Research Article GU J Sci 34(3): 718-731 (2021) DOI: 10.35378/gujs.715296

Gazi University

Journal of Science

http://dergipark.gov.tr/gujs

Efficient Turkish Text Classification Approach for Crisis Management Systems

Saed ALQARALEH *

Hasan Kalyoncu University, Computer Engineering Department, Gaziantep, TURKEY

Highlights

- Design of text classification system that fully supports the Turkish language.
- Building an efficient system that can identify social media data related to the crisis.
- Effect of social media analytics on internet-based crisis management systems.

Article Info

Received: 06 Apr 2020
Accepted: 03 Jan 2021

Keywords

Crises management Systems
Ensemble learning
Text classification

Abstract

In this paper, an effective tweet classification system that fully supports the Turkish language has been developed. The proposed system can be used for mining (classifying) the recently published and publicly available tweets to find the crisis's most related and useful tweets to gain situational awareness, which can help in taking the correct responses in order to prevent or at least decrease the effect of such situations. A deep study was carried out to improve and optimize the proposed system. In more detail, some intensive experiments were performed to investigate the performance of some well-known machine learning algorithms, i.e., K-Nearest Neighbor (KNN), Support Vector Machine (SVM), and Naive Bayes (NB) when used for text (tweets) classification. Then, the performances of the ensemble systems of the studied algorithms and the Random Forest (RF), AdaBoost Classifier (AdaBoost), GradientBoosting Classifier (GBC) ensemble systems have also been observed. As shown in the experimental evaluation and analysis, the proposed approach has stability, robustness, and can achieve quite good performance when processing the Turkish language. The performance of the proposed classifier was also compared with two state-of-the-art text classification approaches, i.e., "Empirical" and “Turkish Deep ".

1. INTRODUCTION

Crises or disasters are situations that people may encounter throughout their lives. Such situations often have many negative effects such as loss of life and property. The importance of crisis management has increased considerably, as disasters such as terrorism, flood, earthquake, fire, traffic, work accidents, etc. negatively affect either directly or indirectly all or a large part of society.

Nowadays, social media reach the point of being involved almost everywhere in our daily lives. Sharing information became more frequent and faster using platforms such as Facebook, Instagram, and Twitter. This allows social media's data repository to be essential and popular in terms of information mining. For instance, it is possible to use social media data to immediately detect a situation that requires an action from a recovery team, like during natural or human-made disasters/crises. Social media can help authority and volunteers to evaluate and make the most efficient discussions. However, based on the very huge amount of data that are usually shared during such a situation, it is essential to have an efficient and accurate computer-based system that can analyze and classify the available data.

Disaster Management System (DMS) aims to observe, detect, and respond to crises as early as possible in addition to providing a safer and more comfortable environment for people affected by the crisis.
Our aim in this study is to build a system that can monitor the published tweets and identify the situations of crises that require help. This was done by building an efficient text classification system that fully

*e-mail: saed.alqaraleh@hku.edu.tr

supports the Turkish language. In general, machine learning (ML) algorithms such as K-Nearest Neighbor (KNN), Support Vector Machine (SVM), and Naive Bayes (NB), etc. and the ensemble learning ones such as Random Forest, AdaBoost and GradientBoosting are very frequently used for text classification. For instance, for building crisis management systems, these algorithms can be used as the system classifier(classify the current input text as a disaster or not disaster). In more detail, the input text is preprocessed first, then its features are extracted(text is converted into numerical values), and finally the classifier is making a decision on the class of the input.

In this paper, we have proposed a new Turkish language text classification approach model that can efficiently classify, identify Social media data related to the crisis, and inform the authority ASAP (almost a real-time response). This leads to gain situational awareness that may help in preventing or decrease the effect of some disaster by taking the correct responses.

Up to our knowledge, supporting the Turkish language for this topic is new. In addition, the performance of some machine learning (ML) algorithms such as K-Nearest Neighbor (KNN), Support Vector Machine (SVM), and Naive Bayes (NB), etc. and the ensemble learning ones such as Random Forest, AdaBoost and GradientBoosting were investigated in order to find the most suitable classifier for the Turkish language. It is worth mentioning that the performance of ensemble learning will outperform the other ML algorithms. However, we have gone one further step in this study by investigating the effect of changing the base classifier of the mentioned ensemble learning algorithms. On the other hand, although it is obvious that Convolutional Neural Networks can improve multiple operations in language processing, such as autocorrecting, autocompleting, data mining (ex. translation), classification and many other tasks, using such technique requires the availability of a large number of training data which is not available yet, i.e, our constructed datasets is the first one related to disasters for the Turkish language. Overall, unlike the approach of [12], which uses Convolutional Neural Networks, our approach does not require a huge amount of training data to achieve an acceptable performance.

2. CMS RECENT DEVELOPMENTS AND STUDIES

In this section, the recent studies and developments related to the proposed approach have been summarized. In [1], an approach that works on detecting the online available disaster's relevant news was proposed. Mainly, this approach supports the English language only and uses natural language processing in order to preprocess the collected news articles and extract their features. In addition, it uses machine learning algorithms such as SVM and NB to classify the scraped news into disaster relevant or irrelevant news. Overall, both NB and SVM was able to achieve comparable performance.

In [2], a system for binary classification of Twitter data was developed. This system which supports the English language only consisted of two stages where the first one uses the Random Forest, SVM, NB, and Decision Trees, while in the second stage, some ensemble learning approaches are used. In addition, the effect of TFIDF, psychometric, and linguistic were investigated. Overall results indicated that ensemble learning can achieve a significant and acceptable performance for building a real-time CMS.

In [3], a comparison study that investigated the performance of machine learning algorithms such as SVM, Random Forest, Gradient Boosting, etc., and various CNN based models for the proposed of classifying the available disaster's text related to some events such as hurricane, earthquake, was performed. Their result showed that Gradient Boosting outperformed the other studied machine learning algorithms. In addition, related to the CNN based models, the authors suggested that selecting CNN's embedding approach is critical and has a significant effect on the classification task. Overall, the studied CNN models were able to outperform all the other classifiers.

In addition, recently some multimodal classification approaches that work on detecting disasters using both text and images were introduced. For instance, a deep learning model that combines the features of both text and images in order to classify the social media posts was presented in [4]. In more detail, after extracting the features of the post's text and image, this approach works on reducing the dimensionality of the extracted feature vectors. Next, the two vectors are concatenated into one vector. Finally, the resulted vector is passed into the system classifier to be classified as relevant or irrelevant.

Unfortunately, up to the time of writing this study, we could not find any study related to crisis management systems related to the Turkish language. However, this study can be considered as an implementation of a Turkish text classification system for crisis response. In the following, we have summarized some recent studies about Turkish text classification.

In [5], a sentiment analysis system for Turkish tweets using Machine Learning and Word Embedding was introduced. This system starts by converting words into a vector using Word2Vec [6, 7]. Then, each tweet is represented by two vectors, i.e., the first is created using the product-based representation model, and the second is produced using the total based representation model. The performances of the two vectors were compared using Random Forest and Support Vector Machine classifiers. According to [5], the performance of studied classifiers varies according to the used data set.

In [8] and [9], the performance of state-of-the-art word embedding approaches for Turkish text classification was investigated. In more detail, in [8], the efficiency of using the bag of words (Bow) and Word2Vec was observed. In addition, this study used the SVM as the system classifier. Overall, this study concludes that Word2Vec is more efficient for classifying the Turkish text. On the other hand, in [9], an unlabeled large corpus of Turkish words was used to train Word2Vec. Then, the resulted word vectors were used to tune and prepare multiple deep neural network models that have been tested using another corpus of 1.5 million samples. It has been found, as a result of this study, that we can gain a 5% to 7% improvement by using a pre-trained Word2Vec for the Turkish text classification.

In [10], a new Turkish dataset related to newspapers and named as TTC-3600 was introduced. This dataset was used to investigate the performance of some ensemble models such as the Rotation Forest. As a result, they found that the performance of the KNN algorithm can be improved using the Rotation Forest algorithm. On the other hand, ensemble models didn't have a significant impact on KNN. Related to the SVM, both Boosting and Bagging have decreased its accuracy.

In [11], the classification of Turkish 1150 documents belonged to 5 different categories using the Naïve Bayes algorithm was introduced. This approach uses the n-gram to create multiple models that use 2-gram, 3-gram, and 4-gram respectively. As a result, using 3-gram achieved the best performance and execution time.

In [12], a CNN model was built for processing the Turkish sentiment analysis task. This model, which we refer to it as Turkish Deep, contains 6 dense layers, where each one has different output dimensions and a 0.5 Dropout is allocated after the second and the fourth layer, while 0.2 Dropout is allocated after the third one. The performance of multiple preprocessing steps and feature represents were investigated also in [12]. Overall results showed that the deep learning method has the potential to build a better solution for sentiment analysis.

3. THE PROPOSED SYSTEM

The main purpose of this study is to investigate the possibility of building an efficient text classifier that can support the Turkish language and has the ability to identify and classify in real-time the available tweets and inform the authority ASAP about the data related to the possible crisis. By doing so, we believe that authority can gain situational awareness and make some decisions in order to prevent or decrease the effect of possible disaster. Figure 1 shows the schematic diagram of the proposed disaster management system. In general, the proposed system consisted of the following stages:

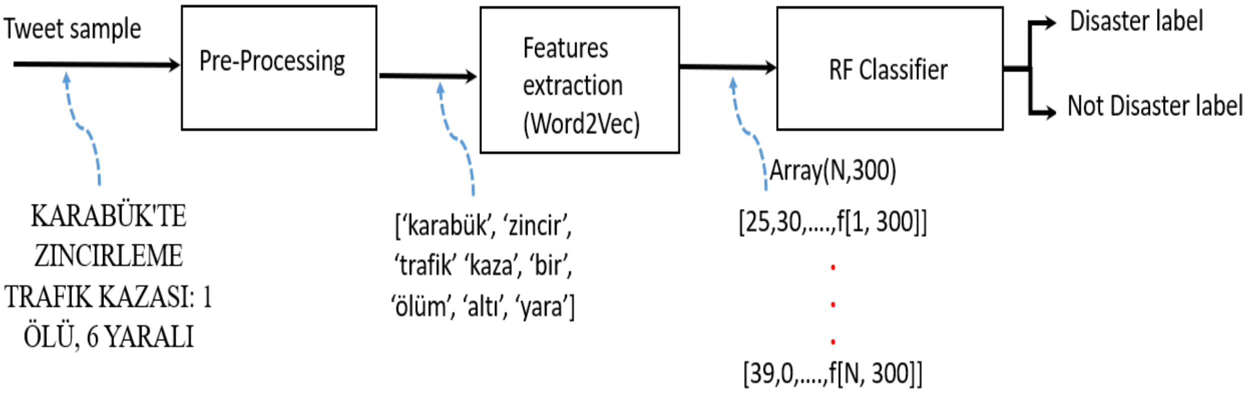

Tweet sample
Pre-Processing
Features extraction (Word2Vec)
RF Classifier
Disaster label
Not Disaster label
KARABÜK'TE ZINCIRLEME TRAFIK KAZASI: 1 ÖLÜ, 6 YARALI
['karabük', 'zincir', 'trafik' 'kaza', 'bir', 'ölüm', 'altı', 'yara']
Array(N,300)
[25,30,....,f[1, 300]]
.
.
.
[39,0,....,f[N, 300]]

Figure 1. *A schematic diagram of the proposed disaster management system*

3.1. Data Aggregation

To overcome the problem of not having any available Turkish dataset about disasters, which can be used for building such a system, we have collected 20,000 tweets, where around 9500 is truly relevant to some disasters that occurred in Turkey, while the remaining tweets are irrelevant, however, it was published during the disasters (most of these were jokes). Figure 2 shows the main steps of constructing this dataset. In more detail, Twitter Streaming API was used to collect the data after selecting a set of Turkish keywords related to disasters, such as "#deprem" which means earthquake, "#Yangın", which means fire, "#trafik kazası" which means traffic accident, "#İş Kazası" which means work accident, "#Sel" which means "Flood", etc. The second step was to annotate these collected tweets, and this process was done by three annotators. Where each one vote whither each tweet is relevant or not, and then majority voting was applied for the final decision. It is worth mentioning that during the annotation process all noisy tweets such as the misleading hashtags and duplicated ones were deleted.

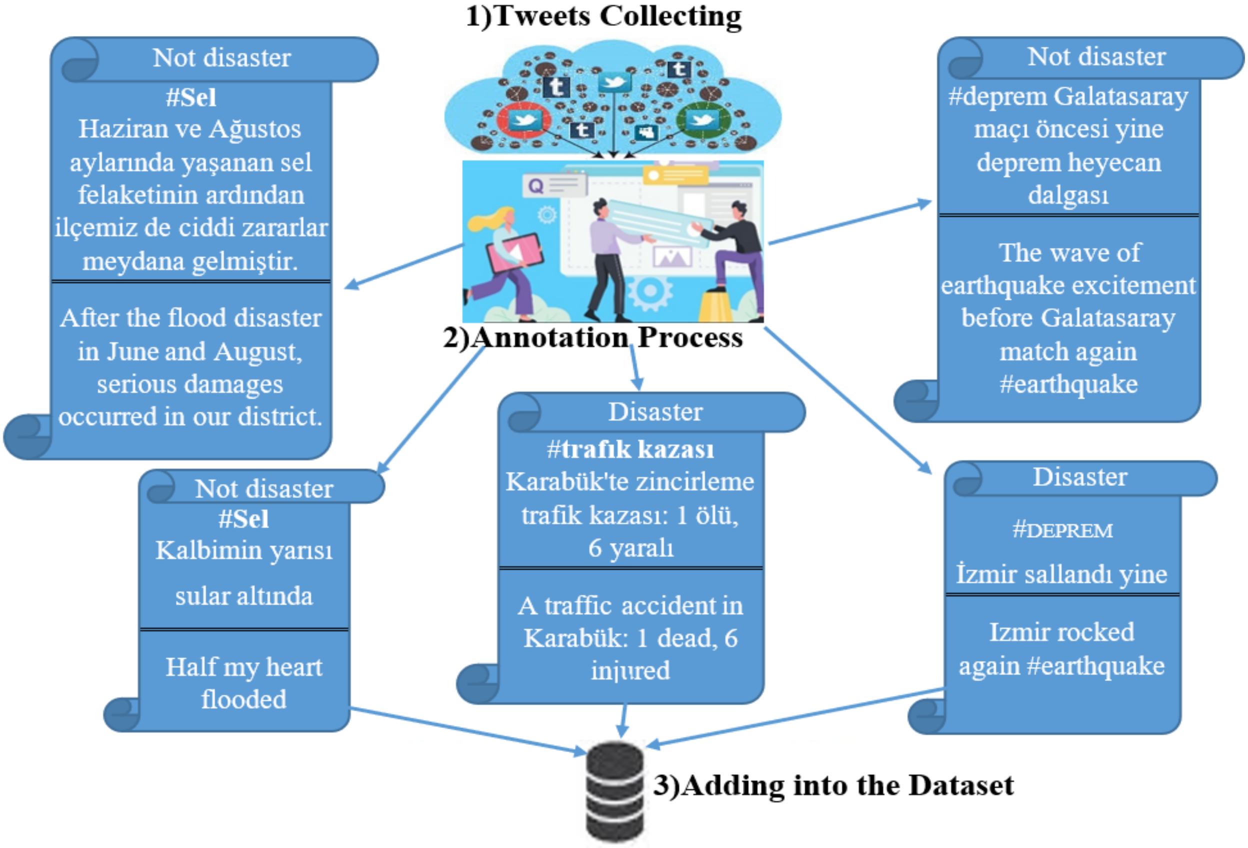

1)Tweets Collecting
Not disaster
#Sel
Haziran ve Ağustos aylarında yaşanan sel felaketinin ardından ilçemiz de ciddi zararlar meydana gelmiştir.
After the flood disaster in June and August, serious damages occurred in our district.
Not disaster
#deprem Galatasaray maçı öncesi yine deprem heyecan dalgası
The wave of earthquake excitement before Galatasaray match again #earthquake
2)Annotation Process
Not disaster
#Sel
Kalbimin yarısı sular altında
Half my heart flooded
Disaster
#trafik kazası
Karabük'te zincirleme trafik kazası: 1 ölü, 6 yaralı
A traffic accident in Karabük: 1 dead, 6 injured
Disaster
#DEPREM
İzmir sallandı yine
Izmir rocked again #earthquake
3)Adding into the Dataset

Figure 2*. Steps of Constructing the Dataset*

3.2. Data Pre-processing

Unlike other languages such as English, the Turkish language requires some extra and spatial pre-processing steps as it is agglutinative and has a different structure. In this work, the following steps, which are based on our preliminary experimental investigations suitable for Turkish are performed.

A) Tokenization: Splitting text strings into tokens or smaller pieces such as sentences, words, or characters. In this study, each tweet is tokenized into words.

B) Eliminate and delete the useless information: In general, tweets may contain some text that may mislead the classification process or it does not give a specific meaning. To avoid such a situation, the following sub-steps are applied:

i. Eliminating stop words and punctuation, which is done using the Turkish corpus of Nltk [13].

ii. Eliminating Hashtags, URL, and reserved words: The hashtag is a '#' (ex. #earthquake) that is often used to publish and notify users about newly published tweet(s). In addition, using the hashtag increase the chance of creating a trending topic. However, users may publish tweets using irrelevant hashtags. In addition, the available URLs do not help or provide us with useful information for the classification process. Furthermore, Twitter reserved words such as “FAV, RT, VIA” are removed also. It is worth mentioning that some other information such as extra space, single-character word, and emoji are also eliminated.

C) Deasciification:’ö’, ’ü’, ‘ğ’, ‘ç’, ’ş’, ’ı’ are Turkish language-specific characters that do not exist in other languages such as English. However, for simplicity users substitute these letters by the equivalent ASCII ones. Hence, deasciification is the process of converting the data(characters) into its correct form.

D) Text Normalization: It includes some sub-steps such as case folding(converting all text to the same letter size either upper or lower, also it includes converting the numbers to word equivalents. In addition, handling the misspellings is applied and this sub-step can be considered as an essential one.

3.3. Feature Extraction

Nowadays, word Embedding can be considered among the most popular approaches that can be efficiently used to map the text into real-valued vectors while preserving the contextual similarity [7]. The FastText [14], Word2Vec [6, 7], and GloVe [15] are examples of the most popular embedding approaches. Our preliminary investigation showed that the Word2Vec is more preferred for the Turkish language. Briefly, Word2Vec is a neural network model consisted of two layers, wherein our case its input is the collected tweets, and its output for each word, as shown in Equation (1), is a vector of 300 float number(f)

$$\text{Embedding}\ (\text{word}_i) = [f_1, f_2, f_3, \ldots., f_{300}]. \tag{1}$$

3.4. Classification

In this work, the performance of K-Nearest Neighbors (KNN), Naïve Bayes (NB), Support Vector Machine (SVM), and some ensemble learning classifying algorithms, i.e., Random Forest, AdaBoost Classifier, and Gradient Boosting classifier have been investigated. In the following, the details of these algorithms are summarized.

i. K-Nearest Neighbor (KNN)

KNN is one of the simplest classification algorithms that classify the current data sample based on the proximity of the K previously classified ones [16, 17]. In the early 1970s, KNN became very popular to be used for pattern recognition and statistical estimation tasks. Related to this study, each tweet will be assigned to the most common class among its nearest neighbors [16].

ii.Naïve Bayes (NB)

Naive Bayes classification algorithm classifies the data based on Bayes theorem. The main purpose of this theorem is to establish a proportional relationship for the class's conditional probabilities for the given data, which is used to determine its probability(membership) to the predefined classes. In other words, probabilities for each class are estimated and calculated using the training set, where in each iteration, if the algorithm encounters a new sample, the previously obtained probabilities are used to find the sample membership probability and the class that maximizes the probability is selected [18].

iii.Support Vector Machine(SVM)

Support Vector Machine (SVM), is one of the supervised machine learning algorithms that can be used for classification and regression analysis. In general, it performs the classification process by finding some boundaries (called hyperplane) that maximize the margin (distance) between the classes, where each data is mapped as a point in high dimensional space using a nonlinear method, and in this new space, the classification is carried out according to the hyperplane, that can distinguish between the classes quite well with more confidence [19].

In addition, the state-of-the-art ensemble based algorithms are briefly summarized [20-24].

i.Random Forest (RF)

Random Forest algorithm is an ensemble classifier method that uses the decision trees as its main classifier [21]. Usually, the random forest ensemble is constructed using a large number of uncorrelated individual DT. Each tree in the system predicts a class for the current sample, then, DTs come together to form the decision, i.e., the class with the most votes becomes the model's prediction. Also, it uses the bagging (bootstrap aggregating) [22] technique to manage the learners.

ii.AdaBoost Classifier (AdaBoost)

Adaboost algorithm was developed by Robert Schapire and Yoav Freund in 1996, it is an ensemble classifier method that creates a strong classifier by linearly combining weak learning algorithms[23]. During training, a weight coefficient is assigned to each sample, and the weak classifier(s) are further trained, to tune its parameters, in each iteration, where the weights of the incorrectly classified samples are increased while the weights of the correctly classified samples are reduced. Overall, the general error is calculated by evaluating the weights of the weak classifiers, then the subsequently added classifier is trained on data that the previous ensemble members failed to classify it correctly.

iii.GradientBoosting Classifier (GBC)

Gradient boosting is another famous ensemble learning approach that is frequently used for regression and classification problems. GBC produces a prediction model by sequentially fitting the base learner to current "pseudo"-residuals by least-squares at each iteration. As shown in [24], the pseudo-residuals are the gradient of the loss functional being minimized, with respect to the model values for each training data point evaluated at the current step. Whereat each iteration, the estimates are updated so that the sum of the residuals is close to zero and the predicted values are close to the true values.

4. EXPERIMENTAL EVALUATION AND ANALYSIS

In this section, multiple experiments were performed to investigate the performance of the proposed system for supporting the Turkish language. In the first experiment, the performance of the KNN, NB, and SVM was investigated. In addition, the performance of building an ensemble system using KNN, NB, SVM, and the performance of following state-of-the-art ensemble approaches RF, AdaBoost, GBC was also studied in the second experiment. Furthermore, the performance of the proposed system which was built based on the results of our intensive investigation was observed in the last two experiments.

In this work, four sub-datasets were constructed from our under-development Turkish dataset, the first dataset related to the Yangın(fire) and contains 2000, the second one is related to the Deprem (earthquake) and contains 4300 tweets, the third one is related to the trafik kazası (traffic accident) that contains 5700, and lastly, the fourth dataset is related to the Sel ("Flood") that contains 1000 tweets. The constructed sub-datasets has an equal number of disaster and non-disaster tweets, i.e., balanced.

Related to the evaluation, we have used the following main standard metrics that are well known used for evaluating classification systems: A) Accuracy, refers to the ratio of the tweets that were correctly classified (Equation (2)). B) Precision refers to the ratio of the correct predictions of the number of total predictions and can be obtained using Equation (3). C) The recall represents the ratio of the correct predictions and the total number of correct tweets in the dataset (Equation (4)). D) F1 score, i.e., the average of both Precision and Recall. F1 can be obtained using Equation (5)

$$\text{Accuracy} = \frac{TP+TN}{\text{N}} \tag{2}$$

$$\text{Precision} = \frac{TP}{TP+FP} \tag{3}$$

$$\text{Recall} = \frac{TP}{TP+FN} \tag{4}$$

$$\text{F1 Score} = \frac{2*\text{Recall} * \text{Precision}}{\text{Recall} + \text{Precision}} \tag{5}$$

where the total number of tweets is represented by N. The True Positive, i.e., the number of correctly predicted tweets as disasters is represented by TP, the True Negative, i.e., is the number of correctly predicted tweets as not-disasters is represented by TN, and the number of disasters tweets which are predicted as not-disasters (False Negative) is represented by FN.

4.1. Implementation Details

This approach was implemented using Python, and some libraries such as Keras, Pandas, NumPy. Related to processing the Turkish language, the "turkish-deasciifier" library was used for the deasciification process. In addition, text normalization, and stemming were done using the "TurkishStemmer". Furthermore, the preprocessing and feature extraction functions were imported from both Keras and Sklearn.

Related to the studied classifiers, KNN, Naïve Bayes, SVM, RF, AdaBoost, and GBC were imported from Sklearn, where the "K" value for the KNN, was defined after testing the values in the range [1, 20], and the test was done using the Grid Search that has the ability to find the best K value. On the other hand, the SVM kernel was set to 'rbf' and its two other parameters, i.e., 'C' and 'gamma' were set by testing [0.1, 1, 10, 100, 1000], [1, 0.1, 0.01, 0.001, 0.0001] respectively. The same process was done to detect the value of the "number of estimators" for the RF Classifier by testing the values {50, 100, 150, and 200}. Finally, the number of trees was 100 for both AdaBoost and GBC.

It is worth mentioning that to ensure the robustness of the results, the 10-fold cross-validation was used to obtain the results of all experiments.

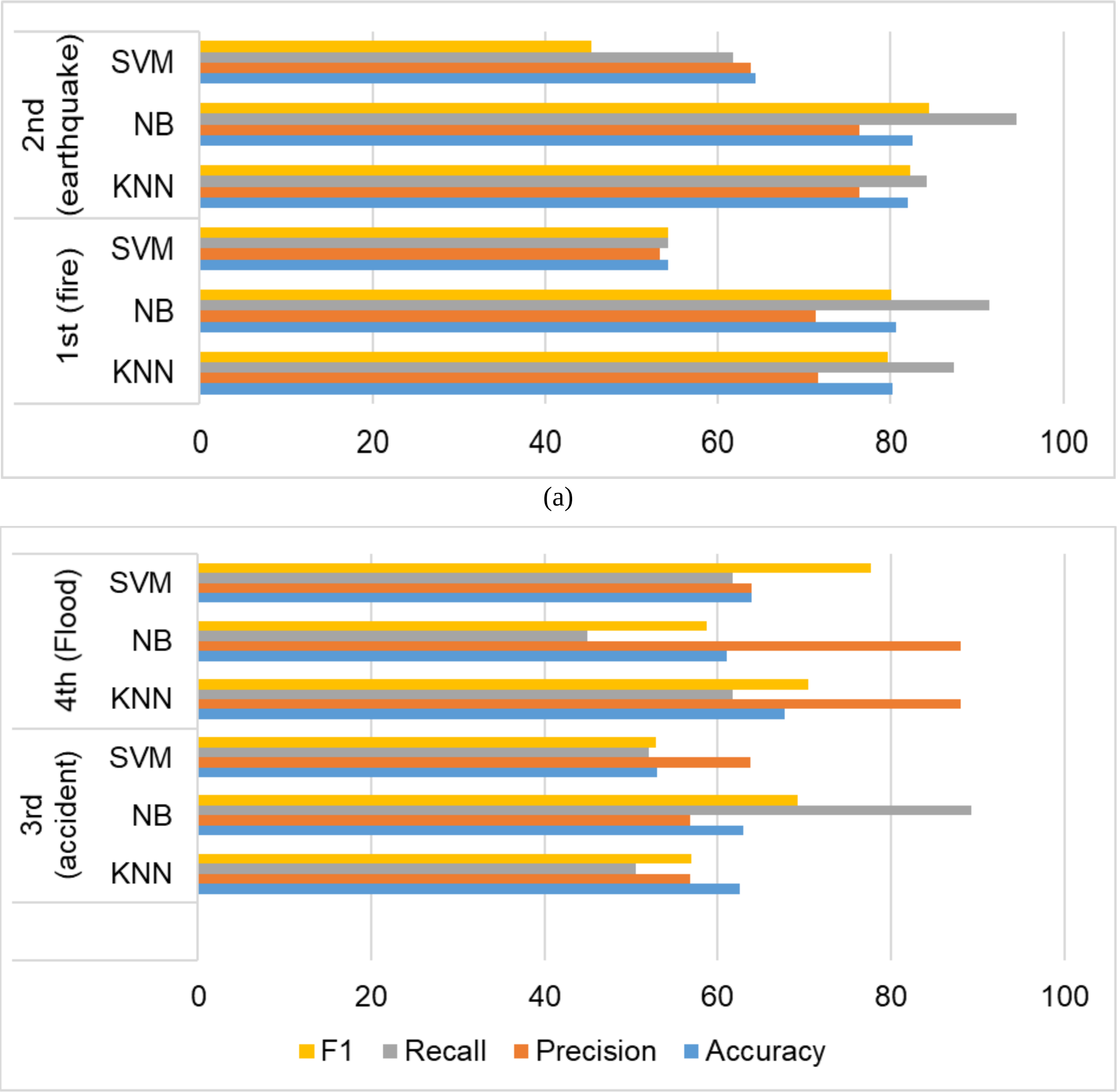
<u>Experiment 1:</u> KNN vs. NB vs. SVM

In this experiment, the robustness and scalability of the mentioned techniques were investigated using the four balanced sub-datasets. Also, in addition to the accuracy, the Precision, Recall, and F1 score, which as mentioned before considers both false positives and false negatives into account has been calculated. Figure 3 (a) shows the performance of the studied algorithms using the first two datasets, and Figure 3 (b) shows the performance of the studied algorithms using the last two datasets. The results can be summarized as follows.

1) For the first three datasets, the NB was able to achieve the highest, while KNN achieved second best, and surprisingly SVM achieved the lowest performance when it is required to process the Turkish language. In addition, it is worth mentioning that NB requires almost half the execution time required by the other two methods.

2) Related to the fourth datasets, which contains one thousand tweets, all algorithms achieved a low performance and the NB got the lowest values. This can be due to the fact that the number of tweets is not enough to train the system. It is worth mentioning that we are currently collecting more tweets related to the topic of this dataset (Flood), and this dataset will not be used for the following experiments.

3) Overall, none of the studied algorithms can be used individually to build an efficient classification system that can handle the Turkish language.



(a)

(b)

Figure 3. *The Accuracy, Precision, Recall, and F1 for the studied algorithms using four different size datasets*

Experiment 2: Performance of Ensemble Systems

As shown in the previous experiment, none of the studied algorithms can be used individually to build an efficient classification system that can handle the Turkish language. On the other hand, it is expected that ensemble learning can overcome the weakness of using the classifiers individually and it has the ability to make the system more accurate, robust, and scalable. In this experiment, the performance of an ensemble system that was consisted of KNN, NB, SVM, and some state-of-the-art ensemble algorithms, i.e., RF, AdaBoost, and GBC was investigated. In more detail, related to the AdaBoost, and GBC, the performance of the default version of these algorithms, i.e., using its base classifier, and its performance after using different base classifiers such as KNN, NB, SVM, etc. was also studied. It is worth mentioning the following:

1) The RF is unlike the other two, was developed based on using the decision tree as a base classifier, and it can't work with any other base classifiers.
2) The structure of AdaBoost does not accept the KNN as a base classifier, similarly, GradientBoosting does not accept the SVM. This due to its structures and the fact that both were built to use weak-classifier. As an alternative, the Logistic regression, which is one of the famous weak-classifier was used in this experiment.

Table 1. *The accuracy of the studied ensemble systems*

Ensemble Learning #	Ensemble Algorithms	Base Classifier	Accuracy (%)		
			First Dataset	Second Dataset	Third Dataset
1st	RF	Default	93.40	89.78	88.44
2nd	AdaBoost	Default	81.10	84.03	80.01
3rd		NB	62.40	60.87	49.19
4th		SVM	73.20	65.49	52.77
5th		Logistic Regression	81.40	84.37	80.21
6th	GBC	Default	82.00	84.56	81.45
7th		KNN	55.20	63.81	65.38
8th		NB	72.08	72.09	70.98
9th		Logistic Regression	77.80	72.19	71.53
10th	{KNN, NB, SVM}		82.40	84.28	83.99

Based on Table 1, the followings can be derived:

1) The results (accuracy) of all the studied ensemble systems using its own base classifier are good and it is clear that such a system is more stable as compared to any single classifier.
2) Overall, the random forest ensemble system can be considered as the best one, as it has achieved the highest accuracy for all the datasets. On the other hand, the ensemble system consisted of {KNN, NB, SVM} was able to achieve the second best performance and outperformed both AdaBoost and GBC.
3) Related to changing the base classifier, we have found that such a process has decreased the performance of GBC using all the studied algorithms. Almost the same situation is done for the AdaBoost, except when using the Logistic Regression, which is as shown before can be considered as the weakest classifier among the studied ones. But, Logistic Regression was able to achieve a bit higher accuracy as compared to the base one.

Experiment 3: Performance of the Proposed System

In this experiment, the robustness and scalability of the proposed approach were investigated using the three constricted Turkish datasets. Also, in addition to the accuracy, the Precision, Recall, and the F1 score have been calculated. In addition, to ensure the robustness of the results, this experiment has been executed five times (5 iterations). Table 2 shows the results of this experiment, and Figure 4 shows the average of the obtained results. Overall, the findings of this experiment can be summarized as follows:

1) Over 90% accuracy has been achieved by the system developed for the Turkish. Hence, the proposed system has the ability to process Turkish datasets effectively.
2) With regard to robustness and scalability, while increasing the number of processed tweets, our system has shown that it is stable and can handle all datasets efficiently.

Table 2. *The Accuracy, Precision, Recall, and F1 Score for the Proposed System*

Iteration	Dataset	Accuracy	Precision	Recall	F1
1st	1st	94.9	97.31	89.78	93.21
	2nd	91.26	92.71	91.46	90.45
	3rd	92.92	90.36	95.2	93.41
2nd	1st	94.8	95.73	89.96	92.51
	2nd	91.41	91.66	88.72	90.10
	3rd	92.92	89.97	95.26	93.24
3rd	1st	94.6	96.14	90.69	93.02
	2nd	91.25	92.48	91.16	91.06
	3rd	92.92	90.52	99.4	93.06
4th	1st	95.2	99.41	92.86	93.23
	2nd	91.88	92.47	88.69	90.48
	3rd	93.48	90.66	95.57	93.75
5th	1st	94.9	97.20	87.26	91.80
	2nd	91.35	91.55	90.45	90.89
	3rd	93.41	88.54	95.71	93.73

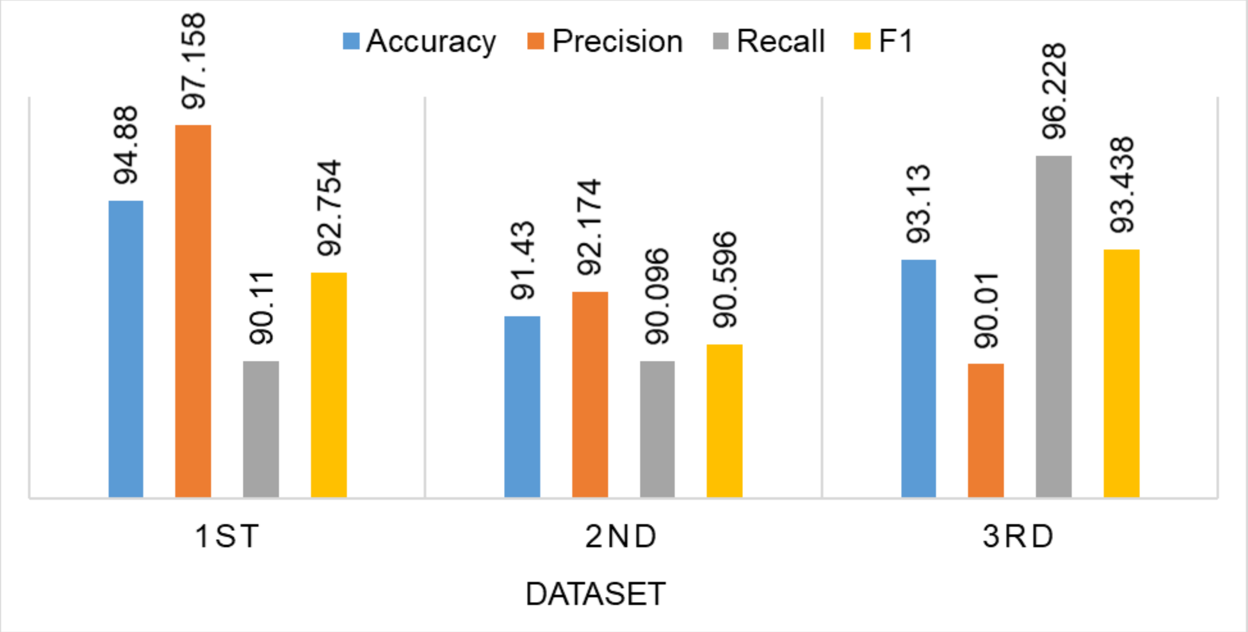


Figure 4. *The Average of Accuracy, Precision, Recall, and F1 for the Proposed System*

Experiment 4: Performance of the Proposed System vs. the Baseline Ensemble System

In this experiment, the performance of the proposed approach was compared with the performance of the best ensemble system that we founded in Experiment 2, i.e., the random forest ensemble system. Figure 5 shows the accuracy of both systems, and it is obvious that the proposed approach has significantly outperformed the mentioned ensemble system. It is worth mentioning that the proposed approach was able to achieve on average 5% as improvement comparing to the RF, which was obtained as a result of integrating both the developed preprocessing model and the Word2Vec.

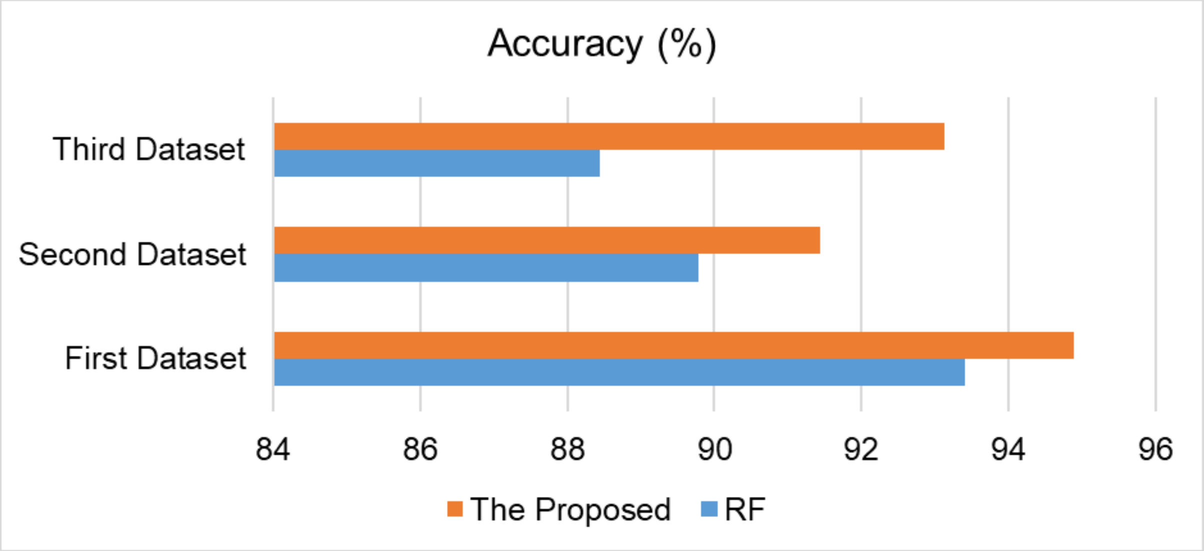


Figure 5*. The Accuracy of the Proposed System and the Random Forest (RF)*

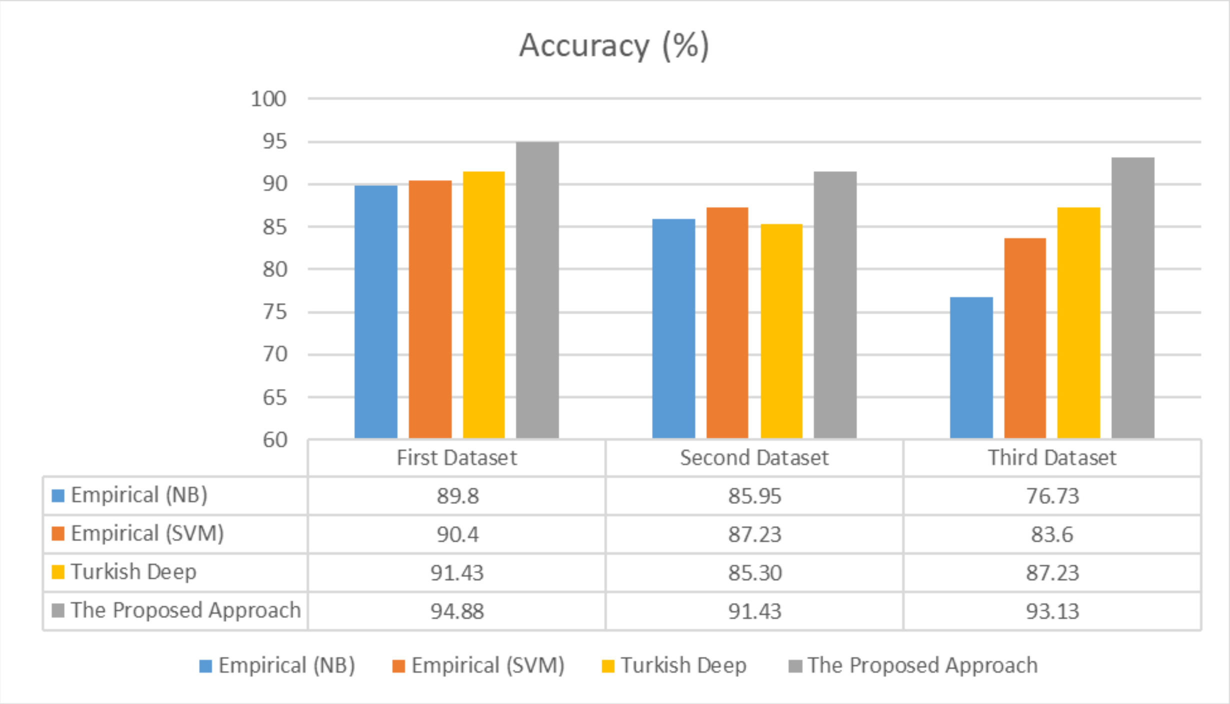
<u>Experiment 5:</u> Performance of the Proposed System vs. *"Empirical" and " Turkish Deep"* Approaches.

In this experiment, the performance of the proposed approach was compared with the approaches presented in [12] and [25]. In detail, we have re-implemented the *Empirical* approach [25], which uses the TF-IDF to represent the text features. Then, this approach uses the Chi-Square for feature selection. In other words, Chi-Square is calculated between each feature and the class. Then, a vector of N features with the highest Chi-Square and non-negative values is selected as the tweet features. Next, both of the SVM and NB was used as the classifier for [25]. Note that we have used 100 as the value of N in this experiment.

Also, we have re-implemented the CNN model presented in [12], which was built for processing the Turkish sentiment analysis task. As mentioned in the related work section, this model, which we refer to it as “Turkish Deep”, contains 6 dense layers, where each one has different output dimensions and a 0.5 Dropout is allocated after the second and the fourth layer, while 0.2 Dropout is allocated after the third one.

Figure 6 shows the experiment results, and it is clear that the proposed approach has outperformed both approaches. Also, related to [24], the SVM, when used as the classifier, was able to outperform the NB classifier and achieve second-best results for the second dataset.

Related to the CNN model of [12], it has achieved the second-best for the first and third datasets, however, as mentioned in the related work section, CNN models are currently the state-of-the-art models for many classification tasks including the text classification. We believe that CNN achieved this results due to lack of a large number of training samples, wherein the case of the used datasets the generated features during the training of CNN are not generalizable enough mainly due to the large number of the CNN’s variables(weights) that need to be trained. In addition, it is suggested that such a CNN model can be further improved by investigating the performance of other types of layers such as Convolutional and embedding layers and increasing the number of training samples. Hence, one of the essential advantages of the ensemble learning algorithms compared to the CNN models is the ability to achieve acceptable performance using less number of training samples.



	First Dataset	Second Dataset	Third Dataset
🟦 Empirical (NB)	89.8	85.95	76.73
🟧 Empirical (SVM)	90.4	87.23	83.6
🟨 Turkish Deep	91.43	85.30	87.23
⬜ The Proposed Approach	94.88	91.43	93.13

Figure 6*. The accuracy of the proposed system and the approach of "Empirical" and " Turkish Deep "*

5. CONCLUSIONS AND FUTURE WORKS

It is obvious that social media analytics can have an impressive effect on many important research field such as internet-based crisis management systems. The main goal of this work was to investigate the possibility of building an efficient system that can classify and identify social media data related to the crisis efficiently, in order to inform the authority asap (almost a real-time response) to gain situational awareness, which may help in preventing or decreasing the effect of some disaster by taking the correct responses.

To achieve our goal, the performance of the K-Nearest Neighbour (KNN), Support Vector Machine (SVM), and Naïve Bayes (NB), in addition to some state of the are ensemble algorithms, i.e., Random Forest (RF), AdaBoost Classifier (AdaBoost), GradientBoosting Classifier (GBC), when using for text (tweets) classifying, has been investigated in details. Unsurprisingly, the ensemble algorithms have significantly beaten the individual classifiers. Hence, it is clear that ensemble learning techniques can improve the performance, robust and stability of the classifying process. Based on this and our pre-investigation, a new effective tweet classification system that fully supports the Turkish language has been developed.

Overall, with the luck of having large datasets that can be used to train a deep learning approach that can achieve a better performance, it is clear that ensemble learning and the proposed system can be used for mining (classifying) the recently published and publicly available tweets to find the crisis's most related and useful tweets to gain situational awareness, which can help in taking the correct responses in order to prevent or at least decrease the effect of such situations. In addition, the proposed approach has the ability to process the Turkish language while achieving very good performance, robustness, and stability.

This study can be expanded in many ways starting with integrating state-of-the-art CNN models, which we believe can further improve building a crisis management system that supports the Turkish language. In addition, building an efficient sub-model for tweets pre-processing is essential and can improve the overall performance significantly.

CONFLICTS OF INTEREST

No conflict of interest was declared by the author.

REFERENCES

[1] Domala, J., Dogra, M., Masrani, V., Fernandes, D., D'souza, K., Fernandes, D., Carvalho, T., “Automated Identification of Disaster News for Crisis Management using Machine Learning and Natural Language Processing”, In 2020 International Conference on Electronics and Sustainable Communication Systems (ICESC), 503-508, (2020).

[2] Alshehri, A., Alahamri, S., “An Ensemble Learning for Detecting Situational Awareness Tweets during Environmental Hazards”, In 2019 IEEE International Systems Conference (SysCon), 1-8, (2019).

[3] Kumar, A., Singh, J. P., Saumya, S., “A Comparative Analysis of Machine Learning Techniques for Disaster-Related Tweet Classification”, In 2019 IEEE R10 Humanitarian Technology Conference (R10-HTC), 222-227, (2019).

[4] Nalluru, G., Pandey, R., Purohit, H., “Relevancy classification of multimodal social media streams for emergency services”, In 2019 IEEE International Conference on Smart Computing (SMARTCOMP), 121-125, (2019).

[5] Ayata, D., Saraçlar, M., Özgür, A., “Turkish tweet sentiment analysis with word embedding and machine learning”, In 2017 25th Signal Processing and Communications Applications Conference (SIU), 1-4, (2017).

[6] Naili, M., Chaibi, A. H., Ghezala, H. H. B., “Comparative study of word embedding methods in topic segmentation”, Procedia Computer Science, 112, 340-349, (2017).

[7] Mikolov, T., Chen, K., Corrado, G., Dean, J., “Efficient estimation of word representations in vector space”, arXiv preprint arXiv: 1301.3781, (2013).

[8] Şahin, G., Turkish document classification based on Word2Vec and SVM classifier”, In 2017 25th Signal Processing and Communications Applications Conference (SIU), 1-4, (2017).

[9] Aydoğan, M., Karci, A., “Improving the accuracy using pre-trained word embeddings on deep neural networks for Turkish text classification”, Physica A: Statistical Mechanics and its Applications, 541, 123288, (2020).

[10] Kılınç, D., Özçift, A., Bozyigit, F., Yıldırım, P., Yücalar, F., Borandag, E., “TTC-3600: A new benchmark dataset for Turkish text categorization”, Journal of Information Science, 43(2), 174-185, (2017).

[11] Kılınç, D., “The Effect of Ensemble Learning Models on Turkish Text Classification”, Celal Bayar Üniversitesi Fen Bilimleri Dergisi, 12(2), (2016).

[12] Demirci, G. M., Keskin, Ş. R., Doğan, G., "Sentiment Analysis in Turkish with Deep Learning", In 2019 IEEE International Conference on Big Data (Big Data), 2215-2221. IEEE, (2019).

[13] BaygIn, M., “Classification of text documents based on Naive Bayes using N-Gram features”, In 2018 International Conference on Artificial Intelligence and Data Processing (IDAP), 1-5, (2018).

[14] Joulin, A., Grave, E., Bojanowski, P., Douze, M., Jégou, H., Mikolov, T., “Fasttext. zip: Compressing text classification models”, arXiv preprint arXiv: 1612.03651, (2016).

[15] Pennington, J., Socher, R., Manning, C. D., “Glove: Global vectors for word representation”, In Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP), 1532-1543, (2014).

[16] Cunningham, P., Delany, S. J., “k-Nearest neighbour classifiers”, Multiple Classifier Systems, 34(8), 1-17, (2007).

[17] Nikhath, A. K., Subrahmanyam, K., Vasavi, R., “Building a K-Nearest Neighbor Classifier for Text Categorization”, International Journal of Computer Science and Information Technologies, 7(1), 254-256, (2016).

[18] Frank, E., Bouckaert, R. R., “Naive bayes for text classification with unbalanced classes”, In European Conference on Principles of Data Mining and Knowledge Discovery, Springer, Berlin, Heidelberg, 503-510, (2006).

[19] Dadgar, S. M. H., Araghi, M. S., Farahani, M. M., “A novel text mining approach based on TF-IDF and Support Vector Machine for news classification”, In 2016 IEEE International Conference on Engineering and Technology (ICETECH), 112-116, (2016).

[20] Dietterich, T. G., “Ensemble methods in machine learning”, In International Workshop on Multiple Classifier Systems, Springer, Berlin, Heidelberg, 1-15, (2000).

[21] Onan, A., Korukoğlu, S., Bulut, H., “Ensemble of keyword extraction methods and classifiers in text classification”, Expert Systems with Applications, 57, 232-247, (2016).

[22] Elith, J., “Machine Learning, Random Forests, and Boosted Regression Trees”, Quantitative Analyses in Wildlife Science, 281, (2019).

[23] Rodriguez, J. J., Kuncheva, L. I., Alonso, C. J., “Rotation Forest: A new classifier ensemble method”, IEEE Transactions on Pattern Analysis and Machine Intelligence, 28(10), 1619-1630, (2006).

[24] Ke, G., Meng, Q., Finley, T., Wang, T., Chen, W., Ma, W., Liu, T. Y., “Lightgbm: A highly efficient gradient boosting decision tree”, In Advances in Neural Information Processing Systems, 3146-3154, (2017).

[25] Ragini, J. R., Anand, P. R., “An empirical analysis and classification of crisis related tweets”, In 2016 IEEE International Conference on Computational Intelligence and Computing Research (ICCIC), 1-4, (2016).