

# Random Forest-Based Sarcastic Tweet Classification Using Multiple Feature Collection



Rajeev Kumar and Jasandeep Kaur

**Abstract** Sarcasm is primary reason behind the faulty classification of the tweets. The tweets of sarcastic nature appear in the different compositions, but mainly deflect the meaning different than their actual composition. This confuses the classification models and produces false results. In the paper, the primary focus remains upon the classification of sarcastic tweets, which has been accomplished using the textual structure. This involves the expressions of speech, part of speech features, punctuations, term sentiment, affection, etc. All of the features are extracted individually from the target tweet and combined altogether to create the cumulative feature for the target tweet. The proposed model has been observed with accuracy slightly higher than 84%, which depicts the clear improvement in comparison with existing models. The random forest-based classification model has outperformed all other candidates deployed under the experiment. The random forest classifier is observed with accuracy of 84.7, which outperforms the SVM (78.6%), KNN (73.1%), and Maximum entropy (80.5%).

**Keywords** Text analytics · Supervised text classification · Sarcasm detection · Support vector machine · Punctuation features · Affection analysis

## 1 Introduction

The field of study which focuses on the interactions of human language and computers is natural language processing. NLP mainly focuses on the intersection of artificial intelligence, computer science, and computational linguistics. To examine, understand, and conclude importance and definition in a wise manner from human language, NLP uses computers. By using NLP, knowledge can be structured and

---

R. Kumar · J. Kaur (✉)

DAV Institute of Engineering and Technology, Jalandhar, Punjab, India

e-mail: [Kaurjasandeep@gmail.com](mailto:Kaurjasandeep@gmail.com)

R. Kumar

e-mail: [Rajeev.daviet@gmail.com](mailto:Rajeev.daviet@gmail.com)

© Springer Nature Singapore Pte Ltd. 2020

S. Tanwar et al. (eds.), *Multimedia Big Data Computing*

for IoT Applications, Intelligent Systems Reference Library 163,

[https://doi.org/10.1007/978-981-13-8759-3\\_5](https://doi.org/10.1007/978-981-13-8759-3_5)

analyzed to do different things like translation, automatic summarization, sentiment analysis, speech recognition, and topic segmentation. NLP is required to analyze text, allowing machine to know how human speaks. It is required for machine translation, automatic question answering, and mining. The exactness in human language is rare and this is the most difficult problem for NLP in computer science. The connection between human and machine is required to know its meaning and not by simply understanding the words. The ill-defined part of language makes NLP a critical task for computers to master and not the learning of language which is quite easy for individuals to learn. On machine learning algorithms NLP is developed. NLP can rely on machine learning than hand-coding big set of rules for automated rule learning by examining a pair of references such as down to a collection of sentences, a large corpus etcetera, and make predictions statistically. To infer, more the information is examined, more the model will be explicit.

## ***1.1 Applications of NLP***

- Machine translation

The procedure through which the conversion of source language text to the target language is done is known as Machine Translation. The pictorial representation below defines all the stages which define it that is from source text to target text [1].

- Automatic summarization

Information overload becomes a problem when humans require acquiring a specific and significant detail from a large amount of knowledge base. Therefore, this application not only understands the emotional meaning containing in the context but also conclude the definition, e.g., gathering information from Social Media.

- Sentiment analysis

To search sentiment among several posts or in the same post in which feeling is not always exhibited clearly, sentiment analysis is used. NLP applications are used by many companies such as this method to know sentiment and opinions electronically through computer to assist to know the thinking of the users related to their products or services. To exemplify, “I love the new Samsung phone” and further wrote “However, it does not sometimes operate well” in this example, an individual is mentioning about the phone along with final benchmarks of its image.

- Text classification

To get the detail which is significant or which can ease few things by permitting predefined categories to a document and fit them is feasible through this classification only.

To exemplify: Spam filtering in email.

- Question answering

For answering the human request, the term of question answering is a capable system and for its popularity, the major gratitude goes to Siri, OK Google, and Chat boxes. It provides authenticity and will go long in the upcoming time, therefore this will remain a challenging task for searching devices and will remain the crucial term of NLP research.

## 1.2 Introduction to Sentiment Analysis

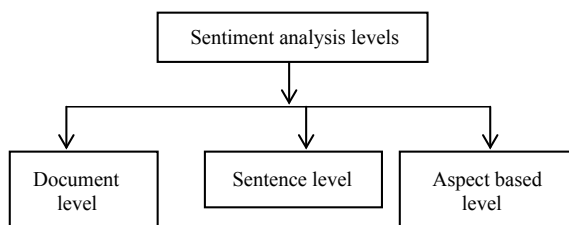
Sentiment analysis is a process to obtain valuable information or sentiment from data. It uses various techniques like text processing, text analysis, natural language, and computational linguistics to process the data. The motive is to find out polarity of a document by analysis of data inside the document. The polarity of document is according to the opinion of the document and that can be either positive or negative or can be neutral polarity. Sentiment analysis is categorized into 3 main areas which are mentioned below.

Sentiment analysis faces many challenges and one of them is sarcasm Detection. As sentiment analysis can be misguided due to the presence of words that have a strong polarity and used as sarcastically, which intended the opposite polarity. Sarcasm is a form of speech in which the speakers convey their message in an implicit way. Sometimes, the naturally uncertain nature of sarcasm makes it hard for humans to decide whether a sentence is sarcastic or not and also it conveys a negative opinion using only positive words or intensified positive words. Therefore, the detection of sarcasm is important for the development and refinement of sentiment analysis (Fig. 1).

## 1.3 Introduction to Sarcasm Detection

Sarcasm is a verbal device, with the intention of putting someone down or is an act of saying one thing while the meaning is opposite. It is mostly used on social media to make a remark that means the opposite of what they say, in order to hurt

**Fig. 1** Different sentiment analysis levels



someone's feelings. The polarity of the statement is also transformed by sarcasm into its opposite. For instance, if someone says, "You have been working hard," he said with heavy sarcasm as the person looked at the empty page.

Phases of sarcasm detection:

- **Dataset Formation:** It is the first step in which dataset can be collected from different sources, e.g., Twitter or posts from Facebook.
- **Data Preprocessing:** In this case, cleaning of data is performed such as removal of URLs, hashtags, tags in the form of @user and unnecessary symbols.
- **Sarcasm Identification:** It involves two different phases, i.e., feature selection and feature extraction. Feature extraction involves Part of speech (), Term presence, Term frequency, Inverse document frequency, Negation and opinion expressions for extracting the features. On the other hand, the lexicon method and statistical method are used in case of feature selection.

## ***1.4 Sarcasm Classification Approaches***

Sarcasm analysis can be implemented using:

- i. Machine Learning Approach
- ii. Lexicon Based Approach
- iii. Hybrid Approach

### **1.4.1 Machine Learning**

It is a field of artificial intelligence that trains the model from the current data in order to predict future outcomes, trends, and behaviors with the new test data. Machine Learning is categorized into Supervised and Unsupervised Learning.

#### **Supervised Learning**

Supervised Learning is used when there is a finite set of classes. In this method, labeled data is needed to train classifiers. In a machine learning based classifier, a training set is used as an automatic classifier to learn the different characteristics of documents, and a test set is used to validate the performance of the automatic classifier. Two steps are involved, i.e., training and testing.

## Unsupervised Learning

This method is used when it is hard to find labeled training documents. It does not depend upon prior training for mine the data. In document level, SA is based on deciding the semantic orientation (SO) of particular phrase within the document. If the average semantic orientation of these phrases is above some predefined threshold, then the document is classified as positive, otherwise it is deemed negative.

### 1.4.2 Lexicon Based Techniques

One of the unsupervised techniques of sentiment analysis is lexicon based technique. There has been a lot of work done based on lexicon. In this classification is performed by comparing the features of a given text in the document against sentiment lexicons. The sentiment values are determined prior to their use. Basically, the sentiment lexicon consists of lists of words and expressions that are used to convey people's subjective feelings and opinions. Three methods to construct sentiment lexicon are:

#### Manual Method

In this approach each opinion word, such as nice (adjective), fast (adverb), love (verb), is selected manually and the corresponding polarity is assigned. This manual approach is a little time consuming and that is why it is never used alone.

#### Dictionary Based Method

This approach has three steps. In the first step, opinion words are constructed with their sentiment orientations manually. Then, in the second step, the seed list is grown by searching for synonyms and antonyms of seed words in a dictionary that is available online such as WordNet. The search results are combined with the seed list with the same polarity as their synonyms in the list or the opposite polarity of their existing antonyms, and the seeking process is started again until no new word is found in the dictionary. In the third step, a correction process is done manually to remove any existent errors. By using machine learning techniques and using additional information in WordNet such as "hyponym, -, it is possible to generate better and richer opinion words lists".

The most important drawback of this simple approach is that it is unable to distinguish between opinion words with respect to their domains. For example, "quiet" is expressing positive sentiment in the context of a car but a negative sentiment for a speakerphone.

### Corpus-Based Method

This method is intended to solve the problem of the dictionary based approach. This method is intended to solve the problem of the dictionary based approach. It consists of two steps. The first step is constructing a seed list of opinion words which have adjective part of speech tags and their polarities. In the second step, a set of linguistic constraints is introduced to search for additional opinion words from the existing corpus as well as their sentiment orientations.

These linguistic constraints are based on the idea of “Sentiment Consistency.” According to sentiment consistency, people usually express the same opinions on both sides of conjunctions (for instance, “and”) and the opposite opinion around disjunctions (for instance, “but”). This idea helps to discover new sentiment words in a collection. For instance, in the sentence “This house is lovely and big.” If we do not have “big” in our seed list, we can conclude from “lovely” and conjunction (“and”) that “big” has the same polarity as “lovely.” Therefore, we can extend our list.

#### 1.4.3 Hybrid Based Techniques

It involves a combination of other approaches namely machine learning and lexical approaches.

## 2 Literature Survey

Tanwar et al. [2] presented a huge amount of multimedia data also defined as MMBD is produced with a rapid incline in the supplying of multimedia devices over the invent of things in “Multimedia big data computing and Internet of Things applications: A taxonomy and process model.” In the present time, there research and development activities do not consider the complexity of MMBD over IoT rather focus on scaler sensor data. This process model mainly directs a number of challenges related to research such as accessibility, scalability, QoS, and reliability requirements.

A survey is presented by Jasandeep Kaur et al. on phases of sarcasm detection and also discusses various approaches based upon the combination of multiple features for classifying the text in “Text Analytical Models for Data Collected from Micro-blogging Portal—A Review”. Moreover, various classification algorithms are deployed for various text analytics systems, which are shortlisted on the basis of the feature engineering mechanism and type of data. For the data collected from Twitter, the Random Forest, SVM, and KNN are used with the punctuation-related, syntax-based, and other features for the sarcasm detection. The Random forest classifier is found the best in comparison with other classification models, where it outperforms the other model by minimum margin of 1.6% from KNN.

Shubhodip Saha et al. proposed an approach for sarcasm detection in Twitter in which textblob is used for preprocessing which includes tokenization, part of speech tagging, parsing, and by using python programming stop words are also removed. For polarity and subjectivity of tweets, RapidMiner is used and weka tool is used for calculating the accuracy of tweets using two classifiers, i.e., Naïve Bayes and SVM. At the end, naïve bayes provides more accuracy as compared to SVM.

A survey is provided by V. Haripriya et al. on various methodologies used to sarcasm detection in Twitter social media data and also done an analysis of various classifiers such as Naïve Bayes, Lexicon Based, and Support Vector Machine. Sarcasm can be determined efficiently only if the existing approaches can deal with large data set but most of the existing approaches can deal with only small datasets. So a deep learning approach is considered as an efficient approach to detect Sarcasm in case of large datasets.

Aditya Joshi et al. described datasets, approaches, trends, and issues in sarcasm detection. Datasets are divided into three classes: short text, long text, and other datasets approach like rule-based, statistical, deep learning-based, shared tasks are discussed and issues in data, issues with features, dealing with dataset skews are the issues for sarcasm detection.

Two approaches are presented by Aditya Joshi et al. in “Expect the unexpected: Harnessing Sentence Completion for Sarcasm Detection” that use sentence completion for sarcasm detection, one is all-words approach and other is incongruous words-only approach. Two datasets are used for the evaluation (i) tweets by [3] contains 2278 tweets out of which 506 are sarcastic annotated manually (ii) discussion forum posts by [4] 752 sarcastic and 752 non-sarcastic tweets manually annotated. For similarity measures, Word2Vec and WordNet similarities are used. The evaluation is configured into overall performance and twofold cross-validation. In overall performance, when Word2Vec similarity is used for the all-words approach an F-score of 54% is obtained but when WordNet is used in Incongruous words-only approach then F-score is 80.24%. In case of two-fold cross validation, when incongruous words-only approach and WordNet similarity are used then F-score is 80.28%.

A pattern-based approach is proposed by Mondher Bouazizi et al. for sarcasm detection on Twitter. They also proposed four different sets of features, i.e., Sentiment-related features, Punctuation-related features, Syntactic and Semantic features, and Pattern-based features. In this approach, the authors proposed more efficient and reliable patterns, i.e., words are divided into two classes: “CI” and “GFI” and this approach achieved 83.1% accuracy, 91.1% precision.

Different supervised classification technique is identified by Anandkumar D. Dave et al. in “A Comprehensive Study of Classification Techniques for Sarcasm Detection on Textual Data” for sarcasm detection and also train SVM classifier for 10X validation along simple Bag-of-words as features and use TFIDF for frequency measurement of the feature. Two datasets were collected (Amazon product reviews and tweets) and preprocessing also done for the removal of noise (spelling mistakes, slang words, user-defined label, etc.) present in the dataset.

An ensemble approach is introduced by Elisabetta Fersini et al. in “Detecting Irony and Sarcasm in Microblogs: The Role of Expressive Signals and Ensemble

Classifiers” in which BMA (Bayesian Model Averaging) along with different classifiers on the basis of their marginal probability predictions and reliabilities. The two main ensemble approaches, i.e., Majority Voting and Bayesian Model Averaging are considered to detect sarcasm and irony. In order to evaluate the proposed BMA approach, Fersini et al. [5] considered baseline classifier the one with the highest accuracy and four configurations: BOW, PP, POS, PP & POS, and the experimental result shows the proposed solution outperforms the traditional classifiers for the well-known Majority Voting mechanism and in this paper sarcasm can be better characterized by PoS tags or ironic statements are captured by pragmatic particles.

Tomas Ptacek et al. represent the first attempt at sarcasm detection on two different languages, i.e., Czech and English in “Sarcasm detection on Czech and English twitter.” For this two different datasets collected 140,000 tweets in Czech and 780,000 English tweets from Twitter Search API and Java Language Detection for the evaluation and two classifiers were used, i.e., Maximum Entropy and Support Vector Machine for the classification. Tests were organized in the 5-fold cross-validation and this approach achieved F-measure of 0.947 and 0.924 on the balanced and imbalanced datasets in English. SVM achieved good results, i.e., F-measure 0.582 on the Czech dataset with the feature set upgraded with patterns.

Two additional features are proposed by Edwin Lunando et al. in “Indonesian Social Media Sentiment Analysis with Sarcasm Detection” to detect sarcasm, i.e., number of interjection words and negativity information, after a common sentiment analysis is conducted. Three different types of experiments were conducted, i.e., experiments on sentiment score, experiments on classification method and experiments for sarcasm detection. In last experiment, the additional features evaluated in the sarcasm classification accuracy which shows that the additional features are effective in sarcasm detection.

A novel bootstrapping algorithm is presented by Ellen Riloff et al. in “Sarcasm as Contrast between a Positive Sentiment and Negative Situation” this naturally learn record of positive sentiment phrases and negative situation phrases from sarcastic tweets. Two baseline systems are created and to train SVM classifiers LIBSVM library is used and 10-fold cross-validation is used to evaluate the classifiers. The SVM achieved 64% precision and 39% recall with both unigram and bigram features and the hybrid approach, applying the contrast method with only positive verb phrases raises the recall from 39 to 42%.

Bruno Ohana et al. present sentiment analysis on film reviews by using hybrid approach which involves a machine learning algorithm namely support vector machine (SVM) and a semantic oriented approach namely sentiwordnet. The features are extracted from sentiwordnet. Training of support vector machine classifier is done on these features. Film reviews are classified by support vector machine afterward. To determine the sentiment orientation of the film reviews, counting of negative and positive term scores has been done.



## 2.1 Research Gaps

1. The word compression method used in the existing model can lower the performance of the sentiment analysis by removing the necessary bias and affecting the total emotion of the text data [6].
2. The existing model offers the accuracy of the nearly 83%, which carries a room for improvement and can be improved up to the higher level. The accuracy of the system can be improved by using the various improvements in the existing model. The system accuracy may be improved by using the above steps [6].
3. The existing model requires high computational power and slower the process of sentiment analysis. The proposed model can be extended to increase the process execution speed of the process. The existing model works in the various levels and uses the multivariate feature descriptors along with the classifier, which includes the overall elapsed time of the sentiment analytical system [6].
4. Only sentiment and emotion clues, which include the polarity and emoticon features, are used to detect the sarcasm in the existing scheme. It analyzes the existence of both positive and negative sentiment-related features, which may lead to false results in many cases [7].
5. The existing approach is best acceptable for the smaller text datasets, where the results have been proved to be efficient for Twitter with 140-word tweets. As Twitter has raised the number of allowed words from 140 to 280, this scheme is no longer efficient for the Twitter data. It must be improved for the larger text databases [8].
6. The sarcasm detection is based upon the different levels of sarcastic tweet in existing scheme. Sarcasm can't be properly described with the particular pre-defined set of rules; hence this scheme can't meet such requirement. A more generalized model can be a better option for sarcasm detection [9].

## 3 Proposed Methodology

### 3.1 To Create a Dataset for Sarcasm Detection

The dataset has been collected from Twitter using the Rest API, and the tweets are captured for the different streams, which includes different keywords for sarcasm, such as #sarcasm #sarcastic, etc. The normal tweets are collected from the natural discussion threads with keywords, such as #happy, #good, etc. A total of 25000 tweets are extracted from the Twitter API used as training data, and 609 tweets for testing purpose.

## 3.2 *Implementation*

The proposed work is implemented in Anaconda framework for complete sarcasm detection model. The configuration of the system is windows 8 (64-bit operating system) having an Intel i3 processor and 3 GB RAM. A detailed explanation of the implementation is done in this section.

### 3.2.1 **Feature Comparison Model**

The new model is designed for classifying tweet data into various categories, which involves the tweet data obtained from Twitter containing several tweets including non-sarcastic and sarcastic tweets. It is basically based upon the mixture of knowledge-based sarcasm detection with feature amalgamation to explore the various aspects of the text in order to recognize the correct type of the tweet. N-gram analysis techniques are used to extract tokens from the message data. Basically, for word-level tokenization is occurred. Mainly, the tokenization process relies on simple heuristics and is distinct from the whitespace characters (such as a space, line breaks) or by punctuation characters. These whitespace and punctuation can or cannot have comprised in the developing accrued record of tokens. On the other hand, there are many cases like hyphenated words, contractions, emoticons, and larger constructs such as URIs. The sarcasm detection technique is quite based upon the tweet category database that uses the n-gram analysis for message data. This model is designed in various components and each component has its own working and design. The new model contains different modules such as tokenization, feature extraction, classification density estimator, stop word filter, etc. Each components have created the final model of proposed work based on the sarcasm detection using feature engineering (or amalgamation) using various aspects of the text data.

### 3.2.2 **Tokenization**

It is the method to extract the keyword data from the input message string. It also enables the automatic sarcasm detection algorithms to find the category of the input tweet data, which gives better results within the lower time and small dictionary as compared to the complete phase dictionary. The proposed model has been analyzed under the N-gram model, which is capable of extracting the word combinations with the higher influence rather than extracting the stag words of less influence.

#### **Algorithm 1: The tokenization method design**

1. Acquire the string from the message body
2. Split the string into the word list
3. Count the number of words in the splitted string
4. Load the STOPWORD data

5. Start the iteration for each word (index)
  - a. Check the word (index) against the STOPWORD list
  - b. If the word (index) match return true
    - i. Filter the word out of the list
  - c. Otherwise, match the word (index) with the supervised data provided for the tokenization
  - d. If the token matches the data in the supervised lists
    - i. Add to the output token list
  - e. Check the token relation with the next word against the phrase data
  - f. If relation found
    - i. Pair both of the words word (index) and word (index + 1)
  - g. Otherwise, return the singular word (index)
  - h. If it's the last word
    - i. Return the word list
  - i. Otherwise GOTO 5(a)

**Feature 1 Contrasting features:** The first feature is entirely based upon the contrasting connotations, which is the most prominent factor showing the sarcastic expressions. The use of the contrasting combinations of emotion or meaning based words and phrases are mainly used to show sarcasm in the sentences. The example of the sarcastic expressions such as “I love being robbed during holidays,” or “I enjoying being cheated by businesses” are the high sarcasm phrases, which are used to show the most common form of sarcastic sentences. From these combinations, two primary things are cleared, which involves the affection and sentiment scores. The sentiment score is calculated by using the following algorithm, which is a kind of supervised sentiment analysis method.

**Algorithm 2: Typical Sentiment analysis model**

1. Perform the data acquisition
2. Perform user list extraction over the input data acquired from the social thread
3. Perform the message level extraction from the input data
4. Apply the supervised tokenization with the localized dictionary to extract each message M out of the total messages N

$$M = f \times (N, 'extract')$$

5. Apply the STOPWORD filtering over the message data denoted by M

$$tokens = \int Remove (M == stopWord)$$

6. Apply the polarization method over the filtered message in step 5

$$polarity = \int Score (M == polarWord)$$

7. Return the polarization value to the decision maker method under the proposed sentiment analysis algorithm

$$weight = \sum polarity$$

8. Classify the message polarity according to the computed weight
  - a. If the computed weight is lesser than 0
    - i. mark the message as negative
  - b. If the weight is higher than 0
    - i. mark the message as positive
  - c. If weight equals zero
    - i. mark message as neutral

To calculate the sentiment score, the dictionary-based sentiment analysis algorithm has been used over the tweet data, where each individual tweet is analyzed under the lexical chain method. The tri-option analysis, which includes the negative, neutral, and positive sentiments in the given tweets, where the negative score range is between  $-1$  and  $-5$  usually, whereas the positive tweet is noticed between  $1$  and  $6$  on a majority. The tweets with sentiment score of zero are considered neutral. The following methods are utilized to determine the affection and sentiment weightage of the individual terms and in the cumulative form.

$$A = \{affect(w) | w \in t\} \quad (3.1)$$

$$S = \{sentiment(w) | w \in t\} \quad (3.2)$$

$$\Delta affect = \max(A) - \min(A) \quad (3.3)$$

$$\Delta sentiment = \max(S) - \min(S) \quad (3.4)$$

Here,  $t$  denotes text contained in the tweet, whereas the  $w$  represents the words in the tweets. The  $affect()$  is the function, which accepts the input of each word one by one and returns the matching affection in the form of affection weight, also known as special sentiment weight. The sentiment is computed by using the  $sentiment()$  function, which works similarly as the  $affect()$ . The difference or contrast of the affection or sentiment is denoted with the symbols of  $\Delta affect$  and  $\Delta sentiment$ . The minimum affection score is subtracted from the maximum affection and a similar step is performed for the sentiment score vector. The contrasting weight is returned to the program.

**Feature 2 Affection analyses:** The tweet data evaluation algorithm based upon the vital combination of above techniques such as sentiment analysis, tokenization, affection, etc. The new model is designed to collect data directly from online source or offline data source. The following algorithm explains the design of the affection model for the proposed model:

**Algorithm 3: Affection analysis method**

1. Acquire the dataset and chooses the raw data form of the read CSV file
2. Count number of rows in raw data,
3. Load STOPWORD list
4. Run the iteration for each message in the raw data
  - a. Extract the current message in the raw data
  - b. Filter the STOPWORDS from the input message data
  - c. Extract the tokens from the input messages data
  - d. Evaluate the affection score of the overall message
  - e. Return the message score to determine the degree of affection
  - f. Add to the detected polarity list of positive, negative or neutral
  - g. If the message is negative
    - i. Acquire the deep emotion supervised lists
    - ii. Determine the message under anger module
    - iii. Determine the message under disgust module
  - h. Return the deep sentiment results

**Feature 3 Punctuation:** This feature is the detailed feature, which counts for the various terms and their individual weights in order to understand the composition of the sentences in the given tweets. This is considered very important, as there is always a unique pattern behind each and every kind of phrase or sentence being written. The composition includes the various terms together such as special characters, punctuations, verbs, adverbs, etc. The following algorithm is used to determine this feature in an elaborative way:

**Algorithm 4: Sentence composition extraction model**

1. Acquire the tweet data obtained from the API
2. Count the rows in the tweet data matrix
3. Iterate for every row in tweet data matrix
  - a. Read the current tweet from the tweet data matrix
  - b. Convert the tweet string to lowercase
  - c. Normalize the string to make it process able through NLP processors
  - d. Replace the URL with the word “url”
  - e. Replace the string “@username” with the word “at\_user”
  - f. Remove the hashtags from the string
  - g. Remove the number values from the input string
  - h. Remove the special characters from the input string
  - i. Convert the string to Unicode string
  - j. Apply the tokenization on the string
  - k. Replace the internet slangs with the original syntactic replacements in the tokens array
  - l. Convert the tokens to the string
  - m. Reapply the tokenization on the re-prepared string
  - n. Remove the stopwords from the extracted keywords under N-gram analysis

- o. Extract the subjective words
  - p. Add the output to the processed array
- 4. Acquire the training data
- 5. Process the training data
- 6. Apply the classification and Return the classification results
- 7. Compute the classification performance
- 8. Return the performance parameters.

### ***3.3 Main News Classification Algorithm***

Automatic sarcasm detection module is majorly defined to compute the density of the keywords related to the categories defined in the training model. The sarcasm detection algorithm is used to return the sentences from the message weight in the numerical form. The supervised classifier is trained with real tweets for the extraction of ontology with the goal that it will find hidden dependencies and also use them for predictions. For classification, supervised learning model and statistical method both are represented by Bayesian classification and also expect a probabilistic model which is used to grab uncertainty about the model by certain probabilities of the outcomes. It can figure out problems of diagnostic and predictive (Fig. 2).

## **4 Results**

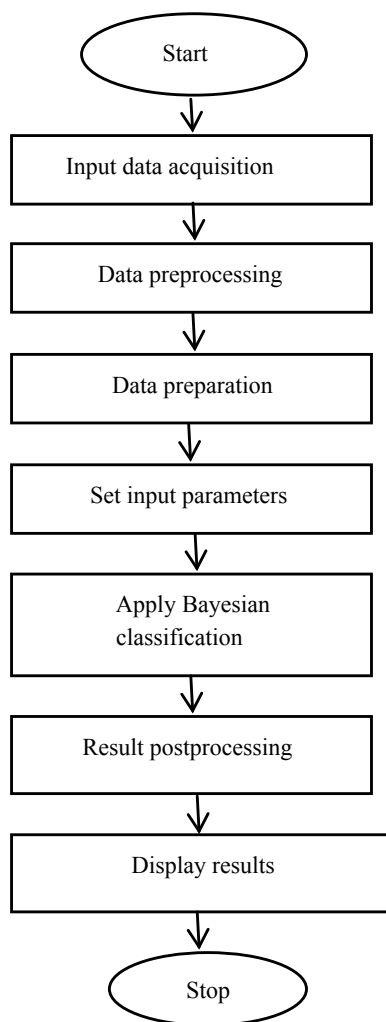
The proposed model has been designed for the sarcasm classification using the text analytical methods over the Twitter dataset. This data contains the various parameters, which includes various features such as affection, sentiment and punctuation related features, syntactic features, pattern related features, etc. In this work, the SVM, Maximum Entropy, KNN, and Random Forest classifiers are applied to the dataset in order to obtain the results.

Afterward, the data is divided into training and testing dataset, which is done using random selection by creating the random number series. The cross-validation split works on the different ratios, such as 10, 20, 30, 40, and 50% for cross-validation, which divided the testing samples accordingly into random groups of training and testing signatures under the prepared sub-datasets.

### ***4.1 Performance Parameters***

The performance evaluation of the proposed model is evaluated using the following parameters:

**Fig. 2** Generalized supervised classification model for sarcasm detection



#### 4.1.1 Accuracy

The overall accuracy is the analysis of the proposed model in the terms of overall accuracy, which is computed by dividing the total number of true cases (including true negative and true positive), by all of the cases.

$$Accuracy := \frac{True\ Positive + True\ Negative}{True\ Positive + True\ Negative + False\ Positive + False\ Negative} \quad (4.1)$$

### 4.1.2 Recall

Recall is the test of the probability of the accuracy, which indicates the performance of the proposed model in the presence of the false negative cases. The false negative cases depict the falsely detected case from the data entries. In recall, the accuracy of the proposed model has been analyzed in the presence of false negative cases:

$$Recall := \frac{True\ Positive}{True\ Positive + False\ Negative} \quad (4.2)$$

### 4.1.3 Precision

The precision depicts the accuracy of the model in the presence of false positive cases. The accuracy of the model depicts the overall impact of the false positive cases, which rejects positive cases. A positive case in our case is when the data entry contains a certain set of parameters from one of the registered category but returns the false result for such entries.

$$Precision := \frac{True\ Positive}{True\ Positive + False\ Positive} \quad (4.3)$$

### 4.1.4 F1-Measure

The F1-Measure is the cumulative parameter to assess the overall impact of the precision and recall in the case to study the overall impact of the false positive and false negative cases over the overall accuracy assessed from the preliminary statistical parameters. The F1-score value is represented in the range of 0 to 1 or 0 to 100, decided as per the maximum ranges of the precision and recall. The following equation is utilized to measure the F1-measure.

$$F1-Measure := 2 * \frac{(R * p)}{R + P} \quad (4.4)$$

where R is recall, and p or P is precision.

### 4.1.5 True Positive

The true positive reading is observed when the target tweet belongs to the sarcastic category and classification result also indicates similar after evaluating the tweet text.



4.1.6 True Negative

The true negative reading is observed when the target tweet is not sarcastic and the classification also confirms its non-sarcastic nature.

4.1.7 False Positive

The true negative reading is observed when the target tweet is not sarcastic, but the classification shows it as sarcastic tweet.

4.1.8 False Negative

The true positive reading is observed when the target tweet belongs to the sarcastic category, but the classification result indicates it as non-sarcastic after evaluating the tweet text.

4.2 Confusion Matrix

Error matrix is another term for confusion matrix and represents a unique table layout which confesses mental image of the execution of classifiers, mostly supervised learning classification. Confusion matrix is a primary kind of contingency table, having two dimensions, i.e., “actual” and “predicted” and in both dimensions sets of classes are same. Normally, positive are those which are identified and negative which are rejected. Therefore, after classification true positive are those which are correctly identified, false positive are describe incorrectly and also represent type 1 error in which no. of samples are incorrectly marked as positive. On the other hand, true negative are those which are correctly rejected, false negative represents incorrectly rejected data and also represent type 2 error in which no. of samples are marked incorrectly negative (Table 1).

Table 1 Confusion matrix

Predicted condition	True condition	
	True positive	False positive (type 1 error)
	False negative (type 2 error)	True negative

### **4.3 Four Different Classifiers for the Classification**

- SVM (Support Vector Machine)
- MaxEnt (Maximum Entropy)
- KNN (K Nearest Neighbor)
- Random Forest

#### **4.3.1 SVM**

The method which is used for classification and regression is known as SVM, in which data is examined and patterns are identified. It is also used for outlier detection. This technique uses the concept of decision planes which define boundaries for decision. Basically, it is classification method in which hyperplane is constructed in multidimensional space to perform classification tasks which classify data into different label classes. The main task of SVM is to identify the right hyperplane to segregate classes. One of the most important techniques of SVM is kernels which transform low dimensional input space into higher dimensional input space and a kernel function that converts not separable problem to separable problem. SVM performs well when margin of separation is clear and also effective in high dimensional spaces.

#### **4.3.2 Maximum Entropy**

This classifier is commonly used in speech and information retrieval problems in NLP. Moreover, MaxEnt does not make assumption in considering the features, conditionally independent of each other, unlike the naïve Bayes. It is based on the application of maximum entropy from all the models that fits the training data. To solve a big number of text classification problems like sentiment analysis, topic classification etcetera, and this classifier can be applied. In terms of estimating the parameters of model, it is required to resolve the optimization problem and due to which mainly it takes more time to train as compared to naïve Bayes. However, in terms of CPU and memory consumption, it is quite competitive as it provides tough results while computing the parameters mentioned earlier.

#### **4.3.3 KNN**

Among all machine learning algorithms, k-nearest neighbor is the smallest one with the maximum vote of its neighbors, an object is classified. It is typically small, positive integer. The assignment of the object is simply done to the category of its closest neighbor if  $k = 1$ . Choosing  $k$  to an odd number is helpful in binary classification problem as tied votes are avoided by it.

The method which is used for KNN can be applied to regression by taking the average value of KNN to be the property value for the object. All the training samples are stored in instance based or lazy learners, which are nearest neighbor classifiers and a new sample is required to be categorized without which it cannot build a classifier. Also, for making projections it can be used.

#### 4.3.4 Random Forest

This [10] was the first paper which brought the concept of ensemble of decision trees which is known random forest, which is composed by combining multiple decision trees. While dealing with the single tree classifier there may be the problem of noise or outliers which may possibly affect the result of the overall classification method, whereas random forest is a type of classifier which is very much robust to noise and outliers because of randomness it provides. Random forest classifier provides two types of randomness, first is with respect to data and second is with respect to features. Random forest classifier uses the concept of bagging and bootstrapping.

### 4.4 Result Evaluation

The results of the proposed model are analyzed for the different classifiers, which includes SVM, maximum entropy, KNN, and random forest algorithms to calculate the sarcastic sentiment in the given set of tweets. The proposed model is cross-validated using the different split ratios, which ranges between 10 and 50%. The results of the simulated are collected in the form of type 1 and 2 errors and statistical accuracy based parameters to calculate overall achievement of work. Table 2 shows statistical accuracy based parameters for the split ratio of 10% with different classification algorithms:

**Table 2** Result analysis of 10% split ratio with statistical accuracy based parameters

Classification algorithms	Precision (%)	Recall (%)	Accuracy (%)	F1-measure (%)
SVM	84.4	77.2	78.6	80.7
MaxEnt	81.0	82.0	80.5	81.5
KNN	77.9	73.1	73.1	75.4
Random forest	78.5	92.3	85.1	84.8

**Table 3** Confusion matrix for SVM classifier of 10% split ratio

	True condition	
Predicted condition	272	50
	80	206

**Table 4** Confusion matrix for maximum entropy classifier of 10% split ratio

	True condition	
Predicted condition	261	61
	57	229

**Table 5** Confusion matrix for KNN classifier of 10% split ratio

	True condition	
Predicted condition	251	71
	92	194

**Table 6** Confusion matrix for random forest classifier of 10% split ratio

	True condition	
Predicted condition	253	69
	21	265

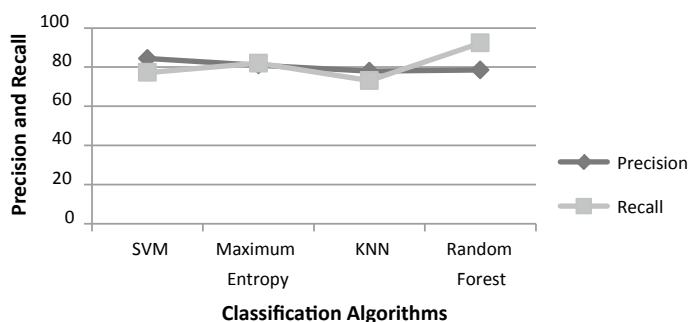
The accuracy based analysis shows the dominance of random forest classifier among all other classification options. The random forest-based model is observed with 92.3% recall, 85.2% overall accuracy and 84.9% f1-error, which are highest among the other options, whereas the 84.4% precision is observed for SVM as highest value, in comparison with random forest (78.5%), which is only exception.

Tables 3, 4, 5, and 6 show the results obtained from the first experiment, which has been conducted with 10% testing data based cross validation. Out of all of the four models, the highest true negative (265) and lowest false negative (21) cases are observed for random forest. On the other hand, the lowest false positives (50) and highest true positives (272) are observed for SVM. Table 2 produces the further accuracy based results on the testing data.

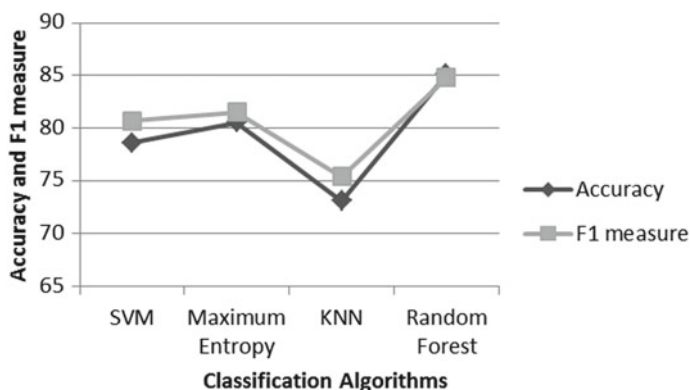
The following line graphs contain two axis, i.e., x-axis and y-axis. In x-axis, there are four different supervised algorithms that are used to classify the data and in y-axis contain a range of data in percentage for precision, recall, accuracy, and f1-measure.

The above Figs. 3 and 4 show the graphical results of the above Table 2. The dominance of random forest based classification can be clearly noticed in the above figure, which is observed higher than others in recall, overall accuracy, and f1-measure based parameters. The random forest classifier is observed best on the basis of recall, accuracy, and f1-measure parameters, whereas for the precision support vector machine classifier is observed the best among all options.

The accuracy based analysis shows the dominance of random forest, where maximum recall (93.2%), overall accuracy (84.9%), and f1-measure (85.4%) are observed, which is significantly higher than other classifiers. In contrast, the SVM classifier



**Fig. 3** Result analysis of 10% split ratio with precision and recall based parameters



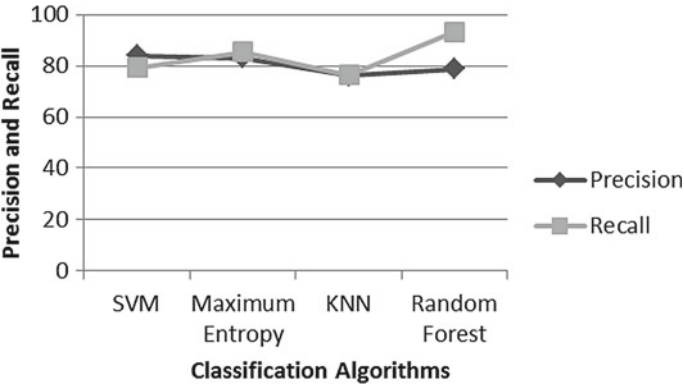
**Fig. 4** Result analysis of 10% split ratio with accuracy and F1-measure based parameters

**Table 7** Result analysis of 20% split ratio with statistical accuracy based parameters

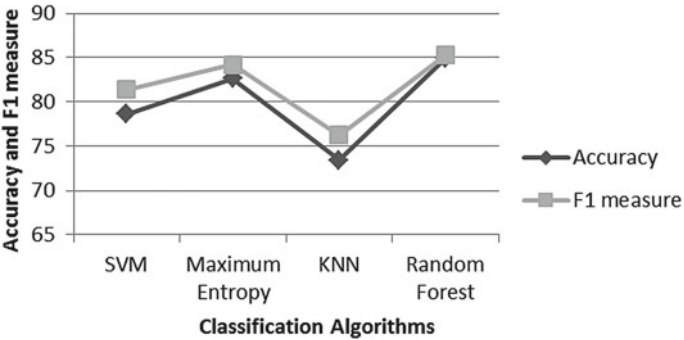
Classification algorithms	Precision (%)	Recall (%)	Accuracy (%)	F1-measure (%)
SVM	83.9	79.1	78.6	81.4
MaxEnt	83.0	85.4	82.6	84.2
KNN	76.1	76.3	73.4	76.2
Random forest	78.7	93.2	84.9	85.3

is observed with 83.9% precision is observed in comparison with random forest (78.8%).

The results of 20% testing ratio based cross-validation is evaluated with all classifiers in another experiment. The highest true negatives (497) and minimum false negatives (39) are observed for random forest among all classifiers. The lowest false positives (109) and the highest true positives (570) are observed for the SVM classifier. The overall accuracy based evaluation is shown in Table 7.



**Fig. 5** Result analysis of 20% split ratio with precision and recall based parameters



**Fig. 6** Result analysis of 20% split ratio with accuracy and F1-measure based parameters

**Table 8** Confusion matrix for SVM classifier of 20% split ratio

Predicted condition	True condition	
	570	109
	150	386

**Table 9** Confusion matrix for Maximum Entropy classifier of 20% split ratio

Predicted condition	True condition	
	564	115
	96	440

In the Figs. 5 and 6 similar to previous figure, random forest classifier is observed with highest recall (>93%), highest accuracy (approx. 84%), and highest f1-measure (>85%), which supports the selection of random forest as best classifier in comparison with other SVM, KNN, and MaxEnt for the sarcasm classification in Twitter data (Tables 8, 9, 10, 11, and 12).

**Table 10** Confusion matrix for KNN classifier of 20% split ratio

	True condition	
Predicted condition	517	162
	160	376

**Table 11** Confusion matrix for Random Forest classifier of 20% split ratio

	True condition	
Predicted condition	535	144
	39	497

**Table 12** Result analysis of 30% split ratio with statistical accuracy based parameters

Classification algorithms	Precision (%)	Recall (%)	Accuracy (%)	F1-measure (%)
SVM	83.9	78.7	78.7	81.2
MaxEnt	81.9	85.7	82.6	83.7
KNN	75	74.2	72.0	74.6
Random forest	78.6	93.4	85.2	85.3

**Table 13** Confusion matrix for SVM classifier of 30% split ratio

	True condition	
Predicted condition	839	161
	226	597

The accuracy based analysis of the classifiers on cross-validation with 30% testing data again shows the dominance of random forest classifier, with only exception of the highest precision in case of SVM. The random forest classifier based model is observed with highest recall (93.4%), overall accuracy (85.2%), and f1-measure (85.3%) in comparison with second highest recall (85.7%), accuracy (82.6%), and f1-measure (83.7%) in case of MAXENT classifier (Tables 13, 14, and 15).

**Table 14** Confusion matrix for Maximum Entropy classifier of 30% split ratio

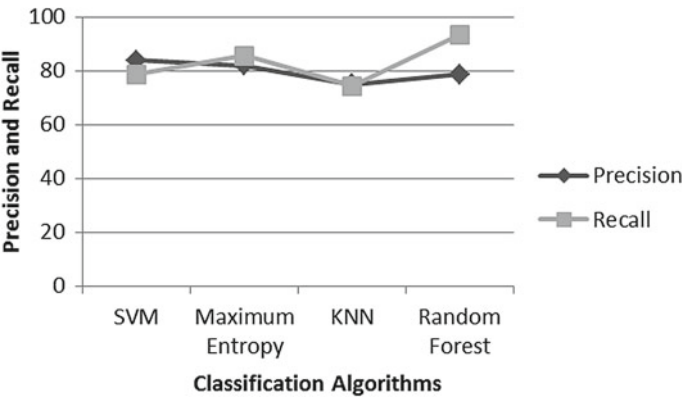
	True condition	
Predicted condition	819	181
	136	687

**Table 15** Confusion matrix for KNN classifier of 30% split ratio

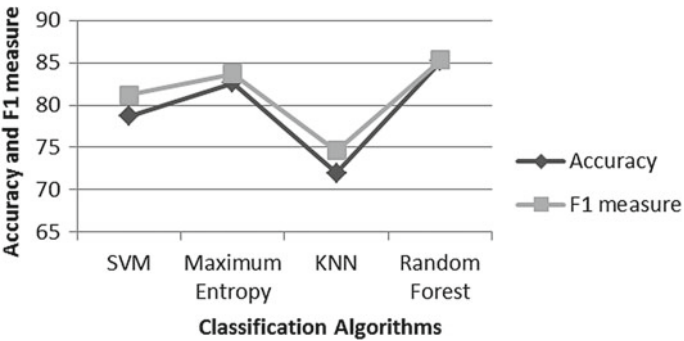
	True condition	
Predicted condition	750	250
	260	563

**Table 16** Confusion matrix for Random Forest classifier of 30% split ratio

Predicted condition	True condition	
	786	214
	55	768



**Fig. 7** Result analysis of 30% split ratio with precision and recall based parameters



**Fig. 8** Result analysis of 30% split ratio with accuracy and F1-measure based parameters

Similarly, as observed from the Table 16, the random forest classifier is also observed with highest true negatives and lowest false negatives, whereas SVM dominates the lowest false positives and highest true positives as per the result synthesis of Table 15.

In the above Figs. 7 and 8, the proposed model based upon random forest is observed with highest recall value (>93%), accuracy (>85%), and f1-measure (>85%), which proves its best performance among other classification methods, whereas the value of recall, accuracy and f1-measure are observed for KNN (74.2, 72.0, and 74.6%), MAXENT (85.7, 82.6, and 83.7%), and SVM (78.7, 78.7, and 81.2%). On the contrary, SVM has been observed with significantly higher on the



**Table 17** Result analysis of 40% split ratio with statistical accuracy based parameters

Classification algorithms	Precision (%)	Recall (%)	Accuracy (%)	F1-measure (%)
SVM	84.3	76.7	77.6	80.3
MaxEnt	83.0	84.7	82.7	83.9
KNN	75.2	72.9	71.4	74.0
Random forest	79.1	92.1	85.0	85.1

**Table 18** Confusion matrix for SVM classifier of 40% split ratio

	True condition	
Predicted condition	1109	206
	336	779

**Table 19** Confusion matrix for maximum Entropy classifier of 40% split ratio

	True condition	
Predicted condition	1092	223
	196	919

**Table 20** Confusion matrix for KNN classifier of 40% split ratio

	True condition	
Predicted condition	989	326
	367	748

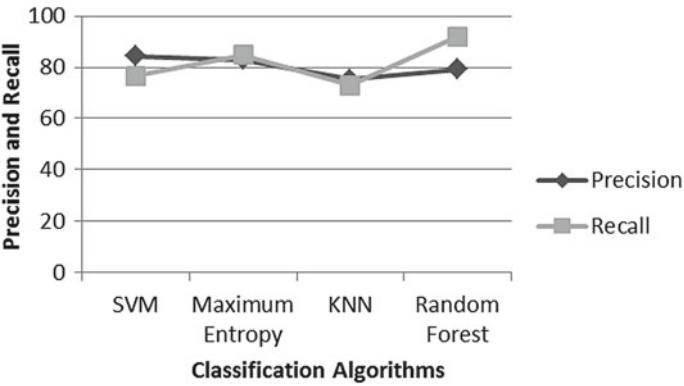
basis of precision (approx. 84%), which is outperformed by the overall accuracy in random forest (85%) in comparison with SVM's 78.7% (Table 17).

The accuracy based analysis shows the dominance of random forest, where maximum recall (92.1%), overall accuracy (85.0%), and f1-measure (85.1%) are observed, which is significantly higher than other classifiers. In contrast, the SVM classifier is observed with 85.3% precision is observed in comparison with random forest (79.1%).

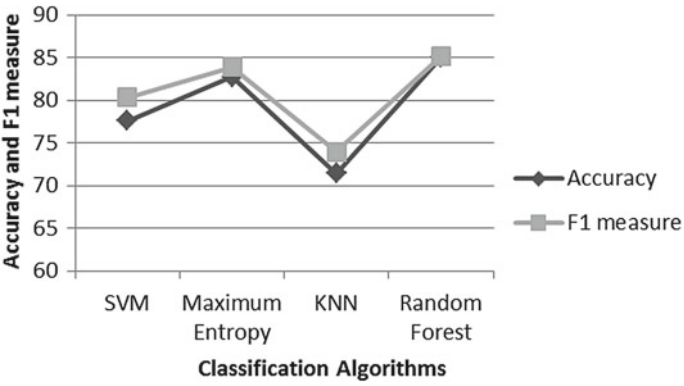
The results of the simulated are collected in the form of type 1 and 2 errors and statistical accuracy based parameters to calculate the overall achievement of work. Above tables (Tables 18, 19, 20 and 21) show the results of type 1 and type 2 errors and statistical accuracy based parameters for the split ratio of 40% with different classification algorithms. The random forest dominates for true negatives and false negatives, whereas SVM dominates for false positives and true positives, which is similar to the previous result analysis of type 1 and 2 errors.

**Table 21** Confusion matrix for random forest classifier of 40% split ratio

	True condition	
Predicted condition	1041	274
	89	1026



**Fig. 9** Result analysis of 40% split ratio with precision and recall based parameters



**Fig. 10** Result analysis of 40% split ratio with accuracy and F1-measure based parameters

**Table 22** Result analysis of 50% split ratio with statistical accuracy based parameters

Classification algorithms	Precision (%)	Recall (%)	Accuracy (%)	F1-measure (%)
SVM	85.2	75.4	77.1	80.0
MaxEnt	83.3	84.7	82.9	84.0
KNN	74.3	72.9	71.3	73.6
Random forest	80.0	91.9	85.4	85.5

The above Figs. 9 and 10 show the graphical results of the above Table 17. The dominance of random forest-based classification can be clearly noticed the above figure, which is observed higher than others in recall, overall accuracy, and f1-measure based parameters (Table 22).

**Table 23** Confusion matrix for SVM classifier of 50% split ratio

	True condition	
Predicted condition	1394	242
	453	948

**Table 24** Confusion matrix for maximum entropy classifier of 50% split ratio

	True condition	
Predicted condition	1363	273
	246	1155

**Table 25** Confusion matrix for KNN classifier of 50% split ratio

	True condition	
Predicted condition	1216	420
	451	950

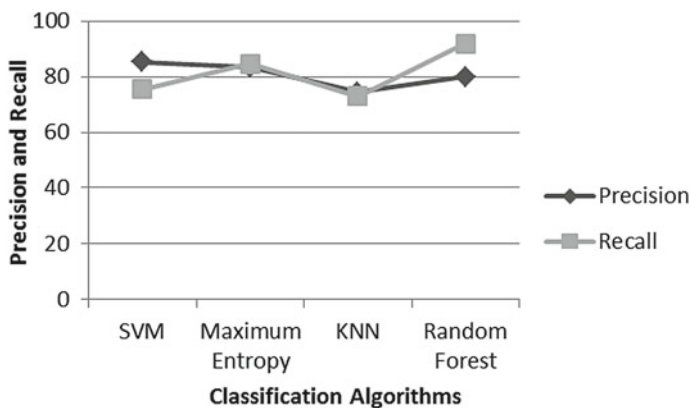
**Table 26** Confusion matrix for random forest classifier of 50% split ratio

	True condition	
Predicted condition	786	214
	55	768

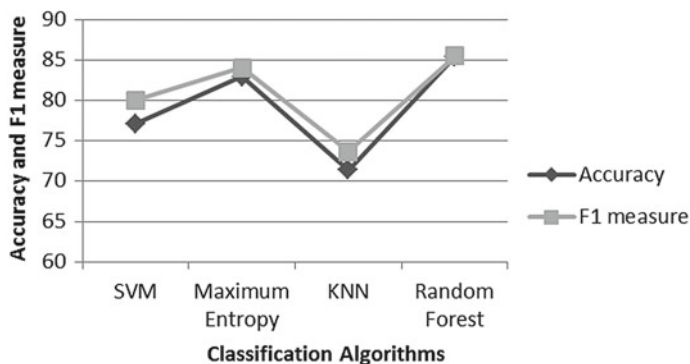
The accuracy based analysis of the classifiers on cross-validation with 50% testing data again shows the dominance of random forest classifier, with only exception of highest precision in case of support vector machine. The random forest classifier based model is observed with highest recall (91.1%), overall accuracy (85.4%), and f1-measure (85.5%) in comparison with second highest recall (84.7%), accuracy (82.9%), and f1-measure (84.0) in case of MaxEnt classifier (Tables 23, 24, and 25).

In the case of Table 26, the highest true negatives (1286) and lowest false negatives (115) are observed for random forest, whereas the highest true positives (1394) and lowest false positives (242) for support vector machine. The overall result is worst in the case of KNN and best in the case of random forest on the basis of observations in Table 26. Table 22 describes the results of 50% cross-validation obtained in the form of statistical accuracy based parameters.

In Figs. 11 and 12, the accuracy based analysis shows the dominance of random forest, where maximum recall (91.9%), overall accuracy (85.4%), and f1-measure (85.5%) are observed, which is significantly higher than other classifiers. In contrast, the SVM classifier is observed with 85.2% precision is observed in comparison with random forest (80.0%).



**Fig. 11** Result analysis of 50% split ratio with precision and recall based parameters



**Fig. 12** Result analysis of 50% split ratio with accuracy and F1-measure based parameters

## 5 Conclusion and Future Scope

The proposed model has been designed for evaluation of the tweet data on the various categories, which involves the tweet data obtained from Twitter containing the several tweets including non-sarcastic and sarcastic tweets using the unique combination of the feature descriptors, which primarily includes the contrasting sentiment this feature is entirely based upon the contrasting connotations, which is the most prominent factor showing the sarcastic expressions, the second feature is affection analysis, i.e., used for the evaluation algorithm based upon the vital combination of above techniques such as sentiment analysis, tokenization, affection, etc., and third feature is punctuation the detailed feature, which counts for the various terms and their individual weights in order to understand the composition of the sentences in the given tweets. The proposed model based upon the supervised classification based upon random forest has been observed the best among the test classification

algorithms, where the random forest is observed with (84.7%) of overall accuracy in comparison with other supervised classification models of SVM (78.6%), logistic regression (80.5%), and KNN (73.1%).

In the future, the proposed model can be further improved by using the more advanced and/or compact feature set, which can provide the more specific information to the sarcastic expressions than the approach used in this paper. The application of feature selection based upon effective algorithms like particle swarm optimization (PSO), genetic algorithm (GA), etc. will be used to attain higher exactness for sarcasm detection.

## References

1. <http://language.worldofcomputing.net/category/machine-translation>
2. S. Tanwar et al., Multimedia big data computing and internet of things applications: a taxonomy and process model. *J. Netw. Comput. Appl.* (2018)
3. E. Riloff, A. Qadir, P. Surve, L. De Silva, N. Gilbert, R. Huang, Sarcasm as contrast between a positive sentiment and negative situation, in *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP)* (2013), pp. 704–714
4. M.A. Walker, J.E.F. Tree, P. Anand, R. Abbott, J. King, A corpus for research on deliberation and debate, in *LREC* (2012), pp. 812–817
5. E. Fersini, F.A. Pozzi, E. Messina, Detecting irony and sarcasm in microblogs: the role of expressive signals and ensemble classifiers, in *Proceedings of the IEEE Conference on Data Science and Advanced Analytics, IEEE* (2015), pp. 1–8
6. M. Bouazizi, T. Ohtsuki, A pattern-based approach for sarcasm detection on Twitter. *IEEE* **4**, 5477–5488 (2016)
7. S. Poria, E. Cambria, D. Hazarika, P. Viji, A deeper look into sarcastic tweets using deep convolutional neural networks (2016)
8. A. Joshi, S. Agrawal, P. Bhattacharyya, M. Carman, Expect the unexpected: harnessing sentence completion for sarcasm detection, in *Proceedings of the International Conference of the Pacific Association for Computational Linguistics* (Springer, Singapore, 2017), pp. 275–287
9. S.K. Bharti, R. Pradhan, K.S. Babu, S.K. Jena, Sarcastic sentiment detection based on types of sarcasm occurring in twitter data. *Int. J. Sem. Web Inf. Syst. (IJSWIS)* **13**, 89–108 (2017)
10. S.K. Bharti, K.S. Babu, S.K. Jena, Parsing-based sarcasm sentiment recognition in twitter data, in *Proceedings of the International Conference on Advances in Social Networks Analysis and Mining, IEEE/ACM* (2015), pp. 1373–1380
11. J. Kaur et al., Text analytical models for data collected from micro-blogging Portal—a review. *J. Emerg. Technol. Innov. Res. (JETIR)* **5**, 81–86 (2018)
12. M. Khodak, N. Saunshi, K. Vodrahali, A large self-annotated corpus for Sarcasm (2018)
13. S. Saha, J. Yadav, P. Ranjan, Proposed approach for sarcasm detection in Twitter. *Indian J. Sci. Technol.* **10** (2017)
14. P. Deshmukh, S. Solanke, Review paper: sarcasm detection and observing user behavioral. *Int. J. Comput. Appl.* **166** (2017)
15. V. Haripriya, P.G. Patil, A survey of sarcasm detection in social media. *Int. J. Res. Appl. Sci. Eng. Technol. (IJRASET)* (2017)
16. A. Joshi, P. Bhattacharya, M.J. Carman, Automatic sarcasm detection: a survey. *ACM Comput. Surv. (CSUR)* **50** (2017)
17. A.D. Dave, N.P. Desai, A comprehensive study of classification techniques for sarcasm detection on textual data, in *Proceedings of the International Conference on Electrical, Electronics and Optimization Techniques (ICEEOT)* (2016), pp. 1985–1991

18. W. Medhat, A. Hassan, H. Korashy, Sentiment analysis algorithms and applications: a survey. *Eng. J.* **5**, 1093–1113 (2014)
19. Tomas Ptacek, Ivan Habernal and Jun Hong, “Sarcasm detection on Czech and English twitter”, *Proceedings of the 25th International Conference on Computational Linguistics: Technical Papers, COLING*, pp. 213–223, 2014
20. D. Khurana, A. Koli, K. Khatter, S. Singh, *Natural language processing: state of the art, current trends and challenges* (2017)
21. E. Lunando, A. Purwarianti, Indonesian social media sentiment analysis with sarcasm detection, in *Proceedings of the International Conference on Advanced Computer Science and Information Systems (ICACSIS), IEEE* ( 2013), pp. 195–198
22. [https://www.google.com/search?q=starting+window+of+spyder&tbm=isch&source=iu&ictx=1&fir=dlJnEECXUMNvM%253A%252CfIHkEljoqWS7zM%252C\\_&usg=\\_\\_3JDt9STOUxjMyHHYZ2UmfL697k%3D&sa=X&ved=0ahUKEwiUoo2zz\\_LaAhVKr48KHcmQDIkQ9QEIajAG#imgsrc=dlJnEEC-XUMNvM](https://www.google.com/search?q=starting+window+of+spyder&tbm=isch&source=iu&ictx=1&fir=dlJnEECXUMNvM%253A%252CfIHkEljoqWS7zM%252C_&usg=__3JDt9STOUxjMyHHYZ2UmfL697k%3D&sa=X&ved=0ahUKEwiUoo2zz_LaAhVKr48KHcmQDIkQ9QEIajAG#imgsrc=dlJnEEC-XUMNvM)
23. [https://en.wikipedia.org/wiki/Spyder\\_\(software\)](https://en.wikipedia.org/wiki/Spyder_(software))
24. <https://wiki.python.org/moin/BeginnersGuide/Overview>
25. D. Dmitry, O. Tsur, A. Rappoport, Enhanced sentiment learning using Twitter hashtags and smileys, in *Proceedings of the 23rd International Conference on Computational Linguistics: posters* (2010), pp. 241–249
26. B. Ohana, B. Tierney, Sentiment classification of reviews using SentiWordNet (2009)
27. S.B. Kotsiantis, I. Zaharakis, P. Pintelas, Supervised machine learning: a review of classification techniques, in *Emerging Artificial Intelligence Applications in Computer Engineering* (2007)
28. A. Pak, P. Paroubek, Twitter as a corpus for sentiment analysis and opinion mining. *LREC* **10**, 1320–1326 (2010)