Improving Crisis Event Detection Rate in Online Social Networks Twitter Stream using Apache Spark

A Bhuveshwari1,*, R Jayanthi2, A Lakshmi Meena3
1,2Assistant Professor, Vellore Institute of Technology, Chennai, India
3Application Development Analyst, Accenture, Chennai, India

*Email: bhuvana.cse14@gmail.com

Abstract. Microblogging is a vital emerging practice among the online users for disseminating information about onset events to establish and enable the fastest communication. In specific, event detection in Twitter streams is a critical task due to continuous scrutiny of stream of events which tremendously demands the instantaneous discovery of anomalies happen on an extensive period of time and spatial information. Event detection and tracking during crisis in huge Twitter stream undergoes critical trials, such as the dynamic feature patterns, imbalanced samples on keywords, hashtags, temporal and spatial geographic heterogeneity in location-of-interest. In this research article, onset event detection framework using topic modelling is proposed in streaming window that addresses dynamic feature patterns challenges. The experiments are carried out in Apache Spark Latent Dirichlet Allocation technique i.e., S-LDA algorithm in Spark environment, which process non-probabilistic streaming dynamic featured corpus. The onset event topics are validated with high accuracy. The proposed system takes dispensation time delay for processing the initial data stream containing tweets is after 192 ms. The system averagely confirms the event detection after 75-100 milliseconds in comparing with its real timestamps.

1. Introduction
In recent era of Internet, the improved concentration towards the direction of real-world disaster event identification is carried out using publicly available open data via Twitter, Facebook and YouTube etc. It supports multimedia services like Flickr, Instagram which include images, audio, video sharing facilities for the user which are semi-structured data sources. The challenge is to quantify the in-depth traces of information sources about the events from location-of-interest. There occurs extensively recognized events including civilian’s protests, local gatherings, or can be wider-reaching globe in significance. It is promising to harvest hidden information by overwhelming the critical challenges involved in Twitter and acts as a convention of online social media. The events are categorized based on locality and entity related in a high stream of twitter handles. Certain events are planned whereas few events are unplanned which trigger as breaking news in Online Social Networks (OSN). Particularly, an event is identified as real-world happening based on temporal and geographical spatial co-ordinates due to dynamics in volume, velocity, veracity, noise which acts as a critical challenge [1]. Alternatively, useful information from characteristic Twitter posts are very minimal and repeatedly holds large amounts of shortened, casual and unbalanced words like hashtags, linguistic mistakes and irregular sentence structures. In this research article, topic modelling on event onset incorporates the textual data inside a cluster of documents corpus to distinguish the intact significance of disaster
events. The tweets from news telecasters stray from normal blog substance so that it is generally long, elegantly composed and rich on semantics.

The pioneering investigation on Topic Detection and Tracking (TDT) [2] method identified the broadcasted news which trigger an alert whenever a new-fangled event occurred. However, the topic modelling generated supporting result of modelling the corpus that looks as if the tweets allowing switching between topics and lexicons, and dependencies among documents and trending topics. Meanwhile, LDA can be applied to determine unseen topics and topic consignment to documents using occurrence of corpus which is a probabilistic method [3]. In common, the topic modeling technique is computationally overpriced which requires equally outsized amount of memory and significant quantity of computational cost. The part of event recognition has in the past been tended to in the ground of information retrieval, yet the exertion has principally been on identifying occasions starting from regular media sources, for example, News encourages, Tweets, Posts and so forth. Most of the onset event detection systems use nearest neighbour clustering algorithm, fuzzy-based classification, and topic modelling technique which extremely produces assured useful effective systems. The acute challenge while handling big data is computational cost and complexity on implementing in high performance computing systems in real-time. Accordingly, some ground-breaking research exertions have been attempted to improve performance.

The enhancements contain expanding versatility of execution, enhancing execution, in parallel and conveyed foundations, and dealing with powerfully fluctuating huge volume of datasets [4] preparation. The performance of topic models is often necessary; during emergency situations to achieve the objectives can be proficient using Apache Spark which is used in our proposed work. The big data analytics for trend detection was monitored [5]. As the informational dataset collection, the continuous web-based social networking streams are considered, it has volume and speed and a few imperatives are to be laid on the framework arrangements, for example, RAM to deal with the application viably. Events ought to be recognized continuously at the earliest opportunity, particularly when the approach is planned to be utilized as a part of basic applications like crisis reaction. In such situations, the event detection technique must be evaluated using corpus based performance but also in terms of how quickly the system is able to detect certain events. In this, there is no utilization in recognizing or distinguishing the event that had happened in the past about which individuals are talking currently. Additionally, while identifying the events, for example, disasters, it must be done as quickly as time permits individuals mindful of it even before they are influenced by it. This should be possible utilizing the parallel handling of the quickly developing informational collection. In the proposed frameworks, people's social connections during emergencies are extracted concerning certain vital features namely time, location, hashtag, users, text, and images.

The rest of the paper is organized as follows. Various related work is discussed in Section II. In Section III, the overview of event burst detection approach is deliberated using a mathematical model. The experimental results are discussed in Section IV and Section V concludes the paper with future directions.

2. Related Work
The goal of identifying events is to monitor real-time OSN streams in order to extract the relevant topics and meaningful information concerning a specific location. In turn, the application domains acquire the account of detecting instantaneous outbreaks namely earthquake, floods, bomb blast and so on. Information concerning burst topics from certain locations during crisis event can be used to alarm emergency management. Using named entity recognition detections are handled [6] in social networks. The geographically located users of OSN are considered as a trustworthy source for detecting disaster event and investigating responses after mass emergency events. The insight of an event changes depending on the way in which the burst topic is identified in streaming document collection [7]. The event are identified and detected in Twitter using selective categorical data based and hypermedia based contents. In twitter, users post referees as ‘tweets’ which is considered to be more vital and the tweets are shared by their followers.
Various research work used burst keywords and tweeting dynamic activities on Twitter to understand the structural properties of information flow during the earthquake in 2011 [8], [9]. An anomaly based clustering technique was used to find relevant corpus for detecting disasters which was investigated thoroughly as an early warning system [10]. Furthermore, researchers developed a framework to bind OSN data with events in two-fold steps [11], [12]. The research work used deep learning modes to detect and classify events [13]. In the first step, the noisy data is removed by building a visual filter on the resulting event keywords. In the second step, text, time, location is crawled through a query on the event happening. The Latent Dirichlet allocation (LDA) - based approaches evade data sparsity by foreseeing the events constructed on latent topics with lesser dimension [14]. It estimates the tweet’s topic mixture using regular probabilistic erudite model.

The OSN users desire to distinguish which events are happening in an unpredictable location, events which are trending regularly, marking of events in a specific range of geographical bounds, events which can occur in future, etc. The Event location handle requests certain valuable Twitter parameters. For illustration, area co-ordinates, put check-ins, geo labels are utilized as spatial parameters. The location based event detection depends on the geotagged data [15] by estimating the location entropy parameter. Be that as it may, the rate of geotagged information is amazingly little compared to non-geotagged information but it could be a well-known metric for measuring the ubiquity of event areas (e.g., points-of-interest). The entropy method to sense events is to recognize named entities and later proceed with effective clustering technique to perceive and break events into discrete topics. The events were further identified by real-time entity [17] and indexing of tweets [18]. The ascent of Social Media stages lately raised tremendous data streams which require new ways to deal with examine the individual information.

Most prevailing approaches extravagance semantics but expressions in tweets may dynamically change, interpreting sensitive topics [19], [20]. The linguistic knowledge used in micro-blogs is extremely easy-going, imprecise, and dynamic. The high-level features are designed by combining tweets to topics and by defining the sentiments of a tweet. The features are divided into two sets (i.e., shared features and tweet features). Shared Features consists of the number of friends, followers, number of times the user is mentioned, user language, favorites, status, verified user. Tweet Features include the number of mentions, Date, Time, mentions, Geographic location of event, location of user, URLs, trending words, length of the message, novelty, message, replies, retweets and the word length in tweet. In this research article, a topic modelling framework is proposed that targets to address the real-time challenges involved. In this model, we use the Latent Dirichlet Allocation i.e., Spark - LDA algorithm which takes the processed data, that is the keywords as input and perform topic modeling using which the events are detected.

3. Proposed Framework

The proposed work (Figure 1) deals with the identification of events using topic modelling and classified, specifically detect events, in a streaming Twitter data. It is a non-trivial problem, due to the scalability, heterogeneity and the diversified quality of the data as well as the presence of irrelevant information.
In order to address the challenges, a dynamic feature constrained onset event detection framework in streaming window of data which contains data collection, pre-processing, energetic highlight extraction and theme displaying. In LDA, each record may be seen as a blend of different themes where each archive is considered to have a set of points that are allotted to it through LDA as appeared in Figure 2. Usually indistinguishable to probabilistic latent semantic analysis (pLSA), but that in LDA the subject dispersion is expected to have a scanty Dirichlet earlier. The inadequate Dirichlet priors encode the intuition that archives cover as it were a little set of points which topics use as it were a little set of words as often as possible. In hone, this comes about in distant better; an improved disambiguation of words and a more exact task of archives to subjects. LDA may be a generalization of the pLSA show, which is proportionate to LDA beneath a uniform Dirichlet earlier conveyance. With plate documentation, the conditions among the numerous factors can be captured concisely. The boxes are "plates" speaking to imitates. The external plate speaks to reports, whereas the internal plate speaks to the rehashed choice of themes and words inside a record.

Therefore, the eventual aim is to define the unidentified, unseen parameters $\theta$, $\phi$, and $Z$. The joint distribution $P(\theta, \phi, Z, W|\alpha, \beta)$ is given as follows.

$$P(\theta, \phi, Z, W|\alpha, \beta) = \prod_{i=1}^{K} P(\phi_i|\beta) \prod_{j=1}^{M} P(\theta_j|\alpha)$$  \hspace{1cm} (1)
The generative process of LDA is as follows.

i. For all the documents, the total number of Multinomial parameter vectors are sampled from dirichlet parameters $\alpha$.

ii. For all topics $\phi$, the remaining Multinomial parameter vectors are sampled form dirichlet parameters $\beta$.

iii. Given topic consignment for $n, z_{ni}$ is tested from the Multinomial dissemination with parameter $\theta_i$ equivalent to each document $i$.

iv. The corpus with index $n$ is tested from the Multinomial distribution with parameter resolute from the topic which is sampled in the last step $\phi_{zi,j}$.

Moreover, the definition of the expected value of a Dirichlet distribution, the estimate for $\theta_{ik}$ is as follows.

$$\theta_{ik} = \frac{n_{ik} + \alpha_k}{\sum_{k=1}^{K} n_{ik} + \alpha_k}$$

where $n_{ik}$ is the number of words in record $i$ that have been doled out to subject $k$. And by the precise same contention, the appraise for $\phi_{kw}$ (the extent of word $w$ in theme $k$) as takes after is given as follows.

$$\phi_{kw} = \frac{n_{kw} + \beta_w}{\sum_{w=1}^{W} n_{kw} + \beta_w}$$

where $n_{kw}$ is the number of times word $w$ is doled out to subject $k$ (over all archives document within the corpus). Learning the different dispersions (the set of topics, their associated word probabilities, the subject of each word, and the particular subject blend of each report) could be an issue of Bayesian inference.

$$\int p(z_d, \theta_d | \alpha, \beta) = \int \sum_{d} \theta_{dz_i} \frac{1}{B(\alpha)} \sum_{k} \theta^{n_{id} + \alpha_k} \theta_d = \frac{B(n_d + \alpha)}{B(\alpha)}$$

where $B(\alpha)$ is a normalizing constant. In order to sample from the posterior, it suffices to compute the posterior up to a constant for a single assignment conditioned on all other assignment, Learning the different dispersions (the set of themes, their related word probabilities, the subject of each word, and the specific theme mixture of each report) could be an issue of Bayesian deduction. In our work, we utilized a variety Bayes estimation of the back dispersion.

4. Experiments and Results

4.1. Pre-processing

The tweets are scraped from twitter and random tweets from real-time dataset from Twitter [21] and loaded for Apache Spark. The dataset is loaded from stored folder. The training data is stored as text file and each tweet is accompanied by the magnitude of its sentiment 0 to 1. The tweets are observed that a threshold of 0.5 is good enough to classify a tweet according to its sentiment. Tweets with lesser threshold were not definitive to be trained as per their mentioned classification. Using Twitter API, the
tweets are crawled during covid-19 lockdown as 40,000 tweets are collected, out of which 35,000 tweets are considered to be valid tweets suitable for training. In data pre-processing, cleaning the tweet data removing punctuations, removing name tags, convert everything into lower case. The input data is pre-trained using tokenization which token in a sentence, and a sentences is a token in a paragraph. An inserting framework may be a straight mapping from the initial space (one-of-k) to a real-valued space where substances can have important connections utilizing pack of words. Part the information into test and prepare information, rearrange information. Load the pre-trained vectors for the Word2Vec model. Using a Keyed Vectors file, we can get the embedding of any word by calling word_vec(word) to obtain all the words in the model's vocabulary. Using a Tokenizer to assign indices and filtering out unfrequented words. Tokenizer creates a map of every unique word and an assigned index to it. The parameter called num_words indicates that we only care about the top 20000 most frequent words. Lexicon padding processing sequences all to the word cloves with same length of 0 to 140 words.

4.2. Training and Testing
In this stage, various basic factors involved in configuring Spark with the goal of efficient utilization of computing cluster are considered. In this arrange, different fundamental variables included in arranging Spark with the objective of productive utilization of computing cluster are considered. The preparing organize starts by instantiating Spark LDA show with the parameters decided previously for accomplishing great parallel execution execution. A few of the factors which acts as an input to the demonstrate are number of subjects to be decided, number of agent centers, number of allotments, memory distributed to the agent, number of emphases to get valuable points, and others. In the next level, the driver program is run on Apache Spark mode which starts the proposed model using RDD and pre-processes it. As a result, a data flow graphical frame is generated by applying RDD transformations. Furthermore, within the cluster computing, the sequencer accomplishes load balancing by significant the partitions possession of data to worker nodes. After the Spark-LDA model completes the task, the total execution time, delay time, detection rates are computed. As a final state of training, the S-LDA task is passed to the inspection and event exploration stage. During this stage, the Spark and RDMA-Spark frameworks are tested and validated for scalability to decide the best defaulting standards for each of the response variables.

4.3. Event Detection Rate
The performance of LDA algorithm with comparison to respective similar algorithms is analysed and graphs are constructed as follows: Significance testing utilizing the earlier over as the testing dispersion does not include Gibbs testing. The fetched of straightforward significance inspecting employing dissemination over z is harder to precise, and will be execution subordinate. However, unreasonably to these strategies, it is expected that the taken a toll of creating samples is directly comparable to Gibbs Sampling. The overall detection cost is inspected more thoroughly using S-LDA method to produce good results. The results constructed on the perplexity values obtained as 0.32. The perplexity values for the proposed LDA algorithm are found to be smaller for small data sets and increase gradually with increase in the size of the dataset.
Figure 3. S-LDA Scheduling Delay

The graphs as shown in the Figure 3 are plotted for the scheduling delay it takes in Apache Spark environment. The throughput and latency analysis of the S-LDA algorithm abruptly vary based upon the number of processors as input. As the number of processors increases, the throughput increases. The comparison between the accuracy of the prediction of the Latent Dirichlet Allocation algorithm we have used with and without including the word classifiers are observed from the graph that is observed to be 80% on average.

Figure 4. S-LDA Time Latency

It clearly shows that implementation with word classifiers gives high accuracy compared to that done without classifiers. The processing delay time needed to start monitoring and confirming event detection is evaluated from the data stream simulation of the benchmarking 'CrisisLexT6' dataset. The size of the incoming tweet is set as 750 tweets per minute. Figure 4 indicates the time latency and delay which is reasonably acceptable in distributed file systems environment. For the three sampling interval are considered as 2, 4 and 6 minutes respectively. The event detection accuracy using S-LDA is shown in Figure 5. The data stream is sent as input to the Apache Spark framework by fixing the location threshold as 300 pts and the hashtag threshold 700 numbers. The proposed S-LDA framework detected the event in 75 seconds which is more efficient with an average time delay on single node.
The proposed Spark based topic modelling shows better detection rate when compared to state of art method [15] which takes 200 seconds for confirming an event using MapReduce algorithm. The proposed S-LDA event confirmation time is recorded and noted in Table 1. It shows that the time taken to confirm an event between the start of event detection and the time at which the event detection is confirmed. The proposed S-LDA framework detected the event in 75 seconds which is more efficient when compared to state-of-art which used 200 seconds for confirming an event as shown in Table 1. In the proposed framework, the average of processing time delay for processing the initial data stream containing tweets is after 192 seconds (Table 1).

**Table 1. Evaluation of Event Detection Rate**

| Time Interval | Proposed S-LDA Framework | State-of-Art Method [15] |
|---------------|--------------------------|--------------------------|
|               | Processing Start Time    | Event Confirmation Time  |
|               |                          | (ms)                     | Event Confirmation Time (ms) |
| 2 minutes     | 121.24                   | 192.4                    | 324.7                      |
| 4 minutes     | 240.8                    | 283                      | 456.1                      |
| 6 minutes     | 361.4                    | 401.4                    | 566.1                      |

![Figure 5. S-LDA Event Detection Accuracy](image)

![Figure 6. S-LDA Event Detection Rate](image)
The results show that the proposed S-LDA framework takes a time delay for extracting the events from the initial time interval to confirm the event detection. The system averagely confirms the event detection after 75-100 milliseconds in comparing with its real timestamps as shown in Figure 6. The proposed system takes dispensation time delay for processing the initial data stream containing tweets is after 192 ms.

5. Conclusion
The emergence of burst topics leads to the occurrence of various categories of event in real-time. The process of identifying events and their associated documents on OSN sites is to monitor real-time social media streams and extract information. Event detection is the area of research which is used for significant application areas which exploits OSN. The proposed framework is tested in Apache Spark environment to handle high speed twitter streams. The framework identifies various ongoing event in diverse temporal and spatial location of interest with minimal time latency. The proposed topic modelling event detection framework obtains 96% accuracy which outperforms the existing statistical state-of-the-art methods. The proposed semantic based sensitive access rule LDA framework precisely detects sensitive topics semantically using statistical topic scheme which incorporates standard knowledge bases for tagging the sensitive topics. The maximum accuracy of 95% is obtained with 88% precision and 86% recall with optimal Spark RDD cluster computation period. The proposed system takes dispensation time delay for processing the initial data stream containing tweets is after 192 ms. The results averagely confirm the event detection after 75-100 milliseconds in comparing with its real timestamps. In future work, the framework can be assist amplified to distinguish up and coming characteristic natural disasters or anomaly dangers as early as conceivable to caution public through well-known news broadcasting sources.

References
[1] Petrovic, S. 2013, “Real-time event detection in massive streams”.
[2] Allan, J., 2002. Introduction to topic detection and tracking. In Topic detection and tracking (pp. 1-16). Springer, Boston, MA.
[3] Goorha, S., Ungar, L. 2010, “Discovery of significant emerging trends’, Proceedings of ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD), USA, pp.57-59.
[4] Shi, L.L., Liu, L., Wu, Y., Jiang, L., Hardy, J. 2017, “Event Detection and User Interest Discovering in Social Media Data Streams”, IEEE Access, Vol. 5, pp.20953-20964.
[5] Mathioudakis, M., Koudas, N. 2010, “TwitterMonitor: trend detection over the twitter stream”, Proceedings of the 2010 ACM International Conference on Management of Data – SIGMOD, Indianapolis, Indiana, pp. 1155-1158.
[6] Ritter, A., Clark, S., Mausam, Etzioni, O. 2011, “Named entity recognition in tweets: an experimental study”, Proceedings of the Conference on Empirical Methods in Natural Language Processing, Edinburgh, United Kingdom, pp. 1524-1534.
[7] Yao, L., Mimno, D., McCallum, A. 2009, “Efficient methods for topic model inference on streaming document collections”, Proceedings of the 15th ACM International Conference on Knowledge Discovery and Data Mining, 2009, vol. 4, p. 937 (2009).
[8] Kleinberg, J. 2003, “Bursty and hierarchical structure in streams”, Data Mining and Knowledge Discovery, Vol. 7, No.4, pp.373-397.
[9] Kogan, M., Palen, L., Anderson, K. M. 2015, “Think Local, Retweet Global: Retweeting by the Geographically-Vulnerable during Hurricane Sandy”, Proceedings of ACM conference CSCW ’15, pp. 981–993.
[10] Doulamis, N.D., Doulamis, A.D., Kokkinos, P., Varvarigos, E.M. 2016, “Event detection in twitter microblogging”, IEEE Transactions on Cybernetics, Vol. 46, No. 12, pp.2810-2824.
[11] Fung, G.G.P.C., Yu, J.X.J., Yu, P.P.S., Lu, H. 2005, “Parameter free bursty events detection in text streams”, Proceedings of the 31st International Conference on Very Large Data Bases - VLDB 2005, Vol. 1, pp. 181–192.
[12] Sakaki, T., Okazaki, M., Matsuo, Y. 2010, “Earthquake shakes Twitter users: real-time event detection by social sensors”, Proceedings of the 19th ACM International Conference on World Wide Web, Raleigh, USA, pp. 851–860.
[13] Bhuwaneshwari, A., J. Timothy Jones Thomas, and P. Kesavan, “Embedded Bi-directional GRU and LSTM Learning Models to Predict Disaster Twitter Data,” Procedia Computer Science, Elsevier 165 (2019): 511-516.
[14] Becker, H., Iter, D., Naaman, M., Gravano, L. 2012, “Identifying content for planned events across social media sites”, Proceedings of 5th ACM International Conference on Web Search and Data Mining, 2012, pp. 533-542.
[15] Nguyen, D.T. and Jung, J.J., 2015. Real-time event detection on social data stream. Mobile Networks and Applications, 20(4), pp.475-486.
[16] Bhuwaneshwari, A., and C. Valliyanmai. “Information entropy based event detection during disaster in cyber-social
networks.” Journal of Intelligent & Fuzzy Systems, IOS Press 36, no. 5 (2019).

[17] Jang, B., Yoon, J. 2018, “Characteristics Analysis of Data from News and Social Network Services”, IEEE Access, Vol.6, pp.18061-18073.

[18] Ardon, S, Bagchi, A, Mahanti, A, Ruhela, A, Seth, A, Tripathy, R. M & Triukose, S. 2013, October, “Spatio-temporal and events based analysis of topic popularity in twitter”, Proceedings of the 22nd ACM International Conference on Information & Knowledge Management, pp. 219-228.

[19] Bhuvaneswari, A & Valliyammai, C. ‘Semantic-based sensitive topic dissemination control mechanism for safe social networking’, Rajsingh, E.B, et al. (eds.), Advances in Big Data and Cloud Computing, Advances in Intelligent Systems and Computing, Volume 645, Chapter No. 17, pp. 197-207. Springer Nature, Singapore, 2018.

[20] Valliyammai. C., Bhuvaneswari A., "Semantics-based sensitive topic diffusion detection framework towards privacy aware Online Social Networks", Cluster Computing, Springer (2018) pp. 1-16.

[21] Link: http://chorusanalytics.co.uk/ [Accessed on January 2021].