International Journal of Computing and Digital Systems
ISSN (2210-142X)
Int. J. Com. Dig. Sys. 15, No.1 (May-24)

http://dx.doi.org/10.12785/ijcds/1501114

Analytical Comparison On Detection Of Sarcasm Using Machine Learning And Deep Learning Techniques

Ameya Parkar[1] and Rajni Bhalla[2]

[1,2]*School of Computer Application,Lovely Professional University, Punjab, India*

Received 8 Jun. 2023, Revised 20 Mar. 2024, Accepted 29 Mar. 2024, Published 1 May. 2024

Abstract: Sentiment Analysis is used in Natural Language processing to detect the opinion of the text/sentence put in by the user. A lot of challenges are faced while detecting the sentiment and one of them is the presence of sarcasm. Sarcasm is very difficult to detect and there could be ambiguity about the presence or absence of sarcasm. Various rule-based methods have been used in the past by researchers to detect sarcasm. However, the results have not been promising. The models developed using machine learning classifiers have gained popularity over the statistical and rule-based methods. Recently, deep learning techniques have been popularly used to detect the presence of sarcasm. In this paper, we have used eight machine language classifiers to detect sarcasm. Deep learning techniques have also being used along with machine learning techniques. An ensemble model has also been trained and tested on both datasets. Bidirectional Encoder Representations from Transformers technique has given the best performance among the deep learning and machine learning techniques with an accuracy score of 92.73% and f-score of 93% on the news headlines dataset and an accuracy score of 75% and f-score of 74% on the Reddit dataset.

Keywords: Sarcasm Detection, Machine Learning (ML), Ensemble model, Deep Learning (DL), Social media

1. Introduction

Social media has become the need of people in their day-to-day lives. People post their opinions, ideas, humor, etc. on social media and share it with other people online. The discussions widely range from sports, politics, movies, etc. and are openly discussed and a lot of information is available online. Websites/Applications such as Twitter allow users to express their own opinions in short text while others such as Reddit, Quora, etc. allow users to express long as well as short opinions. Companies and institutions gather the data relevant to them and try and gauge the public opinion of people about themselves, their products, etc. Sentiment analysis or Opinion mining allows the companies to judge if the people expressing their opinions are talking positively or negatively about them and their products. This helps businesses, organizations, institutes, etc. to understand the sentiment of the people which in turn can lead to promoting and launching a particular product, service, etc. or discarding or making it better. So, sentiment analysis plays an important role and businesses could be putting a lot of effort and money depending on the opinions of the people.

Some users express their sentiments using sarcasm. Sarcasm is the use of text or sentences in which the people mean the opposite of what they want to say. By using sarcasm in their opinions, the polarity of the sentence inverts from positive to negative or vice versa. If the opinions are taken in the form of video then by the gestures of the person and the facial features we can determine if the person is expressing sarcasm or not. If the opinions are taken in the form of audio then by the change of tone we can determine if the person is giving a sarcastic opinion or not. For example, in the context of a cricket game, “Way to go, player” has a very different meaning if said to a player who has got out versus a player who has hit a six. If the player had hit a six, it would be treated as a positive sentiment. However, if the player had got out, it would be a sarcastic opinion. In both cases, judging by the tone and context we can judge if the opinion is sarcastic or not.

If the opinion is only in the form of text then it is very difficult to judge if the opinion is sarcastic or not. In terms of social media such as Twitter, users use hashtags such as #sarcasm, #sarcastic, etc. to denote sarcasm. Some users put emojis such as winking face ;), smiling face, etc. to help the reader understand that the opinion is sarcastic. However, some readers might not understand the emojis and in general might not understand sarcasm. Secondly, the writer might not use the correct hashtags and the correct emojis to express their opinions.

The contributions of this paper are:

1) We collected the dataset from Kaggle. The first dataset is on news headlines and the other dataset is on Reddit posts.

E-mail address: parkar.ameya@gmail.com, rajni.b27@gmail.com

2) Preprocessing techniques like lemmatization were performed on the datasets.
3) Word embedding was done to convert text to vectors.
4) The dataset was split as 80% for training and 20% for testing.
5) ML classifiers and DL classifiers were used to train the model and test it.
6) The model was judged on different performance metrics.

The objectives of this paper are:

1) To understand the concept of sarcasm
2) To study the existing techniques used in the detection of sarcasm.
3) Introduced an ensemble model to detect the presence of sarcasm.
4) Make a comparative analysis of the existing techniques on datasets available online
5) Find the best techniques to detect the presence of sarcasm

2. Literature Review

Reference [1] gathered negative sentiment tweets on Chinese hashtags on COVID-related terms and proposed a model to detect the presence of five hyperbole features. They manually annotated the data into three classes. They used ML classifiers achieved an accuracy of 75% on the hyperbole-based model.
Reference [2] proposed a model to detect sarcasm. Initially, preprocessing steps including POS tagging were undertaken followed by TFIDF. Feature selection was done using the Chi-square technique and Information Gain. The data was then given to a SVM classifier. PSO algorithm was used to optimize the model parameters to improve the classification performance & sarcasm detection. The model performed well on the Kaggle dataset.
Reference [3] used the Arabic sarcasm Twitter dataset and used ML classifiers as well as DL classifiers with a swarm optimization algorithm to detect sarcasm. They tried to reduce the features and got an accuracy of 86.85% on the dataset.
Reference [4] used ML classifiers DL classifiers to detect sarcasm. TFIDF technique was used for the ML classifiers and Glove embeddings for the DL classifiers. LSTM technique gave an accuracy of 93.25%.
Reference [5] detected satire on short articles. They manually handcrafted each of the feature sets. They combined the sets using a deep learning architecture and gave them to machine learning classifiers. Fasttext was used to convert text to vectors. Logistic Regression was the best classifier with an f-score of 94%.
Reference [6] proposed a recurrent model to detect self-deprecation. They worked on 8 Twitter datasets. Initially, the tweets were converted into an embedding layer using GLoVe embeddings/Amazon we/Affection space. It was passed through a convolution layer and features were extracted. 2 attention layers were used after passing through the Bi-GRU layer. Adam optimizer was used followed by a sigmoid function to detect if a tweet contains self-depreciating sarcasm or not. The proposed model gave the best performance metrics across the datasets compared to the standard methods of deep learning.
Reference [7] used six machine classifiers to detect polarity and sarcasm. They compared their work with previous work which was done using deep learning Bi-LSTM technique. They worked on Arabic text and the decision tree classifier gave the best accuracy of 64.4%.
Reference [8] used DL for sarcasm detection by making a framework that had a combination of semantics, sentiments and dimensional information of users. CNN was used to extract the semantics. The bidirectional LSTM technique was used to understand the specific habits of users. They tried their framework on some datasets and got the highest F performance measure of 74.5%.
Reference [9] used t-test method for sarcasm. They used the t-test method to extract features and find the optimal features.
Reference [10] worked with ML algorithms to detect the presence of sarcasm. A few ML classifiers were used while keeping other machine learning algorithms for future use.
Reference [11] collected data from online forum posts and tried to detect the presence of sarcasm. They used a ML classifier and mentioned that semantic features could also be included for detection.
Reference [12] focused on supervised as well as unsupervised learning to detect the presence of sarcasm. Since the dataset was small, Naïve Bayes gave better performance compared to clustering methods as clustering methods require a larger dataset.
Reference [13] used ML classifiers and DL classifiers. ML classifiers were used to detect the presence or absence of a target of sarcasm while DL classifiers were used to accurately determine the target or multiple targets of sarcasm in the reviews.
Reference [14] used a graph relational structure to capture different kinds of expressions which indicated the presence of sarcasm. They tested it on different datasets concluding that external knowledge could also be used.
Reference [15] used the FastText embedding technique along with the BERT model to detect sarcasm. They achieved an accuracy score of 98 and f-score of 98.32. They used the technique on three publicly available datasets from Kaggle.
Related work is mentioned in detail in Table I and Table II.

3. Datasets

We have taken 2 datasets from the Kaggle website. The first dataset is on Headline News and contains 28619 records. It contains 2 columns, the first column for the headlines and the second indicating sarcastic or not. The second dataset is from the Reddit reviews and contains 80000 records. It contains many columns including the comments, the names of the author who wrote the comment, the date, the rating, the upvote, the down vote and the

TABLE I. Related Works

References	Dataset	Size	Language	Performance A:Accuracy P:Precision R: Recall F: F-score
1	Chinese Twitter	6600 tweets	Chinese	Hyperbole Based Sarcasm Detection model A: 75 P: 78 R: 63 F: 70
2	Kaggle	28501 posts	English	IMLB-SDC model F: 94.9
31	Social media	1956 tweets 26709 headlines	English	Twitter dataset A: 88.9 F: 81.5 Headlines dataset A: 81.4 F: 89.87
32	News Headline Dataset Kaggle	26805 headlines	English	DLE SDC model A: 94.05 P: 94.06 R: 94.01 F: 94.03
3	Semeval 2022 Twitter	3102 tweets	Arabic	ANN + Particle Swarm Optimization A: 86.85
16	Articles and writings	1500 articles	English	Algorithm and rule based with a database
19	Reddit (Movies technology) IAC (political debates)	Reddit mov: 8200 Reddit tech: 16094 IAC v1:1965 IAC v2:4646	English	Proposed model (sentiment clues incongruity) Reddit movies: A: 73.96 P: 73.98 R: 74.07 F: 73.42 Reddit technology: A: 74.53 P: 74.85 R: 74.45 F: 73.85
4	Kaggle news articles	Not mentioned	English	DT: A: 59.4 P: 85 R: 14 F: 72 KNN: A: 85.2 P: 97 R: 70 F: 81 RF: A: 87.47 P: 95 R: 81 F: 88 SVM: A: 90.28 P: 94 R: 89 F: 92 CNN: A: 79.23 P: 83 R: 79 F: 74 LSTM: A: 93.25 P: 95 R: 90 F: 93
5	News articles	32000 short news articles	English	SVM: P: 91 R: 90 F: 91 KNN: P: 86 R: 86 F: 86 LR: P: 95 R: 95 F: 94 DT: P: 90 R: 90 F: 90 DA: P: 91 R: 90 F: 91
6	Twitter datasets	D1: 151283 D2: 3892 D3: 1801 D4: 15060 D5: 52576 D6: 41703 D7: 42622	English	CAT-BiGRU model A: 93 P: 92 R: 98 F: 94
17	Twitter	1.5 million tweets	English	PID-EDSDISI method: A: 87 P: 83 R: 80 F: 82
7	Arabic texts	10000 tweets	Arabic	Decision tree A: 64.4
37	Reddit, headlines, tweets	1956 tweets 26709 headlines	English	Accuracy: Reddit: 83.92 Headlines: 90.8 Twitter: 92.8
34	Chinese dataset	4972	Chinese	BERT A;76
8	Social media datasets	13479 short ones 31822 long ones	English	BCNNSEN A:73
9	Online tweets	In hundreds	Hindi	t-test A:94
30	Online records	In hundreds	Hindi	ML classifier A:50
10	Twitter	Not mentioned	English	ML classifiers RF: 76 SVM: 74

All values in Performance column are in percentage

TABLE II. Related Works

References	Dataset	Size	Language	Performance A:Accuracy P:Precision R: Recall F: F-score
18	Chat application	Not mentioned	Indonesian	Manual Analysis methods such as pattern recognition, semantics, etc.
21	Facebook and Instagram	Short Dataset	English	Sentiment strength A:84
29	Online tweets	58609	English	ML classifiers A:83
28	Ecommerce website	Not mentioned	English	ML classifiers A:67
27	Online tweets	Not mentioned	English	ML classifiers A:96
11	Online	Forum posts	English	Classifier NB A: 78 F: 79
24	Ecommerce reviews	1254	English	k means clustering + ML classifiers A: 79
12	Twitter	Tweets	English	Supervised ML classifier A: 65
20	Online tweets	50000 tweets	English	F: 93.4
22	Online tweets	40000	English	Probabilistic CNN approach A: 97
13	Online reviews	Social media data	English	LSTM A:89
25	Online repository	Reviews	English	LSTM and ML classifier A: 95
35	Online tweets	15548	Arabic	BERT A:91
14	Public datasets	Large dataset	English	A:85
36	Online tweets and reviews	4692 lines 1262434 comments 994 tweets	English	BERT F:97
26	Twitter	20500 tweets Bengaluru traffic	English	Accuracy Naïve Bayes: 59.97 Logistic Regression: 79.93 Support Vector Machine: 80.98 Random Forest: 77.38
23	Twitter	1.45 million tweets	English	MapReduce function with Hadoop framework and corpus F: 97
33	News headlines Reddit	44263 headlines 899955 comments	English	Ensemble model News headlines A: 99 Reddit A: 82
15	News Headlines, Reddit, Twitter	26709 headlines 1 million reddit 39780 tweets	English	FastText + BERT A: 98.25 P: 92 R: 98 F: 98.32

All values in Performance column are in percentage

comment being sarcastic or not.

4. Techniques Used To Detect Sarcasm

Initial research on sarcasm used different rule-based classifiers and lexical analysis such as semantic features, sentiment features, pattern-related, syntax-related and so on. In the past few years, researchers have adopted the ML techniques and more recently DL techniques.
We discuss a few of the techniques here:

A. Sarcasm detection by Lexical Analysis

Reference [16] found out the polarity of the sentences and then proceeded to check if the sentence was sarcastic or not by using rule-based methods such as the presence of emoticons, slangs, sarcasm tags, uppercase letters, exclamation marks, etc and they applied it using algorithms. If the sentence had sarcasm then they changed the polarity of the sentence and again performed sentiment analysis to improve the accuracy.
Reference [17] proposed a model to detect emotion, sarcasm and influential users on Twitter. The model detected sarcasm by detecting hashtags in the tweet, different polarity, long sentences with contrasting polarity and comparing a positive feeling with a pessimistic situation. They used POS tagging, and bootstrap algorithm to detect sarcasm. Across all different networks of tweets, accuracy was 87Reference [18] used manual analysis and experimented on Indonesian WhatsApp to detect the presence of sarcasm by looking into semantics, patterns, etc.

B. Sarcasm detection by Word embedding

Reference [19] used the Glove model to create the word embedding layer. They considered sentiment classification along with context incongruity to detect sarcasm. They compared their proposed model with other techniques. They used Reddit datasets and IAC datasets. The model performed better compared to some other techniques and achieved an accuracy of 74.53% and 78.28% on the Reddit and IAC datasets, respectively.
Reference [20] used datasets that were manually labeled as well as labeled using supervision. They achieved an f-score of 93 on working with the distant supervision datasets and a lower score on the manually labeled data. They used CASCADE embeddings.

C. Sarcasm detection by context

Reference [21] used sentiment strength to detect sarcasm. An average of positive and negative strength was used with rules for the presence of sarcasm.
Reference [22] used the word2vec model for user embeddings and the CNN technique was used on the embeddings. A probabilistic approach was used initially as well as context was considered.
Reference [23] used a Hadoop-based framework to detect sarcasm from tweets. They used the MapReduce function and they extracted tweets having hashtags such as sarcasm. They used a corpus of universal words to detect the presence of sarcasm and the dataset was time dependent. Their technique was faster compared to other research techniques and gave an f-score of 97%.

D. Sarcasm detection by Machine Learning

Reference [24] tried different methods to select the correct features from all of the features available. ML classifiers were used to detect the presence of sarcasm. They suggested using clustering with ML classifiers.
Reference [25] used a combination of different ML classifiers and LSTM model with the embeddings of the GLOVE model. An accuracy of 95% was achieved.
Reference [26] used ML techniques to detect sarcasm. They extracted 20500 tweets using Twitter API on Bengaluru city traffic. They used Term Frequency-Inverse Document Frequency to convert text data to vectors. Support Vector classifier gave the best accuracy of 80.98% amongst the classifiers used.
Reference [27] mentioned that tweets are in general short by default and used ML classifiers on the tweets. An accuracy of 96% was achieved using the ML classifiers.
Reference [28] found the opinion of Amazon reviews. They used ML classifiers to find the presence of sarcasm. Once the presence was noted, the opinion of the review was inverted which in turn helped to increase the performance of the ML classifiers.
Reference [29] used ML classifiers on online tweets. They also used some rule-based methods like semantics, patterns, etc. to detect sarcasm.
Reference [30] worked on reviews in the Hindi language. Inverse document frequency was used to convert text to vectors and SVM was used to train and test the model.

E. Sarcasm detection by Deep learning

Reference [31] proposed an ensemble model to detect sarcasm. They used GloVe embeddings and Word2Vec embeddings to convert text into vectors and then used the LSTM technique. They worked on the Twitter dataset and Headlines dataset because Twitter has short phrases while Headlines are generally longer. The LSTM technique used dense layers and the context of the previous sentence was used to determine if the current sentence was sarcastic or not. They achieved an accuracy score of 88.9% on the Twitter dataset and 81.4Reference [32] used a deep learning model to detect sarcasm. Preprocessing was followed by Glove embeddings to convert data into feature vectors. A combination of CNN and RNN was used to detect and classify sarcasm. A hyperparameter tuning process was used to boost the detection of sarcasm.
Reference [33] proposed an ensemble model using CNN, Bi-Directional LSTM and GRU. They used social media datasets and got an accuracy of 99% and 82% on the two datasets, respectively. False negatives were not subtly caught by the model as well as sarcasm expressed politely.

F. Sarcasm detection using Transformers

Reference [34] used semantics and context information initially. They used the BERT model to detect the presence of sarcasm on a Chinese dataset.
Reference [35] used the BERT model on Arabic datasets. They worked on imbalanced datasets and suggested the detection of sarcasm in mixed languages.
Reference [36] used the BERT model to detect sarcasm. They used online datasets which were differing from one another in size. Also, they mentioned that historical data about a user need not be available for detection.
Reference [37] made a model using sentence-based embeddings and autoencoder techniques. They used BERT and USE for sentence embedding and LSTM for autoencoder. The embeddings were passed to SoftMax for final classification. They trained and tested the model on the Reddit corpus dataset, headline news and tweets with an accuracy of 83.92%, 90.8% and 92.8%, respectively.

5. Approaches Used To Detect Sarcasm In This Study

A. Naive Bayes algorithm

It is based on the Bayes theorem and is used in classification for a high-dimensional dataset. It works on the principle of probability and assumes that the features in the dataset are independent of other features in the dataset. P(A/E) = P(E/A) * P(A) / P(E) Where A and E are two events and P(A/E) is the probability of A given that event E has already occurred. We have used 7 fold cross validation in our model.

B. Support Vector Machine

It is a prediction method based on statistical learning frameworks. In SVM, the training data is mapped into points to maximize the space between two classes. It creates a decision line using which we can segregate the total space into classes and put the data points in the correct class. In the case of sarcasm, the words which contain sarcasm are on one side of the best line/hyperplane and the words which do not contain sarcasm are on the other side of the hyperplane. The equation for the hyperplane used is

$$p^T x + c = 0$$

where p represents the vector to the hyperplane, x is the input vector and c represents the distance of the hyperplane from the origin. We have used 3 fold cross-validation in our model.

C. Random Forest

It is an ML classifier that starts with quite a few decision trees and the trees contain data from various points in the dataset. The average of the prediction of the individual decision trees is taken to improve the accuracy of prediction for the dataset. It is an ensemble technique that takes the prediction from each tree and predicts the final output. In our model, we have used 100 trees as estimators. We have used 3 fold cross-validation in our model.

D. Logistic Regression

It is one of the classifier techniques which has a regression function and uses a simple sigmoid function. Independent property is assumed by this model and probability is used as a judging factor to determine the class. We have used the liblinear algorithm for optimization and 100 iterations to converge to a value. We have used 3 fold cross-validation in our model.

E. Gradient Boost classifier

Gradient Boost classifier starts with a decision stump and assigns equal weights to all data points. It increases the weights for incorrectly classified data points and decreases the weights for all correctly classified data points. It works on the principle of decision trees and is an ensemble technique. We used 50 boosting stages initially and then varied it between 50 to 200 with an initial learning rate of 0.1 and varying thereafter and a maximum depth of 5. We have used 3 fold cross-validation in our model.

F. Decision Tree

Decision tree classifier follows a tree structure where internal nodes are the features, branches are the rules and each leaf of the decision tree is the outcome. The working of the algorithm starts from the root of the tree. Each record in the dataset is checked with the value of the root attribute. This process happens for each node in the branch of the tree. It continues till all the nodes are accessed of the tree including the leaf nodes in the tree. We have used the gini criterion for the quality of the split and the best splitter value to choose the best split. We varied the maximum depth of the tree from 2 to 15. We have used 5-fold cross-validation in our model.

G. k nearest neighbor

It is one of the simplest machine learning algorithms and works on the principle that the observations can be classified and a majority vote can be used to determine in which category a particular observation will fall into. The emphasis is on the value of k, which in turn would lead to different possibilities that the observations will fall into. Euclidean distance metric is used in the k nearest neighbor classification technique and the best class is chosen depending on the closest distance the observation falls into. We have used 5 nearest neighbors to begin with. Initially, all neighbors are weighted equally and used a brute force search. We have used the Minkowski metric for standard Euclidean distance with a mean leaf size between 10 to 20. We have used 6-fold cross-validation in our model.

H. Stochastic Gradient Descent

It is an optimization algorithm used to reduce the loss of the cost function. It works on the concept of probability and rather than selecting the entire dataset for every iteration, a sample of each class is chosen so that we reach the optimal value faster. Hyperparameter tuning can be done in every iteration rather than at the end and it in turn saves time as

well as loss value is calculated at the end of each iteration. We have used a modified Huber as the loss parameter that has tolerance to outliers as well as different probability estimates. We have opted for an optimal learning rate with maximum iterations set to 100. We have used 8-fold cross-validation in our model.

I. Long short-term memory (LSTM)

It is a type of recurrent neural network. The technique works well on sequential data as it has a cell state that can store information and hence learn long-term dependencies in the data. We have used a sequential model along with the Relu activation function. The activation is based on the sigmoid function. In terms of loss function, we have used cross entropy and Adam optimizer. It eliminates the problems of vanishing gradient as well as exploding gradient by updating the weights at regular intervals.

J. Recurrent Neural Network (RNN)

It is a type of neural network where the output of the earlier state is considered and is fed to the current state. Forecasting takes place keeping in mind the state previously and this is where RNN solves the problem by having a hidden layer. It keeps in memory the order of the sequence. Generalization is possible because the weights involved are kept the same and in turn, it reduces the number of parameters. We have used a sequential model and Bidirectional LSTM is used with 64 layers. The activation function is relu and we have also used drop out. Adam optimizer is used.

K. Convolution Neural Network (CNN)

It is used in natural language processing as well as correctly classifying images. It consists of 3 layers: a convolution layer, a pooling layer and a fully connected layer. The convolution layer has a filter that checks the presence of features. After multiple iterations, a feature map is created. The values are converted to numbers and patterns are extracted from it. The pooling layer is similar to the convolution layer but it reduces the number of features and in turn, could lead to some information loss. It improves the efficiency of the CNN. The fully connected layer is where the classification happens based on the features of the previous layers. The sequential model of the keras library is used and the sigmoid function for activation. Convolution layers are used with binary cross entropy used for the loss function and Adam optimizer.

L. Global vectors for word representation (GLOVE)

It is an unsupervised deep-learning algorithm. It converts words to vectors. Training is performed on global word-to-word statistics using the co-occurrence technique. It has a huge corpus and it shows interesting relationships between different words. Similar words are clustered together and dissimilar words are in different clusters. We downloaded the pre-trained word vectors for embedding.

M. Bidirectional LSTM

It is putting two recurrent neural networks together. Reverse as well as straight information about the sentence is used in this technique. It allows to run the input in two ways, from past to future as well as future to past. Two hidden states are used to preserve the information. It adds one more LSTM layer which reverses the direction of information flow. The output from both LSTM layers is combined with operations such as addition to predict the words. 10 epochs were run to fit the data in the model.

N. Bidirectional Encoder Representations from Transformers (BERT)

It learns the context meaning between words in the text. The encoder technique is used. Context is learned using the encoder. While training, the data is picked up sentence wise and prediction for subsequent statements takes place. We have used dense layers with 128 neurons and have used Relu as an activation function. Adam optimizer is used along with the Binary cross entropy loss function. A dropout layer is used to filter out 20

O. Ensemble model

An ensemble model is used in this study to detect sarcasm. The ensemble model is used by combining the techniques of Naïve Bayes classifier, Stochastic Gradient classifier and Logistic Regression classifier. The final estimator used is Logistic Regression.

6. Methodology

A. Machine Learning techniques:

Preprocessing techniques such as tokenization, removal of stop words, removal of unwanted symbols, removal of null values, conversion of words to lowercase and normalization were done initially on both the datasets individually. A data frame was created for sarcastic words and another data frame for non-sarcastic words. We used the CountVectorizer function to convert text to vectors. Each dataset was split in the ratio of 80:20 for training and testing, respectively. The model was trained on each of the machine-learning techniques individually. The model was tested on the testing dataset and performance was noted down in the form of evaluation metrics. We have also used an ensemble model to detect sarcasm.

B. Deep Learning techniques:

Preprocessing techniques such as tokenization, removal of stop words, removal of unwanted symbols, removal of null values and normalization were done on both the datasets individually. Tokenizer was used to convert text to vectors. For recurrent neural networks along with long short-term memory technique, soft max activation function was used. Each dataset was split in the ratio of 80:20 for training and testing, respectively. The model was trained on each of the deep learning techniques individually. The model was tested on the testing dataset and performance was noted down in the form of evaluation metrics. Fig. 1 shows the flow of the methodology used in this paper

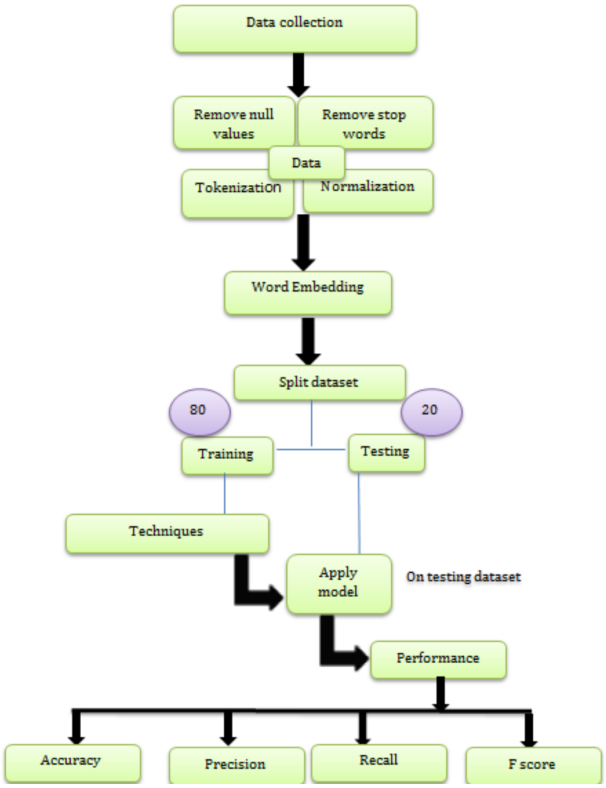

Figure 1. Work flow of the methodology used

7. Performance Measures

The performance of different models & classifiers is judged by using different performance measures. The performance measures used in this study are:

A. *Accuracy(A)*

It is one of the most fundamental measures used to judge the performance of a model which focuses on the predictions which are correctly done for both positive and negative classes.

B. *Precision(P)*

It focuses on the correctly classified positive class samples in relation to the positive class samples.

C. *Recall(R)*

It focuses on the correctly classified positive samples in relation to the rightly marked positive samples and incorrectly marked negative samples.

D. *F-score(F)*

It is one of the best performance measures that uses the average of P and R. For balanced as well as imbalanced datasets, this performance measure works well.

8. Comparative Analysis

The ML and DL algorithms are tested on two datasets. The first is the news headlines dataset and the second is

TABLE III. Machine learning techniques headlines dataset

Methodology	Accuracy	Precision	Recall	F-score
Naive Bayes	82	82	82	82
Stochastic Gradient Descent	81	81	81	81
k Nearest Neighbor	74	76	74	73
Logistic Regression	81	81	81	81
Decision Tree	59	69	59	50
Random Forest	78	78	78	77
Support Vector Machine	81	81	81	81
Gradient Boost	76	76	76	75
Ensemble Model	81	80	81	81

TABLE IV. Machine learning techniques Reddit dataset

Methodology	Accuracy	Precision	Recall	F-score
Naive Bayes	64	64	64	64
Stochastic Gradient Descent	66	66	66	66
k Nearest Neighbor	61	63	61	52
Logistic Regression	65	65	65	65
Decision Tree	58	56	58	45
Random Forest	62	62	62	60
Support Vector Machine	63	63	63	58
Gradient Boost	61	63	61	55
Ensemble Model	65	67	76	65

TABLE V. Deep learning techniques headlines dataset

Methodology	Accuracy	Precision	Recall	F-score
LSTM	73.07	71.68	64.10	67.68
LSTM + RNN	81	81	81	81
RNN	46.12	52	50	50
CNN	82	82	82	82
GLOVE	75.98	76	76	76
Bi-Directional LSTM	70.63	66.92	65.73	66.32
BERT	92.73	93	93	93

TABLE VI. Deep learning techniques Reddit dataset

Methodology	Accuracy	Precision	Recall	F-score
LSTM	63	63	63	63
LSTM + RNN	58	58	58	58
RNN	30	47	30	50
CNN	62	62	62	62
GLOVE	64.70	66.03	62.64	64.29
Bi-Directional LSTM	58.59	59.10	56.47	57.75
BERT	75	76	75	74

the text collected from the Reddit website. The results of testing the ML algorithms on the headlines dataset with the parameters of accuracy, precision, recall and f-score are mentioned in TABLE III and for the Reddit dataset in TABLE IV. The results of testing the DL algorithms on the headlines dataset with the parameters of accuracy, precision, recall and f-score are mentioned in TABLE V and for the Reddit dataset in TABLE VI.

From TABLE III, for the Headline news dataset, we can conclude that the Naïve Bayes classifier and the ensemble model give the best performance. An accuracy of 82 and an f-score of 82 is achieved among the machine learning algorithms.

From TABLE IV, for the Reddit dataset, we can conclude that the Stochastic Gradient Descent classifier and the ensemble model give the best performance. An accuracy of 66 and an f-score of 66 is achieved among the machine learning algorithms.

From TABLE V, for the Headline news dataset, we can conclude that Bidirectional Encoder Representations from Transformers gives the best performance with an accuracy of 92.73 and an f-score of 93 among the deep learning algorithms.

From TABLE VI, for the Reddit dataset, we can conclude that Bidirectional Encoder Representations from Transformers gives the best performance with an accuracy of 75 and an f-score of 74 among the deep learning algorithms.

It can be concluded that the Bidirectional Encoder Representations from Transformers technique gives the best performance.

9. Conclusion

Sentiment Analysis is used to detect the opinion of sentences and classify them as neutral, positive or negative. Sarcasm detection in text is a challenge and it is difficult to spot the presence of sarcasm. In this study, we have used eight machine language classifiers and deep learning techniques to detect sarcasm. Based on publicly accessible datasets of Headline news and Reddit, we train and test the different ML and DL techniques. Preprocessing techniques such as removal of stop words, removal of null values, tokenization and normalization are done on both datasets. Word embedding techniques have been used to convert text to vectors. We have split both datasets in the ratio of 80:20 for training and testing, respectively. Performance metrics such as accuracy, precision, recall and f-score are used. For the Headline news dataset as well as the Reddit dataset, the Bidirectional Encoder Representations from Transformers technique gives the best performance with an accuracy of 92.73% and f-score of 93% on the Headline news dataset and an accuracy of 75% and f-score of 74% on the Reddit dataset. It is to be noted that the performance is better on the Headline news dataset as the headline contains the whole context of the topic while in the Reddit dataset, there are a lot of reviews with less contextual knowledge.

Some of the research gaps that we have spotted are:

1) Detection of sarcasm when it is expressed in a polite way
2) Domain-specific sarcasm detection
3) Pre-processing is done efficiently as when we remove stopwords and other characters, some important information may also be removed

The novelty of this research is that we have analyzed different techniques used by researchers and made a fine comparison to judge the best technique/(s) amongst them to detect sarcasm. In addition, we have added an ensemble model to check if it gives a better performance than the individual techniques used.

For future work, we would go ahead with a better evolutionary method than the existing techniques.

References

[1] B. V. Govindan V, "A machine learning approach in analysing the effect of hyperboles using negative sentiment tweets for sarcasm detection," . *King Saud Univ. - Comput. Inf. Sci*, pp. 5110–5120, 2022.

[2] D. V. P. Prabhavathy, "An intelligent machine learning - based sarcasm detection and classification model on social networks," *Journal of Supercomputing*, pp. 10 575–10 593, 2022.

[3] A. Omar and A. E. Hassanien, "An optimized arabic sarcasm detection in tweets using artificial neural networks," *5th International Conference on Computing and Informatics*, pp. 251–256, 2022.

[4] R. Kumar and S. Sinha, "Comprehensive sarcasm detection using classification models and neural networks," *8th Int. Conf. Adv. Comput. Commun. Syst. ICACCS 2022*, pp. 1309—-1314, 2022.

[5] Y. W. C. N. M. N. M. S. Razali, A. Abdul Halin and S. Doraisamy, "Context-driven satire detection with deep learning," *IEEE Access*, p. 78780–78787, 2022.

[6] A. Kamal and M. Abulaish, "Cat-bigru: Convolution and attention with bi-directional gated recurrent unit for self-deprecating sarcasm detection," *Cognitive Computation*, pp. 91–109, 2021.

[7] M. A. F. M. A. Abdelaal and M. M. Arafa, "Predicting sarcasm and polarity in arabic text automatically: Supervised machine learning approach," *Journal of Theoretical and Applied Information Technology*, p. 2550–2560, 2022.

[8] M. S. P. H. K. T. Y. Du, T. Li and Z. Yang, "An effective sarcasm detection approach based on sentimental context and individual expression habits," *Cognitive Computation*, pp. 78–90, 2022.

[9] S. S. Sonawane and S. R. Kolhe, "Tcsd: Term co-occurrence based sarcasm detection from twitter trends," *Procedia Computer Science*, pp. 830–839, 2019.

[10] G. K. K. S. M. K. Sentamilselvan, P. Suresh and D. Aneri, "Detection on sarcasm using machine learning classifiers and rule based approach," *IOP Conference Series: Materials Science and Engineering*, pp. 1–8, 2021.

[11] S. M. L. M. W. R. Justo, T. Corcoran and M. I. Torres, "Extracting relevant knowledge for the detection of sarcasm and nastiness in the social web," *Knowledge-Based Systems*, pp. 124—-133, 2014.

[12] S. Mukherjee and P. K. Bala, "Sarcasm detection in microblogs using naïve bayes and fuzzy clustering," *Technology in Society*, pp. 19—27, 2017.

[13] V. L. P. Parameswaran, A. Trotman and D. Eyers, "Detecting the target of sarcasm is hard: Really??" *Information Processing and Management*, pp. 01–22, 2021.

[14] W. C. G. Li, F. Lin and B. Liu, "Affection enhanced relational graph attention network for sarcasm detection," *Applied Sciences*, pp. 01–12, 2022.

[15] P. Kumar and G. Sarin, "Welmsd – word embedding and language model based sarcasm detection," *Online Information Review*, pp. 1242–1256, 2022.

[16] A. D. S. M. A. K. B. S. Majumdar, D. Datta and A. Acharya, "Sarcasm analysis and mood retention using nlp techniques," *International Journal of Information Retrieval Research*, pp. 01–23, 2022.

[17] P. Kumaran and S. Chitrakala, "A novel mathematical modeling in shift in emotion for gauging the social influential in big data streams with hybrid sarcasm detection," *Concurrency and Computation: Practice and Experience*, pp. 01–23, 2022.

[18] E. W. R. Afiyati and A. Cherid, "Recognizing the sarcastic statement on whatsapp group with indonesian language text," *IEEE conference 2017 International Conference on Broadband Communication, Wireless Sensors and Powering (BCWSP)*, pp. 1–6, 2018.

[19] X. Z. G. L. W. Chen, F. Lin and B. Liu, "Jointly learning sentimental clues and context incongruity for sarcasm detection," *IEEE Access*, pp. 48 292—48 300, 2022.

[20] S. V. Oprea and W. Magdy, "Exploring author context for detecting intended vs perceived sarcasm," *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pp. 2854—-2859, 2020.

[21] P. Katyayan and N. Joshi, "Sarcasm detection algorithms based on sentiment strength," *Intelligent Data Analysis: From Data Gathering to Data Comprehension*, pp. 289–306, 2020.

[22] K. Sundararajan and A. Palanisamy, "Probabilistic model based context augmented deep learning approach for sarcasm detection in social media," *International Journal of Advanced Science and Technology*, pp. 8461–8479, 2020.

[23] R. K. P. K. S. B. S. K. Bharti, B. Vachha and S. K. Jena, "Sarcastic sentiment detection in tweets streamed in real time: a big data approach," *Digital Communications and Networks*, pp. 108–121, 2016.

[24] H. M. K. Kumar and B. S. Harish, "Sarcasm classification: A novel approach by using content based feature selection method," *8th International Conference on Advances in Computing and Communication (ICACC-2018)*, pp. 378–386, 2018.

[25] B. S. A. A. M. A. Barhoom and S. S. Abu-naser, "Sarcasm detection in headline news using machine and deep learning algorithms," *International Journal of Engineering and Information Systems (IJEAIS)*, pp. 66–73, 2022.

[26] M. H. A. Syrien and A. Syrien, "Sentiment polarity through sarcasm identification and detection in tweets," *Indian Journal of Natural Sciences*, pp. 40 794–40 801, 2022.

[27] H. R. S. M. U. A. P. R. A. Ashwitha, G. Shruthi and T. C. Manjunath, "Sarcasm detection in natural language processing," *Materials Today: Proceedings*, pp. 3324–3331, 2020.

[28] M. V. Rao and C. Sindhu, "Detection of sarcasm on amazon product reviews using machine learning algorithms under sentiment analysis," *2021 Sixth International Conference on Wireless Communications, Signal Processing and Networking (WiSPNET)*, pp. 196–199, 2021.

[29] M. Bouazizi and T. Otsuki, "A pattern-based approach for sarcasm detection on twitter," *IEEE Access*, pp. 5477–5488, 2016.

[30] A. D. Dave and N. P. Desai, "A comprehensive study of classification techniques for sarcasm detection on textual data," *International Conference on Electrical, Electronics, and Optimization Techniques (ICEEOT)*, pp. 1985—-1991, 2016.

[31] B. S. D. K. Sharma and A. Garg, "An ensemble model for detecting sarcasm on social media," *International Conference on Computing for Sustainable Global Development*, pp. 743—-748, 2022.

[32] K. Kavitha and S. Chittieni, "An intelligent metaheuristic optimization with deep convolutional recurrent neural network enabled sarcasm detection and classification model," *International Journal of Advanced Computer Science and Applications*, pp. 304–314, 2022.

[33] A. N. S. S. P. Goel, R. Jain and M. Srivastava, "Sarcasm detection using deep learning and ensemble learning," *Multimedia Tools and Applications*, pp. 01–24, 2022.

[34] Z. Wen, "Sememe knowledge and auxiliary information enhanced approach for sarcasm detection," *Information Processing and Management*, pp. 01–12, 2022.

[35] A. Y. Muaad, "Artificial intelligence-based approach for misogyny

and sarcasm detection from arabic texts," *Computational Intelligence and Neuroscience*, pp. 01–09, 2022.

[36] E. Savini and C. Caragea, "Intermediate-task transfer learning with bert for sarcasm detection," *Mathematics*, pp. 01–14, 2022.

[37] S. A. H. K. D. K. Sharma, B. Singh and R. Sharma, "Sarcasm detection over social media platforms using hybrid auto-encoder-based model," *Electronics*, pp. 01–26, 2022.

Ameya Parkar Research Scholar, School of Computer Application, Lovely Professional University. Area of research is Natural Language Processing.

Dr. Rajni Bhalla Associate Professor, School of Computer Application, Lovely Professional University. Area of research is Data Mining.