Accident Analysis and Prevention 151 (2021) 105973

Contents lists available at ScienceDirect

Accident Analysis and Prevention

journal homepage: www.elsevier.com/locate/aap

Traffic accident detection and condition analysis based on social networking data

Farman Ali [a,*,1], Amjad Ali [b,1], Muhammad Imran [c], Rizwan Ali Naqvi [d], Muhammad Hameed Siddiqi [e], Kyung-Sup Kwak [f,*]

[a] Department of Software, Sejong University, Seoul, South Korea
[b] Department of Computer Science, COMSATS University Islamabad, Lahore Campus, Lahore, Pakistan
[c] College of Applied Computer Science, King Saud University, Riyadh, Saudi Arabia
[d] Department of Unmanned Vehicle Engineering, Sejong University, Seoul, South Korea
[e] Department of Computer Science, Jouf University, Sakaka, Saudi Arabia
[f] Department of Information and Communication Engineering, Inha University, Incheon, South Korea

ARTICLE INFO

Keywords:
Traffic accident detection
Traffic accident analysis
Traffic monitoring system
Ontology
Bi-LSTM

ABSTRACT

Accurate detection of traffic accidents as well as condition analysis are essential to effectively restoring traffic flow and reducing serious injuries and fatalities. This goal can be obtained using an advanced data classification model with a rich source of traffic information. Several systems based on sensors and social networking platforms have been presented recently to detect traffic events and monitor traffic conditions. However, sensor-based systems provide limited information, and may fail owing to the long detection times and high false-alarm rates. In addition, social networking data are unstructured, unpredictable, and contain idioms, jargon, and dynamic topics. The machine learning algorithms utilized for traffic event detection might not extract valuable information from social networking data. In this paper, a social network–based, real-time monitoring framework is proposed for traffic accident detection and condition analysis using ontology and latent Dirichlet allocation (OLDA) and bidirectional long short-term memory (Bi-LSTM). First, the query-based search engine effectively collects traffic information from social networks, and the data preprocessing module transforms it into structured form. Second, the proposed OLDA-based topic modeling method automatically labels each sentence (e.g., *traffic* or *non-traffic*) to identify the exact traffic information. In addition, the ontology-based event recognition approach detects traffic events from traffic-related data. Next, the sentiment analysis technique identifies the polarity of traffic events employing user's opinions, which helps determine accurate conditions of traffic events. Finally, the FastText model and Bi-LSTM with softmax regression are trained for traffic event detection and condition analysis. The proposed framework is evaluated using traffic-related data, comparing OLDA and Bi-LSTM with existing topic modeling methods and traditional classifiers using word embedding models, respectively. Our system outperforms state-of-the-art methods and achieves accuracy of 97 %. This finding demonstrates that the proposed system is more efficient for traffic event detection and condition analysis, in comparison to other existing systems.

1. Introduction

Real-time detection and analysis of traffic accidents is one of the most important and challenging tasks for both travelers and transport management sectors. Traffic accidents affect traffic flow and traffic operations, and causing serious injuries and even fatalities. The World Health Organization (WHO) has stated that 1.35 million people die every year worldwide because of traffic accidents (Aqib et al., 2020). Moreover, the United States of America spends US$160 billion a year, on average, due to traffic events, including traffic accidents and traffic jams, and that figure might reach US$192 billion by the end of 2020 (Wang et al., 2017). A major cause of death between the ages of 15 and

* Corresponding authors.
E-mail addresses: farmankanju@sejong.ac.kr (F. Ali), amjad.ali@cuilahore.edu.pk (A. Ali), dr.m.imran@ieee.org (M. Imran), rizwanali@sejong.ac.kr (R.A. Naqvi), mhsiddiqi@ju.edu.sa (M.H. Siddiqi), kskwak@inha.ac.kr (K.-S. Kwak).
[1] These authors contributed equally to this work and co-first authors.

https://doi.org/10.1016/j.aap.2021.105973
Received 24 September 2020; Received in revised form 24 December 2020; Accepted 29 December 2020
Available online 15 January 2021
0001-4575/

30 in worldwide is traffic accidents (Wang et al., 2017). Researchers classify traffic congestion into recurring and non-recurring events, which refer to predictable and unpredictable incidents, respectively. A traffic accident is one of the main non-recurring events that disturb traffic flow. Therefore, accurate and early detection of traffic accidents in real time and in an efficient manner, along with condition information, is very important to help drivers avoid risk zones and choose the fastest and safest routes. In addition, it also helps the transportation management sector and state police reduce fatalities and restore traffic flow as easily and quickly as possible.

Several detection systems based on sensor devices (e.g., loop detectors, cameras, GPSs) have been presented to collect traffic information, such as vehicle speeds and locations, times, and types of traffic events (Chen et al., 2017; Guerrero-Ibáñez et al., 2018; Wang et al., 2017; Zhang et al., 2019). These systems employ traditional algorithms for the analysis of collected data to detect traffic incidents. However, sensor-based systems have some limitations. First, installation and operation of sensor devices is costly, and provides limited traffic information. In addition, communication errors and detector failure often occur in traffic monitoring systems. The failure of sensors (e.g., loop detectors) could create a serious problem for traffic accident detection in large areas. Second, the existing algorithms utilized for traffic event identification can fail owing to the long detection times and high false-alarm rates. To overcome these limitations, social networking platforms and advanced deep-learning models can be utilized as a source of real-time traffic data and for traffic-accident detection algorithms, respectively.

Recently, various systems have been presented to detect traffic accidents using social networking sites and text mining techniques, as discussed in the related work section below. A crowdsourcing navigation system called Waze has been successfully used in recent years, providing real-time information about accidents, traffic, weather conditions, and blocked roads (Hoseinzadeh et al., 2020; Zhang et al., 2020). However, Waze is incapable of providing reports on traffic events that do not belong to one of its predefined event categories (Vallejos et al., 2020). In addition, Waze was developed for private vehicles; it does not provide information about public transportation. The data from Waze are only available to registered users, and therefore, it is difficult for the public sector (e.g., traffic management offices) to access them. Social media can be a valuable solution for addressing the limitations in Waze. People share their thoughts on social media associated with traffic events (e.g., traffic jams, vehicle crashes, landslides, and so on). Therefore, social networking platforms have become a source of massive amounts of data on traffic events. However, social media data are unstructured, unpredictable, and contain idioms, jargon, and dynamic topics. Extracting valuable information from social media data may be a job too complex for traffic detection systems. Therefore, it is essential to develop a novel model that can extract the most significant traffic-related data, and then explore the extracted data to find traffic events.

To detect traffic-related data, natural language processing (NLP) and machine learning (ML) models have recently been implemented to extract valuable information from unstructured social media data (Cao et al., 2018; Chen et al., 2018; Salas et al., 2018; Zhang et al., 2018). However, there are still limitations in the present methods with respect to handling social media data for traffic accident detection. First, the systems presented include manually labeling the data before applying ML techniques. Manual labeling of a large amount of textual data is almost impossible because it is too time consuming. Second, a traditional method called the word2vec model has been employed for text representation. However, word2vec assigns a different vector to each word, and ignores the morphology of words. It does not represent a new word with a vector if the word is not available in the training dataset. Therefore, the word2vec model fails to effectively perform in the context of social networking data. Third, traditional ML models may not precisely handle the unstructured data from social media for detection and analysis of traffic incidents. Therefore, an intelligent system is required, with advanced techniques that can automatically label traffic data and precisely represent them for condition analysis of traffic incidents.

In this paper, we propose a smart methodological framework for traffic accident detection and condition analysis using ontology and latent Dirichlet allocation (OLDA) and bidirectional long short-term memory (Bi-LSTM). First, data about non-recurring traffic events are effectively collected from the social networking platforms Twitter and Facebook. The collected data are preprocessed in order to convert them into structured form. Second, OLDA is applied to automatically allocate a class label (e.g., *traffic* or *non-traffic*) to each sentence in the preprocessed data. The ontology-based name entity recognition (NER) approach is then used to detect a traffic event and its location. In addition, user sentiments about traffic events are analyzed using SentiWordNet (SWN) to identify the polarity of traffic events. Finally, the FastText model and Bi-LSTM with softmax regression are trained to detect traffic events and predict their polarity. This paper's main contributions are as follows.

- A novel, smart framework is proposed to efficiently analyze social networking data for traffic event detection and condition analysis using OLDA and Bi-LSTM model.
- A new topic modeling module is developed based on OLDA to automatically label each sentence (*traffic or non-traffic*) in order to identify traffic-related information. In addition, an ontology-based event identification system is implemented to accurately extract traffic events mentioned in context.
- A sentiment analysis approach is effectively applied for condition analysis of a traffic event. This approach efficiently utilizes user sentiments about traffic events, and identifies the polarity of each sentiment word to classify the condition of a traffic event as positive, neutral, or negative.
- For document representation, the FastText embedding model accurately transforms formal and informal words from social networking text into low-dimensional vector representations. In addition, Bi-LSTM with softmax regression is utilized to learn semantic information effectively, which increases the accuracy of traffic event detection and polarity prediction.
- The proposed system enhances the efficiency of traffic event detection and achieves accuracy of 97 %, compared with existing systems.

The remainder of this paper is structured as follows. Section 2 discusses the recent traffic event detection systems using sensors and social media data. Section 3 presents the proposed framework, data preprocessing, the topic modeling method, text representation, and classification models. Section 4 evaluates the efficiency of the proposed models. Finally, Section 5 concludes our research work and provides suggestions for future work.

2. Related work

Social networking platforms play an important role in traffic event detection and condition analysis. However, extracting the needed information from social networking data, and then investigating them, is a challenging task in the field of transportation analysis. In addition, accurate labeling of traffic data and precisely representing them is another challenging task for ML-based traffic polarity prediction. Therefore, this section first looks at traffic event detection using sensors and social networking data; we then focus on class labeling of social media data and data classification, and then sentiment analysis–based traffic polarity detection.

2.1. Traffic event detection using sensors and social networking data

In recent years, researchers have utilized different methods, along with traffic-related keywords, to detect and analyze traffic events using sensors and social networking data. Zheng et al. presented a data fusion

architecture to detect non-recurring traffic events using both social media and taxi GPS data (Zheng et al., 2018). They used the social media data to identify serious traffic events and used GPS data to find their times and locations. However, the presented framework can only detect the traffic events, but might not extract an accurate reason for them. Wang et al. proposed a system for traffic event detection using social media and GPS data (Wang et al., 2017). They developed a hybrid model to combine different types of data for estimating traffic congestion in urban areas. However, their presented method is difficult to understand, and achieved a low accuracy rate from data mining. Zhang et al. utilized both social and remote-sensing data to identify risky traffic locations in a city (Zhang et al., 2019). However, their focus was to find risky locations. Their system does not contain a method to handle the noisy data from social media. Guerreiro et al. presented the framework of an intelligent transportation system (ITS) for big data processing (Guerreiro et al., 2016). The main idea of their system is to collect, store, and transform data precisely. However, the large scale of data processing is difficult without data-management and deep-learning approaches. Nallaperuma et al. presented an unsupervised and machine learning–based traffic management system for effective decision making (Nallaperuma et al., 2019). Their system integrated smart sensors and social media data to detect non-recurring traffic events. However, that system needs semantic information to precisely handle both sensor and social media data. Xu et al. presented a 5 W (what, where, when, who, and why) model to detect emergency events in urban areas (Xu et al., 2016). The aim of their model is to use social networking data for the detection of emergency events such as traffic jams, storms, and fire. Their model lacks a feature extraction approach, which may increase the complexity of handling social media data. Gutierrez et al. proposed a Twitter mining-based system to detect traffic events in the United Kingdom (Gutierrez et al., 2015). They used data from different agencies published on Twitter to identify traffic events. However, their system needs a real-time clustering method in order to minimize the ambiguities of traffic-related feature extraction. Lin et al. presented an automated traffic incident detection system based on generative adversarial networks (GANs) (Lin et al., 2020). They utilized temporal rules in order to extract useful variables from traffic data, and then applied random forest to rank the extracted variables. In addition, they employed GANs and SVM to generate new incident samples and detect the traffic incident, respectively. However, SVM can work well only for small and balanced dataset. Therefore, their proposed framework may not properly handle a large amount of textual data related to traffic incidents. Lin and Li introduced a novel method how to use crowdsourcing data for traffic accident analysis in real-time (Lin and Li, 2020). They employed user-generated data and explored algorithms to predict the behavior of traffic flow after the traffic accidents. Their study proved that crowdsourcing data is more effective than single-sourcing data for traffic condition analysis.

2.2. Class labeling and classification

Accurate class labeling of social media data can boost the performance of traffic event detection and polarity classification. Pereira et al. proposed an automatic system that classifies travel-related Tweets from geo-located data (Pereira et al., 2017). They used a word embedding model called paragraph2vec and ML classifiers for Tweet representation and classification, respectively. However, paragraph2vec may not represent textual data with low dimensionality, which can affect the classification results. Angelica-Salas et al. presented a real-time incident detection system based on Twitter analysis methods (Salas et al., 2018). They manually labeled tweets and applied various methods for data classification. However, their system achieved very low accuracy due to the usage of traditional methods. Dabiri and Heaslip presented a deep-learning model for both data representation and classification (Dabiri and Heaslip, 2019). They manually labeled the Tweets and applied the word2vec model for data representation. In addition, they utilized a Convolutional Neural Network (CNN) and a Recurrent Neural Network (RNN) to classify traffic event data into three classes: non-traffic, traffic incident, and traffic information and condition. However, Twitter data are usually informal, short, and contain a lot of special characters. The system by Dabiri and Heaslip may not understand the actual meaning of a word without semantic knowledge. D'Andrea et al. presented a real-time monitoring framework for traffic event detection from Twitter data (D'Andrea et al., 2015). They manually labeled the data (traffic Tweet or non-traffic Tweet) and utilized NLP techniques for data processing and a support vector machine (SVM) as a classification model. However, their traditional methods for data representation and classification may not be applicable to the current informal and unstructured data of social networks. Chen et al. presented a detection system for the extraction of traffic information from Sina Weibo (Chen et al., 2018). They used the word2vec model along with a combination of LSTM and a CNN to detect traffic texts. However, their work can be further improved to extract the time and location of the occurrence of a traffic event for further analysis.

There have been some studies using LDA for topic labeling and data classification (Bastani et al., 2019; Kim et al., 2015; Yang and Rim, 2014; Zhou et al., 2016). Zhang et al. proposed a deep learning–based system for traffic accident detection from social media data (Zhang et al., 2018). They manually labeled the data and utilized two different classifiers (a deep neural network and LSTM) for traffic-related token extraction. In addition, they applied an SVM and supervised LDA (SLDA) to classify social networking data. However, LDA ignores word order, which affects the classification results. Gu et al. presented a filtering system for event information from occurrences on both arterial roads and highways (Gu et al., 2016). They collected Tweets using a data acquisition method, and labeled them as traffic incident events and non-traffic incidents. They used the bag-of-words (BoW) model with LDA to put traffic incident–related texts into five classes: road work, obstacle vehicles, events, hazards and weather, and accidents. However, their system is limited to a certain number of Tweets. Additional Tweets can provide extra information that allows investigating the benefits and costs of the higher Tweet sampling rate. In addition, their system needs an advanced NLP model, which may increase the classification accuracy. Lu et al. proposed a system based on word2vec and LDA to comprehensively describe traffic accidents (Lu et al., 2018). They presented social signals semantically in a description of traffic accidents. They first identify traffic events from social networking platforms, and then fuse the LDA-based detected topic with overall traffic events using the word2vec model. However, their system has some limitations. First, it is only limited to the Sina Webo platform, but it might detect traffic events with more comprehensive details using other social media sources (e.g., Facebook and Twitter). Second, a deep learning model such as LSTM and the RNN might improve the system accuracy in traffic event detection. Roque et al. described how topic modeling can be effectively utilized to detect attributes related to road crashes (Roque et al., 2019). They used LDA-based framework to extract topics from road safety inspection (RSI) reports. In addition, they combined topic modeling method with text mining to enhance the performance of information extraction from RSI reports. Their system results proved that LDA method is appropriate for extracting the attributes of road crashes from RSI. Goh et al. presented various text mining methods to classify the accident narrative (Goh and Ubeynarayana, 2017). They extracted accident information from the US OSHA website, and then classified it using six different ML algorithms. However, they failed to achieve decent accuracy of classification. Their system needs advanced approach with good F1 score for automatic labeling of the accident narratives.

2.3. Sentiment analysis-based traffic polarity prediction

A sentiment analysis method can help to analyze the conditions of different events. Therefore, several methods of sentiment analysis have

been presented to analyze event conditions or to predict feature polarity (Cao et al., 2014; Poria et al., 2016; Reynard and Shirgaokar, 2019; Serrano-Guerrero et al., 2020; Valdivia et al., 2018; Yao and Wang, 2020; Yoo et al., 2018; Young et al., 2018; Zhou et al., 2019). Cao et al. presented a semi-supervised learning method to investigate traffic events using user sentiments (Cao et al., 2018). They first trained their model to estimate the sentiments using emoticons and textual data, and then predicted whether traffic is jammed or not. However, emoticons may not provide enough information to accurately predict the polarity of an individual feature. Therefore, their system achieved low accuracy for sentiment classification. Ali and various colleagues proposed three different frameworks for sentiment analysis of transportation (Ali et al., 2019b, 2019a, 2017). The first system was fuzzy ontology–based sentiment analysis for traffic activity monitoring (traffic jams and accidents) (Ali et al., 2017). This system extracts a user's opinion from social media data, and identifies the polarity of transportation features. Ali et al. proposed another framework based on a word embedding model, LDA, and ML classifiers in order to improve the performance of the previous system (Ali et al., 2019b). This second system extracts meaningful data about traffic from social media, and then generates topics for the extracted data. After that, they apply a lexicon-based model to improve the accuracy of text representation by using a word embedding model. However, they achieved low accuracy for sentiment classification. Ali et al. proposed yet another traffic network monitoring system using a fuzzy ontology and a word embedding model (Ali et al., 2019a). They utilized the fuzzy ontology to provide semantic knowledge about traffic features for the word2vec model. This semantic knowledge improved the performance of traffic feature extraction and sentiment classification using Bi-LSTM. However, all of these systems lack adequately labeled datasets about traffic events. In addition, manually labeling data and utilizing a large number of concepts in the ontology makes the proposed semantic knowledge more complicated, which makes the task of polarity classification computationally complex.

3. Proposed framework

In this section, we present different approaches that are applied to retrieve, process, analyze, and classify social media data about traffic events. Fig. 1 shows the proposed system architecture. When traffic events such as a traffic jam, road slide, traffic accident, road closure, or traffic congestion have occurred, people share information about these events on social media. The shared information can be used as a source of data for traffic accident detection. However, social networking data are unstructured, and contain idioms, jargon, and dynamic topics. It is difficult to extract valuable information from social networks for traffic event detection and condition analysis. Therefore, the proposed methodological framework is composed of several modules, namely query-based crawling, data preprocessing, automatic labeling, text representation, and classification, which operate sequentially in order to successfully detect and analyze traffic events. The main aim of the proposed framework is to automate the detection and condition analysis of traffic accidents and events using OLDA and Bi-LSTM-based text classification. First, query-based real-time data are acquired from Twitter and Facebook using different Application Programming Interface (API) methods (Task 1 in Fig. 1). After data crawling, text mining approaches are applied to preprocess the collected data for further analysis (Task 2 in Fig. 1). OLDA-based topic modeling is employed to label all the data (traffic or non-traffic) in order to identify traffic-related data. The entity extraction approach then extracts traffic events and their locations. In addition, we analyze user sentiments about traffic events to classify event conditions as positive, neutral, or negative (Task 3 in Fig. 1). Word embedding models called FastText and word2vec then represent the traffic event data with a very low-dimensional vector (Task 4 in Fig. 1). Finally, Bi-LSTM with softmax regression is trained to categorize traffic events and predict their polarity (Task 5 in Fig. 1). By analyzing non-recurring traffic events as shown in Fig. 1 the system helps an ITS office to automatically detect traffic accidents and events and predict their polarity.

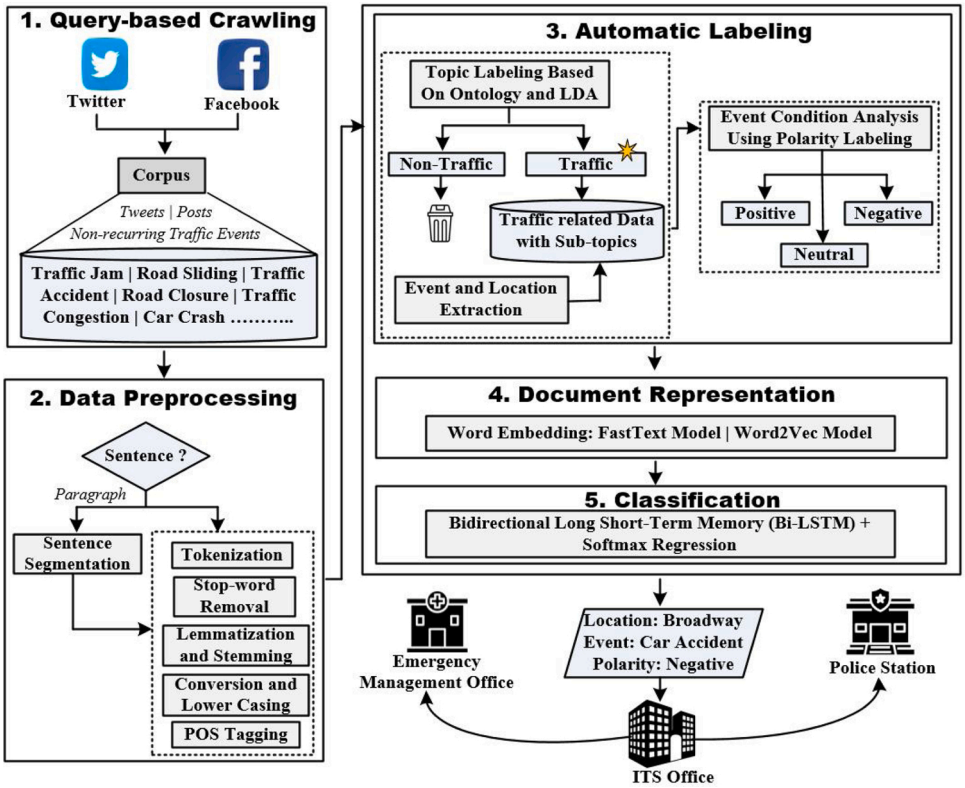


Fig. 1. The proposed system architecture for accident event detection and condition analysis.

3.1. Query-based crawling

This section discusses the procedure for data collection. We selected the most populous metropolitan cities in the U.S.A. where non-recurring traffic events mostly happen: New York, Los Angeles, and Chicago, which have populations of 8.6 million, 4 million, and 2.7 million, respectively. The data on non-recurrent traffic events occurring in these cities were retrieved using the social network platforms Twitter and Facebook. These platforms offer the following API methods that allow users to access the data. Every API approach utilizes various parameters to create a specific request for data collection.

3.1.1. Twitter data

There are several Twitter API methods that allow direct access to public Tweets. These APIs are divided into three different tiers: standard, premium, and enterprise. The use of the standard APIs is free, but it has limited functionality, whereas the other two APIs offer thorough access to Twitter data by charging more than US$2000 per month. We utilized the standard APIs (*REST API*, *Streaming API*, *Get users/search API*, and *Get statuses/user_timeline API*) to retrieve Tweets containing data about non-recurring traffic events. The *REST API* lets users retrieve Twitter data using keyword-based queries (Ali et al., 2019b, 2019a, 2017). To retrieve the most recent related Tweets, the *REST API* queries must be specific in terms of traffic event data. Thus, the constructed queries used in this study contained a set of keywords with Boolean operators OR and AND (e.g., car AND (crash OR collision)), a centroid (latitude, longitude), and a radius (e.g., 0.3 miles). However, queries in *REST API* are limited to 350 per 15 min. Therefore, we employed the most useful 50 queries, which were constructed using 500 keywords related to traffic accidents and events. The *Streaming API* is employed to retrieve real-time Twitter data using queries similar to those in *REST API*. However, users must have a continuous HTTP connection to access the most recent Tweets. The *Get users/search API* employs a query to identify a specific user account on Twitter. The fourth type, the *Get statuses/user_timeline API* approach, uses a unique name identity to retrieve Twitter data posted by a specific user. Both the REST and the streaming APIs are utilized to collect Tweets from general users, whereas the other two APIs collect Tweets from authorized accounts, which for this study means official Twitter accounts of transportation departments. These accounts post formal Tweets about traffic events with exact times and locations, which helps to process them easily. For example, one Tweet about a traffic accident was "*A two-car crash in Los Angeles has injured one person, traffic congestion in the evening at downtown*". From this Tweet, the information about the traffic event and location can easily be extracted using a keyword-based search engine. We selected 30 authorized Twitter accounts that post real-time traffic event information from New York, Los Angeles, and Chicago. We collected 600,000 Tweets about traffic events by employing the abovementioned API methods with queries.

3.1.2. Facebook data

Recently, e-commerce companies, organizations, and traffic incident management offices have been using the Facebook platform to share activities and event information with people. In this study, some transportation departments' pages on Facebook were analyzed for data collection. We first identified the oldest page with the most posts. The Facebook API cannot identify the page-created date. Therefore, we checked the dates of posts published on these pages, and then considered pages that contained old posts. We selected Facebook pages that comprise traffic event information from New York, Los Angeles, and Chicago (e.g., the Metropolitan Transportation Authority, future transportation, the New York Times, the New York State Department of Transportation, and RTA-Regional Transportation Authority-IIIinois). The posts published on those pages between March 2017 and January 2019 were retrieved using the Facebook Graph API with a Java client (RestFB). However, the posts published by fans of the pages were eliminated, and we considered only those posts published by *admin*, which contained exact traffic information for event detection. We collected 5000 posts about traffic events by employing the abovementioned API methods.

3.2. Data preprocessing

Data preprocessing is the second module of the proposed system, which transforms the raw and unstructured data of social networks into a more structured form, as shown in Fig. 2. People share information on social networks in a very casual way, with text that comprises hashtags, special characters, and useless words. It is important to utilize text mining methods to filter out these characters from the corpus data before feeding it to classifiers. However, the posts retrieved from Facebook are in the form of paragraphs. Therefore, we first use a sentence segmentation approach to split those paragraphs into sentences. The following preprocessing methods are then applied in sequence, which presents unstructured social network data in an organized form to easily detect traffic accident events and identify their polarity.

3.2.1. Tokenization

Tokenization is a process of converting complex text into a set of words called *tokens*. Complex text generally contains hashtags, punctuation, non-text characters, and word-space. In the proposed system, an n-gram tokenization approach is employed to eliminate non-alphanumeric characters and split each portion of text into words (a bag-of-words representation). After this step, each text in the corpus is represented as a sequence of words for further processing.

3.2.2. Stop-word removal

This step eliminates stop-words that provide no valuable information for traffic event detection, polarity prediction, or sentiment classification. The most common stop-words are pronouns, prepositions, symbols (@, dates, #, etc.), conjunctions, and articles (a, an, the). In addition, URLs in the text do not comprise information for text analysis. Therefore, using a stop-word handler called Rainbow, we eliminate content that does not help detect traffic events and predict polarity. Note that numbers and negations play an important role in traffic event detection and sentiment classification, respectively. Numbers are utilized to discover places and entities (e.g., 4th Street and Bus 801) while negations help to predict polarity. Thus, the proposed system uses numbers and negations, respectively, to easily find entities and compute polarity about traffic events.

3.2.3. Lemmatization and stemming

Lemmatization plays a key role in sentiment prediction. It unites many words into one. It morphologically analyzes each word of a sentence and generates its base form. A stemming method removes the last character(s) of a word in order to convert the word to its root form. For example, a text about a traffic accident might be "a traffic accident in the area of New York downtown, two cars involved. Injuries have been reported, road closed." In these clauses, the words *involved*, *injuries*, *reported*, and *closed* are converted to their basic forms: *involve*, *injury*, *report*, and *close*, respectively. We employ a suffix-dropping algorithm to convert words to their root form. This approach merges various words that reduce the dimensionality of text representation. Both stemming and lemmatization are extensively employed methods that produce good results. These methods should not be employed together, because they are mutually exclusive.

3.2.4. Conversion and lowercasing

Text on social network platforms is written in an informal way that contains unusual words to express people emotions. For example, people sometimes present their opinions about a traffic jam by using the word *jammed* in the form *jammmed*. In our process, each word is represented in its generic form. In addition, all the text in the corpus is converted to

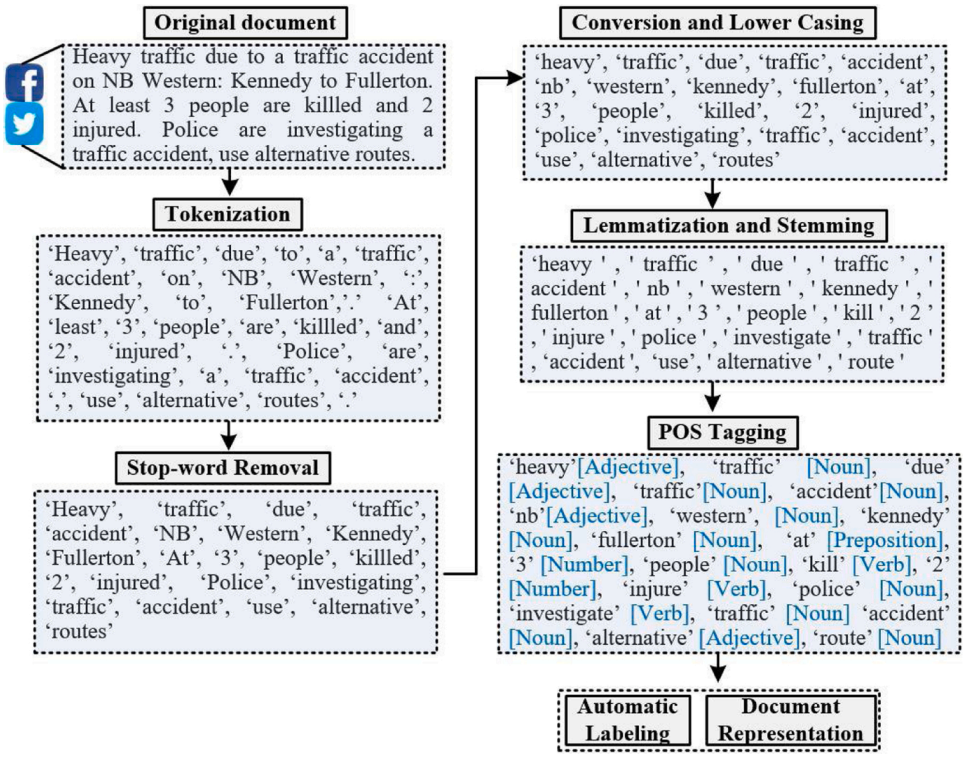


Fig. 2. Data preprocessing steps.

lowercase, which provides a constant format for all the text in order to avoid confusion during text analysis.

3.2.5. *Part-of-Speech (POS) tagging*

Part-of-speech (POS) tagging is the procedure of assigning a label to each word of a sentence. These labels can be *noun*, *verb*, *adjective*, *adverb*, etc. The main aim of this step in preprocessing is to remove any POS that does not constitute a word for traffic event detection. In addition, POS tagging identifies nouns and adjectives, which are the most valuable indicators for entity extraction and sentiment classification. The proposed system tags each word used the Stanford tagger CoreNLP. This step makes entity recognition much easier.

3.3. *OLDA-based topic modeling*

In this section, we utilize an OLDA-based topic modeling method to assign a class label to each sentence of the preprocessed data. The NER approach is then used to detect traffic events and their locations. After event detection, SWN is utilized to detect the polarity of each traffic event based on people's opinions.

3.3.1. *LDA-based topic detection*

After preprocessing, LDA is applied to detect topic trends regarding traffic events from cleaned data. LDA was developed to generate latent topics in textual documents using a probabilistic distribution method (Kim et al., 2015). When social website users want to post something on social networks, they first select a topic based on a current event. Social website users then select different words based on the selected topics. The process of LDA-based topic modeling using preprocessed data is illustrated in Fig. 3. The main idea is that each textual document, D, demonstrates a different latent topic with distributed probability Θd, wherein each topic is described by different words with distributed probability Φtp. Both hyper-parameters α and β are prior distributions of Θd and Φtp, respectively. The hyper-parameter α illustrates the relation between documents and topics, whereas β is the distributed probability of all latent topics. The plates D, K, and N indicate documents in the corpus, the total number of topics, and unique words in document, respectively. There are two main jobs performed by LDA. First, it allocates unique words to different topics in individual documents. Second, it assigns a probability to words in individual topics. However, these two jobs generate confusion for each other. In the first job, LDA might allocate only limited terms/words to each topic. In the second job, LDA might allocate many documents to a single topic. In the proposed system, we detect a group of strongly co-occurring words to adjust these tasks. The generative process of LDA is described in Algorithm 1. In the proposed system, Gibbs sampling is utilized to determine the topic word and document topic with distributed probabilities Φ and Θ, respectively. The following equations are used to compute a k-dimensional distributed random variable for probability α and the distribution of θ, Tp, and W (Ali et al., 2019b; Ren and Hong, 2017):

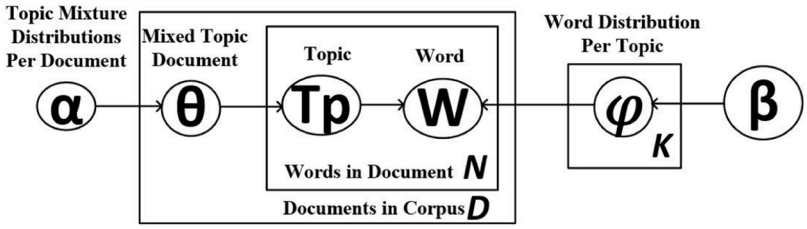


Fig. 3. The LDA model for topic generation.

$$p(\theta\alpha) = \frac{\Gamma\left(\sum_{i=1}^{k} \propto_i\right)}{\prod_{i=1}^{k} \Gamma(\propto_i)} \theta_1^{\alpha_1-1} \cdot \theta_2^{\alpha_2-1} \ldots . \theta_K^{\alpha_{k-1}} \quad (1)$$

$$p(\theta, Tp, W\alpha, \beta) = p(\theta\alpha) \prod_{n=1}^{N} p(Tp_n\theta) P(W_n Tp_n, \ \beta) \quad (2)$$

$$p(D\alpha, \beta) = \prod_{d=1}^{M} \int p(\theta_d|\alpha) \left(\prod_{n=1}^{N_d} \sum_{Z_{dn}} p(Tp_{dn}\theta_d) p(w_{dn}|Tp_{dn}, \beta) \right) \mathrm{d}\theta_d \quad (3)$$

where the parameters Γ(x) and α represent the gamma function and the k-vector, respectively. In this research, n-gram LDA is applied to produce topics on traffic event–related texts. There are two main outputs of LDA: topic β_k and the probabilistic weights in each document, Θ_d. Fig. 4 presents an explanatory example of LDA output on traffic event data (a Tweet and a post in the original document). In this example, W shows words in the traffic event data, e.g. *transportation*, and Tp denotes word allocation to the topic, e.g. the word *transportation* is allocated to topic 1; hence, Tp = 1. LDA summarizes the traffic event data in the form of three main and top topics, β_A, β_B, and β_C, with the largest weights represented by Θ_d, as shown at the bottom of Fig. 4. Each topic contains three words; e.g. topic β_A contains *transportation*, *construction*, and *department*, with the highest probabilities being 0.31, 0.21, and 0.13, respectively.

Here, each document is summarized as a grouping of topics generated by LDA. This helps us to avoid dealing with the entire text in the document, which is a time-consuming task. The results show that it is possible to generate a mixture of most valuable topics using LDA for the representation of traffic event–related textual data. This proves that LDA is an interesting method to automatically discover a large volume of textual data about traffic events, and to overcome the workload of manual labeling. However, the output of LDA has some limitations. First, the LDA results comprise words that are not related to traffic events when other text about the traffic event includes them. Second, when a document comprises words with low probability, LDA ignores the relationship between the topic and the text. Third, LDA cannot acquire semantic relationships among words. In addition, it assigns inadequate words and topics to each topic and document, respectively. Hence, the document vector is insufficient. Fourth, we also use Tweets for traffic event detection, which are a kind of short text. LDA produces noisy topics when Tweets contain very short text. To handle the above limitation, an ontology is used to enhance the LDA output and identify an appropriate label for each piece of traffic event data in the corpus.

3.3.2. OLDA-based traffic event labeling

An ontology shares knowledge between classes in a specific domain using object and data properties (Ali et al., 2019b, 2017; Arauz et al., 2012). Our ontology is developed using a well-known Web Ontology Language (OWL) tool called Protégé. In OLDA-based traffic event labeling, the domain ontology is constructed before using LDA. However, expert knowledge is needed for ontology construction, which ensures accurate semantic relationships among the various classes. Otherwise, the constructed semantic knowledge would be useless. For ontology building, traffic-related keywords and information are collected from social media. Usually, social media data comprise keywords such as *crash* or *accident* that are related to traffic or accident events. However, there is no system based on these vocabularies that can detect traffic events. Therefore, we collected more than 200 news articles on traffic events. From these articles, we selected the most frequently appearing words that represent specific traffic events, as shown in Table 1. In addition, we collected traffic event information from different traffic control management websites about road work, traffic jams, traffic accidents, road closures, and other events. The collected keywords and information are utilized to build ontology-based semantic knowledge for traffic event detection. The proposed ontology examines the topic-level completeness of traffic event narratives.

The LDA output contains a topic with the most-related keywords. Because LDA is an unsupervised method, the generated topics are not labeled with any valuable and significant names. Therefore, the event type is specified in the proposed ontology (e.g., road incident, slow traffic, collision, and car accident are events related to traffic), which connects an unlabeled generated cluster of topics to a real-world event. To extract words semantically related to the topic, we searched each topic in the proposed ontology employing a Description Logic (DL) query. The result of this query comprises various words about events, and each word has sub-words that are stored for further analysis. We apply Term Frequency ($TF = \frac{n_{fi,d}}{\sum_k n_{fk,d}}$) and Inverse Document Frequency ($IDF = \log\frac{|D|}{N_{fi}}$) approaches in order to understand the significance of these words (Chandrashekar et al., 2018). Here, a word is considered a term of the ontology, and a document is the LDA-based topic cluster. The following equation is used to allocate a weight to each topic:

$$Weight_{fi} = \frac{n_{fi,d}}{\sum_k n_{fk,d}} * \log\frac{|D|}{N_{fi}} \quad (4)$$

Here, $n_{fi,d}$, $\sum_k n_{fk,d}$, |D|, and N_{fi} represent the total number of ontology terms existing in document d, the sum of all the ontology terms in

Algorithm 1. Generative process of LDA-based topic modeling.

Data: Preprocessed corpus data
Results: Topics with strongly relevant words
Begin

1 // LDA-based topic modeling
2 Consider $N \sim$ Poisson (φ), where N is a word sequence in doc D
3 **for each** topic tp ϵ T **do**
4 | Draw $\phi_{tp} \sim Dir\ (\beta)$;
5 **end for**
6 **for each** *doc* ϵ *preprocessed corpus data* **do**
7 | Draw topic $\Theta_d \sim Dir\ (\alpha)$;
8 | **for each** *N* unique word of *doc* (W_{d_i}) **do**
9 | | Draw topic $Tp_d \sim$ Multi (Θ_d), P (Tp_d,n | θd);
10 | | Draw a word W_{d_i} from $p\ (w_n \mid Tp_d)$;
11 | **end for**
12 **end for**

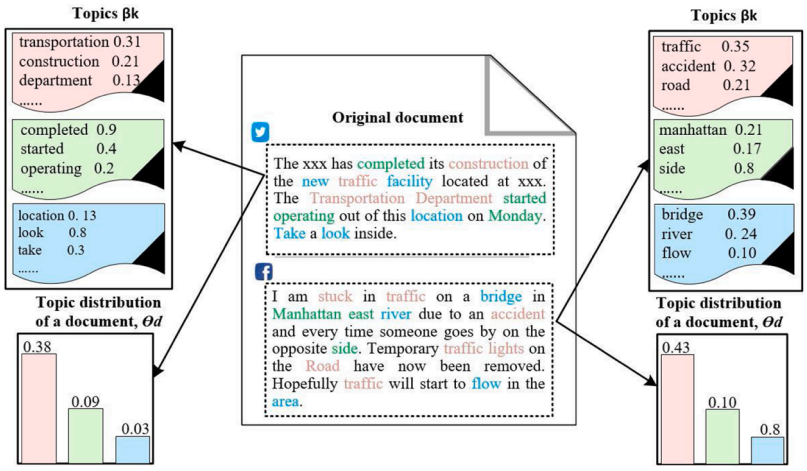


Fig. 4. Example of LDA output.

Table 1
Example traffic event–related keywords.

Accident, driver, car, injuries, crash, traffic, hospital, police, died, vehicle, passenger, killed, closed, injured, dead, struck, jammed, investigation, head, single, bridge, crossing, heavy, light, location, pedestrian, person, road.

document d, the size of the corpus document, and all documents linked with *fi*, respectively. The weight indicates how related the specified event in the ontology is to document d. After finding the weight, the following equation is used to find the similarity between the ontology event and the LDA-based topic:

$$Similarity\ (OntoEvent,\ \mathrm{LDAT}) = \sum_{i=1}^{n} f(ontoF_i, LDAT_i)\ \times weight_i \qquad (5)$$

Here, *OntoEvent*, LDAT, and $weight_i$ indicate specified events in the ontology for the LDA-based generated topic cluster, and the significance of event *i*, respectively. When the similarity score surpasses 4, the first word in the LDA-based topic cluster is considered the topic of the document (i.e., it is semantically linked to the corpus document). The second word of the LDA-based generated cluster is treated as a new topic if the core of the similarity function is less than 4. The abovementioned process is repeated until the relevant topic for the document is detected. If the selected topic belongs to a traffic category, then the document will be labeled *Traffic*; otherwise, it is labeled *Non-Traffic*. The whole scenario is presented in Fig. 5.

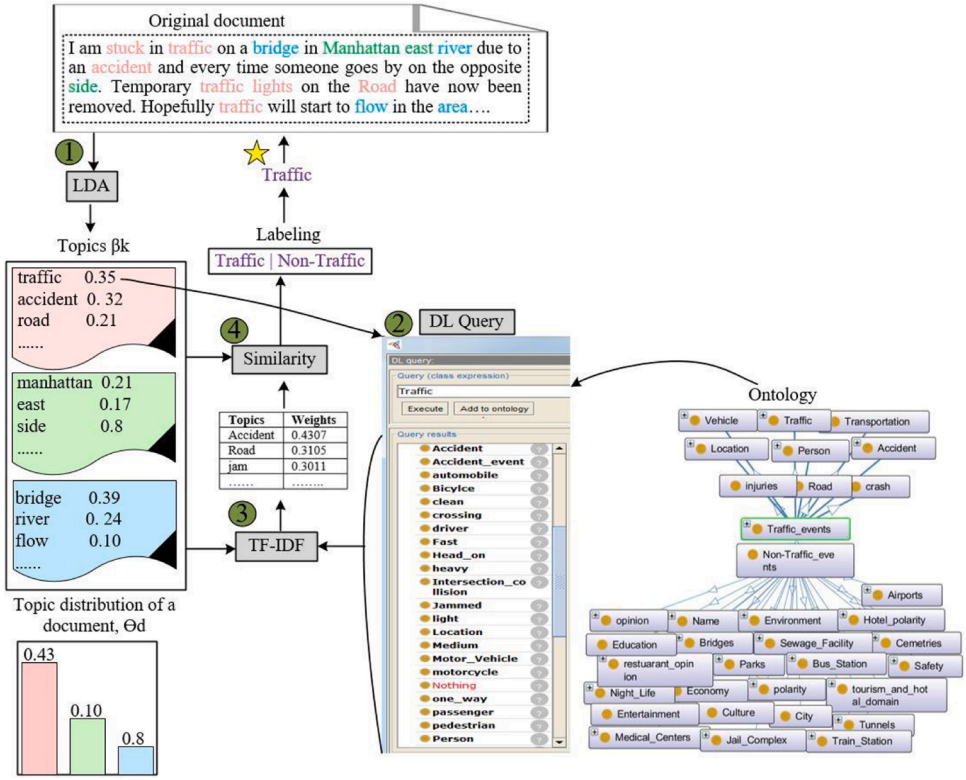


Fig. 5. OLDA-based traffic event labeling.

The results of OLDA-based labeling of the traffic event data are shown in Table 2. As discussed earlier, LDA generates a topic with the most-related words. However, LDA is an unsupervised method, and it cannot automatically label the topics. Human knowledge and judgment are required to investigate the relation and significance of the topics and to intelligently label them. Therefore, after generating topics using LDA, ontology-based semantic knowledge is utilized to label the topics, which are shown in the third column of Table 2. In our research, the data labeled *Non-Traffic* are discarded, and only data labeled *Traffic* are considered for further processing.

3.3.3. Event and location extraction

We developed an ontology-based event extraction system using a collection of traffic event–related entities and their relationships to extract events mentioned in a traffic context. The proposed event extraction system is written using the Protégé OWL and Stanford CoreNLP, written in Java. It is necessary to extract domain-specific event types, such as *traffic accident* and *traffic jam*. Therefore, we deliver event seed words to the ontology, which are discussed above and presented in Table 1. The seed words are keywords that are utilized to extract events from content. In traffic analysis, the location of an event must be identified. However, it is difficult to extract location information because social networking data are informal. They contain special words, and even mistyped words. To solve these problems, we propose a location entity–extraction method using geo-parsing. For each extracted event, the proposed method explores social networking data, and identifies the location name in the textual data.

3.3.4. Event condition analysis using polarity labeling

We applied a sentiment analysis method for event condition analysis. After event and location detection, the system identifies the users' opinion words in each document. It is necessary to find the polarity of all opinion words in each document. Therefore, the system allocates a polarity score to all opinion words using SentiWordNet (Baccianella et al., 2010; Cavalcanti et al., 2011). Each sentence is first preprocessed for POS tagging of each word. Verbs, adjectives, and adverbs are only considered for a search in SWN after POS tagging. SWN first finds the polarity of sentiment words regarding traffic events. The output from finding the polarity of event-related words is then combined to identify the polarity of the whole document. SWN is a dictionary database that connects each *synset* of a *WordNet* to three kinds of score: positive, objective, and negative. It should be noted that SWN allocates no score to an input word if it does not have a meaning for it. We retrieved scores from SWN for each specific opinion word, and then applied the following equations to compute the polarity of each traffic event.

$$P_S\ (E_i) = \sum_{w \in wE_i}^{n} P_S\ SWN_w \tag{6}$$

Table 2
Social networking data and their corresponding labels using OLDA.

	Social network data	Label
1	A two-car crash in Los Angeles has injured one person, traffic congestion in the evening at downtown.	Traffic
2	Heavy traffic due to a traffic accident on NB Western: Kennedy to Fullerton. At least 3 people are killed and 2 injured. Police are investigating a traffic accident, use alternative routes.	Traffic
3	I am stuck in traffic on a bridge in Manhattan east river due to an accident, and every time someone goes by on the opposite side. Temporary traffic lights on the Road have now been removed. Hopefully traffic will start to flow in the area.	Traffic
4	I don't know why I started watching this show!! It's like a horribly slow car crash that won't stop! It just gets worse!	Non-Traffic
5	When it comes to injuries at work, most people may conjure up images of a slip and fall accident, an incident involve.	Non-Traffic
6	Primary Care's Hyden branch closed for week after crash.	Non-Traffic

Table 3
Traffic event types, location detection, and polarity labels.

No.	Traffic event–related sentences	Traffic event type	Location	Polarity
1	A two-car crash in Los Angeles has injured one person, traffic congestion in the evening at downtown.	Car accident	Los Angeles, downtown	Negative
2	Heavy traffic due to a traffic accident on NB Western: Kennedy to Fullerton. At least 3 people are killed and 2 injured. Police are investigating a traffic accident, use alternative routes.	Traffic accident	NB Western, Kennedy and Fullerton	Negative
3	I am stuck in traffic on a bridge in Manhattan east river due to an accident and every time someone goes by on the opposite side. Temporary traffic lights on the Road have now been removed. Hopefully traffic will start to flow in the area.	Traffic accident	Manhattan, east river	Positive

$$Neu_S\ (E_i) = \sum_{w \in wE_i}^{n} Neu_S\ SWN_w \tag{7}$$

$$N_S\ (E_i) = \sum_{w \in wE_i}^{n} N_S\ SWN_w \tag{8}$$

where $P_S\ SWN_w$, $Nue_S\ SWN_w$, and $N_S\ SWN_w$, respectively show positive, neutral, and negative scores for the event. The score is determined for individual word *w* using the arithmetic mean from SWN. Some examples were extensively discussed in (Ali et al., 2018). If $P_S\ (E_i) > N_S\ (E_i)$ *and* $Neu_s\ (E_i)$, then the system determines event polarity to be positive. On the other hand, event polarity is negative if $N_S\ (E_i) > P_S\ (E_i)$ *and* $Neu_S\ (E_i)$. Finally, polarity is neutral if $Neu_S\ (E_i) > P_S\ (E_i)$ *and* $N_S\ (E_i)$. The output from event and location extraction, and the condition analysis using the polarity labeling method, are shown in Table 3.

3.4. Document representation

The traditional document representation method called bag-of-words represents each word of the document as a one-hot vector. The BoW approach represents each word and document by row vector and column vector, respectively. It fills a vector with 1 if the word exists in the document. Otherwise, the vector is filled with 0. However, the dimensionality of one-hot representation is very high. Furthermore, one-hot representation is unable to capture semantic similarities between words. Neural network-based word embedding approaches are an effective method to represent millions of words in a continuous vector space by grouping words with similar meanings. In contrast to one-hot representation, word embedding captures the semantic similarities between words, and represents words with low-dimensional vectors. Various representation methods have been generated and applied to represent each word of the document with a distributed probability (Ali et al., 2019b; Chen et al., 2018; Dabiri and Heaslip, 2019). However, the use of word vectors is a challenging NLP task, such as the classification of social network data. We trained two renowned state-of-the-art models for representation of traffic event data: word2vec and the FastText model.

3.4.1. The word2vec model

The word2vec language model utilizes a neural network approach in

order to learn word embedding from a traffic event dataset with millions of words. It produces a high-dimensional vector space for individual words of the dataset. There are two well-known architectures of the word2vec model: continuous bag-of-words (CBOW) and skip-gram. Both of these models are trained using one hidden layer. However, their learning purposes are different from each other. The learning purpose of CBOW is to predict the middle word in the given context. The skip-gram objective is the reverse of CBOW. Here, the present word is the input, and the surrounding words are the output. In this work, the skip-gram model is trained due to its better work for consistent words. The input layer of skip-gram receives a one-hot vector for each word of the sentence, whereas the output layer generates a probability distribution for all words. Thus, both the input and output layer contain an equal number of neurons corresponding to the number of words in the dataset, represented by *v*. In addition, the size of the hidden layer is fixed to the word dimensionality, represented by *n*. Therefore, the size of the contextual weight matrix, W, is (*v* x *n*). In the skip-gram model, a sequence of words $\{ w_1, w_2, w_3, w_3 \ldots w_T \}$ is received as training words, and the average of the log probability is increased using the following equation (Ali et al., 2019a):

$$\frac{1}{T} = \sum_{t=1}^{T} \sum_{-c \le j \le c, j \ne 0} \log p(w_{t+j} | w_t) \tag{9}$$

where c is the window size of the adjacent word, with classic values 6 or 12. The t, w_t, and w_{t+j} denote the number of unique words in the corpus data, the given word, and the adjacent words, respectively. The skip-gram model represents traffic events in textual data and finds semantic similarities among event-related words. For example, dead is similar in context to died. The top 13,000 words of the traffic event dataset are trained to boost the efficiency of word embedding. After training a traffic event dataset, a 200-dimensional word2vec model is created for document representation. One of the biggest limitations of the word2vec model for social media text is that it assigns a different vector to each word, and ignores the morphology of words. It does not represent a new word by a vector if the word is not available in the training dataset. Social networking data comprise a large amount of text, with both formal and informal words, described by many characters and abbreviations. The word2vec approach is unsuccessful in terms of the needed data for effective learning and memory requirements. Therefore, the above-described approach fails to effectively perform in the context of social networking data.

3.4.2. The FastText model

The FastText model can be utilized to overcome the limitations of the word2vec model (Verheye and Loncke, 2019). This model was developed by Facebook, and transforms words from social networking text into vector representations. The FastText model uses two neural network layers, which is extremely valuable for social media text when considering sub-word information. This model is an extension of skip-gram that breaks the words into a bag of character n-grams and assigns a vector to each character n-gram. Therefore, each word is denoted by the sum of its related n-gram vectors. The FastText model with n-gram characters can produce vectors for those words that are not present in the training data. This model is efficient at enhancing word representations for languages that contain formal and informal words in the form of verbs and nouns.

3.5. Classification based on Bi-LSTM and Softmax regression

In this paper, we compare two deep learning models: the RNN and Bi-LSTM. In addition, we employ softmax regression with deep learning models for the classification of traffic accident–related text.

3.5.1. The RNN model

RNNs have been successfully employed in the tasks of machine translation and natural language processing (Ayata et al., 2017; Gupta et al., 2017; Lipton et al., 2015). They comprise an input layer, a hidden layer, and an output layer. Like the forward neural network, the job of the input and output layers of the RNN is the same. The RNN can be characterized into the conventional RNN and the timeslot RNN. A conventional RNN has no concept of time. Thus, given sentence *s* of *n* words, the conventional RNN only considers the current word in the training stage. In contrast, a timeslot RNN shares information between time steps, as shown in Fig. 6. Let us consider the clause *"heavy traffic due to traffic accident"* as input for an RNN. The timeslot RNN takes input time t = 1 as the token *heavy* for the input sentence, as shown in Fig. 6. Mathematically, the forward RNN for the past and current predictions at each time step can be described as follows (Gupta et al., 2017):

$$h_t = g\left(w_I x_t + w_R\, h_{t-1} + b_h\right) \tag{10}$$

$$h_t = g\left(w_y h_t + b_y\right) \tag{11}$$

where, $g, x_t, w, h_t,$ and b represent activation function, current input, weight, hidden state, and biases, respectively. The tanh activation formula is mostly utilized for g. Furthermore, the softmax activation function is employed in the output layer. Eqs. 10 and 11 calculate the past and current predications, respectively. However, the RNN has no power to learn longstanding dependencies, which happens when the gap between the data in one time step and the required time step become large. This creates exploding and vanishing gradient problems during backpropagation. To address such problems, LSTM was developed to handle longstanding dependencies and overcome vanishing and exploding gradient problems.

3.5.2. Bi-LSTM model

LSTM is an advanced form of a Recurrent Neural Network with the same type of architecture. Both the RNN and LSTM pass the information

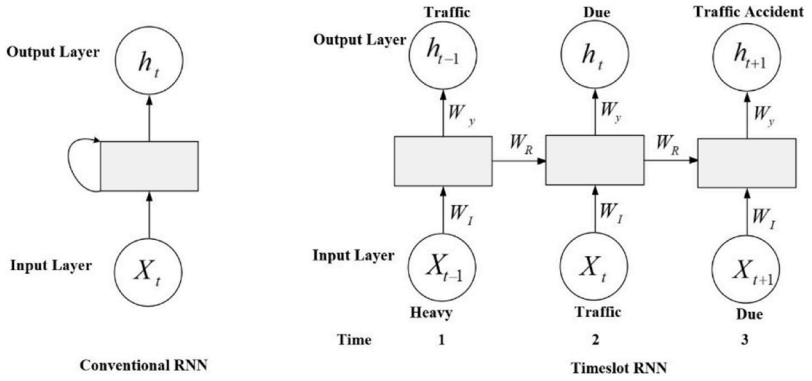


Fig. 6. The architecture of an RNN.

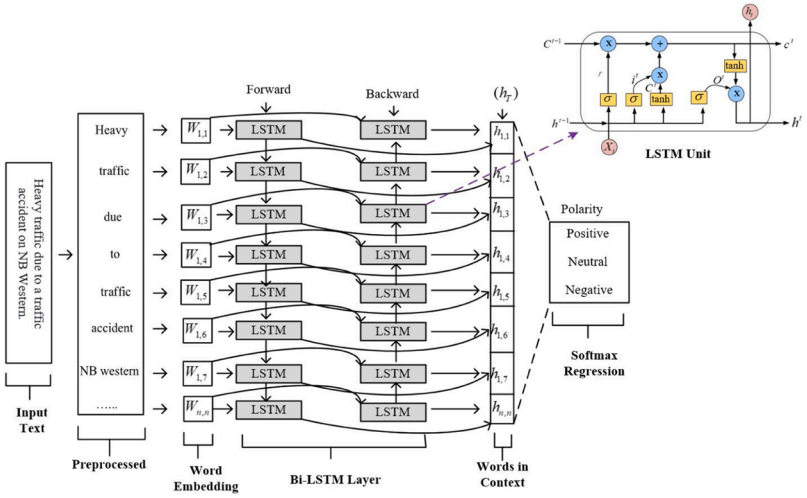


Fig. 7. Bi-LSTM for polarity prediction.

from one stage to another. The LSTM model displays extreme success in longstanding dependencies (Lamurias et al., 2018). However, a single LSTM model is limited to predicting the output based on the previous information. Therefore, a single LSTM–based system may predict the wrong output without knowing the forward information. For example, the input text "heavy… due to traffic accident" lost the word *traffic*. The single LSTM uses only *heavy* to produce the lost word *traffic*. However, the system may produce the wrong word based on the training data, such as *truck*, *instrument*, etc. Therefore, we propose a Bi-LSTM model that contains both past (*heavy*) and upcoming (*traffic*) information, and that can predict the lost word *traffic* easily. Two LSTM models run parallel in Bi-LSTM–based systems, as shown in Fig. 7. One LSTM runs from the start of the input data, and other runs from the end of the input data. This method allows the Bi-LSTM model to hold both previous and upcoming information. For example, when the sequence data vectors are given to the Bi-LSTM model as input, the first and second LSTM models, respectively, learn the data sequence starting at the beginning from left to right, and starting at the end from right to left. In this way, the Bi-LSTM model fully holds the information on traffic events for polarity prediction.

In Fig. 7, the LSTM unit is used in the hidden layers that have the ability to keep previous information for a short time. The main element of an LSTM unit is memory cell C_t, which is updated employing input gate i_t and forget gate f_t. The input gate decides which information should to be kept in the memory cell. The forget gate decides which information should be dumped from the memory cell. At each time step, C_t for the forward LSTM can be updated using the following equations:

$$u_t^{(f)} = tanh\left(w_{xu}^{(f)}x_t + w_{hu}^{(f)}h_{t-1} + b_u^{(f)}\right) \quad (12)$$

$$i_t^{(f)} = \sigma\left(w_{xi}^{(f)}x_t + w_{hi}^{(f)}h_{t-1} + b_i^{(f)}\right) \quad (13)$$

$$f_t^{(f)} = \sigma\left(w_{xf}^{(f)}x_t + w_{hf}^{(f)}h_{t-1} + b_f^{(f)}\right) \quad (14)$$

$$C_t^{(f)} = f_t^{(f)} \odot C_{t-1}^{(f)} + i_t^{(f)} \odot u_t^{(f)} \quad (15)$$

$$O_t^{(f)} = \sigma\left(w_{xo}^{(f)}x_t + w_{ho}^{(f)}h_{t-1} + b_o^{(f)}\right) \quad (16)$$

$$fh_t = O_t^{(f)} \odot tanh\left(C_t^{(f)}\right) \quad (17)$$

At each time step, C_t for the backward LSTM can be updated using the following equations:

$$u_t^{(b)} = tanh\left(w_{xu}^{(b)}x_t + w_{hu}^{(b)}h_{t+1} + b_u^{(b)}\right) \quad (18)$$

$$i_t^{(b)} = \sigma\left(w_{xi}^{(b)}x_t + w_{hi}^{(b)}h_{t+1} + b_i^{(b)}\right) \quad (19)$$

$$f_t^{(b)} = \sigma\left(w_{xf}^{(b)}x_t + w_{hf}^{(b)}h_{t+1} + b_f^{(b)}\right) \quad (20)$$

$$C_t^{(b)} = f_t^{(b)} \odot C_{t-1}^{(b)} + i_t^{(b)} \odot u_t^{(b)} \quad (21)$$

$$O_t^{(b)} = \sigma\left(w_{xo}^{(b)}x_t + w_{ho}^{(b)}h_{t+1} + b_o^{(b)}\right) \quad (22)$$

$$bh_t = O_t^{(b)} \odot tanh\left(C_t^{(b)}\right) \quad (23)$$

where $w_{xi}, b_i, w_{xu}, w_{hu}, w_{xo}, b_o, w_{xf}$, and b_f are parameters to be learned, and x_t is the input data of the LSTM model. The output of the forward and backward LSTM is fh_t and bh_t, respectively. For each time step t, the final output of Bi-LSTM h_T is generated by combining the forward and backward LSTM units as follows:

$$h_T = w_x^{(h)} fh_t + w_h^{(h)} bh_t + b^{(h)} \quad (24)$$

The Bi-LSTM contains fh_t and bh_t, which describe the past and future information, respectively. Bi-LSTM combines the features of the past and future context, and considers them the output of the model.

3.5.3. Softmax regression

The final layer of the proposed model is softmax regression. It generates the results of multiple classes in the form of probability. Softmax regression takes the output of the Bi-LSTM model, h_T, as input, which is still a vector of dimension d. As discussed above, we predict polarity *pol* for each sentence in document D with words N. The polarity $pol^{'}$ can be predicted as follows:

$$p(polX) = softmax\ (X) = \frac{\exp(X)}{\sum_{j=0}^{i}\exp(X^i)}\ ,\ X = W_s h_T + b_s \quad (25)$$

In the above equation, X and i describe the input of time step j and the sentiment category, respectively. The maximum probability as the predicted polarity of the input data is as follows:

$$pol^{'} = arg\ \max_{pol} p(pol|x_k) \quad (26)$$

then utilize cross entropy to compute the loss function for all the labeled sentences, as follows:

$$\text{Loss} = \frac{1}{M}\sum_{s=1}^{M} Y_s.\ \log p(pol_s|X_s) \quad (27)$$

where s represents the nth input sentence. The softmax regression

identifies the probability outputs in the interval 0–1. The probability output shows that the polarity of traffic events is positive, neutral, or negative.

4. Experiments

The data obtained from social networking sites regarding traffic events were discussed in Subsection 3.1. The data preprocessing for the use of topic modeling methods and classifier models was explained in Subsection 3.2. The proposed approaches were presented in subsections 3.3–3.5. This section evaluates the proposed model's performance using real-world data on non-recurring traffic events.

4.1. Performance evaluation

In this section, we discuss the various experiments conducted in order to show how well the proposed OLDA can assign topics, and how well Bi-LSTM can predict the polarity of traffic events.

Dataset: We developed a crawler based on Twitter and Facebook APIs to collect data about traffic events. First, the most populous metropolitan cities in the U.S.A. were selected, where non-recurring traffic events mostly happen: New York, Los Angeles, and Chicago. Then, the most useful 500 keywords related to traffic events were employed to construct different types of queries. These keyword-based queries were applied to collect the most relevant data on non-recurring traffic events from social networking platforms. In our dataset, we collected 600,000 tweets and 5000 posts from Twitter and Facebook, respectively. The dataset obtained from social networking sites were extensively discussed in Section 3.1. The proposed framework is based on ontology, which contains expert knowledge about traffic events. Therefore, this approach can only be applied to collect and handle traffic-related data from any social application. The collected data were analyzed using a preprocessing module. The dataset was then automatically labeled using the proposed method as discussed in Section 3.3. The datasets were divided into two subsets: training and testing. Datasets of 70 % and 30 % were used for training and testing, respectively, of the classifier models.

To evaluate the performance of the proposed OLDA model, we first determined the optimum number of topic keywords (Tp) in OLDA clustering based on different numbers of iterations. The OLDA was then compared with other topic modeling approaches: LDA and SLDA (Fu et al., 2015; Li et al., 2015; Zhang et al., 2018). Furthermore, we trained well-known word embedding models (string2word, word2vec, and FastText) for representation of the traffic event data. The proposed Bi-LSTM model and other classifiers (random forest [RF], the SVM, logistic regression [LR], the RNN, and LSTM) were applied to predict the polarity of traffic events based on the trained embedding models. The performance of the proposed Bi-LSTM model was then compared with other classifiers. Furthermore, the proposed Bi-LSTM model and the other classifiers were evaluated based on individual data from three cities and with combined data. This system was implemented using Waikato Environment for Knowledge Analysis (Weka) and the Protégé OWL tool with Java. Performance was measured based on accuracy (Acc), precision (Pre), recall (Rec), and function measure (F-measure).

4.2. Evaluation results

This section presents the experimental results described in Subsection 4.1. First, the results of 100, 200, 400, and 600 iterations were evaluated based on the accuracy with different numbers for Tp in clustering. The proposed OLDA was then compared with other topic modeling approaches using traffic event datasets under Tp = 12. The accuracy of these approaches for both traffic and non-traffic data was compared in order to identify the best approach to traffic event detection. In addition, Pre, Rec, and F-measure (FM) of the proposed OLDA were also compared against LDA and SLDA.

4.2.1. Test results from different numbers of topic keywords in clustering

The accuracy of topic modeling is also affected by the number of topic keywords in clustering. After data preprocessing, various numbers of keywords were used to find the optimum set of keywords in clustering for topic modeling. In the first experiment, we tested keyword sets that comprised 4, 8, 12, 16, and 20 keywords. Fig. 8 illustrates the accuracy of OLDA-based clustering for the aforementioned keyword sets when the number of iterations was 100, 200, 400, and 600. As can be seen, all the iterations for a set comprising 12 keywords achieved higher clustering accuracy than other keyword sets. In Fig. 8, we circled in blue the clustering accuracy of all iterations for the 12-keyword set. The achieved results show that the performance of OLDA is better for topic modeling with a 12-keyword set in clustering than with other keyword sets.

4.2.2. Results comparison of the proposed OLDA with other class labeling approaches

In the second experiment, we performed topic modeling at Tp = 12 using traffic event datasets. The main goal was to allocate a class label (*traffic* or *non-traffic*) to each sentence. Fig. 9 illustrates the obtained results from all topic modeling approaches on the traffic event datasets. In more detail, the figure presents the performance metrics for each topic modeling approach. According to Fig. 9, the proposed OLDA obtained the best results, with Acc, Pre, Rec, and FM at 80 %, 76.6 %, 77.5 %, and 77.1 %, respectively. The obtained results indicate that the proposed model can accurately label sentences if precise datasets with

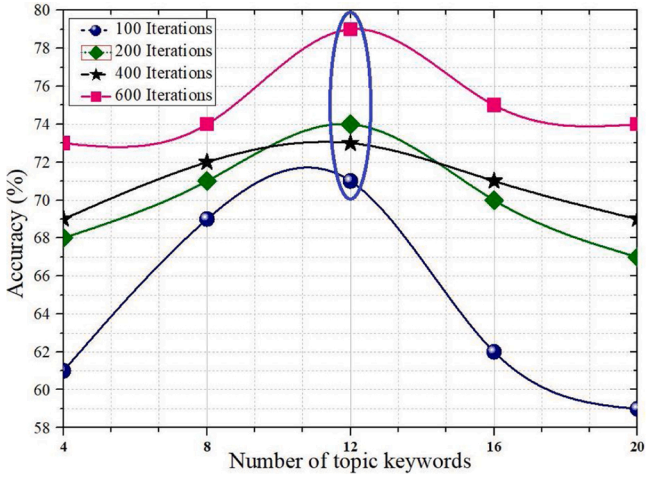


Fig. 8. Accuracy of OLDA-based clustering with different numbers of topic keywords.

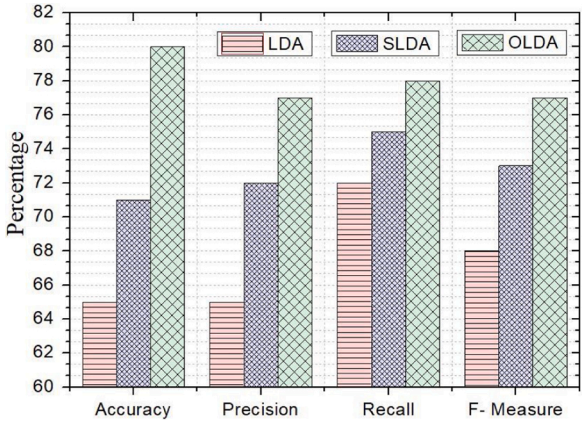


Fig. 9. Results comparison of topic modeling approaches compared to the proposed model.

Table 4
Polarity prediction results of the classifier models with word2vec and FastText.

Classifier model	Word2vec				FastText			
	Acc	Pre	Rec	FM	Acc	Pre	Rec	FM
RF	56.6	77.5	56.7	65.4	63.3	64.4	63.3	63.2
SVM	68.3	79.4	68.4	73.4	71.6	71.8	71.7	71.6
LG	70.9	72.6	71.0	71.7	74.8	75.6	74.8	75.1
RNN	75.4	83.5	75.5	79.2	80.0	80.6	80.0	80.2
LSTM	83.3	83.3	83.3	83.2	85.0	88.0	85.0	86.4
Bi-LSTM	88.3	88.9	91.4	90.1	**93.5**	92.4	90.4	91.3

the correct number of keywords are assigned to the proposed OLDA. The obtained results also illustrate that the proposed OLDA can perform better than LDA and SLDA in terms of social networking handling for topic modeling or class labeling.

4.3. Polarity prediction results for condition analysis of traffic events

In the third experiment, the proposed Bi-LSTM model was compared with RF, LG, the SVM, the RNN, and LSTM in terms of polarity prediction of traffic events. It should be noted that the RNN, LSTM, Bi-LSTM, word2vec, and FastText are extensively discussed, along with their parameters, in Sections 3.4 and 3.5. We used RF with 100 iterations and seed 1, and LG and SVM with a training parameter ridge estimator and a radial basis function, respectively (Ali et al., 2020, 2019b). The results obtained from the proposed Bi-LSTM and the other five classifier models are shown in Table 4, which presents the performance metrics of the models. These classifiers were applied to predict the polarity of traffic events using two types of word embedding models: word2vec and FastText. The results of all baseline models were compared in order to evaluate the performance of word embedding models. Bi-LSTM achieved higher classification accuracy of 88.3 %, in comparison to other classifiers using the word2vec model. Furthermore, the accuracy of all classifiers largely increased when using the FastText model. As shown in Table 4, the baseline model accuracy increased, compared to the accuracy of these models using word2vec. The highest accuracy (at 93.5 %) was from the proposed Bi-LSTM. We can see that FastText outperforms word2vec in terms of polarity prediction of traffic events. In addition, the results of this experiment indicate that the FastText model can precisely represent traffic event data with a very low-dimensional vector for ML models. These results also show that Bi-LSTM with FastText can accurately predict the conditions of traffic events.

In the fourth experiment, we made different models by combining OLDA with word embedding models and ML classifiers. After topic modeling, the pre-trained word embedding models with ML classifiers were utilized in order to predict the polarity of traffic events. The performance of these models was evaluated using datasets from three different cities, namely New York, Los Angeles, and Chicago. These cities have different rates of traffic accidents, and different road conditions and population densities. Table 5 presents the polarity prediction results using OLDA with different word embedding models and ML classifiers: OLDA + word2Vec + RNN, OLDA + word2Vec + LSTM, OLDA + word2Vec + Bi-LSTM, OLDA + FastText + RNN, OLDA + FastText + LSTM, and OLDA + FastText + Bi LSTM. The achieved results show that the proposed OLDA performed well with the FastText and Bi-LSTM model. The results also indicate that OLDA with FastText improved the performance of the baseline models in terms of polarity prediction. As can be seen, the accuracy of the RNN, LSTM, and Bi-LSTM with OLDA and FastText was higher by 5.8 %, 3.3 %, and 5.8 % with respect to the RNN, LSTM, and Bi-LSTM with OLDA and word2Vec, respectively, using the New York dataset. We see that OLDA with FastText and Bi-LSTM was more efficient for polarity prediction than OLDA with other word embedding and classifier models. The accuracy of each model fluctuated as the dataset changed. This may be due to the variances in traffic events of the different cities. However, the best accuracy obtained by the proposed model was 90 %, 92.9 %, and 94.4 % for New York, Los Angeles, and Chicago, respectively. The obtained results show that the proposed model can precisely handle, represent, and classify traffic event–related datasets from different areas.

In the fifth experiment, the performance of all models was evaluated using a combined dataset of the three different cities. Fig. 10 shows the results of polarity prediction for the traffic events. As illustrated, the best accuracy was achieved with the proposed OLDA + FastText + Bi-LSTM (97 %). This accuracy is higher by 5%, 12 %, 6%, 12 %, and 17 % compared to OLDA + word2Vec + RNN, OLDA + word2Vec + LSTM, OLDA + word2Vec + Bi-LSTM, OLDA + FastText + RNN, and OLDA + FastText + LSTM, respectively. The results show that integration with the OLDA model can provide better performance than models without OLDA (as compared to the results in Table 4). For example, the prediction accuracy of OLDA with FastText and Bi-LSTM was much higher than FastText and Bi-LSTM without OLDA. The obtained results also indicate that Bi-LSTM and LSTM models outperformed the RNN. For example, OLDA + FastText + Bi-LSTM and OLDA + FastText + LSTM outperformed the RNN. Furthermore, Bi-LSTM outperformed all other classifiers in all the experiments, which shows that the proposed model with a high-volume dataset can perform better than other models in terms of polarity prediction of traffic events.

4.4. Computational cost of the proposed model

The computational cost of the proposed model is examined in this section. Fig.11 shows the running time and comparison of the proposed model. We first extracted the data about traffic jam from the whole dataset. The submodules of the proposed system are then applied to predict the polarity of traffic jam. Fig. 11 (a) shows the running time of all the subsequent tasks for the polarity prediction of the single traffic event. Data preprocessing, topic and polarity labeling, word embedding, and polarity classification takes 2.2, 4.1, 5, and 6.3 s, respectively. Fig. 11 (b) shows the comparison of the proposed model with other models of traffic event detection and condition analysis using combined datasets. As it can be seen in Fig. 11 (b), the proposed OLDA + FastText + Bi-LSTM is slightly slower than other models. This is due to two main reasons. First, FastText is an extension of word2Vec model, which utilizes n-gram characters to produce vectors for those words that are not present in the training data. Second, simple LSTM runs only on input sequence, whereas Bi-LSTM runs two LSTM in parallel, one on input sequence and other on the reverse of the input sequence. Therefore, both FastText and Bi-LSTM takes more time to produce extra

Table 5
Traffic event detection and condition analysis results of all models using different area datasets.

Models	New York				Los Angeles				Chicago			
	Acc	Pre	Rec	FM	Acc	Pre	Rec	FM	Acc	Pre	Rec	FM
OLDA + word2Vec + RNN	74.8	73.9	70.8	72.3	77.0	76.4	73.3	74.8	79.2	80.0	72.3	76.0
OLDA + word2Vec + LSTM	81.1	81.6	75.5	78.5	83.7	84.2	83.7	83.6	82.9	85.2	76.7	80.7
OLDA + word2Vec + Bi-LSTM	84.2	86.0	84.2	84.5	87.7	76.9	95.2	85.1	91.9	84.0	95.5	89.4
OLDA + FastText + RNN	80.6	80.0	76.6	78.3	82.5	82.7	82.6	82.5	83.7	84.4	83.7	83.6
OLDA + FastText + LSTM	84.4	84.5	81.7	83.1	86.0	83.9	86.7	85.2	89.7	90.4	89.7	89.7
OLDA + FastText + Bi-LSTM	**90.0**	92.4	90.4	91.3	**92.9**	87.0	95.2	90.9	**94.4**	86.4	100	92.7

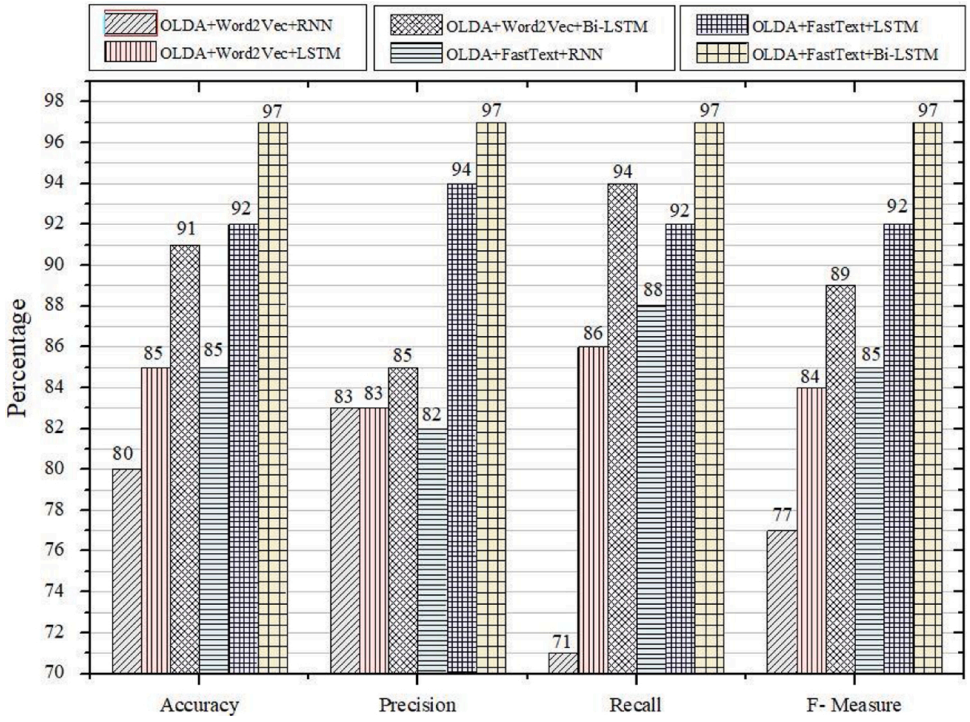
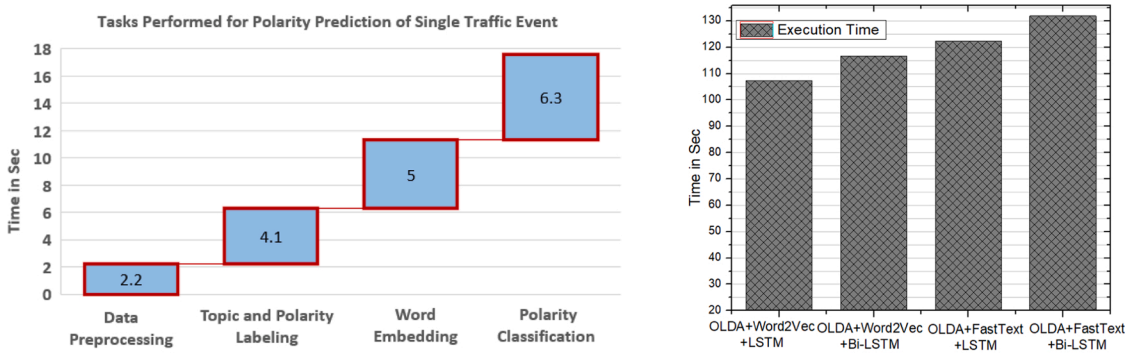


Fig. 10. Comparison of the proposed model with other models based on combined datasets.



(a) (b)

Fig. 11. Computational cost of the proposed method: (a) time required for all the subsequent tasks for the polarity prediction of single traffic event and (b) comparison with other models for traffic event detection and condition analysis.

Table 6
Detailed comparison of existing methods using traffic event datasets retrieved from social media.

Rank	Authors / Year	Class labeling methods	Word embedding models	Classifiers	Overall Acc / FM (%)
1	Cao et al. (Cao et al., 2018) / 2018	Manually (Traffic / Non-traffic)	Traditional methods (bag-of-words)	GRU	56.6 / -
2	Ali et al. (Ali et al., 2019a) / 2019	Manually (Positive / Negative)	Word2vec + fuzzy ontology	Bi-LSTM	84.0 / -
3	Zhang et al. (Zhang et al., 2018) / 2018	Manually (Accident / Non-accident)	Traditional methods (bag-of-words)	DBN	85.1 / -
4	Angelica-Salas et al. (Salas et al., 2018) / 2018	Manually (Traffic/ Non-traffic)	Traditional methods (bag-of-words)	SVM	88.2 / 88.6
5	Ali et al. (Ali et al., 2017) / 2017	Manually (Positive / Negative)	Traditional methods (bag-of-words)	Fuzzy ontology	89.0 / 93.6
6	Chen et al. (Chen et al., 2018) / 2019	Manually (Traffic/ Non-traffic)	Word2vec model	CNN + LSTM	- / 90.5
7	Ali et al. (Ali et al., 2019b) / 2019	Manually (Positive / Negative)	Word2vec model	Deep learning	93.0 / 88.0
8	**Proposed method**	OLDA-based automatic labeling (Traffic/ Non-traffic)	FastText	Bi-LSTM	97.0 / 97.0

vectors of word and learn past and future information about traffic events, respectively.

4.5. Comparison with existing systems

We compared our proposed system against state-of-the-art methods in terms of traffic event detection and polarity prediction, as presented in Table 6. Cao et al. (Cao et al., 2018) utilized a bag-of-words model with a Gated Recurrent Unit (GRU) for traffic sentiment classification, and achieved accuracy of 56.6 %. Ali et al. (Ali et al., 2019a) represented traffic feature–related text using word2vec with a fuzzy ontology, classifying them with Bi-LSTM, and obtained accuracy of 84.0 %. Zhang et al. (Zhang et al., 2018) applied a bag-of-words model with a Deep Belief Network (DBN) for traffic accident detection and obtained accuracy of 85.1 %. Angelica-Salas et al. (Salas et al., 2018) employed a bag-of-words model with an SVM for traffic incident detection, and achieved accuracy and F-measure of 88.2 % and 88.6 %, respectively. Ali et al. (Ali et al., 2017) utilized traditional NLP approaches to text representation and a fuzzy ontology for polarity classification, and obtained accuracy and F-measure of 89 % and 93.6 %, respectively. Chen et al. (Chen et al., 2018) attained an overall F-measure of 90.5 % by utilizing the word2vec model with a CNN and LSTM for traffic information detection. Ali et al. (Ali et al., 2019b) employed the word2vec model with deep learning for polarity classification of traffic features, and achieved accuracy and F-measure of 93 % and 88 %, respectively.

As can be clearly seen in Table 6, the proposed system significantly outperformed the existing systems. Our system's accuracy is shown in the last row of Table 6. In this experiment, OLDA and FastText, respectively, were used to automatically label and precisely represent the data, whereas Bi-LSTM was applied for polarity prediction. The proposed system achieved individual accuracy of 80 % and 97 % for topic labeling and polarity prediction, respectively. The obtained results show that the proposed techniques might help in the development of smart transportation systems for traffic event detection and polarity prediction.

5. Conclusion

In this paper, we presented a social networking–based, real-time traffic monitoring framework using an OLDA/Bi-LSTM model to improve the accuracy of traffic event detection and condition analysis, and to help the transport management sector and state police reduce fatalities and restore traffic flow as easily and quickly as possible. Several sensible issues were discussed, including traffic-information collection from social networks using query-based crawling, transformation of unstructured data into structured form using a data preprocessing module, OLDA-based class labeling of each sentence, exact traffic-related entity extraction using an ontology, polarity identification of traffic events based on a sentiment analysis method, document representation employing word embedding models, and traffic event detection and condition analysis based on deep learning models. The proposed framework offers a detection and analysis system that extracts the most valuable traffic information from unstructured data, and labels and represents them to accurately detect and analyze traffic events. Indeed, the proposed OLDA automatically labels each sentence to effectively identify the exact information on the traffic event. The sentiment analysis technique identifies the polarity of the traffic event, which helps know accurate conditions of the traffic event. The proposed word embedding model transforms formal and informal words into low-dimensional vector representations in order to improve the accuracy of classification tasks. This novel framework not only identifies the traffic event–related data but also analyzes them to find their exact condition, which enhances the performance of traffic monitoring systems. Also, this technique may be linked to various information extraction and class labeling systems, text representation models, and polarity prediction systems, since it extracts meaningful data from unstructured data, labels the data accurately, and represents these labeled data with low-dimensional vectors in order to improve the performance of traffic event detection and condition analysis.

In future work, the performance of information extraction and topic modeling can be further improved to extract more desirable data, and reduce the complexity of class labeling. Furthermore, a new word embedding model called BERT will be trained for document representation to more easily handle social networking data. Finally, a more refined technique will be investigated for event extraction and event polarity prediction to achieve effective results.

CRediT authorship contribution statement

Farman Ali: Conceptualization, Methodology, Software, Visualization, Writing - original draft. **Amjad Ali:** Resources, Validation, Formal analysis, Software. **Muhammad Imran:** Investigation, Writing - review & editing. **Rizwan Ali Naqvi:** Writing - review & editing. **Muhammad Hameed Siddiqi:** Data curation, Resources. **Kyung-Sup Kwak:** Funding acquisition, Project administration, Supervision.

Declaration of Competing Interest

The authors report no declarations of interest.

Acknowledgments

This work was supported in part by a National Research Foundation of Korea grant funded by the Korean Government (Ministry of Science and ICT-NRF-2020R1A2B5B02002478), and in part by Sejong Universitythrough its faculty research program. Imran's work was supported by the Deanship of Scientific Research at King Saud University through research group project number RG-1435-051.

References

Ali, F., Kwak, D., Khan, P., Islam, S.M.R., Kim, K.H., Kwak, K.S., 2017. Fuzzy ontology-based sentiment analysis of transportation and city feature reviews for safe traveling. Transp. Res. Part C Emerg. Technol. 77 https://doi.org/10.1016/j.trc.2017.01.014.

Ali, F., Ei-Sappagh, S., Khan, P., Kwak, K.S., 2018. Feature-based transportation sentiment analysis using fuzzy ontology and SentiWordNet. 9th Int. Conf. Inf. Commun. Technol. Converg. ICT Converg. Powered by Smart Intell. ICTC 2018 D 1350–1355. https://doi.org/10.1109/ICTC.2018.8539607.

Ali, F., El-Sappagh, S., Kwak, D., 2019a. Fuzzy ontology and LSTM-based text mining: a transportation network monitoring system for assisting travel. Sensors (Switzerland) 19, 2. https://doi.org/10.3390/s19020234.

Ali, F., Kwak, D., Khan, P., El-Sappagh, S., Ali, A., Ullah, S., Kim, K.H., Kwak, K.-S., 2019b. Transportation sentiment analysis using word embedding and ontology-based topic modeling. Knowledge-Based Syst. https://doi.org/10.1016/j.knosys.2019.02.033 xxxx.

Ali, F., El-Sappagh, S., Islam, S.M.R., Kwak, D., Ali, A., Imran, M., Kwak, K.-S., 2020. A smart healthcare monitoring system for heart disease prediction based on ensemble deep learning and feature fusion. Inf. Fusion 63 (December 2019), 208–222. https://doi.org/10.1016/j.inffus.2020.06.008.

Aqib, M., Mehmood, R., Alzahrani, A., Katib, I., 2020. In-Memory Deep Learning Computations on GPUs for Prediction of Road Traffic Incidents Using Big Data Fusion. https://doi.org/10.1007/978-3-030-13705-2_4.

Arauz, P.L., Gomez-Romero, J., Bobillo, F., 2012. A fuzzy ontology extension of wordnet and eurowordnet for specialized knowledge. 10th Terminol. Knowl. Eng. Conf. New Front. Constr. Symbiosis Terminol. Knowl. Eng. TKE 2012 139–154. June.

Ayata, D., Saraclar, M., Ozgur, A., 2017. BUSEM at SemEval-2017 task 4A sentiment analysis with word embedding and Long short term memory RNN approaches. Proc. 11th Int. Work. Semant. Eval. 775–781.

Baccianella, S., Esuli, A., Sebastiani, F., 2010. SentiWordNet 3. 0: an enhanced lexical resource for sentiment analysis and opinion mining SentiWordNet. Analysis 0, 1–12 doi:10.1.1.61.7217.

Bastani, K., Namavari, H., Shaffer, J., 2019. Latent Dirichlet allocation (LDA) for topic modeling of the CFPB consumer complaints. Expert Syst. Appl. 127, 256–271. https://doi.org/10.1016/j.eswa.2019.03.001.

Cao, J., Zeng, K., Wang, H., Cheng, J., Qiao, F., Wen, D., Gao, Y., 2014. Web-based traffic sentiment analysis: methods and applications. IEEE trans. Intell. Transp. Syst. 15 (2), 844–853. https://doi.org/10.1109/TITS.2013.2291241.

Cao, D., Wang, S., Lin, D., 2018. Chinese microblog users' sentiment-based traffic condition analysis. Soft Comput. 22 (21), 7005–7014. https://doi.org/10.1007/s00500-018-3293-8.

Cavalcanti, D.C., Prudêncio, R.B.C., Pradhan, S.S., Shah, J.Y., Pietrobon, R.S., 2011. Good to be bad? Distinguishing between positive and negative citations in scientific impact. Proc. - Int. Conf. Tools with Artif. Intell. ICTAI June 2014 156–162. https://doi.org/10.1109/ICTAI.2011.32.

Chandrashekar, M., Nagulapati, R., Lee, Y., 2018. Ontology mapping framework with feature extraction and semantic embeddings. Proc. - 2018 IEEE Int. Conf. Healthc. Informatics Work. ICHI-W 2018 34–42. https://doi.org/10.1109/ICHI-W.2018.00012.

Chen, C., Luan, T.H., Guan, X., 2017. Connected Vehicular Transportation. July, pp. 2–14.

Chen, Y., Lv, Y., Member, S., Wang, X., Li, L., Member, S., Wang, F., 2018. Texts with deep learning approaches. IEEE Trans. Intell. Transp. Syst. PP 8, 1–10. https://doi.org/10.1109/TITS.2018.2871269.

D'Andrea, E., Ducange, P., Lazzerini, B., Marcelloni, F., 2015. Real-time detection of traffic from twitter stream analysis. IEEE trans. Intell. Transp. Syst. 16 (4), 2269–2283. https://doi.org/10.1109/TITS.2015.2404431.

Dabiri, S., Heaslip, K., 2019. Developing a Twitter-based traffic event detection model using deep learning architectures. Expert Syst. Appl. 118, 425–439. https://doi.org/10.1016/j.eswa.2018.10.017.

Fu, Y., Yan, M., Zhang, X., Xu, L., Yang, D., Kymer, J.D., 2015. Automated classification of software change messages by semi-supervised Latent Dirichlet Allocation. Inf. Softw. Technol. 57 (1), 369–377. https://doi.org/10.1016/j.infsof.2014.05.017.

Goh, Y.M., Ubeynarayana, C.U., 2017. Construction accident narrative classification: an evaluation of text mining techniques. Accid. Anal. Prev. 108, 122–130. https://doi.org/10.1016/j.aap.2017.08.026. May.

Gu, Y., Qian, Z., Chen, F., 2016. From Twitter to detector: real-time traffic incident detection using social media data. Transp. Res. Part C Emerg. Technol. 67, 321–342. https://doi.org/10.1016/j.trc.2016.02.011.

Guerreiro, G., Figueiras, P., Silva, R., Costa, R., Jardim-Goncalves, R., 2016. An architecture for big data processing on intelligent transportation systems. An application scenario on highway traffic flows. 2016 IEEE 8th Int. Conf. Intell. Syst. IS 2016 - Proc. 65–71. https://doi.org/10.1109/IS.2016.7737393.

Guerrero-Ibáñez, J., Zeadally, S., Contreras-Castillo, J., 2018. Sensor technologies for intelligent transportation systems. Sensors (Switzerland) 18 (4), 1–24. https://doi.org/10.3390/s18041212.

Gupta, S., Namavari, A., Smith, T.O., 2017. Word Sense Disambiguation Using Skip-Gram and LSTM Models, pp. 1–9.

Gutierrez, C., Figuerias, P., Oliveira, P., Costa, R., Jardim-Goncalves, R., 2015. Twitter mining for traffic events detection. Proc. 2015 Sci. Inf. Conf. SAI 2015 371–378. https://doi.org/10.1109/SAI.2015.7237170.

Hoseinzadeh, N., Liu, Y., Han, L.D., Brakewood, C., Mohammadnazar, A., 2020. Quality of location-based crowdsourced speed data on surface streets: a case study of Waze and Bluetooth speed data in Sevierville. TN. Comput. Environ. Urban Syst. 83, 101518. https://doi.org/10.1016/j.compenvurbsys.2020.101518. June.

Kim, E.H.J., Jeong, Y.K., Kim, Y., Kang, K.Y., Song, M., 2015. Topic-based content and sentiment analysis of Ebola virus on Twitter and in the news. J. Inf. Sci. 42 (6), 763–781. https://doi.org/10.1177/0165551515608733.

Lamurias, A., Clarke, L.A., Couto, F.M., 2018. BO-LSTM: classifying relations via long short-term memory networks along biomedical ontologies. bioRxiv 336719. https://doi.org/10.1101/336719.

Li, X., Ouyang, J., Zhou, X., Lu, Y., Liu, Y., 2015. Supervised labeled latent dirichlet allocation for document categorization. Appl. Intell. 42 (3), 581–593. https://doi.org/10.1007/s10489-014-0595-0.

Lin, Y., Li, R., 2020. Real-time traffic accidents post-impact prediction: based on crowdsourcing data. Accid. Anal. Prev. 145, 1–11. https://doi.org/10.1016/j.aap.2020.105696. July.

Lin, Y., Li, L., Jing, H., Ran, B., Sun, D., 2020. Automated traffic incident detection with a smaller dataset based on generative adversarial networks. Accid. Anal. Prev. 144, 105628. https://doi.org/10.1016/j.aap.2020.105628. September 2019.

Lipton, Z.C., Kale, D.C., Elkan, C., Wetzel, R., 2015. Learning to Diagnose with LSTM Recurrent Neural Networks, pp. 1–18. https://doi.org/10.14722/ndss.2015.23268.

Lu, H., Shi, K., Zhu, Y., Lv, Y., Niu, Z., 2018. Sensing urban transportation events from multi-channel social signals with the Word2vec fusion model. Sensors (Basel). 18, 12. https://doi.org/10.3390/s18124093.

Nallaperuma, D., Nawaratne, R., Bandaragoda, T., Adikari, A., Nguyen, S., Kempitiya, T., De Silva, D., Alahakoon, D., Pothuhera, D., 2019. Online incremental machine learning platform for big data-driven smart traffic management. IEEE trans. Intell. Transp. Syst. 20 (12), 4679–4690. https://doi.org/10.1109/TITS.2019.2924883.

Pereira, J., Pasquali, A., Saleiro, P., Rossetti, R., 2017. Transportation in social media: an automatic classifier for travel-related tweets. Lect. Notes Comput. Sci. (including Subser. Lect. Notes Artif. Intell. Lect. Notes Bioinformatics) 10423 LNAI 355–366. https://doi.org/10.1007/978-3-319-65340-2_30.

Poria, S., Cambria, E., Gelbukh, A., 2016. Aspect extraction for opinion mining with a deep convolutional neural network. Knowledge-Based Syst. 108, 42–49. https://doi.org/10.1016/j.knosys.2016.06.009.

Ren, G., Hong, T., 2017. Investigating online destination images using a topic-based sentiment analysis approach. Sustain. 9, 10. https://doi.org/10.3390/su9101765.

Reynard, D., Shirgaokar, M., 2019. Harnessing the power of machine learning: can Twitter data be useful in guiding resource allocation decisions during a natural disaster? Transp. Res. Part D Transp. Environ. 77 (March), 449–463. https://doi.org/10.1016/j.trd.2019.03.002.

Roque, C., Lourenço Cardoso, J., Connell, T., Schermers, G., Weber, R., 2019. Topic analysis of Road safety inspections using latent dirichlet allocation: a case study of roadside safety in Irish main roads. Accid. Anal. Prev. 131 (April), 336–349. https://doi.org/10.1016/j.aap.2019.07.021.

Salas, A., Georgakis, P., Petalas, Y., 2018. Incident detection using data from social media. IEEE Conf. Intell. Transp. Syst. Proceedings, ITSC 2018-March 751–755. https://doi.org/10.1109/ITSC.2017.8317967.

Serrano-Guerrero, J., Chiclana, F., Olivas, J.A., Romero, F.P., Homapour, E., 2020. A T1OWA fuzzy linguistic aggregation methodology for searching feature-based opinions. Knowledge-Based Syst. 189, 105131. https://doi.org/10.1016/j.knosys.2019.105131 xxxx.

Valdivia, A., Luzón, M.V., Cambria, E., Herrera, F., 2018. Consensus vote models for detecting and filtering neutrality in sentiment analysis. Inf. Fusion 44 (October 2017), 126–135. https://doi.org/10.1016/j.inffus.2018.03.007.

Vallejos, S., Alonso, D.G., Caimmi, B., Berdun, L., Armentano, M.G., Soria, Á., 2020. Mining social networks to detect traffic incidents. Inf. Syst. Front. Barth 2009. https://doi.org/10.1007/s10796-020-09994-3.

Verheye, J., Loncke, S., 2019. Influence of Twitter Sentiment Using Word Representations on.

Wang, S., Zhang, X., Cao, J., He, L., Stenneth, L., Yu, P.S., Li, Z., Huang, Z., 2017. Computing urban traffic congestions by incorporating sparse GPS probe data and social media data. Acm Trans. Inf. Syst. Secur. 35, 4. https://doi.org/10.1145/3057281.

Xu, Z., Liu, Y., Yen, N., Mei, L., Luo, X., Wei, X., Hu, C., 2016. Crowdsourcing based description of urban emergency events using social media big data. IEEE Trans. Cloud Comput. 7161 https://doi.org/10.1109/tcc.2016.2517638 c, 1–1.

Yang, M.C., Rim, H.C., 2014. Identifying interesting Twitter contents using topical analysis. Expert Syst. Appl. 41 (9), 4330–4336. https://doi.org/10.1016/j.eswa.2013.12.051.

Yao, F., Wang, Y., 2020. Domain-specific sentiment analysis for tweets during hurricanes (DSSA-H): a domain-adversarial neural-network-based approach. Comput. Environ. Urban Syst. 83, 101522. https://doi.org/10.1016/j.compenvurbsys.2020.101522. February.

Yoo, S.Y., Song, J.I., Jeong, O.R., 2018. Social media contents based sentiment analysis and prediction system. Expert Syst. Appl. 105, 102–111. https://doi.org/10.1016/j.eswa.2018.03.055.

Young, T., Hazarika, D., Poria, S., Cambria, E., 2018. Recent trends in deep learning based natural language processing [Review Article]. IEEE Comput. Intell. Mag. 13 (3), 55–75. https://doi.org/10.1109/MCI.2018.2840738.

Zhang, Z., He, Q., Gao, J., Ni, M., 2018. A deep learning approach for detecting traffic accidents from social media data. Transp. Res. Part C Emerg. Technol. 86 (November 2017), 580–596. https://doi.org/10.1016/j.trc.2017.11.027.

Zhang, Y., Lu, Y., Zhang, D., Shang, L., Wang, D., 2019. RiskSens: a multi-view learning approach to identifying risky traffic locations in intelligent transportation systems using social and remote sensing. Proc. - 2018 IEEE Int. Conf. Big Data, Big Data 2018, 1544–1553. https://doi.org/10.1109/BigData.2018.8621996.

Zhang, W., Hong, Z., Chen, W., 2020. Hierarchical pricing mechanism with financial stability for decentralized crowdsourcing: a smart contract approach. IEEE Internet Things J. 1–16. https://doi.org/10.1109/jiot.2020.3007268. XX XX.

Zheng, Z., Wang, C., Wang, P., Xiong, Y., Zhang, F., Lv, Y., 2018. Framework for fusing traffic information from social and physical transportation data. PLoS One 13 (8), 1–19. https://doi.org/10.1371/journal.pone.0201531.

Zhou, Y., De, S., Moessner, K., 2016. Real world city event extraction from twitter data streams. Procedia Comput. Sci. 58, 443–448. https://doi.org/10.1016/j.procs.2016.09.069. DaMIS.

Zhou, J., Lu, Y., Dai, H.N., Wang, H., Xiao, H., 2019. Sentiment analysis of Chinese microblog based on stacked bidirectional LSTM. IEEE Access 7, 38856–38866. https://doi.org/10.1109/ACCESS.2019.2905048.