Benchmarking Machine Learning Algorithms for Twitter Sentiment Prediction

Mr. Rahul Shivhare[1], Shivika Tomar[2], Paras Goel[2], Manav Gupta[4]
[1]*Assistant Professor, Department of Information Technology, MIET*
[2,3,4] *Department of Information technology, MIET*

***Abstract*— With a focus on the comparison of the performance of different algorithms, this study examines sentiment analysis on Twitter data through machine learning approaches. The dataset's tweets contain four sentiment classes: positive, negative, neutral, and irrelevant. TF-IDF vectorization was part of the preprocessing steps for converting raw text into numerical features suitable for model training. Multinomial Naive Bayes, Logistic Regression, Support Vector Machine, Decision Tree, Random Forest, and Artificial Neural Network (ANN) were the six machine learning algorithms that were implemented. When these models' accuracies for classification were measured, the highest accuracy was that of the Random Forest model (97.8%), followed by that of the ANN (97.3%). Though they did a great job, basic models like Naive Bayes and Logistic Regression lagged behind the advanced approaches. The results indicate how efficiently complicated models carry out sentiment analysis operations and provide the foundation for subsequent studies on ensemble methods or another model optimization.**

Keywords—Count Vectorizer, Multinomial Naïve Bayes, NLP Pipeline, Accuracy

I. INTRODUCTION

Social networking sites like Twitter have become powerful communication and public expression tools in the modern digital age. Businesses, governments, and researchers can gain immensely by analyzing tweet sentiment to discover patterns, measure public sentiment, and make informed decisions. Four sentiment categories—positive, negative, neutral, and irrelevant—are used to classify tweets for this project's sentiment analysis of Twitter feed.

The project investigates and compares a range of machine learning algorithms to achieve accurate sentiment classification, ranging from classical methods like Logistic Regression and Multinomial Naive Bayes to sophisticated methods like Support Vector Machines, Decision Trees, Random Forests, and Artificial Neural Networks (ANNs). Tens of thousands of labeled tweets made up the dataset, which was preprocessed using TF-IDF vectorization to convert raw text into numerical features.

Top-one classification accuracy was used to compare the performance of each algorithm so that their relative effectiveness can be directly compared. The Random Forest model performed better than the ANN, demonstrating their capacity to handle complex patterns in text data. Beyond identifying the most appropriate algorithms for sentiment analysis, this comparison study shows how machine learning can be used to extract meaningful insights from unstructured social media data.

II. LITERATURE REVIEW

The majority of the research on sentiment analysis of Twitter fall into two categories: supervised approaches [1,2,3,4,5,6] and lexicon-based approaches [7,8,9]. Supervised approaches train a classifier (e.g., Naive Bayes, Support Vector Machine, Random Forest, etc.) with a set of features together such as POS tags, word N-grams, and tweet context features like hashtags, retweets, emoticons, capital words, etc. Lexicon-based approaches like Senti WordNet utilize existing lexicons of words and their sentiment orientation to measure the overall sentiment inclination of a text [10]. Not much multiple class sentiment analysis studies exist. Most of them tried to assign numerical sentiment scores to the texts or classify the strength of the sentiment into more than one sentiment strength levels (e.g., extremely negative, negative, neutral, positive, and very positive) [11], [12].

A recommendation system for emoticons was proposed by Liang et al. [13] to assist authors in selecting the appropriate emoticon to be used in posted messages to convey their intended meaning. Indeed, the term "multi-class classification" has classically been used to denote classifying a tweet or a text with one of a given set of sentiment strengths. Classifying tweets with one of the following sentiment classes—i.e., "very negative," "negative," "neutral," "positive," and "very positive"—or simply classifying with a score between -1 and 1 that also indicates the polarity and strength of the sentiment was a common task of categorization [14][15].

Statistical machine learning algorithms are not applicable to more complex text classification tasks, although they perform well in simple sentiment analysis tasks [16], [17]. Deep learning models, however, provide excellent performance in computer vision [20], speech recognition [19], and sentiment analysis [18].

LeCun et al. [21] proposed a CNN model to learn the local responses from temporal or spatial data and obtain the features of the text. To decrease the training parameters and the computational complexity, he applied the weight sharing technique of the CNN model. CNN cannot learn the correlation sequence, however, and its performance heavily relies on choosing the proper window size [22].

Even Go et al. [23] performed one of the earliest works on Twitter sentiment analysis, classifying the tweets into positive or negative and defining the problem as a two-class classification.

To define about 3 lac tweets, Go et al. [23] and Pak and Paroubek [24] employed emoticons such as labelling. Unlike Go et al., Pak & Paroubek tried to classify tweets into good, negative, or neutral categories while developing tweet classification as more than two class classification problems. They subsequently employed distinctive feature sets such as Unigrams, Bigrams, different n-grams, and n-gram position to explore the implementation of Support Vector Machine and Multiple Naïve Bayes. Based on the findings, MNB, n-grams, and POS tagging were the best combination.

III. METHODOLOGY

A. Necessary Modules

1. Pandas:

Part of data processing in This Twitter Sentiment processing project is the Pandas library. Capable of manipulating data effectively it makes dataset's structured format easy to manage. With Pandas Data Frames we Can load the dataset. It has twitter_training.csv and twitter_validation.csv files. This gives a tabular structure for easy analysis. It can manage missing values and get particular columns easily. For example, columns like Tweet and Sentiment. it also has exploratory data evaluation support. This helps To get an idea of the sentiment class distribution. These are Irrelevant, Negative, Neutral and Positive. Pandas has strong grouping And filtering support. These are important. They allow precise insights into Data trends. Pandas supports feature engineering and Visualization tasks. it is compatible with libraries like NumPy and matplotlib. Pandas is A vital part in Managing Data pipeline in This project. It makes tasks like combining datasets ,converting language to machine-readable format and Exporting processed data easy.

2. NumPy:

Twitter Sentiment Analysis Project Relies extensively on NumPy toolkit. This Toolkit provides Efficient array manipulation and numerical computations. While working with high-dimensional TF-IDF feature matrices in text vectorization NumPy saves the day. This is the foundation for Handling numerical data in Python. It ensures smooth data handling. The backbone for handling datasets in this project is NumPy. Nd array. It facilitates efficient operations and storage on 15,000-dimensional feature vectors. Calculations for determining model predictions and accuracy are simplified by NumPy functions. These are NumPy mean and NumPy. argmax. In addition, array slicing, matrix manipulations and Smooth Data type conversions are facilitated with NumPy. This is necessary for preprocessing and preparing Data for machine learning models. NumPy usage enhances interoperability. It is capable of working with other libraries. Libraries like scikit-learn and pandas ensure smooth data transfer across project pipeline.

3. Scikit-learn:

Scikit-learn is a general-purpose, wide Python library with applications across multiple machine learning operations such as classification, regression, and clustering. Scikit-learn was heavily utilized in the development, training, and testing of machine learning models for this Twitter sentiment analysis project. Scikit-learn offered simple-to-use tools for text data preprocessing such as TF-IDF vectorization, which converted tweets into numerical features that machine learning algorithms can process. Scikit-learn offered simple usage of models such as Multinomial Naive Bayes, Logistic Regression, Support Vector Machines, Decision Trees, and Random Forest. These models were simply trainable and testable using scikit-learn's consistent and simple-to-use API. Scikit-learn functions also enabled simple hyperparameter tuning, scoring accuracy, and performance measurement based on metrics such as classification accuracy.

B. Preprocessing

1. Tokenization

Tokenization, arguably our system's most important operation, is the activity of dividing incoming tweets into separate units of information called tokens. Tokens allow the system to parse and interpret the text of tweets by dividing specific words or phrases. For executing the Multinomial Naïve Bayes Classification Algorithm and extracting valuable features, the activity is a prerequisite. Finally, tokenization makes sentiment classification more efficient and increases the performance of our cutting-edge Twitter sentiment analysis software by increasing our system's capacity for detecting trends, sentiment, and nuances in unstructured text.

2. Punctuation removal

One Eliminating punctuation is a key step in sentiment analysis. This step improves machine learning and offers clean textual Data. This action assists an exclusive focus on text. it gets rid of the text of different types of punctuation. these Include commas and periods, and Exclamation or Question marks. The action Erases punctuation. This emphasizes the correct words. The Attention is diverted away from unwanted symbols. Noise is minimized. Python's regular expressions are used in this operation. In addition to string manipulation techniques, they are used in this stage. This stage is integrated and is smoothly passed to the preprocessing pipeline. By getting rid of punctuation, the clearness and consistency in feature extraction are improved. The Process of TF-IDF vectorization. can also be integrated in This operation. It Guarantees that text is standardized. The Goal is to enhance models for sentiment analysis. In this operation, Python’s regular expressions play a crucial role. They work with string manipulation techniques. Both are essential for this stage. They are integrated easily into the preprocessing pipeline. Punctuation is removed and finally, models for sentiment analysis are improved.

3. Digit Removal

In order to avoid irrelevant numeric data from affecting sentiment categorization digitization is required. This is a major process in Twitter sentiment research. Tweets contain numeric numbers. those are not useful resource sentiment categorization. These Numbers are dates, numbers or user codes. In the cleaning phases Digits Are found and eliminated from text. This is executed through preprocessing packages or regular expressions. these are completed in python. noise is removed from the data. This helps the Models concentrate on significant words and phrases. The feature extraction process becomes more effective and focused. Methods such as TF-IDF vectorization are improved. Therefore, the sentiment analysis is more accurate with numbers removed.

4. Stop Words Removal

Words like "is," "the," or "and," are referred to as stop words. They are frequently used but do not carry much meaning in sentiment analysis. By eliminating these words machine Learning models become more precise. it also makes them more efficient. The aim is to pull the focus towards the more important phrases in the text. During the process of improving the input Data For this project, stop words Were eliminated. The process Took during the preprocessing stage Techniques employed were: altering the list of stop words using tools Options were: Nltk or the scikit-learn tool. The purpose? To get rid of words irrelevant to the context. The purpose was also to keep related terms advantage of this process is that the noise within the dataset was minimized. This step allowed machine learning Models to concentrate on words that conveyed sentiment. The outcome? Improved classification results were yielded.

5. Stemming

Stemming is the basis of sentiment analysis. It is important to text preprocessing This process brings words to their root or base form Breaking a word into its Constituent parts enables it to be analyzed This will provide you with its basic meaning. In this research, stemming was used To Normalize word variants. Words such as "running and "runner were recalled to their Common origin. The origin was run." By doing this the feature space's dimensions reduced. The models then could concentrate Better on individual word meanings. This kind of concentration is referred to as generalization. Stemming improves the accuracy of sentiment classification. It does this for the Twitter dataset. It excludes word endings and suffixes. This makes models treat similar semantically terms as the same. It Is an important tool. It is important to mention that Twitter data can Be highly noisy. In the case of data in This set stemming simplifies most words. There is a chance of losing meaning. Regardless of this, the technique still Helps in reducing dimensions. It also helps models in Concentrating better. Therefore, stemming is an excellent tool in sentiment analysis.

6. Lemmatization

For consistency in Sentiment analysis tasks, lemmatization comes into play. It is a pre-processing process. Its intention is to maintain consistency across varied word forms. The Process is regarding breaking words down to the root form. For instance, "running" and "ran" Are brought down to lemma "run." This process doesn't merely handle the word form issues in the dataset. It aids in lowering the feature set complexity. This results in greater model accuracy. Lemmatization was employed within this study. It Was for term normalization within tweets. This resulted in improved machine learning model performance. The outcome was A more Accurate sentiment categorization.

7. Creating N-grams

Preprocessing of data from Twitter was required for this Sentiment analysis research. This comprised creating n-grams These n-grams are sequential groups of n-word units. These are useful for retaining patterns and context Individual words or unigrams may miss out on these. In order to make n-grams we used Count vectorizer And TfidfVectorizer. They are included in scikit-learn. N-grams ranged from bigrams (two-word combinations) to trigrams (three-word combinations). Through the production of n-grams, we Improved models' sentiment comprehension. This allowed Them to Account for the interaction between adjacent words As well. N-grams also enriched our representation of features for tweets. They helped towards a better performance in consistently classifying sentiment for many classes.

8. Count Vectorizer

Sentiment analysis test turned raw text data into numbers. This was Done using scikit-learn's Count Vectorizer tool. It counts the number of times Each Word appears in a tweet .This results In a matrix With sparse data. Every word In the corpus Is associated with a specific feature. This method is an important step in converting raw data into structured form. It's essential for Machine learning models. Count Vectorizer is key in training Different models .Two such Models Are Multinomial Naive Bayes and Support Vector Machine. Its job is to classify sentiment in Twitter messages. The tool captures word frequencies. This provides important information for Sentiment analysis experiments. It can Be a tough task. Count Vectorizer makes it manageable.

IV. MODELS USED

1. Multiple Naïve Bayes

Used in this sentiment analysis study was a probabilistic classifier. This is the Bayes' Theorem based classifier. We are referring to Multinomial Naïve Bayes. It Makes an assumption. The concept is that the presence of a word in a tweet is independent of other words. This model is ideal. It is ideal For Text classification issues.

Issues Where word frequency is important. Examples are sentiment analysis. Here, we Have Multinomial Naïve Bayes. It concluded probability of each class of sentiment. These classes are Positive, Negative, Neutral and irrelevant. It concluded using word frequencies. The Classifier employed training data's dataset. The dataset was converted into a usable form. This conversion used an approach of count vectorization. Another approach that was used is TF-IDF. This classifier gave a sound baseline. This baseline was with which we were able to compare against other models. The accuracy was around 74.37%.

2. Decision Tree

The Decision Tree model was used within this research. It classified sentiment of tweets. It is a supervised learning approach. It produces tree-like decisions in this model. It separates Data into features' values. Each leaf node represents An emotion class. Examples of classes: irrelevant, negative, neutral, or positive. Each internal node represents a feature. In our case, the word frequencies by Count Vectorizer were employed. They trained The model on preprocessed data. The Decision Tree had an impressive accuracy rate 95.6%. This Tree operates with complex patterns in text data.

3. Logistic Regression

The sentiment classification of the study was based on Logistic Regression. It was employed As a baseline model. Logistic Regression Is a linear model. It makes use of the input data of the tweets It predicts the likelihood of the specific sentiment. Sentiment can be Either irrelevant, negative, neutral and positive. Output probabilities of Logistic Regression are mapped to a sentiment class. This is achieved through sigmoid function. Accuracy in this model is remarkable It's at 85.49%. This model's success in the project is established. The model Is easy. It's quick .It can manage an enormous feature Space well.

4. Random Forest

This Study employed Twitter data. It was annotated with the Random Forest algorithm sentiment. The goal was to increase accuracy. Furthermore, there was a requirement to avoid overfitting. The Random Forest method Is a type of Ensemble learning. It constructs numerous decision trees. Subsequently, it Aggregates The predictions from these trees. A final prediction is generated when the Output of each tree is aggregated. These trees are trained on randomly selected parts of data. Increasing model's generalization ability, high classification accuracy of 97.8% was obtained. In The prediction of sentiments Positive, Negative and Neutral this technique performed better than other models. It also picked out complex data patterns successfully. Generating prediction, the Random Forest model performed better than other models. It did so for sentiments Like Positive, Negative and Neutral. The model also picked out data patterns successfully.

5. Support Vector Machine

Support Vector Machine (SVM) proves to be a powerful supervised learning method. It has a very wide range of applications in the area of Twitter Sentiment analysis. SVM's capability to find an optimal hyperplane is equally strong. The hyperplane clearly classifies data points into sentiment classes. The classes here include Positive, Negative ,Neutral, and Irrelevant. Implementation of scikit-learn is the second step. By this, tweets get transformed. The Numerical features Are created using TF-IDF vectorization. These Features are then passed Directly into the SVM model. Multi-dimensionality of Data is no issue. SVM model takes care of it. Sparse features also get taken care of. It's a high performer. This can be seen in its high accuracy marks. Numbers tell a lot. SVM has an accuracy of 89.69%. Accuracy is a gigantic Strength of SVM. Its Accuracy Is Only Surpassed by Its scalability. It definitely is A safe bet for text classification tasks.

6. Artificial Neural Network

An Artificial Neural Network (ANN) was used for the project. It classified Twitter messages according to their sentiment. Ann leveraged its ability to identify intricate patterns and relationships. This Powered it to attain a High accuracy level. The input layer handled TFIDF vectorized tweets They were then passed through One or more hidden layers. The hidden layers utilized activation functions Such as ReLU The function played an important role to achieve non-linear features. Sentiment labels such As Positive and Negative were forecasted. The forecasts were created by The final output layer. The layer applied a SoftMax activation. Outputs were in the form of Irrelevant, Negative, Neutral or Positive labels. The ANN was Effective in processing complex textual data. It demonstrated high accuracy at 97.3%. The execution was superb. This again indicates the efficacy of ANN in dealing with complex text input.

V. RESULTS OBTAINED

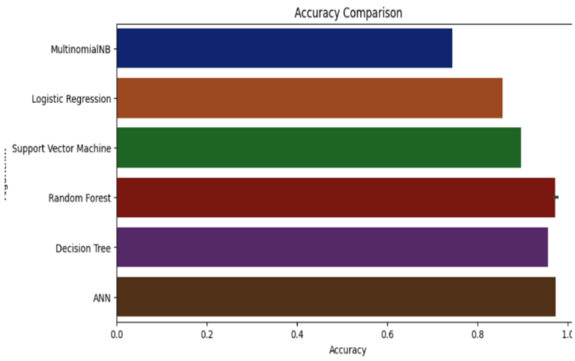

Figure 1 Accuracy Comparison Chart

MultinomialNB --> 0.7437437437437437
Logistic Regression --> 0.8548548548548549
Support Vector Machine --> 0.8968968968968969
Random Forest --> 0.977977977977978
Decision Tree --> 0.955955955955956
ANN --> 0.972972972972973

Figure 2 Accuracy values

After comparing all the models which are explained above we obtain maximum accuracy with the use of Random Forest which is 97.79%. The classification report and confusion matrix are given below

```
Classification Report for Random Forest:
              precision    recall  f1-score   support

  Irrelevant       0.99      0.98      0.99       171
    Negative       0.96      0.98      0.97       266
     Neutral       0.98      0.96      0.97       285
    Positive       0.98      0.99      0.98       277

    accuracy                           0.98       999
   macro avg       0.98      0.98      0.98       999
weighted avg       0.98      0.98      0.98       999

Random Forest Accuracy: 0.9780
```

Figure 3 Classification Report

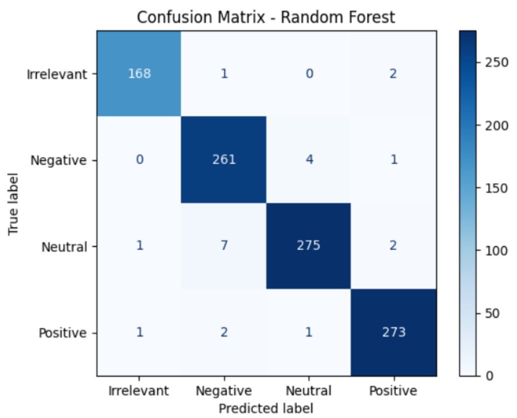

Figure 4 Confusion Matrix

The project effectively compared numerous Machine learning models for Twitter Sentiment analysis. This analysis was on Twitter data. The findings of this comparison were remarkable. Random Forest beat ANN with 97.8% accuracy. The Artificial Neural Network fell short. It came second Position With a 97.3% accuracy. The Findings are clear. Complex models beat less complex algorithms. Algorithms such as Naive Bayes and Logistic Regression belong to this group. These models Beat these more basic algorithms. They are superior at detecting nuances of feeling. Random Forest is A Flexible machine-learning model. It is applied across many fields. Twitter sentiment analysis is one among many uses.

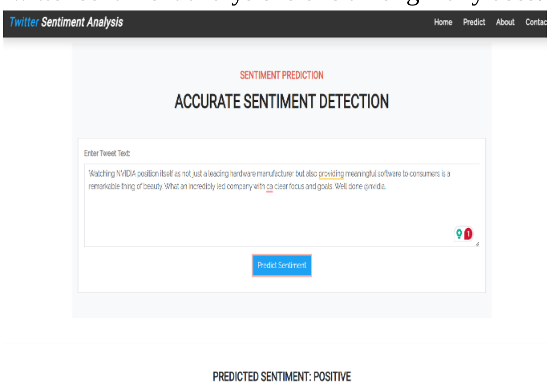

Figure 5 Testing Model on real positive tweet

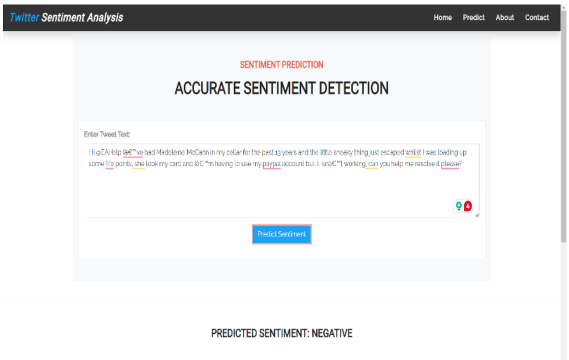

Figure 6 Testing Model on real negative tweet

Algorithm	Accuracy
Multinomial Naive Bayes	74.37%
Logistic Regression	85.48%
Support Vector Machine	89.69%
Random Forest	97.80%
Decision Tree	95.59%
Artificial Neural Network (ANN)	97.30%

Table 1 Accuracy Comparison in percentage

VI. CONCLUSION

Objective of research was to compare and analyze Different machine learning models. This assessment was done for classifying sentiments in Twitter data. Research wanted to give specific attention to models' ability to capture sentiment nuances. There were six Algorithms Used in this research which were Multinomial Naive Bayes, Logistic Regression Support Vector Machine, Decision Tree, Random Forest Artificial Neural Network Data for this analysis was vectorized using TF-IDF It was seen that models exhibited discernible differences in their performance. This Difference was when they were tested using top-one classification accuracy.

One significant Performance was Of the Random Forest. It achieved the Highest performance of 97.8% accuracy. It was closely followed by the ANN that recorded an accuracy of 97.3%. These models exhibited strength in handling a diverse range of intricate text data. They were able to elegantly walk through the complexity Involved In interpreting it. But they demonstrated lesser capacity Than other models to find sophisticated patterns in the data. Models like the Naive Bayes and the Logistic Regression, with their Simpler designs, were prone to this behavior.

REFERENCES

[1] Saif, H., He, Y. & Alani, H., Semantic sentiment analysis of twitter. The Semantic Web–ISWC 2012, pp.508-524,2012.

[2] Kiritchenko, S., Zhu, X., & Mohammad, S. M. Sentiment Analysis of Short Informal Texts. Journal of Artificial Intelligence Research, 50, pp.723–762,2014.

[3] da Silva, N. F. F., Hruschka, E. R., &Hruschka, E. R.. Tweet sentiment analysis with classifier ensembles. Decision Support Systems, 66, pp.170-179,2014.

[4] Hagen, M., Potthast, M., Büchner, M., & Stein, B.. Twitter Sentiment Detection via Ensemble Classification Using Averaged Confidence Scores. In European Conference on Information Retrieval, Springer, Cham, 2015, pp. 741-754,2015.

[5] Jianqiang Z, Xueliang C. Combining Semantic and Prior Polarity for Boosting Twitter Sentiment Analysis. In Proc. Smart City/SocialCom/SustainCom (SmartCity), 2015 IEEE International Conference on. IEEE, pp.832-837,2015.

[6] Jianqiang Z. Combing Semantic and Prior Polarity Features for Boosting Twitter Sentiment Analysis Using Ensemble Learning. In Proc. Data Science in Cyberspace (DSC), IEEE International Conference on. IEEE, pp.709-714,2016.

[7] Thelwall, M., Buckley, K., Paltoglou, G. Sentiment strength detection for the social web. J.American Society for Information Science and Technology 63(1), pp.163–173,2012.

[8] Paltoglou, G., &Thelwall, M. Twitter, MySpace, Digg Unsupervised Sentiment Analysis in Social Media. ACM Transactions on Intelligent Systems and Technology, 3(4), pp.1-19,2012.

[9] Montejo-Ráez, A., Martínez-Cámara, E., Martín-Valdivia, M. T., &Ureña-López, L. A..A knowledge-based approach for polarity classification in Twitter.Journal of the Association for Information Science and Technology, 65(2), pp.414-425, 2014.

[10] Baccianella S, Esuli A & Sebastiani F. "Sentiwordnet 3.0: An enhanced lexical resource for sentiment analysis and opinion mining". In Proc. the Seventh conference on International Language Resources and Evaluation, Malta. Retrieved May. Valletta, Malta, pp.2200-2204, 2010.

[11] Y. H. P. P. Priyadarshana, K. I. H. Gunathunga, K. K. A. N. N. Perera, L. Ranathunga, P. M. Karunaratne, and T. M. Thanthriwatta, Sentiment analysis: Measuring sentiment strength of call centre conversations, in Proc. IEEE ICECCT, Mar. 2015, pp. 19.

[12] R.Srivastava and M.P.S.Bhatia, Quantifying modi edopinion strength: Afuzzy inference system for sentiment analysis, in Proc. Int. Conf. Adv. Comput., Commun. Informat., Aug. 2013, pp. 15121519..

[13] W. B. Liang, H. C. Wang, Y. A. Chu, and C. H. Wu, Emoticon recommendation in microblog using affective trajectory model, in Proc. AsiaPaci c Signal Inf. Proc. Assoc. Ann. Summit Conf. (APSIPA), Dec. 2014, pp. 15.

[14] R. Srivastava and M.P.S. Bhatia, "Quantifying modified opinion strength: A fuzzy inference system for Sentiment Analysis," in Proc. Int. Conf. Advanced in Computing, Communications and Informatics, pp.1512–1519, Aug. 2013.

[15] Y.H.P.P. Priyadarshana, K.I.H. Gunathunga, K.K.A. Nipuni N.Perera, L. Ranathunga, P.M. Karunaratne, and T.M. Thanthriwatta, "Sentiment analysis: Measuring sentiment strength of call centre conversations," in Proc. IEEE ICECCT, pp.1–9, March 2015.

[16] Q. Huang, R. Chen, X. Zheng, and Z. Dong, Deep sentiment representation based on CNN and LSTM, in Proc. Int. Conf. Green Informat., Aug. 2017, pp. 3033.

[17] M. S. Neethu and R. Rajasree, Sentiment analysis in Twitter using machine learning techniques, in Proc. 4th Int. Conf. Comput., Commun. Netw. Technol. (ICCCNT), Jul. 2013, pp. 15.

[18] J.Schmidhuber, Deeplearning in neural networks: An overview, Neural Netw., vol. 61, pp. 85117, Jan. 2015.

[19] L. Brocki and K. Marasek, Deep belief neural networks and bidirectional long-short term memory hybrid for speech recognition, Arch. Acoust., vol. 40, no. 2, pp. 191195, Jun. 2015.

[20] V. Campos, B. Jou, and X. Giró-i-Nieto, From pixels to sentiment: Fine tuning CNNsforvisualsentimentprediction, ImageVis.Comput.,vol.65, pp. 1522, Sep. 2017.

[21] Y. Lecun, L. Bottou, Y. Bengio, and P. Haffner, Gradient-based learning applied to document recognition, Proc. IEEE, vol. 86, no. 11, pp. 22782324, Nov. 1998.

[22] G. Xu, Y. Meng, X. Qiu, Z. Yu, and X. Wu, Sentiment analysis of commenttextsbasedonBiLSTM, IEEEAccess,vol.7,pp. 5152251532, 2019.
[23] A. Go, R. Bhayani, and L. Huang. "Twitter sentiment classification using distant supervision." CS224N Project Report, Stanford 1, no. 12 2009.
[24] A. Pak, and P. Paroubek. "Twitter as a corpus for sentiment analysis and opinion mining." In LREc, vol. 10, no. 2010, pp. 1320-1326. 2010.