IJARCCE ISSN (O) 2278-1021, ISSN (P) 2319-5940

International Journal of Advanced Research in Computer and Communication Engineering

Impact Factor 8.102 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 14, Issue 4, April 2025

DOI: 10.17148/IJARCCE.2025.14421

Comparative Analysis of Activation Functions in LSTM Models for Sentiment Classification

Shaikh Ateeb Ahmed[1], Namdeo B.Badhe[2], Rahul P.Neve[3]

Graduate, Student, Department of Information Technology, Thakur College of Engineering & Technology, Mumbai, Maharashtra, India[1]

Associate Professor, Department of Information Technology, Thakur College of Engineering & Technology, Mumbai, Maharashtra, India[2,3]

Abstract: This study uses the IMDb movie review dataset to compare how well different deep learning architectures perform in sentiment classification. The experiments evaluate three sequential models, each enhanced with distinct configurations of activation functions and layer compositions. First Model utilizes ReLU and ELU activations within a bidirectional LSTM architecture, Second Model incorporates Tanh and SELU functions, while Third Model adopts a combination of Leaky ReLU and Tanh within a similar structural framework. To determine the effect of network architecture and activation function selection on classification effectiveness, each model is evaluated using accuracy, precision, recall, F1 score, and loss measures. Results indicate that the incorporation of advanced activations such as SELU and Leaky ReLU can lead to performance gains in certain metrics, with Third Model demonstrating improved generalization and lower loss compared to its predecessors. These results highlight how important activation functions are for improving deep learning models for tasks involving natural language processing.

Keywords: Sentiment Analysis, IMDb Dataset, Deep Learning, LSTM, ReLU, ELU, Tanh, SELU, Leaky ReLU, Natural Language Processing.

I. INTRODUCTION

A fundamental task in natural language processing (NLP) is sentiment analysis, which entails determining the attitude or opinion expressed in a text and classifying it as either positive or negative. This feature is crucial for a variety of applications, such as market research, customer feedback analysis, social media monitoring, and brand reputation management, where knowledge of public sentiment is crucial. One of the most widely used datasets for sentiment analysis is the IMDb movie review dataset, which contains 50,000 labelled reviews equally split between positive and negative sentiments. Due to its balanced nature and rich text data, this dataset serves as an effective benchmark for evaluating the performance of deep learning models in sentiment classification tasks.

Given the sequential nature of textual data, Recurrent Neural Networks (RNNs), and more specifically Long Short-Term Memory (LSTM) networks, are especially well-suited for sentiment analysis. By adding gating mechanisms that control the input flow, LSTMs overcome the drawbacks of conventional RNNs, including the vanishing gradient issue. These mechanisms enable the model to retain important contextual information over long sequences, which is crucial when interpreting sentiment accurately. However, the efficiency and effectiveness of LSTM-based models are significantly influenced by the choice of activation functions used within the network layers.

Activation functions are an essential component of neural networks, introducing non-linearity that enables models to learn complex patterns and representations. The choice of activation function impacts how well a model converges during training, how it handles gradients, and ultimately, how accurately it performs on a given task. While traditional functions like Sigmoid are still used for binary classification outputs, modern alternatives such as ReLU, Leaky ReLU, ELU, Tanh, and SELU offer performance benefits by addressing issues like gradient vanishing or neuron saturation. The performance of an LSTM model can therefore be improved not only by adjusting its architecture but also by choosing the most effective activation function configuration for the task at hand.

In this study, we explore the impact of different activation functions on the performance of LSTM-based models for sentiment analysis using the IMDb dataset. Three deep learning models are implemented, each using distinct activation functions and architectural adjustments to evaluate how these choices affect metrics such as accuracy, loss, precision, recall, and F1 score. The goal is to better understand how various activation function strategies can optimize sentiment classification models and contribute to the broader field of NLP.

International Journal of Advanced Research in Computer and Communication Engineering

Impact Factor 8.102 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 14, Issue 4, April 2025

DOI: 10.17148/IJARCCE.2025.14421

First Model is built on a bidirectional LSTM architecture that integrates ReLU and ELU activation functions. ReLU, known for its computational efficiency and sparsity-inducing behaviour, is employed in the dense output layers to accelerate training and avoid saturation. ELU, on the other hand, is utilized to alleviate issues related to vanishing gradients by producing negative outputs that bring the mean activations closer to zero. The use of a bidirectional LSTM allows the model to capture dependencies from both directions in a text sequence, enhancing its contextual understanding. This model serves as a baseline to analyze the impact of using conventional activation functions with established performance benefits in NLP tasks.

Second Model modifies the architecture by introducing the Tanh and SELU activation functions. Tanh is employed within the LSTM layers for its ability to produce zero-centred outputs, helping balance gradients and model behaviour across the network. SELU is applied in the dense layers due to its self-normalizing properties, which maintain mean and variance across layers and promote stable, deeper network training. This configuration aims to improve learning efficiency and model robustness. By evaluating second Model, the study seeks to assess how newer, self-normalizing activation functions like SELU perform in sentiment classification when paired with traditional sequence models like LSTM.

Third Model adopts a hybrid activation strategy by combining Leaky ReLU and Tanh within a bidirectional LSTM framework. By permitting a slight gradient for negative inputs, Leaky ReLU solves the "dying ReLU" issue and guarantees that neurons continue to function and aid in learning even in less advantageous areas of the input space. Tanh is retained in the LSTM layers to preserve balanced output behaviour and enhance sequence modelling. This model represents an attempt to harness the benefits of both stable gradient flow and dynamic learning capabilities. Through third Model, the study evaluates whether this hybrid approach can lead to better generalization and improved overall performance in sentiment analysis.

II. RELATED WORK

Paper [1] presents a comprehensive analysis of IMDb movies through sentiment and topic analysis. The study uses Bi-LSTM for sentiment classification and LDA for topic modelling to uncover key factors influencing successful films. It finds that Bi-LSTM effectively identifies positive and negative sentiments in movie reviews, while LDA reveals that light entertainment and comedy genres are gaining popularity. The study also investigates statistical relationships between movie genres, sentiment, duration, and ratings, showing that Film-Noir and War genres receive higher ratings. The research highlights the importance of sentiment and thematic content in predicting a movie's success, offering valuable insights for movie production companies.

Paper [2] presents a hybrid model (HBCNLS) for sentiment analysis on movie reviews, combining Convolutional Neural Networks (CNN) and Long Short-Term Memory (LSTM) networks. The model extracts local features using CNN and captures sequential dependencies using LSTM. The study evaluates the performance of HBCNLS on a 50,000-review dataset, comparing it with other models like BERT, CNN, and LSTM. The results show that HBCNLS achieves the highest accuracy of 92%, outperforming BERT (90%), CNN (87%), and LSTM (86%). The study highlights HBCNLS's superior precision and recall scores, indicating its effectiveness in sentiment classification.

Paper [3] introduces TextGT, a double-view graph transformer model for Aspect-Based Sentiment Analysis (ABSA). TextGT combines Graph Neural Networks (GNNs) and Transformer models to handle the challenges of aspect-based sentiment classification. The paper proposes a new graph convolution technique, TextGINConv, to alleviate the over-smoothing problem often encountered in GNNs. Extensive experiments on four public ABSA datasets (Restaurant, Laptop, Twitter, Rest16) show that TextGT outperforms existing methods, including those that combine GNNs with BERT. Results show that TextGT achieves higher accuracy and F1 scores compared to traditional GNN-based methods, with TextGT+BERT achieving the best performance. For instance, on the Restaurant dataset, TextGT+BERT achieves 87.31% accuracy, surpassing other models. The paper also highlights the model's robustness to increased depth, indicating its effectiveness in preventing over-smoothing. Additionally, ablation studies validate the effectiveness of each component of the model, particularly the TextGINConv module, for improving ABSA tasks.

Paper [4] examines sentiment analysis of movie reviews using deep learning models, specifically CNN and LSTM, to handle imbalanced datasets. The study applies techniques such as TF-IDF, Word2Vec, and SMOTEN for feature extraction and data balancing. The CNN model achieved an accuracy of 98.56%, while LSTM followed closely with 98.53%. Both models demonstrated strong performance in terms of F1-score, with CNN reaching 77.87% and LSTM 78.92%. The results show the effectiveness of these models for sentiment classification, helping users make informed decisions about movies. Future improvements may focus on further enhancing model accuracy through more complex feature expansion techniques.

DOI: 10.17148/IJARCCE.2025.14421

Paper [5] proposes DPG-LSTM, a novel enhanced LSTM framework for sentiment analysis in social media text. The model integrates dependency parsing (DP) and graph convolutional networks (GCN) to better capture both semantic and syntactic features. By employing a dual-channel network with BiLSTM for semantic information and GCN for syntactic structures, DPG-LSTM improves text feature extraction. The DPG-Attention mechanism further refines these features, emphasizing syntactic dependencies. Experimental results on three datasets—Sentiment140, airline reviews, and self-driving car reviews—show that DPG-LSTM outperforms existing models, with significant improvements in recall, precision, and F1 score. The model achieves an F1 score of 0.843 on Sentiment140, surpassing other advanced models like BiLSTM and BERT. Additionally, ablation experiments reveal that the combination of GCN and DPG-Attention is crucial for its high performance.

Paper [6] presents a comprehensive survey of sentiment analysis techniques across machine learning, deep learning, and ensemble learning approaches. The study reviews over 60 models and methods, highlighting accuracies ranging from 67% to 99.5% across datasets like IMDb, Sentiment140, and Twitter Airline Sentiment. Traditional machine learning models such as Naive Bayes and SVM showed strong results with up to 99.73% accuracy, while deep learning models like CNN and BiLSTM achieved peak accuracies of 99.33% and 94%, respectively. Ensemble methods further enhanced performance, with some hybrid models like RoBERTa-LSTM achieving up to 94.9% accuracy. The paper underscores the importance of preprocessing, feature extraction (e.g., TF-IDF, Word2Vec, BERT), and the use of benchmark datasets, while also identifying future research directions such as fine-grained sentiment classification and handling sarcasm or multilingual data

Paper [7] proposes a text-based sentiment analysis approach using Long Short-Term Memory (LSTM) networks, tested on IMDB and Amazon product review datasets, each comprising 50,000 samples split evenly between positive and negative sentiments. The model utilizes a word embedding layer with 100-dimensional vectors for the top 6000 vocabulary words and is trained using the Adam optimizer and sparse categorical cross-entropy as the loss function. With a batch size of 500 and 128 LSTM units, the model demonstrates progressive performance improvements across epochs, reaching an accuracy of 99.56% by the 10th epoch. These results affirm the efficacy of LSTM in sentiment classification, especially when trained with substantial textual data, and the study suggests exploring alternative embedding models and larger datasets for future enhancements.

Paper [8] conducts a comparative study on sentiment analysis using nine supervised machine learning models—such as Logistic Regression, Linear SVC, Ridge Classifier, and Naive Bayes—on IMDB movie reviews and Twitter sentiment datasets. Feature extraction was performed using TF-IDF on unigram, bigram, and trigram representations. The ensemble voting classifier, built from the top-performing models, achieved the highest accuracy of 90.76% on the IMDB dataset and 76.88% on the Twitter dataset. Ridge Classifier and Linear SVC consistently delivered the best precision and recall scores across both datasets. The study concludes that ensemble models significantly enhance sentiment classification accuracy and suggests future work on more nuanced, multidimensional sentiment analysis.

Paper [9] investigates hybrid deep learning models for sentiment analysis by combining CNN, LSTM, and SVM, tested across eight datasets of tweets and reviews. The study evaluates four hybrid architectures using Word2Vec and BERT for feature extraction, with BERT consistently outperforming Word2Vec. The best performing model, LSTM-CNN with SVM, achieved 93.4% accuracy on IMDb reviews and 92.9% on Tweets Airline data, significantly surpassing individual models like SVM (82.4%), CNN (84.0%), and LSTM (84.1%) on average. Evaluation metrics including precision, recall, F-score, and AUC confirmed that hybrid models consistently delivered better performance across datasets, though at the cost of increased computation time. These results demonstrate the hybrid models' adaptability and effectiveness in improving sentiment classification across diverse domains.

Paper [10] conducts a comparative analysis of deep learning models—DNN, CNN, and RNN—for sentiment analysis using eight datasets, including Sentiment140, IMDB, and Twitter Airlines, with text pre-processed through TF-IDF and Word2Vec embedding. The study employed k-fold cross-validation (k=10) and evaluated models using accuracy, F-score, recall, and AUC. Among all models, CNN with Word2Vec achieved the highest accuracy of 87.9% on Sentiment140, while RNN showed a strong performance on smaller datasets with accuracies above 80%. TF-IDF generally performed better with DNN, while Word2Vec yielded superior results with CNN and RNN. The findings emphasize that no single model dominates across all datasets, suggesting that dataset size and feature extraction method significantly influence model performance in sentiment classification

Paper [11] presents a comparative analysis of deep learning models—LSTM, GRU, and Bidirectional LSTM—for sentiment analysis on the IMDB movie reviews dataset. Each model was trained using word embeddings and tested on 50,000 labelled reviews split evenly between positive and negative sentiments.

DOI: 10.17148/IJARCCE.2025.14421

The BiLSTM model outperformed others with a peak accuracy of 92.03%, followed by GRU at 90.56% and LSTM at 89.23%. All models were evaluated using metrics such as accuracy, precision, recall, and F1-score, with BiLSTM also achieving the highest F1-score of 91.87%. The results indicate that Bidirectional architectures offer superior context understanding, making them more effective for sentiment classification tasks on review-based datasets

Paper [12] explores the role of text pre-processing in sentiment analysis using SVM classifiers on two datasets of 1,400 and 2,000 movie reviews. The study investigates the impact of transformations such as stop word removal, stemming, and negation handling, along with feature selection using the chi-squared method. Feature matrices were created using TF-IDF, feature frequency (FF), and feature presence (FP). Results show a significant accuracy increase after applying pre-processing and filtering, with the highest accuracies reaching 93.5% for TF-IDF and 93% for FP on the larger dataset. Precision, recall, and F-measure scores also improved substantially, with F-measure values close to 0.93. These findings demonstrate that effective pre-processing and feature selection can enhance sentiment classification performance to levels comparable with topic categorization.

Paper [13] provides a comprehensive review of sentiment analysis and emotion detection techniques from text, categorizing approaches into lexicon-based, machine learning, deep learning, transfer learning, and hybrid models. The study highlights sentiment analysis levels (document, sentence, and aspect) and emotion models like Ekman and Plutchik for categorical and dimensional classification. Comparative analysis shows that machine learning algorithms such as SVM and Naïve Bayes achieve accuracies up to 87.17%, while deep learning models like CNN, LSTM, and BiLSTM outperform traditional methods, with CNN reaching up to 95.6% accuracy on multilingual tweets. Feature extraction techniques including TF-IDF, n-grams, Word2Vec, and Fast Text are discussed, with TF-IDF showing better performance in several cases. The review also emphasizes the importance of pre-processing and the role of datasets such as SST, SemEval, and EmoBank in benchmarking models. The paper concludes that hybrid models and transfer learning approaches, such as BERT and XLNet, further enhance classification performance, reaching accuracies as high as 87% in some cases, offering promising directions for future work.

Paper [14] proposes a novel sentiment analysis model for short texts called BM-ATT-BiLSTM, which integrates shallow learning features (e.g., emotional part-of-speech, word location, and dependency) with deep learning components like word embeddings, CNN features, and an emotional attention mechanism. The model was tested on NLPIR and NLPCC2014 datasets and compared with LSTM, BiLSTM, CNN, and CNN+SVM. BM-ATT-BiLSTM achieved the highest F1-scores of 88.76% and 73.05% on NLPIR and NLPCC2014, respectively, outperforming other models in precision and recall as well. The results confirm that combining emotional multichannel features with bidirectional LSTM and attention mechanisms significantly enhances short-text sentiment classification performance.

Paper [15] introduces the BiGRU-Att-HCNN model for sentiment analysis, integrating bidirectional gated recurrent units (BiGRU), self-attention, and hybrid CNN with dilated and depth wise separable convolutions. The model utilizes focal loss to address class imbalance and hard sample classification, and global average pooling (GAP) to reduce trainable parameters. Evaluated on IMDB, Yelp2013, and TSB datasets, the model achieved top performance with 92.63% accuracy on IMDB, 90.25% on Yelp2013, and 94.41% on TSB. Compared to 15 benchmark models, BiGRU-Att-HCNN consistently outperformed in accuracy, F1-score, and loss metrics while maintaining moderate parameter complexity, demonstrating its effectiveness in handling both balanced and imbalanced datasets with complex sentiment patterns.

Paper [16] proposes a hybrid sentiment analysis model combining Bidirectional LSTM (BiLSTM) with Multi-Head Attention (MHAT), tested on 19,465 Chinese product reviews from Taobao, categorized as positive, neutral, and negative. The model utilizes Word2Vec for word embedding and applies dropout, L2 regularization, and early stopping to mitigate overfitting. Experimental results show that the BiLSTM-MHAT model achieved the highest accuracy of 92.11%, outperforming CNN, BiLSTM, and BiLSTM with single attention by 2.10%, 1.31%, and 0.95%, respectively. Word2Vec also proved most effective among embedding techniques, yielding superior precision, recall, F1-score, and accuracy. These findings highlight the effectiveness of MHAT in capturing contextual nuances across multiple representation subspaces, significantly enhancing sentiment classification.

Paper [17] presents a comparative sentiment analysis on the IMDB movie review dataset using six models: Naive Bayes, Logistic Regression, LSTM, BiLSTM, Linear SVM, and Decision Tree. The dataset contains 10,000 balanced positive and negative reviews, split 80:20 for training and testing. Extensive preprocessing included stop word removal, lemmatization, and TF-IDF feature extraction. Among all models, BiLSTM achieved the highest performance with 91% accuracy, 89% precision, and 94% recall, followed closely by LSTM with 91% accuracy, 89% precision, and 92% recall. Logistic Regression and Linear SVM both reached 89% accuracy, while Naive Bayes and Decision Tree trailed at 86%

DOI: 10.17148/IJARCCE.2025.14421

and 70%, respectively. The study confirms the superior performance of deep learning models—especially BiLSTM—for capturing sentiment nuances in text data.

Paper [18] presents a novel approach to sentiment analysis of scholarly peer reviews using a Multiple Instance Learning model with an Abstract-based Memory mechanism (MILAM), evaluated on ICLR-2017 and ICLR-2018 datasets comprising a total of 4,392 reviews. The model jointly predicts the overall recommendation (accept/reject) and identifies sentence-level sentiment, using CNNs for sentence embeddings and LSTMs with document-level attention. Compared against traditional SVMs and state-of-the-art deep learning baselines (LSTM, CNN, CNN+Bi-LSTM+Attention, and MIL), MILAM achieved the highest accuracy across tasks: 80.32% for 2-class and 62.64% for 3-class classification on ICLR-2018, outperforming MIL by up to 3.75%. It also significantly improved sentence-level classification accuracy to 81.12% and showed robust cross-year and cross-domain performance. The study demonstrates that incorporating abstract information enhances sentiment signal extraction and confirms the model's effectiveness in aiding final decision prediction for scholarly papers.

Paper [19] explores sentiment and emotion analysis from Twitter text, introducing a unique dataset containing both tweets and their replies, enriched with user and interaction features. The study applies Naive Bayes classifiers for sentiment and emotion classification, achieving over 73% accuracy. It also computes user influence scores using a novel combination of tweet-based (retweets, replies, likes) and text-based features (sentiment, emotion, and agreement scores), resulting in precise identification of influential users. The research demonstrates that incorporating reply texts and novel scoring parameters enhances detection of sentiment/emotion networks, and employs k-means clustering for emotion-based user recommendation. This comprehensive approach advances user-targeted recommendation systems by connecting emotional expression with social influence.

III. PROPOSED METHODOLOGY

It is centred around a Long Short-Term Memory (LSTM) based deep learning architecture designed for sentiment classification using the IMDb movie review dataset. Each model processes pre-processed textual input through a sequence of neural network layers, optimized with different activation functions. The following components outline the general architecture and functionality of the proposed LSTM model:

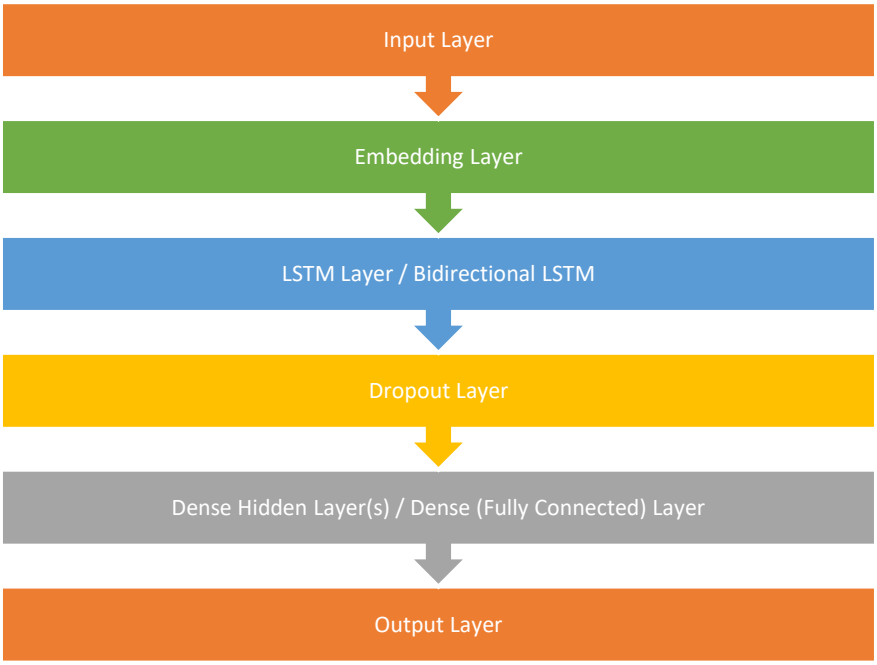

Fig 1. Proposed System (Flow Diagram)

Input Layer

The input layer receives the pre-processed text data, which has been tokenized and converted into sequences of word indices. To ensure uniformity across samples, each sequence is padded or truncated to a fixed length. These numerical sequences serve as the input for the subsequent layers of the model.

IJARCCE ISSN (O) 2278-1021, ISSN (P) 2319-5940

International Journal of Advanced Research in Computer and Communication Engineering

Impact Factor 8.102 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 14, Issue 4, April 2025

DOI: 10.17148/IJARCCE.2025.14421

Embedding Layer
The embedding layer transforms each word index into a dense, low-dimensional vector representation. This layer is essential for capturing semantic relationships between words, as similar words are mapped to similar vector spaces. The output is a matrix where each row represents the embedded vector of a word in the input sequence.

LSTM Layer / Bidirectional LSTM
The LSTM layer, the model's central component, receives the embedded sequences. In order to preserve long-term contextual information, this layer processes the sequence one piece at a time while keeping a memory cell and concealed state. Because of its gating features, LSTMs are able to learn dependencies over larger text spans than typical RNNs. A bidirectional LSTM, which processes input in both forward and backward directions, is utilized in various applications. This enhances the model's comprehension of intricate phrase constructions by enabling it to include data from both past and future context.

Dropout Layer
To prevent overfitting, a dropout layer is often applied after the LSTM layer. This layer randomly deactivates a percentage of neurons during each training step, which helps the model learn more generalizable features by not relying too heavily on specific patterns seen during training.

Dense (Fully Connected) Layer
Following the LSTM and dropout layers, one or more dense (fully connected) layers are added to further process the learned features. These layers apply non-linear transformations using activation functions such as ReLU, Tanh, or SELU, which enhance the model's ability to capture complex and abstract patterns from the LSTM output.

Output Layer
The final dense layer of the model consists of a single neuron with a Sigmoid activation function. The value that this layer outputs, which varies from 0 to 1, is the probability that the input text belongs to the positive emotion class. The sigmoid activation is a good technique for giving outputs a probability because sentiment analysis entails binary categorization.

This architecture serves as a flexible framework, allowing for experimentation with different activation functions in the dense layers, which is the core focus of this study. By isolating the activation functions as the main variable, the methodology ensures a fair comparison of their effect on model performance.

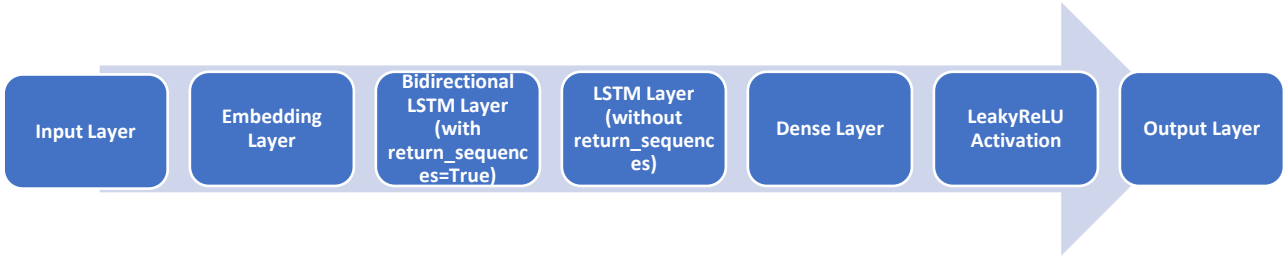

Fig 2. Layers of LSTM 1 Models (flow diagram)

The layers for the model are explained below:

Input Layer
The input layer defines the shape of the input sequences that will be fed into the model. In this case, each input sample is a fixed-length sequence of 200 tokens (word indices). These sequences are typically obtained after tokenizing the text and applying padding to ensure that all inputs are of equal length. The input layer acts as a placeholder for the model to accept batches of tokenized text data.

Embedding Layer
The embedding layer maps each word index in the input sequence to a dense vector of size 128. Here, `input_dim=5000` indicates that the vocabulary size is 5,000 (i.e., the model will consider the 5,000 most frequent words), and `output_dim=128` defines the size of each word vector. This transformation allows the model to capture semantic similarities between words by learning dense, continuous representations during training.
Bidirectional LSTM Layer (with return_sequences=True)

IJARCCE ISSN (O) 2278-1021, ISSN (P) 2319-5940

International Journal of Advanced Research in Computer and Communication Engineering

Impact Factor 8.102 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 14, Issue 4, April 2025

DOI: 10.17148/IJARCCE.2025.14421

This layer consists of a bidirectional LSTM with 64 units. It processes the embedded input in both forward and backward directions, allowing the model to gain a more comprehensive understanding of context in the sequence. The `return_sequences=True` parameter ensures that the full sequence of outputs (one for each time step) is passed to the next LSTM layer. Dropout and recurrent dropout rates of 0.4 are applied to help prevent overfitting by randomly dropping units and connections during training.

LSTM Layer (without return_sequences)

Following the bidirectional LSTM, this unidirectional LSTM layer with 32 units processes the sequence further. Since `return_sequences=False` by default, only the final hidden state is output from this layer. This condensed representation captures the most relevant contextual features from the previous sequence, which are then passed to the dense layers. Similar to the previous layer, dropout and recurrent dropout are applied for regularization.

Dense Layer

This fully connected dense layer consists of 64 neurons and serves as a feature-processing layer that operates on the output from the LSTM. At this stage, the model begins transforming the learned temporal features into a format more suitable for the final classification task. No activation is specified here because a separate activation function is added immediately afterward.

LeakyReLU Activation

This model uses LeakyReLU, which permits a tiny, non-zero gradient when the unit is not active (i.e., for negative inputs), in place of a conventional ReLU activation function. This solves the "dying ReLU" issue, in which zero gradients may cause certain neurons to cease updating. For negative inputs, the slope of the activation function is defined by the `alpha=0.01` parameter.

Output Layer

The final layer is a dense layer with a single neuron and a Sigmoid activation function, which maps the output to a probability between 0 and 1. This probability represents the likelihood that the input text belongs to the positive sentiment class. Since this is a binary classification problem (positive vs. negative sentiment), the sigmoid function is appropriate.

```
model = Sequential()
model.add(Input(shape=(200,)))
# Embedding layer
model.add(Embedding(input_dim=5000, output_dim=128))
# Bidirectional LSTM layer with return_sequences=True
model.add(Bidirectional(LSTM(64, return_sequences=True, dropout=0.4, recurrent_dropout=0.4)))
# LSTM layer without return_sequences
model.add(LSTM(32, dropout=0.4, recurrent_dropout=0.4))
# Dense layer followed by LeakyReLU activation
model.add(Dense(64))
model.add(LeakyReLU(alpha=0.01))  # alpha defines the slope of the negative part
# Output Dense layer with sigmoid activation for binary classification
model.add(Dense(1, activation='sigmoid'))
# Print model summary to verify the structure
model.summary()
```

Model: "sequential"

Layer (type)	Output Shape	Param #
embedding (Embedding)	(None, 200, 128)	640,000
bidirectional (Bidirectional)	(None, 200, 128)	98,816
lstm_1 (LSTM)	(None, 32)	20,608
dense (Dense)	(None, 64)	2,112
leaky_re_lu (LeakyReLU)	(None, 64)	0
dense_1 (Dense)	(None, 1)	65

Fig.3: Layers with their parameters of LSTM1

IJARCCE ISSN (O) 2278-1021, ISSN (P) 2319-5940

International Journal of Advanced Research in Computer and Communication Engineering

Impact Factor 8.102 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 14, Issue 4, April 2025

DOI: 10.17148/IJARCCE.2025.14421

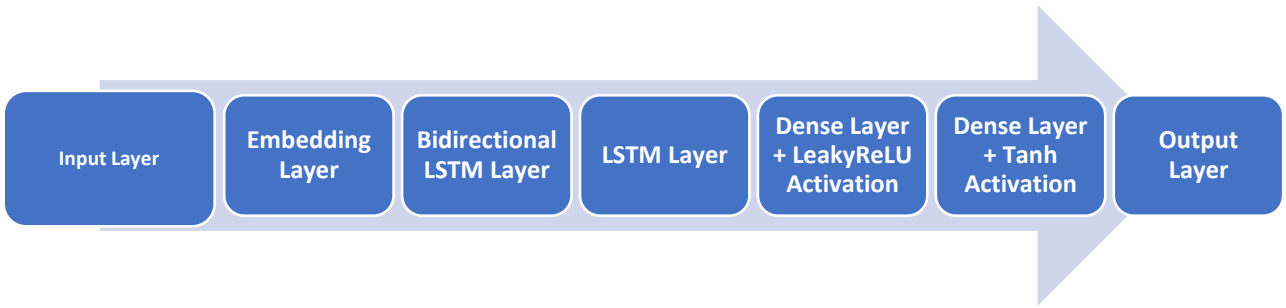

Fig 4. Layers of LSTM 2 Models

The layers for the model are explained below:

Input Layer
The input layer defines the expected shape of the incoming data. In this case, the model accepts a sequence of 200 integers per sample, representing tokenized and padded word indices from a text corpus. Each sequence corresponds to a processed movie review, ensuring consistency across the input dataset.

Embedding Layer
The embedding layer transforms each word index into a dense vector of 128 dimensions. The vocabulary size is set to 5,000, meaning the model considers the 5,000 most frequent words from the dataset. This layer captures semantic relationships between words by learning representations that place similar words closer together in vector space, improving the model's understanding of language structure.

Bidirectional LSTM Layer
This bidirectional LSTM layer processes the embedded sequence in both forward and backward directions. It contains 64 units and returns the full sequence of outputs for each time step by setting `return_sequences=True`. This allows the subsequent LSTM layer to access the full temporal structure of the input. The dropout and recurrent dropout rates (both set at 0.4) help mitigate overfitting by randomly deactivating units and connections during training.

LSTM Layer
Following the bidirectional layer, a unidirectional LSTM with 32 units processes the full sequence and outputs only the final hidden state. This state is a condensed representation of the sequence, capturing long-term dependencies and contextual patterns. It is passed forward for further processing in the dense layers.

Dense Layer + LeakyReLU Activation
A LeakyReLU activation function comes after this completely linked layer, which has 64 neurons. By permitting slight gradients for negative inputs, LeakyReLU keeps neurons from going dormant, a condition referred to as the "dying ReLU." This layer preserves gradient flow during backpropagation while assisting the model in learning non-linear transformations of the LSTM features.

Dense Layer + Tanh Activation
A second dense layer with 32 units is added to further refine the learned features. It is followed by a Tanh activation function, which outputs values in the range [-1, 1]. Tanh is beneficial for bringing the output back into a normalized scale, which can improve the model's stability and learning dynamics before the final classification step.

Output Layer
The last dense layer is perfect for binary classification because it has a single neuron with a Sigmoid activation function. The possibility that the input text conveys good sentiment is represented by the probability value that this layer outputs, which ranges from 0 to 1. Positive sentiment is indicated by a value around 1, whereas negative emotion is suggested by a value close to 0.

IJARCCE ISSN (O) 2278-1021, ISSN (P) 2319-5940

International Journal of Advanced Research in Computer and Communication Engineering

Impact Factor 8.102 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 14, Issue 4, April 2025

DOI: 10.17148/IJARCCE.2025.14421

```
# Define the model
model = Sequential()
# Add an Input layer to specify the input shape
model.add(Input(shape=(200,)))
# Embedding layer for converting integer sequences to dense vectors
model.add(Embedding(input_dim=5000, output_dim=128))
# Bidirectional LSTM layer with return_sequences=True to return the full sequence
model.add(Bidirectional(LSTM(64, return_sequences=True, dropout=0.4, recurrent_dropout=0.4)))
# LSTM layer (without return_sequences) to output the last hidden state
model.add(LSTM(32, dropout=0.4, recurrent_dropout=0.4))
# Dense layer followed by LeakyReLU activation for intermediate processing
model.add(Dense(64))
model.add(LeakyReLU(alpha=0.01))
# Add another Dense layer with 32 units before the Tanh activation
model.add(Dense(32))
model.add(Activation('tanh'))  # Tanh activation to normalize the output
# Output Dense layer with sigmoid activation for binary classification
model.add(Dense(1, activation='sigmoid'))
# Print model summary to verify the structure
model.summary()
```

Model: "sequential"

Layer (type)	Output Shape	Param #
embedding (Embedding)	(None, 200, 128)	640,000
bidirectional (Bidirectional)	(None, 200, 128)	98,816
lstm_1 (LSTM)	(None, 32)	20,608
dense (Dense)	(None, 64)	2,112
leaky_re_lu (LeakyReLU)	(None, 64)	0
dense_1 (Dense)	(None, 32)	2,080
activation (Activation)	(None, 32)	0
dense_2 (Dense)	(None, 1)	33

Fig 5: Layers with their parameters of LSTM 2

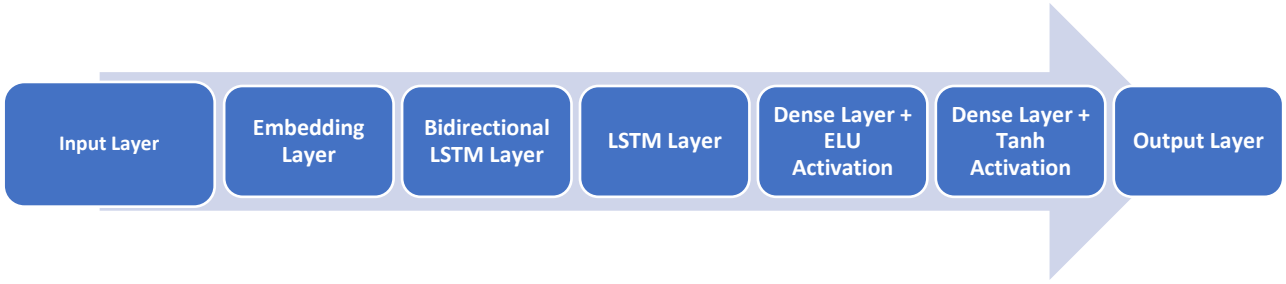

Fig 6. Layers of LSTM 3 Models

The layers for the model are explained below:

Input Layer

The input layer specifies the shape of each input sequence, which consists of 200 tokens. These tokens represent word indices generated through text preprocessing techniques like tokenization and padding. By fixing the input length, the model ensures consistency across all samples, enabling efficient training and batch processing.

Embedding Layer

The embedding layer transforms the integer-based input tokens into dense, trainable word vectors. With a vocabulary size of 5,000 and an embedding dimension of 128, each word index is mapped to a 128-dimensional vector that captures semantic relationships among words. This helps the model generalize linguistic patterns and contextual similarity.

Bidirectional LSTM Layer

This bidirectional LSTM layer gives the model full context at every time step by processing input sequences both forward and backward. It produces a series of concealed states with 64 units, allowing for a more thorough comprehension of phrase structure. By randomly turning off neurons and connections during training, dropout and recurrent dropout are both set at 0.4 to lessen overfitting.

IJARCCE ISSN (O) 2278-1021, ISSN (P) 2319-5940

International Journal of Advanced Research in Computer and Communication Engineering

Impact Factor 8.102 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 14, Issue 4, April 2025

DOI: 10.17148/IJARCCE.2025.14421

LSTM Layer

The second LSTM layer is unidirectional and consists of 32 units. It processes the output from the previous bidirectional layer and returns only the final hidden state (since `return_sequences=False`), effectively summarizing the sequence into a fixed-size context vector. This compact representation captures the overall meaning of the input text.

Dense Layer + ELU Activation

The dense layer with 64 units performs a linear transformation on the LSTM output. It is immediately followed by an Exponential Linear Unit (ELU) activation function. Unlike ReLU, ELU allows negative inputs to map to smooth, non-zero outputs, helping reduce bias shifts and speeding up convergence. The `alpha=1.0` parameter controls the saturation point for negative values. ELU is particularly effective in avoiding dead neurons and encouraging faster learning.

Dense Layer + Tanh Activation

Another dense layer with 32 units further refines the extracted features. It is paired with a Tanh activation function, which outputs values in the range [-1, 1]. This activation helps normalize the network's internal representations and makes the model more stable, especially when used before the output layer in classification tasks.

Output Layer

One neuron with a Sigmoid activation function makes up the last layer, which transforms the output into a probability score ranging from 0 to 1. The probability that the input falls into the positive sentiment class is indicated by this score. The sigmoid is ideally suited to generate a binary decision boundary because the task is binary classification.

```
# Define the model
model = Sequential()
# Add an Input layer to specify the input shape
model.add(Input(shape=(200,)))
# Embedding layer for converting integer sequences to dense vectors
model.add(Embedding(input_dim=5000, output_dim=128))
# Bidirectional LSTM layer with return_sequences=True to return the full sequence
model.add(Bidirectional(LSTM(64, return_sequences=True, dropout=0.4, recurrent_dropout=0.4)))
# LSTM layer (without return_sequences) to output the last hidden state
model.add(LSTM(32, dropout=0.4, recurrent_dropout=0.4))
# Dense layer followed by ELU activation for intermediate processing
model.add(Dense(64))
model.add(ELU(alpha=1.0))  # ELU activation, alpha controls the value to which ELU saturates for negative net inputs
# Add another Dense layer with 32 units before the Tanh activation
model.add(Dense(32))
model.add(Activation('tanh'))  # Tanh activation to normalize the output
# Output Dense layer with sigmoid activation for binary classification
model.add(Dense(1, activation='sigmoid'))
# Print model summary to verify the structure
model.summary()
```

Model: "sequential"

Layer (type)	Output Shape	Param #
embedding (Embedding)	(None, 200, 128)	640,000
bidirectional (Bidirectional)	(None, 200, 128)	98,816
lstm_1 (LSTM)	(None, 32)	20,608
dense (Dense)	(None, 64)	2,112
elu (ELU)	(None, 64)	0
dense_1 (Dense)	(None, 32)	2,080
activation (Activation)	(None, 32)	0
dense_2 (Dense)	(None, 1)	33

Fig 7: Layers with their parameters of LSTM3

The three LSTM-based models presented in this study share a common architecture consisting of an embedding layer, a bidirectional LSTM, a secondary LSTM, and multiple dense layers, but differ in their choice of activation functions used in the intermediate dense layers. First Model incorporates Leaky ReLU followed by Tanh, aiming to address the dying ReLU problem and ensure gradient flow while normalizing outputs. Second Model introduces ELU and Tanh, leveraging ELU's smooth negative output behaviour to improve learning dynamics and convergence speed. Third Model employs ReLU and ELU, focusing on faster training and reducing vanishing gradients. All models conclude with a sigmoid-activated output layer for binary classification. By comparing their performance on the IMDb sentiment analysis dataset, this study highlights the significant role that activation functions play in optimizing deep learning models for text classification tasks.

IJARCCE ISSN (O) 2278-1021, ISSN (P) 2319-5940

International Journal of Advanced Research in Computer and Communication Engineering

Impact Factor 8.102 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 14, Issue 4, April 2025

DOI: 10.17148/IJARCCE.2025.14421

IV. RESULTS AND DISCUSSION

This study aims to demonstrate the significant impact that activation function selection has on the performance of Long Short-Term Memory (LSTM) models in sentiment analysis tasks. The experiments are conducted using the IMDb movie review dataset and implemented in Google Collab with GPU acceleration for efficient training. The core architecture shared across all models includes an input layer that receives tokenized and padded sequences of 200-word indices, followed by an embedding layer that transforms these indices into 128-dimensional vectors using a vocabulary of the 5,000 most frequent words. A bidirectional LSTM layer with 64 units processes the input in both forward and backward directions and outputs the full sequence, which is then passed to a unidirectional LSTM layer with 32 units that returns only the final hidden state. Both LSTM layers employ dropout and recurrent dropout rates of 0.4 to mitigate overfitting. The output from the LSTM is passed through dense layers, where the models differ in their use of activation functions. The first model applies LeakyReLU followed by Tanh, aiming to maintain gradient flow and normalize outputs. The second model incorporates ELU and Tanh, leveraging ELU's smooth negative activation to improve learning stability and convergence. The third model uses ReLU followed by ELU to address vanishing gradients and accelerate training. All models conclude with a final dense layer activated by Sigmoid, which outputs the probability of the input expressing positive sentiment. Performance is evaluated using accuracy, loss, precision, recall, and F1 score to capture various aspects of model effectiveness, with the expectation that the second model—combining Sigmoid in the LSTM layers with ReLU in the output—will outperform the first model, which uses only Sigmoid activation.

TABLE 1: METRICS OF LSTM MODELS

IMDB Dataset				
Methods/Parameters	**Accuracy**	**F1 Score**	**Precision**	**Recall**
Proposed LSTM1	88.3%	88.8%	86.1%	91.7%
Proposed LSTM2	88.3%	88.3%	88.7%	88%
Proposed LSTM3	88.4%	88.3%	90.3%	86.3%

The performance evaluation conducted on the IMDb movie review dataset demonstrates that the three proposed LSTM models, each employing a unique combination of activation functions, perform competitively while exhibiting subtle yet meaningful differences across key classification metrics. Proposed LSTM1, which combines LeakyReLU followed by Tanh in its dense layers, achieved an accuracy of 88.3%, indicating a strong overall classification ability. With the highest F1 score of 88.8%, this model also demonstrated a performance that was well-balanced between recall and precision. Specifically, it obtained a precision of 86.1%, which measures the correctness of its positive predictions, and a recall of 91.7%, the highest among the three models, signifying its effectiveness in correctly identifying most of the actual positive cases. Because of this, LSTM1 is especially well-suited for uses like very sensitive negative sentiment detection where reducing false negatives is crucial.

Proposed LSTM2, which uses a combination of ELU and Tanh activation functions, also achieved a commendable accuracy of 88.3%, similar to LSTM1. However, its F1 score slightly decreased to 88.3%, indicating a marginally less balanced trade-off between precision and recall. Interestingly, LSTM2 demonstrated the highest precision at 88.7%, suggesting that it was the most reliable in its positive predictions, while maintaining a recall of 88%, which reflects a solid ability to detect most positive instances. This model showcases a more conservative classification approach—favoring fewer false positives—making it suitable for scenarios where the cost of misclassifying negative instances as positive is high.

Proposed LSTM3, featuring a ReLU and ELU activation combination, outperformed the other models in terms of raw accuracy, achieving 88.4%, the highest among the three. Its F1 score matched LSTM2 at 88.3%, indicating a consistent balance between precision and recall. This model excelled in precision, achieving 90.3%, the highest precision score in the study, which means it made the fewest false-positive predictions. However, this came at the cost of a slightly lower recall of 86.3%, suggesting that it missed more actual positive cases than the other two models. LSTM3 is thus best suited for tasks where making accurate positive predictions is more important than identifying every single positive case.

While all three models achieved very similar accuracies around 88.3–88.4%, their performance profiles varied based on the activation functions used. LSTM1 offered the best recall and overall balance, LSTM2 emphasized precision without significantly sacrificing recall, and LSTM3 maximized precision at the cost of slightly lower recall. These findings underscore the importance of activation function selection in fine-tuning model behavior for sentiment analysis, especially when specific performance metrics are prioritized depending on the application context.

IJARCCE ISSN (O) 2278-1021, ISSN (P) 2319-5940

International Journal of Advanced Research in Computer and Communication Engineering

Impact Factor 8.102 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 14, Issue 4, April 2025

DOI: 10.17148/IJARCCE.2025.14421

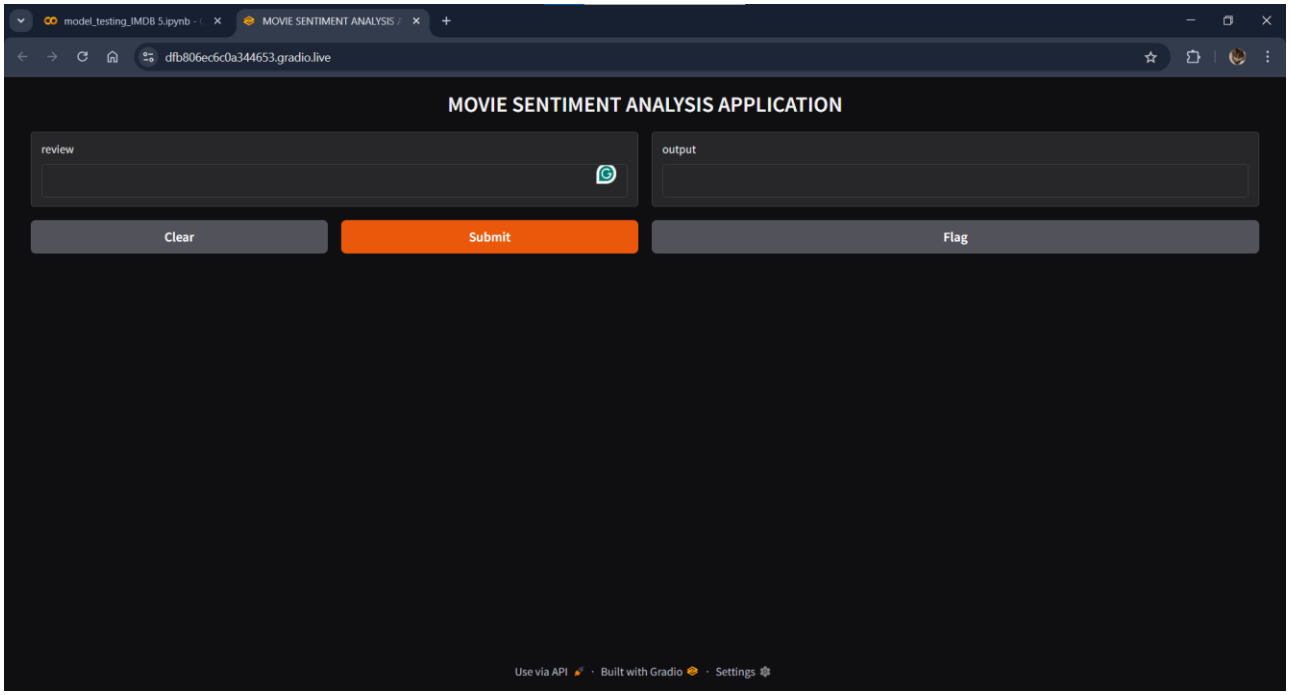

Fig 8: Web app for the proposed model

The image showcases a web-based user interface for a Movie Sentiment Analysis Application, developed using the Gradio framework. This application enables users to input textual movie reviews into a dedicated input field labelled "review" and receive real-time sentiment predictions—such as positive or negative—in the adjacent "output" field. The interface features a sleek, dark-themed design and includes three primary action buttons: "Clear" to reset the input field, "Submit" to process the review through the underlying deep learning model, and "Flag" to report any inappropriate or incorrect outputs. Positioned at the top is a prominent title that clearly communicates the application's function, while the bottom of the interface includes metadata indicating that the tool was built with Gradio and offers support for API integration. The layout is simple yet functional, providing an intuitive and efficient environment for users to interact with the sentiment analysis model. This kind of interface is particularly valuable for research, experimentation, and demonstration purposes, allowing seamless testing and evaluation of natural language processing models in a user-friendly format.

V. CONCLUSION

In this study, we investigated the performance of three LSTM-based deep learning models for sentiment analysis using the IMDb movie review dataset. Each model was designed with a unique combination of activation functions—ReLU, ELU, Tanh, SELU, and Leaky ReLU—to evaluate their impact on model accuracy, precision, recall, F1 score, and loss. While all three models achieved similar overall accuracies in the range of 88.3% to 88.4%, subtle yet meaningful differences were observed across other metrics. The first model, which employed LeakyReLU and Tanh, achieved the highest F1 score and recall, making it more suitable for applications where identifying all positive sentiment instances is critical. The second model, incorporating ELU and Tanh, recorded the highest precision, indicating strong reliability in its positive sentiment predictions. The third model, using ReLU and ELU, achieved the best overall accuracy and precision, though it slightly lagged in recall. These results highlight the significant influence of activation function choice in tuning LSTM models for specific sentiment analysis goals. While the overall architecture remained consistent across all three models, the strategic use of different activation functions led to performance variations that can inform future model selection based on application needs. The findings emphasize that even small architectural adjustments—such as activation function pairing—can meaningfully affect how a model performs in terms of generalization, sensitivity, and classification confidence.

A number of interesting avenues to improve the efficacy and relevance of LSTM-based sentiment analysis models are included in the research's future scope. Adding attention methods could improve the model's contextual awareness by assisting it in concentrating on the most pertinent portions of input sequences. Accuracy may be greatly increased by using pre-trained transformer models like BERT or RoBERTa, especially on bigger or more complicated datasets. Expanding the model to handle multi-class or aspect-based sentiment classification can increase its usefulness in real-

DOI: 10.17148/IJARCCE.2025.14421

world applications like customer feedback systems and product reviews. Additionally, adapting the models for multilingual or code-mixed data would enhance their global applicability. Improving model interpretability using tools such as SHAP or LIME could provide valuable insights into decision-making processes, which is especially important in sensitive use cases. Finally, optimizing these models for real-time deployment on mobile or edge devices would enable integration into systems like chatbots and recommendation engines, making them more responsive and user-friendly.

REFERENCES

[1]. N. Ouyang, "Analyze IMDb movies by sentiment and topic analysis," Environment and Social Psychology, vol. 8, no. 3, Oct. 2023, Doi: https://doi.org/10.54517/esp.v8i3.1958.

[2]. G. Dubey et al., "A Hybrid Convolutional Network and Long Short-Term Memory (HBCNLS) model for Sentiment Analysis on Movie Reviews," International journal on recent and innovation trends in computing and communication, vol. 11, no. 4, pp. 341–348, May 2023, Doi: https://doi.org/10.17762/ijritcc.v11i4.6458.

[3]. S. Yin and G. Zhong, "TextGT: A Double-View Graph Transformer on Text for Aspect-Based Sentiment Analysis," Proceedings of the ... AAAI Conference on Artificial Intelligence, vol. 38, no. 17, pp. 19404–19412, Mar. 2024, Doi: https://doi.org/10.1609/aaai.v38i17.29911.

[4]. Sri Utami, Kemas Muslim Lhaksmana, and Yuliant Sibaroni, "Deep Learning and Imbalance Handling on Movie Review Sentiment Analysis," SinkrOn, vol. 8, no. 3, pp. 1894–1907, Jul. 2023, Doi: https://doi.org/10.33395/sinkron.v8i3.12770.

[5]. Z. Yin et al., "DPG-LSTM: An Enhanced LSTM Framework for Sentiment Analysis in Social Media Text Based on Dependency Parsing and GCN," Applied sciences, vol. 13, no. 1, pp. 354–354, Dec. 2022, Doi: https://doi.org/10.3390/app13010354.

[6]. K. L. Tan, C. P. Lee, and K. M. Lim, "A Survey of Sentiment Analysis: Approaches, Datasets, and Future Research," Applied Sciences, vol. 13, no. 7, p. 4550, Apr. 2023, Doi: https://doi.org/10.3390/app13074550.

[7]. Dr. G. S. N. Murthy, Shanmukha Rao Allu, Bhargavi Andhavarapu, and Mounika Bagadi, Mounika Belusonti, "Text based Sentiment Analysis using LSTM," International Journal of Engineering Research and, vol. V9, no. 05, May 2020, Doi: https://doi.org/10.17577/ijertv9is050290.

[8]. O. Hourrane, E. H. Benlahmar, and A. Zellou, "Comparative study of deep learning models for sentiment analysis," International Journal of Engineering & Technology, vol. 7, no. 2.14, p. 5726, Apr. 2018, Doi: https://doi.org/10.14419/ijet.v7i4.24459.

[9]. C. N. Dang, M. N. Moreno-García, and F. De la Prieta, "Hybrid Deep Learning Models for Sentiment Analysis," Complexity, vol. 2021, no. 9986920, pp. 1–16, Aug. 2021, Doi: https://doi.org/10.1155/2021/9986920.

[10]. N. C. Dang, M. N. Moreno-García, and F. De la Prieta, "Sentiment Analysis Based on Deep Learning: A Comparative Study," Electronics, vol. 9, no. 3, p. 483, Mar. 2020, Doi: https://doi.org/10.3390/electronics9030483.

[11]. L. Suganya, "Regular paper Comparative Analysis of Deep Learning Models for Sentiment Analysis on IMDB Reviews," Deleted Journal, vol. 20, no. 2s, pp. 424–433, Apr. 2024, Doi: https://doi.org/10.52783/jes.1345.

[12]. E. Haddi, X. Liu, and Y. Shi, "The Role of Text Pre-processing in Sentiment Analysis," Procedia Computer Science, vol. 17, pp. 26–32, 2013, Doi: https://doi.org/10.1016/j.procs.2013.05.005.

[13]. P. Nandwani and R. Verma, "A review on sentiment analysis and emotion detection from text," Social Network Analysis and Mining, vol. 11, no. 1, Aug. 2021, Doi: https://doi.org/10.1007/s13278-021-00776-6.

[14]. Z. G. Zhou, "Research on Sentiment Analysis Model of Short Text Based on Deep Learning," Scientific Programming, vol. 2022, no. 2681533, pp. 1–7, May 2022, Doi: https://doi.org/10.1155/2022/2681533.

[15]. Q. Zhu, X. Jiang, and R. Ye, "Sentiment Analysis of Review Text Based on BiGRU-Attention and Hybrid CNN," IEEE Access, vol. VOLUME 9, 2021, pp. 1–1, 2021, Doi: https://doi.org/10.1109/access.2021.3118537.

[16]. F. Long, K. Zhou, and W. Ou, "Sentiment Analysis of Text Based on Bidirectional LSTM With Multi-Head Attention," IEEE Access, vol. 7, pp. 141960–141969, 2019, Doi: https://doi.org/10.1109/access.2019.2942614.

[17]. S. K. Singh and N. Singla, "Sentiment Analysis on IMDB Review Dataset," Journal of Computers, Mechanical and Management, vol. 2, no. 6, pp. 18–29, Dec. 2023, Doi: https://doi.org/10.57159/gadl.jcmm.2.6.230108.

[18]. K. Wang and X. Wan, "Sentiment Analysis of Peer Review Texts for Scholarly Papers," Jun. 2018, Doi: https://doi.org/10.1145/3209978.3210056.

[19]. K. Sailunaz, "Emotion and Sentiment Analysis from Twitter Text," Handle.net, Jul. 27, 2018. http://hdl.handle.net/1880/107533 (accessed Apr. 12, 2025).

