DOI: 10.21917/ijsc.2023.0446

AN INTEGRATED DEEP LEARNING APPROACH FOR SENTIMENT ANALYSIS ON TWITTER DATA

N.S. Prabakaran[1] and S. Karthik[2]
[1]*TATA Consultancy Services, Bengaluru, India*
[2]*Department of Computer Science and Engineering, SNS College of Technology, India*

Abstract
Analysis of Sentiment (SA) is a computational technique that seeks to extract subjective evaluations, attitudes, and emotional states from online platforms, specifically social media sites like Twitter. The subject has gained significant traction within the research community. The predominant emphasis of traditional sentiment analysis lies in the analysis of textual data. Twitter is widely recognized as a prominent online social networking platform that facilitates microblogging, wherein users share updates pertaining to various subjects through concise messages known as tweets. Twitter is a widely utilized platform that enables individuals to articulate their perspectives and emotions through the medium of tweets. Sentiment analysis refers to the computational methods of classification of given data in text format, categorizing it as either positive, negative, or neutral. The main goal of this study is to use deep learning methods for SA. The purpose is to guess sentiment and then evaluate the results based on accuracy, recall, and f-score. In this paper, a hybrid approach combining Particle Swarm Optimization (PSO) and Genetic Algorithm (GA) techniques, specifically referred to as PSOGA, is proposed to optimize features for a modified neural network (MNN). The final step is to use the K-fold cross-validation method to assess the results. The dataset was obtained through the utilization of the Ruby Twitter API. The ultimate outcome is juxtaposed with the preceding Cuckoo Search (CS) algorithm that had been terminated.

Keywords:
Tweeter, Deep Learning, K-Fold Cross Validation, HDFS, Modified Neural Network

1. INTRODUCTION

In the last few decades, social media has become the primary vehicle for the worldwide distribution of news and information. Social media plays a significant role in enabling individuals to express their opinions on various current or real-world events. Information can be disseminated among individuals via various platforms, such as Facebook, Twitter, and other commonly employed social media applications. These platforms allow users to not only share information but also express their sentiments through the utilization of tweets or textual emotions [1]. Tweets may discuss a wide range of products, services, and circumstances, allowing users to share their thoughts and feelings about them, encompassing both positive and negative sentiments. Sentiment analysis enables businesses to identify customer perspectives by analyzing online reviews and feedback, thereby allowing them to discern customer sentiment towards products and their respective features [2].

Sentiment analysis (SA) provides users with an assessment of the satisfaction level associated with product information prior to making a purchase. Marketers and firms utilize analysis data to gain insights into their products or services, enabling them to tailor their offerings to meet the specific needs and preferences of users. Textual retrieval of data techniques primarily center on the processing, searching, and analysis of factual data. Facts possess an inherent objective nature; however, there exist additional textual elements that convey subjective attributes. The primary focus of SA is the examination of thoughts, feelings, appraisals, beliefs, and emotions, as they constitute the central elements of the analyzed content. The proliferation of online sources such as blogs and social networks has led to a substantial increase in available information, thereby presenting numerous challenging opportunities for the development of new applications. One potential approach to predicting recommendations generated by a recommendation system involves incorporating sentiment analysis to consider positive or negative opinions regarding the items in question.

The primary objective of this analysis is to distinguish text sentiment division, treating it as an issue of classification. The primary source of SA is derived from social media platforms [3]. Numerous social networking platforms consistently generate intricate data pertaining to sentiments. Numerous individuals express their viewpoints through social media platforms, such as tweets and microblogs, in order to delve into evaluations of online products [4]. The aforementioned data necessitates analysis in order to discern the polarity of various elements such as movie reviews, products, and political issues within a country. In recent times, numerous researchers have put forth deep learning approaches for addressing various tasks in natural language processing (NLP).

2. LITERATURE REVIEW

In their study, [5] introduced a novel approach known as frequentiment for automated sentiment analysis of user opinions derived from a dataset of Amazon reviews. Building upon the methodology employed in this study, the researchers proceeded to construct a lexicon of words by employing probabilistic frequency calculations to determine the occurrence of words within the text. Furthermore, they assessed the impact of class by segregating the attributes identified within the input text data. The researchers conducted an analysis of the results generated by lexicons based on unigrams, bigrams, and trigrams. They employed both unsupervised and supervised machine learning approaches to evaluate the outcomes. Further, they evaluated the efficacy of 37 distinct machine learning techniques by comparing their results on the Amazon dataset. The researchers assert that within the existing body of literature, this particular domain of sentiment analysis is regarded as one of the most comprehensive.

In their study, [6] demonstrated the process of developing a application for Windows that facilitates real-time data analysis on twitter. WordNet, the SMS dictionary, and the Stanford dictionary are only some of the natural language processing resources that this software draws upon in its preparation of textual data.

In this study, [7] focuses on the issue of sentiment classification using a dataset from Twitter. The author used a number of machine and deep learning approaches for categorization and emotional analysis. When compared to other methods, the accuracy achieved by using a Convolutional Neural Network (CNN) in combination with the long-short-term memory (LSTM) approach has been reported to be significantly higher.

In [8] conducted a comprehensive survey on SA utilizing a Twitter database. In this paper, the author presents a comparative examination of various methods employed in opinion mining, specifically focusing on ML and lexicon-based approaches. Additionally, the study includes an examination of evaluation metrics utilized in this field. This paper also addresses the diverse challenges and applications associated with sentiment analysis. The author concludes that high-quality data directly impacts the reliability of findings, emphasizing the importance of cleaner data. The utilization of a bigram model yields superior accuracy in sentiment analysis when compared to alternative models. The survey findings indicate that the integration of machine learning techniques with the opinion lexicon method can enhance the precision of sentiment classification and enable adaptability across diverse domains and languages.

In their study, [9] aimed to implement sentiment analysis through the utilization of machine learning methodologies. The primary goal of this research was to predict sentiment and subsequently analyze the obtained results. This study employs Support Vector Machine (SVM), Decision Tree (DT), Navie Bayes (NB), and Random Forest (RF) techniques for the analysis. The DT method demonstrates superior accuracy compared to other methods.

In their study, [10] introduced a deep learning approach to conduct sentiment analysis on Twitter data. The proposed approach utilizes a CNNLSTM architecture in deep learning, incorporating pre-trained embeddings. The objectives of this approach include the automatic extraction of characteristics for use in sentiment analysis and the categorization of assessments and views, which are categorized into two different classes, namely positive or negative. This work utilizes benchmark data. An 81.20% accuracy rate has been achieved by comparing the performance of the CNN-LSTM-based deep learning methodology to that of baseline machine learning approaches.

Zhang and Zheng [11] conducted a comparative study on the topic of text sentiment analysis. This Chinese text is being considered for SA, which we will think of as a problem of binary classification to establish the text's polarity, specifically its positive or negative emotional tendencies. The researchers conducted various preprocessing tasks on the data, including cleaning the data, word segmentation, stop word removal, selection of features, and classification. The TF-IDF (Term Frequency-Inverse Document Frequency) approach was used to calculate the importance of each attribute. The authors employ a SVM model for the purpose of text representation and subsequently utilize both SVM and Extreme Learning Machine (ELM) algorithms to analyze the emotions conveyed within the text.

In their study, Yadav and Shitole [12], put forth a sentiment analysis approach for Twitter data, employing three machine learning algorithms: Logistic Regression (LR), Linear Support Vector Classification (LSVC), and Multinomial Naive Bayes (NB). In this research study, the author has determined that the utilization of sentiment features, as opposed to conventional text classification methods, leads to a higher level of accuracy. The Linear Support Vector Classification (SVC) model demonstrates superior accuracy, achieving a rate of 83.71%, surpassing the accuracy of the other two models.

3. PROPOSED SYSTEM

In our proposed study, the processing of natural language in an efficient manner is conducted through a series of stages. Initially, the input data from Twitter is subjected to preprocessing techniques such as tokenization, lemmatization, misspelled correction, and removal of repeated characters. The purpose of these procedures is to reformat data that is currently in an unstructured state. After the words have been preprocessed, they are next arranged systematically using HDFS (Hadoop Distributed File System). This facilitates the execution of MapReduce operations to eliminate duplicate words and establish the desired structured format. Following this step, the features are extracted. Thirdly, the features are extracted from the resulting word after the HDFS process. The features encompassed in this context include Unigram, Bigram, Trigram, N-gram, and Connotative. In the fourth step, the task is to assign a ranking to all of the featured words. Subsequently, the Deep Learning Modified Neural Network employs the extracted features ranking value to categorize the data into either of these three classes: positive, negative, or neutral categories for the purpose of sentiment analysis. The weight parameters of the Deep Learning Modified Neural Network are optimized through the utilization of Particle Swarm Optimization and Genetic algorithms (PSOGA). The training error in Deep learning-modified neural networks can be reduced with the addition of hidden layers. It appears from the simulation results that the suggested solution outperforms comparable approaches. Ultimately, the outcome was evaluated through the utilization of the K-fold cross-validation technique. The Fig.1 depicts the proposed system's process.

3.1 DATA COLLECTION

This paper presents a methodology for collecting labeled data using the Ruby Twitter API. Additionally, an organized set of pre-processing steps is proposed to enhance the manageability of tweets for natural language processing techniques.

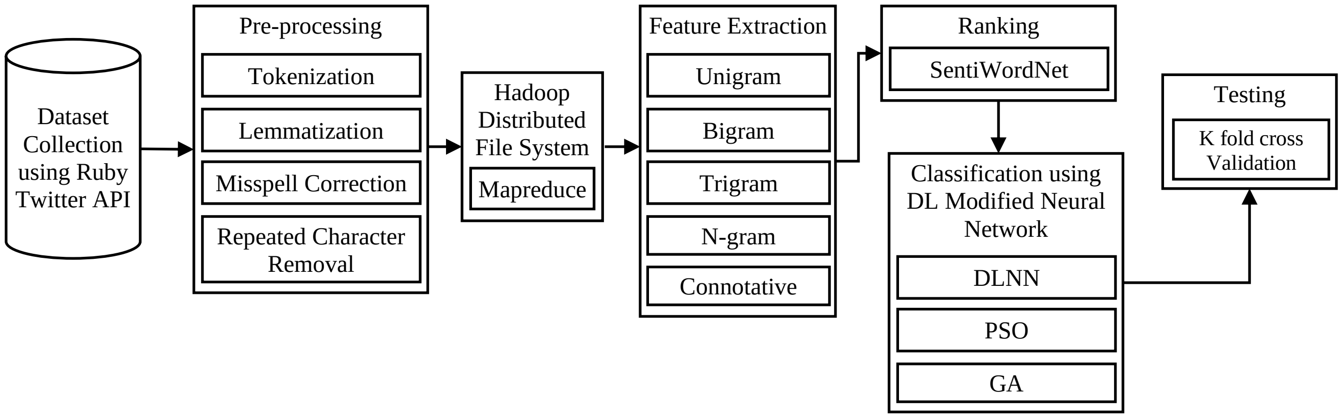

Fig.1. System Architecture for proposed work

3.2 PRE-PROCESSING

The initial step involved the retrieval of data from the database. The input data undergoes preprocessing in order to transform unstructured information into a structured format. This is achieved through a process called tokenization, where the data points are separated into individual words. Additionally, any irrelevant words are eliminated from the dataset.

3.2.1 Tokenization:

Tokenization is a crucial preliminary process in SA for Twitter data. The process entails decomposing a textual document, such as a tweet, into discrete entities referred to as tokens. In the realm of natural language processing, tokens are commonly represented as words. However, it is important to note that tokens can encompass a broader range of entities, including numerical values, punctuation symbols, and even emoticons. The significance lies in the ability to conduct sentiment analysis on specific words or phrases inside a tweet, thereby enabling the extraction of valuable insights.

Algorithm for Tokenization

Input: Selected Tweets

Output: Tokenized Tweets

For all words in Processed Tweets

Tokenize the word passing to Tweet Tokenizer Method

Append Tokenize Sentence

Return Tokenize Sentence

3.2.2 Lemmatization:

Lemmatization is a semantic procedure that entails the reduction of words to their fundamental or root form, referred to as the lemma. The objective of lemmatization is to standardize words in such a way that various imposed forms of a word are regarded as a singular token.

Algorithm for Lemmatization

Input: Tokenized Word

Output: lemmatized Word

For word in word Tokens

Initialize StemmedSentence variable to empty list

If length of word greater than 2

Method call for stemming the word using PorterStemmer object.

Method call for Lemmatizing the word using WordNetLemmatizer object

Return Lemmatized Sentence List

3.2.3 Misspell Correction:

The management of misspellings constitutes a significant component of text analysis, particularly in the context of analyzing Twitter data. Spelling errors have the potential to impact the precision of sentiment analysis, topic modeling, and other natural language processing (NLP) tasks.

3.2.4 Repeated character Removal:

Character repetition, also referred to as removing repeated characters, is a technique employed in the preprocessing stage of text analysis. This technique proves to be advantageous in a multitude of natural language processing applications. The process entails the reduction of consecutive instances of the same character within a word to a singular occurrence. This procedure aids in standardizing the text and mitigating the interference caused by an overabundance of repeated characters.

3.3 HDFS MAP REDUCE

Hadoop is a freely available framework that offers a system of distributed computers designed for the purpose of processing and analyzing extensive datasets. The Hadoop Distributed File System (HDFS) is a crucial element of the Hadoop framework, enabling the decentralized storage of data across numerous machines. An additional significant element is MapReduce, which is a programming paradigm and execution framework utilized for the parallel processing and analysis of data across a group of machines. The utilization of the MapReduce framework within the Hadoop system facilitates the accomplishment of scalable and proficient data processing for extensive datasets. This is achieved through the partitioning of data into smaller segments and the subsequent distribution of processing tasks across numerous nodes within a cluster [14,15].

3.4 FEATURE EXTRACTION

The features that are retrieved from our dataset include both unigrams and bigrams. These are the specific sorts of features that are extracted. For the purpose of conducting our study, we first generate a frequency distribution for the unigrams and bigrams that may be discovered inside the dataset, and then we choose the

N unigrams and bigrams that are most prevalent within that distribution.

3.4.1 Unigrams:

One of the most frequently employed and straightforward techniques for text classification involves considering the occurrence of individual words or tokens within the text. The process involves extracting individual words from the initial dataset and subsequently generating a frequency distribution based on these words. The dataset yields a total of 181,232 distinct words. Among these words, the majority of the words located towards the higher end of the frequency spectrum can be considered as noise, as they occur infrequently and have minimal impact on the classification process. Hence, we exclusively employ the highest N words from these sources to construct our vocabulary. In the case of sparse vector classification, *N* is set to 15000, while for dense vector classification, N is set to 90000 [16].

3.4.2 Bigrams:

Bigrams refer to pairs of words that appear consecutively in a given dataset or corpus. The aforementioned features provide an effective means of representing negation in natural language, as exemplified by the phrase This is not good. A dataset was used to extract a total of 1,954,953 unique bigrams. Among these options, the majority of the bigrams located at the tail end of the frequency spectrum can be considered as noise, as they occur infrequently and have minimal impact on the classification process. Consequently, we exclusively utilize the top 10000 bigrams in these sources in order to construct our vocabulary [17].

3.5 RANKING-SENTIWORDNET

SentiWordNet is a database created from WordNet that contains subjective information about individual words. SentiWordNet assigns numerical values to each phrase that reflect its emotion value and provides a definition for each term, which provides additional information about the word. For English speakers looking for a specialized lexical tool, SentiWordNet is an excellent option. Each definition in this dictionary is a set of words that have the same grammatical category and meaning regardless of their POS label. Each group is connected to three sentiment number ratings that show the degree to which the words it includes are positive, negative, or neutral. These values range from 0 to 100. The scores in question exhibit a range spanning from 0.0 to 1.0, and the cumulative total of these scores within each group amounts to 1.0. The term excellent is exclusively classified as an adjective, according to the information shown in Table.1, with a score of 1.0 for positive and 0.0 for negative. The term cold possesses a negative sentiment score of 0.75 when used as an adjective to describe a low or insufficient temperature. Additionally, when employed as a noun to refer to a mild viral infection, it carries a negative sentiment score of 0.125.

Table.1. Example of Pre-processing

	Text Data
Selected Tweet	I am connected with world cup and its GOOD Connecting each other with team World Cup Song connects Worldcupppp 2014 Brazi11112014
Tokenized Tweet	[I, am, connected, with, world, cup, and, its, GOOD, Connecting, each, other, with, team, World, up, Song, connects, Worldcupppp, 2014, Brazi11111[1], 2014]
Lemmatized Tweet	[connect, with, world, cup, and, it, good, connect, each, other, with, team, world, cup, song, connect, worldcupppp, 2014, brazi1111[1], 2014]
Misspell Correction	[connect, with, world, cup, and, it, good, connect, each, other, with, team, world, cup, song, connect, worldcup, 2014, brazil, 2014]
Repeated Character Removal	[connect, with, world, cup, and, it, good, connect, each, other, with, team, world, cup, song, connect, **worldcup,** 2014, brazil, 2014]
Sanitized POS tags with word	(I, None) (am, v) (connect, a) (with, None) (world, n) (cup, n) (and, None) (it, v) (GOOD, a) (Connect, n) (each, None) (other, a) (with, one) (team, n) (World, n) (Cup, n) (Song, n) (connect, n) (Worldcup, n) (2014, None) (Brazil, n) (2014, None)
Sentiment Score for the synset term obtained from the SentiWordNet Database	<i.n.01: PosScore=0.0 NegScore=0.0> <be.v.01: PosScore=0.25 NegScore=0.125> <universe.n.01: PosScore=0.0 NegScore=0.0> <cup.n.01: PosScore=0.0 NegScore=0.0> <good.a.01: PosScore=0.75 NegScore=0.0> <each.s.01: PosScore=0.0 NegScore=0.0> <other.a.01: PosScore-0.0 NegScore=0.625> <team.n.01: PosScore=0.0 NegScore=0.0> <universe.n.01: PosScore=0.0 NegScore=0.0> <cup.n.01: PosScore=0.0 NegScore=0.0> <song.n.01: PosScore=0.0 NegScore=0.0> <brazil.n.01: PosScore=0.0 NegScore=0.0>

3.6 CLASSIFICATION USING MODIFIED CONVOLUTIONAL NEURAL NETWORK (MCNN)

The features were categorized using the MCNN algorithm. In this study, the researchers employ a CNN for the purpose of classification. Through the use of a hybrid PSOGA method, the values of the weights are optimized. The MCNN (Multi-Convolutional Neural Network) is a class of trainable algorithms that have the ability to acquire knowledge and solve complex problems. This is achieved through the utilization of training data, which consists of a collection of input-output pairs, where the desired outputs (targets) are provided. The system is composed of a collection of neurons, represented by functions, that are interconnected in various layers. Each layer consists of neurons. The problem that needs to be addressed is indicated by input patterns, which are transmitted through the layers. The information is represented and processed through the synaptic weights that are associated with it. The optimization of the weight value is achieved through the utilization of the CS method, as described in the subsequent section. In the context of Deep Learning, the Modified Neural Network incorporates an augmentation wherein the number of hidden layers is expanded with the objective of mitigating the training error. The architectural configuration of the Multi-Column Convolutional Neural Network (MCNN) is visually depicted in Fig.2.

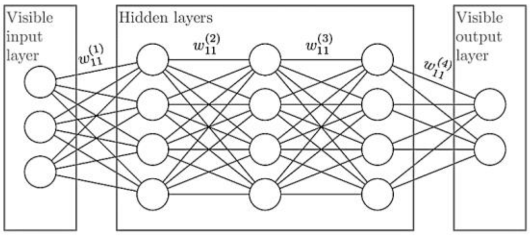

Fig.2. Structure of DCNN with three Hidden layer

Algorithm for MCNN

The Modified Deep Learning CNN algorithm is follows:

Step 1: Generate random weights using a uniform distribution and place them inside the range [0, 1] before assigning them to the neurons in the hidden layer as well as the output layer. It is imperative to uphold a uniform weight value for all neurons within the input layer.

Step 2: The following equation is used to determine the output of the hidden layer in the network:

$$Hidden_{out} = B_i + \sum_{i=0}^{j} X_i W_j \tag{1}$$

The bias value is denoted as B_i, the input characteristics from the features that were obtained are represented as X_i, and the weighted average value of the input that was provided features is indicated by W_j.

Step 3: It is possible to calculate the final output unit by multiplying the hidden unit by the total amount of weight of the output from the hidden layer, as shown in the Eq.(2). This will give you the final output unit.

$$Output_{layer} = B_i + \sum_{i=0}^{j} Hidden_{out_i} W_{jk_i} \tag{2}$$

where, $Output_{layer}$ is the hidden unit and W_{jki} is the hidden layer weights and $output_{layer}$ denotes the output unit.

The activation function utilized for the output layer is determined as:

$$Act_{fun} = \frac{1}{\left(1+e^{-\left(output_{layer}\right)}\right)} \tag{3}$$

Step 4: Recognize the learning error as offered beneath

$$Learning_{error} = [target_{output}\text{-}Output_{layer}] \tag{4}$$

In order to maximize the output value, it is necessary to optimize the weight in the hidden layer through the utilization of the Particle Swarm Optimization with Genetic Algorithm (PSOGA) algorithm.

3.6.1 PSOGA:

Particle Swarm Optimization (PSO) is a straightforward optimization technique that is easily applicable and requires minimal parameter modification. The Particle Swarm Optimization (PSO) algorithm is straightforward to implement and requires only a small number of parameters. Moreover, it is extensively employed to address optimization challenges and the issue of feature selection. The Genetic Algorithm is widely recognized for its adaptability in facilitating the integration of other methodologies, resulting in improved solutions. Both feature selection and optimization of parameters of SVM are commonly performed using genetic algorithms. Furthermore, it is widely recognized for its ease of alignment and applicability in classification tasks, similar to other optimization problems.

Algorithm: Hybrid PSOGA Algorithm - feature optimization

Training tweets are utilized for the purposes of tokenization and preprocessing, resulting in the creation of a training set.

Input: Feature set from input layer

Output: Optimized features for hidden layer

Step 1: Initialize the population from input layer

Step 2: for each particle X in I do

Step 3: if $f(X_i) == f(pd_i)$ then $pd_i = X_i$; End if

Step 4: if $f(X_i) == f(nd_t)$ then

$$nd_i = X_i;$$

Else

$$gd_i = (gd_i, X_i)$$

End for

Step 5: End if the conditions satisfied else

Step 6: Goto step 3

Step 7: update new particle population S

Step 8: while isNotTerminated () do

$P_p(S)=P(S)$.optimizePositive

$P_q(S) = P(S)$.optimizeNegative

Step 9: Mutate $P_p(S)$

Step 10: Mutate $P_q(S)$

Step 11: Evaluate $P_p(S)$ and $P_q(S)$

Step 12: Return Optimizes $P_p(S)$, $P_q(S)$

Step 13: end While

3.6.2 K-fold Cross Validation:

The practice of cross-validation is common in the field of predictive modeling. This technique includes separating the initial dataset into two unique subsets—a training set and a test set—so that each may be utilized independently. The primary sample is randomly split into k different subgroups of the same size as part of the k-fold cross-validation methodology. One particular subset out of all of the available subsets has been singled out as the validation data for the objective of putting the model through its paces, while the remaining k-1 subsets have been put to use to educate the model. After then, the cross-validation method is repeated k times, where k stands for the total number of folds. During every cycle, one of the k subsets is utilized only for the purpose of providing the data needed for validation, while the other subsets are put to use for training purposes. Calculating the accuracy that is averaged out across all k-folds is what is used to get the final accuracy. The approach of cross-validation with tenfold replication is one that is often used in a large number of investigations. All of the examples that are contained inside the dataset are employed for the purposes of the 10-fold cross validation, and these instances are then divided into 10 separate groups. There are a total of ten of these groups; nine of them are used for training reasons, while the other group is set aside for examination. Iterations of the method are carried out for a total of 10 times, after which the average accuracy obtained across all of the iterations is calculated.

3.7 RESULT ANALYSIS AND DISCUSSION

Text preprocessing is performed on all of the gathered tweets in this phase. This procedure comprises separating each word into a token using the tokenization approach. The following step is the extraction of characteristics from the tweets, followed by the development of scores and an orientation for the tweets. The efficiency and effectiveness of our proposed classifier are assessed by evaluating its accuracy, precision, and recall metrics.

Table.2. Dataset Description

		Positive	Neutral	Negative
Train Data	45000	18240	14313	1244
Test Data	43218	16182	17342	9694

Performance analysis is performed by evaluating following performance metrics:

Accuracy: The term accuracy is frequently employed in academic discourse to denote the degree of correctness or precision exhibited by an estimation, calculation, or prediction. The measure is frequently represented as either a percentage or a decimal ranging from 0 to 1, with 1 denoting complete accuracy and 0 indicating a complete lack of accuracy.

$$Accuracy\,(ACC) = \frac{TP+TN}{TP+TN+FP+FN} \tag{5}$$

Specificity: It is sometimes referred to as the positive predictive value, and it indicates the percentage of relevant recovered occurrences instead of the percentage of accurate picked items.

$$Specificity = \frac{TP}{TP+FP} \tag{6}$$

Sensitivity: Recall is commonly acknowledged as the ratio of relevant instances that are successfully retrieved, or alternatively, it can be understood as the proportion of correctly identified items that are chosen.

$$Sensitivity = \frac{TP}{TP+FN} \tag{7}$$

F-Score: The metric in question is a composite measure that combines precision and recall metrics. It can be conceptualized as a weighted harmonic average or as a comprehensive evaluation of the trade-off between these two metrics. The concepts of sensitivity and specificity are fundamental measures used in various academic disciplines, particularly in the field of statistics and medical research. Sensitivity is calculated by using equation (8)

$$F-Score = 2\cdot\frac{(Sensitivity)(Specificity)}{(Sensitivity)+(Specificity)} \tag{8}$$

Table.3. Overall performance of Proposed work

	Accuracy	Specificity	Sensitivity	F-Score
Positive	93.5	89.4	83.2	89.6
Negative	89.3	87.3	86.5	86.3
Neutral	88.4	85.9	81.4	84.9
Overall	90.4	87.5	83.7	86.9

The hybrid algorithm proposed in this study is utilized for the purpose of classifying the emotions and polarity of tweets. The tweets were categorized into three distinct classes, specifically referred to as positive, neutral, and negative. The first category comprises tweets related to the themes of happiness, fun, and love, whereas the second category consists of tweets associated with hate, anger, and sadness. The Table.3 displays the classification result that the suggested hybrid approach, which combines particle swarm optimization and genetic algorithm, was able to achieve.

The classifier exhibits an overall accuracy of approximately 90.4%. Specifically, the positive accuracy stands at around 93.5%, the negative accuracy at approximately 89.3%, and the neutral accuracy at 88.4%. The observed precision for both positive and negative tweets is significantly high, reaching 87.5%. The recall measure acquired a value of roughly 83.7% for both the positive and the negative polarities, while the F-measure obtained a value of approximately 86.9% for each polarity.

The task of multi-class classification involves the categorization of emotion classes, specifically those pertaining to anger, disgust, fear, joy, sadness, surprise, and N.A. Fig.4 illustrates the classifications of feelings that have been made for each of the classes using the method that we have suggested. The analysis indicates that the class labeled as anger contains a greater number of datasets.

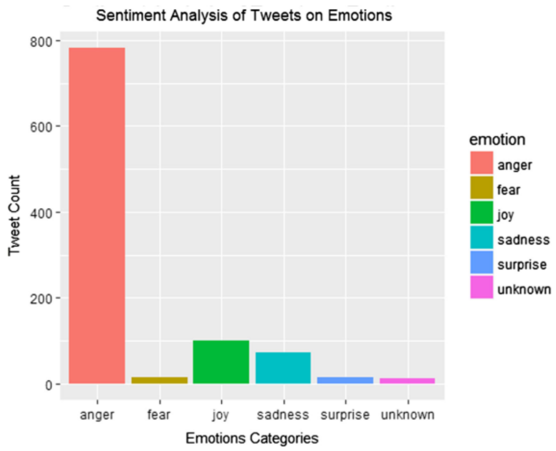

Fig.4. Classified emotions on proposed work

In this investigation, we examine the efficacy of our suggested work by contrasting it with two different approaches that are already in use like Cuckoo Search (CS) and Modified Decision Tree (MDT) with MANN. Additionally, the table provided in Fig.4 displays various other models that are currently in existence. The performance comparison demonstrates that our proposed methodology attains higher levels of accuracy in comparison to other methods that have been previously presented.

Table.4. Performance comparison Proposed Vs Existing

	Accuracy	Specificity	Sensitivity	F-Score
CS-MANN	84.2	77.6	79.4	82.8
PSOGA-MCNN	90.4	83.7	87.5	86.9
KNN	67	70.5	69.3	67.9
SVM	68	69	68.1	68.7

KNN+SVM	76	68.5	68.1	77.5
GA	86	87.3	87.9	87.5
PSO	88	88.7	89.1	81.4

Table.5. Performance of K-Fold validation

Epoch	Acc_ Training	Loss_ Training	Loss_ Validation	Acc_ Validation
Epoch-1	0.6781	0.5431	0.5724	0.6687
Epoch-2	0.6984	0.5791	0.5894	0.6739
Epoch-3	0.7031	0.5657	0.5885	0.6746
Epoch-4	0.6968	0.5687	0.5728	0.6821
Epoch-5	0.7003	0.5625	0.5739	0.6737
Epoch-6	0.7064	0.5698	0.5787	0.6808
Epoch-7	0.7032	0.5579	0.5797	0.6814
Epoch-8	0.7073	0.5519	0.5789	0.6823

Within the framework of the K-fold cross validation method, the dataset is partitioned into k different subsets. One subset is held out for validation purposes, while the remaining subsets are combined and used for training. The trained model is then evaluated against the held-out subset. The aforementioned procedure is iterated k times, with each iteration representing a distinct fold, wherein a distinct portion is withheld. After that, we take the scores that we got for each fold and average them in order to get a more accurate indication of how well our model works. The Table.5 contains the findings that were obtained using the k-fold approach.

According to the evidence, training-phase accuracy is better than validation-phase accuracy. Additionally, the loss during the validation phase is greater than the accuracy. Ultimately, the performance of the work we suggested surpasses that of all other existing models.

4. CONCLUSION AND FUTURE WORK

This article proposes an innovative hybrid technique for the categorization of sentiment analysis data. The integration of PSO and GA, referred to as PSOGA, has demonstrated superior performance in comparison to alternative algorithms currently available. The utilization of a machine learning classifier in this optimization technique yields a classification accuracy exceeding 90% for sentiment analysis of tweets, categorizing them into the classes of positive, negative, and neutral. The proposed model undergoes two primary stages in its training process: preprocessing and feature generation. In subsequent research endeavors, it is possible to expand upon our proposed methodology by incorporating alternative classifiers. Additionally, a significant advancement in sentiment classification could be achieved by altering the optimization problem.

REFERENCES

[1] Saurabh Singh, "Twitter Sentiments Analysis using Machine Learning", *International Journal of Scientific Research in Computer Science, Engineering and Information Technology*, Vol. 6, No. 4, pp. 1-9, 2020.

[2] Senthil Murugan Nagarajan and Usha Devi Gandhi, "Classifying Streaming of Twitter Data based on Sentiment Analysis using Hybridization", *Neural Computing and Applications*, Vol. 31, pp. 1425-1433, 2018.

[3] R.B. Shamantha and P. Rai, "Sentiment Analysis using Machine Learning Classifiers: Evaluation of Performance", *Proceedings of International Conference on Computer and Communication Systems*, pp. 21-25, 2019.

[4] P. Tyagi and R.C. Tripathi, "A Review Towards the Sentiment Analysis Techniques for the Analysis of Twitter Data", *Proceedings of International Conference on Advanced Computing and Software Engineering*, pp. 1-6, 2019.

[5] Łukasz Augustyniak, Piotr Szymanski, Tomasz Kajdanowicz and Włodzimierz Tuliglowicz, "Comprehensive Study on Lexicon-based Ensemble Classification Sentiment Analysis", *Entropy*, Vol. 18, No. 1, pp. 1-7, 2015.

[6] B.M. Bandgar and Binod Kumar, "Real Time Extraction and Processing of Social Tweets", *International Journal of Computer Sciences and Engineering*, Vol. 3, No. 3, pp. 1-13, 2015.

[7] Vedurumudi Priyanka, "Twitter Sentiment Analysis using Deep Learning", *SSRN*, Vol. 89, pp. 1-17, 2021.

[8] Vishal A. Kharde and S.S. Sonawane, "Sentiment Analysis of Twitter Data: A Survey of Techniques", *International Journal of Computer Applications*, Vol. 139, No. 11, pp. 1-14, 2016.

[9] Bandaru Mounika and M.S.V.S. Bhadri Raju, "Sentiment Analysis of Twitter Data using Machine Learning Approaches", *International Journal of Research*, Vol. 5, No. 4, pp. 1-13, 2018.

[10] Vishu Tyagi, Ashwini Kumar and Sanjoy Das, "Sentiment Analysis on Twitter Data using Deep Learning approach", *Proceedings of International Conference on Advances in Computing, Communication Control and Networking*, pp. 1-5, 2020.

[11] S. Zhai and Z.M. Zhang, "Semi Supervised Auto Encoder for Sentiment Analysis", *Proceedings of International Conference on Artificial Intelligence*, pp. 14-18, 2016.

[12] Nikhil Yadav and Ajitkumar Shitole, "*Twitter Sentiment Analysis using Supervised Machine Learning*", Springer, 2020.