Anti-social behaviour analysis using random forest and word to vector approach

Nidhi Chandra*, Sunil Kumar Khatri and Subhranil Som

Amity School of Engineering and Technology,
Amity University, Noida,
Uttar Pradesh, India
Email: nsrivastava5@amity.edu
Email: sunilkkhatri@gmail.com
Email: ssom@amity.edu
*Corresponding author

Abstract: Social Networking and micro-blogging applications provide active platforms for communications, sharing thoughts and ideas. Processing natural text coming from varied social platforms possess many technical challenges such as processing messages written in slang, informal short messages, classifying messages into different labels and category based on the meaning. Maximum natural text processing and interpretation systems use *n*-gram language models, which can be simple and powerful most of the time. Random forest ensemble-based classifier has the potential to generalise the unseen data as compared to *n*-gram language models. Anti-social messages are a significant problem in social media. In this paper we present an approach to classify the natural language text as anti-social text using Random Forest classifier. In this paper we are addressing the challenge to identify anti-social messages using this algorithm using vector ensemble technique to classify anti-social text in offline mode. Word to vector approach has been used for word embeddings to train the model. This paper combines word to vector approach with random forest classifier using a multilayer network.

Keywords: natural language processing; random forest; ensemble classifier; anti-social behaviour analysis; word to vector.

Reference to this paper should be made as follows: Chandra, N., Khatri, S.K. and Som, S. (2022) 'Anti-social behaviour analysis using random forest and word to vector approach', *Int. J. Applied Management Science*, Vol. 14, No. 1, pp.38–56.

Biographical notes: Nidhi Chandra is a faculty at Department of Computer Science & Engineering, ASET and a Research Scholar of Amity University, Uttar Pradesh, India. She obtained her Master's degree in Computer Science from Guru Gobind Singh Indraprastha University, Delhi. With over 14 years of academic experience and 2 years of industry experience, her research interest lies in natural language processing, semantic analysis and data mining.

Sunil Kumar Khatri is working as Director of Campus, Amity University Tashkent, Uzbekistan at Amity Education Group. He is a fellow of IETE, Sr. Life Member of CSI, IEEE, IASCSIT and Member of IAENG. He specialises in software reliability and testing, data mining and warehousing, network security, soft computing and pattern recognition. He has been conferred "IT Innovation & Excellence Award for Contribution in the field of IT and

Computer Science Education" by Knowledge Resource Development & Welfare Group at IIT, Delhi in 2012.

Subhranil Som has 14 years of teaching and research experience and is working as Associate Professor in Amity University, Noida (NCR), India. He completed PhD in Computer Science and Engineering from University of Kalyani, West Bengal in the year of 2012. He was as a Principal Investigator of an UGC funded project. His specialisation includes cryptography and network security and robotics. He was attached with a WHO's International Research Project on "e-Health for Health Care Delivery", University of New South Wales, Sydney, Australia. He is holding prestigious memberships of various reputed international technical and research organisations.

1 Introduction

There are multiple sources of text data and it spreads across in a web of documents over the internet and otherwise. About 85 to 95% of data is estimated to be in an unstructured format which also comprises of this text and one can find it through internet searches such as Google, Yahoo, etc. To find insight into this large proportion of data one needs to perform refining and analysis of this textual data using various data science methodologies and tools. The first step is to convert text into data frames using various API's in languages such as *R* and Spark, etc. following which one can go ahead with text transformation. The text transformation comprises of text cleansing which involves word stemming, word replacement, punctuation and stop-word removal, word case transformation, etc.

Several techniques and methodologies have evolved in the area of text mining. These techniques provide the basic fundamental framework to achieve results in this area. The framework might not over all the steps and methods but critical cover steps. Additionally there are modelling methodologies in a succinct and clear manner as they might be complicated. The process of data gathering and its compilation is a huge topic in itself and is beyond the scope of this document but the preferred approach would be to use some kind of tidy framework which would help us to leverage tibbles (*R* data frames) or standard data frames in many of our steps and tidytext functions to help in transition to different text mining structures like corpus.

As a first step we need to transform text files into data frames, post which data preparation can begin with text transformation. Many steps have been applied on the natural text or transform the natural text such as Change case where capital letters are changed to lowercase, numbers, punctuation's, Stopwords and whitespaces have been eliminated from the natural text and word will be replaced by the synonyms and word will be rolled back to the root word.

With the help of these transformations one can create compact datasets which helps simplifying the structure which further helps identifying relationships between words thereby helping in enhanced understanding.

One of the strong methods which help to group documents as per their main topic involves creating Topic models. These topic models use the concept of probabilistic modelling of term frequencies that occurs in a specific document. This particular fitted model can be leveraged for estimation of document similarities and uses the layer of

latent variables to identify similarities between specific keywords. Document can be classified or categorised based on the distribution of terms in the document or by comparing with the document which have approximately similar term distribution.

To perform quantitative analysis of text one needs to do semantic analysis of sentences and word tagging based on parts of speech. There are eight parts of speech in the English language: noun, pronoun, verb, adjective, adverb, preposition, conjunction, and interjection. The part of speech indicates how the word functions in meaning as well as grammatically within the sentence. An individual word can function as more than one part of speech when used in different circumstances. Understanding parts of speech is essential for determining the correct meaning of the word. Semantics of word and statistical property of word for example term frequency would be sufficient for analysing text.

Opinion mining or Polarity identification many times is called as sentiment analysis. It helps know the positivity and negativity of text. While analysis of polarity in *R* language, it allocates scores to each word to help in analysis through average and standard deviations assigned to polarity by groups such as various authors, text and topics through the use of in build polarity dictionaries.

The algorithm leverages polarity dictionaries to tag words with positive classified, negative classified or neutral classified sentiments. The clustering of tagged words is done with four words before and two word after the tagged words. These clusters are then tagged with valence shifters which could be neutral, negate, amplified and de-amplified. The sequence of weights based on their numeric values and locations are then applied to both terms and group of similar terms. The formulation is then done through summation which is then divided by square root of the number of words in that sentence.

The term formality measure helps share an understanding on text relations with the reader or listener's speech. This helps to understand the comfort zone of the text producer person against the audience or understanding the location where communication had taken place. The key property of informal text is that it is contextual in nature. This formality measure is called *F*-Measure. Diversity in relation to text mining is referred to as count of different words being used against the total number of words used.

The useful tool to understand the distribution of words throughout the document to explore text and figuring out patterns is dispersion or lexical dispersion, through which analysis is conducted with the call of specific word or words of interest. The result can be plotted between terms or occurrence of terms in the text over time for analysis and predictions.

1.1 Random forest

Random forest algorithms are based on the principle of result aggregation of multiple predictors to help in good predictive analysis of data compared to any specific predictor. Collections of predictors are referred to as ensemble and hence the overall technique is referred to as Ensemble learning.

Random forest many times also referred as random decision forests are used to build efficient predictive models for classification and regression problems. This method also referred as Ensemble method leverages different learning models to get best predictive results. In this case, the model builds forest of random uncorrelated decision trees to reach for best possible results.

One of the key benefits of ensemble learning model is that it makes use of many machine learning models to get the better performance. Each individual model is weak by in itself, however when used in combined strength its model becomes strong thereby generating more accurate results.

Decision trees works on similar lines as tree-based data structures where in the root node creates binary splits moving down until specific criteria is met. This is also referred to as Top-Down approach. This binary splitting helps in providing predictive value based on internal nodes which leads to final node. In the context of classification, the output of decision tree is a predicted target class for all created terminal nodes. Decision trees show cases high variance while utilising varied training and test sets for similar data due to their over fitting on this data. This sometimes tends to show bad performance on un-seen data. This is one of the main reasons for limited usage of this algorithm in predictive modelling. However application of ensemble methods helps us to create models that makes use of underlying decision trees as a base to generate excellent result.

Using the process of bootstrap aggregation one can create ensemble of trees by generating multiple training sets with replacement. With these training sets CART – Classification and Regression tree model can be trained for each subsample.

Bootstrap Aggregation or Bagging is one of the very powerful ensemble methods. Using this approach one can reduce variance by averaging the ensemble's result through the creation of majority-votes model. Bagging trees do not needs to be pruned for growth which results in high variance and lower bias which is one of the capabilities that helps in fine-tuned predictive power. The drawback of this algorithm is that feature space utilisation creates correlation risk between trees thereby increasing model bias.

The bagging tree has a limitation that it makes use of complete feature space during creation of splits in the trees. One runs the risk of forest of correlated trees due to the fact that some variables in feature space are indicating certain predictions, thereby running into the risk of generating forest of correlated trees. This leads to increasing if bias and reducing variance. The tweaking of bagging tree methodology is advantageous for model's predictive power.

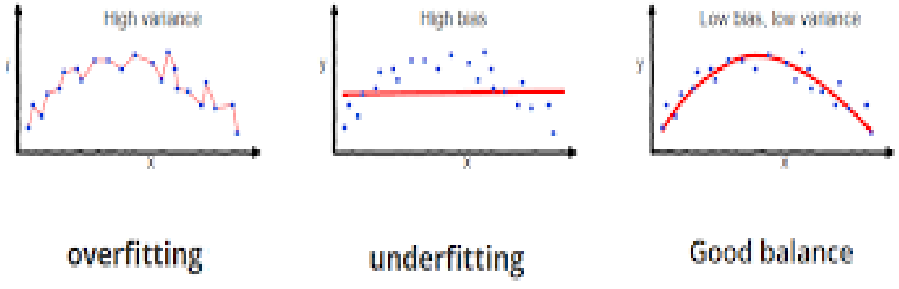
One of the aims of Random forest is to decrease correlation issues mentioned earlier through utilisation of subsample of feature space for each split. The aim of this is to de-correlate and prune trees through stop criteria for node split. For example – large number of decision trees are considered strong as compared to single decision trees. The outputs are aggregated when large decision trees are used with results strong ensembles.

The strength and weakness of algorithm depends on its bias and variance exhibited by all machine learning models as depicted in Figure 1. These are measured by training machine learning model on data which is divided in multiple sections and performing comparative analysis of generated outputs to the actual output values of data. Bias is a measure of delta between predicted values and actual values and it occurs when many simplistic assumptions are made by algorithm.

The random forest algorithm's functional objective is to work on high-variance and low-bias decision trees and convert them into low-bias and low-variance model. Through the process of aggregation multiple outputs of each decision tree, the random forest algorithm helps to reduce the variance which on many occasions creates erroneous decision trees. With the help of majority voting technique one can perform averaging of results generated by individual trees. This helps prevent generating results which drifts away from the actual values. This creates variations between actual and predicted values. Variance is a measure of scattering of predicted values from actual values. Performance

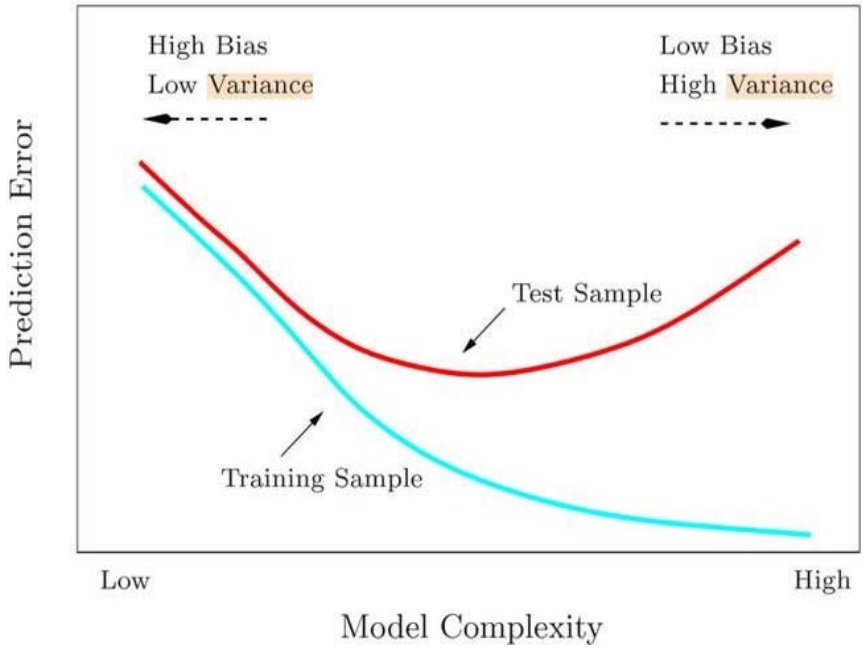
of model on trained data set. It occurs due to adverse performance of model on trained dataset against untrained data sets. The influence of algorithm through data specifics is determined by higher variance.

Figure 1 High bias, high variance and just fit



Under ideal conditions both bias and variance should be low indicating the fact that prediction values of model are closer to correct values for different data across same dataset. High variance results in over fitting leading to the fact that machine learning algorithms tends to model random noise available in training data set. Models emanating from high-variance data are complex and difficult to model. This can be validated from graph below with *f* (model complexity) and error of test and training predictions as shown in Figure 2.

Figure 2 Bias/variance trade-off



The decision trees are trained by random forest through subsets of training data where in the decision tree split is done through random selection of attributes of data. The randomness of data helps make sure that ML models are different from one another, resulting in spread of potential errors throughout model and cancelled by the majority voting decision strategy.

Let's try to understand application of random forest through an example. Let's consider an example of book publisher website which takes recommendations from buyers for its titles. The search and selection of each individual buyer is considered as decision tree with its data captured. The feedback data is captured through an online form and applied to random forest algorithm for each individual book from multiple users. This helps provides baseline recommendations for each individual books one needs to buy.

The errors occur buyer's feedbacks are guided by their own biases. To encounter this kind of situations the suggestions of users (decision tree) are combined and majority vote decision is applied on their suggestions creating a random forest. There are situations where in similar data is provided by multiple buyers resulting in highly biased and correlated data which also lacks variance. To cater to such kind of scenario users are provided with fixed wider range of recommendations so that they have minimum criteria to make for recommendations. Applying majority vote casting facilitates to eliminate outliners thereby offering accurate and varied list of tips of books.

2 Literature survey

The perfectness of classification and regression algorithms depends upon accuracy and execution of Random Forest algorithm. Since Random Forest comes under ensemble techniques, its precision and execution could be improved through test of its base classifier. Thereby to have a perfect ensemble model there should be diverse, precise and accurate base classifiers.

The random forest area has numerous Meta studying strategies connected to it. These strategies are primarily based on the fundamentals that base classifiers forms random forest area set of rules. Current random forest algorithm model's overall performance need to be examined and in comparison in opposition to this model.

Many more progressions strategies had been implemented on random forest area as semi-supervised random forest area, rotational forest area and fuzzy random forest. Random forest area does no longer require unique parallel classifiers for the evaluation of multi-class issue. However, it has some impediment that it needs expansive measure of named records to move to the high-quality dimension of execution. Along these strains, to triumph over this problem Semi-Supervised Random forest calculation was proposed by Leistner et al. (2009)

The accuracy of classification can be increase by joining the individual classifiers to frame Multiple Classifier Systems. Multiple Classifier Systems are primarily based on a forest formed by fuzzy decision trees known as Fuzzy Random Forest. This method is proposed by Bonissone et al. (2010)

Freedom of expression is one of the most aggressively contested rights of the cutting-edge international. Even as censorship of unfastened shifting on line content consisting of Twitter tweets curtails the freedom of speech, but unregulated opprobrious tweets discourage loose discussions within the digital model (Silva et al., 2016). Hate speech

detection is a tough studies hassle because of ambiguity inside the clear demarcation of offensive, abusive and hateful textual content due to variations inside the way human beings express themselves in a linguistically diverse social putting. A first-rate venture in tracking online content produced on social media websites like Twitter, Facebook and Reddit is the humongous quantity of statistics being generated at a quick tempo from varying demographic, cultural, linguistic and non-secular communities.

Many more approaches have been applied to identify natural text that is coming under the category of cyber trolling, cyber criticism or cyber threat.

Imran et al. (2015, 2016a, 2016b) demonstrated human-annotated Twitter corpora collected all through 19 unique crises that occurred among 2013 and 2015. To demonstrate the utility of the annotations, we educate system getting to know classifiers. Moreover, we submit first biggest word2vec word embedding's educated on 52 million crisis-related tweets. To deal with tweets language issues, the case study introduced human-annotated normalised lexical assets for extraordinary lexical versions. To make feel of massive amounts of Twitter messages posted during crises, the approach applied a simple operation, that is, the automatic categorisation of messages into the types of interest. That is a multiclass categorisation hassle in which instances are categorised into considered one of numerous training. The case study use 3 famous classification algorithms i.e. Naive Bayes (NB), support Vector Machines (SVM), and Random forest (RF). Proposed a machine-learning classifiers to empirically validate the effectiveness of the annotated data sets. The simulation also offers word2vec word embedding's to train and test on 52 million messages. The simulation consider that those sources and the tools constructed using them will assist enhance automated natural language processing of crisis related messages and ultimately be beneficial for humanitarian groups.

Hasanuzzaman et al. (2017) present a supervised learning strategy to detect racist language on Twitter based on word embedding that incorporate demographic (Age, Gender, and Location) information. The methodology achieves reasonable classification accuracy over a gold standard dataset (F1=76.3%) and significantly improves over the classification performance of demographic-agnostic models. First step is to construct demographic word embedding's from the unlabelled data. Then, it precedes over with the context-aware high-level low-dimensional features form a succinct input representation forth specific task of racist tweet detection.

Zhao et al. (2019) presented an iForest, an interactive visualisation system that enables users interpret random forest models from diverse views. The system reveals relations between input features and output predictions, as a result allowing users to flexibly tweak feature values to monitor prediction changes. It also enables users audit the decision process of predictions to explore the underlying working mechanisms. Our assessment results show that iForest can effectively assist users in understanding random forest models and their predictions. The implementation proposes a visual analytic system aiming at interpreting random forest models and predictions. In addition to providing users with all the tree information, study summarises the decision paths or the selection paths in random forests, which eventually reflect the working mechanism of the model and reduces users 'mental burden of interpretation. To illustrate the effectiveness of system, two usage scenarios and a qualitative user study are conducted.

Dong et al. (2015) presented a target detection method, which is geared toward detecting and identifying target pixels-based totally on unique spectral signatures, and is of super hobby in Hyper-Spectral Image (HSI) processing. Target detection can be taken into consideration as essentially a binary type. Random Forests were effectively

implemented to the category of HSI statistics. But, random forests need a large amount of categorized information to achieve an amassing performance, which may be difficult to attain in target detection. The author proposes an efficient metric learning detector based on random forests, named the Random Forest Metric Learning (RFML) algorithm, which mixes semi-multiple metrics with random forests to better separate the desired targets and history.

Madisetty et al. (2018) presented an approach that combines both deep learning and traditional feature-based models using a multilayer neural network which acts as a meta-classifier. Study evaluates the method on two data sets, one data set is balanced, and another one is imbalanced. Author has proposed a neural network-based ensemble technique consisting of deep learning methods and traditional feature-based methods to detect the spam at tweet level. The simulation is achieved with multiple word embedding's using Convolutional Neural Network.

3 The language modelling of random forest

The motivation to form a language model is to illustrate the probability of a phrase string. S denotes a string of *N* phrases this is $s = s_1,\ s_2,\ s_3,...,s_N$. Then, by using the chain rule of possibility, we've got

$$P(S) = P(s_1) \times \prod_{i=2}^{N} P(s_i \mid s_1, \ldots, s_{i-1})a \tag{1}$$

To calculate the probability $P\ (s_i \mid s_1,\ s_{i-1})$, one would have to train the corpus that incorporates huge variety of words. In any real time languages ecosystem with a small vocabulary size, as we try to increase the accuracy the chances we anticipates collapse, hence history of words p_1, p_{i-1} or strings s_i may create equivalence training classes. One of the highly leveraged language models, *n*-gram language fashions, makes use of the identities of rest *m*–1 words or string as equivalence training classes. In an *n*-gram model, we've

$$P(S) = P(s_1) \times \prod_{i=2}^{N} P(s_i \mid s_{i-n+1}^{i-1} \tag{2}$$

where we have used s_{i-n+1}^{i-1} denote the word sequence $s_{i-n+1}, \ldots, s_{i-1}$.

The maximum likelihood estimate of $P(s_i \mid s_{i-n+1}^{i-1}$ is

$$P(s_i \mid s_{i-n+1}^{i-1} = \frac{c(s_{i-n+1}^{i})}{c(s_{i-n+1}^{i-1})} \tag{3}$$

where $c(s_{i-n+1}^{i}$ notes the number of occasions the string $s_{i-n+1}, \ldots, s_i$ is been found in the training data.

When we refer to n-gram language model, the collection of word $P_{i-n+1},...,P_{i-1}$ is referred to as history of predicting P_i. To perform classification of histories to equivalence class's decision tree is being used in decision tree model, with the property

that histories within similar equivalence elegance shares similar distribution over the words being predicted. The reason behind decision tree to be very attractive in language modelling is that it shares the ephemeral way while dealing with data scarceness problem. As verified from few training information, a tree keeps growing till it satisfies certain criteria.

Multiple statistics have been consumed as a function of the preventing criterion to predict the size of the selection tree. Decision tree is grown through an association of series of node splitting. The node comprises of hard and fast of histories. These set of histories are then split into subsets based on facts extracted from schooling facts. All the histories are kept in the root node initially where this node is the only leaf node of a selection tree. The node splitting is then applied to split this set of histories into two subspaces depending on information received in training sunspace. The leaf identified at each level of decision tree is selected for splitting, and then newly created nodes are marked as leaves. To increase the log probability of schooling facts, splitting criteria's are being used. The split uses most effective statistics related to the node under attention.

Allow us to anticipate that we've got a choice Tree node NDT beneath attention for splitting. H (NDT) is denoted as collection of histories that is been visible within the training information which may attain NDT node. There are *n*–1 objects within context of n-gram records. The location within the records comprise of space among the phrase in history with predicted phrase. The splits that are considered are selected functions within records.

Given a function in the records, we can outline $\alpha i(t)$ at function *i*. = to be the set of histories belonging to tree, such that all of them have phrase *t* at position *i*, it's far clean that $h(p) = Ut\ \alpha i(t)$ for every function? Within the records. For every *i*, our set of rules makes use of αi (.) = as primary factors to assemble two subsets, R_i and S_i, to form the premise of a probable cut up. Consequently, a node incorporates two questions about records: (1) is the records in R_i ? And (2) is the records in S_i ? If a record has an answer "yes" to (1), it will continue to the left toddler of the node. Further, if it has a solution "yes" to (2), it'll continue to the proper baby. If the answers to both questions are "no", the history will not continue in addition.

For straightforwardness, we omit the subscript *i* in later discussion since we always consider one position at a time. Initially, we split *h* (*p*) into two non-empty disjoint subsets, *R* and *S*, using the elements α_i (.). Let us denote the log-likelihood of the training data associated with *p* under the split as *L*(*R*). If we use the machine learning estimates for probabilities, we will have

$$L(R) = \sum_{w} \left(N(w,R) \log \frac{N(w,R)}{N(R)} + N(w,S) \log \frac{N(w,S)}{N(S)} \right) \tag{4}$$
$$N(R)\ logN(R) - N(S)\ logN(R)$$

where *N* (*w*) refers to the word count, *w* that are following all histories in (.) and *N* (.) refers to corresponding overall depend. One can note that only counts are considered in equation four, and a perfect record shape is been used to save them from computation. To identify the first-class subset *R* and *S* one would need to transfer elements within *R* and *S*

and vice versa. Suppose $\alpha(t)\varepsilon R$ is the term we need to move. The log-probability when we shift α (t) from R to S can be computed with the use of equation (4) with the following changes:

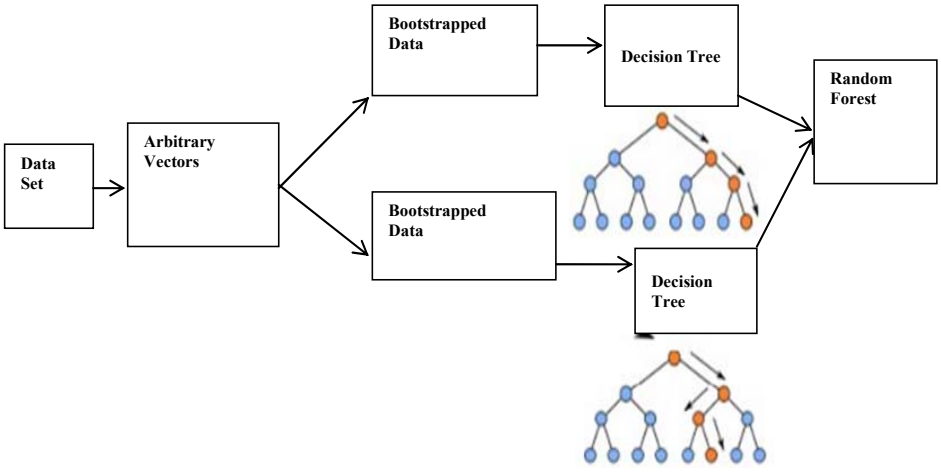
$$\begin{aligned} N(w,R) &\rightarrow N(w,R)-N(w,\alpha(t)) \\ N(w,S) &\rightarrow N(w,S)+N(w,\alpha(t)) \\ N(R) &\rightarrow N(R)-N(\alpha(t)) \\ N(S) &\rightarrow N(S)+N(\alpha(t)) \end{aligned} \tag{5}$$

The move is accepted to modify the count only if there is an increase in log likelihood which results from tentative move, else the element remain at its original location. There is a modification within the subset of R and S post every flow is done if no other flow can improve the log-likelihood of final subsets R* and S* to further assist save the total log-likelihood increase. After checking the positions within history, one can select the set with largest log-likelihood for splitting the node.

4 Random forest formations

There are two stages in the random forest algorithm, one is the creation of random forests, and the other is to make a prediction from the first stage random forest classifier. The entire process is shown below in Figure 3, and the use of the algorithm is easy to understand.

Figure 3 Random forest creation steps

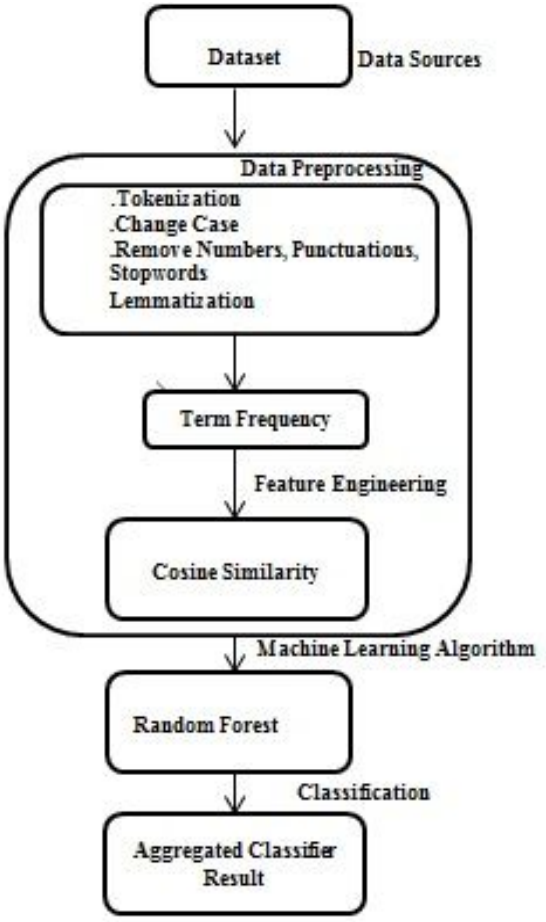


We will make the prediction in the next stage with the creation of the random forest classifier. The prediction of random forests is achieved as, Takes the test features and uses the rules of each randomly created decision tree to predict the result and store the predicted result (target). For each predicted target, calculate the votes. Consider the predicted highly voted goal as the final prediction.

5 Proposed methodology

The sentiment analysis algorithm makes use of bag of words concept to mine sentiment data from online sites such as blogging or social media networks The concept behind bag-of-word approach is to consider the count of individual words as a feature vector. The algorithms, such as Naive Bayes, Maximum Entropy, etc., often suffer from biasness towards a class and hence are appropriate to use in many cases ensemble method for performing classifications. To increase the accuracy of predictions, feature vector of similar words and context sense identities are been used.

Figure 4 Random forest-based classification in natural language processing



The sentiment analysis of social media data helps in ascertain overall mood of the public for political analysis or to understand behaviour of customers towards certain product which help businesses increase customer outreach through understanding views and opinions implied in text. With the use of NLP algorithms one can understand semantic meaning of text which helps in analysis of text information.

Specific NLP techniques used for vector formation are – stemming, removing stop words, part of speech tagging, named entity recognition. The constraint associated with bag of word algorithm is its consideration for individual words and frequencies during feature vector creation.

This problem can be overcome by using semantics during the extraction of feature vector where in semantic relationships can be explored with the help of WordNet's synsets which helps cover overall domain with original keyword.

Figure 4 illustrated the data set have been pre-processed to get the weighted term and the document matrix. We have used and anti-social data corpus that comprises of text related to racist comments, suicidal notes and confessions scripts. The data has been labelled and pre-processed by applying various text transformations.

Term Frequency and document term matrix have been used as the feature extraction of the text. Using this document term matrix Random Forest generation techniques have been applied to classify the text.

6 Experimental scenarios

Natural language processing is concerned with interaction between Natural Language text and computer. Natural Language processing is one of the major components of artificial intelligence and computational intelligence. Natural Language processing provides a seamless interaction between computers and human beings. It helps in interpretation natural text.

Computational linguistic is an emerging field and is applicable in machine translation, speech recognition, intelligent web searching, information retrieval and intelligent spelling checker. Understanding word frequency is the prime factor to evaluate after pre-processing the natural text. In natural text collocations exists. Collocations are collections of two or more tokens that tend to exist together and labelled as Unigram, Bi-gram, and Tri-Gram and so on.

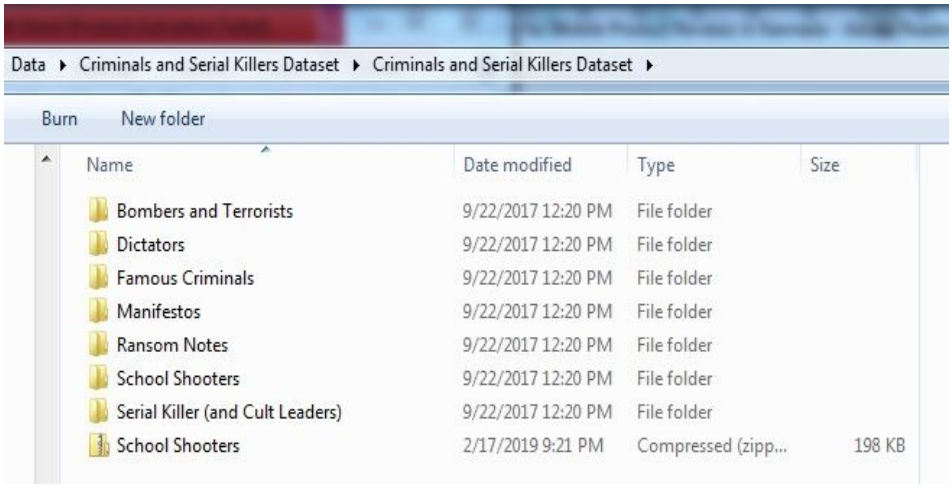
6.1 Data set and feature selection

For the dataset, weekly tweets have been collected that are prone to the cyber trolling for example racist and anti-national and secular comments. This test representation is used as the training sample, for the base classifier.

A similar entity model with 148 files serves as the training set for the random forest ensemble. It predicts the polarity of the sentiment of new document extracted from the social media. Ensemble based algorithm have been to classify the polarity of the submitted document.

The Anti-Social Behaviour corpus is a collection of aggressive, violent, and antagonistic texts and acts a training set for polarity detection. The texts were gathered from various blog posts and information-web sites which (Munezero et al., 2013) may want to conclusively identify as being Anti-Social Behaviour. In general 148 files have been diagnosed as Anti-Social Behaviour and have been shown in Figure 5. The collection is all English texts, having subjects together with: serial killer manifestos, antisocial texts, terrorism, violence-based totally texts, and suicide notes. (Chandra et al., 2017) also presented an approach to classify text that comes under anti-social category using *k*-NN classifier using which (Munezero et al., 2013) ASB corpus.

Figure 5 Data set depicting anti-social corpus



6.2 Transformation of text data

The workflow starts with a records set collection and marking of the statistics. The method starts by taking into consideration and analysing an input document for e.g. a csv file, that incorporates the evaluated texts and its related sentiment label. The data set or the corpus accommodates of the text and a sentiment column. The process starts with the reading of input file and selecting the contributing fields or columns by discarding the non-contributing columns. The contributing columns are sentiments and the labels.

6.3 Text pre-processing steps

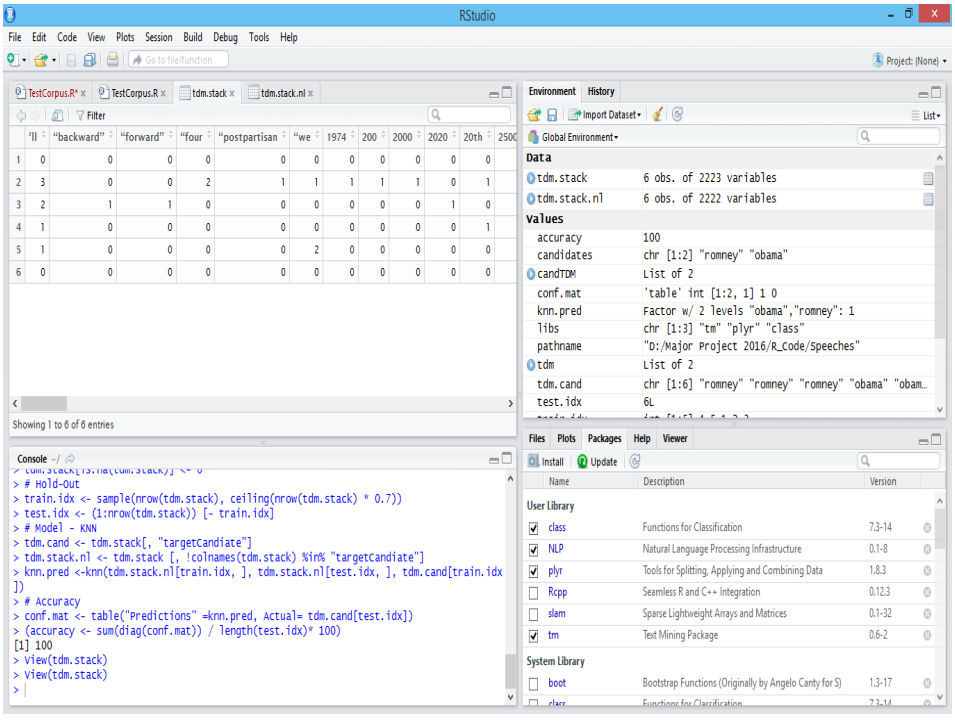
In Natural Language Processing String is the fundamental data type used to represent contents. Splitting the text into smaller parts is called tokens and the process is known as Tokenisation. One can perform tokenisation into individual sentences. This is the most crucial steps in NLP. NLTK Library to perform tokenisation there are two methods i.e. PunktWordTokenizer through that each word is kept instead of creating an entirely new token, another method is word PunctTokenizer that provides splitting by making punctuation an entirely new token. Regular expression-based tokeniser can also be used by RegexpTokenizor function. Regular expression-based tokeniser splits a string into substrings using a regular expression. Regular Expression matches either the tokens/words or the separators between tokens i.e. gaps or spaces. All the pre-processing functions have been shown in Figure 6.

With a purpose to perform processing on natural language textual content, we need to carry out normalisation that in particular includes removal of punctuation, converting the entire textual content into lowercase or uppercase, converting numbers into words, increasing abbreviations, canonicalisation of text, and so forth. The outcome as a term document matrix has been projected in Figure 7.

Figure 6 Text transformation steps

```
# remove URLs
stripURL = function(x) {
  gsub("www[^[:space:]]+|htt[^[:space:]]+", "", x)
}
corpus <- tm_map(corpus, content_transformer(stripURL))

corpus <- tm_map(corpus, removeNumbers)
corpus <- tm_map(corpus, removePunctuation)
corpus <- tm_map(corpus, stripWhitespace)

comments$comment_description <- as.character(unlist(sapply(corpus, `[`, "content")))

remove(corpus)

# remove rows with duplicate comment

library(dplyr)
comments = comments %>% distinct(., comment_description, .keep_all =  TRUE)
```

Figure 7 Term document matrix representation in *R*



Next step after normalising text is to correct or substitute tokens. This step can be achieved via replacing words using regular expression. Text contractions can be avoided via replacing text for example doesn't can be replaced by does not. The process has an important task where repeating characters will be deleted and word will be replaced but their synonyms. NLTK package has nltk.metrics which provides various evaluations of text and similarity measures. Edit Distance is the distance in terms of characters to be inserted, substituted, and deleted to get the similar text. Various operations are applied in

edit distance to get the similar text as copying letters from the first string to the second string, substituting a letter with another, deleting a letter in the first string, inserting a letter in the second string, etc.

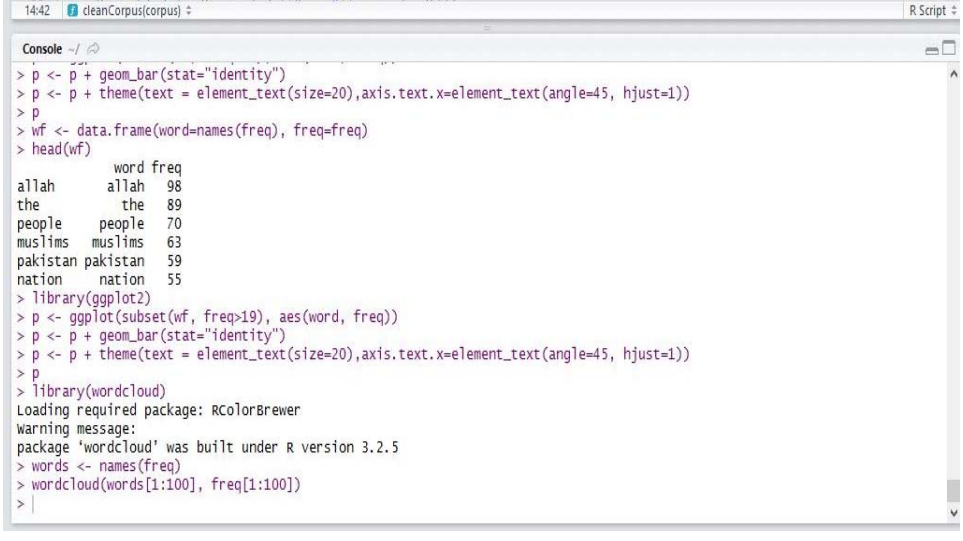
6.4 Extracting features and creating vector

We reach an imperative factor in this analysis after all these pre-processing transformations steps to prepare the text .These pre-processing steps gives the weighted terms. All these weighted terms form the basis to form term vectors. While creating the "record vector" we first identify collection or bag of words for which a table needs to be created for the usage of the "BoW creator" node. The file vector takes under consideration "bow" as input to help create the document vector.

We filter the terms with order less than 10 files from the bow vector created in previous step. We count all the distinct documents, filtering the selected terms and containing selected terms with the aid of grouping with the aid of phrases.

Using these words which have been extracted from the documents, a document vector is then created. The report vector is then represented numerically as a file and classified as tree classifier. The "record vector" could be represented as "bit vector" or "numerical vector". The rankings or frequencies can be used as numerical values and used using "TF" or "IDF" nodes thereby creating a bit vector as shown in Figure 8.

Figure 8 Term frequency representation in *R*



6.5 Random forest

At the same time as constructing the decision trees of random forest, a random pattern of some *M* training predictors are chosen as split criteria from the authentic *N* training set, all through split in a tree, it's far occurring each time.

Algorithm 1: Random forest for anti-social behaviour analysis

1: procedure RF-ASBA

2: Let N is the size of the training set

3: Let the training set is the sample of the same size N, drawn with substitute from the original data

4: If there are M input variable, a number $m<<M$ is specified such that each node, m variable are selected at random out of the M and the best split on these m is used to split the node. The value of m is held constant during the forest growing.

5: Split the node into child nodes using the best split. Tree is grown to the largest extent possible. There is no pruning.

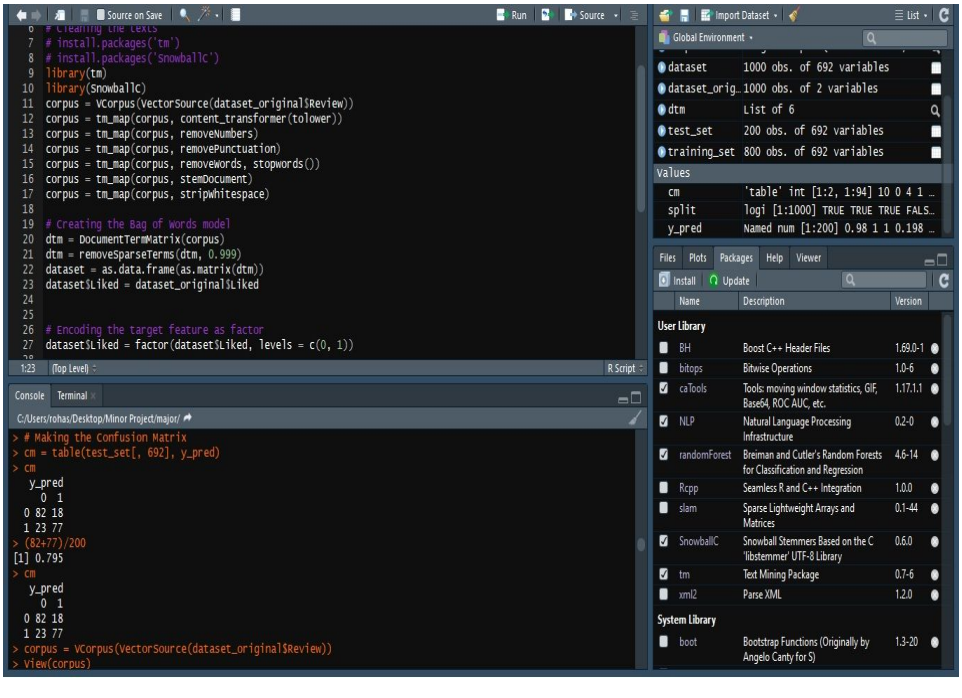
6: Predict the label/class of the test using majority vote.

7: end Procedure

End

Ensemble-based algorithm outputs an aggregated value of each tree. The decision trees are created with the use of random subset which is created from the training data having a fixed probability distribution. Deeply grown tree has low bias and high variance as depicted in the Figure 2. Owing this behaviour they can learn irregular patterns and over fit their training sets. It gives improvement over bagged tree because they decor relate the trees in the random forest.

Figure 9 Confusion matrix



6.6 Classification

To classify label of data or to categorise label of statistics there are numerous conventional data mining algorithms are available like DT, SVM and ensembles. Machine learning algorithms can be divided into category Supervised Learning Algorithm and Unsupervised Learning Algorithms. As all of us realise supervised Learning algorithms are based on a subjective variable. In this study, the subject label is Sentiment label. Sentiment label can be kept as category inside the documents. The subject or classified label is pulled out from source document and there after attached as string column with the help of category to class. Category is used like a subjective class to help with classification process. The output of classification is achieved as Confusion Matrix as shown in Figure 9. The Confusion Matrix is further realised to calculate the accuracy of the classifier.

7 Results

The output of our RF ensemble classifier is confusion matrix of the predicting classifier. The evaluation measures are calculated using the confusion matrix tabulated as Table 1. A diagonal value in the table contains the true positives for the respective label. For a specific label the remaining diagonal values are true negatives.

Table 1 Confusion matrix results

n=200	*Predicted: No*	*Predicted: YES*
Actual: NO	82	18
Actual: YES	23	77

The following is a list of computations from Figure 9. This can be calculated from a confusion matrix:

i *Accuracy*: The precision of the classifier

(TP+TN)/total = (82+77)/200 = 0.795

ii *Misclassification rate*: The frequency of misclassification of data

(FP+FN)/total = (23+18)/200 = 0.205

Equivalent to 1 – Accuracy i.e. “Error Rate”

iii *True positive rate*: When the actual answer is yes, the prediction is also yes

TP/actual yes = 77/100 = 0.77 i.e. “Recall” or “Sensitivity”

iv *False positive rate*: When the actual answer is no, the prediction is also no

FP/actual no = 18/100 = 0.18

v *Specificity*: How frequently does it predict no, when the actual answer is no

TN/actual no = 82/100 = 0.82

Equivalent to 1 – FPR

vi *Precision*: When the algorithm predicts yes, how frequently is it correct

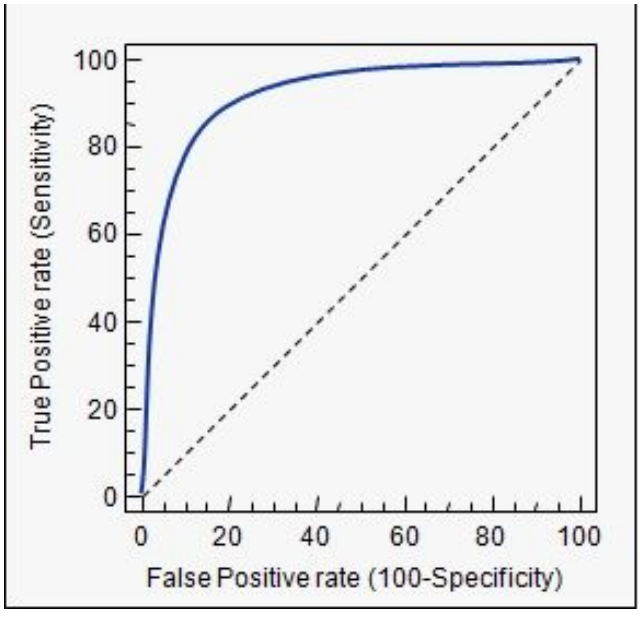
TP/predicted yes = 77/95 = 0.810

vii *Prevalence*: The yes condition is occurring how often in the sample?

Actual yes/total = 100/200 = 0.5

In this scenario we have applied decision tree as a classification algorithm on a 70% training set and 30% testing set. Decision tree accuracy is 79.5% as depicted in Figure 10.

Figure 10 ROC curve



8 Conclusions

Random forest works great with the data used to create them but they are not flexible when it comes to classifying new samples but random forests combine the simplicity of decision trees with flexibility resulting in a vast improvement in accuracy. Bootstrapped Data set, selects samples randomly from the original data set and same sample can be selected twice in bootstrapped data set. Remote sensing used in ETM devices to acquire images of the earth's surface and accuracy is higher and training time is less. It is used in multiclass object detection and it provides better detection in complicated enviornments. It is also used in a game console called Kinect, helps in tracking body movements and recreates it in the game.

Random forest really shines in the handling of missing data. It estimates missing data. Radom forest can maintain accuracy when a large proportion of data is missing. For example if sample is coming from different regions with different statically properties

and data is missing. For example in case of demographic data collection one data has missing count of children's and other has missing size of the home. With these properties two different decision trees can be built and can be used to guess which one fits better.

The approach has been devised which transforms the unlabelled raw corpus into labelled data by mapping the target word to its context word, and learns the representation of words in a classification task. Anti-social text classification at tweet level is difficult and is a challenging process. In this paper, we have proposed an ensemble technique consisting of word to vector and traditional feature-based methods to detect anti-social text at tweet level. The experiment starts with the word embedding's using word to vector method and text is classified by random forest technique. Results depicts the performance of the proposed approach is superior to the baseline classifier methods.

References

Bonissone, P., Candenas, J., Garrido, M. and Diaz, R. (2010) 'A fuzzy random forest: fundamental for design and construction', *Studies in Fuzziness and Soft Computing*, Vol. 249, pp.23–42.

Chandra, N., Khatri, S.K. and Som, S. (2017) 'Anti social comment classification based on k-NN algorithm', *Proceedings of the 6th International Conference on Reliability, Infocom Technologies and Optimization (Trends and Future Directions) (ICRITO)*, IEEE, pp.348–354.

Dong, Y., Du, B. and Zhang, L. (2015) 'Target detection based on random forest metric learning', *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*.

Hasanuzzaman, M., Dias, G. and Way, A. (2017) 'Demographic word embeddings for racism detection on twitter', *Proceedings of the the 8th International Joint Conference on Natural Language Processing*, pp.926–936.

Imran, M., Castillo, C., Diaz, F. and Vieweg, S. (2015) 'Processing social media messages in mass emergency: a survey', *ACM Computing Surveys (CSUR)*, Vol. 47, No. 4, p.67.

Imran, M., Meier, P., Castillo, C., Lesa, A. and Herranz, M.G. (2016a) 'Enabling digital health by automatic classification of short messages', *Proceedings of the 6th International Conference on Digital Health Conference*, ACM, pp.61–65.

Imran, M., Mitra, P. and Castillo, C. (2016b) *Twitter as a Lifeline: Human-annotated Twitter Corpora for NLP of Crisis-related Messages*, CoRR, abs/1605.05894.

Leistner, C., Saffari, A., Santner, J., Godec, M. and Bischof, H. (2009) 'Semi-supervised random forests', *Proceedings of the ICCV IEEE Conference*, pp.506–513.

Madisetty, S. and Desarkar, M.S. (2018) 'A neural network-based ensemble approach for spam detection in twitter', *IEEE Transactions on Computational Social Systems*, Vol. 5, No. 4, pp.973–984.

Munezero, M., Mozgovoy, M., Kakkonen, T., Klyuev, V. and Sutinen, E. (2013) 'Antisocial behavior corpus for harmful language detection', *Federate Conference in Computer Science*, Krakow, Poland.

Silva, L., Mondal, M., Correa, D., Benevenuto, F. and Weber, I. (2016) 'Analyzing the targets of hate in online social media', *Proceedings of the 10th International Conference on Web and Social Media, ICWSM 2016*.

Zhao, X., Wu, Y., Lee, D.L. and Cui, W. (2019) 'iForest: interpreting random forests via visual analytics', *IEEE Transactions on Visualisation and Computer Graphics*, IEEE, Vol. 25, No. 1, pp.407–416.