International Journal for Innovative Engineering and Management Research
A Peer Revieved Open Access International Journal
www.ijiemr.org

COPY RIGHT

ELSEVIER
SSRN

IJIEMR Transactions, online available on 5th Jan 2023. Link :http://www.ijiemr.org/downloads.php?vol=Volume-12&issue=Issue 01

DOI: 10.48047/IJIEMR/V12/ISSUE 01/16

Title **Deep Sentiment Analysis on Twitter Data Using CNN-LSTM Approach**

Volume 12, ISSUE 01, Pages: 161-172

Paper Authors
Harika Vanam , Dr. Jeberson Retna Raj R

USE THIS BARCODE TO ACCESS YOUR ONLINE PAPER

To Secure Your Paper As Per UGC Guidelines We Are Providing A Electronic Bar Code

International Journal for Innovative Engineering and Management Research
A Peer Revieved Open Access International Journal
www.ijiemr.org

Deep Sentiment Analysis on Twitter Data Using CNN-LSTM Approach

Harika Vanam [1], Dr. Jeberson Retna Raj R [2]
[1] Computer Science & Engineering, Sathyabama Institute of Science and Technology (Deemed to be University) Chennai, India
[2] Information Technology, Sathyabama Institute of Science and Technology (Deemed to be University) Chennai, India

Abstract
Although conducting a sentiment analysis of public opinion as expressed on social media platforms such as Twitter and Facebook appear to have obvious benefits, there are still a few issues to be resolved. Hybrid methods may produce fewer sentiment errors when used on complex training data. The dependability of various methodologies is investigated in this body of work using a variety of other datasets. We compare hybrid models to single models in this work throughout different domains and datasets. Our company's learning algorithms use tweets and reviews as inputs, which perform in-depth sentiment analysis. In this study, we aim to solve the challenging problem of detecting the moods of Twitter users based on the information they publish to the platform. To do sentiment analysis, our organization uses a variety of machine learning and deep learning techniques. We pay close attention to customer feedback patterns. On the Kaggle public leaderboard, we finally apply a majority vote ensemble technique to achieve a classification accuracy of 83.58%. This strategy combines three of our most successful models into a single prediction. One method for improving sentiment analysis accuracy is using deep learning models (SVM XGBoost, and CNN-LSTM). The results of the studies showed that, when applied to datasets, the proposed model had an accuracy of 91.34%.

Keywords: Deep learning, Sentiment analysis, CNN, LSTM, Twitter

1. Introduction

Using natural language processing, text analysis, and computational linguistics, the sentiment analysis method locates and extracts subjective information from the source material [1, 2]. The use of Twitter as a forum for the expression of ideas and

opinions has increased. Every year, more tweets are being received, and this trend continues. Many difficulties arise while trying to manage this enormous volume of data. In this project, we employ Hadoop technology to process a sizable amount of data and conduct data analysis. With the robust open-source Hadoop framework, we may carry out effective operations on dispersed data among several nodes. The Map Reduce computing paradigm is an accompanying technology that produces and processes enormous data sets on a cluster in a parallel and distributed manner [3, 4]. DNNs are productive in terms of parameters and computation. With the machine and deep learning growth in the present days in leading sectors, each has advantageous and disadvantageous traits. Each layer learns to abstract the inputs [5]. CNN requires less hyperparameter tweaking [6] and monitoring, whereas LSTM requires more but computes faster. A neural network model is the LSTM model. The LSTM requires substantially more time to learn. Therefore, combining two or more procedures reduces the chance of making technical errors while maximizing the benefits of each [7].

Sentiment can be improved by 2%–6% using lexicon-based sentiment analysis. A collaborative hybrid system might perform better than a unitary one. Despite this, the integrated model performs better than the competition, depending on the task. Machine learning methods like CNN, LSTM, and SVM [8] evaluate several datasets. However, the lengths of tweets and reviews and the subjects covered differ amongst datasets. Sample sizes expressed emotions, and unrelated data all vary. As a result, it could be necessary to make specific alterations to various methods for various types of input data. Sentiment analysis is one technique that might not be suitable in every circumstance.

Twitter is a well-known social networking site where users can interact by exchanging brief messages, or "tweets," with one another. People can use this technique to communicate their ideas and emotions on various subjects. Customers and advertisers, among others, have conducted on these tweets to learn more about investigation in the current market research [9, 10].

We have also increased the precision of our sentiment analysis-based estimates due to recent developments in machine learning techniques. The "tweets" in this report will be subjected to sentiment analysis (SA) utilizing various machine

learning techniques. We attempt to classify a tweet's polarity as either positive or negative. The main emotion in a tweet should be used to establish its final classification if it contains both positive and negative emotions. We employ a dataset from Kaggle that has been crawled and classified as either positive or negative. Usernames, hashtags, and emoticons are all included in the data; these elements must be processed and put into a consistent format.

Additionally, we need to extract significant textual components like bigrams and unigrams, which are a sort of "tweet" encoding. We conduct sentiment analysis using several machine learning techniques and the features gathered. On the other hand, relying on certain models offers a low level of accuracy. As a result, we combined multiple models that were the best overall to create a model ensemble. A variety of classifiers are combined in the meta-supervised learning technique known as "deep learning" to increase prediction accuracy. The results of our experiments and observations are then presented.

2. Related Works

Building hybrid models that enhance picture recognition is simpler when a CNN model and an SVM are integrated. Convolutional networks give SVMs their properties, but SoftMax uses the first employed CNN. The information was utilized to evaluate both a hybrid model and a hybrid technique for textual sentiment categorization. According to some data, using SVMs for deep learning classification may help with image recognition [10]. The authors suggested utilizing a CNNLSTM-based deep learning system with a pre-trained embedding technique to evaluate emotions and classify evaluations or opinions as positive or negative [11]. This system would automatically learn to extract features for assessing and classifying emotions. The baseline machine learning technologies and the CNN-LSTM deep learning technology were contrasted. Their suggested course of action outperforms the datasets for applying the findings.

In [12], the authors analyzed their information and compares all the hybridization approaches. Across all datasets, combining deep learning and support vector machine (SVM) models yields superior results to any model used alone. Single models lose against hybrid models. In recent years, DL approaches have demonstrated significant potential in

sentiment analysis compared to less reliable conventional methods. When applied to feature maps, linear transforms eliminate redundant or similar features. The ghost unit is in charge of producing ghost features by deleting any qualities that are replicated or otherwise identical from each intrinsic feature. CNN needs much less hyperparameter adjusting and supervision, whereas LSTM gives better results despite requiring more time to compute. Although the integrated model's efficiency depends on the work, it consistently prevails. For sentiment analysis, hybrid deep learning models must be built using LSTM networks, CNNs, and SVMs. We assessed the accuracy and drawbacks of SVM, LSTM, and CNN using hybrid models. Combining deep learning and SVM in hybrid models can increase SA accuracy. Based on the experiment results, the proposed model's accuracy was 91.3% for dataset type 1 and 91.5% for dataset type 8.

Twitter users have two alternative ways to share their opinions and thoughts about a range of goods, subjects, and events: tweets and retweets. The authors of [13] claim that when people follow and examine tweets like these, they get user input. Instead of manually reading through millions of tweets to analyze this data, sentiment analysis is used due to the enormous volume of collected data. People sometimes use emoticons like happy, sad, and neutral faces to convey their emotions. As a result, many websites have a tonne of "raw data," or unprocessed information. The main objective is to select the techniques using ML classifiers. The study classifies fine-grained emotions expressed in tweets concerning vaccinations using machine learning and a method called "deep learning" (89974 tweets). The study's data included both tagged and unlabeled instances on some occasions. It can also recognize emoticons in tweets using machine-learning frameworks like Textblob, Vadar, and NLTK.

The authors summarized a few major research methodologies in SA investigations [14]. They investigated several approaches to machine learning, such as conventional models, deep learning models, and others. In recent years, DL models have distinguished themselves as a superior approach for processing natural languages, whereas classical models are the most effective for SA. The effectiveness of deep learning models like CNN and LSTM is evaluated on a range of datasets. To determine the

text's polarity, which can have positive and negative emotional inclinations, the author [15] considers Chinese texts for SA, which can be regarded as having two-classification difficulties. In their investigation, the text is represented by the SVM model, and the emotions exhibited in the text are examined using both the SVM and ELM models. They cleaned the data, divided it, eliminated any stop words, chose particular attributes, and finally classified it. The feature weights were computed using the TF-IDF (term frequency–inverse document frequency).

3. Materials and Methods

In this current paper, we use CNN-LSTM as the proposed model to construct the sentiment analysis of tweets that takes the Kaggle dataset and preprocess features before applying the CNN-LSTM model. The CNN model creates processes sent to the LSTM; fully connected layers come before that, as shown in Figure 1.

Dataset: Tweets and their related emotions are included in the data, presented as files with comma-separated values. The training dataset's tweet id, sentiment, and Twitter fields are filled out in a CSV file. The id of a tweet, or the unique number that identifies it, can be either 1 (positive) or 0 (negative), while sentiment can be either (negative). Although the columns in the test dataset are also in a CSV file, they are formatted as follows: tweet id, tweet. Text, emoticons, symbols, URLs, and allusions to particular people are all included in the collection. When predicting people's moods, words and emoticons perform better than URLs and allusions to specific people. As a result, references and URLs are no longer necessary. There are two million tweets in the training dataset and 800,000 in the test dataset, respectively. Additionally, several misspelled words, excessive punctuation, and various symbols are repeated throughout the same paragraph. To normalize the dataset, the tweets must be normalized.

Data Preprocessing:

Only the raw tweets collected from Twitter can be used to create

irrelevant dataset because most of the people uses social media.

Tweets contain a variety of unique attributes that must be retrieved, such as retweets, emoticons, user mentions, and so on.

International Journal for Innovative Engineering and Management Research
A Peer Revieved Open Access International Journal
www.ijiemr.org

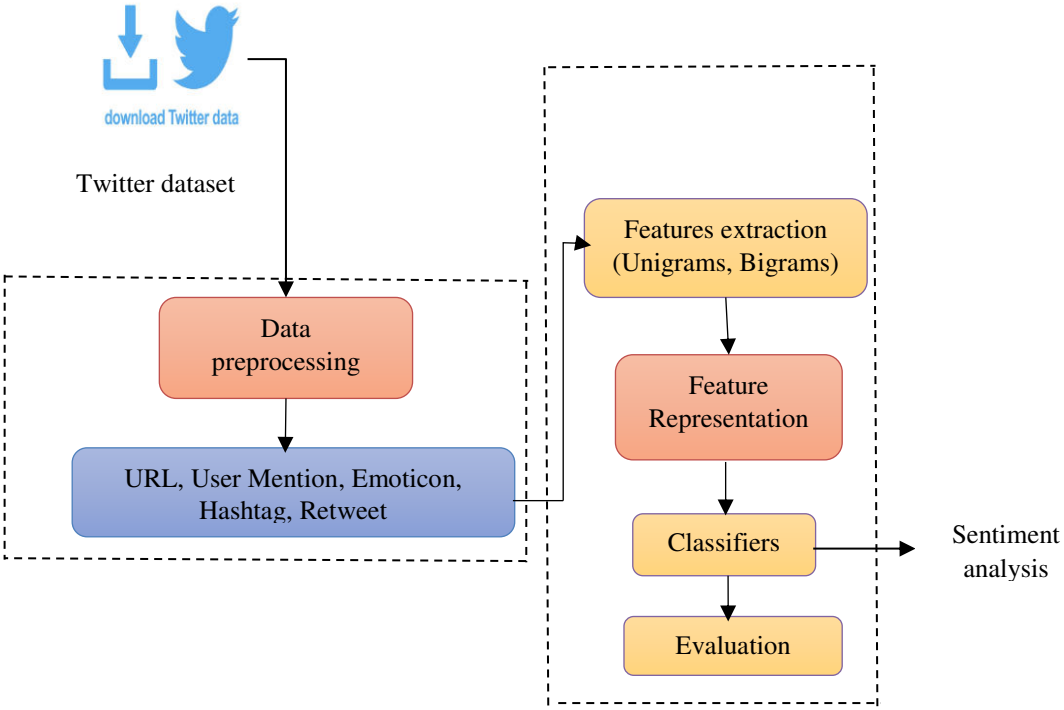

Figure 1: Block diagram of the proposed model

As a result, the raw data from Twitter must be standardized to from which many classifiers are trained with minimal effort. We used many preparation procedures to standardize the data and reduce the overall size of the dataset. As shown in the example below, we perform general pre-processing on the tweets.

• Change the uppercase characters in the tweet to lowercase characters.
• If there are two or more dots, substitute a space (.).
• Remove all spaces and quotation marks (" and ") at the end of the tweet.

Feature Extraction

Bigrams and unigrams are two distinct sorts of features that we produce using our dataset. The dataset's unigrams and bigrams are first compiled into a frequency distribution, from which the top N are chosen for further examination.

Unigrams. One of the most fundamental and frequently used criteria for text categorization of "tokens," within the text. As an illustration, we chose specific words to generate for terms in the sample. Because they do not appear frequently enough to impact categorization, the bulk

of phrases at the low end of the frequency range are regarded as noise.

Bigrams. The bigrams, which exist in the corpus in a certain sequence, are among the word combinations in the dataset. These characteristics accurately mimic real-world negation, such as the adverb "not good," for example. A total of 1954953 revealed after it was searched. Several bigrams are the noise at the low end of the frequency range, and they need to appear more frequently to impact classification. As a result, we only use the top 10,000 bigrams to generate our vocabulary.

Feature Representation

Unigrams and bigrams are extracted, and each tweet is expressed as a feature vector using either a sparse or dense vector form, depending on the classification technique.

Sparse Vector Representation (SVR)

As bigram features are used, the SVR of each tweet when just unigrams are taken into account is either 15000 or 25000. when bigrams and unigrams are considered. Similarly, in all other situations, it is zero, resulting in a sparse vector.

• **presence**. The presence feature type places a one at the indexes of unigrams and bigrams present in a tweet and a zero everywhere else in the feature vector.

• **frequency.** One approach to express a term's inverse document frequency will have '+ve' representation at the unigram (and bigram) indices to indicate tweet count if the frequency feature type is used; otherwise, it will have 0. The inverse-document frequency of the term then scales the frequency of each phrase to give terms that are important to the job larger values (*idf*).

$$idf(t) = log\left(\frac{1+n_d}{1+df(d,t)}\right) + 1 \quad (1)$$

where n_d is the total number of documents and $df(d; t)$ is the number of documents in which the term t occurs.

Dense Vector Representation

In this current study we use size of 90000 for dense vector representation, equivalent to the top 90000 words in the dataset. Because each word is assigned an integer index based on its rank (beginning with 1), the word that appears most frequently assigns 1. Following that, tweet is defined by a dense vector constructed from these indexes.

Classifiers

In this study, we classified tweets using the SVM, XGBoost, and a CNN-LSTM deep learning model.

SVM. This is an example of a binary linear classifier that does not consider probabilities. The maximum-margin hyperplane that splits a training set of points with $b_i = 1$ and $b_i = -1$ for a set of points denoted by $(a_i; b_i)$, where b denotes class and a_n denotes feature vector respectively that must be calculated. This will be done for the set of points. The hyperplane equation looks like this:

$w \cdot a - y = 0$

For further expansion of margin levels, we used term signified by γ, as

$$\max_{w,\gamma} \gamma, s.t. \forall i, \gamma \leq b_i(w.a_i + y) \quad (2)$$

in order to tell the difference between the two spots.

XGBoost. We work hard to reduce the loss function, as shown below. To employ the boosting techniques to the dataset we used ensemble for all the K models are combined as follows.

$$\hat{x}_l = \sum_{i=1}^{I} f_i(y_l), f_i \in F \quad (3)$$

The letters y_l represent the input and $\hat{x}_l$ output, respectively, while F represents the tree space. This method generates a prediction model composed of weak prediction decision trees using a gradient-boosting strategy.

The CNN-LSTM Architecture

A CNN-LSTM Neural Network Design Proposal The proposed CNN-LSTM neural network design is depicted in Figure 2. In the first layer, we used Glove 6B 300D as a word embedding model, and our embedding learned all of the terms from the Sentiment140 training datasets. The vocabulary size is 15 000, the output dimension of the embedding layers is a 30 by 300 matrix, and the maximum batch length is 30. To avoid overfitting, the embedding layer communicates with the spatial dropout layer at a rate of 0.2 times per second.

The output is fed into the first 1-dimensional CNN layer, where each filter has a size of five, and the CNN has one dimension. Then we'll define sixty-four filters. As a result, we can train 64 distinct characteristics on the network's second layer. As a result, the second neural network layer generates a 26 by 64 matrix, which is input into the 64-dimensional Bi-LSTM layer to capture long-range correlations and extract features. The outcome will be added to a dense layer of 128 by 512 neurons. To lower some

randomly selected matrix weights, the dense layer's output is reduced by a factor of 0.5. Finally, a completely linked dense layer with a sigmoid function that yields a vector as input (positive and negative) is used to predict with two units. The vector output consists of 512 neurons with a value of one.

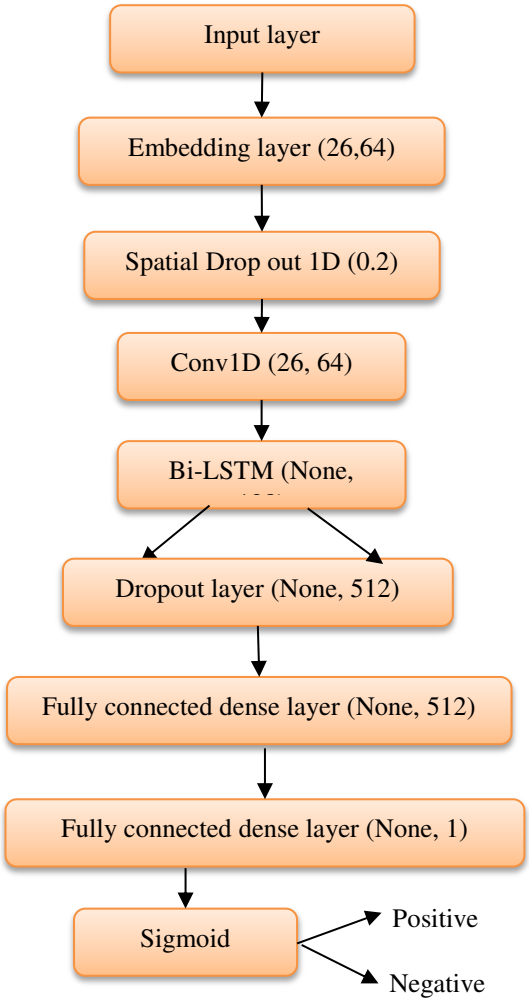

Figure 2: Block diagram of Sentiment analysis (CNN-LSTM)

We used a binary cross-entropy loss function and an Adam optimizer to build the model, with the learning rate for the Adam optimizer set to 0.001 across 10 epochs and a batch size of 1024.

Results and Discussion

Using SVM, XGBoost, and CNN-LSTM, we examine deep learning and assess it using a single preprocessed text data set. We evaluated the completeness and accuracy of the models. All the experiments are done using python 3.6 on windows 10 with an i5 processor and 8GB RAM computer. Table 1 lists all the results generated from the three models, SVM, XGBoost, and CNN-LSTM, after applying sparse vector representation.

Table 1: Results of three models SVM, XGBoost and CNN-LSTM after applying sparse vector representation

Models	Frequency	Presence
SVM	81.66	81.22
XGBoost	82.78	82.55
CNN-LSTM	91.25	91.34

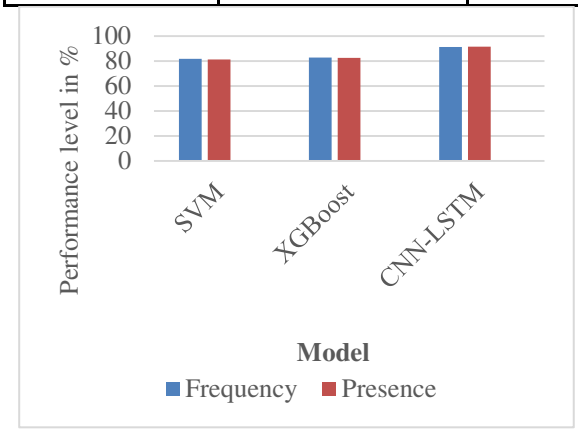

Figure 3: Performance comparison of models' frequency and presence on Kaggle public leaderboard

International Journal for Innovative Engineering and Management Research
A Peer Revieved Open Access International Journal
www.ijiemr.org

Table 2: Models used for accuracies on Kaggle dataset

Models	Accuracy in %	Execution time in sec
SVM	87.64	13.84
XGBoost	89.51	12.45
CNN-LSTM	92.58	2.345

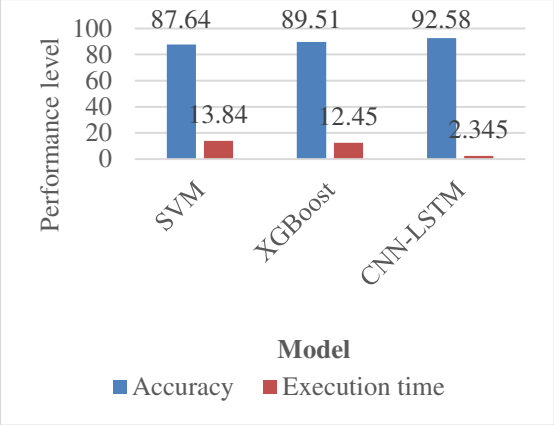

Figure 4: Performance comparison of models Accuracy and execution time on Kaggle public leaderboard

The comparison results show that our proposed CNN-LSTM model significantly outperforms many baseline approaches. The model's improved performance demonstrates this. When applied to large standard datasets for sentiment analysis, our method achieves an average accuracy of 92.58%. To extract features from the Twitter dataset, we used a CNN-LSTM model. This model includes a dense layer as well as 128 bidirectional LSTM-stacked layers. As a result, we were able to achieve higher levels of accuracy. Figures 5 and 6 show the training and validation behavior of our model's performance and loss across all ten epochs as part of the analysis. This was done to improve understanding of the data.

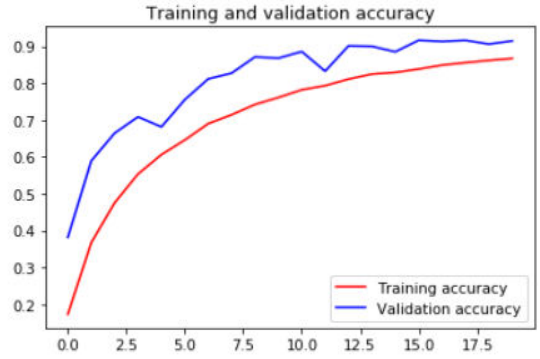

Figure 5: Training and Validation accuracy

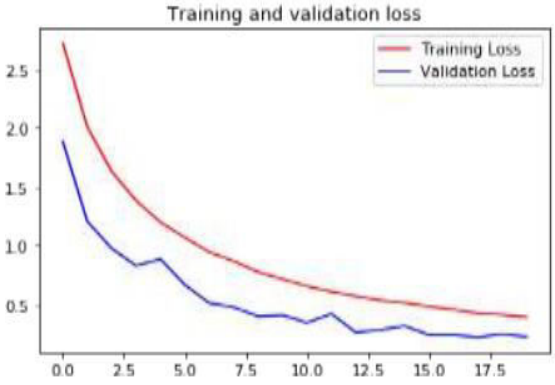

Figure 6: Training and Validation loss

Conclusion

The tweets contained various elements, such as text, symbols, URLs, hashtags, and mentions of other platform users. We do some preliminary processing on the tweets before we begin training to ensure they can be appropriately incorporated into the models. This ensures that the tweets are accurate and informative. We used a

variety of machine learning algorithms, including XGBoost, SVM, and convolutional neural networks, to classify the user's perspective on the matter as expressed in the tweet. Bigrams and unigrams were examined as two different forms of classification features, and it was found that adding bigrams to the feature vector improved accuracy. Bigrams were found to be less effective than unigrams. Finally, the proposed CNN-LSTM classifier achieved highest accuracy of 92.58% and less execution time of 2.345 sec on Kaggle public leaderboard.

References

[1] Prajapati, Yash & Khande, Rajeshree & Parasar, Akanksha. (2022). Sentiment Analysis of Emotion Detection Using Natural Language Processing. 10.1007/978-981-19-3951-8_18.

[2] Chong, Wei & Selvaretnam, Bhawani & Soon, Lay-Ki. (2014). Natural Language Processing for Sentiment Analysis: An Exploratory Analysis on Tweets. 212-217. 10.1109/ICAIET.2014.43.

[3] Pan, Wu & Ha, Li. (2014). Study of Map-Reduce over Hadoop Based Cloud Computing Environment. Applied Mechanics and Materials. 509. 175-181. 10.4028/www.scientific.net/AMM.509.175.

[4] Leu, Jenq-Shiou & Yee, Yun-Sun & Chen, Wa-Lin. (2010). Comparison of Map-Reduce and SQL on Large-Scale Data Processing. Journal of the Chinese Institute of Engineers. 36. 244 - 248. 10.1109/ISPA.2010.40.

[5] T. Sureshkumar, "Sentimental analysis of product review data using deep learning," International Journal for Research in Applied Science and Engineering Technology, vol. 9, no. 3, pp. 1028–1030, 2021.

[6] N. Bhat and B. Nethravathi, "Sentimental analysis using convolutional neural network," International Journal of Engineering Applied Sciences and Technology, vol. 5, no. 3, pp. 422–426, 2020.

[7] B. Mannepalli and K. Mannepalli, "Enhanced deep hierarchical long short-term memory and bidirectional long shortterm memory for Tamil emotional speech recognition using data augmentation and spatial features," Pertanika Journal of Science and Technology, vol. 29, no. 4, 2021

[8] R. Muralidharan, R. P. Vijai Ganesh, and T. Kanagasabapathy, "Analyzing ELearning platform reviews using sentimental evaluation with SVM classifier," Journal of Physics: Conference Series, vol. 1767, no. 1, Article ID 012012, 2021.

[9] Jagdale, Jayashree & Reha, Ali & Emmanuel, Mulimira. (2022). Sentimental Evaluation of Sensitive Tweets Using Hybrid Sentiment Analysis Model. 10.1007/978-981-16-7330-6_65.

[10] M. Fikri, T. Azhar, and T. S. Sabrila, "Perbandingan metode na¨ıve bayes dan support vector machine pada analysis sentiment twitter," SMATIKA JURNAL, vol. 10, no. 02, pp. 71–76, 2020.

[11] Tyagi, Vishu & Kumar, Ashwini & Das, Sanjoy. (2020). Sentiment Analysis on Twitter Data Using Deep Learning approach. 187-190.

10.1109/ICACCCN51052.2020.9362853.

[12] Al-Abyadh, Mohammed & Iesa, Mohamed & Azeem, Hani & Singh, Devesh & Kumar, Pardeep & Abdulamir, Mohamed & Jalali, Asadullah. (2022). Deep Sentiment Analysis of Twitter Data Using a Hybrid Ghost Convolution Neural Network Model. Computational Intelligence and Neuroscience. 2022. 10.1155/2022/6595799.

[13] Dabade, Manisha. (2021). Sentiment Analysis of Twitter Data by Using Deep Learning and Machine Learning. Turkish Journal of Computer and Mathematics Education (TURCOMAT). 12. 962-970. 10.17762/turcomat.v12i6.2375.

[14] Singhal, P. and Bhattacharyya, P., 2016. Sentiment analysis and deep learning: a survey. Center for Indian Language Technology, Indian Institute of Technology, Bombay.

[15] Zhang, X. and Zheng, X., 2016, July. Comparison of text sentiment analysis based on machine learning. In 2016 15th International Symposium on Parallel and Distributed Computing (ISPDC) (pp. 230 233). IEEE.