}(t_1)|}{|U^h_{all}(t_1)|}$$

Now that we have the values $THF(t_i)$ at every time slice t_i, we aggregate them by taking average.

$$THF_{avg} = \frac{1}{N} \sum_{i=1}^{N} THF(t_i)$$

Algorithm 2 is used for computing THF_{avg} for a given surge S_h. We hypothesize that the value of THF_{avg} is higher for the tweets regarding an event as compared to the tweets regarding random topics.

3.4 Methodology

We collected the tweets for a continuous time period – say from 15^{th} Jan 2018 to 4^{th} Mar 2018 using Twitter API and grouped them by the hashtags. The details of the dataset is given in Table 1. Then we detected the surges. After that, we calculated the THF values for the surges. Then we conducted a survey and labelled the surges to be event or non event. This is considered to be the

Algorithm 2 Computation of Temporal Herding Factor

Divide the surge S_h into equal time slices $\{t_1, t_2, ..., t_N\}$
for each tweet tw in S_h **do**
  $uid = userId$ of tw
  $d = time$ of tw
  if tw is a retweet **then**
    Append uid to the list $U^h_{RT}(t)$ if not already present.
    Where t is the time slice in which d belongs.
  else
    Append uid to the list $U^h_T(t)$ if not already present.
    Where t is the time slice in which d belongs.
  end if
end for
for each time slice t in S_h **do**
  $U^h_{all}(t) = U^h_{RT}(t) \cup U^h_T(t)$
  if $t = t_1$ **then**
    $N_{RT} = \mid U^h_{RT}(t) \mid$
    $N_{all} = \mid U^h_{all}(t) \mid$
  else
    $U^h_{all}(t-1) = U^h_{RT}(t-1) \cup U^h_T(t-1)$
    $N_{RT} = \mid U^h_{RT}(t) - U^h_{all}(t-1) \mid$
    $N_{all} = \mid U^h_{all}(t) - U^h_{all}(t-1) \mid$
  end if
  if $N_{all} = 0$ **then**
    $THF(t) = 0$
  else
    $THF(t) = \frac{N_{RT}}{N_{all}}$
  end if
end for
$THF_{avg} = \frac{1}{N} \sum_{i=1}^{N} THF(t_i)$

ground truth. Then we split the surges into 50:50 in order to be used as training and testing sets. Using the training set, we calculated the threshold of the THF value that can be attributed to some event. We verify the method by testing the threshold on the testing set. Our method is outlined in Figure 1.

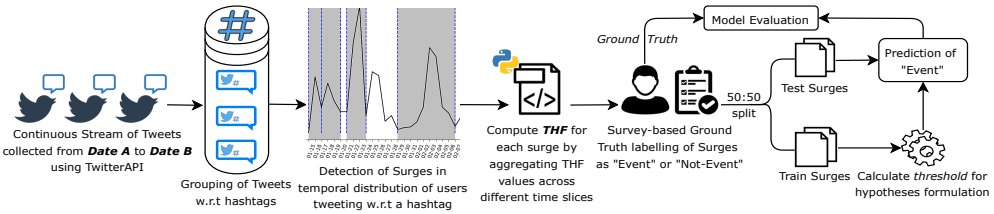


Fig. 1: Outline of our method

4 Evaluation

In this section, we evaluate our event detection model described in the previous section. The 3.3M dataset that is used for evaluation was downloaded using the Twitter API. The details of this dataset are given in Table 1.

Table 1: Details of the 3.3M dataset with Geolocation as near:INDIA within:1500mi

Total number of tweets	$3,360,608$
Start date	15^{th} Jan 2018
End date	4^{th} Mar 2018
Number of days	49
Number of unique users	$431,081$
Number of hashtags	27

We took $28,415$ tweets randomly out of all the tweets scraped for each day. This was done so that the dataset can be uniformly distributed over all the days. The number $28,415$ was chosen because this was the smallest number of tweets that were captured on a single day during this time period. After this sampling, we had $1,363,920$ tweets posted by $280,286$ unique users. We detected 244 surges in our dataset. Out of these 244 surges, we dropped 41 surges since they had significantly less number of tweets (i.e., less than 50 tweets). It is to be noted here that all hashtags in the 3.3M dataset, irrespective of whether they are trending or non-trending, have been considered to find the surges. We computed the THF values for each one of the 203 surges.

Labelling: We conducted a comprehensive survey on the tweets in each of the Candidate surges and labelled them as events or non-events. We randomly picked 50 tweets from each surge. Each of these surges are then annotated by 3 people. The survey was conducted amongst 27 volunteers of age group 17 - 23 years, who were familiar with Twitter. The questions that were asked in the survey form are:

- How many tweets are talking about an event? (this is to find out all the tweets regarding events)
- How many tweets are there in the largest set of tweets that are talking about the same event? (this is to find the largest cluster among the tweets that are talking about an event)

If the answer to the second question is more than 17 (33% of 50 tweets), we label them as events. The inter-annotator agreement measured through the Fleiss' Kappa coefficient is 0.73.

Here, we say the *largest set of tweets* because usually events are distinguishable as there is a clear difference in the number of tweets talking about the same thing. For events, a lot of tweets could be found that talk about it.

We labelled surges as events and non-events. Here, *Events* imply that the corresponding surge has at least one event. The THF_{avg} value was computed for all the surges in each method. We considered manual classification labelling as the ground truth. We then divided our dataset randomly into two equal parts – one for training and the other for testing purposes. We randomly selected 50% of our data for the training set. We calculated the Mean and Standard Deviation of the THF_{avg} values corresponding to the surges in the training set that are labeled as events. We then selected threshold T as discussed further in order to define a range as our hypothesis for predicting whether a surge corresponds to an event or not. The hypothesis is as follows:
Hypothesis: If $Mean_E - (T \times SD_E) < THF_{avg} < Mean_E + (T \times SD_E)$, then there is at least one event in the corresponding surge.
In the above, $Mean_E$ and SD_E represent the mean and standard deviation of the THF_{avg} values respectively, corresponding to the surges in the training set that are labelled as events. T is the number of standard deviations we consider to detect the outliers.
Choosing Best T by Multiple Runs: In the hypothesis given above, choosing the right value of T is a crucial part. In order to select the value of T that gives the most accurate results, we evaluated our method on different values of

$$T \in \{1, 1.25, 1.5, 1.75, 2, 2.25, 2.5\}.$$

Further, we carried out 10 random runs of training-testing on our dataset for each T in the above set. The results are given in Table 2.

Table 2: The results of our methods on the 3.3M dataset.

Method	**THF**
Precision	0.76
Recall	0.89
F1-score	0.82

Figure 2 shows the events with respect to the hashtag *#trndnl*. The shaded portion corresponds to events. The start and end time slices of the events are marked with blue vertical dotted lines while the unshaded regions does not represent events.

4.1 Bot Detection and elimination

When we closely scrutinized the remaining 203 surges, we found that in certain surges, there were less users and disproportionately large number of tweets. This suggested that there could be presence of bots in the dataset. We checked whether the users are bots or not by using the Botometer [7] (formerly known

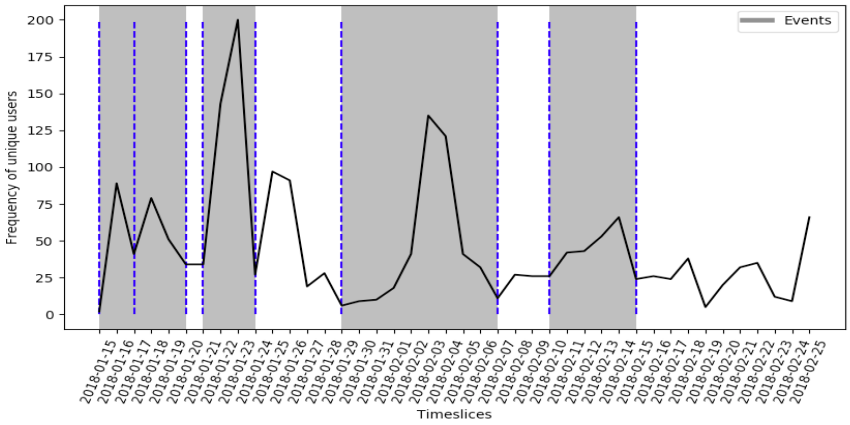


Fig. 2: Shaded region denotes events – related to #trndnl.

as BotOrNot). Botometer has an API that helps us to determine bots. The API returns a Complete Automation Probability (CAP) score for a requested user – this is the probability of the Twitter user being a bot.

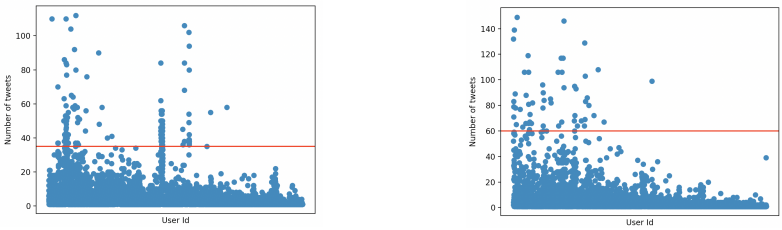
It would be naive to check each user in the dataset against the Botometer. Hence, we obtained the candidate users (who are more probable to be bots) based on a threshold T chosen by visual inspection of the scatter plot of the total number of tweets made by each unique user. We used this plot to determine the T as a value from which the data points become more dispersed visually. If the total number of tweets by the user (across all surges) is more than T, then it is considered as a candidate bot. Each candidate bot is checked with botometer. If its CAP score is high (i.e., CAP score $> 50\%$), then the candidate bot is labelled as a bot. This CAP threshold of 50% is chosen to ensure that no bots with significant automation probability are missed being detected.

Bot Removal: After detecting the bots, an important question to be answered is whether to remove only the tweets by such bots or to eliminate the entire surge that contains the tweets of the bots from our dataset. The former approach has the advantage to prevent much loss of the data. However, one problem with the former approach is that it neglects the influence of the original tweets by the bots. We may remove the original tweets of the bots, but these tweets might have been retweeted by other users. So, we considered both the cases (without bot removal also as one) and named the methods as 1 and 2 as described in the sequel.

Method 1: Without any bot removal In this method, we assumed that the dataset has no bots present in it. Based on this assumption, we directly computed the THF values for each one of the 203 surges.

Method 2: Removal of entire surge Based on the scatter plot (see Fig. 3 (a)), threshold $T = 35$ was set for determining candidate bots. The total number of

unique users whose activity was more than T were 148. Out of them, 7 were detected as bots since they were having CAP scores greater than 50%. Out of 203 surges, the total number of surges with the presence of at least one bot were 22. All these surges were entirely removed from the dataset and the *THF* values were computed for each one of the remaining 181 surges.



(a) 3.3M Dataset (b) Generic Dataset (Section 5)

Fig. 3: Scatter plots of total number of tweets of users to choose threshold for candidate bot determination (scaled to show only users with < 150 tweets)

Finally, Methods 1 and 2 gave 203 and 181 surges, respectively. For each method, we took the set of surges and manually labelled them as *Events* or non-events by using the procedure described in the sequel.

4.2 Comparing with the Event Detection Method Twitinfo

In this section, we report the results of the comparison between our model and the Twitinfo model; both implemented on the 3.3M dataset. We compare our results with Twitinfo model because that is the closest to our approach.
Detecting events using Twitinfo on our Dataset: Using the same values of T and α as Twitinfo uses, would not be an optimal decision since the granularity at which they mostly analyze the tweets is at minute-level whereas we deal with the day-level analysis in the THF_{avg} model. As a result, we test the Twitinfo algorithm on our dataset using different values of T and α.

Table 3: Twitinfo Results on our dataset: Values of T and α giving Best Precision, Recall and F1-scores.

Parameters		Event		
T	α	Precision	Recall	F1-score
1.5	0.01	1.0	0.28	0.44
0.25	0.9	0.29	0.84	0.43
3.0	0.225	0.81	0.53	0.64

Our dataset consists of 27 different hashtags each consisting of multiple surges. For each hashtag h we do the following: We first construct the timeline for h by selecting all the surges related to that hashtag and consider a combined time-sorted collection of all the tweets belonging to these surges; we bin the day-wise tweet frequency in order to generate the required timeline for h; we then run the *peak finding algorithm* of Twitinfo on this timeline for each h in the dataset and for different combinations of T and α generated from the sets $Threshold$ and $Alpha$. Like Twitinfo, we also use $p = 5$.

$$Threshold = \{0.25, 0.5, 0.75, 1, 1.25, 1.5, 1.75, 2, 2.25, 2.5, 2.75, 3\}$$

$$Alpha = \{0.01, 0.025, 0.05, 0.075, 0.09, 0.1, 0.125, 0.15, 0.175, 0.2, 0.225, 0.25, 0.275, 3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9\}$$

For each surge S in our dataset, if the time slice corresponding to the largest value of that surge is present in any one of the windows that the Twitinfo outputs, then S is classified as an Event. Table 3 summarizes the values of T and α that produces the best Precision, Recall, and F1-score values on our dataset for each of the Events. We can see that the best results are obtained for $T = 3.0$ and $\alpha = 0.225$. However, Recall in detecting Events is relatively poor in this case.

Table 4: Comparison of the best results of our methods and the Twitinfo method on our dataset.

	Method 1	**Method 2**	**Twitinfo**
Precision	0.76	0.79	0.81
Recall	0.89	0.83	0.53
F1-score	0.82	0.81	0.64

The results of our method along with Twitinfo method are tabulated in Table 4. We can see from the table that our method outperforms Twitinfo method in recall of detecting Events by a large margin and therefore, also in F1-scores by a large margin. The importance of precision and recall depends on the application. In some cases, precision can be more important while in other cases, recall or even both can be equally important in event detection. For example, in case of disaster events, recall can be more important since no one will want to miss out any disaster event. However, in this paper we consider both the precision as well as recall to be equally important. Additionally, Table 4 shows that *THF* method (for both the Methods) reports the best F1-score.

5 Generality of the Hypothesis

We feel that our hypothesis may be generic in nature since it clearly detects *Events* even though the dataset is not scraped with respect to any specific event.

To test the generality of our hypothesis, we verify the hypothesis on a generic dataset. This dataset is not restricted to any particular region and is from a different time period. We downloaded the Twitter firehose dataset that was used in [17]. This dataset is also listed in the *ICWSM* website[2] and is publicly available. The details of this dataset are given in Table 5.

Table 5: Details of public dataset with Geotagging as None

Total number of tweets	$1,280,000$
Number of days	29
Start date	14^{th} *Dec* 2011
End date	11^{th} *Jan* 2012

Since there are no geotagging restrictions, this dataset is not region-specific. There are 77 hashtags and 547 surges in the dataset. Out of them, majority of the tweets in 423 surges are non-English. Most of the events in this dataset are from the Middle Eastern region. This could be a reason for such a large number of non-English tweets. There were 14 surges that were too small – having less than 50 tweets. Also, since manual labelling of non-English tweets was not possible, we discarded all the surges that had majority of non-English tweets. Hence, we were left with 110 surges.

Now, for bot detection and removal, we plotted a scatter plot of distribution of the total number of tweets by different users as described in Section 4 (See Figure 3 (b)). By visual inference, we selected a threshold $T = 60$ for detecting candidate bots. 89 users having their total tweet count greater than 60 were labelled as candidate bots. We followed the method described previously on the Generic Dataset also. Botometer tagged 4 users as bot and they were present across 47 surges.

To test the hypothesis, we manually labeled the surges that represent events as described in previous sections. We then tested the same hypothesis that we formulated from our dataset, on these surges. The results of the methods on Generic Dataset are summarized in the Table 6 for $T = 1$.

6 Discussion

The novelty in this work is that we are considering the behavioral traits to identify the presence of events. The advantage of our method is that it is scalable as it

[2] https://www.icwsm.org/2018/datasets/

Table 6: Test results on the Generic dataset

Method	T	Precision	Recall	F1-score
Method 1	1	0.70	0.97	0.81
Method 2	1	0.70	0.97	0.81

uses only the metadata to detect the presence of events. This is evident when we compare the time required for the analysis. Our method used only 48.83 seconds whereas Twitinfo method used 678.24 seconds for finding the events on the same dataset. However, both our work and Twitinfo use the same granularity as one day, whereas Twitinfo was originally introduced with analyzing at a granularity of minutes. Also, our method will detect only whether there is at least one event corresponding to the surge or not. In the case of multiple events occurring at the same time period of the surge with the same hashtag, our method will not be able to detect them as separate events.

If we want to detect the specific events separately, we may use clustering. However, the clustering needs to be done on only a very small dataset – the tweets corresponding to the surge under consideration – compared to doing clustering in the whole dataset. This makes it more scalable compared to the event detection methods using clustering on the entire dataset.

7 Conclusion

In this paper, we proposed a method to detect events from Twitter data, based on our hypothesis that herding occurs in surges during events. We formulated a parameter called *Temporal Herding Factor (THF)* to detect traces of herding from the metadata of the tweets – for this we used a dataset with 3.3M tweets and found a boundary *THF* value to classify surges as an event or not. The results that we got support our hypothesis that indeed herding in surges can be seen as an indicator for detecting events. Results obtained from our method show that it performs better than the state of the art method Twitinfo.

Moreover, we tested our method on an openly available dataset (Generic dataset). We used the same boundary values that we calculated from 3.3M dataset and showed that our algorithm works with F1-score of 0.81 even with the Generic dataset. This shows that our approach can be generalized. It is to be noted here that the 3.3M dataset and the Generic dataset are from different time periods. Upon acceptance of the paper, we will make the dataset and code openly available for reproducibility.

In future, we would like to formulate more parameters that represent herding to verify our hypothesis. Also, we would like to check whether the boundary values need fine-tuning for different geographical regions. Through this way we intend to find the most optimal boundary values. Also, we would like to verify our hypothesis using other methods to detect or measure herding.

References

1. Adedoyin-Olowe, M., Gaber, M.M., Dancausa, C.M., Stahl, F., Gomes, J.B.: A rule dynamics approach to event detection in Twitter with its application to sports and politics. Expert Systems with Applications **55**, 351–360 (2016)
2. Alvanaki, F., Sebastian, M., Ramamritham, K., Weikum, G.: Enblogue: emergent topic detection in web 2.0 streams. In: Proceedings of the 2011 ACM SIGMOD International Conference on Management of data. pp. 1271–1274. ACM (2011)
3. Asch, S.E.: Studies of independence and conformity: I. a minority of one against a unanimous majority. Psychological monographs: General and applied **70**(9), 1 (1956)
4. Banerjee, A.V.: A simple model of herd behavior. The quarterly journal of economics **107**(3), 797–817 (1992)
5. Cai, H., Yang, Y., Li, X., Huang, Z.: What are popular: exploring Twitter features for event detection, tracking and visualization. In: Proceedings of the 23rd ACM international conference on Multimedia. pp. 89–98. ACM (2015)
6. Cote, J., Sanders, D.: Herding behavior: Explanations and implications. Behavioral Research in Accounting **9** (1997)
7. Davis, C.A., Varol, O., Ferrara, E., Flammini, A., Menczer, F.: Botornot: A system to evaluate social bots. In: Proceedings of the 25th international conference companion on world wide web. pp. 273–274 (2016)
8. Fang, Y., Zhang, H., Ye, Y., Li, X.: Detecting hot topics from Twitter: A multiview approach. Journal of Information Science **40**(5), 578–593 (2014)
9. Gaglio, S., Re, G.L., Morana, M.: Real-time detection of Twitter social events from the user's perspective. In: Communications (ICC), 2015 IEEE International Conference on. pp. 1207–1212. IEEE (2015)
10. Gaglio, S., Re, G.L., Morana, M.: A framework for real-time Twitter data analysis. Computer Communications **73**, 236–242 (2016)
11. Hasan, M., Orgun, M.A., Schwitter, R.: TwitterNews+: a framework for real time event detection from the Twitter data stream. In: International Conference on Social Informatics. pp. 224–239. Springer (2016)
12. Hasan, M., Orgun, M.A., Schwitter, R.: A survey on real-time event detection from the Twitter data stream. Journal of Information Science (2017)
13. Li, C., Sun, A., Datta, A.: Twevent: segment-based event detection from tweets. In: Proceedings of the 21st ACM international conference on Information and knowledge management. pp. 155–164. ACM (2012)
14. Li, H., Sakamoto, Y.: Re-tweet count matters: social influences on sharing of disaster-related tweets. Journal of Homeland Security and Emergency Management **12**(3), 737–761 (2015)
15. Marcus, A., Bernstein, M.S., Badar, O., Karger, D.R., Madden, S., Miller, R.C.: Twitinfo: aggregating and visualizing microblogs for event exploration. In: Proceedings of the SIGCHI conference on Human factors in computing systems. pp. 227–236. ACM (2011)
16. Mathioudakis, M., Koudas, N.: Twittermonitor: trend detection over the Twitter stream. In: Proceedings of the 2010 ACM SIGMOD International Conference on Management of data. pp. 1155–1158. ACM (2010)
17. Morstatter, F., Pfeffer, J., Liu, H., Carley, K.M.: Is the sample good enough? comparing data from Twitter's streaming API with Twitter's firehose. In: ICWSM (2013)

18. Ozdikis, O., Senkul, P., Oguztuzun, H.: Semantic expansion of tweet contents for enhanced event detection in Twitter. In: Proceedings of the 2012 International Conference on Advances in Social Networks Analysis and Mining (ASONAM 2012). pp. 20–24. IEEE Computer Society (2012)
19. Petkos, G., Papadopoulos, S., Aiello, L., Skraba, R., Kompatsiaris, Y.: A soft frequent pattern mining approach for textual topic detection. In: Proceedings of the 4th International Conference on Web Intelligence, Mining and Semantics (WIMS14). p. 25. ACM (2014)
20. Raafat, R.M., Chater, N., Frith, C.: Herding in humans. Trends in cognitive sciences **13**(10), 420–428 (2009)
21. Shepard, D.: Nonparametric bayes pachinko allocation for super-event detection in Twitter. In: TENCON 2014-2014 IEEE Region 10 Conference. pp. 1–5. IEEE (2014)
22. Sivaraman, N.K., Muthiah, S.B., Agarwal, P., Todwal, L.: Social synchrony in online social networks and its application in event detection from twitter data. In: Proceedings of the The 2020 IEEE/WIC/ACM International Joint Conference on Web Intelligence and Intelligent Agent Technology (WI-IAT'20) (2020)
23. Thapen, N., Simmie, D., Hankin, C.: The early bird catches the term: combining Twitter and news data for event detection and situational awareness. Journal of biomedical semantics **7**(1), 61 (2016)
24. Unankard, S., Li, X., Sharaf, M.A.: Emerging event detection in social networks with location sensitivity. World Wide Web **18**(5), 1393–1417 (2015)