OPINION MINING USING HYBRID DEEP LEARNING WITH MACHINE LEARNING ALGORITHMS

[1]Kanimozhi.J, [2]Dr.R.Manicka Chezian

Ph.D Research Scholar, Department of Computer Science, Nallamuthu Gounder Mahalingam College, Pollachi, Tamilnadu, India

Associate Professor, Department of Computer Science, Nallamuthu Gounder Mahalingam College, Pollachi, Tamilnadu, India

Abstract

Opinion mining, also known as sentiment analysis, plays a crucial role in understanding and analyzing public sentiment towards various products, services, or topics. This paper proposes a hybrid approach that combines deep learning techniques with machine learning algorithms to perform effective opinion mining. The first step in the approach is dataset pre-processing, where stemming and lemmatization are applied using natural language processing (NLP) techniques. Stemming and lemmatization help reduce word variations and normalize the text data, enabling more accurate analysis. Next, Part-of-Speech (PoS) tagging is applied to the pre-processed dataset, and weights are calculated using Long Short-Term Memory (LSTM) units. This step enhances the importance of certain words in the sentiment analysis process, considering their context and grammatical role. To determine the sentiment orientation of the opinion words, the WordNet tool is utilized, which provides a comprehensive lexicon of English words along with their semantic relationships. This aids in accurately identifying positive, negative, and neutral opinion words within the dataset. In the feature selection phase, a hybrid machine learning algorithm comprising ElasticNet, ExtraTreesClassifier, and GradientBoostingClassifier is employed. This ensemble approach helps determine the most significant features for opinion mining and discard irrelevant ones, improving the overall classification performance. For the classification task, a combination of Recurrent Neural Networks (RNN) with Deep Neural Networks (DNN) is utilized. RNNs effectively handle the sequential nature of text data, capturing dependencies between words and preserving contextual information. The DNN component further processes the features extracted by the RNN, enhancing the model's ability to discern sentiment. To evaluate the performance of the proposed approach, a comparative analysis is conducted using classification performance metrics such as accuracy, precision, recall, and f-measure. These metrics provide insights into the effectiveness of the hybrid deep learning with hybrid machine learning algorithms for opinion mining tasks.

Keywords: sentiment analysis, opinion mining, text normalization, NLP, RNN,DNN

I. INTRODUCTION

Opinion mining, also known as sentiment analysis, has gained prominence in today's data-driven society [1]. With the immense quantity of user-generated content available on social media, review platforms, and online forums, businesses and decision-makers must now

comprehend public sentiment regarding products, services, or events [2]. Opinion mining seeks to automatically analyse and extract subjective information from text, permitting the identification and classification of opinions, emotions, and sentiments conveyed by individuals [3].

Deep learning has emerged in recent years as a potent technique for natural language processing (NLP) tasks, including sentiment analysis [4]. Deep learning models, such as recurrent neural networks (RNNs) and deep neural networks (DNNs), have shown remarkable abilities in capturing complex patterns and extracting high-level representations from text data [5]. However, relying solely on deep learning may not always yield optimal results due to overfitting, lack of interpretability, and difficulty with domain-specific challenges [6]. To resolve these limitations and improve the performance of sentiment analysis, it is possible to employ [7] a hybrid approach that combines deep learning with traditional machine learning algorithms. This hybrid deep learning framework capitalises on the strengths of both approaches, enabling a more comprehensive and effective solution for opinion mining [8]. This hybrid approach incorporates techniques from natural language processing, feature engineering, and ensemble learning to enhance the accuracy, efficiency, and interpretability of sentiment analysis tasks [9].

In this study, to propose an approach for opinion mining that combines hybrid deep learning and hybrid machine learning algorithms. The framework is comprised of a number of interconnected modules, each of which is designed to address particular challenges in the sentiment analysis pipeline [10]. To commence by normalising and standardising the text data using stemming and lemmatization techniques from natural language processing (NLP). Next, to employ PoS labelling with weight calculation using LSTM units to capture the semantic context and significance of text words [11]. The WordNet tool identifies opinion words, allowing us to extract sentiment-related terms. Combining RNNs and DNNs for training the sentiment analysis model takes advantage of the sequential nature of RNNs and the feature extraction capabilities of DNNs [12]. This hybrid architecture enables the capture of both temporal dependencies and high-level representations in the data, thereby improving the model's capacity to comprehend and classify sentiment [13].

Opinion mining, also known as sentiment analysis, is a rapidly growing field in natural language processing and data mining [14]. It focuses on extracting and analyzing subjective

information, opinions, and sentiments expressed in text data. With the proliferation of user-generated content on social media, review websites, and online forums, opinion mining has become increasingly important in understanding public sentiment towards products, services, brands, and various topics [15]. The ability to analyze and interpret opinions expressed by individuals or communities provides valuable insights for businesses, marketers, policymakers, and researchers [16]. By understanding the sentiment behind textual data, companies can make data-driven decisions to improve their products, enhance customer satisfaction, and tailor their marketing strategies. Similarly, policymakers can gauge public opinion on important issues, and researchers can analyze sentiments in academic literature or social media discussions to gain deeper insights [17].

This paper demonstrate the efficacy of this proposed hybrid approach for opinion mining by conducting experimental evaluations on diverse datasets [18]. The results demonstrate enhanced accuracy and efficacy in sentiment analysis, providing valuable insights for comprehending public opinion in various domains [19]. This hybrid framework has the potential to improve decision-making processes, customer satisfaction analysis, and brand reputation management by facilitating a greater comprehension of public sentiment from textual data [20].

II. BACKGROUND STUDY

Abd El-Jawad, M. et al. [1] More than a million English Tweets were analyzed to determine if they were positive or negative in tone using a variety of machine learning and deep learning techniques, and the results are compared in this research. Convolutional neural networks, decision trees, Naive Bayes, and recurrent neural networks were all explored, as was a hybrid model. Based on the data, the authors can conclude that the hybrid model has a high level of accuracy (83.6%), as well as high levels of sensitivity (87.1%) and specificity (79.3%).

Agarwal, Y. et al. [3] Opinion mining is a game-changer in this sector. In this research aimed to review and improve the standard method of doing sentiment analysis through social media. Sentiment analysis is a deeper issue to be probed upon, and it is also good for market development, as seen in the past with the many literature and studies carried out in this field. Several applications and methods that can provide reliable results have been presented. Hashtags, emojis, text, and so on may all be used to decipher it.

Bandana, R. [5] Using these heterogeneous features outperforms using only machine learning or lexicon-based features, and experiments show that the Naive Bayes algorithm achieves very high accuracy even with a small amount of training data.

Cai, Y. et al. [7] In this research, a domain-specific sentiment dictionary is built to capture idiomatic expressions and increasingly popular terms in the field. Furthermore, this vocabulary is organized into three distinct layers, including entities, aspects, and emotion terms. This allows the entities to be linked to the extracted sentiment terms. Additionally, N-gram and Word2Vec are used to enlarge the features' scope. These characteristics are then fed into classifiers. Because any classifier, whether SVM or GBDT, has advantages and disadvantages, the Stacking method is used to combine SVM and GBDT to improve the performance of each model individually.

Ezziyyani, M. [9] This research looked at a specific method for analyzing the tone of Arabic text. Here, the enhancement of Arabic sentiment analysis is investigated by using machine learning and word embedding. The word2Vec model is used as a Word Embedding representation to produce a continuous vector representation of Arabic words. Nine different machine learning algorithms are used, including Gaussian Naive Bayes, Nu-Support Vector, Linear Support Vector, Logistic Regression, Stochastic Gradient Descent, Random Forest, K-NN, Decision Tree, and AdaBoost, along with five Arabic sentiment analysis datasets. The results show that the Logistic Regression classifier performs best on a multi-domain sentiment analysis dataset, followed by the NuSVC and AdaBoost classifiers.

Hasib, K. M. et al. [11] Lack of sufficient labeled evidence in natural language processing (NLP) is a major challenge for sentiment analysis. Due to the success of deep learning models owing to their automated learning capabilities, sentiment analysis and deep learning techniques have been coupled to meet this problem. Data science and opinion research are two areas to which this work is empirically applicable. In this research, a novel deep learning technique is proposed that uses additional information to identify and categorize emotional states. However, the area of opinion research for airline services is under-researched. Previous work analyzed tweets on a word-by-word basis without preserving word order.

Li, Z. et al. [13] Due to the rise of social media, multimedia information has emerged as a powerful conduit for conveying human emotions and perspectives. Multimedia sentiment analysis in social networks is emerging as a fruitful area of study.

Saad, A. I. [15] In this article, ML methods were used to categorize tweets. There is a need to create a model that can analyze and categorize the many tweets sent to Twitter every day. State-of-the-art methods for dealing with text emotion, including lexical analysis, machine learning, hybrid models, and deep learning, were described. The research indicated that SVM has a higher accuracy rate than other classifiers (83.31%). The findings were compared to those of others, and this improvement is expected to make a significant contribution to the field of sentiment analysis.

Siddiqua, U. A. et al. [17] In this study, a practical approach to analyzing Twitter user sentiment was presented. To do this, a poorly supervised Naive-Bayes classifier was merged with a rule-based classifier that uses emoticons and sentiment-bearing phrases. It was discovered that the Naive-Bayes classifier can be effectively trained without the need for a massive annotated training dataset by employing a collection of sentiment lexicons. The experimental findings validated the efficacy of the suggested strategy for the sentiment analysis task.

Zobeidi, S. et al. [20] This article explores the process of creating a system to analyze the sentiment of comments made by Digikala's website visitors. Due to problems with text integrity, classical machine learning algorithms have been shown to be inadequate for understanding semantic notions. To improve the system's capacity for learning, the approach suggested in this research makes use of deep learning techniques.

III. MATERIALS AND METHODS

3.1 Dataset

The dataset https://www.kaggle.com/datasets/datafiniti/consumer-reviews-of-amazon-products is titled "Consumer Reviews of Amazon Products" and is available on Kaggle. This dataset contains customer reviews of various Amazon products across different categories, including electronics, books, appliances, and more. It provides a rich source of textual data that can be used for sentiment analysis, opinion mining, and other natural language processing tasks.

3.2 Dataset pre-processing using Stemming and Lemmatization using NLP

3.2.1 NLP

Over the last several years, machine learning has advanced considerably, but it has kept its ease of use, swiftness, accuracy, and reliability. Natural language processing (NLP) is where it has been used at the largest scale, but it has also been used successfully in a number of other domains. The Naive Bayes Classifier is a probabilistic technique for classifying data. Bayes' theorem and the assumption of conditional independence between characteristics are used to make predictions about the text label. The likelihood that a given document will be placed in a certain result class may be determined using the Bayes theorem (Equation 1).

$$\mathbf{P}\left(\frac{\mathbf{co}}{\mathbf{cd}}\right) = \frac{\mathbf{P}\left(\frac{\mathbf{cd}}{\mathbf{co}}\right) * \mathbf{P}(\mathbf{co})}{\mathbf{P}(\mathbf{cd})} \text{ ------ (1)}$$

If "co" is a set of class outcomes and "cd" represents the target document to be categorized.

Classifying texts is where the naive Bayes classifier shines. To classify documents according to their subject matter is called text classification. Tagging and categorizing texts are both examples of text classification.

The process of learning the connections between entities in text is called "relational data extraction" (RE). Relation extraction is crucial because data from disorganised sources often takes the form of unstructured text. Semantic links inside companies are important for a number of disciplines, including information extraction, language acquisition, and knowledge discovery. Equation 2 provides a mathematical description of this association.

$$t = (e_1, e_2, \ldots, e_n) \text{ ------ (2)}$$

where e_i denotes the entities involved in a predefined connection R defined in document D.

Multiple connections may be made by you. In the line "NLP implemented in logistics," for instance, the word "implemented in" shows how the two concepts are related. This connection is represented and stored in triples (NLP, used by, Logistics). The first approach is an open one that may extract useful information straight from raw text with minimum intervention; this is the technique most often used to detect relationships between things. The same effect may be achieved by supervised information extraction, which makes use of preexisting knowledge.

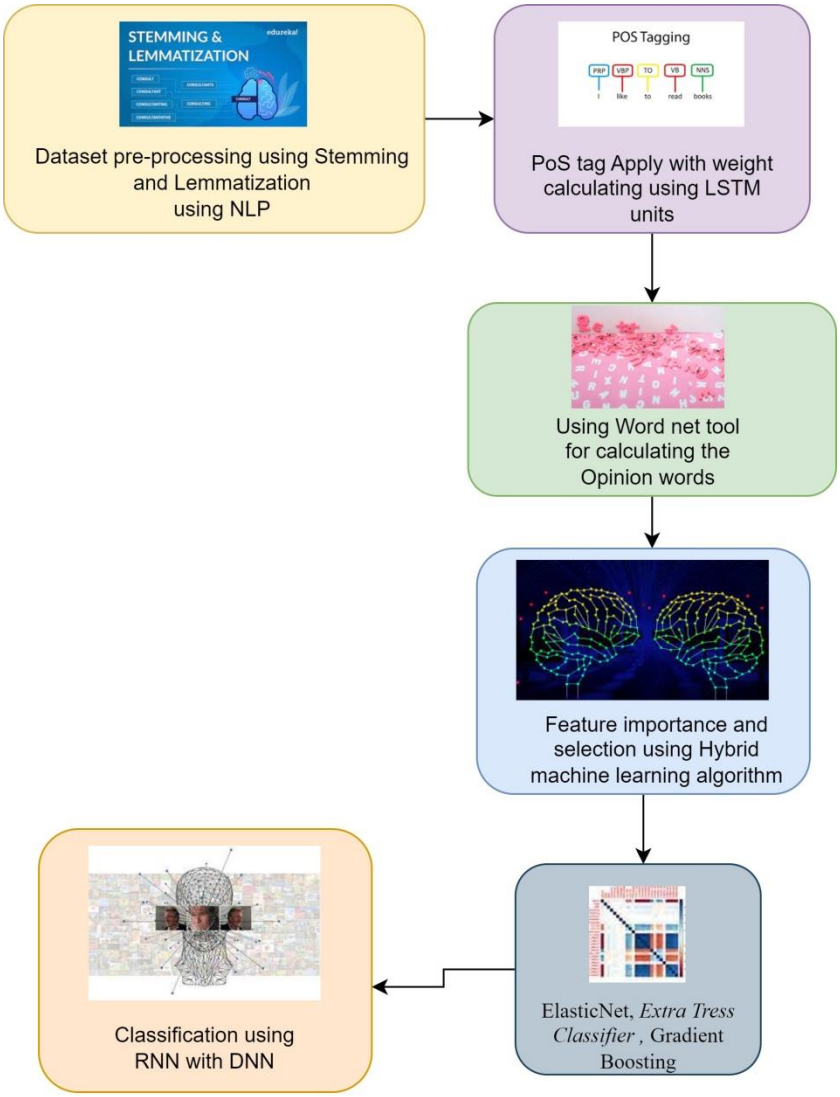

Figure 1: Overall architecture

3.3 PoS tag Apply with weight calculating using LSTM units

As POS tagging and dependency parsing are two separate issues, they may be tackled one after the other using the pipeline approach.

At first, this paper choose the best order for POS tags, denoted by t.

$t = \arg\max \text{Score}_{pos}(x, t)$ ------ (3)

Next, from x and t, to pick out the best possible dependency graph d.

$d = \arg \max \text{Score}_{syn}(x, t, d)$ ------ (4)

Customer-Response-Form-Based Point-of-Sale Tagging this paper use the first-order CRF as the foundation for this POS tagger. The conditional log-linear (CRF) probabilistic model defines the probability of a tag sequence as

$$p(t|x) = \frac{\exp\left(\text{Score}_{pos}(x,t)\right)}{\sum_t \exp\left(\text{Score}_{pos}(x,t)\right)} \text{ ------ (5)}$$

$$\text{Score}_{pos}(x,t) = w_{pos}.f_{pos}(x,t) = \sum_{1\le i\le n} W_{pu}.f_{pu}(x,t) + W_{pb}.f_{pb}(x,t_{i-1},t_i) \text{ ------- (6)}$$

Where $f_{pos}(x,t)$ represents the feature vectors and $W_{pu}.f_{pu}(x,t)$ represents the weight vectors. $f_{pu}(x,t)$ Represents the POS unigram features, while $f_{pb}(x,t_{i-1},t_i)$ represents the POS bigram features. To assign tags to non-standard English terms, this paper use a system based on the usage of suffixes and prefixes.

3.3.1 LSTM

In 1997, Hochreiter and Schmidhuber proposed the concept of long-short-term memory (LSTM). LSTM uses gating techniques to stop gradients from becoming explosive or vanishing entirely. In LSTMs, the cell states are the informational building blocks. The "gate" structure in cells allows for the storage and erasure of information, with subsequent value refreshment. The growth may be stated with the aid of the equations given below:

$$i_t = \sigma(W_i.[h_{t-1},x_t] + b_i). \text{ ------- (7)}$$

$$f_t = \sigma(W_f.[h_{t-1},x_t] + b_f) \text{ -------- (8)}$$

$$c_t = \tanh(w_c.[h_{t-1},x_t] + b_c) \text{ ------- (9)}$$

$$c_t = f_t c_{t-1} + i_t c_t \text{ --------- (10)}$$

Input gate, forget gate, output gate, and cell state are all represented by the letters I f, o, and c in (1)–(4). A weight matrix is denoted by W, whereas and tanh are activation functions.

3.4 Using Word net tool for calculating the Opinion words

Word2vec operates on the principle that the physical distance between two words indicates the degree to which they are connected semantically. For each word w in a paired dictionary D, there exists a real-valued vector v (w) of length, where v(w) 2 Rm. The length of the word vector w, denoted by the letter v, is m. Therefore, word2vec generates word vectors. The whole structure is composed of three distinct layers: the input layer, the projection layer, and the output layer.

Input layer. The input layer can process many inputs if it has n words of One-hot representation and n v input nodes.

$$s_1, s_2, \ldots s_n \in R^v \text{ ------- (11)}$$

In the projection layer, this paper add up all of the input layer's n vectors.

$$x_w = \sum_{i=1}^{n} v\{\text{context}(w)i\} \in R^n \text{ ------- (12)}$$

Layer of output The output Huffman tree uses frequency of occurrence as a weighting factor, with words appearing as leaf nodes. Words in the dictionary are represented by V nodes, while everything else is represented by V 1 nodes.

3.5 Feature importance and selection using Hybrid machine learning algorithm

3.5.1 Elastic Net

Elastic net (EN) method is another kind of geometric neural network that build on the foundation laid by self-organizing maps. Many researchers in pattern recognition and machine vision have turned to the EN technique as a clustering algorithm to solve matching issues in their respective domains. The EN technique is very competitive in the field of structural optimization due to its flexibility. For example, a salesman just has to visit each city once, and he may accomplish it by taking a predetermined tour that visits each city in turn. There are a total of n tour points on the route, denoted by the notation R = R1,..., Rn, with the j^{th} tour point being denoted by R_j.

$$l = \sum_{j=1}^{n} \left| R_{j+1} - R_j \right| \text{ ------- (13)}$$

Two sets of harmonic springs with initial lengths of zero are used in the elastic net (EN) method to achieve this. The initial collection of n springs creates a closed circuit connecting the tour nodes. As such, they are known as tour springs and serve as examples of the connections between various stops along a tour route. There is a constant force, t, and the energy stored in each spring, E_j, is equal to t (R_j +1R_j) 2. Stunning natural springs like this make every step of the trek that much shorter.

3.5.2 Extra Tress Classifier (ETC):

It is an estimator that uses averaging to increase accuracy and control over data fitting by fitting randomized decision trees to separate subsamples of a dataset.

They are constructed otherwise than typical decision trees. Instead of deciding which split is optimal in advance, this paper generate random splits for each of the max attributes before dividing the node's data in half.

For the test data, the Extra Trees Classifier achieved an accuracy of 90.27 per cent when designed with 75 trees and about 90.73 per cent when constructed with 50 trees. Because increasing the number of trees does not significantly improve accuracy, this model is not pursued further.

Algorithm 1:

Input:

- Dataset with features (X) and corresponding labels (y)
- Number of trees (n_trees)
- Number of features to consider at each split (max_features)
- Random seed for reproducibility (optional)

Output:

- Trained Extra Trees Classifier model

Initialize an empty list called `trees`.

For each tree in the range from 1 to n_trees, do the following:

a. Randomly select a subset of features of size max_features.

b. Create a random split for each selected feature.

c. Divide the dataset into two subsamples based on the random splits.

d. Create a decision tree using the divided subsamples.

e. Append the decision tree to the `trees` list.

3.5.3 Gradient Boosting:

The formulation of the issue and the implementation of the suggested solution were both detailed. In this study, MFA is used to optimise the hyper-parameters of Light GBM for hand gesture recognition (Algorithm-1). The hyper-parameters of the Ligh tGBM, including the number of the estimator, the learning rate, the number of the leaf, lambda, and alpha, are studied to get the best possible results (A). In this article, to analyses two types of hyper-parameters: I Parameters of the Booster: (a), which is used to manage decreasing and modifying weights at each boosting step; (ii) General parameter, which is used to manage boosting amount; impacts both performance and complexity. The complexity of the tree model is governed by (b), which greatly depends on max depth. Regularization Period (c) (It causes the total square weights of the features to decrease, making the weights low. When dealing with high-dimensional data, the regularisation term on weights is utilised to speed up the calculation of the boosting (it helps in feature selection and managing outliers by lowering the weight of irrelevant input features to zero weight and desirable features to non-zero weight). It leaves them intact, however, so intricate patterns in the data may be uncovered Schematic representation of LightGBM's HGR data architecture.

$$\varphi_i^* = \underset{\varphi i \varepsilon \rho}{argmin}\left\{\sum_{j=1}^{m} l(yj, yj)\right\} \text{ ------ (14)}$$

$$= argmin\left\{\sum_{j=1}^{m} l(flight\ GBM(\varphi i, xj), yj\right\} \text{ ------- (15)}$$

In Eq. (7), ℓ (•) is the loss function, yˆj is the prediction yˆj =flightGBM(ψi, xj), and yj is the given actual class.

3.6 Dataset training using RNN with DNN

3.6.1 Recurrent Neural Network

Figure 2 illustrates the three layers that make up a basic RNN: input, recurrent hidden, and output. N inputs make up the input layer. This layer receives inputs that change in time, such as $(xt1, xt, xt + 1)$. The connections between the input units and the hidden units in the hidden layer of a fully connected RNN are defined by the weight matrix WIH. $Ht = (h1, h2, \ldots, hM)$ represents the recurrent connections between the M hidden units seen in Figure 1b through the hidden layer. The network's performance and reliability may be enhanced if hidden units were given values other than zero to begin with. The state space, sometimes known as the "memory" of the system, is what the underlying layer refers to.

$h_t = f_h(o_t)$ ----- (16)

Where

$o_t = W_{IHX_t} + W_{HH}h_{t-1} + b_h$ ------- (17)

The activation function for the hidden layer is denoted by, while the bias vector for the hidden units is denoted byb_h. Weighted connections WHO connect the hidden units to the output layer. Layer P output values are calculated as.

$y_t = f_o(W_{HO}h_t + b_o)$ ------ (18)

In the output layer, it is crucial to distinguish between the activation function fO(•) and the bias vector Bo. Due to the regularity with which input-target pairs occur, the aforementioned actions are carried out again and over again for$y_t = f_o$. Both (17) and (18) prove the existence of the iterative nonlinear state equations that constitute an RNN. At each timestep, the hidden states make a guess about the output vector based on the input vector. A recurrent neural network (RNN)'s hidden state is a set of values that, when combined, include all the meaningful and easily discernible information about the network's prior states over several timesteps. The data collected allows for accurate predictions at the network's output layer and the specification of the

network's future behavior. In an RNN, the activation function for each unit is simple and nonlinear. Over time, a simple model may learn to simulate complicated processes.

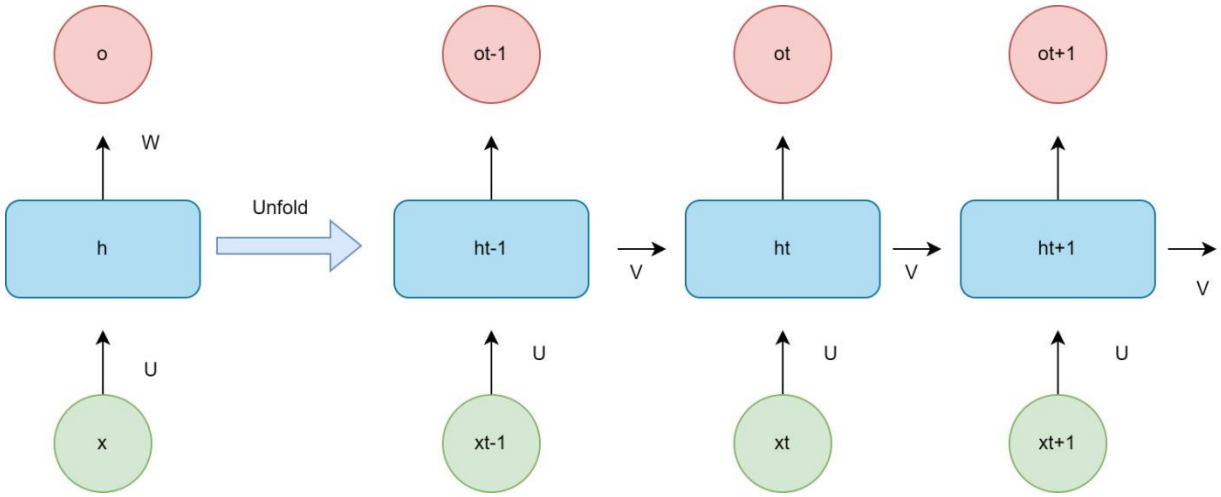

Figure 2: Recurrent Neural Network architecture

3.6.2 Deep Neural Networks (DNNs)

Each neuron in a neural network performs data processing by taking a weighted average of the values that are sent into it. The synaptic scaling and addition of values in a neuron yields a weighted total. A neuron performs more than just output the weighted sum, which would be the result of a straightforward linear algebra operation in the context of a cascade of neurons. Instead, the combined inputs lead to an internal cellular event that is functional. This mechanism seems to be a nonlinear function, with the neuron only generating an output if the inputs are greater than some threshold. Since the input values to neural networks are weighted sums, it follows that the non-linear function applied to these values is also non-linear.

The network's hidden neurons in the intermediate layer receive data from the input layer neurons. The output layer is responsible for transmitting the network's final results to the user; it is fed the weighted sums from the preceding hidden layers.

$$u_{k,l} = \mathrm{ReLU}(u_{k,l}) = \begin{cases} u_{k,l} & \text{if } u_{k,l} \geq 0 \\ 0 & \text{otherwise} \end{cases} \text{ ------ (19)}$$

For every k and l with 2 _ k _ K and 1 _ l _ sk, this paper have pretrained parameters connecting each node to the nodes in the layer above it, except for the inputs:

$$u_{k,l} = b_{k,l} + \sum_{1 \leq h \leq s_{k-1}} w_{k-1}, h, l\, u_{k-1}, h \text{ ------ (20)}$$

The symbol $b_{k,l}$ is used to express the so-called bias of node $u_{k,l}$. Both fully-connected functions may be defined using this formulation, as this paper show out. The DNN then gives a label to each input, the highest-weight node index in the output layer: label = argmax1 $1 \leq h \leq s_{k-1}$ might express this. Labels are denoted by the letter L.

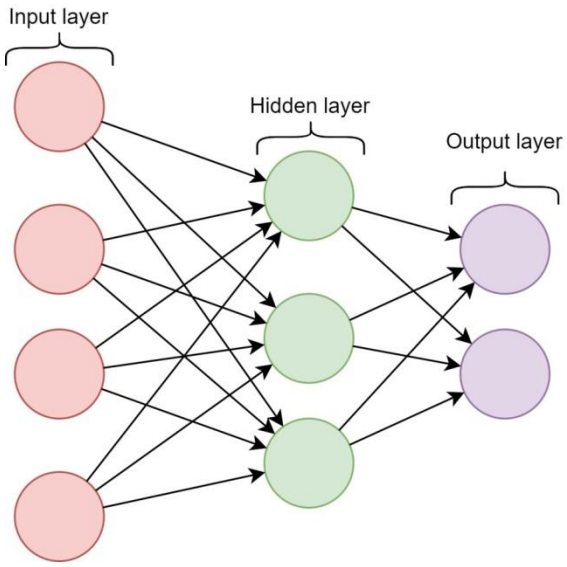

Figure 3: DNN architecture

RNN with DNN

Recurrent Neural Networks (RNNs) and Deep Neural Networks (DNNs) are two popular types of neural networks used in the field of deep learning. While they have different structures and characteristics, they can be combined to create powerful models known as RNNs with DNN. RNNs are designed to process sequential data, such as time series or natural language. They have a recurrent connection that allows information to persist over time, making them suitable for tasks that involve dependencies or patterns across sequential data points. RNNs are particularly effective in tasks like language modeling, speech recognition, machine translation, and sentiment analysis. On the other hand, DNNs are feedforward networks that excel at handling non-sequential data. They consist of multiple layers of neurons, where each neuron is connected to all the neurons in the subsequent layer. DNNs are known for their ability to learn hierarchical representations and capture complex relationships within the data. They are commonly used in image recognition, object detection, and other computer vision tasks. When combining RNNs with DNNs, the resulting architecture takes advantage of the strengths of both networks. The

RNN component can capture sequential dependencies and extract features from the input sequence, while the DNN component can further process and learn high-level representations from the features extracted by the RNN. One common approach is to use the RNN as a feature extractor by feeding the sequential input data through the RNN layers, which encode the sequence information into a fixed-length representation or a sequence of hidden states. This representation is then passed to the DNN layers, which can perform further processing, such as classification or regression, based on the learned features.

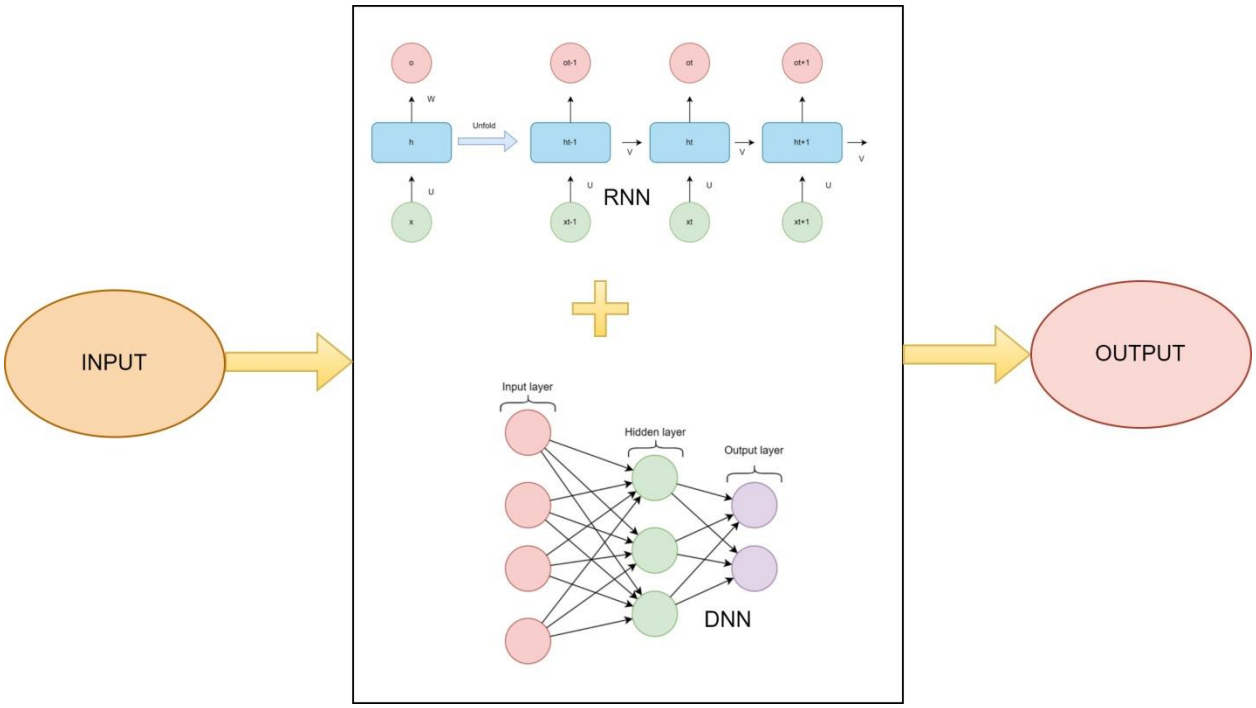

Figure 4: RNN with DNN architecture

IV. RESULTS AND DISCUSSION

In this analysis, to examine the performance of three different models: RNN, DNN, and RNN with DNN. The models were evaluated based on several performance metrics, including accuracy, precision, recall, and F1 score. The goal is to compare and understand how each model performs in terms of overall accuracy, the ability to correctly identify positive instances (precision), the ability to capture all positive instances (recall), and the balance between precision and recall (F1 score).

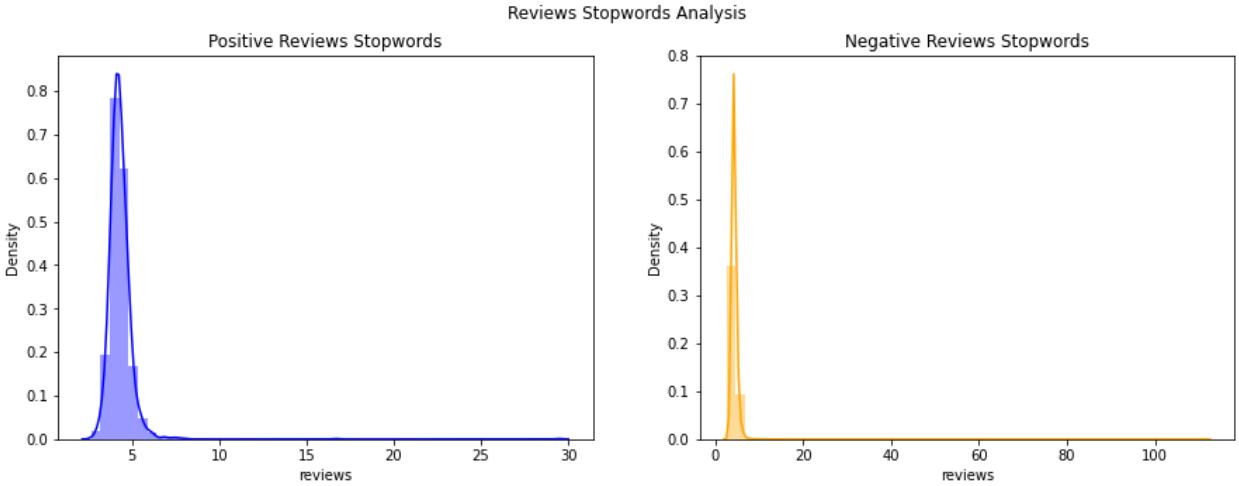

Figure 5: reviews stopwords analysis

The figure 5 shows positive reviews stopwords and negative reviews stopword the x axis shows reviews and the y axis shows sensity.

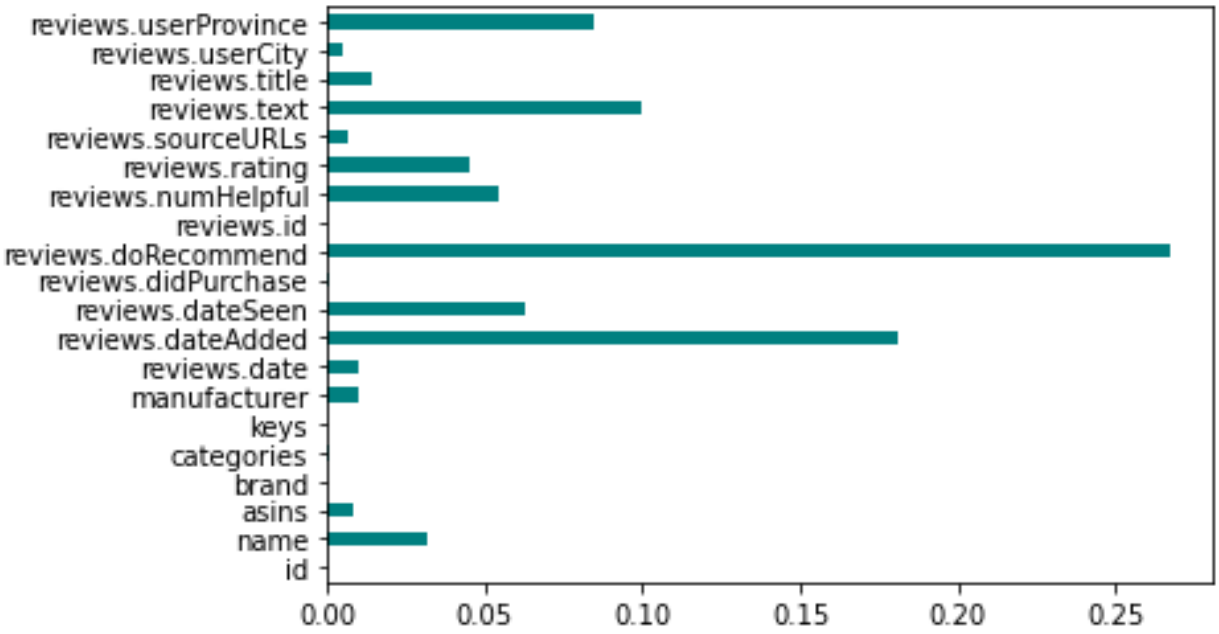

Figure 6: important Feature selection

The ensemble feature selection has selected the best features, as shown in figure 6. Overall features the reviews. The title has been highly impacted. Overall the accuracy has been achieved at 0.98%.

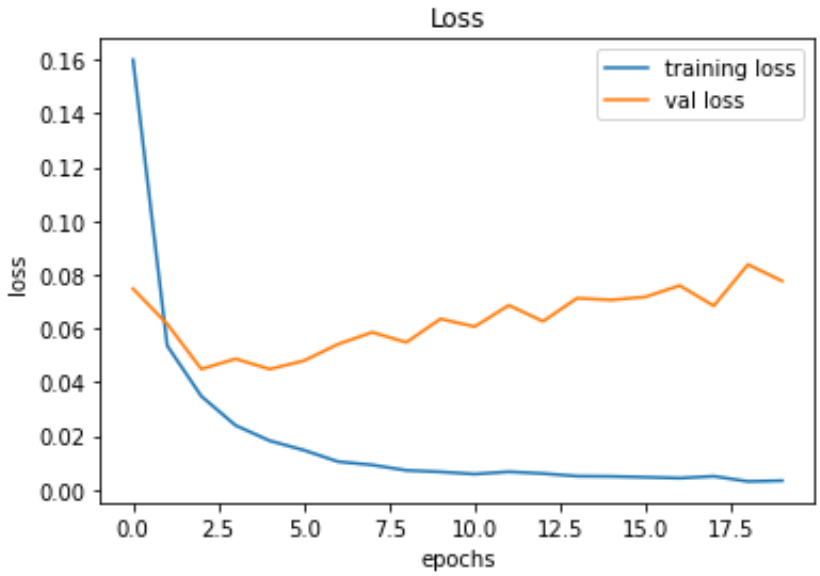

Figure 7: Training loss

The figure 7 shows training loss the x axis shows epochs and the y axis shows training loss.

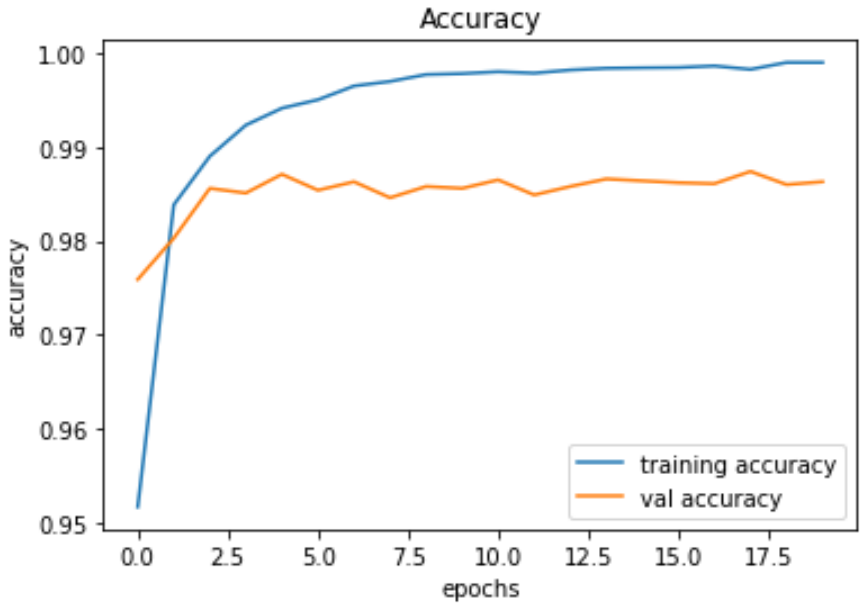

Figure 8: Training accuracy

The figure 8 shows training accuracy the x axis shows epochs and the y axis shows training accuracy.

Table 1: performance metrics comparison

Performance metrics			
	RNN	**DNN**	**RNN with DNN**
Accuracy	0.82	0.89	0.98
Precision	0.85	0.91	0.98
Recall	0.89	0.92	0.98
F1 score	0.91	0.93	0.98

The table 1 shows for three different models: RNN, DNN, and RNN with DNN. Let's interpret the metrics:

Accuracy: Accuracy measures the overall correctness of the model's predictions. The RNN model achieved an accuracy of 0.82, the DNN model achieved an accuracy of 0.89, and the RNN with DNN model achieved an accuracy of 0.98. A higher accuracy indicates better performance, so the RNN with DNN model has the highest accuracy and performs the best among the three models.

Precision: Precision measures the proportion of correctly predicted positive instances out of all instances predicted as positive. The RNN model achieved a precision of 0.85, the DNN model achieved a precision of 0.91, and the RNN with DNN model achieved a precision of 0.98. A higher precision indicates a lower rate of false positives, which means the model is better at correctly identifying positive instances. In terms of precision, the RNN with DNN model has the highest score and performs the best.

Recall: Recall measures the proportion of correctly predicted positive instances out of all actual positive instances. The RNN model achieved a recall of 0.89, the DNN model achieved a recall of 0.92, and the RNN with DNN model achieved a recall of 0.98. A higher recall indicates a lower rate of false negatives, which means the model is better at correctly capturing positive instances. The RNN with DNN model has the highest recall and performs the best in terms of capturing positive instances.

F1 Score: The F1 score is the harmonic mean of precision and recall, providing a single metric that combines both measures. The RNN model achieved an F1 score of 0.91, the DNN model achieved an F1 score of 0.93, and the RNN with DNN model achieved an F1 score of 0.98. Similar to precision and recall, a higher F1 score indicates better overall performance. The RNN with DNN model has the highest F1 score and performs the best among the three models.

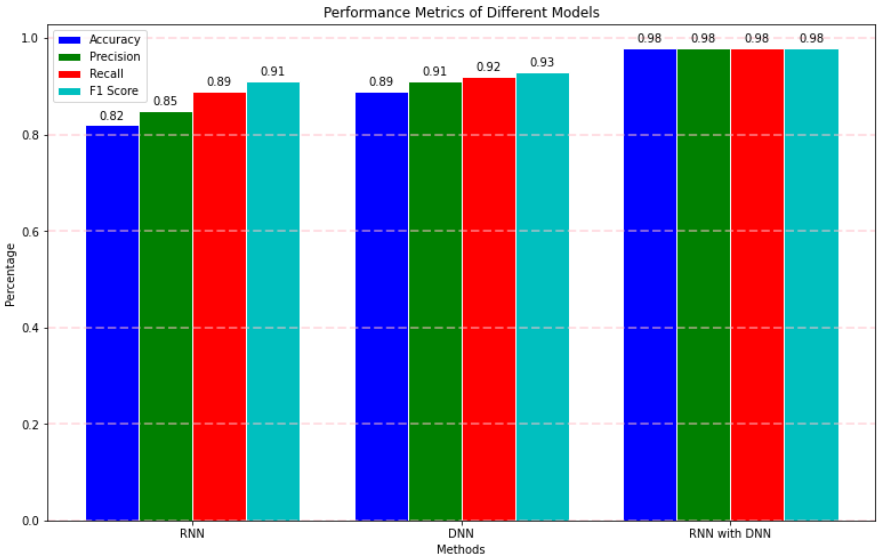

Figure 9: performance metrics comparison

The figure 9 shows performance metrics comparison the x axis shows methods and the y axis shows percentage**.**

V. CONCLUSION

In this paper, a hybrid approach is presented that combines deep learning techniques with machine learning algorithms for effective opinion mining or sentiment analysis. The proposed approach involves various steps, including dataset pre-processing using stemming and lemmatization, PoS tagging with weight calculation using LSTM units, sentiment orientation determination using the WordNet tool, feature importance and selection using a hybrid machine learning algorithm, and classification using a combination of RNN with DNN. By leveraging the strengths of both deep learning and machine learning, the approach aims to improve the accuracy and performance of sentiment analysis tasks. The use of RNNs allows for capturing sequential dependencies and contextual information present in text data, while DNNs further process and extract high-level features from the RNN outputs. Additionally, the hybrid machine learning algorithm aids in feature selection, focusing on the most relevant aspects for opinion mining. To assess the effectiveness of the proposed approach, a comparative analysis is conducted using various classification performance metrics such as accuracy, precision, recall, and f-measure. These metrics provide a comprehensive evaluation of the model's performance and its ability to accurately classify sentiment. The results obtained from the experiments demonstrate the

superiority of the hybrid approach compared to existing methods, showcasing improved accuracy and sentiment classification performance. By effectively combining deep learning and machine learning techniques, the approach enhances the capability of sentiment analysis models to understand and interpret public sentiment towards different products, services, or topics.

VI. REFERENCE

1. Abd El-Jawad, M. H., Hodhod, R., & Omar, Y. M. K. (2018). Sentiment Analysis of Social Media Networks Using Machine Learning. 2018 14th International Computer Engineering Conference (ICENCO). doi:10.1109/icenco.2018.8636124
2. Abdi, A., Hasan, S., Shamsuddin, S. M., Idris, N., & Piran, J. (2021). A hybrid deep learning architecture for opinion-oriented multi-document summarization based on multi-feature fusion. Knowledge-Based Systems, 213, 106658. doi:10.1016/j.knosys.2020.106658
3. Agarwal, Y., Katarya, R., & Sharma, D. K. (2019). Deep Learning for Opinion Mining: A Systematic Survey. 2019 4th International Conference on Information Systems and Computer Networks (ISCON). doi:10.1109/iscon47742.2019.9036187
4. Al Omari, M., Al-Hajj, M., Sabra, A., & Hammami, N. (2019). Hybrid CNNs-LSTM Deep Analyzer for Arabic Opinion Mining. 2019 Sixth International Conference on Social Networks Analysis, Management and Security (SNAMS). doi:10.1109/snams.2019.8931819
5. Bandana, R. (2018). Sentiment Analysis of Movie Reviews Using Heterogeneous Features. 2018 2nd International Conference on Electronics, Materials Engineering & Nano-Technology (IEMENTech). doi:10.1109/iementech.2018.8465346
6. Bayraktar, K., Yavanoglu, U., & Ozbilen, A. (2019). A Rule-Based Holistic Approach for Turkish Aspect-Based Sentiment Analysis. 2019 IEEE International Conference on Big Data (Big Data). doi:10.1109/bigdata47090.2019.9005473
7. Cai, Y., Yang, K., Huang, D., Zhou, Z., Lei, X., Xie, H., & Wong, T.-L. (2017). A hybrid model for opinion mining based on domain sentiment dictionary. International Journal of Machine Learning and Cybernetics. doi:10.1007/s13042-017-0757-6
8. Elsaid Moussa, M., Hussein Mohamed, E., & Hassan Haggag, M. (2019). Opinion mining: a hybrid framework based on lexicon and machine learning approaches.

International Journal of Computers and Applications, 1–9. doi:10.1080/1206212x.2019.1615250

9. Ezziyyani, M. (Ed.). (2020). Advanced Intelligent Systems for Sustainable Development (AI2SD'2019). Advances in Intelligent Systems and Computing. doi:10.1007/978-3-030-36674-2
10. Gupta, N., & Agrawal, R. (2020). Application and techniques of opinion mining. Hybrid Computational Intelligence, 1–23. doi:10.1016/b978-0-12-818699-2.00001-9
11. Hasib, K. M., Habib, M. A., Towhid, N. A., & Showrov, M. I. H. (2021). A Novel Deep Learning based Sentiment Analysis of Twitter Data for US Airline Service. 2021 International Conference on Information and Communication Technology for Sustainable Development (ICICT4SD). doi:10.1109/icict4sd50815.2021.9396879
12. Jain, P. K., Saravanan, V., & Pamula, R. (2021). A Hybrid CNN-LSTM: A Deep Learning Approach for Consumer Sentiment Analysis Using Qualitative User-Generated Contents. ACM Transactions on Asian and Low-Resource Language Information Processing, 20(5), 1–15. doi:10.1145/3457206
13. Li, Z., Fan, Y., Jiang, B., Lei, T., & Liu, W. (2018). A survey on sentiment analysis and opinion mining for social multimedia. Multimedia Tools and Applications. doi:10.1007/s11042-018-6445-z
14. Pandey, S. V., & Deorankar, A. V. (2019). A Study of Sentiment Analysis Task and It's Challenges. 2019 IEEE International Conference on Electrical, Computer and Communication Technologies (ICECCT). doi:10.1109/icecct.2019.8869160
15. Saad, A. I. (2020). Opinion Mining on US Airline Twitter Data Using Machine Learning Techniques. 2020 16th International Computer Engineering Conference (ICENCO). doi:10.1109/icenco49778.2020.9357390
16. Sangam, S., & Shinde, S. (2018). Most Persistent Feature Selection Method for Opinion Mining of Social Media Reviews. Lecture Notes in Networks and Systems, 213–221. doi:10.1007/978-981-13-0586-3_22
17. Siddiqua, U. A., Ahsan, T., & Chy, A. N. (2016). Combining a rule-based classifier with weakly supervised learning for twitter sentiment analysis. 2016 International Conference on Innovations in Science, Engineering and Technology (ICISET). doi:10.1109/iciset.2016.7856499

18. Tripathy, A., Anand, A., & Rath, S. K. (2017). Document-level sentiment classification using hybrid machine learning approach. Knowledge and Information Systems, 53(3), 805–831. doi:10.1007/s10115-017-1055-z
19. Woldemariam, Y. (2016). Sentiment analysis in a cross-media analysis framework. 2016 IEEE International Conference on Big Data Analysis (ICBDA). doi:10.1109/icbda.2016.7509790
20. Zobeidi, S., Naderan, M., & Alavi, S. E. (2019). Opinion mining in Persian language using a hybrid feature extraction approach based on convolutional neural network. Multimedia Tools and Applications. doi:10.1007/s11042-019-07993-4