



Article

# An Emergency Event Detection Ensemble Model Based on Big Data

Khalid Alfalqi \* and Martine Bellaiche \*

Computer and Software Engineering Department, École Polytechnique de Montréal,  
Montréal, QC H3T 1J4, Canada

\* khalid.alfalqi@polymtl.ca (K.A.); martine.bellaiche@polymtl.ca (M.B.)

**Abstract:** Emergency events arise when a serious, unexpected, and often dangerous threat affects normal life. Hence, knowing what is occurring during and after emergency events is critical to mitigate the effect of the incident on humans' life, on the environment and our infrastructures, as well as the inherent financial consequences. Social network utilization in emergency event detection models can play an important role as information is shared and users' status is updated once an emergency event occurs. Besides, big data proved its significance as a tool to assist and alleviate emergency events by processing an enormous amount of data over a short time interval. This paper shows that it is necessary to have an appropriate emergency event detection ensemble model (EEDEM) to respond quickly once such unfortunate events occur. Furthermore, it integrates Snapchat maps to propose a novel method to pinpoint the exact location of an emergency event. Moreover, merging social networks and big data can accelerate the emergency event detection system: social network data, such as those from Twitter and Snapchat, allow us to manage, monitor, analyze and detect emergency events. The main objective of this paper is to propose a novel and efficient big data-based EEDEM to pinpoint the exact location of emergency events by employing the collected data from social networks, such as "Twitter" and "Snapchat", while integrating big data (BD) and machine learning (ML). Furthermore, this paper evaluates the performance of five ML base models and the proposed ensemble approach to detect emergency events. Results show that the proposed ensemble approach achieved a very high accuracy of 99.87% which outperform the other base models. Moreover, the proposed base models yields a high level of accuracy: 99.72%, 99.70% for LSTM and decision tree, respectively, with an acceptable training time.



**Citation:** Alfalqi, K.; Bellaiche, M. An Emergency Event Detection Ensemble Model Based on Big Data. *Big Data Cogn. Comput.* **2022**, *6*, 42. <https://doi.org/10.3390/bdcc6020042>

Academic Editors: Miltiadis D. Lytras and Andreea Claudia Serban

Received: 7 February 2022

Accepted: 14 April 2022

Published: 16 April 2022

**Publisher's Note:** MDPI stays neutral with regard to jurisdictional claims in published maps and institutional affiliations.



**Copyright:** © 2022 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

## 1. Introduction

Emergency events arise when a serious, unexpected, and often dangerous threat affects normal life. Both natural catastrophes (such as earthquakes, cyclones and floods) and man-made casualties (like fires, explosions and moving vehicle accidents) are examples of emergency events. Hence, knowing what is happening during and after an emergency is critical to mitigate the consequences of the event on humans' lives, infrastructures, environment and finances [1]. As a result, obtaining information rapidly and precisely is crucial for immediate responses. One of the most trustworthy sources of information remains social media platforms. Nowadays, social media platforms have emerged as prominent sources of real-time information regarding emergency events. Individuals at the incident location, and those nearby, may immediately upload valuable information on such platforms—the usage of which has exploded due to the prevalence of cellphones and internet connectivity [2]. It is thus not surprising that social networks are online communication platforms that many people use worldwide [3]. There are different types of social network platforms. One of which is Twitter: a platform that enable people to communicate by sending short messages of no more than 280 characters. Another platform

is Snapchat: an application that allows users to share their personal stories by sending, to their friends and relatives, images and videos which are removed 24 h after they are posted [4].

Emergency events create spikes in social networks' information rates which then requires fast data processing to handle such events. During emergency events, the information on social network platforms generates massive data that cannot be handled by traditional data processing methods, thus requiring the use of big data processing techniques [5]. Big data is a collection of more comprehensive methods that seek to study, save, and supervise massive data within an adequate time frame [6]. It provides decision-makers with unique capabilities to analyze and understand a set of circumstances to ensure they choose the most appropriate decisions when emergencies strike [7].

A robust emergency event detection system is needed to limit the disastrous impacts of such events. Social networks and big data play a critical role to expedite the discovery of emergency events. Leveraging data from social networks can help to manage, monitor, analyze and detect emergency events. However, when gathering real-time data from social networks, the main research gap concerns identifying the exact location of the emergency events.

Several research articles attempted to tackle this drawback by presenting promising suggestions. Alomari E et al. [8] utilized Twitter, big data, and social media platforms to develop an auto labeling system for traffic-related event detection. Their approach detected traffic events automatically from tweets using distributed machine learning and Apache Spark. However, this paper does not suggest the use of Snapchat maps to detect the exact location of traffic-related events. Instead, it presents multiple methods to extract the event location: from the tweet's message, from the hashtag if it mentions a specific place, from a predefined list of account names specialized in posting about traffic conditions, or by checking coordinates of a tweet. In addition, while this paper focuses only on traffic-related event detection, our approach provides a broader overview and can detect different types of emergency events.

Furthermore, the novelty of our approach is using the Snapchat hotspot maps as a reliable source to detect precise emergency event locations without prior knowledge of the location or the emergency event itself. Accordingly, our approach first identifies the emergency event locations through the Snapchat heat map. Then, after detecting the exact location of such events, we leverage Twitter stream API to collect the tweets in the identified locations in near real-time.

This paper proposes a new and efficient emergency event detection ensemble model (EEDEM) to address the aforementioned research gap. The main objective of this model is to find the exact location of an emergency event by utilizing the collected data from social networks (SN) with the integration of big data (BD) and machine learning (ML) [9].

The suggested approach aspires to acquire the best performance in terms of accuracy while reducing the processing time to fulfill the goals of this paper. More particularly, the contributions of this paper can be stated as follow:

- Propose a data collection model to collect the data from social networks by leveraging the Snapchat API and Twitter API.
- Develop a new and efficient emergency event detection ensemble model (EEDEM) by utilizing the collected data from social networks (SN) with the integration of big data (BD) and machine learning (ML).
- Identify the exact location of an emergency event by observing the Snapchat hotspot map to detect any potential emergency events and then utilizing Twitter API to collect the dataset of such events.
- Propose an ensemble voting model to evaluate the performance of our approach and improve the accuracy of the base models.

The rest of the paper is organized as follows: Section 2 "Related Works" explains the related works of three different research domains studied in this research. Section 3 "Background" presents a brief background of Big data layer architecture, the Apache spark streaming and the machine learning models. Section 4 "Methodology" describes the

research methodology steps, techniques and models used for the experimentation. Section 5 “Evaluation approach”, provides the details of the selected case study which highlights the reasons behind this selection and experimental environment setup, evaluation metrics, the performance analysis for our proposed approach and discusses study results. Finally, Section 6 “Conclusion and Discussion”, concludes the research work and provides future research directions.

## 2. Related Work

This section illustrates a literature review pertaining to three research fields studied in this paper. First, we shed light on up-to-date studies that address machine learning classification methods in emergency events detection using social networks. Then, we elaborate on current big data methods in emergency events detection using social networks. After that, we address the recent usage of Snapchat in emergency events detection.

### 2.1. Machine Learning Classification Methods in Emergency Events Detection Using Social Networks

Septianto et al. [10] presented an innovative idea to collect traffic flow data from Twitter in Jakarta city utilizing a machine learning classifier called ‘Nave Bayes Classifier (NBC)’. They built a software that can display Jakarta’s traffic conditions in real time and then sort the data to be injected into Google Map. In addition, they utilised an illustrative forecasting models relying on up-to-date data to oversee Jakarta’s roads during designated times using NBC to encourage drivers to take different paths instead of heading towards traffic jams.

Toujani et al. [11] proposed a unique approach that identifies event information following a natural catastrophe by employing social networks as the main reference. Then, they cluster individuals into levels founded on the phase of hazard. This clustering procedure is profitable for reporters to simplify the method of deriving information in emergencies. Furthermore, they utilized the fuzzy theory steps on these events to promote clustering excellency and eradicate opacity in the collected data.

Kumara et al. [12] suggested a procedure to bolster the after-disaster response methods by specifying the right location and the type of crisis founded on three stages. Nevertheless, this procedure has some limitations and requires further optimization.

The above approaches deliver ambitious perspectives on detecting emergency events by applying machine learning to the data from social networks. However, none of the above-mentioned papers have used big data technologies.

### 2.2. Big Data Techniques in Emergency Events Detection Using Social Networks

Similar works have used big data with data collected from social networks. Hagras et al. [13] employed the Latent Dircherilet Allocation (LDA) topic analysis method to categorize and assess tweets correlating with the Japan Tsunami. Yet, this method can be expanded by using additional datasets to boost the accuracy and expedite the processing in real-time.

Ragini et al. [14] introduced a strategy for crisis governance by incorporating emotion analysis and ML algorithms. They used a Support Vector Machine (SVM) to analyze the data obtained from Twitter via human sentiments, both optimistic and pessimistic, and then categorize them based on their necessities. Although the suggested strategy simplifies emergency crews’ mission to recognize the disastrous case and take suitable measures directly, it has some problems in employing social network data for crisis mitigation—specifically the unclarity in obtaining crisis data through various sources and the lack of the suitable criterion. However, these problems can be expiated by gathering data from diverse venues to categorize the data efficiently and enhance precision.

Salas et al. [15] employed Apache Spark to gather tweets and build a model using the SVM classification approach to classify them. Then, they used Name Entity Recognition (NER) and Wikipedia to obtain the location information.

Lau, R.Y. et al. [16] suggested the Latent Dirichlet Allocation (LDA) to classify datasets retrieved from Twitter and Weibo. Then, they used SVM, KNN and NB to classify the data by utilizing Spark MLlib. In addition, they created a classification ensemble technique to automatically detect specific crowded traffic events.

Bhuvaneswari et al. [17] presented an end-to-end framework to enhance emergency events detection rate by deploying topic modelling to the data collected from Twitter stream API. They built their real-time framework based on Apache Spark and the LDA technique. The event detection approach reached a high accuracy of %96 and the event could be detected after only 75–100 milliseconds of its occurrence.

Alomari E et al. [18] utilized Twitter and the big data approach to design a dictionary that facilitates the detection of traffic events in Saudi Arabia. In addition, they used sentiment analysis based on the lexicon approach for Arabic and Saudi dialect words.

The aforementioned studies present novel ideas that apply big data techniques to the data collected from social network platforms following emergency events. However, none of the previous papers consider using both Twitter along with Snapchat hotspot maps to pinpoint the exact location of emergency events. By leveraging the Snapchat hotspot map and Twitter, and combining it with big data techniques, we can precisely identify the locations of emergency events and thus accelerate the detection processing.

### 2.3. Using Snapchat in Emergency Events Detection

On the other hand, some recent work proposed using Snapchat maps to solve similar issues. For example, Al-ghamdi N et al. [19] suggested using Snapchat heat maps to analyze the crowd behavior at the Grand Holy Mosque in Mecca, Saudi Arabia. Furthermore, a similar approach was proposed by Alageeli N et al. [20] to analyze the sentiment of Riyadh's season visitors. They predefined the locations of three major events in Riyadh's season and then used Snapchat maps to extract crowd behavior patterns. However, these studies utilized the Snapchat map as a social sensor to track and analyze the visitors' activities. Moreover, Juhasz et al. [21] utilized Snapchat and different social network platforms to compare their activity patterns. Likewise, Lamba et al. [22] crawled the Snapchat map API to collect data in predefined cities to annotate them into dangerous driving or not dangerous driving.

Even though aforementioned papers utilized the Snapchat map in their approaches, they used it in predefined locations. Unlike our approach, we propose using the Snapchat heat maps as a reliable source of precise emergency event location detection without prior knowledge of the location.

## 3. Background

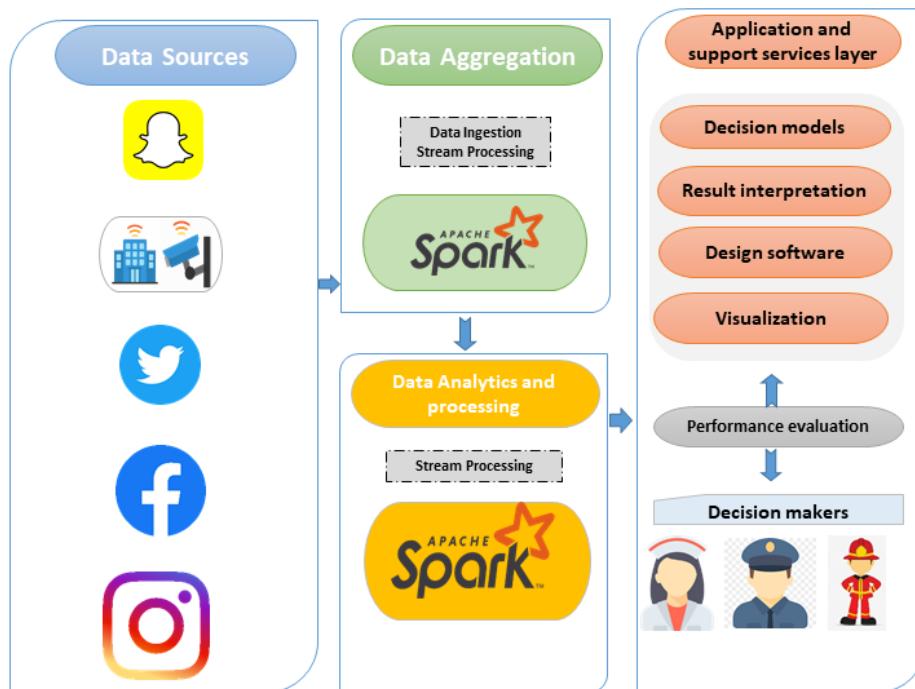
This section provides a brief overview of relevant background concepts inherent to this study, which include big data, Apache Spark and machine learning algorithms.

### 3.1. Big Data

Big Data architecture facilitates the data flow development to suit both batch processing and stream processing. This architecture comprises four layers that promote safe data flow. These layers are merely a method of categorizing components that conduct specific tasks. Figure 1 shows the overall layers of big data.

- The data resource layer: This layer is responsible for managing and collecting all potential data sources to strengthen the scheme [6].
- The data aggregation layer: Adding multiple heterogeneous data sources inevitably improves the efficiency and usefulness of the framework to allow taking the most appropriate decisions. However, it may also raise framework instability and add certain difficulties. The main responsibility for this layer consists of gathering all the data from different channels before injecting it into a multi-source database [23].

- The data analytic and processing layer: The main data processing layer includes a range of specific resources to acquire, store, retrieve, search and analyse data. A mix of various big data analytic platforms can be used to build a near real-time system.
- The application and support services layer: An integrated web-based computer framework will offer decision-makers (e.g. emergency services, public safety staff, police, fire departments) the necessary information. The main goal of this layer consists of enhancing the decision-making cycle with a continuous stream of the required information, as well as the latest trends for further perspective [23].



**Figure 1.** Overall of big data layers.

### 3.2. Machine Learning

Machine learning is described as the phases that involve learning given data in order to obtain adequate knowledge and then create the targeted outcome. Machine learning can be supervised, unsupervised or semi-supervised [24]. The following algorithms are used in our approach: long short-term memory (LSTM), decision tree (DT), support vector machine (SVM), K nearest neighbor (KNN) and Naïve Bayes (NB).

- Long Short-Term Memory (LSTM):** is a kind of neural network layer that is usually adopted in Recurrent Neural Networks (RNN) which overcomes RNN's vanishing gradient problem. A cell, an input gate, an output gate, and a forget gate comprise a standard LSTM unit. The cell retains data for unlimited time frames, and the three gates monitor data transmission to and from the cell [25].
- The decision tree (DT):** is a kind of supervised classifiers technique used for classification and regression. Due to its strength and effectiveness, it became a standard tool for machine learning and big data problems. It contains three types of nodes: decision nodes, chance nodes and end nodes [26].
- Support vector machine (SVM):** is used for both classification and regression problems. The goal of SVM is to create the most appropriate hyperplane and split the dataset into two classes. SVM is a robust machine learning technique which can be deployed in a broad range of subjects such as linear, nonlinear classification, and regression. It is among the most widely used machine learning techniques [27].
- K Nearest Neighbor (KNN):** is a machine learning approach that matches raw data to data that has already been categorized with a stipulated training data class. The

disparity between the raw and trained data is used to assess this comparison. The nearest neighbors are detected with the data set with the lowest distance estimates [28].

- **Naïve Bayes (NB):** is beneficial to define data sets with a significant volume of knowledge since it performs quickly and is easily enforced. The NB algorithm is a Bayesian classification approach; hence, it is founded on the Bayes probability concept and generates probability tables for each variable individually [29].

#### 4. Research Methodology

In this section, to tackle the research gap and fulfill our goals, the core aspects of this research must be elaborated by describing the required steps. First, Apache Spark Streaming is integrated to empower stream processing. Second, a data collection model is proposed to collect and label “Snapchat” and “Twitter” data to prepare the dataset. Third, data preprocessing steps are applied so that the machine learning algorithms can recognize the data. Fourth, feature extraction algorithms are utilized to help construct machine learning classifiers. Fifth, machine learning classifiers for emergency event detection models are proposed to facilitate the classification process. Finally, we deploy an ensemble based emergency events detection approach and conduct a performance evaluation to depict the detection accuracy of the base models and the proposed voting ensemble model.

##### 4.1. Build Big Data Processing—Apache Spark Streaming

This research is developed on the Apache Spark platform, which empowers the stream processing of different data sources. Spark streaming is widely used to process near real-time data from numerous sources. Spark streaming is an enhancement of the main Spark API that provides extensible, elevated, fault-tolerant continuous data stream processing. Spark collects live, raw data feeds and separates them into packets, which are subsequently processed by the Spark engine to provide the latest version of results. Its fundamental concept is the Discretized Stream, which depicts a flow of data separated into small batches. DStreams are founded on the basis of Resilient Distributed Datasets (RDDs). This allows Spark streaming to collaborate with other Spark components including MLlib and Spark SQL [30]. Spark ML and Spark SQL are the primary libraries that have been used. Spark.ML, an extension package integrated into core Spark, provides a higher application programming interface stacked on top of data to develop and enhance machine learning [31]. Furthermore, the python built-in libraries were used to process the dataset.

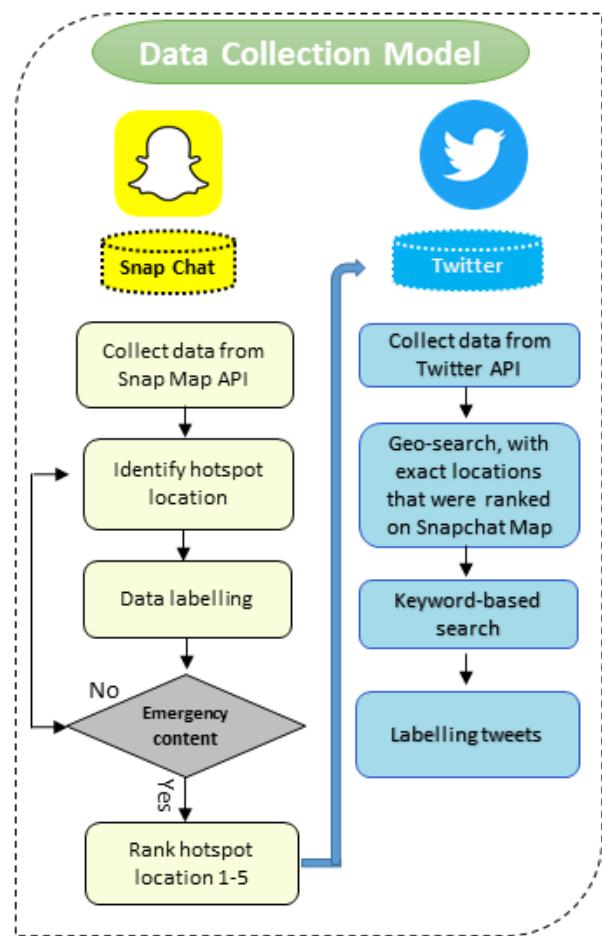
##### 4.2. Data Collection Model

The most popular social network platforms Twitter and Snapchat are used in the proposed emergency event detection model. The collected dataset can take the shape of pictures, videos or texts. Figure 2 shows the proposed data collection model for both Snapchat and Twitter platforms.

###### 4.2.1. Snapchat Data Collection

To identify the precise location from the Snapchat platform, we need to develop a classification model using snaps from the Snapchat Map. This interactive map enables us to share our location with others. It is a distinctive characteristic whereby any content can be publically available, and immediate updates are generated while Snapchat is accessible. Furthermore, content posted on a Snapchat Map is geo-tagged and indicated in a precise area [22]. Thus, by monitoring the Snapchat Map, any hotspot area on the map could indicate an emergency event zone [32]. Accordingly, snaps in that area are scrapped to be labelled and marked as either emergency event-content or non-emergency event-content. Then, each hotspot location is ranked based on the highest number of snaps in that location, to prioritize the locations. For this purpose, we developed a wrapper for Snap Map’s API to create Node.js JavaScript that collect snaps posted at the precise hotspot location.

The collected snaps can be image or video; therefore, the images and the videos will be classified using an image classification technique.



**Figure 2.** Data collection model.

#### 4.2.2. Twitter Data Collection

After identifying and ranking hotspot zones on the Snapchat Map, the top five hotspots are used to collect Twitter data. The benefit of this step is to narrow down the tweets' geo-location search based on the high-ranked hotspot areas and reduce processing times [21]. In this research, tweets will be collected through the Twitter Search API; geo-tagged tweets are collected in the exact locations that were identified and ranked on Snapchat Map. To illustrate, the collection process is carried out with the following steps:

1. Geographical Search “Geo-Search”: it searches within a bounding box of the most highly ranked locations from a Snapchat Map, to collect all tweets within the identified zone.
2. Keyword-Based Search: it searches, within the collected tweets, keywords and hashtags which are considered relevant to the emergency event. This step is intended to extract tweets that include at least one keyword or hashtag connected to the emergency situation to minimize non-relevant data [12].
3. Labelling Tweets: according to the type of emergency event (by matching keywords and hashtags data to the emergency event detected on the Snapchat), tweets are associated with one of two labels: related or not to the case study [33,34].

#### 4.3. Data Preprocessing

The collected tweets are stored in MongoDB, one of the most prominent document-oriented databases [35]. Thus, we utilized MongoDB to store the tweets acquired through the Twitter API before it is exported to the Apache Spark Data Frame. The datasets is preprocessed at this stage to ensure machine learning algorithms can recognize the data in the next steps. Specific processing tasks are conducted on the acquired Twitter datasets. The main preprocessing steps used are: tokenization, normalization, lower case text conversion,

as well as the removal of stop words, punctuation, white space and repeated characters. Algorithm 1 shows the preprocessing steps

---

**Algorithm 1:** Prepossessing
 

---

```

Data: TweetList; stopWordsList; punctuationList;
Result: Normalized tokens
1 spark ← SparkConnector()
2 tweets_DF['fullText'] ← spark.load(TweetList)
3 for tweet in tweets_DF do
4     clean_Tweet ← remove_repeating_char(tweet)
5     clean_Tweet ← casefoldingText(clean_Tweet)
6     clean_Tweet ← remove_mentions(clean_Tweet)
7     clean_Tweet ← remove_links(clean_Tweet)
8     clean_Tweet ← remove_RTs(clean_Tweet)
9     clean_Tweet2 - []
10    for char in clean_Tweet do
11        if char not in punctuation List AND not char.isdigit() AND not char is English
12            Char() then
13                | clean_Tweet2.append(char)
14            end
15    end
16    tokens_List ← tokenizingText(clean_Tweet2)
17    normalizedTokens - []
18    for token in tokens_List do
19        if token not in stop Words List then
20            | normalizedTokens.append(token)
21        end
22    end
23    normalizedTokens ← tweets_DF ['Tokens']
end
```

---

#### 4.4. Feature Extraction

Feature extraction is an essential component to develop machine learning algorithms. Its objective is to transform raw data into controllable parameters (a collection of features) while retaining data accuracy. Additionally, it empowers us to choose the essential features while developing a classifier model. Different feature extraction techniques such as TF-IDF and Bag of Words [36] were used in this research, are defined as follows:

1. **TF-IDF:** It is an acronym for Term Frequency-Inverse Document Frequency. It considers the number of occurrences of a phrase, in a document, to determine its significance [37].
2. **Bag of Words:** The full text is displayed as a list of words in the Bag of Words (BOW) feature. The frequency of each word is employed as a feature while training an algorithm. Substantial preprocessing is required to ensure that the bag of words feature provides acceptable accuracy. The preparation stage excludes extraneous words from the database, allowing the classifier to avoid any redundant features throughout the process of learning [36].

#### 4.5. Applying Machine Learning Classifiers

According to the rules developed throughout the learning or training stages, classification in machine learning helps predict a group or class of an input dataset. To filter tweets into related or non-related emergency events, a classifier is built using ML classification algorithms [36]. We have a balanced dataset as the number of samples for the negative class (not related to the case study) is 49.6%, while the positive class (related to the case study) is 50.4%.

Also, we need to tune the parameters after data processing and feature extraction to reach the optimal performance modelling. Grid search is a method to find the optimal tuning parameter values. To achieve this, a collection of tuning-parameter possibility values must be defined and assessed [18]. Table 1 shows the hyper-parameters of the models.

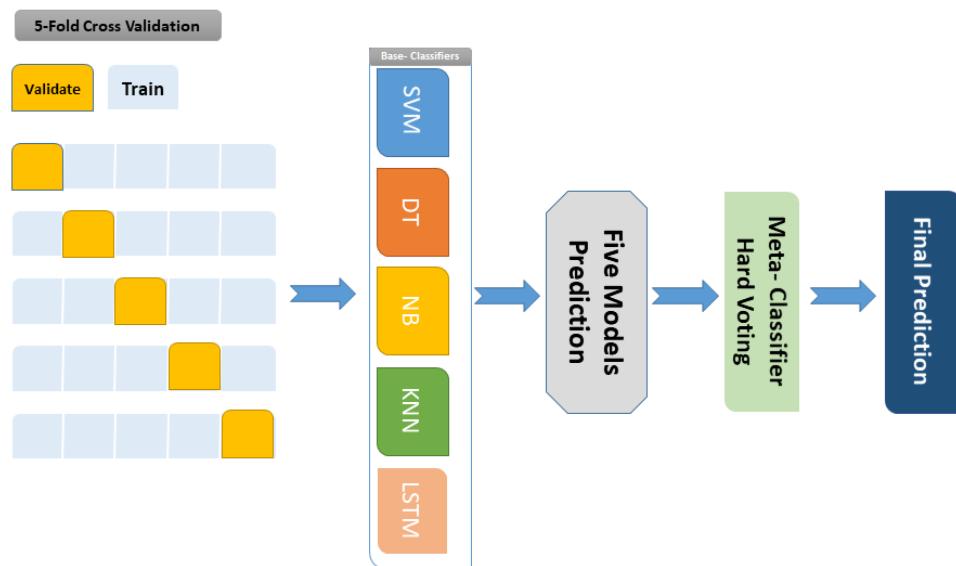
**Table 1.** Hyper-parameters of the models.

Models	Hyper-Parameters
LSTM	dropout_rate = 0.2, embed_dim = 32, hidden_unit= 16 , optimizers = RMSprop, output = 3 neurons, activation = tanh + linear , epochs = 10, batch_size = 128
SVM	C = 1, gamma = 0.5 , kernel = rbf, probability is False, shrinking = True, tol = 0.001
KNN	n_neighbors = 8.
Naive Bayes	var_smoothing = $1 \times 10^{-8}$
Decision Tree	criterion = 'entropy', splitter = "best"
The Proposed Ensemble	Estimators = LSTM, SVM, KNN, Naive Bayes and Decision Tree, Voting = hard

Then, a Spark ML library was utilized to develop and train the models. In order to classify the dataset, five base-models were built with LSTM, SVM, Naive Bayes, decision tree and K Nearest Neighbor algorithms [38].

#### 4.6. Proposed Voting Ensemble for Emergency Event Detection Model

In this step, we use the five trained models combined together to deploy the voting ensemble approach. We applied the 5-Fold cross-validation to split the dataset and train the models, with data split into five parts. Each round uses 5-1 parts to train the classifier and the residual part as a validation set. The 5-Fold cross-validation is depicted in Figure 3.

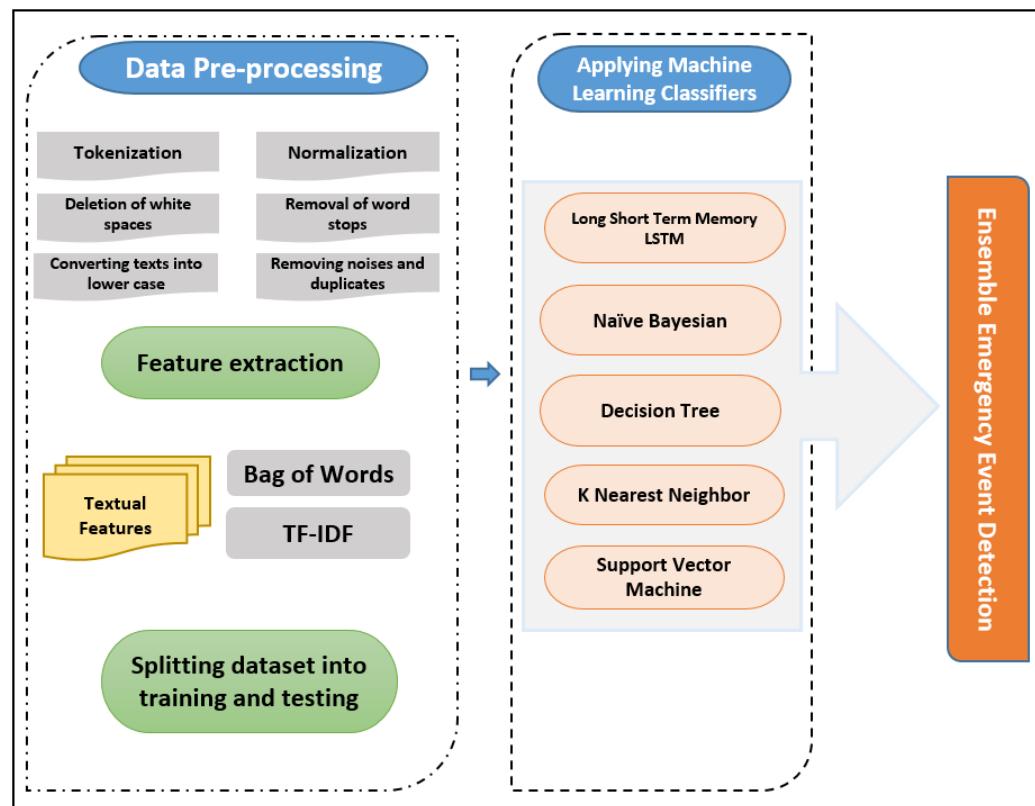


**Figure 3.** The 5-Fold cross validation.

The evaluation is based on a majority voting (Hard Voting) approach of the five base classifiers (LSTM, SVM, Naive Bayes, decision tree, and K Nearest Neighbor). As a result, the proposed hard voting ensemble model combines five base classifiers predictions to give the final prediction [39].

After that, we will conduct a performance evaluation to depict the detection accuracy of all models.

Figure 4 shows the proposed steps in Sections 4.3–4.6 of the emergency event detection ensemble model.



**Figure 4.** The proposed steps of the emergency event detection ensemble model.

## 5. Evaluation Approach

This section provides the details regarding the experimental environment setup, the evaluation metrics, and the performance analysis for the proposed approach.

### 5.1. Case Study

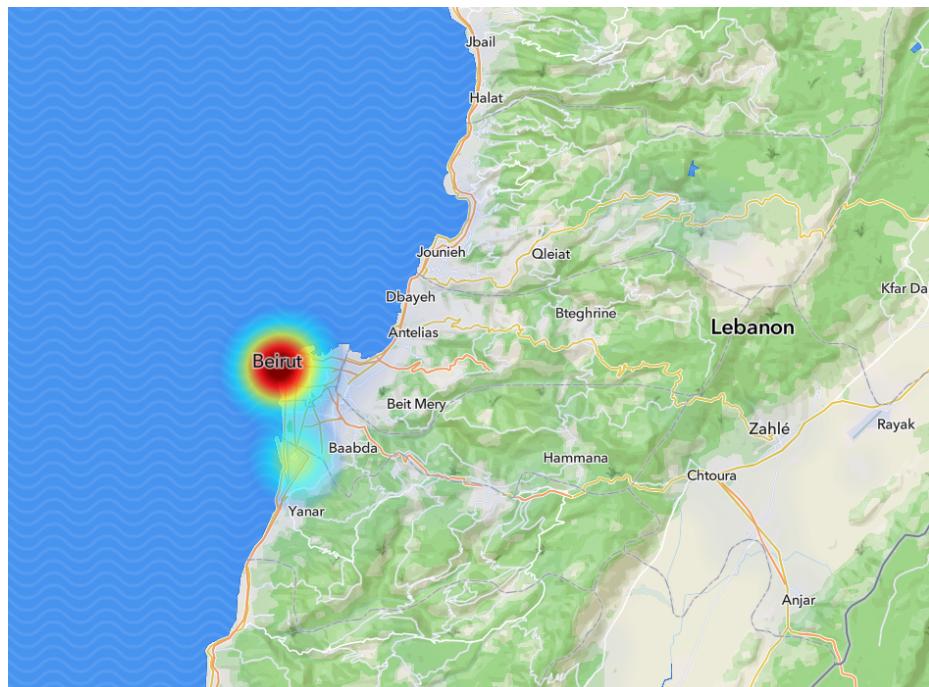
#### The Beirut Port Explosion

On 4 August 2020, a major explosion, one of the most enormous non-nuclear blasts in history, damaged a large section of Beirut. Roughly 2750 tons of ammonium nitrate (which equals to 1.1 kilotons of TNT) stockpiled in a facility near the port detonated, inflicting significant damage throughout the capital. The shock was felt in countries as far away as Turkey, Syria and Cyprus. The explosion killed at least 200 people, wounded over 6000 and an estimated 300,000 inhabitants became homeless after the \$15 billion in property destruction [40]. Figure 5 shows the words cloud for the tweets related to the case study.

Using the developed Node.js Google function wrapper, the search for the hotspot location with a Snapchat map revealed more than five hotspots on 4 August 2020. However, based on the ranking criteria of the Snapchat data collection in Section 4.2.1, the hotspot location which has the most snaps at that time was the Beirut port explosion. Figure 6 shows the location of Beirut Port explosion on the Snapchat hotspot map. After Identifying the location of this emergency event, we collected snaps in this location and then classify them using Convolutional Neural Network(CNN) with the help of the transfer learning approach; by utilizing the pre-trained model as a feature extractor. The pre-train model is ResNet50, a variant of the ResNet model that has 48 convolution layers and 1 MaxPool, and 1 Average Pool layer. Then we have modified the output dense layer to suit our case study. Table 2 present the structure of this model.



**Figure 5.** Words cloud of tweets.



**Figure 6.** Explosion location of Beirut Port.

**Table 2.** Structure of the CNN model.

Models	Hyper-Parameters
CNN	<pre> optimizer='adam',      loss='binary_crossentropy',      metrics = ['accuracy'], activation = 'sigmoid' input_2 (InputLayer), Output Shape [(None, 224, 224, 3)] resnet50(Functional), Output Shape (None, 7, 7, 2048) global_average_pooling2d (GlobalAveragePooling2D) Output Shape, (None, 2048) dense (Dense), Output Shape (None, 6) </pre>

### 5.2. Experimental Setup

Our models were built with different Python packages, including Scikit-Learn, Keras and TensorFlow. Furthermore, experiments and results were compiled and the TensorFlow environment was implemented on Google Colabotary, with 16GB RAM and a 108GB disk. The experimental platform is a laptop with 11th Generation Intel® Core™ i7 processor

NVIDIA® GeForce® MX450 (2 GB GDDR6 dedicated) 16 GB RAM; 64-bit operating system, x64-based processor, Windows 11.

To evaluate the performance of our approach, the dataset was first collected. The approach used the most common social networks platforms, Snapchat and Twitter, to collect the data in order to identify the location of an emergency event without any prior knowledge. Therefore, after identifying the exact location of the emergency event in Beirut from the Snapchat map, tweets were collected from Twitter API on 4 August 2020. The geographical search was applied to collect all tweets within the chosen area.

After all tweets in the location of this very case study were collected, we automatically applied the selected keywords search to retain solely the tweets that include keywords related to the case study using the Twitter API. The following specific keywords were used: "Lebanon Explosion, Explosion in Beirut, Huge explosion Lebanon, Beirut port exploded, Beirut Blasts, Lebanon Blasts, Disaster in Beirut, Disaster in Lebanon and Beirut". The Twitter API collects tweets and formats them using JavaScript Object Notation (JSON). Every tweet has a distinct Identifier, the actual text, a tweet timestamp (indicating the time tweet was posted), and numerous additional attributes including "username" and "location".

Finally, a total of 50,244 tweets were collected. However, since the obtained dataset contained both English and non-English tweets, only the English tweets were kept for the experiment. Moreover, replies to tweets, retweets as well as quoted tweets were also removed. Consequently, after deleting the aforementioned tweets, the total number of tweets went down to 20,144.

Furthermore, the dataset was labelled as either Beirut-related or not related using python script. After we collected the dataset and then filtered it using the related keyword search in the previous step, we stored the filtered tweets that contain keywords related to our case study in a separate JSON file. Then we labelled them as related to our case study using our script, which will add a new column to the dataset and assign the label to all tweets in this file. Moreover, we created another JSON file and stored the tweets that were excluded from the first JSON file, which did not contain any keyword related to our case study. Then we labelled them as not-related to our case study using our script, which will add a new column to the dataset and assign the label to all tweets in this file. Subsequently, both files were merged into a single JSON file to prepare our dataset for further processing. Furthermore, we applied the 5-Fold cross-validation approach to split the dataset. The dataset was randomly partitioned into 5-sized subsamples using a K-fold cross-validation method. One k section was utilized for validation assessment, while the residual k-1 sections were used for classifier training. In the future, we plan to improve our study by the using the 10-fold cross-validation approach.

Moreover, the preprocessing steps were applied to extract the most important features from the data to reduce the computational overhead. Then, detection models were built using the trained dataset. Finally, the trained models were integrated into the Apache Spark streaming and used the test dataset to evaluate the performance of the proposed approach.

### 5.3. Performance Evaluation

The evaluation metrics play a substantial aspect in achieving the desired classification during the training stage. Furthermore, collecting appropriate measurement criteria is crucial for differentiate and acquire the ideal classifier [41]. After applying the machine learning classifiers mentioned above for emergency events detection, the accuracy of the proposed model was tested through diverse performance evaluation criteria such as [42]:

- **Precision (P):** It is used to subtract the number of accurately predicted positive patterns from the total number of expected positive patterns in a positive category.

Precision can be calculated using the following equation.

$$\text{Precision} = \frac{TP}{TP + FP} \quad (1)$$

- **Recall (R):** It is used to determine the percentage of correctly classified positive patterns. Recall can be calculated using the following equation.

$$\text{Recall} = \frac{TP}{TP + FN} \quad (2)$$

- **F-Measure (FM):** Also known as the F1-score, which is a metric to determine accuracy on a given data. It is used to examine binary classification algorithms that label data as "positive" or "negative". The F1-score can be calculated using the following equation.

$$F1 = 2 \times \frac{(precision \times recall)}{precision + recall} \quad (3)$$

- **Accuracy (ACC):** It is used to calculate the proportion of accurate classification compared to the total number of instances examined. Accuracy can be calculated using the following equation.

$$\text{Accuracy} = \frac{(TP + TN)}{TP + FP + TN + FN} \quad (4)$$

True Positive (TP): It refers to the model's correct classified number of instances [43]. False Positive (FP) refers to the number of negative instances identified incorrectly as positive instances [43].

False Negative (FN): refers to the number of positive instances identified incorrectly as negative instances [43].

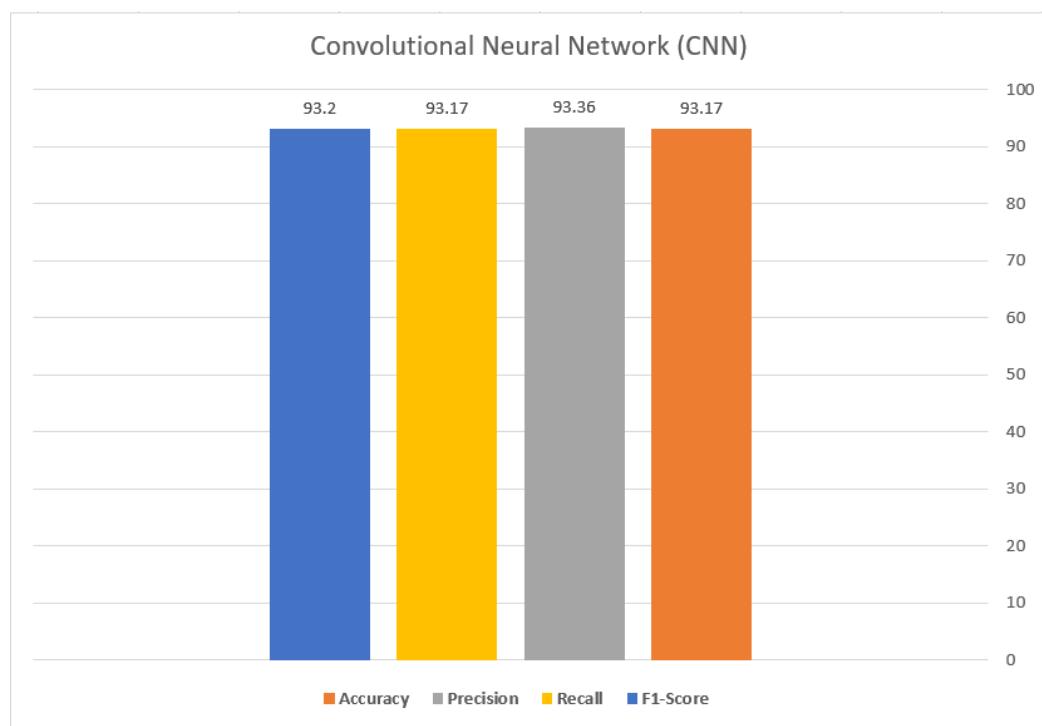
True Negative (TN): The true negative values refer to the number of negative instances that were correctly classified by the model [43].

- **Throughput:** It represents the number of results delivered over a given period of time. In the framework of this study, this is measured in units of flow.

#### 5.4. Results

The experiments to evaluate our approach were conducted locally, in the environment set up described in the previous subsection. The proposed Snapchat classification model achieved a high accuracy of 93.17%. Figure 7 shows the performance evaluation of the Snapchat classification model.

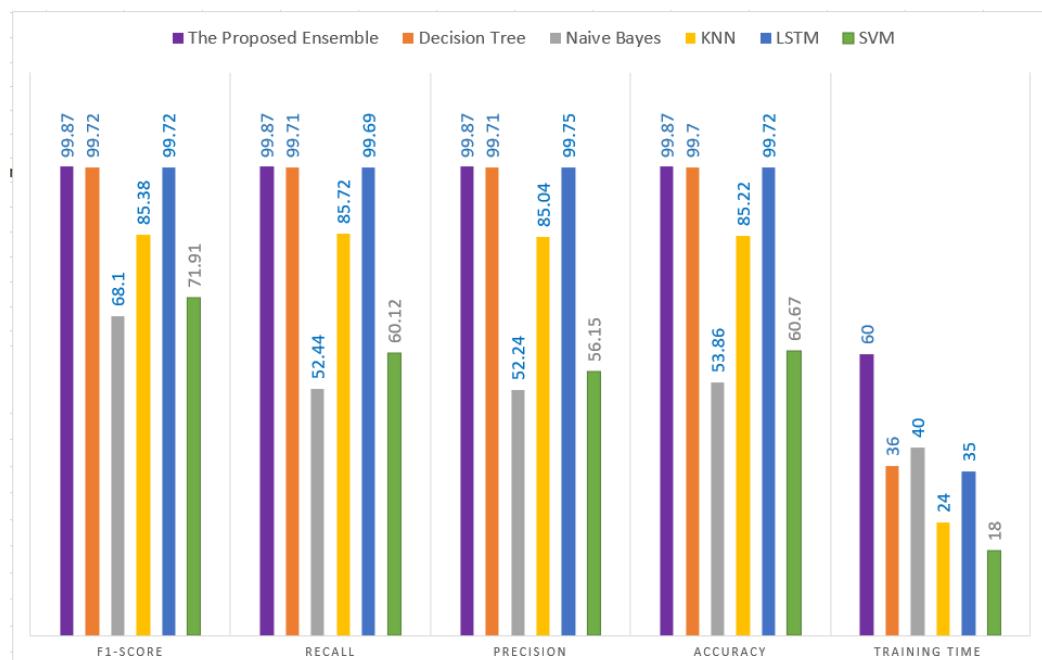
Also, Table 3 shows the results of the Twitter classification models. The experimental results proved that our proposed ensemble approach achieved a very high accuracy of 99.87%, outperforming the other base models. In addition, it can be noted from Table 3 that three base models (LSTM, decision tree and KNN) achieved high accuracy of 99.72%, 99.70% and 85.22%, respectively. Table 3 shows that the LSTM base model achieved higher results than the other base models except for recall, where decision tree outperforms it with 99.71%. LSTM achieved 99.72% for accuracy, F1-score, 99.72% for precision and 99.69% for recall. The difference between the results achieved by LSTM and decision tree is negligible. However, the LSTM performed better in training time (35 s). Even though KNN required less training time (24 s) than decision tree (36 s), decision tree outperformed it with 99.71% for precision and recall, 99.72% for f1-score and 99.70% for accuracy. Moreover, SVM required less training time (18 s.) among all other base models, but it achieved a very low accuracy of 60.67%. In addition, Table 3 shows that Naive Bayes achieved the lowest accuracy among all base models with the most training time needed. Figure 8 shows the performance evaluation for each model separately.



**Figure 7.** The performance evaluation of the Snapchat classification model.

**Table 3.** Results of Models Classification.

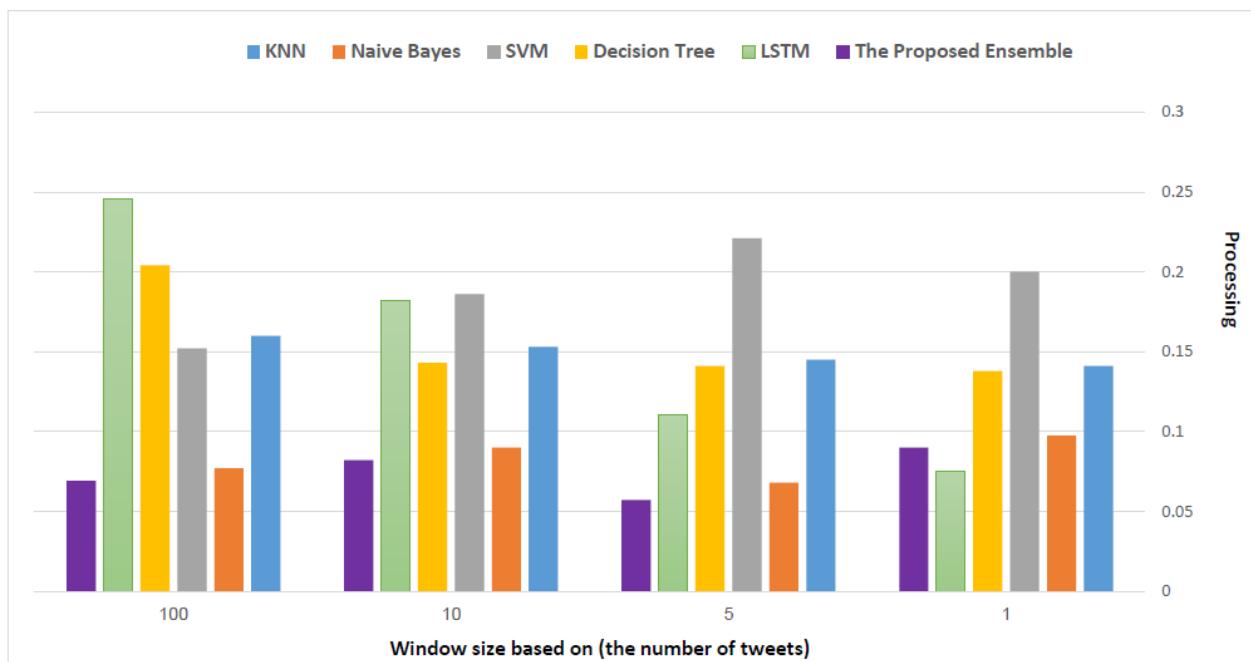
Model	Precision	Recall	F1-Score	Accuracy	Training Time (s)
LSTM	99.75	99.69	99.72	99.72	35
SVM	56.15	60.12	71.91	60.67	18
KNN	85.04	85.72	85.38	85.22	24
Naive Bayes	52.24	52.44	68.10	53.86	40
Decision Tree	99.71	99.71	99.72	99.70	36
<b>The Proposed Ensemble</b>	<b>99.87</b>	<b>99.87</b>	<b>99.87</b>	<b>99.87</b>	<b>60</b>



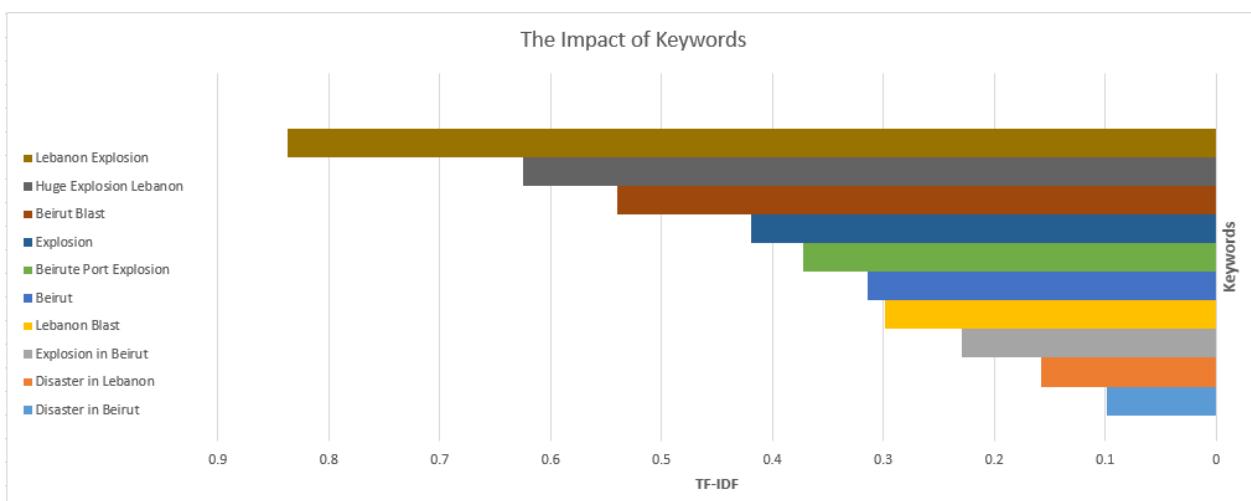
**Figure 8.** The performance evaluation of each model separately.

The models were integrated into Apache Spark streaming to assess the processing time using a test dataset. As shown in Figure 9, different time windows were used to stream the data from 1 to 100 s. Obviously, the Naive Bayes model processes the data much faster than all other base models in all window frame times—except for window 1. Indeed, in window 1, the LSTM model takes less time to process the data. However, as the window size increases, the processing time of the LSTM model decreases. Moreover, as we can observe from Figure 9, the proposed ensemble model outperforms the other models except in window 1. However, as the window size increases, the processing time of the proposed ensemble model increases.

By analyzing the impacts of the selected keywords which were applied to the collected tweets, Figure 10 shows that the keyword “Lebanon Explosion” yields a high score of 0.837. The whole list of the keywords are presented in Figure 10.



**Figure 9.** Processing time of models classification based on window size.



**Figure 10.** The impacts of the selected keywords.

## 6. Conclusions and Discussion

This paper indicated that it is necessary to have an accurate event detection model to react quickly and efficiently to emergencies. This paper's main contribution lies in the fact that it uses a Snapchat map to pinpoint the precise location of an emergency event. Monitoring the Snapchat map allows detecting the hotspot locations and analyzing them by labelling every snap in that particular hotspot as either associated or not to an emergency event. Then, a data collection model was proposed to collect the data from the Twitter API. Subsequently, the preprocessing steps were applied so that the machine learning algorithms could recognize the data, and feature extraction algorithms were utilized to help construct a machine learning classifier. Then, five machine learning base classifiers for emergency event detection models are proposed to construct our proposed ensemble approach. Finally, a performance evaluation of the proposed models was conducted to evaluate the model in terms of accuracy detection achievement.

Apache Spark streaming was integrated into our data to assess the model processing time using the aforementioned evaluation metrics. As a result, the proposed ensemble approach achieved accuracy levels of 99.87% which outperformed the other base models. Moreover, LSTM and decision tree yield 99.72%, 99.70% for accuracy, respectively, with an adequate training time. Therefore, the LSTM model is considered superior to the other base models. However, in our approach we applied only one deep learning model "LSTM" to the collected dataset from Twitter. Because the LSTM model is very effective in the sequential problem of NLP, also, as sequential data such as text and time series are the backbone of the Twitter platform, we utilized LSTM in this study. Moreover, even though the ensemble learning technique is beneficial for improving quality, accuracy, and decreasing predictions errors compared to the single learning machine model, the complexity of the learning process has significantly increased, which eventually led to the increase in the training time of the model.

Future work expected that the performance would improve if more feature extractions and more deep learning classifiers were added to the approach. Furthermore, our system shows good evidence of using the Snapchat hotspot map as a reliable source to detect the location of emergency events. However, to improve the scope and profundity of the work concerning what can be detected using the Snapchat platform, we need to enrich our approach. Therefore, we plan to enhance this work by employing computer vision methods to the collected dataset from Snapchat API. Accordingly, by applying such techniques, we can have a broader analysis of the dataset to detect emergency events and assess the severity of such events. Moreover, our approach focuses on the data collected from social network platforms without considering other data sources. Therefore, we will boost our work by exploring IoT sensors and satellite imagery sources.

**Author Contributions:** Conceptualization, K.A.; methodology, K.A.; software, K.A.; validation, K.A.; formal analysis, K.A.; investigation, K.A.; resources, K.A.; data curation, K.A.; writing original draft preparation, K.A.; writing review and editing, K.A.; visualization, K.A.; supervision, M.B.; project administration, K.A. and M.B.; funding acquisition, K.A. and M.B. All authors have read and agreed to the published version of the manuscript.

**Funding:** This research received no external funding.

**Institutional Review Board Statement:** Not applicable.

**Informed Consent Statement:** Not applicable.

**Data Availability Statement:** Not applicable.

**Conflicts of Interest:** The authors declare no conflict of interest.

## References

1. Rong, Y.; Liu, Y.; Pei, Z. A novel multiple attribute decision-making approach for evaluation of emergency management schemes under picture fuzzy environment. *Int. J. Mach. Learn. Cybern.* **2021**, *13*, 633–661. [[CrossRef](#)]
2. Lee, J.; Wood, J.; Kim, J. Tracing the Trends in Sustainability and Social Media Research Using Topic Modeling. *Sustainability* **2021**, *13*, 1269. [[CrossRef](#)]
3. Usf.edu. Introduction to Social Media, University Communications and Marketing. 2020. Available online: <https://www.usf.edu/ucm/marketing/intro-social-media.aspx> (accessed on 1 January 2021).
4. Koch, J. *Teach Introduction to Education*; SAGE Publications: New York, NY, USA, 2018; p. 119.
5. Oussous, A.; Benjelloun, F.; Lahcen, A.A.; Belfkih, S. Big Data technologies: A survey. *J. King Saud Univ. Comput. Inf. Sci.* **2018**, *30*, 431–448. [[CrossRef](#)]
6. Bhadani, A.; Jothimani, D. Big data: Challenges, opportunities and realities. In *Effective Big Data Management and Opportunities for Implementation*; Singh, M.K., Kumar, D.G., Eds.; IGI Global: Pennsylvania, PA, USA, 2016; pp. 1–24.
7. Horita, F.E.; de Albuquerque, J.P.; Marchezini, V.; Mendiondo, E.M. Bridging the gap between decision-making and emerging big data sources: An application of a model-based framework to disaster management in Brazil. *Decis. Support Syst.* **2017**, *97*, 12–22. [[CrossRef](#)]
8. Alomari, E.; Katib, I.; Albeshri, A.; Yigitcanlar, T.; Mehmood, R. Iktishaf+: A Big Data Tool with Automatic Labeling for Road Traffic Social Sensing and Event Detection Using Distributed Machine Learning. *Sensors* **2021**, *21*, 2993. [[CrossRef](#)] [[PubMed](#)]
9. Alfalqi, K.; Bellaiche, M. IoT-Based Disaster Detection Model Using Social Networks and Machine Learning. In Proceedings of the 2021 4th International Conference on Artificial Intelligence and Big Data (ICAIBD), Chengdu, China, 28–31 May 2021; pp. 92–97. [[CrossRef](#)]
10. Septianto, G.R.; Mukti, F.F.; Nasrun, M.; Gozali, A.A. Jakarta congestion mapping and classification from twitter data extraction using tokenization and naïve bayes classifier. In Proceedings of the 2015 Asia Pacific Conference on Multimedia and Broadcasting, Bali, Indonesia, 23–25 April 2015. [[CrossRef](#)]
11. Toujani, R.; Akaichi, J. Event news detection and citizens community structure for disaster management in social networks. *Online Inf. Rev.* **2019**, *43*, 113–132. [[CrossRef](#)]
12. Banujan, K.; Kumara, T.G.S.B.; Paik, I. Twitter and Online News analytics for Enhancing Post-Natural Disaster Management Activities. In Proceedings of the 2018 9th International Conference on Awareness Science and Technology (iCAST), Fukuoka, Japan, 19–21 September 2018; pp. 302–307. [[CrossRef](#)]
13. Hagrás, M.; Hassan, G.; Farag, N. Towards Natural Disasters Detection from Twitter Using Topic Modelling. In Proceedings of the 2017 European Conference on Electrical Engineering and Computer Science (EECS), Bern, Switzerland, 17–19 November 2017.
14. Ragini, J.R.; Anand, P.R.; Bhaskar, V. Big data analytics for disaster response and recovery through sentiment analysis. *Int. J. Inf. Manag.* **2018**, *42*, 13–24. [[CrossRef](#)]
15. Salas, A.; Georgakis, P.; Nwagbososo, C.; Ammari, A.; Petalas, I. Traffic event detection framework using social media. In Proceedings of the 2017 IEEE International Conference on Smart Grid and Smart Cities (ICSGSC), Singapore, 23–26 July 2017; pp. 303–307.
16. Lau, R.Y. Toward a social sensor based framework for intelligent transportation. In Proceedings of the 2017 IEEE 18th International Symposium on A World of Wireless, Mobile and Multimedia Networks (WoWMoM), Hong Kong, China, 12–15 June 2017; pp. 1–6. [[CrossRef](#)]
17. Bhuvaneswari, A.; Jayanthi, R.; Meena, A.L. Improving Crisis Event Detection Rate in Online Social Networks Twitter Stream using Apache Spark. *J. Phys.: Conf. Ser.* **2021**, *1950*, 012077. [[CrossRef](#)]
18. Alomari, E.; Mehmood, R.; Katib, I. Sentiment Analysis of Arabic Tweets for Road Traffic Congestion and Event Detection. In *Smart Infrastructure and Applications*; Springer: Berlin/Heidelberg, Germany, 2019; pp. 37–54. *\_2*. [[CrossRef](#)]
19. Alghamdi, N.; Alrajebah, N.; Al-Megren, S. Crowd Behavior Analysis using Snap Map: A Preliminary Study on the Grand Holy Mosque in Mecca. In Proceedings of the 2019 on Computer Supported Cooperative Work and Social Computing (CSCW'19), Austin, TX, USA, 9–13 November 2019; pp. 137–141. [[CrossRef](#)]
20. Alghamdi, N.; Alageeli, N.; Abu Sharh, D.; Alqahtani, M.; Al-Razgan, M. An Eye on Riyadh Tourist Season: Using Geo-tagged Snapchat Posts to Analyse Tourists Impression. In Proceedings of the 2020 2nd International Conference on Computer and Information Sciences (ICCIS), Sakaka, Saudi Arabia, 13–15 October 2020; pp. 1–6. [[CrossRef](#)]
21. Juhász, L.; Hochmair, H.H. Analyzing the spatial and temporal dynamics of Snapchat. In Proceedings of the AnaLysis, Integration, Vision, Engagement (VGI-ALIVE) Workshop, Lund, Sweden, 12 June 2018.
22. Lamba, H.; Srikanth, S.; Pailla, D. *Driving the Last Mile: Characterizing and Understanding Distracted Driving Posts on Social Networks*; Association for the Advancement of Artificial Intelligence: Menlo Park, CA, USA, 2019.
23. Shah, S.A.; Seker, D.Z.; Rathore, M.M.; Hameed, S.; Ben Yahia, S.; Draheim, D. Towards Disaster Resilient Smart Cities: Can Internet of Things and Big Data Analytics Be the Game Changers? *IEEE Access* **2019**, *7*, 91885–91903. [[CrossRef](#)]
24. Kwekha-Rashid, A.S.; Abduljabbar, H.N.; Alhayani, B. Coronavirus disease (COVID-19) cases analysis using machine-learning applications. *Appl. Nanosci.* **2021**. [[CrossRef](#)]
25. Yang, J.; Qu, J.; Mi, Q.; Li, Q. A CNN-LSTM Model for Tailings Dam Risk Prediction. *IEEE Access* **2020**, *8*, 206491–206502. [[CrossRef](#)]

26. Zhou, H.; Zhang, J.; Zhou, Y.; Guo, X.; Ma, Y. A feature selection algorithm of decision tree based on feature weight. *Expert Syst. Appl.* **2020**, *164*, 113842. [CrossRef]
27. Olowononi, F.O.; Rawat, D.B.; Liu, C. Resilient Machine Learning for Networked Cyber Physical Systems: A Survey for Machine Learning Security to Securing Machine Learning for CPS. *IEEE Commun. Surv. Tutorials* **2020**, *23*, 524–552. [CrossRef]
28. Bout, E.; Loscri, V.; Gallais, A. How Machine Learning Changes the Nature of Cyberattacks on IoT Networks: A Survey. *IEEE Commun. Surv. Tutorials* **2021**, *24*, 248–279. [CrossRef]
29. Gu, J.; Lu, S. An effective intrusion detection approach using SVM with naïve Bayes feature embedding. *Comput. Secur.* **2020**, *103*, 102158. [CrossRef]
30. Qolomany, B.; Al-Fuqaha, A.; Gupta, A.; Benhaddou, D.; Alwajidi, S.; Qadir, J.; Fong, A.C. Leveraging Machine Learning and Big Data for Smart Buildings: A Comprehensive Survey. *IEEE Access* **2019**, *7*, 90316–90356. [CrossRef]
31. Podhoranyi, M. A comprehensive social media data processing and analytics architecture by using big data platforms: A case study of twitter flood-risk messages. *Earth Sci. Inform.* **2021**, *14*, 913–929. [CrossRef]
32. Juhász, L.; Hochmair, H. Comparing the Spatial and Temporal Activity Patterns between Snapchat, Twitter and Flickr in Florida. *GIForum* **2019**, *1*, 134–147.\_01\_s134. [CrossRef]
33. Hernandez-Suarez, A.; Sanchez-Perez, G.; Toscano-Medina, K.; Perez-Meana, H.; Portillo-Portillo, J.; Sanchez, V.; Villalba, L.J.G. Using Twitter Data to Monitor Natural Disaster Social Dynamics: A Recurrent Neural Network Approach with Word Embeddings and Kernel Density Estimation. *Sensors* **2019**, *19*, 1746. [CrossRef]
34. Said, N.; Ahmad, K.; Riegler, M.; Pogorelov, K.; Hassan, L.; Ahmad, N.; Conci, N. Natural disasters detection in social media and satellite imagery: A survey. *Multimedia Tools Appl.* **2019**, *78*, 31267–31302. [CrossRef]
35. Eyada, M.M.; Saber, W.; El Genidy, M.M.; Amer, F. Performance Evaluation of IoT Data Management Using MongoDB Versus MySQL Databases in Different Cloud Environments. *IEEE Access* **2020**, *8*, 110656–110668. [CrossRef]
36. Wijeratne, S.; Sheth, A.; Bhatt, S.; Balasuriya, L.; Al-Olimat, H.S.; Gaur, M.; Yazdavar, A.H.; Thirunarayan, K. *Feature Engineering for Twitter-Based Applications*; CRC Press: Boca Raton, FL, USA, 2018; pp. 359–393. [CrossRef]
37. De Pablo, Á.; Araque, O.; Iglesias, C.A. Transfer Learning with Social Media Content in the Ride-Hailing Domain by Using a Hybrid Machine Learning Architecture. *Electronics* **2022**, *11*, 189. [CrossRef]
38. Hasan, A.; Moin, S.; Karim, A.; Shamshirband, S. Machine Learning-Based Sentiment Analysis for Twitter Accounts. *Math. Comput. Appl.* **2018**, *23*, 11. [CrossRef]
39. Awan, F.M.; Saleem, Y.; Minerva, R.; Crespi, N. A Comparative Analysis of Machine/Deep Learning Models for Parking Space Availability Prediction. *Sensors* **2020**, *20*, 322. [CrossRef] [PubMed]
40. Ramzy, A.; Peltier, E. What We Know about the Beirut Explosions. *The New York Times*. Available online: <https://www.nytimes.com/2020/08/05/world/middleeast/beirut-explosion-what-happened.html> (accessed on 4 October 2021).
41. Miguel, J.; Caballé, S.; Xhafa, F.; Prieto, J. A massive data processing approach for effective trustworthiness in online learning groups. *Concurr. Comput. Pr. Exp.* **2014**, *27*, 1988–2003. [CrossRef]
42. Hossin, M.; Sulaiman, M.N. A Review on Evaluation Metrics for Data Classification Evaluations. *Int. J. Data Min. Knowl. Manag. Process* **2015**, *5*, 1–11. [CrossRef]
43. Grandini, M.; Bagli, E.; Visani, G. Metrics for Multi-Class Classification: An Overview. *arXiv* **2020**, arXiv:abs/2008.05756.