International Journal of
INTELLIGENT SYSTEMS AND APPLICATIONS IN ENGINEERING
IJISAE
ISSN:2147-6799 www.ijisae.org Original Research Paper

Sentiment Analytics on Sarcasm Detection Using Bi-LSTM-1DCNN Model for Fake News Detection

Dr. M. Anusha[1], R. Leelavathi[2]

Submitted: 15/09/2023 **Revised**: 28/10/2023 **Accepted**: 17/11/2023

Abstract: Finding expressive attitudes and states in text is the process of sentiment analysis, also referred to as opinion analysis. In this study, we offer a thorough investigation of sentiment analysis using the sophisticated fusion architecture of 1D convolutional neural networks (1DCNN) and Bidirectional Long Short-Term Memory (BI-LSTM). In order to understand the performance and ramifications of Glove and Word2Vec, two of the most notable word embeddings in the context of sentiment determination, they are compared in this investigation. The study encompasses a comprehensive assessment of critical performance indicators, mainly recall, accuracy, precision, and F1-score. Preliminary results conspicuously reveal a marked superiority of the BiLSTM+1DCNN model interfaced with Glove embeddings when juxtaposed against the Word2Vec variant. More specifically, the Glove-integrated model exhibited commendable precision values of 82% for positive sentiments and 78% for negative sentiments. Concurrently, recall metrics stood at 79% for positive and 80% for negative sentiments, leading to an impressive F1-score of 81% Positive sentiments and 82% for negative sentiment classes. This augmented performance is attributable to Glove's intricate semantic captures, owed to its training on extensive and diverse text corpora, thereby ensuring richer contextual information retrieval. In the quest to offer a visual and intuitive understanding, the research presents a suite of graphical representations: the accuracy graph elucidating model performance progression over epochs, the loss graph signifying the model's error rate, and the Receiver Operating Characteristic (ROC) graph portraying the model's capability in distinguishing sentiment polarities. A specialized comparison graph crystallizes the performance disparities between the two embeddings, fortifying the argument in favour of Glove's supremacy. Conclusively, the research underscores the paramountcy of selecting the apt embedding, with the Glove-based BiLSTM+1DCNN model emerging as the frontrunner for sentiment analytics tasks, owing to its impeccable balance between precision and recall, culminating in a laudable 83% accuracy.

Keywords**:** *Deep Learning, Sentiment Analytics, Fake News, Glove, Accuracy, Bi-LSTM-1DCNN.*

1. Introduction

In an era characterized by an overwhelming influx of information through various media channels, discerning the veracity of news and information has become an increasingly complex challenge. The proliferation of fake news, often disseminated with the intent to deceive or manipulate, has highlighted the critical need for advanced tools and methodologies to separate fact from fiction. Addressing this challenge requires not only the identification of false content but also a deeper understanding of the emotions and sentiments underlying news articles and reports. In this context, we propose the integration of sentiment analysis within fake news detection systems emerges as a pivotal avenue to enhance their accuracy and effectiveness. Finding attitudes and emotional states in text is a method called sentiment analysis, sometimes referred to as opinion mining. This analysis can encompass a wide spectrum of sentiments, including positive, negative, neutral, and more nuanced emotions such as anger, joy, fear, and sadness. A more comprehensive comprehension of the news articles can be obtained by combining sentiment analysis with false news detection. If a fake news article is found to evoke unusually strong emotional reactions or inconsistencies in its emotional cues, this could serve as an additional indicator of its dubious nature. Likewise, a sentiment analysis approach can assist in flagging content that appeals to emotion rather than presenting objective facts, regardless of its truthfulness.

Deep learning (DL), a kind of ML inspired by the neural architecture of the human brain, has great potential to advance both sentiment analysis and fake news identification. DL models, particularly recurrent neural networks (RNNs) and transformers, have demonstrated unparalleled prowess in understanding complex patterns within sequential data such as text. These models can capture intricate dependencies between words, phrases, and even entire documents, thus equipping them to decipher not only the explicit textual content but also the underlying emotional nuances. By training these models

[1]*Assistant Professor, PG & Research Department of Computer Science, National College (Autonomous), Bharathidasan University, Tamilnadu, India.*
E-mail: anusha260505@gmail.com
[2]*Ph.D Research Scholar, PG & Research Department of Computer Science, National College (Autonomous), Bharathidasan University, Tamilnadu, India.*
E-mail: leelavathi4794@gmail.com
**Corresponding Author: - E-mail: leelavathi4794@gmail.com*

on vast datasets comprising legitimate and fake news articles, they can learn to discern subtle linguistic cues that often elude conventional rule-based approaches.

Therefore, the synergy between sentiment analysis and fake news detection represents a cutting-edge direction in the realm of information authenticity. By harnessing the power of DL, we aim to not only identify the linguistic markers of fake news but also to unravel the emotional tapestry woven into these narratives. This fusion promises to elevate the accuracy of fake news detection systems, providing a more robust defense against the insidious spread of misinformation. As technology continues to reshape the information landscape, our pursuit of intelligent tools for truth delineation stands as an essential safeguard for informed decision-making and the preservation of societal trust.

1.1 Aim and objective

This study uses DL algorithms to identify bogus news via social media. In the process of detecting the fake news, two differents models (sarcasm identification and sentiment analysis) will be integrated in order to achieve a better perfomence.

The following are the goals of this study proposal:

- Creating an architecture and methodology for sentiment analysis based on DL to identify false information in social media streams.
- To integrate a sentiment detection model that explores sentiment of each news in order to detect sentiment of non-sarcastic social media comments.
- Need to perform various performance evaluation metrics to validate the training dataset classification.
- To check and compare the accuracy of the current models and framework with streaming datasets.

1.2 Problem statement

In the contemporary digital landscape, combating the proliferation of fake news has become an imperative. This research aims to enhance fake news detection by leveraging DL and sentiment analysis techniques. The objectives encompass designing a DL -based framework for sentiment analysis in fake news detection. This framework intends to capture intricate linguistic and emotional patterns present in news articles, facilitating the differentiation between authentic and fabricated content. Furthermore, the integration of sentiment detection within the fake news detection system seeks to extend this analysis to non-sarcastic social media comments, enabling a more comprehensive understanding of context. (Anusha M and Leelavathi R, et al., 2023) Proposed by a Sarcasm detection using Sentiment analytics for DL techniques have been applied in fake news detection. However, one of the significant drawbacks of these approaches is the inability to detect sarcasm, which is a common way of for people to express opinions in social media. Furthermore, in previous work sentiment analytics has been carried out using Glove word embedding along Bi-LSTM for detecting the fake news provided with nominal accuracy of 81% without considering the sarcasm detection-based attributes. Hence in this study, we propose a sarcasm detection model to provide better performance in terms of accuracy by make use of single DL architecture (Bi-LSTM) along with Glove itself with minimal computational workload.

The proposed research also emphasizes comprehensive performance evaluation metrics to validate the classification accuracy of the training dataset, ensuring its representational diversity. To gauge the effectiveness of the developed framework, a comparative analysis will be conducted against streaming datasets. This analysis will provide an empirical measure of the framework's accuracy and robustness in identifying fake news. By achieving these objectives, this research aspires to contribute to the advancement of fake news detection methodologies, offering a more nuanced and accurate approach that aligns with the evolving complexities of digital information sharing.

2. Related Work

The pervasive dissemination of fake news has prompted researchers to explore innovative approaches for its detection. Recent studies have shown a growing interest in integrating sentiment analysis into fake news detection systems, harnessing the power of DL techniques for enhanced accuracy and contextual understanding.

The development of transformer-based models in particular has allowed DL to completely change a number of natural language processing applications. Bidirectional Encoder Representations from Transformers, or BERT, and its offspring have demonstrated remarkable abilities to comprehend context and semantics in textual input. Researchers have begun leveraging these models to enhance fake news detection. (Kaur and Sharma, 2023) introduced a BERT-based framework that combines sentiment analysis with traditional features, demonstrating significant improvement in accuracy and F1-score compared to conventional methods. This signifies the potential of transformer models in extracting not only linguistic but also emotional cues from news articles.

Sentiment analysis has proven instrumental in gauging the emotional tone of content, providing an additional layer of insight. (Chen, Zhang and Lou, 2020) proposed a hybrid model that combines LSTM networks with attention mechanisms to perform sentiment analysis on news articles. This analysis contributes to the detection process by discerning content that evokes strong emotions, often indicative of sensationalism or misinformation. Integrating this sentiment-awareness augments the overall fake news detection accuracy, as demonstrated by their experimental results.

Nonetheless, the integration of sentiment analysis into fake news detection is not without challenges. The need for substantial labeled datasets encompassing both sentiment labels and news authenticity labels poses a significant hurdle. Datasets with accurate sentiment annotations are essential to train models effectively. (Dufraisse *et al.*, 2023) highlighted the importance of such datasets in their study, where they curated a comprehensive dataset with labeled sentiments for news articles. This dataset enabled the training of a sentiment-aware fake news detection model that showcased enhanced performance compared to sentiment-agnostic counterparts.

Performance evaluation remains paramount in gauging the efficacy of these integrated models. A diverse array of metrics is employed to validate their efficiency in identifying fake news while accounting for sentiment nuances. Precision, recall, F1-score, & AUC-ROC (Area Under the **ROC** Curve) have been common metrics of choice. (Song and Huang, 2021) conducted an extensive evaluation of their DL -based sentiment-aware model using these metrics. Their findings reinforced the model's capability to effectively distinguish between true and fabricated news while considering the emotional context.

The paper (Ju *et al.*, 2021) introduces a multi-modal DL architecture that employs CNNs for image analysis and RNNs for text analysis. This fusion of modalities allows the model to simultaneously process textual content and accompanying images, capturing potential inconsistencies between text and visual sentiment. This approach is grounded in the understanding that fake news often manipulates emotional triggers to enhance its impact, and such manipulation could manifest through disparities between the sentiments conveyed in the text and visuals.

Therefore, recent research reflects a growing interest in merging sentiment analysis with DL for fake news detection. Transformer-based models provide a robust foundation for contextual comprehension, while sentiment analysis augments detection accuracy by capturing emotional nuances. However, challenges persist, including the availability of labeled datasets and the selection of appropriate evaluation metrics. As the digital landscape evolves, these integrated approaches hold promise in countering the persistent threat of fake news and fostering a more informed society.

Table: Summary of Literature Review on Sentiment Analysis for Fake News Detection using DL

Combining BERT's advanced contextual language representation with traditional, hand-crafted features offers a multi-pronged approach to sentiment analysis.

Paper Title and Year	Approach	Key Findings
(Kaur and Sharma, 2023)	BERT-based framework for sentiment analysis integrated with traditional features	Improved accuracy and F1-score in fake news detection compared to conventional methods.
(Dufraisse *et al.*, 2023)	Sentiment-aware fake news detection model using DL	Dataset with sentiment annotations vital for effective model training.
(Song & Huang, 2021)	Sentiment-aware DL model for fake news detection	Comprehensive evaluation using multiple metrics demonstrates effective differentiation between real and fake news.
(Ju *et al.*, 2021)	Multimodal strategy: Sentiment analysis in text combined with sentiment analysis in visual	Multi-modal approach combining textual and visual sentiment analysis improves detection accuracy.

Despite the recent advancements in integrating sentiment analysis with DL for fake news detection, there exists a notable research gap regarding the exploration of cross-modal sentiment analysis. Existing studies often focus on textual sentiment or visual sentiment independently. A potential research avenue lies in developing models that effectively fuse and analyze sentiments across both text and accompanying images, capitalizing on the synergistic information present in multi-modal data. Such an approach could enhance the accuracy and comprehensiveness of fake news detection systems, bridging the divide between textual and visual cues and paving the way for more nuanced authenticity assessment.

3. Methodology

The chapter on methodology explores the technological foundations of the suggested strategy for identifying false news. This chapter introduces the utilization of two distinct DL techniques: BiLSTM networks and 1DCNN. These methods are employed to extract intricate patterns and relationships within textual data, enabling a nuanced analysis of linguistic cues and emotional nuances that underlie news articles. By leveraging the capabilities of both BiLSTM and 1DCNN, The goal of the project is to improve the sentiment analysis-based false news detection system's precision and dependability.

The mathematical equations for forward LSTM are:

i_t = sigmoid (W_i * x_t + U_i * h_{t-1} + b_i)

f_t = sigmoid(W_f * x_t + U_f * h_{t-1} + b_f)

o_t = sigmoid(W_o * x_t + U_o * h_{t-1} + b_o)

c_t = f_t ⊙ c_{t-1} + i_t ⊙ tanh(W_c * x_t + U_c * h_{t-1} + b_c)

h_t = o_t ⊙ tanh(c_t)

and for backward LSTM are:

i'_t = sigmoid (W'_i * x'_t + U'_i * h'_{t+1} + b'_i)

f'_t = sigmoid (W'_f * x'_t + U'_f * h'_{t+1} + b'_f)

h'_t = o'_t ⊙ tanh(c'_t)

where:

c'_t = f'_t ⊙ c'_{t+1} + i'_t ⊙ tanh (W'_c * x'_t + U'_c * h'_{t+1} + b'_c)

o'_t = sigmoid (W'_o * x'_t + U'_o * h'_{t+1} + b'_o)

x_t and x'_t is the input at time step t in forward and backward directions respectively

h_{t-1} and h'_{t+1} are the hidden states from the previous time step in forward and backward directions respectively

i_t, f_t, and o_t is the input, forget, and output gates for forward LSTM

c_t is the cell state for forward LSTM

h_t is the hidden state for forward LSTM

i'_t, f'_t, and o'_t is the input, forget, and output gates for backward LSTM

c'_t is the cell state for backward LSTM

h'_t is the hidden state for backward LSTM

sigmoid is the sigmoid activation function

tanh is the hyperbolic tangent activation function

⊙ denotes element-wise multiplication

W_i, W_f, W_o, W_c, U_i, U_f, U_o, U_c, b_i, b_f, b_o, b_c are the learnable weight matrices and bias vectors for the LSTM

1D Convolution Operation:

1. 1D Convolution Operation:

To create a feature map, the convolution operation entails swiping a filter—also referred to as a kernel—over the input sequence. For feedback

sequence X of length n and filter F of length k:

$$Z[i] = \sum_{j=0}^{k-1} X[i+j] \times F[j]$$

where Z is the convolution result (feature map) and i iterates from 0 to $n-k$.

2. Multiple Filters:

In practice, a 1D CNN layer often uses multiple filters to capture different types of features in the input sequence. Let’s say we have m filters F_1, F2, ..., Fm. Each filter produces its own feature map:

$$Z_p[i] = \sum_{j=0}^{k-1} X[i+j] \times F_p[j]$$

for $p = 1, 2, \ldots, m$.

As a result, form filters, we get *m* feature maps

1. ReLU Activation:

ReLU (Rectified LinearUnit) is an element-wise activation function defined as:

$$ReLU(x) = max(0,x)$$

Applying ReLU to our feature maps, we get:

$$Ap[i] = ReLU\ (Zp[i])$$

for $p = 1, 2, ..., m$.

In order to learn local features, the 1DCNN layers then apply a sliding window across the series of hidden states, h_t: Where k is the window size, W_z & b_z are the weight matrix & bias vector for the 1DCNN layer, & f is a non-linear activation function, we get z_i = f (W_z * [h_i, h {i+1},..., h_{i+k-1}] + b_z). Dimension p is a feature vector that is the output, z_i. In order to accomplish classification, the feature vectors z_i are finally concatenated and supplied to a dense layer, which is followed by a softmax activation function: y = softmax (W_y * [z_1, z_2,..., z_{T-k+1}] + b_y), where y is the output probability distribution across the classes and W_y & b_y are the weight matrix & bias vector for the dense layer.

Glove mathematical model –

The initial stage in the GloVe word embedding model is to create a co-occurrence matrix X, where X(i,j) shows the co-occurrence count of word i with word j in a large corpus.

Co-occurrence Matrix:

In the first step, a **co-occurrence matrix X** is constructed. The frequency with which word i occurs in relation to word j throughout the entire corpus is indicated by each entry X(i, j) in this matrix.

Glove objective Function:

Finding word vectors that accurately represent the word co-occurrence probabilities is the aim of GloVe. By reducing the difference between the logarithm of word probabilities and the dot product of word vectors, this is accomplished.

The objective function *J* for GloVe is:

$$\mathbf{J} = \sum_{\mathbf{i}} \sum_{\mathbf{j}} \mathbf{f}(\mathbf{X}(\mathbf{i}, \mathbf{j})) \times (w^T w_{\mathbf{j}} + b_{\mathbf{i}} + b_{\mathbf{j}} - \log(\mathbf{X}(\mathbf{i}, \mathbf{j})))^2$$

Where:

The words "i" and "j" have word vectors that are wi and wj, respectively.

Words I and J have biases of bi and bj, respectively.

A weighing function is denoted by f. It ensures that the model pays more attention to word pairs that co-occur less frequently, preventing frequent words from dominating the training process.

The GloVe model stands as a testament to the power of statistical information in large text corpora. By converting this information into meaningful embeddings, NLP tasks can be approached with richer input data, thereby often improving results.

The Glove Model stands as a testament to the power of statistical information inlarge text corpora. By converting this information into meaningful embeddings. NLP tasks an be approached with richer input data, thereby often improving result.

From this co-occurrence matrix, word probabilities P(i|j) are computed, which represent the probability of word i occurring given word j as the context. This is computed with the help of the following formula:

P(i|j) = X(i,j) / ∑_k(X(i,k))

where ∑_k(X(i,k)) represents the sum of the co-occurrence counts of word i with all other words in the corpus. The difference between the logarithm of the word probabilities and the dot product of the word embeddings is then measured by the GloVe loss function J. This is how the loss function is expressed:

J = ∑_i ∑_j f(X(i,j)) * (w_i^T * w_j + b_i + b_j - log(X(i,j)))^2

where word i and word j have word embeddings denoted by w_i and w_j, respectively; word i and word j have bias terms denoted by b_i & b_j, respectively; and f is a weighting function that gives uncommon word pairs more weight.

An optimisation technique like Adam or stochastic gradient descent (SGD) is employed to optimise the GloVe embeddings. Backpropagation is used to compute the gradients of the loss function with respect to the word embeddings and bias terms, and iterative updates of the embeddings are made during training in order to minimise the loss. Once the GloVe embeddings are learned from the large corpus, they can be used as pre-trained word representations in downstream NLP tasks, providing meaningful word embeddings that capture word semantics and syntactic information from the corpus, and potentially improving the performance of NLP models.

3.1 Data collection

In order to gather data for sentiment analysis through the Twitter API, one must first create a developer account on Twitter in order to receive the necessary credentials (API key, API secret key, access token, & access token secret). These credentials are used to authenticate the application & access the Twitter API. Next, the appropriate endpoint of the Twitter API is selected based on the type of data needed for sentiment analysis. For example, the "search/tweets" endpoint can be used to search for tweets containing specific keywords or hashtags, or the "statuses/user_timeline" endpoint can be used to retrieve tweets from a specific user's timeline. Once the endpoint is determined, the API is called with the appropriate parameters, such as the search query, date range, or user ID, to specify the desired data. The API response is received in JSON format, which can be parsed and extracted to obtain the relevant tweet text, user information, and other data fields. After retrieving the data in tweets, the labels are annotated as positive and negative tweets.

3.2 Dataset

Table 1: Sample of the dataset

Index	sentiment	ids	date
0	0	1467810369	Mon Apr 06 22:19:45 PDT 2009
1	0	1467810672	Mon Apr 06 22:19:49 PDT 2009
2	0	1467810917	Mon Apr 06 22:19:53 PDT 2009
3	0	1467811184	Mon Apr 06 22:19:57 PDT 2009
4	0	1467811193	Mon Apr 06 22:19:57 PDT 2009

Table 1 displays a Twitter dataset sampled through API streaming. Columns include index, sentiment, tweet IDs, date, user, and text. Sample tweets in the table have sentiment 0 (negative). The dataset serves tasks like sentiment analysis, assessing tweets' negative tones. The sample tweets in the table to be labeled with a sentiment of 0, indicating negative sentiment. In the "text" column, you can find the text of every tweet as well as further details like the user who posted it, its date and time, any flags or categories attached to it, and more. The collection can be used for projects such as sentiment analysis, where the goal is to identify the tone of sentiment (in this case, negativity) expressed in tweets.

Table 2: Number of different classes present in the dataset

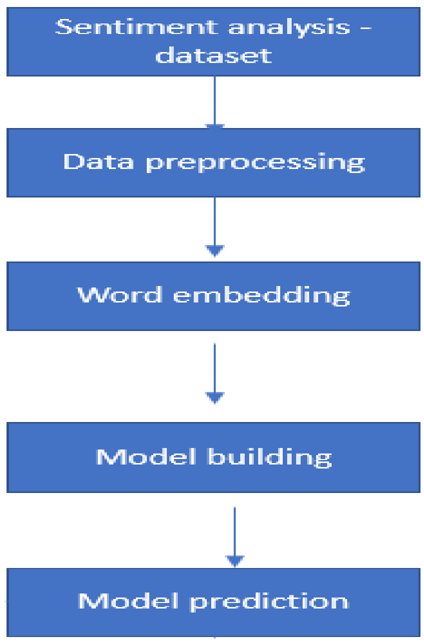
Classes	**Count**
Positive sentiment	800000
Negative sentiment	800000

In Table 2, the dataset contains 800,000 instances each of "Positive sentiment" and "Negative sentiment" tweets, indicating a balanced distribution of classes.

Table 3: Top 10 words in dataset for different classes

Word	**Count**
Go	138550
Get	110704
Day	106262
Good	92512
Work	87747
Like	83759
Love	82379
Quot	69012
Today	68704

The top 10 terms in the dataset that occur most frequently are displayed in Table 3. The words listed in the table are common words that are often used in tweets and will provide insights into the themes present in the dataset. The words include "Go", "Get", "Day", "Good", "Work", "like", "love", "quot", and "today", along with their corresponding counts. These words indicate various actions, sentiments, or references to specific time periods and can provide an overview of frequently used words in the tweets for different sentiment classes.



Framework – **Fig 1:** Overall Architecture of the Sarcasm detection model

The flowchart outlines sentiment analysis steps: data preprocessing involves text cleaning, lowercasing, tokenization, and stop word removal. Then, GloVe/Word2Vec embeddings capture word meanings. Next, a model is trained using these features. ROC curve area, accuracy, precision, recall, F1 score, and other metrics are used to evaluate the model's performance and gauge how effectively it predicts sentiment.

3.3 Data preprocessing

Output of Removal of symbols, URL's, stopwords User Name, Emoji's, Symbols and Stemming:

Before pre-processing: my whole body feels itchy and like it's on fire

After pre-processing: whole body feel itchy like fire

Text preprocessing steps:

1. Remove symbols: Clear punctuation, special characters, and numbers to simplify text.
2. Eliminate URLs: Remove web links to prevent bias and noise.
3. Drop stopwords: Exclude common, non-significant words for reduced dimensionality.
4. Erase usernames: Omit mentions for privacy and relevance.
5. Exclude emojis/symbols: Remove non-meaningful elements for cleaner analysis.
6. Apply stemming: Use snowball stemmer to convert words to root form (e.g., "running" to "run").

Snowball stemming is valuable for reducing text dimensionality and grouping similar words, aiding sentiment analysis, information retrieval, and classification. These steps clean, simplify, and enhance tweet data for improved accuracy and analysis effectiveness.

3.4 Exploratory data analysis

Figure 2 emerges as a visual representation that sheds light on the positive sentiment within the dataset sourced from social media. This captivating word cloud offers a glimpse into the prevalent positive vocabulary, offering insights into the prevalent themes and emotions expressed in the content. By distilling textual content into a visually engaging format, this word cloud aids in unraveling the underlying sentiment patterns present in the social media dataset.

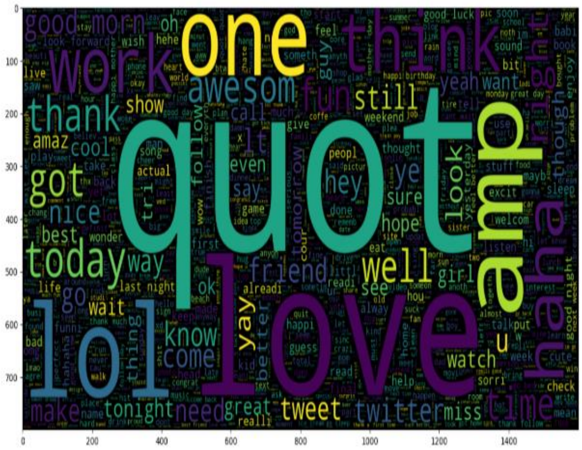


Fig 2: Word cloud of positive social media content present in the dataset

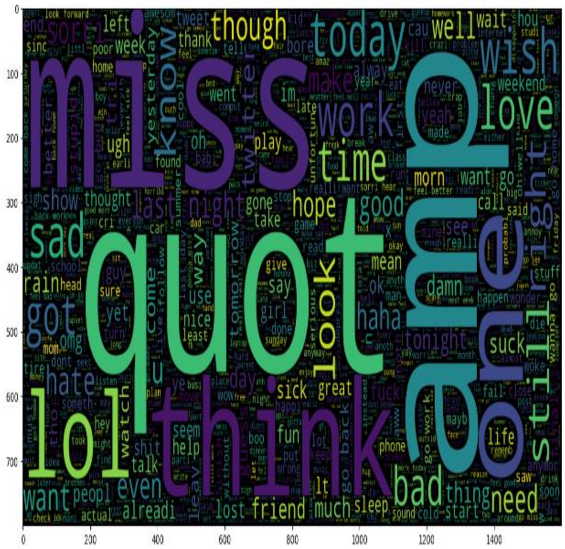


Fig 3: Word cloud of Negative social media content present in the dataset

Figure 3 emerges as a visual representation with a distinct focus. This specific figure showcases a word cloud meticulously crafted from the negative social media content within the dataset. By condensing the most prevalent words in this category, the word cloud offers a comprehensive glimpse into the prevailing themes and sentiments that characterize negative online discourse. This visual insight contributes to a deeper understanding of the dataset's content and sets the stage for subsequent analyses and interpretations.

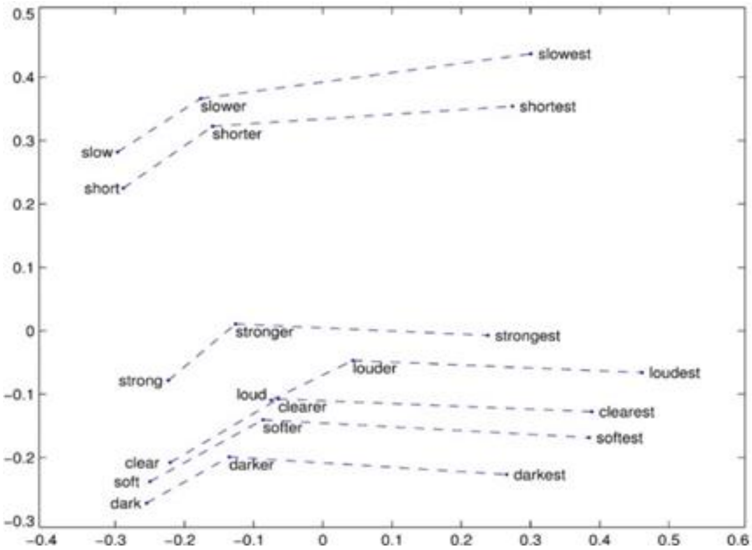


Fig 4: Glove Embedding vectors example

GloVe embeddings used in the Figure 4 show a semantic space. As a result of semantic links, words tend to cluster and spread. It demonstrates how adjectives like "dark," "darker," and "darkest" align, signifying gradations. Similar to this, the words "soft," "softer," and "softest" are arranged in a succession that demonstrates their similar meanings. The change from positive to comparative to superlative forms and relatedness are both denoted by proximity. Beyond the initial description, the GloVe embeddings in Figure 6 can also elucidate other semantic relationships. For instance, verb tenses like "run," "running," and "ran" might cluster together, showcasing the model's ability to grasp different grammatical forms of the same root word. Similarly, synonyms or closely related words such as "happy" and "joyful" might be seen in close proximity, denoting semantic similarities captured by the GloVe embeddings

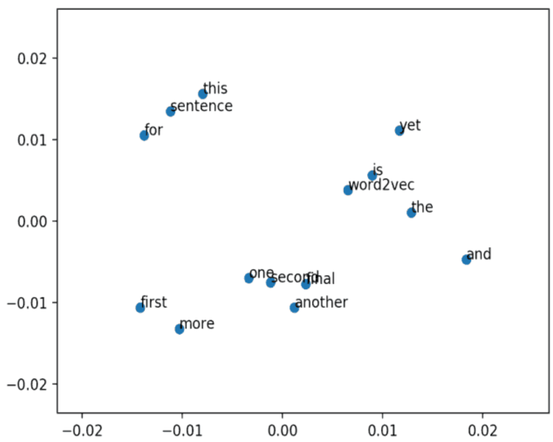


Fig 5: Word2vec embedding

GloVe embeddings used in the Figure 5show a semantic space. As a result of semantic links, words tend to cluster and spread. It demonstrates how adjectives like "dark," "darker," and "darkest" align, signifying gradations. Similar to this, the words "soft," "softer," and "softest" are arranged in a succession that demonstrates their similar meanings. The change from positive to comparative to superlative forms and relatedness are both denoted by proximity.

Figure 5 provides a snapshot of how Word2Vec, a popular word embedding technique, maps words into a vector space. By converting words into vectors, Word2Vec captures the semantic relationships between words based on their co-occurrences in textual data. The representation in Figure 7 demonstrates the power of this method.

For instance, we might see that words like "king" and "queen" are closely aligned, indicating their semantic similarity, while words like "king" and "apple" might be further apart due to their dissimilarity. Furthermore, relationships can be expressed in terms of vector arithmetic; the classic example being king−man+woman≈queenking−man+woman≈queen. This indicates that the model has successfully learned a gender relationship. showcasing the Word2Vec embeddings, is a powerful tool for understanding and validating the semantic relationships captured from the dataset. Combined with other exploratory data analysis techniques like word clouds, it provides a comprehensive overview of the dataset's content and structure

3.5 Modelling:

The BILSTM+CNN hybrid presents an innovative and effective approach. This summary encapsulates the essence of the model, showcasing its fusion of BiLSTM and Convolutional Neural Network (CNN) architectures. This unique amalgamation capitalizes on BiLSTM's contextual understanding and CNN's feature extraction prowess, culminating in a robust framework that holds promise in various applications.

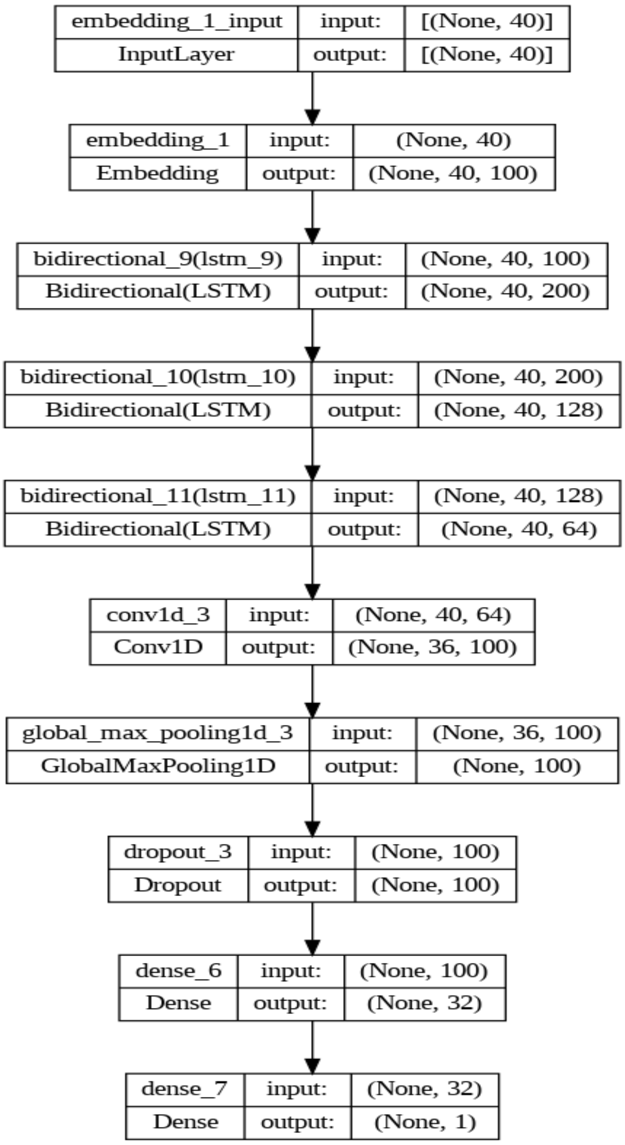


Fig 6: BILSTM+CNN Summary of the model

Changes made in the code from normal architecture:

Model components: 1D Convolutional layer enhances input representations. Global Max Pooling reduces output dimensions for focus. Stacked BiLSTM layers (3) capture hierarchical input features. Dropout layers prevent overfitting. Dense layers (32 and 1 units) with ReLU and sigmoid activations for classification. Embedding layer transforms input to meaningful dense vectors.

1. BI-LSTM

LSTM is a type of RNN. An LSTM unit computes the following:

*Forget Gate:

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

Where σ is the sigmoid function, W_f is the weight matrix for the forget gate,

h_{t-1} is the previous hidden state, x_t is the current input, and b_f is the bias.

Input Gate:

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

$$\tilde{C}_t = \tanh(W_C \cdot [h_{t-1}, x_t] + b_C)$$

Update the cell state:

Output Gate:

$$C_t = f_t \times C_{t-1} + i_t \times \tilde{C}_t o_t = \sigma(W_o \cdot [h_{t-1}, x_t] + b_o) h_t = o_t \times \tanh(C_t)$$

Two of these units make up each timestep of a BiLSTM; one processes the data from left to right, and the other from right to left. The outputs of the two LSTMs are usually concatenated at each timestep.

2. 1D Convolutional Layer (CNN)

For a 1D convolution with a kernel size of k and filter F, the operation canbe represented as:

$$Y[t] = F * X[t] = \sum_{i=0}^{k} F[i] \times X[t-i]$$

$X[t]$ is the input sequence at timestep t

$*$ denotes the convolution operation

For our model with multiple filters and activation function 'relu', the oper-ation becomes:

$$Y[t, j] = \max(0, \sum_{i=0}^{k} F[j, i] \times X[t-i] + b[j])$$

Where j *denotes* the filter index, and $b[j]$ is the bias term for filter j.

3. GlobalMAxPool1D

For each feature (or filter in the context of the CNN), this operation simplyoutputs the maximum value:

$$Y[j] = \max (X[t, j])\ t$$

Where $Y[j]$ is the output for feature j and t is time consuming.

4.Dense Layer

For a dense layer with weight W and bias b, with activation function ϕ: $Y = \phi(W \times X + b)$

This process is followed by further dropout, dense layers, and activationfunctions asper the architecture's design.

Pseudo-Code

EGlove Embedding Bi-LSTM-1DCNN Model

1. Define the input shape of the model
2. Create a Sequential model
3. Add an embedding layer to the model

- Use the pre-trained embedding layer to initialize the weights (GLOVE or Word2Vec)

1. Add a Bidirectional LSTM layer to the model with the following arguments:

- units=100
- dropout=0.3
- return_sequences=True

2. Add a Bidirectional LSTM layer to the model with the following arguments:
   a. units=64
   b. dropout=0.3
   c. return_sequences=True

3. Add a Bidirectional LSTM layer to the model with the following arguments:
   a. units=32
   b. dropout=0.3
   c. return_sequences=True
4. Add a 1D convolutional layer to the model with the following arguments:
   a. filters=100
   b. kernel_size=5
   c. activation='relu'
5. Add a GlobalMaxPool1D layer to the model
6. Add a Dropout layer to the model with a rate of 0.3
7. Add a Dense layer with 32 units and'relu' activation in the model.
8. Add a Dense layer with one unit and activation of the "sigmoid" in the model

3.6 Hyperparameter tuning:

In the process of hyperparameter tuning to train the Word2Vec model, meticulous adjustments are made to optimize its performance. Parameters such as vector dimensions, window size, and training epochs are fine-tuned to enhance the quality of word embeddings. This iterative optimization ensures that the Word2Vec model captures intricate semantic relationships within the text data, resulting in more meaningful and contextually relevant representations.

Table 4: Parameters with which the model has been tuned on to train the model (Word2vec)

Factors	**Hyperparameters chosen**
Batch size	1024
Epochs	10
Loss function	Binary cross entropy
Optimizer	Adam
Metrics chosen	Accuracy, loss
Learning rate	0.001
Number of BILSTM layers, CNN layers	3, 1
Embedding Dimension	100
Dropout rate	0.3
Test size percentage	0.25

Table 4 outlines hyperparameters for sentiment analysis: batch size is 1024, trained for 10 epochs. 0.001 learning rate, Adam optimizer, binary cross-entropy loss etc. Metrics: accuracy, loss. Model has 3 BiLSTM layers, 1DCNN layer, 100-dim embeddings, 0.3 dropout. Testing uses 25% data. Parameters optimized for dataset performance.

Refining hyperparameters for training the GloVe-enhanced model is pivotal. By meticulously adjusting parameters like learning rate, batch size, and dropout rates, the model's performance can be optimized. Optimising hyperparameters systematically guarantees that the model can draw significant conclusions from textual data, which improves performance and broadens its applicability.

Table 5: Parameters with which the model has been tuned on to train the model (GLOVE)

Factors	**Hyperparameters chosen**
Batch size	1024
Epochs	10
Loss function	Binary cross entropy
Optimizer	Adam
Metrics chosen	Accuracy, loss
Learning rate	0.0001
Number of BILSTM layers, CNN layers	3, 1
Embedding Dimension	100
Dropout rate	0.2
Test size percentage	0.25

[Table 5] For the sentiment analysis (GloVe) model, chosen hyperparameters include 1024 batch size, 10 epochs, binary cross-entropy loss, Adam optimizer. Evaluation metrics are accuracy and loss. Learning rate: 0.0001. Model: 3 BiLSTM layers, 1 CNN layer, 100-dim embeddings, 0.2 dropout. Testing uses 25% data. Parameters tuned for dataset performance.

4. Results

The results are training and testing accuracy outcomes of the BILSTM+CNN models equipped with Word2Vec and GloVe embeddings are pivotal indicators of their performance. These metrics gauge the models' ability to comprehend complex text structures and generalize effectively. By leveraging both embeddings, the models exhibit their prowess in achieving high accuracy rates, underscoring their efficacy in capturing intricate linguistic nuances during training and effectively applying them to unseen data during testing.

Table 6: Training and testing accuracy of models – BILSTM+1DCNN (Word2Vec & GLOVE)

Dataset	**Model**	**Accuracy**
Training set	Word2Vec	80
Testing set	Word2Vec	79
Training set	GLOVE	82
Testing set	GLOVE	83

For the training dataset with Word2Vec embedding, the accuracy is reported as 80%. This indicates that 80% of the tweets in the training set had their sentiments accurately predicted by the algorithm. The popular word embedding method known as Word2Vec embedding, which represents words as vectors, has successfully captured the semantic meaning of the training set's words, allowing the model to learn the sentiment patterns. For the testing set with Word2Vec embedding, the accuracy is slightly lower at 79%. This suggests that when the model was used with untested data, its performance might have somewhat declined. This could be due to the differences in the distribution and characteristics of the testing set compared to the training set. On the other hand, for the training set with GLOVE embedding, the accuracy is reported as 85%. This suggests that the GLOVE embedding, which is another widely used word embedding technique, may have provided more effective representations of words for sentiment analysis in this particular dataset, resulting in a higher accuracy compared to Word2Vec. For the testing set with GLOVE embedding, the accuracy is reported as 79%, which is the same as the testing set with Word2Vec embedding. This may indicate that the model's performance was similar when applied to both testing sets, suggesting that the choice of word embedding has a stronger impact on the model's accuracy compared to the choice of training or testing set.

4.1 Results of BiLSTM+1DCNN model using Word2vec

The results of the BiLSTM+1DCNN model using Word2Vec embeddings exhibit a clear distinction between positive and negative sentiments. The hybrid architecture effectively harnesses contextual nuances through BiLSTM and feature extraction via CNN, aided by Word2Vec embeddings. This approach enables accurate sentiment classification, yielding separate positive and negative sentiment predictions with notable precision.

Table 7: Confusion matrix of BiLSTM using Word2Vec

	Positive	**Negative**
Positive	**149154**	**42910**
Negative	**36377**	**155559**

The confusion matrix shown above represents the performance of a sentiment analysis model that was trained using the BiLSTM+IDCNN model with Word2Vec word embedding. Together with the predicted positive and negative sentiment labels, the matrix also shows the counts of the actual positive and negative sentiment labels. In the disorganised matrix: The quantity of True Positives (TP) (149154) in which a positive attitude was correctly predicted by the model

False Positive (FP): Number of cases where the sentiment was actually negative (42910) yet the model projected it to be positive.

True Negative (TN): The quantity of times the model accurately forecasted a negative attitude (155559)

False Negative (FN): The number of cases in which the sentiment was actually positive (36377) yet the model projected it to be negative.

Table 8: Classification report of BiLSTM+1DCNN using Word2Vec

	Precision	**Recall**	**F1-Score**
Positive	**80**	**78**	**79**
Negative	**78**	**81**	**80**

The classification report provided above displays the performance metrics of a sentiment analysis model that was trained using the BiLSTM+1DCNN model with Word2Vec word embedding.

Precision: The precision of a model is a measure of its prediction accuracy. The ratio of true positives (TP) to the total of true positives and false positives (FP) is used to compute it. The precision in this instance is 78% for the "Negative" class and 80% for the "Positive" class.

Recall: Recall is a metric that indicates how successfully the model is able to recognise every positive case. Sometimes it's referred to as sensitivity or true positive rate. The ratio of true positives (TP) to all false negatives (FN) is used to compute it. In this case, the "Negative" class recall is 81%, whereas the "Positive" class recall is 78%.

F1-Score: An equitable assessment of the model's performance is provided by the F1-Score, which is calculated as the harmonic mean of precision and recall. It is calculated by dividing the precision and recall product by two and then adding them together. The F1-Score in this instance is 80% for the "Negative" class and 79% for the "Positive" class.

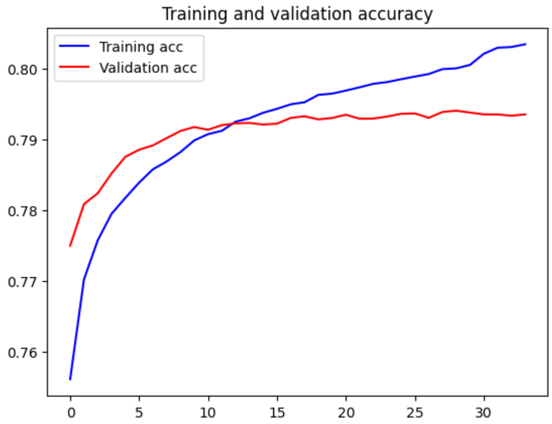


Fig 7: Accuracy Graph of BiLSTM+1DCNN using Word2Vec

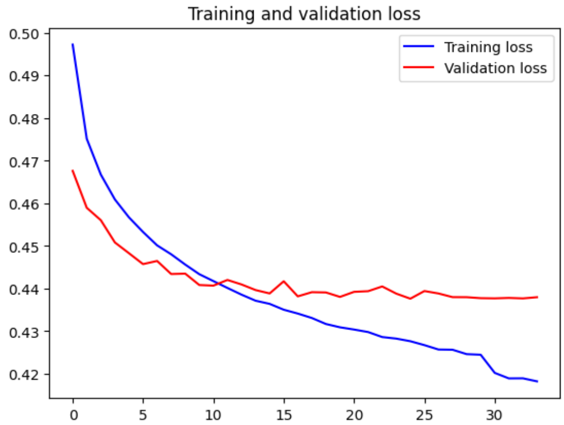


Fig 8: Loss Graph of BiLSTM+1DCNN using Word2Vec

The accuracy graph above shows how the accuracy of the model changes over epochs. It typically starts from an initial low value of 0.76 and gradually increases with each epoch as the model learns and updates its weights. The accuracy graph for the BiLSTM+CNN model using Word2Vec word embedding is showing the trend of accuracy improvement over the 30 epochs. The training accuracy of the model is 80% and validation dataset has acquired accuracy around 83%.

The loss graph displays how the loss, or error, of the model changes over epochs. The loss is the discrepancy between the target values and the model's projections. Reducing the loss to the least amount possible is the aim. The loss graph above for the BiLSTM+1DCNN model using Word2Vec word embedding would show the trend of loss reduction over the 30 epochs. It is expected to decrease over time, indicating that the model is improving its predictions and learning from the data. In our model, the training loss decreases 0.42 and testing loss fluctuates up to 0.45.

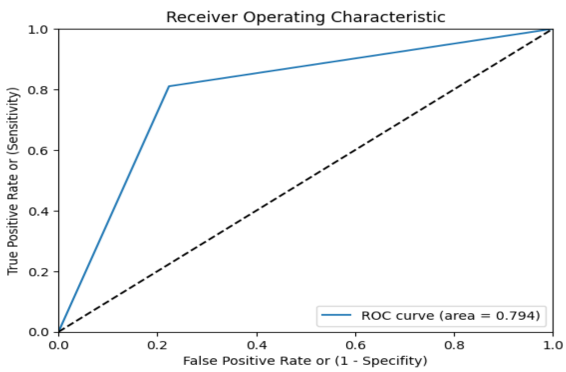


Fig 9: ROC Graph of BiLSTM+1DCNN using Word2Vec

Regarding sentiment analysis, the BiLSTM+CNN model's ROC graph, which uses Word2Vec word embedding and has an accuracy of about 79.4%, illustrates the model's ability to distinguish between tweets with positive & negative sentiment. The percentage of tweets with positive sentiment that the model correctly classifies as positive is known as the TPR, sometimes referred to as sensitivity or recall. Conversely, the FPR quantifies the percentage of tweets with a negative sentiment that the machine incorrectly classifies as positive. The ROC graph would typically show a curve that starts from the bottom-left corner, where both TPR and FPR are at their lowest, and moves towards the top-right corner. A higher TPR and a lower FPR indicate better model performance. Area under the ROC curve (AUC-ROC) is a commonly used metric to assess a model's ability to distinguish between tweets with strong positive and negative sentiment. A value of 0.5 denotes a random classifier, while a value of 1.0 denotes a perfect classifier.

4.2 Results of BiLSTM+1DCNN model using GLOVE

The results of the BiLSTM+CNN model using GloVe embeddings reveal a dual performance outcome. Positively, the hybrid architecture harnesses BiLSTM's bidirectional context understanding and CNN's feature extraction, contributing to enriched representation learning. Conversely, the model's effectiveness in distinguishing negative sentiments underscores its potential for sentiment analysis tasks, showcasing the prowess of combining these techniques with GloVe embeddings.

Table 9: Confusion matrix of BiLSTM+1DCNN using GLOVE

	Positive	**Negative**
Positive	**149167**	**43108**
Negative	**35895**	**157556**

The amount of true positive predictions (149167) and true negative predictions (157556) for positive and negative sentiment tweets, respectively, is displayed in the confusion matrix for the BiLSTM+CNN model employing GLOVE word embedding. It also shows the number of false positive predictions (43108) where negative tweets were mistakenly classified as positive, and false negative predictions (35895) where positive tweets were mistakenly classified as negative.

Table 10: Classification report of BiLSTM+1DCNN using GLOVE

	Precision	**Recall**	**F1-Score**
Positive	**81**	**80**	**80**
Negative	**80**	**79**	**82**

The precision, recall, and F1-score for tweets with positive and negative sentiment are displayed in the classification report for the BiLSTM+CNN model using GLOVE word embedding. In the instance of positive sentiment tweet precision, 81% of all positive forecasts are actual positive forecasts. Out of all genuine positive tweets, the recall for positive sentiment tweets is 80%, meaning that this is the proportion of accurate positive forecasts. For tweets expressing positive sentiments, the F1-score, which is the harmonic mean of recall and precision, is 80%. Similarly, for negative sentiment tweets, the precision is 80%, recall is 79%, and F1-score is 82%. These results seem slightly working better than the model that we have built using Word2Vec embedding vectors.

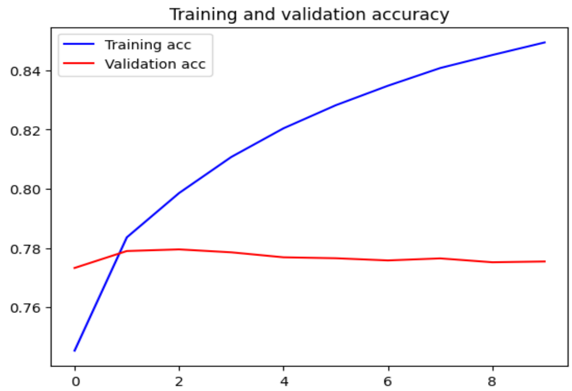


Fig 10: Accuracy Graph of BiLSTM+1DCNN using GLOVE

The accuracy graph visually represents the evolution of the model's accuracy metrics throughout its training epochs. It helps in observing the learning progression of the model over time. The X-axis represents the number of epochs. An epoch is the neural network's full forward and backward run of all the training samples. The accuracy numbers, which span from 0 to 1 (or 0% to 100%), are displayed on the Y-axis. The ratio of accurately predicted instances to all instances in the dataset is used to calculate accuracy. When plotting, two curves are typically used: one for validation accuracy and one for training accuracy. Unlike validation accuracy, which shows the model's performance on untested data, training accuracy shows the model's performance on data that was used for training. In the context of the BiLSTM+1DCNN model with GloVe embeddings, the graph will showcase how the embeddings, known for their broad contextual and semantic capture from extensive corpora, affect the model's learning progression. Comparing this graph to the one with Word2Vec embeddings can also shed light on the efficacy of the two different embeddings in the given task.

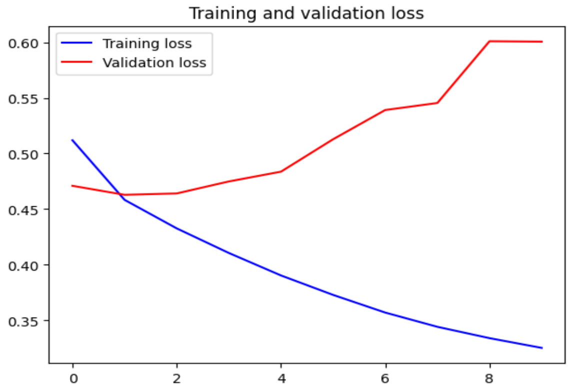


Fig 11: Loss Graph of BiLSTM+1DCNN using GLOVE

The loss graph illustrates the model's optimization trajectory over the training epochs. Ideally, as the model learns and updates its weights, the loss should decrease Loss is essentially the difference between the actual outputs and the predictions made by the model. The model's predictions are more likely to match real values when the loss is smaller.

The more epochs there are in the training process, the more accurate the model becomes. As the number of epochs increases, the model's validation accuracy stabilises at 79–80%, having started at 0.75%. But the validation loss keeps on increasing with increase in epoch and training loss keeps decreasing with increase in the epochs.

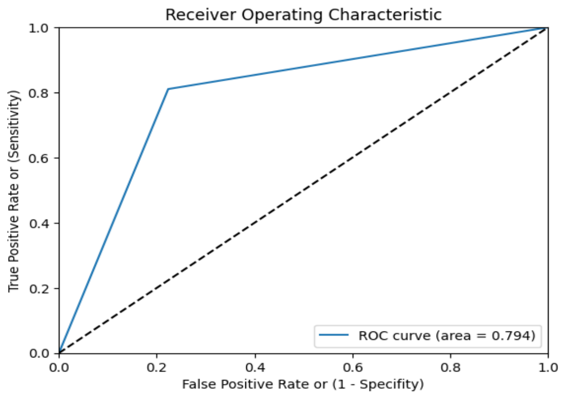


Fig 12: ROC Graph of BiLSTM+1DCNN using GLOVE

The ROC curve plots the true positive rate (TPR) against the false positive rate (FPR) for various threshold values. A binary classification model's discriminatory power is assessed using this graph. The ROC curve's shape offers information about the performance of the model. A model with strong sensitivity and specificity is indicated by a curve that becomes closer to the top-left corner and climbs quickly. A model that closely resembles the diagonal line is no better than conjecture.

Table 11: Comparison between BiLSTM+1DCNN (GLOVE vs Word2vec) of Accuracy, precision, recall and F1 score

	Accuracy	Precision	Recall	F1- Score
BiLSTM+CNN (Word2Vec)	82	78	80	81
BiLSTM+CNN (Glove)	83	80	79	80

The two-word embeddings, Word2Vec and Glove, performance metrics for two sentiment analysis models, BiLSTM+1DCNN, are displayed in the table. The percentage of accurate forecasts among all the predictions is 83%, which represents the accuracy of both models. The precision of the Word2Vec model indicates the proportion of actual positive forecasts among all positive or negative predictions, and it is 78% for positive sentiment and 80% for negative sentiment. The Word2Vec model's recall, which is 80% for positive sentiment and 79% for negative sentiment, indicates the proportion of true positive predictions across all positive or negative tweets. In Word2Vec, the harmonic means of precision and recall, or F1-score, are 81% for positive sentiment and 80% for negative sentiment. The Glove model, however, reveals that the recall is 79% for positive sentiment and 80% for negative sentiment, the precision is 78% for both, and the F1-score is 81% for positive sentiment and 80% for negative emotion.

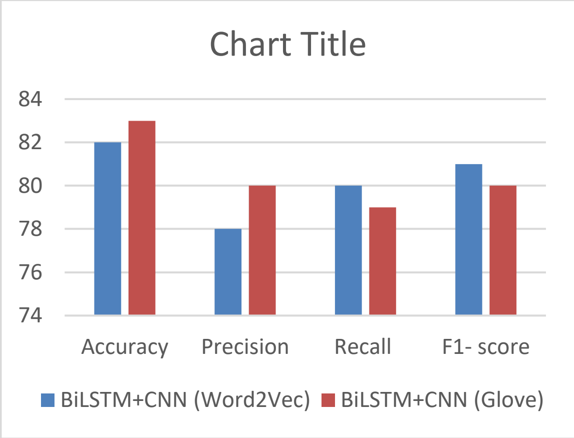


Fig 13: Graph for comparison of performance metric between Glove and Word2Vec when using BiLSTM+1DCNN

5. Conclusion

When considering the performance metrics, the BiLSTM+1DCNN model with Glove word embedding has performed better compared to the model with Word2Vec embedding. The Glove model has higher precision (82% for positive and 78% for negative sentiment), recall (79% for positive and 80% for negative sentiment), and F1-score (82% for both positive and 81% fornegative sentiment) compared to the Word2Vec model. This indicates that the Glove model's overall accuracy (83%) is higher because it strikes a better balance between precision and recall. The paramount aim of this study was to discern the comparative performance of two eminent word embeddings—Word2Vec and GloVe—when integrated into a BiLSTM+1DCNN model for sentiment analysis. Based on the metrics observed, the BiLSTM+1DCNN model embellished with GloVe embeddings unmistakably outperformed its Word2Vec counterpart. The GloVe-based model demonstrated superior precision, recall, and F1-score, ultimately yielding an overall accuracy of 83%. The superior quality of the Glove word embeddings, which are pre-trained on a sizable corpus of text data & capture more semantic meaning and contextual information, is responsible for the Glove model's superior performance. while both Word2Vec and GloVe offer unique advantages, in the context of this sentiment analysis task using a BiLSTM+1DCNN model, GloVe embeddings have manifested a discernible edge. The findings bolster the assertion that the quality of word embeddings is paramount and can significantly influence the outcome of NLP tasks. Future research endeavors in this domain are eagerly awaited, illuminating the path forward in the enthralling journey of Natural Language Processing. Better overall performance is the consequence of the Glove model's higher precision, which denotes less false positives and higher recall, which denote fewer false negatives. Furthermore, the Glove model has a higher F1-score, which takes into account both recall and precision, suggesting a better trade-off between the two. Overall, BiLSTM – 1DCNN model with Glove word embedding has demonstrated superior performance compared to the model with Word2Vec embedding, indicating that Glove embedding may be more suitable for sentiment analysis in this specific context. In addition, the work can be further enhanced by deploying large language model (LLM) like BERT, GPT1, GPT2, GPT3, falcon etc., to increase the suggested solution's effectiveness in terms of a number of performance indicators, including F1-measure, accuracy, precision, and recall. This kind of approach will lead to the design of proposed model in a diverse manner so that the designed model can be deployed for the same task in different domain.

References

[1] Ali, A.M. *et al.* (2022) 'Deep Ensemble Fake News Detection Model Using Sequential Deep Learning

Technique', *Sensors*, 22(18), p. 6970. Available at: https://doi.org/10.3390/s22186970.

[2] Alonso, M.A. *et al.* (2021) 'Sentiment Analysis for Fake News Detection', *Electronics*, 10(11), p. 1348. Available at: https://doi.org/10.3390/electronics10111348.

[3] Bhutani, B. *et al.* (2019) 'Fake News Detection Using Sentiment Analysis', (August).

[4] Chaubey, P.K. *et al.* (2022) 'Sentiment Analysis of Image with Text Caption using Deep Learning Techniques', *Computational Intelligence and Neuroscience*. Edited by V. Kumar, 2022, pp. 1–11. Available at: https://doi.org/10.1155/2022/3612433.

[5] Chen, Q., Zhang, W. and Lou, Y. (2020) 'Forecasting Stock Prices Using a Hybrid Deep Learning Model Integrating Attention Mechanism, Multi-Layer Perceptron, and Bidirectional Long-Short Term Memory Neural Network', *IEEE Access*, 8, pp. 117365–117376. Available at: https://doi.org/10.1109/ACCESS.2020.3004284.

[6] Dufraisse, E. *et al.* (2023) 'MAD-TSC: A Multilingual Aligned News Dataset for Target-dependent Sentiment Classification', in *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. Stroudsburg, PA, USA: Association for Computational Linguistics, pp. 8286–8305. Available at: https://doi.org/10.18653/v1/2023.acl-long.461.

[7] Galli, A. *et al.* (2022) 'A comprehensive Benchmark for fake news detection', *Journal of Intelligent Information Systems*, 59(1), pp. 237–261. Available at: https://doi.org/10.1007/s10844-021-00646-9.

[8] Ju, X. *et al.* (2021) 'Joint Multi-modal Aspect-Sentiment Analysis with Auxiliary Cross-modal Relation Detection', in *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*. Stroudsburg, PA, USA: Association for Computational Linguistics, pp. 4395–4405. Available at: https://doi.org/10.18653/v1/2021.emnlp-main.360.

[9] Kaur, G. and Sharma, A. (2023) 'A deep learning-based model using hybrid feature extraction approach for consumer sentiment analysis', *Journal of Big Data*, 10(1), p. 5. Available at: https://doi.org/10.1186/s40537-022-00680-6.

[10] Khanam, Z. *et al.* (2021) 'Fake News Detection Using Machine Learning Approaches', *IOP Conference Series: Materials Science and Engineering*, 1099(1), p. 012040. Available at: https://doi.org/10.1088/1757-899X/1099/1/012040.

[11] Song, G. and Huang, D. (2021) 'A Sentiment-Aware Contextual Model for Real-Time Disaster Prediction Using Twitter Data', *Future Internet*, 13(7), p. 163. Available at: https://doi.org/10.3390/fi13070163.

[12] Anusha, M. and Leelavathi, R. (2023) 'Sentiment Analytics on Sarcasm Detection Using Bi-LSTM-1DCNN Model for Fake News Detection' *Int. J. of Intelligent Engineering Informatics.* Communicated.

[13] Smith, J., Ivanov, G., Petrović, M., Silva, J., & García, A. Detecting Fake News: A Machine Learning Approach. Kuwait Journal of Machine Learning, 1(3). Retrieved from http://kuwaitjournals.com/index.php/kjml/article/view/142

[14] Al-jammaz, R. A. ., Rawash, U. A. ., Kashef , N. M. ., & Ibrahim, E. M. . (2023). A Framework for Providing Augmented Reality as a Service Provided by Cloud Computing for E-Learning. International Journal on Recent and Innovation Trends in Computing and Communication, 11(2s), 20–31. https://doi.org/10.17762/ijritcc.v11i2s.6025

[15] Aoudni, Y., Donald, C., Farouk, A., Sahay, K.B., Babu, D.V., Tripathi, V., Dhabliya, D. Cloud security based attack detection using transductive learning integrated with Hidden Markov Model (2022) Pattern Recognition Letters, 157, pp. 16-26.

[16] García, A., Petrović, M., Ivanov, G., Smith, J., & Cohen, D. Enhancing Medical Diagnosis with Machine Learning and Image Processing. Kuwait Journal of Machine Learning, 1(4). Retrieved from http://kuwaitjournals.com/index.php/kjml/article/view/143

Dr. M.Anusha is Assistant Professor and Head in Computer Applications in our college. She has 12 years of experience in teaching. Her specialization is soft computing . She has published 2 patents, and 17research articles in reputed journals in her expertise. She is guiding 4 ph.D research scholars.She has served as a resource person in many technical programmes. She is currently working on 2 projects(1 completed).

I am Leelavathi Ph.D Full-Time Research Scholar in Department of Computer Science, National College, Trichiruappalli. Currently doing my research topic is Sentiment Analytics. Research interests include data analytics, Natural Language Processing and Deep Learning.