Earth Science Informatics (2021) 14:913–929
https://doi.org/10.1007/s12145-021-00601-w

RESEARCH ARTICLE

A comprehensive social media data processing and analytics architecture by using big data platforms: a case study of twitter flood-risk messages

Michal Podhoranyi [1]

Received: 18 December 2020 / Accepted: 1 March 2021 / Published online: 11 March 2021

Abstract
The main objective of the article is to propose an advanced architecture and workflow based on Apache Hadoop and Apache Spark big data platforms. The primary purpose of the presented architecture is collecting, storing, processing, and analysing intensive data from social media streams. This paper presents how the proposed architecture and data workflow can be applied to analyse Tweets with a specific flood topic. The secondary objective, trying to describe the flood alert situation by using only Tweet messages and exploring the informative potential of such data is demonstrated as well. The predictive machine learning approach based on Bayes Theorem was utilized to classify flood and no flood messages. For this study, approximately 100,000 Twitter messages were processed and analysed. Messages were related to the flooding domain and collected over a period of 5 days (14 May – 18 May 2018). Spark application was developed to run data processing commands automatically and to generate the appropriate output data. Results confirmed the advantages of many well-known features of Spark and Hadoop in social media data processing. It was noted that such technologies are prepared to deal with social media data streams, but there are still challenges that one has to take into account. Based on the flood tweet analysis, it was observed that Twitter messages with some considerations are informative enough to be used to estimate general flood alert situations in particular regions. Text analysis techniques proved that Twitter messages contain valuable flood-spatial information.

Keywords Hadoop · Spark · Data extraction · Social network · Floods

Introduction

Nowadays, the use of social media data is growing significantly, thanks to open data policy and technological advances providing data relatively fast and easy. Scientific societies have already applied such data in economic, cultural, political or environmental domains. However, the most common use case that was utilized in many works is in emergency management. In this case, social media are beneficial because, in most situations, the first information about an emergency usually appears on social networks (Kim and Hastak 2018). Traditional sources such as newspapers, radio or TV, which are responsible for spreading information, are almost always delayed compared to social media and provide only one-way communication (Schneider and Check 2010).

Online social media channels, such as Twitter, enable people not only use the medium for interaction with other people but also to be a part of the emergency management process by sharing and discussing important events (Yaqub et al. 2017). This data-rich medium produces and disseminates information with a different level of precision and truthfulness almost every second, and therefore one has to be careful which data will be engaged in the analysing process. These intensive data are generated constantly and can be characterized as big data because they possess all big data key parameters such as volume, velocity, and variety (Hill et al. 2014; Pradeep and Sundar 2020). Therefore, integrating Twitter data in emergency processes introduces not only technical but also data analysing and processing challenges. Twitter data processing requires adequate storage capacity, database with fast data access and easy data handling. All mentioned features are provided by big data platforms that are able to process data in a distributed manner.

Communicated by: H. Babaie

✉ Michal Podhoranyi
michal.podhoranyi@vsb.cz

[1] IT4Innovations – VSB Technical University, 17.listopadu 15, 70833 Ostrava, Czech Republic

This paper presents both technical as well as data analysing challenges. The first technical part introduces the big data architecture, while the second part presents data analysing techniques and outcomes. As a use case, global Twitter data were chosen as an initial dataset. Specifically, for this study, floods were selected as an appropriate event and data domain.

The structure of the article is organized as follows. Section *Objectives and Contributions (in Introduction)* presents the main goals and general contributions of the paper. Section *Related work* is dedicated to reviewing the current state of literature. Section *Methodology* provides the methodological approaches used to build the proposed architecture. It contains details about data application and Tweet processing as well. Section *Data Analysis* summarizes the results based on content, text and localization extraction analysis. Last sections, *Discussion, Future work, and Conclusion* are devoted to resume conclusions and future challenges.

Objectives and contributions

The main objective of the article is to propose an advanced architecture and workflow based on Apache Hadoop and Apache Spark big data platforms. The primary purpose of the presented architecture is collecting, storing, processing, and analysing intensive data from social media streams (Fig. 1). This paper presents how the proposed architecture and data workflow can be applied to analyse Tweets with the predefined flood topic. The secondary objective, trying to describe the flood situation by using only Tweet messages and exploring the informative potential of such data is demonstrated as well.

(Martinez-Rojas et al. 2018) define the main areas that require additional research in Twitter-based emergency management: interoperability, diversity, credibility, visualization, and regulatory initiatives. *Interoperability* is regarding mechanisms that allow automatically extracting and processing relevant information as well as identifying false information. Automatic detection of tweets according to the user who posts the message and what is the object is hidden under *the diversity* term. *Credibility* deals with the control of false information, whereas *regulatory initiatives* take the creation of procedures, policies, and regulations as an important object to deal with. Finally, *visualization* presents that an appropriate visualization method can improve a decision-making process.

The main contribution of the article is associated with interoperability and partially with diversity. Specifically, the first and major contribution is in the proposed big data architecture based on the combination of Apache Spark and Apache Hadoop platforms. The system provides an automatic set of steps aimed to extract the demanded information by using social media streams in near real-time. This article presents a near real-time processing solution since micro-batch processing was used. Micro-batch processing is a type of traditional batch processing which runs batch processes on smaller accumulations of data. This approach is suitable when we need fast processed data but not necessarily in real-time. Average processing times from the data ingesting to the visualization is approx. 2–3 s (for our processing pipeline).

The second contribution of the paper is in the Spark application that enables to process data in memory with Spark engine and at the same time, it takes advantage of Apache Hadoop Yarn cluster. Additionally, the application implements advanced techniques for data processing optimization.

From an analysing point of view, flood-related tweets were processed and analysed. This article aims to use text analysis methods and location estimation techniques to analyse the flood situation by using only the information included in the processed Tweets. The location estimation technique based on the gazetteer was investigated in detail, and all pros and cons were evaluated and described.

Related work

This chapter investigates two directions of related studies: the role of Twitter as a popular social media in scientific works and frameworks for big data processing.

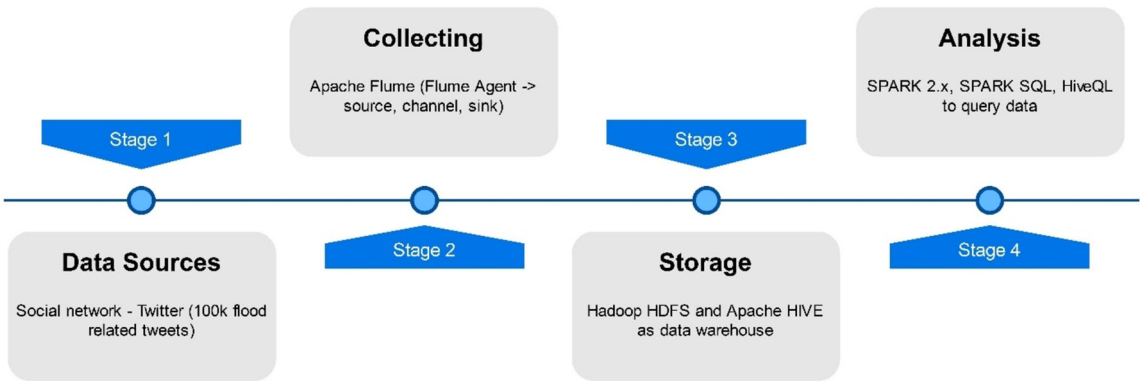

Fig. 1 General workflow

Twitter data analysing

Social networks have a range of roles within daily routines in modern society. In general, the common purposes of almost all social media are communication, information, advertising, and social events (Chianese and Piccialli 2016). However, they are also used in a more sophisticated way as supporting tools for analysing social and emergency events (Son et al. 2020). The engagements of social media in these domains have been presented by many scholars. For instance, (Yoo et al. 2016) used Twitter data during Hurricane Sandy where information diffusion theory for characterizing diffusion rates was applied, (Martin et al. 2019) applied Twitter messages to depict city events and evaluate their spatiotemporal characteristics, (Muralidharan et al. 2011) compared non-profit organizations and media by using Facebook and Twitter data during the Haiti earthquake in 2010 or (Lansley and Longley 2016) explored the use of an unsupervised learning algorithm to classify geo-tagged Tweets from Inner London. Currently, very attractive are the studies presented in (Melo and Figueiredo 2020; Wang et al. 2021) focused on COVID-19 twitter data analyses.

Among all social networks, Twitter is currently the most studied social medium in the emergency field (Simon et al. 2015). Twitter is ideal for these studies due to its open data policy (data are accessible through public APIs) and for its data content containing extractable spatial and temporal information. (Martinez-Rojas et al., 2018) revised the documents associated with Twitter data and emergencies and tried to determine which phase of the emergency is the most popular - Before, During or After. It was observed that the After phase is currently the most attractive for processing.

One of the most useful directions of Twitter data research is in natural disaster data processing, such as hurricanes (Huang and Xiao 2015; Vera-Burgos and Padgett 2020), earthquakes (Muralidharan et al. 2011), heavy rains or floods (Wang et al. 2018; Arthur et al. 2018). According to the literature review of (Martinez-Rojas et al. 2018), the most attractive events for research are earthquakes followed by floods and hurricanes, but widely explored are also storms and typhoons.

Research into the content of flood-related tweets has ranged from providing early detection data (Jongman et al. 2015) through depth extraction from posted photos (Fohringer et al. 2015) to studying the floods (Eilander et al. 2016) or analysing the social network after a disaster (Kim and Hastak 2018). All aforementioned studies proved the usefulness of extracted data from Twitter in flooding research.

There are a variety of textual analysis techniques for extracting required information from text-based sources. When it comes to a content analysing of Twitter data, many authors used n-gram method for identifying frequent text sequences. (Rossi et al. 2018; Al-Daihani and Abrahams 2016) applied this method for revealing the most discussed topics in the input datasets. Other studies usually utilized classical statistical methods or sentiment analysis. Typical applications of textual analysis are in political and emergency science (Yaqub et al. 2017).

Framework for data processing

In general, most of the social data-based applications focused on natural disasters, regardless of the phase of disaster they aim to solve, requiring fast collecting and analysing of a big amount of information (Landwehr et al. 2016). Such functionality delivers distributed frameworks for large-scale data processing. There are two major representatives of such frameworks - Apache Hadoop (Lu et al. 2020) and Apache Spark (Shafiee et al. 2018). Both open source software frameworks are considered as standards for developing data-intensive applications. These big data technologies have been implemented in many scientific areas for developing data-intensive analytics. As a proper example serves the research of (Shafiee et al., 2018) that enhanced water system models by integrating big data technologies. Apache Spark was used for integrating frequent data into water-related models and analysis. (Martin et al. 2019) proposed big data architecture and workflow based on Apache Spark. Data about city events in Valencia were processed and analysed. On the other hand, (Zvara et al. 2019) showed the optimization techniques of various data streams by means of the developed tracing engine and the CosmoHub web application based on Hadoop is presented in (Tallada et al. 2020).

After reviewing the literature from diverse directions, it was observed that the major effort of a scientific society is dedicated more to the specific local events (e.g., Haiti earthquake, Hurricane Sandy) than to the global perspective. It was observed that the objectives of locally-based studies focused mainly on general information, response or risk assessment. There are also presences of general perspective studies that are often associated with a detection of a given emergency in the early phase (Martinez-Rojas et al. 2018). Additionally, it has been identified in the literature that many studies do not provide detailed information about the mechanism of social data processing. They present only textual analysis or data visualization techniques and technology that is behind staying hidden. Social data and big data technologies are in a close relationship, and therefore we think that scientists should pay more attention to the processing technology/architecture. Studies, which focused on the whole pipeline of social data processing (data ingesting, storing, processing, analysing, and visualizing), are rarely published.

Methodology

Architecture and components

Choosing the appropriate data ingestion tool is one of the key challenges faced by data processing architectures. With the right data ingestion tool, one can instantly import, process, and store data from various data sources (e.g., social media data, network traffic data, and log data). There are plenty of options on the current data market such as Apache Flume, Apache Sqoop, Apache Kafka, Apache NIFI or Apache Storm. All mentioned tools offer similar functionality but with different levels of customization.

As a suitable data ingestion tool for this study, Apache Flume was selected and used for the first step of the data handling process. The main competitive advantage of Flume is in easy Twitter agent implementation, simple architecture, and efficiency. Furthermore, it is designed specifically for Hadoop and distributed file systems (e.g., HDFS), and therefore it fits spotlessly to our architecture. There is no need (in our case) to use complex software such as Kafka or Nifi, which require a more robust implementation strategy.

Apache Flume is a complex, distributed, and available service for collecting, aggregating, and moving large amounts of various data to a centralized data store. The main structure of the Flume is designed for the continuous data ingestion into Hadoop (HDFS) (Osman 2019). It runs in the form of one or more agents (JVM independent daemon processes) that contain three components: Source, Channel and Sink. In a simplified form, a Flume source consumes events (basic units) delivered to it by an external source and then stores the events into one or more channels. The sink removes the events from the channel and puts them into external storage like HDFS (Flume 1.9.0 User Guide 2020). This study designed Flume architecture as follows (Fig. 2):

- Source type: Twitter – source was connected to the Twitter interface to download tweets continuously. All data were serialized by AVRO serialization system. AVRO stores the data definition in JSON format.
- Channel type: Memory Channel – The events were stored into an in-memory queue. The drawback of this channel is the inability of data recovery in the case of agent failure. It acts like a buffer and a bridge between source and sink.
- Sink type: HDFS Sink – The sink delivered data into the Hadoop cluster/HDFS. HDFS was the final destination for ingested data.

Given below is a configuration file used for data ingestion in this experiment (Fig. 3). The line of source code, *TwitterAgent.sources.Twitter.keywords = flood*, corresponds to the filtered "flood" keyword.

HDFS Hadoop cluster

The Hortonworks Data Platform (HDP) was used as an appropriate open-source platform for our processing tasks. The HDP is a secure, enterprise-ready Hadoop distribution and consists of all important Apache Hadoop projects including Hadoop Distributed File System (HDFS), Hive, Yarn, Ambari or HBase.

Currently, there are two major Hadoop distributions on the data market – HDP and Cloudera. The main advantage of the HDP with respect to the Cloudera is in its completely free nature whilst Cloudera provides paid services (Cloudera has a commercial license, while Hortonworks has an open- source license). Both platforms are based on the same core of Apache Hadoop, and therefore, they are expected to have more similarities than differences. Despite many similarities and the same core, both platforms provide several key differences: (1) Hortonworks doesn't have any proprietary SW, whilst Cloudera has Cloudera Manager, Cloudera Search and Impala (SQL interface) (2) HDP is a native component on the windows server (3) slightly different security strategies.

The HDFS Hadoop cluster was established for two reasons: (1) distributed storage and (2) data- parallel processing. For the experiment, a small 10-node master/slave cluster was configured to fulfil the defined goals. The basic structure of the cluster is shown in the Fig. 4. The NameNode is the Hadoop single master that manages the file system and access to files.

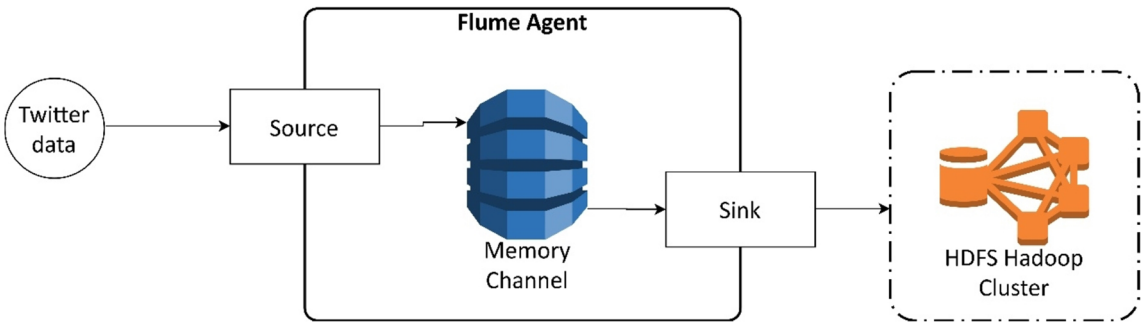

Fig. 2 Flume architecture

Fig. 3 Flume configuration file

```
TwitterAgent.sources = Twitter
TwitterAgent.channels = MemChannel
TwitterAgent.sinks = HDFS
# Describing/Configuring the source
TwitterAgent.sources.Twitter.type = xxxxx.TwitterSource
TwitterAgent.sources.Twitter.consumerKey= xxxxx
TwitterAgent.sources.Twitter.consumerSecret= xxxxx
TwitterAgent.sources.Twitter.accessToken= xxxxx
TwitterAgent.sources.Twitter.accessTokenSecret= xxxxx
TwitterAgent.sources.Twitter.maxBatchSize = 10
TwitterAgent.sources.Twitter.maxBatchDurationMillis = 200
TwitterAgent.sources.Twitter.keywords=flood
# Describing/Configuring the sink
TwitterAgent.sinks.HDFS.channel=MemChannel
TwitterAgent.sinks.HDFS.type=hdfs
TwitterAgent.sinks.HDFS.hdfs.path=hdfs://master.eu:8020/user/flume/tweets
TwitterAgent.sinks.HDFS.hdfs.fileType=DataStream
TwitterAgent.sinks.HDFS.hdfs.writeformat=Text
TwitterAgent.sinks.HDFS.hdfs.batchSize=100
TwitterAgent.sinks.HDFS.hdfs.rollSize=1024
TwitterAgent.sinks.HDFS.hdfs.rollCount=1000
TwitterAgent.sinks.HDFS.hdfs.rollInterval=30
# Describing/Configuring the channel
TwitterAgent.channels.MemChannel.type=memory
TwitterAgent.channels.MemChannel.capacity=10000
TwitterAgent.channels.MemChannel.transactionCapacity=1000
TwitterAgent.sources.Twitter.channels = MemChannel
TwitterAgent.sinks.HDFS.channel = MemChannel
```

The DataNodes mainly work as managers of the actual data physically stored on the nodes.

To get data into the preprepared cluster, the Flume agent was used to transfer data from Twitter API to HDFS. Consequently, the data were distributed across the nodes in the cluster and split into large blocks (size 128 MB) and independently replicated at multiple data nodes (Replication factor = 3). Replication factor is the number of times Hadoop framework replicates each Data Block. All blocks are stored on the local file system on the DataNodes. Block is replicated to provide a fault tolerance mechanism to the system, and therefore the system is able to recover yourself after a node in the cluster fails.

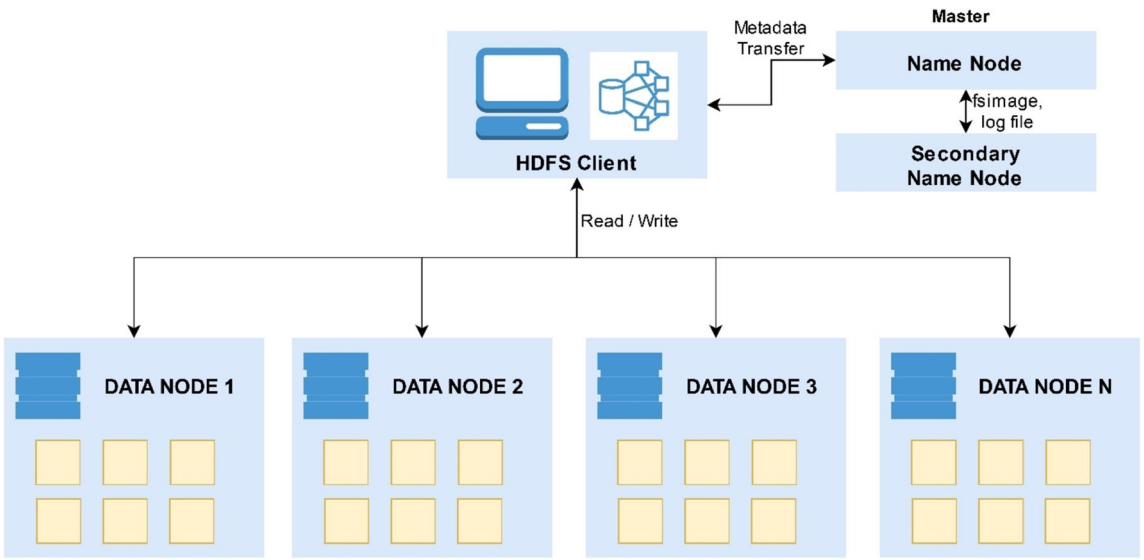

Fig. 4 HDFS Hadoop cluster architecture

All data processing systems require an effective resource manager providing convenient access to available resources at run time. Such resources may include CPU, memory, disk or network.

Apache Hadoop Yarn was applied as a technology for resource management in the proposed architecture. Yarn as a resource negotiator and job scheduler belongs to the Apache Hadoop's core components. While HDFS is the storage layer of Hadoop, Yarn is the resource management layer of Hadoop architecture. In the proposed cluster architecture, Apache Hadoop Yarn resides between HDFS and the processing engine that is used for running applications. In our case, the slower processing engine MapReduce was replaced by Apache Spark engine. There is no doubt that Yarn provides a big benefits such as dynamic resource allocation, supports multiple scheduling methods or scalability. Perhaps, the most significant benefit is the fundamental idea of Yarn to split up the functionalities of resource management and job scheduling into separate daemons to support varied types of processing and applications.

Hive and spark engine

Choosing the right database for the experiment was about understanding the needs of input data and architecture. Most importantly, we needed to select the database that supports the right data structure, size, and speed to meet the requirements of our experiment. Apache Hive was found as optimal to play a database role in the architecture.

The Apache Hive is ideal for analysing, reading, writing, and managing large datasets that are stored in distributed storage and queried by HiveQL (SQL abstraction for integrating SQL-like queries). There are a few main reasons for choosing Hive as a central warehouse: (1) It supports built-in connectors for various types of formats (CSV/TSV, JSON). (2) It provides schema on-Read (load as is without any changing or transformation, data interpretation during reading). This operation is faster than on-Write and improves performance. (3) Easy access to data via HiveQL, (4) Built on top of Apache Hadoop, (5) Query execution via Apache Spark engine. (6) Provides advanced techniques for large dataset analyses stored in HDFS. However, the Apache Hive is designed to meet mainly traditional warehousing tasks, and therefore it is not fully prepared for OLTP (Online Transaction Processing) workloads.

Several analysable Twitter fields were selected and consequently imported into the Hive data warehouse (e.g., ID, text, retweet count). The import command (Create External Table) can be modified to include more Twitter data, but it must have the same structure (name, data type) as the JSON fields in Twitter documentation. In this phase, the Hive is still unable to read the raw JSON data stored in HDFS. To do so, JSON SerDes (Serializers/Deserializers) was developed to map structured JSON data into tables. In general, the SerDes interface allows the user to instruct Hive how the data fields should be processed. The Hive Deserializer converts data fields into a Java object that is readable by the Hive and Serializer takes Java object and converts it into an appropriate format used for storing in the HDFS. ADD JAR is the command to invoke SerDes residing in the query for external table creation.

As Apache Spark engine has more advantages over the Map-Reduce like in-memory computation (Spark may be up to 100 times faster in memory, 10 times faster on disk), the Spark was decided to be a primary processing engine. Spark was developed to decrease the processing time of the Hadoop ecosystem and to overcome the limitations of MapReduce. Spark offers more features than Hadoop, but there is still a limitation - Spark does not provide its file management system, and therefore it is beneficial to integrate it with Hadoop. As shown in the below schema (Fig. 5), the Spark is a central component of the proposed architecture, but the MapReduce engine still can be used optionally for different use cases if necessary.

Spark can be run in a standalone cluster mode or take advantage of clusters based on a management framework like Yarn. The Spark application was developed in Scala and submitted to a preprepared Hadoop cluster using the master URL. When Spark application runs on Yarn, functionalities like resource management or scheduling are under the control of its management framework (Spark executors run in the form of containers). For submitting an application, a spark-submit script was used with the specified master flag. The connectivity between Spark application (Scala code) and Yarn cluster is provided by SparkContext telling to Spark how to access the cluster.

Spark application

The following section provides details about the Spark application preparation. The purpose of the application is in automated data transformation and operations defined in the workflow. Scala, as a native Spark language, was selected for developing of the application source code. In general, each Spark application consists of a driver and an executor process. The driver process deals primarily with the main function and task distribution across the executors, whereas the executors mainly do the assigned tasks. The application was run in the Hadoop Yarn cluster deployment mode, thus resource management, security, and scheduling are under the control of Yarn, and everything runs inside the cluster.

The structure of Spark application is depicted in (Fig. 6). In the first part, the SparkSession object was created as an entry point to the Spark SQL. SparkSession allows basic configurations related to accessing the Spark SQL services or loading tables from Hive warehouse. In the next step, Spark SQL was used to read data from an existing Hive deployment.

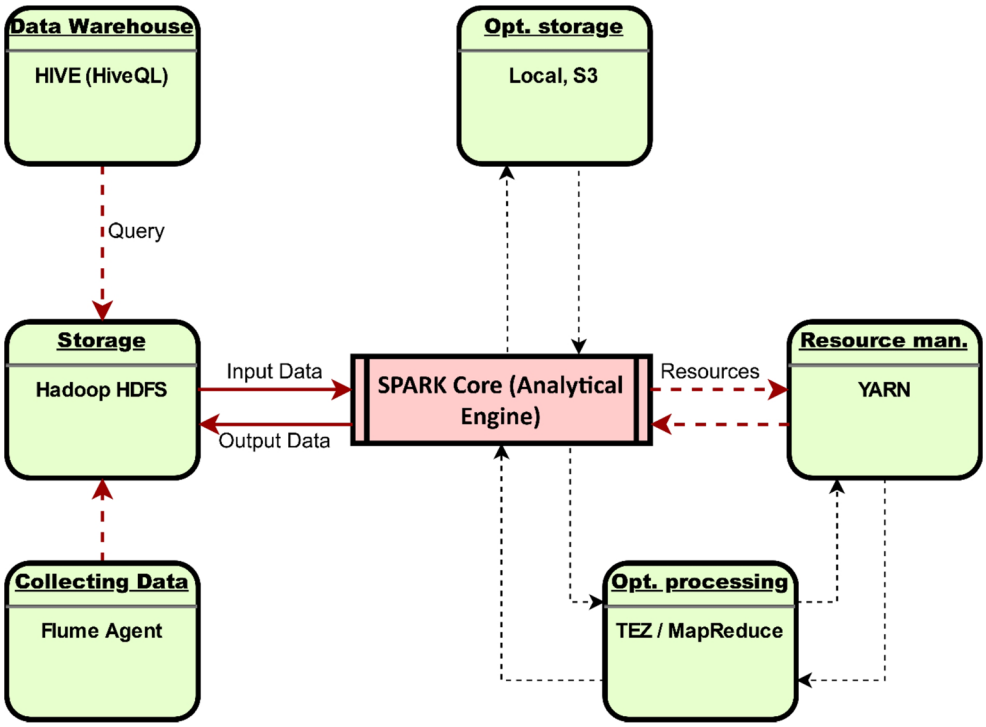

Fig. 5 Proposed architecture with Spark engine as core

Specifically, tables with Tweets and geographical names were loaded for further processing. After the comprehensive analytical part that was made on the top of Tweet table and performed with an essential statistic, word occurrence and context analysis (n-grams), the most interesting part of the application was followed. Extraction of geographical locations from Tweet table by using the gazetteer table, the Cross Join operation was used to produce results. The Cross Join operation provides outcomes based on multiplying of rows in the first table by the number of rows in the second table (Cartesian result). This part of the code was time-consuming since all combinations of rows had to be paired and processed. Result phase was affected by the distributed character of the data, and therefore the repartitioning operation was applied as well.

Twitter data collection and preprocessing

Twitter belongs to the most popular social networks with 500 million Tweets daily on average (Crannell et al. 2016; Ozturk and Ayvaz 2018). It provides social networking and valuable data that can be used for increasing public awareness about emergencies or monitoring and detecting disasters. Twitter is a micro-blogging service, which is a type of service using short text messages (known as Tweets) as a form of communication (Rossi et al. 2018). However, there are limits up to 280 characters for every Tweet (former maximum 140 characters). User's posts are public by default, and therefore can be obtained by Twitter API and consequently processed under the limitations specified in terms of service. Twitter data are ideal

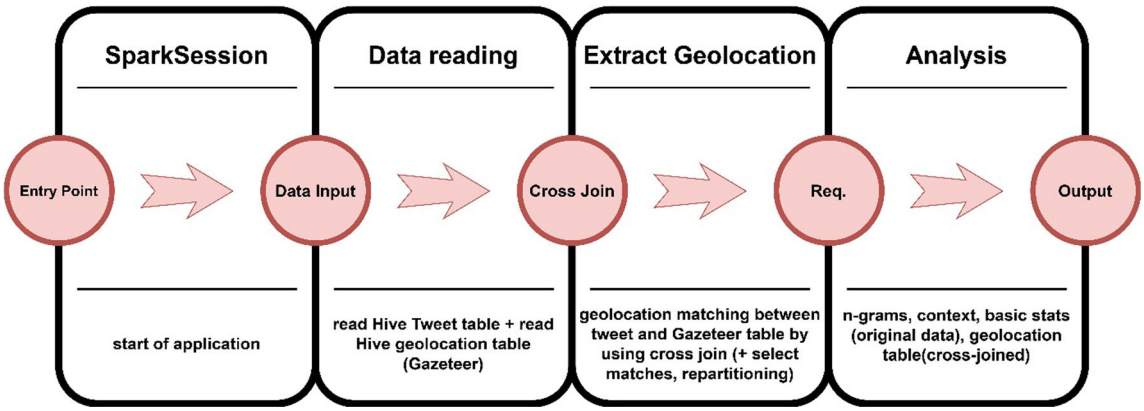

Fig. 6 Spark application structure

for scientific purposes for many reasons: (1) open data policy, (2) data availability is simple (API), (3) data often include spatial and temporal information, and (4) availability for a wide range of analyses.

Twitter data were streamed using Twitter API for five days from May 14 to May 18, 2018, and the keyword "flood" was set as a filter to ensure that all tweets will be strictly associated with the flood topic. Since "flood" is a multimeaning word, we applied measures for Tweet classification by using a machine learning approach. In this period, 99,989 Tweets were captured with 106,197 of "flood" word occurrence. For this dataset, only English written Tweets were taken into account for further analysing and processing.

Collected Tweets are formatted as the open standard file format JSON (JavaScript Object Notation). It is language-independent format based on key (string) value (string, number, boolean, array or object) pairs with named attributes and corresponding values. JSON is a text format usually used for transferring structured data to and from a server. Twitter API provides data encoded in JSON. JSON includes all fundamental objects (e.g., Tweet, Users), which encapsulate the core attributes describing the object (Twitter Developer 2020). For instance, each Tweet object consists of an author, ID, text message or geo-data, and each User object contains the Twitter name, ID or number of followers.

After the downloading process, the data classification and cleaning phase was following. This phase consists of reducing noise and removing irrelevant data to increase accuracy. The Twitter data require complex cleaning processes to ensure that text analysis identifies only valid and representative information.

Data classification by using machine learning

Social data classification by using machine learning approaches is a relatively new area of research. In the last few years, it focused on detecting spam and spammers, specific news (natural disasters, business, and politics), negative messages or categorizing emails (Alom et al. 2020; Bermejo et al. 2011). This chapter discusses why we need a machine learning approach and what our main flood data classification issue is.

We collected thousands of Tweets from Twitters API containing the specified keyword "flood". Since the word "flood" is a multiple-meaning word and not always represents the flood as a natural disaster (what is desired) in the context, we decided to apply processes that will be able to filter out noncorrect meanings. For instance, Tweets refer to floods of tears, a flood of feelings or flood of people are highly undesirable and have to be discarded, whereas Tweets related to a flood as a natural disaster are eligible.

The main goal of this part of our research is to develop a classifier, which will separate real flood Tweets from the non-flood (two-class classifier). For this purpose, we decided to utilize an algorithm called Multinomial Naïve Bayes, implementing the Bayes algorithm for multinomially distributed data. In this study, we deal with specific flood-oriented data that often contain sensitive information about a possible or upcoming natural disaster. Misclassification of such emergency data can lead to false warning messages, and therefore to get model accuracy as high as possible is one of our priorities.

Some of the most popular text classification algorithms include the Naïve Bayes group of algorithms and support vector machines (SVM). Since the SVM approaches require significant computational resources and complex "multidimensional" datasets to achieve optimal accuracy, the Naïve Bayes type of text classification was decided to be more suitable for our purpose. It provides a perfect balance between processing speed, computational resources, and accuracy on large datasets. Types of Naïve Bayes classifier: (1) Multinomial (our choice, best fit to our data) - able to classify data into multiple categories, and the predictors are the frequency of the words present in the input dataset (2) Bernoulli – less complex than Multinomial because the predictors are boolean variables only (3) Gaussian – the predictors use continues value (not discrete).

Naïve Bayes classifier

The naïve Bayes classifier is a standard probabilistic classifier assuming independence among terms. This model not only considers the terms appearing in each tweet but also the frequency of appearance. The classifier assigns each tweet to predefined classes with a strong independence assumption between objects. The naïve Bayes is a conditional probability algorithm calculating the probability of each tag for a given tweet. The method used for computing probabilities is based on Bayes Theorem. The formula for Bayes Theorem describes as (Zhang and Sakhanenko 2019):

$$P(A|B) = \frac{P(B|A)*P(A)}{P(B)} \quad (1)$$

It provides the posterior probability P (A|B), the prior probability P(A), the marginal probability P(B), and the conditional probability P (B|A).

Specifically, the Multinomial Naïve Bayes algorithm was utilized for tweet classification where the data are typically represented as word vector counts. The basic assumption to use it is that each feature (Tweet) is independent and equal (no effect on each other and the same weight). This kind of algorithm has been used in many studies (Harzevili and Alizadeh 2018; Jiang et al. 2016), and it has proven that accuracy, speed, and no excessive computational resources are not the only advantages. Experiments demonstrate that Bayes classifier is useful in many complex real-world situations and can outperform many similar purpose classifiers (Baesens et al. 2003).

Data and model preparation

Before we applied the machine learning classifier, we first divided our Tweet dataset into identifiable classes to know what kind of flood meanings our data include. We identified four classes of flood tweets: (1) Tweets with the flood as a natural disaster, (2) Religious or biblical flood tweets, (3) Tweets about tears and feelings (4) Other nonspecific meanings. Numerically expressed, first-class covered 74% of all Tweets while the others gradually 9%, 8%, and 7%. For flood tracking, we needed only the first class, and therefore we focused only on this group.

For the ML model purpose, the tweet dataset was split into a training and test data set (we used 6000 tweets for ML model train and test). The training dataset (75% of data) was used to train an algorithm while the test dataset (25%) to evaluate how good our already trained algorithm is. We manually classified a dataset of 6 k tweets into two classes – Real-Flood (first class) and Non-Flood (second class). Consequently, the labelled dataset was applied for training a Naïve Bayes algorithm. Since our dataset contains text only, we used word frequencies (n-grams) to calculate probabilities. Specifically, we decided to utilize unigrams, and we ignored sentence structure and word order.

The final step was to compute the probability for both classes and compare them which one is higher. In our case, the approach is based on probability calculation for each word and multiply them (P(word1|Flood) x P(word2|Flood) x P(word3|Flood)).

Scikit learn (python library) helped us to build a Multinomial Naïve Bayes model in Python. Laplace smoothing parameter was set to 0.7 to avoid issues with zero equality. There are many techniques for improving the performance of our model. For better performance, we removed stopwords (words providing little or no information such as do, is, was) and penalized words that appeared frequently in the dataset. Consequently, we removed special characters '#', '@' and punctuation by using Regexp Tokenizer. As the last process, we applied stemmer (English SnowballStemmer) for producing morphological variants of a root word (lower case conversion is available within the stemmer function).

Performance evaluation

The effectiveness and performance evaluation of the model was evaluated by selected standard metrics (confusion matrix, accuracy, f1-score, precision, and recall). Confusion matrix (CM) uses a matrix to describe the performance of a classification model. It often includes data about true positive/negative and false positive/negative predicted cases. Accuracy (A) is the ratio of the total number of correctly classified cases (for both classes) over the total number of all cases. F1-score (F1) is a weighted harmonic mean of the recall and precision. Precision (P) is defined as the ratio of true positives to the sum of true and false positives. Recall (Rc) demonstrates the ratio of true positives to the sum of true positives and false negatives.

Confusion matrix revealed the performance of a classifier. The matrix compared the actual target values with those predicted by the machine learning model (Fig. 7, left). The classification report provides key metrics for the test dataset (listed below, Fig. 7, right):

The model, which was trained on the training set, demonstrated *accuracy 0.907* on the test set. According to the classification report metrics, we considered the results as satisfactory. Only the precision value for class 0 reported less accurate results since for class 0 we got more false positives (for class 1 false positives are significantly less).

Cleaning phase

The cleaning phase involved several filters and text adjustment techniques (see chapter 3.3.2) (Lansley and Longley 2016; Al-Daihani and Abrahams 2016). Some additional techniques not mentioned in chapter 3.3.2 were applied as well: (1) Twitter abbreviations such as RT (retweets) or MT (modified tweet) were omitted, (2) Hyperlinks (after HTTP) and user names were excluded from the analysis as well. (3) Line break elements ("\n") were removed from the block of text to keep the consistency of the database. All techniques were performed directly in the database with the help of Hive and Regex functions.

The Twitter data set was also investigated to identify spam and duplicate Tweets. Authors (Ozturk and Ayvaz 2018) noticed that the iterative search of Twitters API caused duplicates with missing values. Twitter API is not a live stream, and it provides tweets at a particular time window, and therefore it is likely that API can return set of overlapping Tweets. In this use case, we found out the same issue with duplicates caused by an overlapping time window, and therefore all duplicates with nonvalue records were immediately removed - 0.7% (695) of Tweets.

Presence of spam is a well-known matter on Twitter. To identify and remove spam in this dataset, we used a simple assumption based on post frequency. (Twitter User Data 2020) discovered that users tweeting over 150 times a day can be marked as bots. Even though it was discovered that around 10.5% of Twitter accounts might be bots (Chu et al. 2012), no suspicious accounts were detected in the "flood" Tweet dataset (only common frequencies were observed).

The trained ML model was applied to the cleaned and classified dataset of 99, 289 Tweets (minus overlapped (695)) and 9007 of them were identified as wrong meaning Tweets.

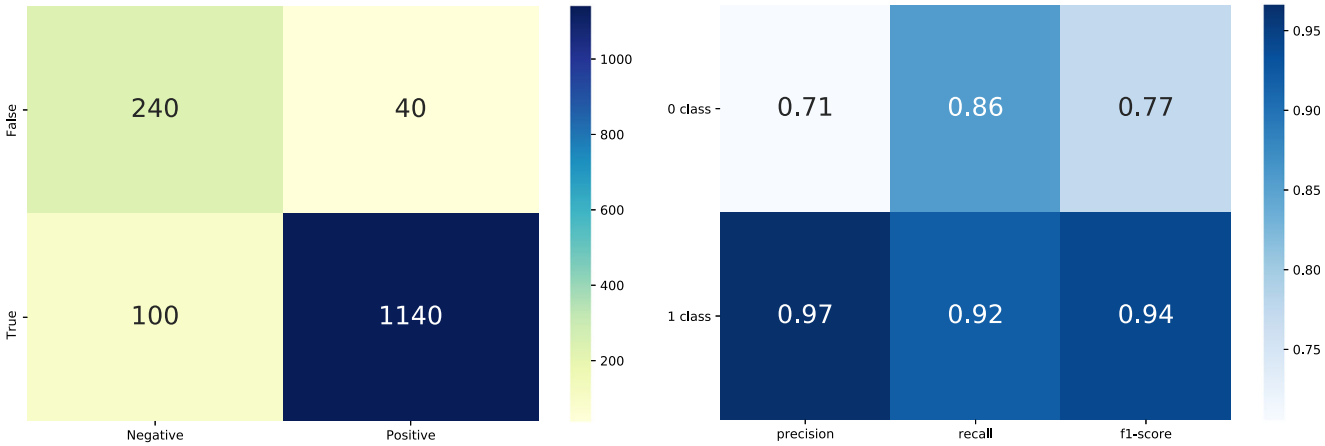

Fig. 7 Confusion matrix (left) and classification report (right)

Data analysis

The preprocessed flood dataset contains:

- 89,587 is the number of real flood Tweets
- 1117 Tweets containing the exact geographical coordinates (geotagged tweets)

Content analysis

Text and content analysis techniques were used to show a qualitative aspect of the dataset. Word pattern frequencies were analysed for flood-related Tweets by using Word Clouds. Word patterns were extracted by the language analysis pipeline identifying specific patterns of terms with required features. Built-in function ngrams() provided by HIVE was utilized to find the most frequent n-grams (word patterns) from the input dataset. In this analysis, we applied commands to return unigram (single word), 2-g (two-word), and 3-g (three-word) occurrences from given text sequences. N-grams allowed us to develop Word Cloud images, which revealed the frequency and usage of single and multiwords.

Figure 8 represents Word Clouds of the most frequent single words (left) and two words (right). It was observed that the top frequent single word was "warning" with occurrence 12,596 (term "flood" as filter word was excluded). The term "warning" was often used in the announcements about an imminent or occurring flood event in the warned state, county or city. The second most frequent word was "flash" (occurrence 11,020), which was used in the context related to a particular type of event - flash flood. It indicates that a flash flood is occurring or will occur in the area. The third most repeated word was "watch" (10,318), which was also used in the context of "soft" flood alert announcements. Other less frequent single words with their occurrences were: rain (6911), river (6574), time (6461), issued (6361), water (6171), flooding (5835), people (5505), advisory (5374), county (5181), area (4324), weather (3875) and heavy (3353).

Top word pair (two-word sequence or bigram) was determined as "flood warning" (8725). It also indicates the presence of a flood threat in the area. Other most frequent pairs only confirmed the probability of approaching a flash flood disaster: flash flood (7205), flood watch (5417), flood advisory (3802), flood-affected (1196), warning issued (1092) and heavy rain (857).

Fig. 8 Word Clouds of the most frequent single words (left) and two-words (right)

It was noted that the most frequent trigrams were "flash flood warning" (3952) and "flash flood watch" (1700). These terms are probably related to the two types of alerts for flash floods that are issued by the National Weather Service (NWS) (Flood Warning Vs. Watch 2020). The NWS "flash flood watch" means that conditions are favourable for flash floods while "flash flood warning" represents that a flash flood is approaching.

Based on the content of the frequency analysis, it was observed that Tweet messages provided valuable information to determine the type of flood ("flash") and the initial phase of the flood event ("warning" and "watch"). It appears that the majority of Tweets refer to approaching flood disasters, and therefore different types of flood warning announcements were captured. However, it has to be taken into account that flood warnings cannot only be issued but also cancelled, changed, extended, updated or expired.

Figure 9 depicts the most repeated unigrams for cities (left) and rivers (right) that appeared in the dataset. An additional Word Cloud output showed the probable territory affected by the flood alerts. Word Clouds displayed as the most prominent words Alexandria (U.S. city, Virginia), Columbia (U.S. city, Florida) and in the "river" part it is Columbia River (U.S., Washington) and Des Moines River (Iowa). The remaining displayed top words are also associated only with U.S. regions. For instance, there were recorded Montgomery (city, Pennsylvania), Nelson (city, Virginia), Roanoke (city, Virginia), Portage (river, Michigan) or White (river, Arkansas).

It was identified that almost 98% of the captured geographical regions belong to U.S. territories and only 2% to the rest of the world (e.g., Japan, Germany). It is interesting to observe that most posts were recorded for states in the East-Central part of U.S.A - Virginia (1314), Pennsylvania (276), and Michigan (296). Upon further investigation, it was found that the abnormally big value for Virginia was caused by frequent retweeting of the original tweets (almost 80% of all Tweets were retweets).

Four Word Clouds were extracted, and several interesting word patterns appeared. This analysis demonstrated that the Word Cloud method produces effective summaries for Tweet based datasets. In our case, the analysis revealed the approaching flood to the East-Central part of U.S.A.

Time distribution and user analysis

Figure 10 summarizes the distribution of tweets over time (the period from May 14 to May 18). It was observed that high posting activity was recorded during Wednesday and Thursday. It indicates the presence of user reactions to the current weather conditions by using flood posts. 28% of flood tweets were posted during Wednesday and only 13% on Friday. Wednesday was used mainly to issue warnings, whereas Friday was covered with expired, changed, extended or updated warnings. Friday was also evaluated as the day with the lowest posting activity.

In the world of social media, user-generated content such as messages or comments are automatically generated every single moment. The content of Twitter messages strongly depends on the type of user. In the case of emergency messages containing often sensitive warning information, who post it is a relevant question. People can almost anonymously post anything, and therefore one has to be careful whose messages will be processed. In an emergency case, mainly tweets from valid and official accounts should be engaged.

In this dataset, the original tweets were mostly posted by users/accounts that are oriented on weather alerts or storm tracking. For instance, National Weather Service, Dynamic Weather Agency, Storm Spotter or Weather Alerts were the main contributors to announce the state of flood warnings. It is interesting to notice that most official agencies and less single person accounts were posting original emergency information. Single person users/accounts used the original messages to reshare them, and sometimes, they add as a comment more specific or locally detailed information. Photos and videos were sparsely attached to the original or reshared messages.

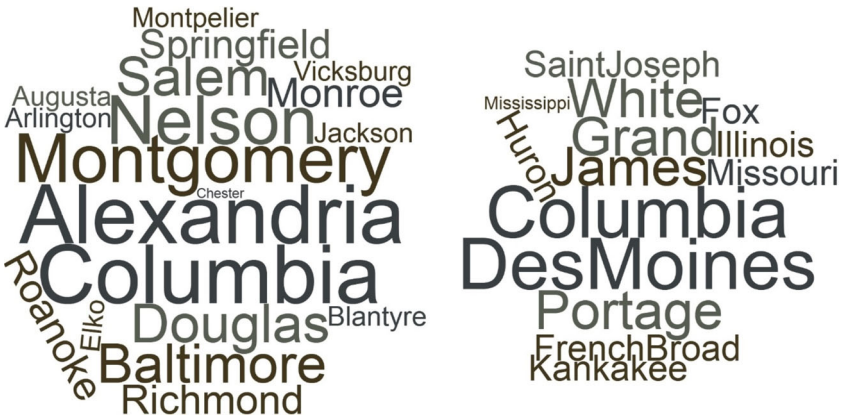

Fig. 9 Word Clouds of the most frequent cities (left) and rivers (right)

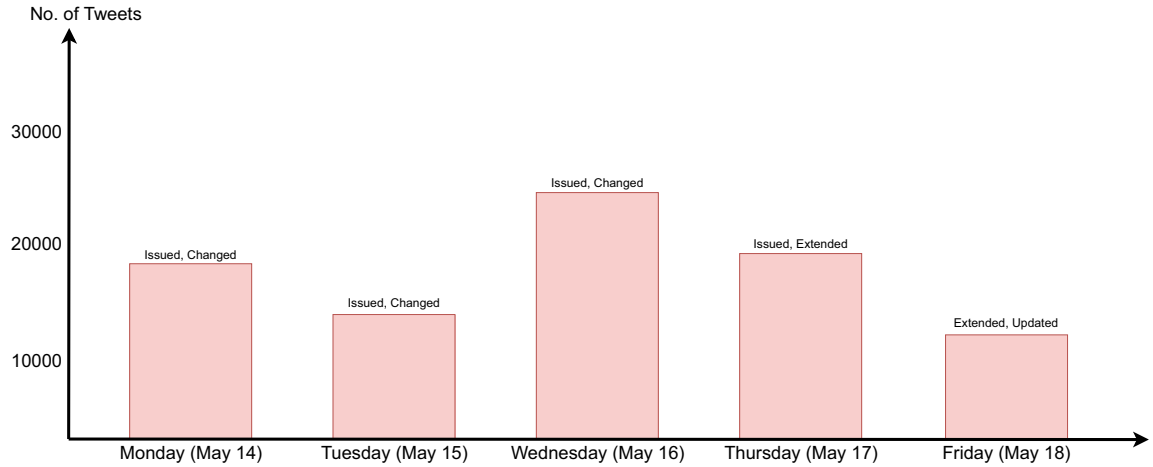

Fig. 10 Tweets - time distribution

Spatial features in twitter messages

Twitter messages include several geo-location possibilities that can be used for spatial information extraction. The first and most accurate method is obtaining locations from geotagged tweets. These types of tweets contain geographical coordinates in the form of latitude and longitude and usually are related to the time of posting (GPS has to be enabled). The second option is encapsulated in the user profile as a location attribute where a Twitter user can define its home location with a character restricted textual form. A different source of geographical location is a tweet text itself. It is a free-text field that is limited to 280 characters, which occasionally can contain mentions about geographical names such as cities, rivers or countries (the unigrams approach was already presented in chapter 4.1). Each of these location sources has its pros and cons, and therefore they should be utilized very carefully. Moreover, in the case of GPS and user profile locations, even if these locations are identified correctly, they might not correspond to the place affected by the flood event. For example, user profile locations are almost always related to the home address of the account holder, and therefore these locations are not suitable for flood location analysis. A similar issue can appear in GPS locations, these locations refer to the place of a device and again not to the place of the flood area.

In the next subchapter, *the gazetteer location method* (Ozdikis et al. 2017) is described in detail, including a discussion about its strengths, weaknesses, and challenges. The chosen method, based on the text itself, was decided as an appropriate experiment to utilize in this study.

Extraction of flood localization (*gazetteer location method*)

Tweet messages require text processing to extract the expected spatial information. To do so, the text was preprocessed (chapter 3.4) and consequently cross-joined with a gazetteer. World Cities database (World Cities Database 2020) was utilized as a gazetteer of cities and towns. The database provides accurate and up-to-date information about cities all around the world with fields such as latitude, longitude, population, density or country. The cross-join results were based on searching for a particular city or town (gazetteer) in each Tweet message (Fig. 11). The cross-join created combinations of every row from two tables: Tweet messages and gazetteer (field: city) and provide outputs based on *where* clause (matching strings from both tables).

The process of spatial data extraction revealed many limitations and weaknesses (described in Table 1). These limitations were often related to the contextual issues and global character of the dataset.

All listed weaknesses and limitations had a definite impact on the final results. It was observed that the proposed method of geo-location extraction produces redundant and noise data, and therefore it is not always possible to obtain high-quality spatial data. The biggest issue is considered fake detection that was mainly caused by enabling substring search and detecting cities with the same names. However, there are methods to mitigate the impact of the limitations. To improve the results, regular expressions were implemented to detect only exact standalone words (according to the gazetteer) and no substrings. Regular expressions were also used to deal with upper and lower case issues. An additional measure, taking only original messages into account could decrease redundancy. It was observed that 80% of important information is included in the original tweets (OT), not in re-tweets (RT). However, this measure was not applied in this case due to preserving a complete content history.

After the implementation of above-mentioned measures, overall, 29,705 geo-locations were captured in the Tweet dataset. 362 of them were evaluated as fakes and 29,211 as duplicates (re-sharing information). 132 locations were confirmed as unique and related to the flood topic (130 in the US and 2 in Japan). Main flood-tweet contributors in the USA were parts: Maryland, Virginia and West Virginia. Florida (Southeast), due to the fast-

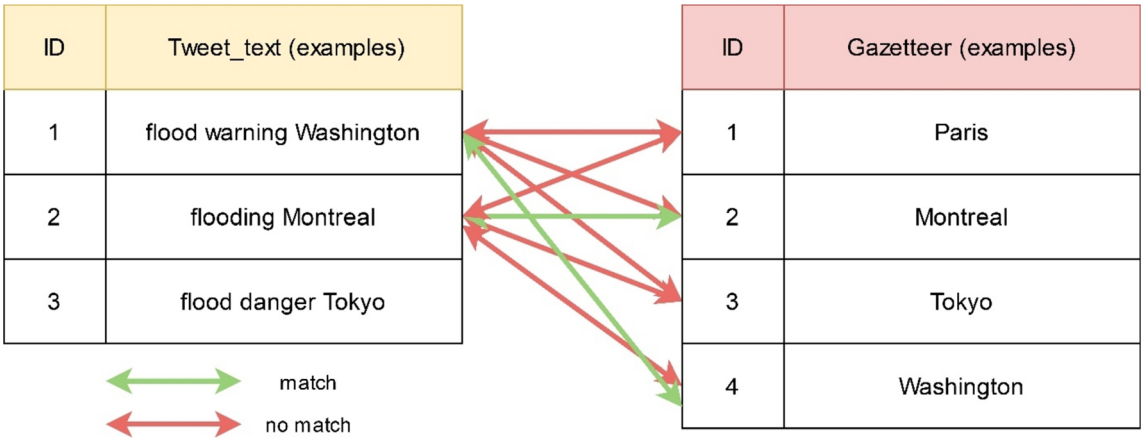

Fig. 11 Fundamentals of cross-join searching with clause

approaching tornado, was evaluated as a flood-warning zone as well. Figure 12 depicts a sample of filtered data – flood confirmed locations with edited Tweets.

This part of the research proves that gazetteer based methods can be used for searching locations in global datasets, but the possibility of missing, inaccurate or redundant information has to be taken into account. Main benefits were found in a simple implementation, relatively fast processing, and possible modularity (Table 2). We believe that gazetteer adjustment just to a particular country or region could deliver more accurate data.

Official flood and weather summary: May 14–18, 2018

This section is designed to provide the real flood situation occurring in the regions between 14-May-2018 and 18-May-2018. We collected only valid flood and weather reports from official organizations to compare with the extracted data. Severe weather was reported mainly for U.S. territories, and therefore primary data sources included services such as U.S. Geological Survey (USGS) and NOAAS National Weather Service (NWS). A brief overview of the situation in the USA is given below (edited highlights) (National Climate Report 2018; Floodlist 2018):

Table 1 Limitations and weaknesses

No.	Limitation	Description
1	Items in gazetteer	What is not in the gazetteer, it does not exist in the processing engine, and therefore cannot find it.
2	Duplicity	Results provided more than one location for the same geo-name. Georgetown (US, Gambia, Guyana), Nelson (US, Canada), Alexandria (US, Romania, Egypt) or Chester (US, UK), were localized in more than one country (+need additional information about a state for proper localization).
3	Non-geographic meaning	Geographical location was detected but with different meaning. For instance, sale (Australian city), colon (Cuba) or alert (Canada) were incorrectly evaluated as geo–locations.
4	Data redundancy	The same city was recorded multiple times due to resharing the same message.
5	Substring detections	Cities like Wa (Ghana), Ho (Ghana), Ita (Paraguay) or Po (Burkina Faso) were detected many times as substrings (e.g. Wa – Warning, Ho – Bartholomew, Ita – Ouachita River, Po - responders).
6	A large amount of data.	This method produces a large amount of data that must be filtered to obtain proper results (high response rate).
Tweet related limitations		
7	Spatial resolution	There are no rules or procedures how to announce locations in Tweets, and therefore various spatial resolutions appeared, ranging from country (Japan), state (US), cities (Washington) to smaller spatial units, such as local regions or streets.
8	Abbreviations	Mainly for US states (e.g., AZ, ID, IL) and sometimes user-defined city codes (e.g., Wash, Nash). These types of city codes are difficult to detect since they are proposed beyond any rules.
9	Spelling mistakes and grammatical errors.	Some of the geo-locations can be lost due to user spelling mistakes or grammatical errors.

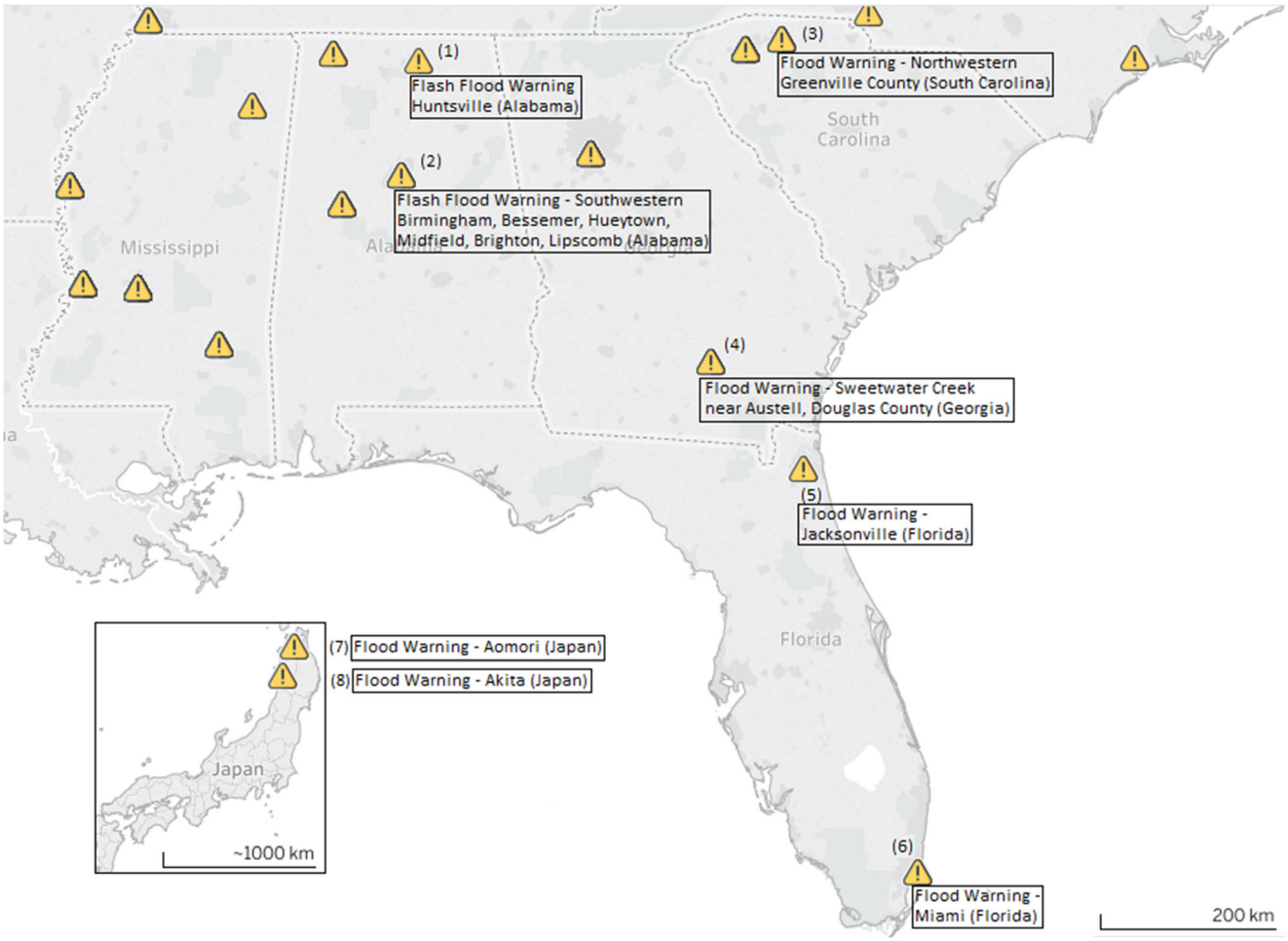

Fig. 12 Sample of filtered data (USA and Japan) – flood confirmed locations with edited Tweet messages

- *USGS Original report*: Severe storms caused major damage in Northeastern USA on Tuesday, 15 May, 2018. Strong winds caused most of the damage.
- Intense rainfall (up to 10 in.) over the past two days (May 15–17, 2018).
- Flood conditions are anticipated to persist and (or) recur over the next several days as additional rainfall (4 to 5 in.) is forecast.
- Severe flooding was reported in parts of Maryland, in particular Montgomery and Fredrick counties, where up to 6 in. of rain fell during the storm. Hail up to 2.5 in. (63.5 mm) was also reported.
- Severe flooding was reported in parts of Maryland, in particular Montgomery County, where up to 6 in. of rain fell during the storm. Hail up to 2.5 in. (63.5 mm) was also reported.
- *NWS 16 May 2018*: Flash Flood Emergency for Maryland. Flash flooding is already occurring.
- Flooding and mudslides were widespread across the region, and Florida and Maryland each had their wettest May on record.

Table 2 Stregths and benefits

No.	Strengths	Description
1	Simplicity	A simple implementation to data platforms.
2	Relatively fast	Depends on data volume and processing technology.
3	Modularity	Gazetteer can be extended with missing geolocations or adapted to specific local names. Usefulness enhances if combined with other geo-databases.
4	Large data	Easy to apply to large datasets.
5	Resources	No need of extra resources (overall inexpensive method).

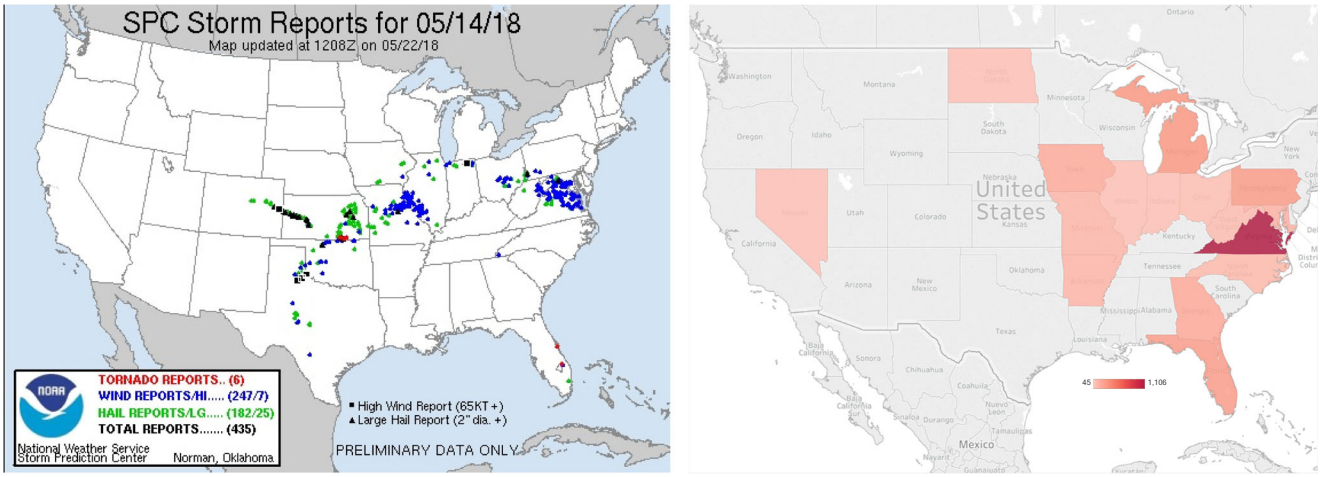

Fig. 13 Comparison of NOAA Storm report and flood-tweet map (sum, spatial unit: U.S. states) for Monday 14 May 2018

- Flooding alerts have been issued across much of the Mid-Atlantic and into parts of the Carolina's due to very heavy rain that's expected across the region over the next two days at least days.
- 17 May 2018: Tropical moisture continues to flow across parts of the Southeast (Florida). This pattern will keep rain showers and thunderstorms.

The most dramatic events were reported from Northeast (Maryland) and Southeast (Florida) of the USA. While Maryland was affected by flash flooding, Florida reported the occurrence of tornados. Mid-Atlantic, as the most endangered area, was covered by issued flooding alerts and warnings. It is obvious that the main keywords and geolocations from official reports correspond with the extracted word clouds (e.g., flash flood, heavy rain, flood warning). The gazetteer method correctly determined Mid-Atlantic as the most affected area with flood warnings.

The NOAAS National Weather Service published Storm reports in the form of a map including tornado, wind, and hail reports (Fig. 13 left) (Storm Prediction Center 2018). It showed us regions affected by severe weather, and thus areas with high probability of flood warnings and alerts. Comparison with the extracted data revealed similarities between the flood tweet distribution (gazetteer method) and storm report map (Fig. 13 right). Both sources identified the most endangered areas, Central and Northeast part of the U.S.

Discussion, future work and limitations

The weakness of the designed data infrastructure is in data visualisation. This step was omitted from the automatic process because the ecosystems do not support such advanced visualisations (e.g. maps). Therefore, some outcomes were plotted separately using suitable tools (e.g., Tableau). This step will be the object of interest in the future, and thus it is intended to extend an automatic processing chain with a data visualisation step by using available data visualisation technologies. This task is challenging and at the same time important because proper visualisation can improve data interpretation. During flood emergencies, a large number of Tweets are generated, and therefore proper visualisations can highlight important aspects that the user should focus on.

Another future work regards to analysis and extension to more geolocation methods and languages. For instance, the Czech language will be challenging to implement in such an environment due to the complicated language structure containing many special characters. From the location method point of view, it could be interesting to adjust a gazetteer just to the specific region or country. This approach could bring more accurate results with less redundancy. It could eliminate the errors that arose from the global nature of the dataset.

A deeper study for specific days will be performed to obtain more detail-oriented information. The expectations are to analyse the content of flood-related tweets and user posting behaviour during days with a high and low occurrence of warnings.

From this analysis, it has been seen that flood tweets are generally informative, and therefore it would be interesting to cross-join not only with city-based searching but also with rivers, counties or countries. We think that the comparison and evaluation of the mentioned spatial units separately and consequently together could bring enhanced information about the flood locations.

Conclusion

The main aim of this study was to propose and implement an architecture for social data processing by using big data platforms and perform analysis to discover useful flood information from the Twitter dataset (focus on spatial features).

Comprehensive in-memory data processing and analytics architecture was designed to complete tasks related to social data processing. The experimental architecture employed the two most used big data platforms Apache Spark and Apache Hadoop. Spark processing engine was applied to use Hadoop Yarn cluster and its HIVE data warehouse. Spark Scala application was developed to run data processing commands automatically and to generate the appropriate output data.

The experiment confirmed the advantages of many well-known features of Spark and Hadoop in social media data processing. Mentioned data platforms were effective in data ingesting, storing as well as processing and analysing. However, implementing such advanced technologies requires significant computing resources to deal with in-memory computations (RAM, storage area). It was observed that such technologies are prepared to deal with social media data streams, but there are still challenges that one has to take into account. The main challenge is related to the specific nature and structure of social data.

For this study, approximately 100 k Twitter messages were processed and analysed. Messages were related to the flooding domain and collected over a period of 5 days (14 May – 18 May 2018). These data included all regions and only English written messages.

According to ML modelling results, it can be claimed that the classification of messages containing the multimeaning flood term is challenging. We decided to use the Machine learning approach, specifically, the Multinomial Naïve Bayes algorithm, to achieve two-class classification goal (flood and no-flood). The key model performance indicators proved that the selected Bayes algorithm is accurate enough to be used as "separator" for multiple-meaning words and messages.

It was observed that Twitter messages with some considerations are informative enough to be used to estimate general flood alert situations in particular regions. These regions are limited mainly by the number of Twitter users, and in this case by language as well. Text analysis techniques used in this study proved that Twitter messages contain valuable general and spatial flood information. Further data analysing revealed that the gazetteer method can be utilized to extract geographical localization with some limitations.

Other data analysis issues were associated with the English language, data quality available on Twitter and data sparsity, which was caused by the distribution of Twitter users all around the world.

It was shown how big data platforms can be used together with Twitter data analysis and automatically extract useful information in near real-time. Big data platforms proved to be effective in social media data processing and analysing.

We can conclude that the analysing of social media data in near real-time is still a nontrivial matter. There is still several technical and data analysing challenges that need further research. This work provides a particular solution that we hope will be helpful for researchers working in this area.

Code availability Not applicable.

Authors' contributions Not applicable.

Funding This work was supported by The Ministry of Education, Youth and Sports from the National Programme of Sustainability (NPS II) project "IT4Innovations excellence in science - LQ1602".

Data availability Tweets are available for free from the Twitter API.

Declarations

Conflicts of interest/competing interests The authors declare that they have no competing interests.

References

Al-Daihani SM, Abrahams A (2016) A text mining analysis of academic libraries' tweets. J Acad Libr 42:135–143. https://doi.org/10.1016/j.acalib.2015.12.014

Alom Z, Carminati B, Ferrari E (2020) A deep learning model for twitter spam detection. Online Soc Netw Media 18:100079. https://doi.org/10.1016/j.osnem.2020.100079

Arthur R, Boulton CA, Shotton H, Williams HTP (2018) Social sensing of floods in the UK. PLoS One 13:1–18. https://doi.org/10.1371/journal.pone.0189327

Baesens B, Gestel TV, Viaene S, Stepanova M, Suykens J, Vanthienen J (2003) Benchmarking state-of-the-art classification algorithms for credit scoring. J Oper Res Soc 54:627–635. https://doi.org/10.1057/palgrave.jors.2601545

Bermejo P, Gamez JA, Puerta JM (2011) Improving the performance of Naïve Bayes multinomial in email foldering by introducing distribution-based balance of datasets. Expert Syst Appl 38:2072–2080. https://doi.org/10.1016/j.eswa.2010.07.146

Chianese A, Piccialli F (2016) International workshop on Data Mining of Iot Systems (DaMIS): a service oriented framework for analysing social network activities. Procedia Comput Sci 98:509–514. https://doi.org/10.1016/j.procs.2016.09.087

Chu Z, Gianvecchio S, Wang H, Jajodia S (2012) Detecting automation of twitter accounts: are you a human, bot, or cyborg? IEEE T Depend Secure 9:811–824. https://doi.org/10.1109/TDSC.2012.75

Crannell WC, Clark E, Jones C, James TA, Moore J (2016) A pattern-matched twitter analysis of US cancer-patient sentiments. J Surg Res 206:536–542. https://doi.org/10.1016/j.jss.2016.06.050

Eilander D, Trambauer P, Wagemaker J, Loenen AV (2016) Harvesting social media for generation of near real-time flood maps. Procedia Eng 154:176–183. https://doi.org/10.1016/j.proeng.2016.07.441

Flood Warning Vs. Watch (2020) https://www.weather.gov/safety/flood-watch-warning. Accessed 5 November 2020

Floodlist (2018) USA – Deadly Storms Hit North East, Flash Floods in Maryland. http://floodlist.com/america/usa/usa-storms-north-east-flash-floods-maryland-may-2018.

Flume 1.9.0 User Guide (2020) https://flume.apache.org/FlumeUserGuide.html. Accessed 5 November 2020

Fohringer J, Dransch D, Kreibich H, Schroter K (2015) Social media as an information source for rapid flood inundation mapping. Nat Hazards Earth Syst Sci 15:2725–2738. https://doi.org/10.5194/nhess-15-2725-2015

Harzevili NS, Alizadeh SH (2018) Mixture of latent multinomial naive Bayes classifier. Appl Soft Comput 69:516–527. https://doi.org/10.1016/j.asoc.2018.04.020

Hill D, Kerkez B, Rasekh A, Ostfeld A, Minsker B, Banks MK (2014) Sensing and cyberinfrastructure for smarter water management: the promise and challenge of ubiquity. J Water Res Pl 140. https://doi.org/10.1061/(ASCE)WR.1943-5452.0000449, 01814002

Huang Q, Xiao Y (2015) Geographic situational awareness: mining tweets for disaster preparedness, emergency response, impact, and recovery. ISPRS Int Geo-Inf 4:1549–1568. https://doi.org/10.3390/ijgi4031549

Jiang L, Wang S, Li C, Zhang L (2016) Structure extended multinomial naive Bayes. Inform Sciences 329:346–356. https://doi.org/10.1016/j.ins.2015.09.037

Jongman B, Wagemaker J, Romero BR, Perez ECD (2015) Early flood detection for rapid humanitarian response: harnessing near real-time satellite and twitter signals. ISPRS Int J Geo-Information 4:2246–2266. https://doi.org/10.3390/ijgi4042246

Kim J, Hastak M (2018) Social network analysis. Int J Inform Manage 38:86–96. https://doi.org/10.1016/j.ijinfomgt.2017.08.003

Landwehr PM, Wei W, Kowalchuck M, Carley KM (2016) Using tweets to support disaster planning, warning and response. Safety Sci 90:33–47. https://doi.org/10.1016/j.ssci.2016.04.012

Lansley G, Longley PA (2016) The geography of twitter topics in London. Comput Environ Urban Syst 58:85–96. https://doi.org/10.1016/j.compenvurbsys.2016.04.002

Lu HC, Hwang FJ, Huang YH (2020) Parallel and distributed architecture of genetic algorithm on apache Hadoop and spark. Appl Soft Comput 95:106497. https://doi.org/10.1016/j.asoc.2020.106497

Martin A, Julian ABA, Cos-Gayon F (2019) Analysis of twitter messages using big data tools to evaluate and locate the activity in the city of Valencia (Spain). Cities 86:37–50. https://doi.org/10.1016/j.cities.2018.12.014

Martinez-Rojas M, Pardo-Ferreira MDC, Rubio-Romero JC (2018) Twitter as a tool for the management and analysis of emergency situations: a systematic literature review. Int J Inform Manage 43:196–208. https://doi.org/10.1016/j.ijinfomgt.2018.07.008

Melo TD, Figueiredo CMS (2020) A first public dataset from Brazilian twitter and news on COVID-19 in Portuguese. Data Brief 32:106179. https://doi.org/10.1016/j.dib.2020.106179

Muralidharan S, Rasmussen L, Patterson D, Shin JH (2011) Hope for Haiti: an analysis of Facebook and twitter usage during the earthquake relief efforts. Public Relat Rev 37:175–177. https://doi.org/10.1016/j.pubrev.2011.01.010

National Climate Report - May 2018 (2018) https://www.ncdc.noaa.gov/sotc/national/201805.

Osman AMS (2019) A novel big data analytics framework for smart cities. Future Gener Comp Sy 91:620–633. https://doi.org/10.1016/j.future.2018.06.046

Ozdikis O, Oguztuzun H, Karagoz P (2017) A survey on location estimation techniques for events detected in twitter. Knowl Inf Syst 52:291–339. https://doi.org/10.1007/s10115-016-1007-z

Ozturk N, Ayvaz S (2018) Sentiment analysis on twitter: a text mining approach to the Syrian refugee crisis. Telemat Inform 35:136–147. https://doi.org/10.1016/j.tele.2017.10.006

Pradeep D, Sundar C (2020) QAOC: novel query analysis and ontology-based clustering for data management in Hadoop. Future Gener Comp Sy 108:849–860. https://doi.org/10.1016/j.future.2020.03.010

Rossi C, Acerbo FS, Ylinen K, Juga I, Nurmi P, Bosca A, Tarasconi F, Cristoforetti M, Alikadic A (2018) Early detection and information extraction for weather-induced foods using social media streams. Int J Disast Risk Re 30:145–157. https://doi.org/10.1016/j.ijdrr.2018.03.002

Schneider S, Check P (2010) Read all about it: the role of the media in improving construction safety and health. J Saf Res 41:283–287. https://doi.org/10.1016/j.jsr.2010.05.001

Shafiee ME, Barker Z, Rasekh A (2018) Enhancing water system models by integrating big data. Sustain Cities Soc 37:485–491. https://doi.org/10.1016/j.scs.2017.11.042

Simon T, Goldberg A, Adini B (2015) Socializing in emergencies – a review of the use of social media in emergency situations. Int J Inf Manag 35:609–619. https://doi.org/10.1016/j.ijinfomgt.2015.07.001

Son J, Lee J, Oh O, Lee HK, Woo J (2020) Using a heuristic-systematic model to assess the twitter user profile's impact on disaster tweet credibility. Int J Inform Manage 54:102176. https://doi.org/10.1016/j.ijinfomgt.2020.102176

Storm Prediction Center (2018) https://www.spc.noaa.gov/exper/archive/event.php?date=20180514.

Tallada P, Carretero J, Casals J, Acosta-Silva C, Serrano S, Caubet M, Castander FJ, Cesar E, Crocce M, Delfino M, Eriksen M, Fosalba P, Gaztanaga E, Merino G, Neissner C, Tonello N (2020) CosmoHub: interactive exploration and distribution of astronomical data on Hadoop. Astron Comput 32:100391. https://doi.org/10.1016/j.ascom.2020.100391

Twitter Developer (2020) https://developer.twitter.com/en/docs/tutorials.

Twitter User Data (2020) An In-Depth Look at the Most Active Twitter User Data. https://sysomos.com/inside-twitter/most-active-twitter-user-data.

Vera-Burgos CM, Padgett DRG (2020) Using twitter for crisis communications in a natural disaster: hurricane Harvey. Heliyon 6:e04804. https://doi.org/10.1016/j.heliyon.2020.e04804

Wang RQ, Mao H, Wang Y, Rae C, Shaw W (2018) Hyper-resolution monitoring of urban flooding with social media and crowdsourcing data. Comput Geosci 111:139–147. https://doi.org/10.1016/j.cageo.2017.11.008

Wang Y, Hao H, Platt LS (2021) Examining risk and crisis communications of government agencies and stakeholders during early-stages of COVID-19 on twitter. Comput Hum Behav 114:106568. https://doi.org/10.1016/j.chb.2020.106568

World Cities Database (2020) https://simplemaps.com/data/world-cities.

Yaqub U, Chun SA, Atluri V, Vaidya J (2017) Analysis of political discourse on twitter in the context of the 2016 US presidential elections. Gov Inform Q 34:613–626. https://doi.org/10.1016/j.giq.2017.11.001

Yoo E, Rand W, Eftekhar M, Rabinovich E (2016) Evaluating information diffusion speed and its determinants in social media networks during humanitarian crises. J Oper Manag 45:123–133. https://doi.org/10.1016/j.jom.2016.05.007

Zhang YC, Sakhanenko L (2019) The naive Bayes classifier for functional data. Stat Probab Lett 152:137–146. https://doi.org/10.1016/j.spl.2019.04.017

Zvara Z, Szabo PGN, Balazs B, Benczur A (2019) Optimizing distributed data stream processing by tracing. Future Gener Comp Sy 90:578–591. https://doi.org/10.1016/j.future.2018.06.047