sciendo

*Transport and Telecommunication, 2020, volume 21, no. 4, 285–294*
*Transport and Telecommunication Institute, Lomonosova 1, Riga, LV-1019, Latvia*
*DOI 10.2478/ttj-2020-0023*

# EXPLORING AN ENSEMBLE OF TEXTUAL MACHINE LEARNING METHODOLOGIES FOR TRAFFIC EVENT DETECTION AND CLASSIFICATION

***Konstantinos Kokkinos [1], Eftihia Nathanail [2]***

*[1]Energy Systems Department, University of Thessaly, Larissa, Greece*
*[2]Civil Engineering Department, University of Thessaly, Volos, Greece*
*kokkinos@uth.gr, enath@uth.gr*

……

Late research has established the critical environmental, health and social impacts of traffic in highly populated urban regions. Apart from traffic monitoring, textual analysis of geo-located social media responses can provide an intelligent means in detecting and classifying traffic related events. This paper deals with the content analysis of Twitter textual data using an ensemble of supervised and unsupervised Machine Learning methods in order to cluster and properly classify traffic related events. Voluminous textual data was gathered using innovative Twitter APIs and managed by Big Data cloud methodologies via an Apache Spark system. Events were detected using a traffic related typology and the clustering K-Means model, where related event classification was achieved applying Support Vector Machines (SVM), Convolutional Neural Networks (CNN) and Long Short Term Memory (LSTM) networks. We provide experimental results for 2-class and 3-class classification examples indicating that the ensemble performs with accuracy and F-score reaching 98.5%.

**Keywords:** Textual, Traffic, Clustering, Classification, Ensemble, Deep-Learning.

## 1. Introduction

Intelligent traffic management has been realized as a "state-of-the-art" approach in modern and smart cities with great effect to local and regional economy, residents' health and environmental sustainability. Related to traffic, diverse information can be acquired using various heterogeneous and distributed sources over the area of interest. Most of these sources refer to personal smart equipment that either belongs to residents (drivers, passengers, cyclists, pedestrians etc.) or it is attached to and transmits from the vehicles. Most of the transmitted information is geo-located and timestamped, therefore it allows the development of novel spatiotemporal frameworks that can support decision making for traffic regulation and congestion avoidance (Gu *et al*., 2016).

The application of machine learning (ML) approaches to sensor data emanated from the Big Data repositories has been used to provide an end-to-end bi-directional characterization and classification of various traffic attributes such as travel speed and time, traffic flow and occupancy (Aqib *et al*. 2019). At the same time, people tend to post on social networks traffic related information especially in road hot spots and closures due to weather conditions, car accidents and road constructions (Alhumoud, 2019). Social network platforms such as Twitter and Facebook, through application programming interfaces (APIs), can allow timely dissemination of unstructured traffic-related information, if properly collected, processed and analysed (Essien *et al*., 2020). Owing to the enormity, veracity, variability and velocity of textual data, we are obliged to apply also Big Data methodologies and infrastructures in order to acquire and analyze them. Towards this direction, the work of D'Andrea *et al*. (2015) suggested a novel text mining model that applies SVM, *k*-NN and C4.5 decision tree models for classification on real-time tweets gathered in various areas of the Italian road network to indicating traffic events. At the same time, research works by Wongcharoen & Senivongse (2016) and Suma *et al*. (2017) have analysed heterogeneous textual data from Twitter to detect, classify and predict road traffic events for smart cities planning procedures.

However, the current state of the art research does not answer the following two challenges. Firstly, even though there exists adequate research in general DM modelling, we lack investigation and comparative analysis of ML algorithms for the detection of traffic related events (Alotaibi *et al*., 2019).

The second challenge relates to the interoperability of various techniques in creating a holistic approach when clustering and/or classifying textual data in general and especially for the Big Data case. As a conclusion, to the best of our knowledge, no research has been published that utilizes ensemble Data Mining (DM) modelling of Big Data textual data for detecting traffic related events.

Based on the above, within the context of this research, we attempted to identify traffic related events based on Twitter textual data, treated with innovative ML techniques. In particular, we included: (a) clustering of textual data to indicate areas of high usage, and (b) classification in textual traffic-related, geo-located and timestamped information.

The contribution of this work is two-fold: (1) it provides an exemplary methodology for clustering numeric geo-located traffic data from social media to dynamically calculate the regions of traffic hot spots in an urban area and (2) it provides a more integrated approach in classifying traffic events using an ensemble of binary and ternary classifiers to achieve greater accuracy in prediction. It should be noted that the proposed ensemble configuration combines the classification abilities of the participating classifiers even though this setup generates additional computational burden. For object instances that belong to the same category, the majority voting approach or the predicted probability average was used to find the best case scenario. However, to overcome the Big Data consequences due to the high rate of tweet generation (8,000 tweets/sec, hundreds of trillions per week according to 2018 evaluations) we used cloud infrastructures and services dedicated to Big Data handling and we adopted cluster computing frameworks such as Spark that utilizes MapReduce rather than the entire Hadoop ecosystem.

The structure of the manuscript is as follows: Section 2 is devoted to the state of the art description that relates to the clustering and event detection and classification of social media textual data specifically oriented to traffic management. After we set up the typology of the events in relation to the problem, in section 3 we also set the theoretical background of all the DM models used in our ensemble configuration. Section 4 illustrates the results of the implementation of the aforementioned ensemble along with a discussion on the validity of the experimental verification. Finally, section 5 contains the conclusions and future challenges in relation to the problem.

## 2. State of the Art

In the recent years, many researchers have proposed various prediction models based on textual data. In particular, though, when dealing with data from social network platforms such as Twitter, researchers also face several additional challenges due to voluminous and complex Big Data generated. Fortunately, sophisticated yet effective platforms and frameworks have been developed to deal with the complexity and the scalability of these problems (Tsai *et al*., 2015; Shim *et al*., 2015; Osman, 2019). Additionally, Twitter provides access to geo-tagged tweets via appropriate Restful service APIs (Goswami, & Kumar, 2019; Mazoyer *et al*., 2020). Apart from many static attributes that can be gathered from tweets, by using appropriate filtration mechanisms we can focus on specific locations and areas of interest and demonstrate user activities capturing their moving patterns (Xin, & MacEachren, 2020).

Clustering is the process of making groups of similar objects, throughout the whole data set, so that they are treated similarly within the group. Each cluster is assigned a unique label as an identifier. On the other hand, classification is a DM function that accurately predicts the group or cluster in which an unidentified object of the data set belongs to. For the first case, the density of positions of the GPS-transmitted tweets can be estimated using the popular K-Means algorithm, which computes the Euclidean distance as the cost function for the centroids of the clusters (Kokkinos *et al*., 2018). For the second case, there exists a repertoire of classifiers, mostly depending on the number of individual categories of transport events studied and the morphology of the acquired data. Such methodologies include Support Vector Machines (SVM), Multi-Layer Perceptron Neural Networks (MLP-NN) and the newest Deep Learning approach models such as Convolutional Neural Networks (CNN) and Long Short Term Memory networks (LSTM) (Tanuja *et al*., 2020).

At the same time, many flow prediction models have been introduced in the literature with most of them using regression methodologies such as k-NN (Chang *et al*., 2012) SVR (Lu *et al*., 2020) and MLP-ANN (Ding, 2019). However, there are a few disadvantages in using these methods, since the inclusion of non-numeric data obliges for model alteration. Furthermore, we also need to overcome the inherited linear architectures of the models, since in many cases, traffic flow and occurred traffic densities are predicted based on historical data and present correlation incidents affected by categorical valued multivariate subsystems. Also, gradient descent algorithms such as MLP-ANN can train ANNs and deep neural network architectures (either in unsupervised or supervised manner) are needed to overcome the poor prediction outcomes.

## 3. Methodology

### 3.1. Framework architecture and data pre-processing

To integrate related solutions and methodologies on several urban challenges in correspondence to traffic we propose a framework which is illustrated in Figure 1. The framework comprises three main components of processing namely: (1) Data collection and textual data pre-processing component, (2) Feature extraction and application of clustering and classification model component and (3) Evaluation component. Initially, the data is collected via the use of Social Feed Manager (2020). This is an online software that allows easy social media data harvesting and supports several exporting formats. We use an export wrapper to automatically filter and store only the geo-located information and text without revealing any personal authorship or other information details. At the same time, all stored textual data are fed to a set of pre-processing functions which are sequentially applied to the tweets of interest preparing them for feature extraction and the actual DM methodologies application. More specifically, the pre-processing is a

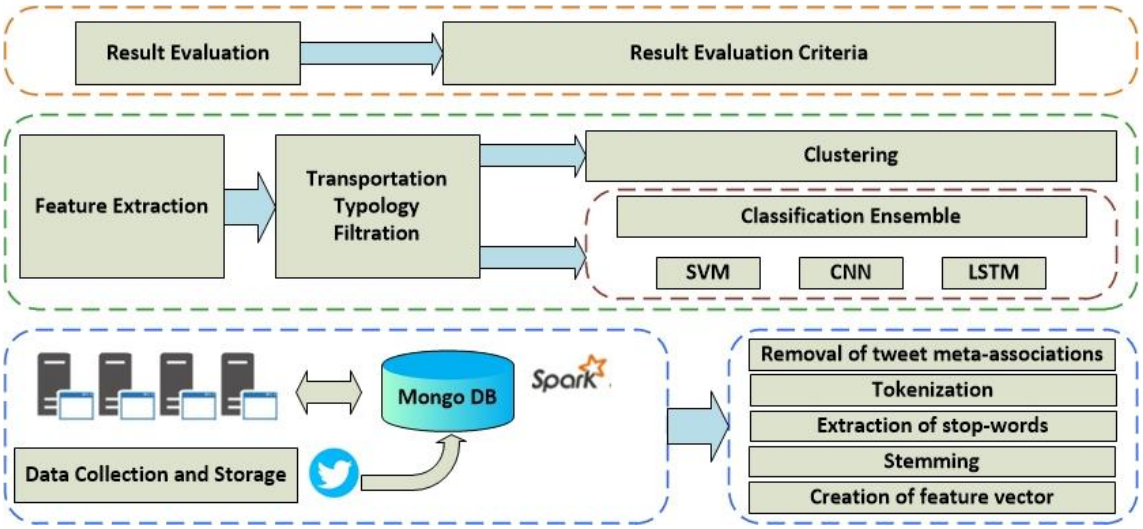



*Figure 1.* Logical architecture of the proposed framework

five-step operation consisting of: (a) removal of all the meta-associations of the textual data including the hashtags, redirections, links, mentions and finally converting all content to lower case characters, (b) tokenization where the corpus is enriched with syllables and words extracting all non-text characters, (c) extraction of non-significant words from the corpus such as articles and pronouns using the methodology by Zhou & Ca (2011), (d) removal of word suffices and grouping of words with similar meaning, a process well known as stemming (Porter, 1980), and finally, (e) the creation of the feature vector after the calculation of the Inverse Frequency Index (IDF) of the corpus content using the occurrence index of the stemmed tokens (Aiello *et al*., 2013).

### 3.2. Clustering of geo-located textual data using the unsupervised K-Means algorithm

Clustering is the method of partitioning all collection objects into a predefined set of clusters in such a way that objects of the same cluster share common attributes and are differentiated from objects of different clusters. For the case of textual data collections, correlation is the primary criterion for the clustering since data is not labelled and therefore, the clustering algorithm must reveal the underlying structure of the collection. Since we are referring only to geo-located data from tweets, the results of any clustering technique can be easily interpreted graphically.

For our case, we used the K-Means algorithm to identify "hot spots" of areas in the broader region of New York City, USA as a case study. More specifically, with the use of Social Feed Manager we gathered Twitter geo-located data from the virtual rectangle bounded by selected Lat-Log coordinates (left upper corner: Hawthorne NJ Lat: 40.939825, Log: -74.160612, right lower corner: Jones Beach State Park NY Lat: 40.597646, Log: -73.505552). The data was gathered to analyze this concept during holiday

period (Christmas period of 2017 – 2018 Dec. 11th to Jan. 3rd). All together, 173,236 tweets were examined, pre-processed and stored accordingly using the Spark MongoDB cloud services. The data set was divided into a training set (70% of the data base) and a testing set (the rest). As the application of the K-Means algorithm needs pre-definition of the number of clusters, after feeding the data into a single machine Spark ML we set K=7 as the predefined number of clusters and the method calculated the epicentres of intensified transportation activities. Figure 2 depicts the "hot spots" of the study area during the respective period. The aforementioned method is blind and unsupervised focusing only or computing the centroids of the clusters at hand and can be applied at any time retroactively in a region of interest. The same can be applied for the microscopic spatiotemporal case when not only the geo-location but also the time of tweet transmission is of interest.

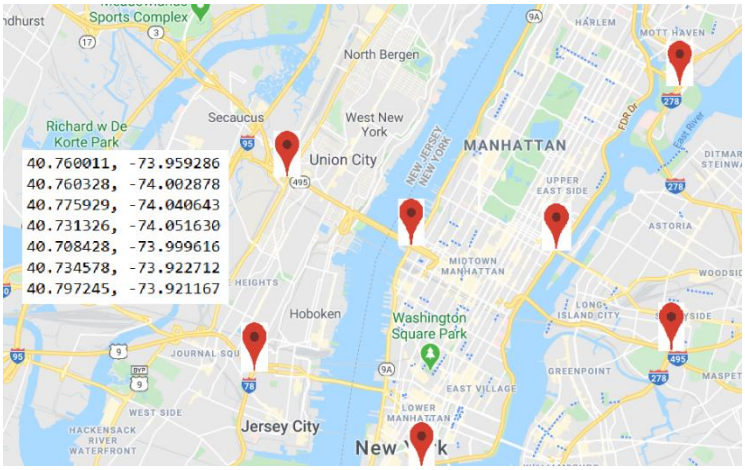



*Figure 2.* Hot spots in the study area

Additional filtering on the tweet content can be performed to detect the orientation of congestion probability increase. Searching in the collection including query keywords such as: traffic, accident, parked car etc. according to the discussed typology can fine tune the outcome.

### 3.3. Classification of textual data via SVM and sentiment analysis

The application of Support Vector Machines (SVM) and the proof of their performance in sentiment analysis is well described in (Abirami & Gayathri, 2017). The SVM models have shown to outperform traditional text categorization ML techniques such as Naïve Bayes networks. In opposition to the probabilistic Bayesian classifiers, the SVM classifier works with near optimum performance, especially in cases of binary classification. In this research, the idea was that the SVM succeeds to establish a hyperplane as the boundary between the two categories. In the developed hyperplane, we achieved not only the separation of tweets into two categories, but we also ensured that the categories separation was done with the biggest possible margin. Specifically, let $\vec{w}$ be the hyperplane vector above and $r_j$ the inclusion binary variable corresponding to positive or negative for the correct class selected for each tweet $\vec{t}_j$, i.e., $r_j \in \{1, -1\}$. Then the solution $\vec{w}$ resulting from the optimization problem can be presented as

$$\vec{w} = \sum_j m_j r_j \vec{t}_j, \quad m_j \geq 0, \tag{1}$$

where the $m_j$'s can be calculated from the dual optimization problem. Note that all the tweets $t_j$ for which the corresponding $m_j$ is positive, is said to be a support vector because these are the only ones that participate in the maximization of the solution $\vec{w}$ above. After the aforementioned optimization function is computed, then any new tweet $\vec{t}_j$ can be classified to one or the other side of the hyperplane (Khairnar & Kinikar, 2013). To achieve the generation of features, as mentioned in the data preprocessing phase, the corpus model was used via inverse document frequency (IDF). More specifically, after the tokenization and the stemming, all processed tweets are converted to numeric equivalent format where each of the words participating in the tweet is assigned a "weight of participation" in the corpus which is based on the term frequency- inverse document frequency (TF-IDF). The TF-IDF is a critical factor in natural language processing since it is depended on the frequency of each tokenized and stemmed word in the tweet and at the same time in the corpus of all tweets, as follows:

$$TF = T_j, \tag{2}$$

$$IDF = log\left[\frac{N}{1+\vec{t}_{j,}}\right], \tag{3}$$

$$TF - IDF = T_j \times log\left[\frac{N}{1+\vec{t}_{j,}}\right], \tag{4}$$

where $T_j$ is the total number of occurrences of the stemmed word under consideration in the tweet $\vec{t}_{j,}$ and $N$ is the total number of tweets in the collection/corpus which contains the corresponding word.

### 3.4. Classification of textual data via Convolutional Neural Networks

CNN is an extension of feed forward networks with differentiating only on the way neurons are connected between layers. Instead of one to one correspondence on the connectivity pattern, neurons in CNN abut to a smaller region of the adjacent layer (otherwise called the receptive layer). The neural network structure is designed in such a way that layers, namely the convolution layer and the pooling layer, are going in pairs, thus every convolutional layer is followed by a pooling layer. More specifically, in the convolutional layer the system convolves a filter on the outcome of the previous layer. The result of the filtering process is defined to be a feature collection. The synthesis of all feature collections, resulted by the application of several filter simultaneously in each layer, makes the content of the pooling layer. The learning process in CNN is local and considers the creation of the feature collections in each convolution layer. However, it is enough for the network to successfully classify words belonging to different classes according to a typology. A typical CNN is depicted in Figure 3.

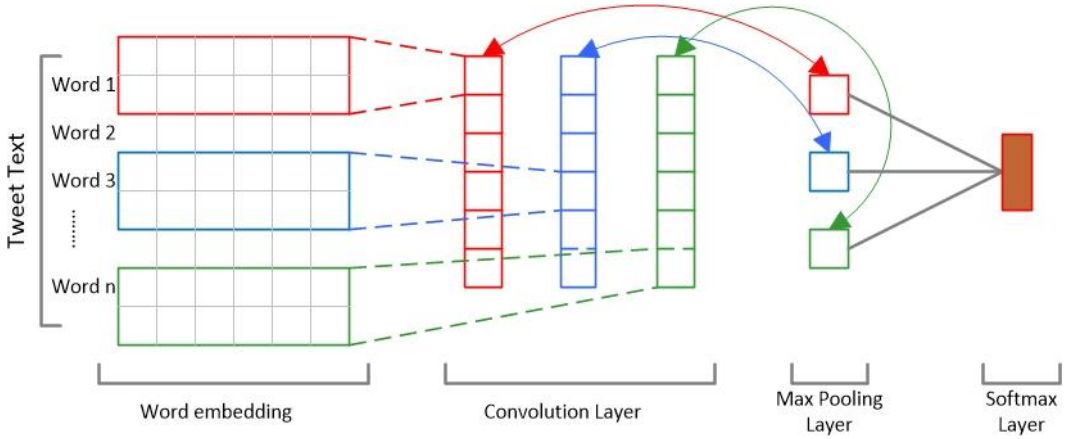



*Figure 3.* Convolutional neural network architecture for the tweet textual analysis (from Dabiri & Heaslip, 2019).

The process of using the CNN structure is initiated with the creation of word embedding. According to Dabiri & Heaslip (2019), after the tokenization and the stemming filtration, all the tweet words are converted to vectors of real numbers in an order that words with high similarity in meaning are adjacent to the vector space. It should be noted that this conversion to numeric data set decreases tremendously the data dimensionality. When we deal with unrelated words, the sparsity of the vector space does not allow the CNN to generalize and reduce the feature collections or maps. On the other hand, dense embedding highlights the computational power of the model. According to Kim (2014), if we define $\boldsymbol{n}$ to be the cardinality of the word set and $\boldsymbol{d}$ the width of the filter used for the production of the feature map, the application of such a filter $\boldsymbol{F}$ produces a feature map

$$\boldsymbol{c} = [c_1, \dots\dots, c_j, \dots\dots, c_{L-n+1}]^T \tag{5}$$

with $L$ being the maximum possible length of the tweet collection (corpus), regardless of the fact that the tweets have variable word count and every $\boldsymbol{c_j}$ is computed according to the dot-product:

$$\boldsymbol{c_j} = f(F.X_{j:j+n-1} + b) \tag{6}$$

where $\boldsymbol{b}$ is the bias introduced with $\boldsymbol{X}$ to be the matrix of features and $\boldsymbol{f}$ to be a non-linear activation function.

It should be noted that the pooling layer is called max since the basic idea here is to take out of the whole feature map only one specific feature to feed the Softmax layer (Kim, 2014). In each iteration, the Softmax layer receives the final vector of feature maps, i.e. it is a vector of vectors that defines the class in which the sample under evaluation lies. The preprocessing and the exclusion of all unnecessary tokens in the corpus not only improves the system's performance (better results in binary classification confusion matrices) but also the computational complexity of the algorithm, thus it decreases the computational time.

### 3.5. Classification of textual data via Long Short Term Memory Neural Networks

The ML methodologies shown above, and especially the K-Means clustering model and the SVM classification model, do not depict satisfactory performance in cases of high dimensional data with high and sparse feature vectors. Also, CNN are limited in extracting meaningless only keyword features, thus they miss to pinpoint the meaning of a traffic event within a tweet message. To cover the cases of high dimensionality data, we included in the proposed ensemble the use of bidirectional LSTM neural networks, which is a type of Recurrent Neural Networks (RNN), as a classic mode of deep learning mostly used for classification of voluminous data such as images. The typical LSTM unit is shown in Figure 4 where the core system comprises a cell $C_t$, a hidden state $h_t$, an input gate $i_t$, an output gate $O_t$ and a forget gate $f_t$. The network is able to store and remember values over variable time periods when at the same time the three aforementioned gates aim in regulating the incoming and outgoing information from the network.

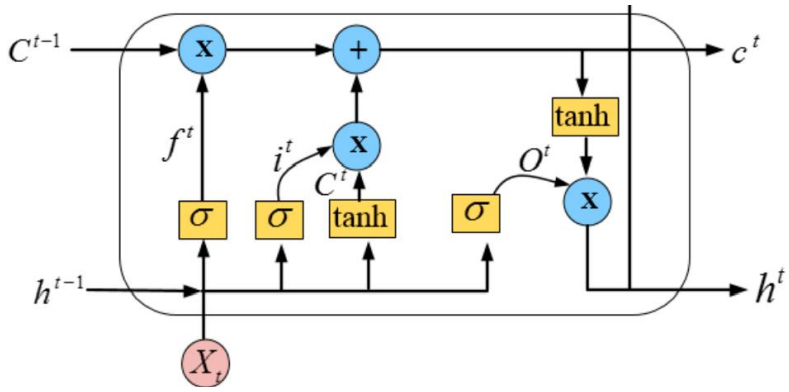



*Figure 4*. LSTM neural network architecture: the use of the input, output and forget gates (Ali *et al*., 2019)

As Karita *et al*. (2019) indicate, RNNs perform better even than CNNs when sentiment classification of high dimensionality is at hand. The LSTM neural network is an extension of RNN that computes the output as follows:

$$\vec{\boldsymbol{C}} = tanh(w_{xc}x^t + w_{hc}h^{t-1} + b_c)\,, \tag{7}$$

$$\boldsymbol{i^t} = \sigma(w_{xi}x^t + w_{hi}h^{t-1} + w_{Ci}C^{t-1} + b_i)\,, \tag{8}$$

$$\boldsymbol{f^t} = \sigma(w_{xf}x^t + w_{hf}h^{t-1} + w_{Cf}C^{t-1} + b_f)\,, \tag{9}$$

$$\boldsymbol{C^t} = f^t \odot C^{t-1} + i^t \odot \vec{C}\,, \tag{10}$$

$$\boldsymbol{O^t} = \sigma(w_{xo}x^t + w_{ho}h^{t-1} + w_{Co}C^{t-1} + b_o)\,, \tag{11}$$

$$\boldsymbol{h^t} = O^t \odot \tanh\,(C^t)\,, \tag{12}$$

where $w_{xc}, w_{hc}$ correspond to the weights of the previous cell and hidden state, $w_{xi}, w_{hi}$ are the weights of the cell and hidden state input, $w_{xf}, w_{hf}$ are the weights of the cell and the hidden forget state, $w_{xo}, w_{ho}$ are the corresponding weights for the output, $b_c, b_i, b_f$ are the biases for the cell, input and forget gates, $i^t$ is the new input gate, $f^t$ is the new forget gate, $O^t$ is the new output gate and $h^t$ is the new hidden state. Finally, the Softmax activation function of the network is calculated in Equation 13, where *c* represents the feature category and *k* represents the number of steps/iterations:

$$Softmax^{kt} = \frac{\exp(x^{kt})}{\sum_k^c \exp(x^{kt})}. \quad (13)$$

The model presented above takes as input the outcome of the word embedding process converting the stemmed words into a matrix of vectors where each one of the vectors corresponds to the words participating in the tweet after the cleaning process. In the general case of voluminous data, the sparsity of the matrix is decreased however the size depends on the size of the corpus which is very big. The process of classification starts with feeding the aforementioned vectors into the neural network and the outcome is directed to the Softmax layer (a layer of neurons activated by the sigmoidal function). The network's performance heavily depends on the training, showing at times reliance only on specific words, so for this reason the introduced biases trigger to create better language regularization.

## 4. Results and Discussion

In the case of classification and for the evaluation of all three models, we used the same data set as in clustering, with 173.236 tweets which were cleaned and preprocessed to check the containment of traffic related keywords. In this case, the partition of training-testing sets was 80%-20%, respectively. The reason is that for the case of classification we do not simply use numeric geo-located data, but we also create a big corpus of word embedding vectors for the Bi-LSTM and the other models.

We performed two classification experiments and compared the performance of the three classifiers. For the first experiment, we performed a binary classification of traffic conditions. The two classes assumed weather caused or not weather caused traffic conditions, which were searched in the data set through the words {traffic, rain, snow, sleet, accident, slowdown, congestion, stuck, thunder, crash} (Kokkinos *et al*., 2018). For the second experiment, we performed a ternary classification experiment on the same pre-processed set of tweets. Here, the experiment pursued classification of tweets related to three specific cases which are expected to affect traffic, and in particular accidents, seasonal events (for example, Christmas) and other external events (for example, a music concert). The queries assumed the same search words as in the first experiment, further enriched with the words {game, strike, demonstration, flight, Christmas, year, accident, crash, ambulance, shopping} (Kokkinos *et al*., 2018). To avoid the variability of tweet length and the corresponding variable on the feature and word embedding vectors, we made a threshold on the number of most tuples used to create the keywords of each tweet.

The Hyper-Parameters examined in the study with both experiments are depicted in Table 1.

**Table 1.** Hyper-Parameter attributes and values

| Neural Network Type | Hyper-Parameter Attributes | Attribute Values |
|---|---|---|
| CNN | Kernel size | 3 |
| | Pool size | 2 |
| | Number of filters | 256 |
| | Hidden size | 128 |
| LSTM | Hidden size | 50 |
| Both Networks | Batch size | 32 |
| | Epoch | 25 |

Additional parameters were: (a) the Learning Rate of the system, (b) the Dropout Rate and (c) the number of recurrent units participating. We did not performe any additional validation performance tests due to inability to synchronize Spark cloud services with the extensive or even exhaustive search of all the parameter space.

For the case of the binary classification experiment, the case study allowed us to calculate the performance of the classifier using counters such as: True Positive (TP), True Negative (TN), False Positive (FP) and False Negative (FN). By using these counters, we could compute the following performance metrics: Accuracy (Acc) (Equation 14), Precision (Prec) (Equation 15), Recall (Rec) (Equation 16) and F-Score (Equation 17). The above metrics were also used for the 3-class classification.

$$Acc = (TP + TN)/(TP + FP + FN + TN), \quad (14)$$

$$Prec = TP/(TP + FP), \quad (15)$$

$$Rec = TP/(TP + FN), \quad (16)$$

$$F - Score = (1 + \beta^2)\frac{Prec.\ Rec}{(\beta^2.Prec) + Rec}, \quad (17)$$

where $\beta = 1$ for class-balanced datasets.

Table 2 depicts our findings for the binary classifier, made by all three classifiers of the ensemble. It should be noted that we calculated the precision, the recall and the F-score for the three models separately for each class, so that the confusion matrix could also be derived automatically. However, for the case of accuracy, we computed one value considering all cases, regardless of the cause (weather or not-weather). Table 2 shows that SVM performed better than the other models, as it was expected due to the well-known general success of SVM as a binary classifier.

**Table 2.** Classification performance for the case of binary classifiers

| Classifier | Accuracy (%) | Precision (%) by class | | Recall (%) by class | | F-Score (%) by class | |
|---|---|---|---|---|---|---|---|
| | | Weather caused | Not-Weather caused | Weather caused | Not-Weather caused | Weather caused | Not-Weather caused |
| **SVM** | **92.73** | 92.06 | 93.4 | 92.80 | 92.66 | 93.02 | 92.44 |
| **CNN** | **90.11** | 89.45 | 90.22 | 90.38 | 91.12 | 88.67 | 90.82 |
| **LSTM** | **91.27** | 90.47 | 90.56 | 90.93 | 92.01 | 91.26 | 92.40 |

Tables 3, 4 and 5 present the performance of the three classifiers of the ensemble separately for each class in the three-class classification. According to the results, LSTM Recurrent Neural Nets outperformed the others, in the vast majority of indicators, showcasing that the Deep Learning approach is more dependable than the others in the ensemble.

**Table 3.** Three-class classification for the case of SVM

| | Due to accidents | Seasonality affected | External events |
|---|---|---|---|
| Accuracy (%) | **81.31** | | |
| Precision (%) by class | 77.65 | 79.60 | 83.61 |
| Recall (%) by class | 82.44 | 81.68 | 80.56 |
| F-Score (%) by class | 81.79 | 81.17 | 81.38 |

**Table 4.** Three-class classification for the case of CNN

| | Due to accidents | Seasonality affected | External events |
|---|---|---|---|
| Accuracy (%) | **81.76** | | |
| Precision (%) by class | 79.65 | 80.92 | 84.72 |
| Recall (%) by class | 82.34 | 82.18 | 80.49 |
| F-Score (%) by class | 82.21 | 81.28 | 81.79 |

**Table 5.** Three-class classification for the case of LSTM

| | Due to accidents | Seasonality affected | External events |
|---|---|---|---|
| Accuracy (%) | **82.69** | | |
| Precision (%) by class | 81.32 | 81.45 | 83.98 |
| Recall (%) by class | 85.75 | 81.87 | 81.32 |
| F-Score (%) by class | 84.73 | 81.03 | 82.08 |

However, the use of the ensemble is not only justified by the above results, but it is also recommended in the cases that one model does not perform with accuracy like the others.

## 5. Conclusions

In this research work, we proposed an ensemble framework of three classifiers and one clustering model to detect traffic events to work as a decision support system for detecting traffic-related events via the use of traditional Data Mining and the modern Deep Learning methodologies. The large amount of tweets generated in the study area required the use of Big Data oriented mechanisms and cloud services to store, filter, pre-process and apply the aforementioned information retrieval models. Firstly, we gathered raw tweet geo-located data GPS-referenced on the broader area of New York City, using the Social Feed Manager online software which is a user friendly online web application of the Twitter API. Then we applied a set of pre-processing techniques in order to remove tweet meta-associations, tokenize the raw data, extract stop and unneeded words from the tweets, transform the remaining words into their stem version and formulate the whole sample ready for feature vectors to be created. The application of clustering via K-Means algorithm resulted in a set of "crisp neighbourhoods" with clear cut borders as expected, since the K-Means model has been proven to be of excellent performance in the case of numeric oriented datasets such as GPS locations.

For the classification, the ensemble of three classifiers (SVM, CNN and LSTM) was applied either as a binary or a ternary classifier framework. In the subcase of the binary classification, the SVM performed the best, with accuracies reaching 92.73%, followed by LSTM and the CNN to be the last in performance. However, in the subcase of the ternary classification, LSTM proved to be the most effective classifier due to its iterative processing and the amount of hidden layers used along with the bidirectional processing. Further, the use of word embedding in the creation of the feature vectors is the main factor that supports the superiority of the LSTM model performance compared to the other two models in this domain.

It is worth investigating the balancing of tweets in each class, which is also a very critical factor, since the performance metrics directly depend on it. It is a challenge to study the effect in the training of the models for the cases of under-represented classes having a strong confidence that this will improve the performance of all the participating models in the ensemble by reducing overfitting. Although solely traffic-related Twitter data is not proven to support decision making, the present research indicated that the proposed methods are very promising in detecting such traffic-related events, based on such data.

## Acknowledgements

This paper is a revised and expanded version of the paper "Applying Unsupervised and Supervised Machine Learning Methodologies in Social Media Textual Traffic Data" presented at the 4th Conference on Sustainable Urban Mobility, Skiathos Island, Greece, May 24-25, 2018.

## References

1. Abirami, A. and Gayathri, V. (2017) A survey on sentiment analysis methods and approach, in *Advanced Computing (ICoAC), 2016 Eighth International Conference on, 2017*: IEEE, pp. 72-76.
2. Aiello, L-C., Petkos, G., Martin, C., Corney, D., Papadopoulos, S., Skraba, R., Göker, A. (2013) Sensing trending topics in Twitter, *IEEE Transactions in Multimedia*, 15(6), pp. 1268–1282.
3. Alhumoud, S. (2019) Twitter Analysis for Intelligent Transportation. *The Computer Journal* 62, 1547–1556. https://doi.org/10.1093/comjnl/bxy129
4. Ali, F., El-Sappagh, S., Kwak, D. (2019) Fuzzy ontology and LSTM-based text mining: A transportation network monitoring system for assisting travel. *Sensors*, 19(2), 234.
5. Alotaibi, S.; Mehmood, R.; Katib, I. (2019) Sentiment Analysis of Arabic Tweets in Smart Cities: A Review of Saudi Dialect. In *Proceedings of the 2019 Fourth International Conference on Fog and Mobile Edge Computing (FMEC)*, Rome, Italy, June 10–13, pp. 330–335.
6. Aqib, M., Mehmood, R., Alzahrani, A., Katib, I., Albeshri, A., Altowaijri, S.M. (2019) Smarter Traffic Prediction Using Big Data, In-Memory Computing, Deep Learning and GPUs. *Sensors* 19, 2206. https://doi.org/10.3390/s19092206
7. Chang, H., Lee, Y., Yoon, B., Baek, S. (2012) Dynamic near-term traffic flow prediction: system-oriented approach based on past experiences. *IET Intel. Transport Systems* 6, 292–305. doi:10.1049/iet-its.2011.0123.
8. D'Andrea, E., Ducange, P., Lazzerini, B., & Marcelloni, F. (2015) Real-time detection of traffic from twitter stream analysis. *IEEE transactions on intelligent transportation systems*, 16(4), 2269-2283.
9. Dabiri, S., Heaslip, K. (2019) Developing a Twitter-based traffic event detection model using deep learning architectures. *Expert systems with applications*, Vol. 118, pp. 425-439.
10. Ding, J. (2019) Investigation on the Traffic Flow Based on Wireless Sensor Network Technologies Combined with FA-BPNN Models, *Journal of Internet Technology*, Vol. 20, No. 2, pp. 589-597.
11. Essien, A., Petrounias, I., Sampaio, P., & Sampaio, S. (2020) A deep-learning model for urban traffic flow prediction with traffic events mined from twitter. *World Wide Web*, 1-24. https://doi.org/10.1007/s11280-020-00800-3
12. Goswami, A., & Kumar, A. (2019) Event Detection Using Twitter Platform. In *Digital Business,* pp. 429-480. Springer, Cham.
13. Gu, Y., Qian, Z. S., & Chen, F. (2016) From Twitter to detector: Real-time traffic incident detection using social media data. *Transportation research part C: emerging technologies*, 67, pp. 321-342. https://doi.org/10.1016/j.trc.2016.02.011
14. Karita, S., Chen, N., Hayashi, T., Hori, T., Inaguma, H., Jiang, Z., Someki, M., Soplin, N.E.Y., Yamamoto, R., Wang, X. and Watanabe, S. (2019) A comparative study on transformer vs RNN in speech applications. In: *2019 IEEE Automatic Speech Recognition and Understanding Workshop (ASRU),* pp. 449-456.

15. Khairnar, J., and Kinikar, M. (2013) Machine learning algorithms for opinion mining and sentiment classification, *International Journal of Scientific and Research Publications*, 3(6), pp. 1-6.
16. Kim, Y., (2014) Convolutional neural networks for sentence classification. *arXiv preprint arXiv*:1408.5882.
17. Kokkinos, K., Nathanail, E., & Papageorgiou, E. (2018) Applying Unsupervised and Supervised Machine Learning Methodologies in Social Media Textual Traffic Data. In: *The 4th Conference on Sustainable Urban Mobility* (pp. 665-672). Springer, Cham.
18. Lu, Z., Xia, J., Wang, M., Nie, Q., & Ou, J. (2020) Short-term traffic flow forecasting via multi-regime modeling and ensemble learning. Applied Sciences, 10(1), 356.
19. Mazoyer, B., Cagé, J., Hervé, N., & Hudelot, C. (2020) A french corpus for event detection on twitter. In *Proceedings of the 12th Language Resources and Evaluation Conference* pp. 6220-6227.
20. Mostafaeipour, A., Rafsanjani, A. J., Ahmadi, M., & Dhanraj, J. A. (2020) Investigating the performance of Hadoop and Spark platforms on machine learning algorithms. *Journal of SuperComputing*.
21. Osman, A. M. S. (2019) A novel big data analytics framework for smart cities. *Future Generation Computer Systems*, 91, 620-633.
22. Porter, M. F. (1980) An algorithm for suffix stripping, Program: Electron. Library *Inf. Syst.*, 14(3), pp 130–137.
23. Shim, J. P., French, A. M., Guo, C., & Jablonski, J. (2015) Big data and analytics: Issues, solutions, and ROI. *Communications of the Association for Information Systems*, 37(1), 39.
24. Social Feed Manager, (2020) *A social network data acquisition software*, https://gwu-libraries.github.io/sfm-ui/ (last accessed, July 20th, 2020).
25. Suma, S., Mehmood, R., Albugami, N., Katib, I., & Albeshri, A. (2017) Enabling next generation logistics and planning for smarter societies. *Procedia Computer Science*, 109, 1122-1127.
26. Tanuja, U., Gururaj, H. L., & Janhavi, V. (2020) A Machine Learning Algorithm for Classification, Analyzation and Prediction of Multimedia Messages in Social Networks. In *Proceedings of First International Conference on Computing, Communications, and Cyber-Security* (IC4S 2019) (pp. 485-499). Springer, Singapore.
27. Tsai, C. W., Lai, C. F., Chao, H. C., & Vasilakos, A. V. (2015) Big data analytics: a survey. *Journal of Big data*, 2(1), pp. 1-32.
28. Wongcharoen, S., & Senivongse, T. (2016) Twitter analysis of road traffic congestion severity estimation. In *13th International Joint Conference on Computer Science and Software Engineering (JCSSE)* (pp. 1-6). IEEE.
29. Xin, Y., & MacEachren, A. M. (2020) Characterizing traveling fans: a workflow for event-oriented travel pattern analysis using Twitter data. *International Journal of Geographical Information Science*, 1-20.
30. Zhou, Y., and Cao, Z.-W. (2011) Research on the construction and filter method of stop-word list in text preprocessing. In: *Proc. 4th ICICTA, Shenzhen, China,* vol. 1, pp. 217–221.