SentivarLSTM: STUDENT SENTIMENT VARIATION ANALYSIS THROUGH DATASET NORMALIZATION

1.Mr. M. Subramaniakumar, 2.Dr. D. Maheswari
1Research Scholar, Rathnavel Subramaniam College of Arts and Science, Sulur
2Head & Research Coordinator Rathnavel Subramaniam College of Arts and Science, Sulur

Abstract
Data complexity, volume, and inconsistencies have made student performance prediction a more daunting endeavour. This paper introduces a novel approach, SentiVarLSTM, designed specifically for analyzing sentiments in student-generated textual data, with a focus on student sentimentanalysisandfeedbackevaluation.Thisresearchusesthreetypesofeducationaldatasets for student sentiment analysis. SentiVarLSTM employs Sentiment Variation Analysis through data set normalization, employing essential pre-processing steps such as tokenization, lowercasing, stop words removal, and text cleaning. This model offers valuable insights into the sentiment dynamics of student feedback, aiding educators and researchers in understanding and improving the overall student experience.
Keywords: Dataset Normalization, Part-of-Speech, Tokenization, Stop Words Removal, SentiVarLSTM

I. INTRODUCTION

Sentiment analysis, a pivotal aspect of natural language processing, plays a crucial role in deciphering the underlying emotions and opinions embedded in textual data [1]. As the field evolves, addressing the nuances of sentiment variation becomes paramount for accurate and nuanced analyses [2]. This paper introduces a novel methodology, referred to as Sentiment Variation Analysis through data set normalization(SentiVarLSTM)[3].By a malgamatin advanced techniques such as tokenization, lowercasing, stop words removal, and PoS tagging using LSTM neural networks, SentiVarLSTM aims to provide a comprehensive solution to sentimentanalysis[4-6].Thekeyinnovationliesinthesubsequentsteps:thecalculationofweights for PoS tags, extraction of opinion words through WordNet, and the assignment of sentiment scores to tokens [7-9]. Importantly, the normalization of sentiment scores to a common scale enhances the interpretability and comparability of results.

Leveraging a Long Short-Term Memory (LSTM) neural network, the methodology includes Part-of-Speech (PoS) tagging to capture contextual information crucial for understanding student sentiments. To enhance the student sentiment analysis process, weights for PoS tags are computed based on their relevance in determining sentiments, and opinion words are identified using WordNet.

Through this innovative approach, SentiVarLSTM endeavors to advance the field of sentiment analysis by effectively addressing variation in sentiment across diverse datasets [10-13]. Subsequently, sentiment scores are assigned to each token, taking into account both PoStag weights and identified opinion words. To ensure interpretability and enable meaningful comparisons, sentiment scores are normalized to a common scale. SentiVarLSTM presents a robust framework tailored for student sentiment analysis, effectively addressing variations in student-generated text through comprehensive dataset normalization.

Sentimentanalysishasemergedasacrucialaspectofnaturallanguageprocessing, aiming to decipher the emotions and opinions expressed in textual data [14]. In the vast landscape of sentiment analysis, understanding the variation in sentiments across different datasets is imperative for accurate and robust model performance [15]. This paper introduces SentiVarLSTM, a specialized approach for analyzing sentiments in student-generated textual data, aiming to enhance student sentiment analysis and feedback evaluation. By employing Sentiment Variation Analysis through dataset normalization and leveraging a LSTM neural network, the methodology incorporates essential pre-processing steps and PoStagging to capture contextual information [16].

The model calculates weights for PoStags and identifies opinion words using WordNet, assigning sentiment scores to each token. Normalization of these scores ensures interpretability and facilitates meaningful comparisons [17]. SentiVarLSTM offers a robust framework for student sentiment analysis, demonstrated through rigorous experiments and comparisons with existing methods, providing valuable insights for educators and researchers to enhance the overall student experience [18]. Dataset normalization plays a pivotal role in this process, as this helps standardize diverse datasets, making them comparable and enhancing the reliability of sentiment analysis outcomes. However, the inherent heterogeneity in language usage, cultural nuances, and context across different datasets poses a challenge for creating universal sentiment analysis models [19-22]. This variability can lead to biased predictions and reduced model generalization when applied to new and diverse datasets. Dataset normalization is the process of standardizing datasets by applying various techniques to mitigate these variations. This involves addressing differences in language style, vocabulary, and sentiment expression within and across datasets. Through effective normalization, the goal is to create a level playing field for sentiment analysis models, enabling them to perform consistently across different data sources [23-25].

The main contribution of the paper is:

- Dataset preprocessing using SentiVarLSTM

What follows is the outline for the rest of the article. Section 2 has several writers discussing various student sentiment analysis methodologies. In Section 3, the SentiVarLSTM model has been represented. The investigation's findings are summarized in Section 4. A discussion of the outcome and potential future research makes up Section 5.

II. BACKGROUNDSTUDY

Abid, F., et al [1] The author wrap up these authors work with concrete results; to begin, the author used POS tagging and domain-specific pre-processing on multi-source datasets to improve distributed word representations' efficiency; these representations now outperform vocabulary sense. Word2Vec, GloVe, and fast Text were just a few of the word representations that the author have covered, and the author have also shown how to combine these representations using a weighted mechanism instead of using a single learning model.

An, H., et al [3] Important information for the tourism industry could be derived from the ability to assess tourist attractions using environmental variables like time of year, weather, and season. This information could pave the way for better sightseeing and travel suggestions. It was suggested that a platform without ratings but with more information and reviews would be beneficial as the present platforms that collect site reviews do not have enough data.

Guo, Y. et al [7] after analyzing and quantifying public emotion using natural language processing technical tools, this paper proposes a multilayer's LSTM model to forecasts to ckprice and return. It then compares the model's prediction effect using two inputs: one based only on historical stock technical indicators and the other with news sentiment score as well. A model that incorporates sentiment scores from news articles into its stock prediction process outperforms a model that solely uses stock time-series data, according to experimental results based on three high cap companies (Amazon, Microsoft, and Tesla). This improved model achieves lower MSE and MAE values.

Hemanth, J., et al [9] it was possible to sum up this study by saying that QR codes were rapidly becoming widely used. As a result, the author can empower the end-user by developing a safe and automated approach to decrease the prevalence of counterfeit drugs in the market. The suggested system's strength was the sense of security it provides through the use of hashing, which prevents unauthorized users from accessing the original metadata and also uses very little system resources.

Kaiser, M. S., et al [11] Here, the author provide a bigram and bidirectional LSTM-based approach to detecting and fixing Bangla real-world errors. Due to the lack of sufficient standard resources, the dataset utilized here has a significant influence on the model training process. On the other hand, this suggested method has fixed the real-world error with an accuracy of 82.86%, which occurs when a word was grammatically correct but alters the original meaning of a statement.

Ombabi,A.H.,etal[13] there has been a meteoricrise in User-Generated Content(UGC) on social media platforms in recent years, and this content provides incredibly useful information for many different kinds of uses. Analyzing social data to determine the public's preferences was the main focus of sentiment analysis. Notwithstanding careful examination of syntactic and semantic norms, as well as the word dependencies of the input sentence, sentiment analysis for Arabic was difficult to do.

Sangeetha, K., et al [15] the author used the feedback from the Vietnamese students to inform these authors work. The author presented a technique that combines many heads with embedding and long short-term memory layers. The baseline models were out performed by these authors fusion model. With these authors model, the author can improve performance for longer sentence sequences while simultaneously paying close attention or referral words. The decrease in the quantity of neutral sentences has an effect, as the author has seen. The author intends to use the fusion model to run experiments on several datasets containing student comments.

2.1Problem definition

This paper addresses the challenge of analyzing sentiments in student-generated textual data, focusing on sentiment analysis and feedback evaluation. It introduces SentiVarLSTM, a novel approach that employs Sentiment Variation Analysis through dataset normalization. The method tackles issues such as variations in student-generated text through essential pre-processing steps, leveraging a LSTM neural network and PoS tagging to capture contextual information crucial for understanding student sentiments. The goal is to enhance sentiment analysis by computing weights for PoS tags and identifying opinion words using WordNet, ultimately providing educators and researchers with valuable insights to understand and improve the overall student experience.

III. EXISTING METHODOLOGIES

In this section, this research outlines the workflow employed in our study on student sentiment analysis using the SentiVarLSTM approach. Leveraging insights from two distinct datasets obtained from Kaggle, this research present a comprehensive overview of our methodology, encompassing dataset preprocessing, the application of LSTM neural networks, and the integration of PoS tagging for contextual analysis.

Dataset collection

In this section, this research utilizes three different educational datasets for student sentiment analysis. The first dataset, collected from the Kaggle website (https://www.kaggle.com/datasets/jayaprakashpondy/student-feedback), comprises a CSV file with a size of 4.28MB.The dataset is distributed across four files, with the primary data stored in "feedback_student.csv." This file consists of three attributes and a total of 2346 records. The second dataset, obta

186 records. The third dataset was obtained from Export Comments as well and can be accessed through the following link: https://exportcomments.com/done/b273e2b8-b359-4d66-9a11-6d7f05527904. We collected the educational video links from YouTube for analysis purposes.

Dataset preprocessing using SentiVarLSTM

LSTM

LSTMs are particularly effective in natural language processing tasks and sequential data analysis. Unlike traditional RNNs referred by Shaik et al. (2023), LSTMs incorporate memory cells, allowing them to capture and remember information for extended periods, making them well-suited for tasks requiring an understanding of context and dependencies over time. This architecture has proven valuable in applications such as language modeling, machine translation, and sentiment analysis, where contextual understanding and long-range dependencies are crucial for accurate predictions and classifications.

OneofthemainreasonsLSTMnetworksaresovaluableintheapplicationsmentionedby C. Li, G. Zhan and Z. Li (2018) is their capacity to effectively capture the long-term associations between sequences. While their performance is improved in numerous applications due to their capacity to perceive contextual information from bidirectional view points, implementation issues including computational intensity and over fitting should be taken into account. When it comes to jobs that call for a sophisticated grasp of sequential data, LSTM networks are a solid bet.

Improved upon the original LSTM design is the two-layer LSTM network architecture. Sunlight irradiance is another kind of time-series data type. The use of LSTM networks in irradiance prediction is still limited. In this study, this research shall prove that it works in this field.

$h_t=(UX_t+Wh_{t-1})$ -------------------------(1)

$h'=f(U'X+W'h$

) -----------------(2)

t t $t+1$

- h_t: Hidden state at time t, representing the memory of the network at that time.
- f: Activation function.
- U: Weight matrix for the input.
- X_t: Input at time t.
- W: Weight matrix for there current connections.
- h_{t+1}: Hidden state from the previous time step.

O_t=($V'h'_t$+Vh_t) ---------------------------- (3)

- O_t:Output at time t.
- g: Activation functions for the output.
- V': Weight matrix for the backward layer output.
- V:Weight matrix for the forward layer output.
- h': Hidden state in the backward layer.
- h_t:Hidden state in the forward layer.

$$s=n\ t=1\ \frac{\Sigma}{(y_t-O_t)}\quad n- \qquad (4)$$

- s: Loss or error, calculated as the average difference between the true values $y_t$and the predicted values $O_t$over the sequence.
- n:Number of ime steps.

The only data that LSTM model scan utilize to predict future outputs is data from previous time sequences. A number of challenges, nevertheless, bind the current production to both the past and the future. The LSTM receives the structure as its input. The set $X_{0,1},...,X_N$.By calculating the forward layer from moments1toN,this research is able to acquire and record the output from the implied layer at every instant. Starting at moment N and continuing all the way to moment 1, the backward layer performs computations backwards, burying the result of each moment in the opposite direction. The results of this study are obtained by summing the outputs of the forward and backward layers at each time step. The values of $Y_{0,1},...,Ye_N$.

The well-liked LSTM network design out performs conventional RNNs in capturing cross-temporal relationships. The long-term dependence issue is fixed and RNN's ability to identify and benefit from dependencies in long-distance data is improved with this upgrade. LSTM RNNs employs internal gates to solve the gradient issue. Being are cursive neural network, LSTM takes in data from one unit and uses it to train other units. In order to finish the classification process, the buried layer's output is used.

Tokenizing LSTM

Tokenizing LSTM refers to the application of a LSTM neural network in the context of tokenization, a fundamental step in natural language processing. This approach involves leveraging the sequential learning capabilities of LSTMs to tokenize input text, breaking it down into individual words or tokens. The model learns the sequential patterns and dependencies between words in a given context, enhancing the tokenization process's effectiveness. By employing Tokenizing LSTM, the model gains the ability to understand the hierarchical and sequential structure of language, contributing to more accurate and context-aware tokenization in various NLP applications.

The calculation formula is shown in Equation

$f_t=(W_{(f)}x_t+U_{(f)}h_{t-1}+b_{(f)})$ -------------------------------(5)

$i_t=(W_{(i)}x_t+U_i h_{t-1}+b_{(i)})$ -------------------------(6)

$c_t=\tilde{c}+i_t\odot \tan h(W_{(c)}x_t+U_{(c)}h_{t-1}+b_{(c)})$----------------------------- (7)

$\tilde{c}=f_t\odot c_{t-1}$ ---------------------- (8)

$h_t=O_t\odot \tan h(C_t)$ ---------------------- (9)

At time t, the input transforms into x_t, and F, the sigmoid activation function, which is the inverse of element multiplication, is represented by it. In order to prevent gradient disappearance or explosion, Tokenizing LSTM makes use of c_t to learn longer information dependency. One kind of reset memory unit is the forget gate f_t, which regulates the input and output of the unit via its input and output gates, O_t and f_t, respectively.

Tokenizing LSTM has the limitation of not being able to store data in a reversed manner. In order to access both the forward and backward features simultaneously, the bidirectional tokenizing LSTM network was proposed. Improving and speeding up learning while supplying more context information is achieved by combining forward and backward Tokenizing LSTM into Tokenizing.

$p = soft(W_c s + b_c)$ ------------------------------ (10)

Algorithm1:TokenizingLSTM
Input: • X_0, X_1,..., X_N are time series data: Stock prices and other time-dependent data are examples of sequential input data. 1. Forward Layer Computation (h_t): o$h_t = f(UX_t + Wh_{t-1})$ 2. Backward Layer Computation (h'_t): o$h'_t = f(U'X_t + W'h'_{t+1})$ 3. Network Output Calculation(O_t): o$O_t = g(V'h'_t + Vh_t)$ 4. Mean Squared Error Calculation(s): o$s = \frac{\sum_{t=1}^{n}(y_t - O_t)}{n}$ 5. Bidirectional Computation of Outputs(Y_t): o$h_t = O_t \odot \tanh(C_t)$ 6. Final Classification using Softmax(p): o$p = softmax(W_c s + b_c)$ **Output:** • Results at the end(Y_0,Y_1,...,Y_N):Values that the Tokenizing LSTM network predicts at each time step.

3.3.3Termfrequency–inversedocument frequency

There are three main schools of thought when it comes to text mining: data mining, knowledge discovery in databases, and information extraction. Such methods are widely employed in many fields of research, including economics referred by Al-Obaydy et al. (2022). TF-IDF is the most popular statistical method for determining a document's relative importance to the corpus in terms of a single word.-IDF.

To calculate TF-IDF, this research multiplies the two statistics (Eqs. (11) and (12)) together, yielding TF-IDF as provided by Eq. (13).

$(t,d) = \frac{f_{t,}}{|\{f_{t'},d : t' \in d\}|}$ ---------- (11)

$IDF(t,D)=log \frac{|D|}{|\{d\in D:t\in d\}|}$ - (12)

$TF-(t,d,D)=TF(t,d)\times IDF(t,D)$-------------------------------- (13)

Financial institutions utilize TF-IDF to filter through textual data from several news, social media, blog, and other platforms for accurate market research and stock prediction. In this study, this research proposes combining historical stock market data and the TF-IDF distance measure to generate a feature weight matrix. This research conducts our tests in Python using the Scikit-learn module, and this research derives the TF-IDF values with Tf-idf Vectorizer from the sklearn. feature_extraction.text package. This research combines TF-IDF with neural networks, and presents a short summary of relevant past work to highlight the relevance of the proposed approach in the financial market.

$(t,d)=0.5+0.5.\frac{ft,}{max\{f_{t',d}:t'\in d\}}$---------- (14)

Whereas the Inverse Document Frequency shifts the emphases is from often occurring words to seldom occurring terms. If a word is frequent or uncommon across all papers, its frequency in the inverse document used as a proxy for the value of the information it conveys. This is the inverse percent of text documents on a logarithmic scale. The inverse document frequency equation is shown in (15).

$idf(t,D)=log \frac{N}{|\{d\in D:t\in d\}|}$- (15)

Finally, this research gets TF-IDF by solving (16).In TF-IDF, high weights are backed by words and documents with low TF-IDF values; as a result, weights must screen out generic phrases.

$(t,d,D)=tf(t,d).idf(t,D)$ --------------------------- (16)

Algorithm2:TF-IDF
Input: • Document corpus(D)containing multiple documents • List of terms(t)to calculate TF-IDF scores for • Preprocessed document-term frequency matrix$tf(t,d)$for terms in documents • Maximum frequency of any term in a document(max $\{tf(t,d)$) **Steps:** 1.Calculatethetermfrequency(TF) for each term in each document using:

$TF(t, d) = \frac{f_{t,d}}{|\{f_{t'},d : t' \in d\}|}$

$IDF(t,D) = log \frac{|D|}{|\{d' \in D : t \in d'\}|}$

$(t,d) = 0.5 + 0.5.\frac{f_{t,d}}{max\{f_{t',d} : t' \in d\}}$

Output:

- TF-IDF scores for each term $tf(t,d)$

3.3.4 SentiVarLSTM

Student Sentiment Analysis using SentiVarLSTM involves a multi-step approach to deciphering sentiment nuances in textual data. The process begins with dataset pre-processing, including tokenization, lowercasing, stop words removal, and text cleaning to ensure consistency and eliminate irrelevant information. The unique contribution of SentiVarLSTM lies in its application of a LSTM neural network for PoS tagging during this pre-processing phase, capturing intricate contextual information. Subsequently, the model calculates weights for PoS tags based on their relevance in sentiment determination. Leveraging WordNet, a lexical database, opinion words are identified, enriching the understanding of sentiment expressions. Sentiment scores are then assigned to each token, incorporating both PoStag weights and the presence of opinion words. Notably, SentiVarLSTM's architecture spans 19 layers, augmenting its capacity to comprehend and analyze complex sentiment variations. By virtue of its comprehensive dataset normalization and sophisticated methodology, SentiVarLSTM emerges as a robust framework for sentiment analysis, rendering it invaluable for deciphering and interpreting sentiments across a spectrum of textual contexts.

TheSentiVarLSTM model and structure are two of the sophisticated recurrent architectures proposed as remedies for the aforementioned RNN issues. When faced with problems requiring sequential solutions and with long-term limitations, SentiVarLSTM is said to perform well. Although several popular SentiVarLSTM variants have been developed in recent years, a comprehensive evaluation of these variants shows that none of them significantly outperform the basic SentiVarLSTM pattern. So, the normal SentiVarLSTM design is part of the suggested topology in this study; this research get into that further later.

A typical SentiVarLSTM design differs from RNN architecture primarily in the presence of hidden units. The SentiVarLSTM layer makes frequent references to a large number of the hidden nodes in SentiVarLSTM. Like RNNs, SentiVarLSTMs use a Memory unit that takes in input at $x_t$and outputs a value, ht, at t, where t is the target time. The architecture consists of 19 layers, including input layers, convolutional layers, max-pooling layers, dropout layers, and concatenation layers. These layers collectively contribute to the model's ability to effectively capture and analyze sentiment dynamics in student-generated textual data.

Convolution operation:= W_x+b,where$z$ is the output, W is the weight matrix, x is the input, and b is the bias.

Maxpooling: Selects the maximum value from a window.

Dropout: During training, randomly sets a fraction of input units to zero.

Upsampling: Increases the spatial dimensions of the feature maps.

These layers collectively contribute to the feature extraction and classification capabilities of the SentivarLSTM model, making it suitable for analyzing student sentiment in educational datasets. The model architecture consists of 19 layers, including convolutional, maxpooling, dropout, and upsampling layers, which enable effective feature extraction and sentiment analysis. During training and character flipping, the complex unit examines the output of the previous cell(c_t), the currentcell (C_t), and the priorunitcell (h_{t-1}) out pdate the outpu to the Previous cell.

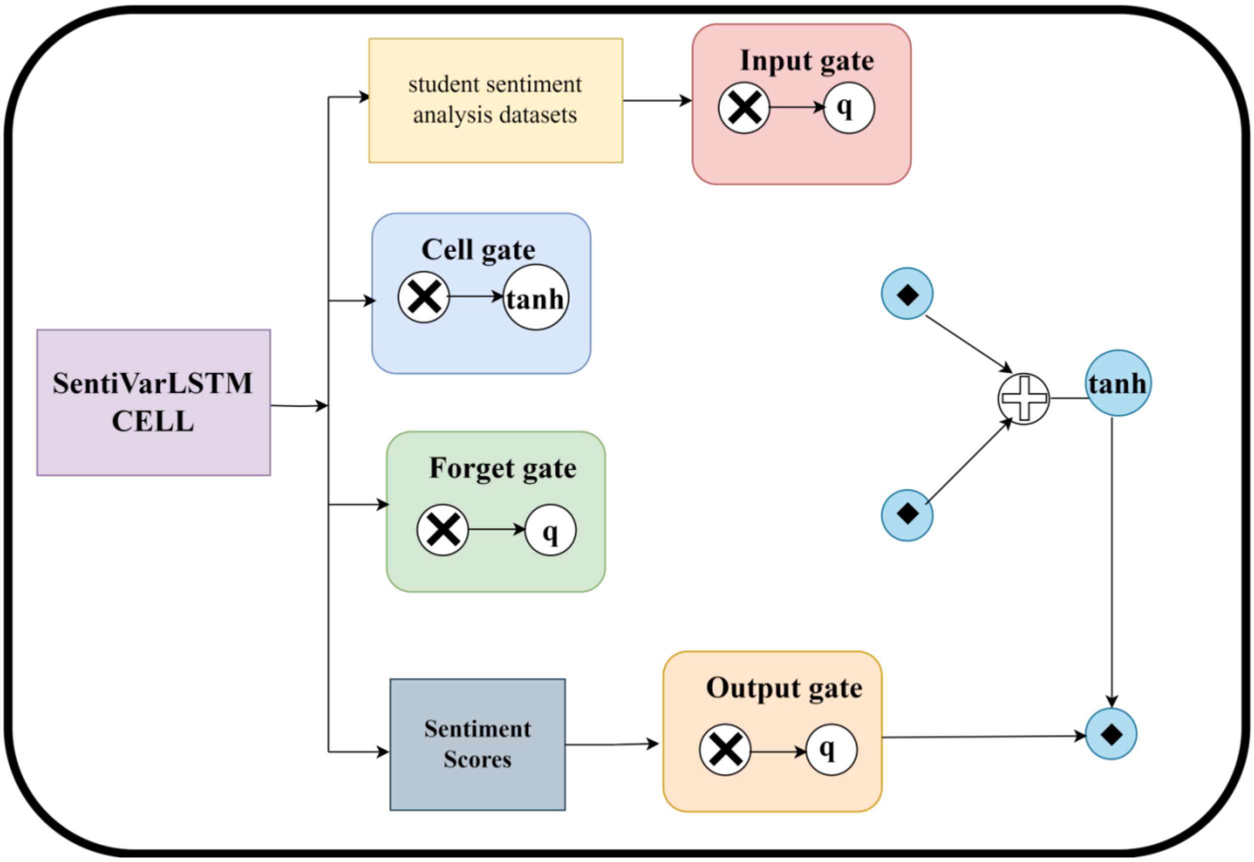

Figure2:SentiVarLSTMarchitecture

$f_t= \sigma_g(W_f x_t+ U_f h_{t-1}+ b_f)$ -------------(1) $i_t= \sigma_g(W_t x_t+ U_i h_{t-1}+ b_i)$ (2)

$o_t=\sigma_g(W_o x_t+U_o h_{t-1}+b_o)$ ---------------- (3)

c_t=tanh($w_c x_t$+$U_c h_{t-1}$+b_c) ------------------ (4)

The above equations have the value indices U_f, U_i, U_o, and uc connect the three inputs and hidden state to the previous cell activation level, while the weighted vectors W_f,,, and Wc provide nonlinear activation information to the entry and input data channel. There are four potential educational sequences:, BI, BO, and BC. In conventional thinking, the scale parameter (σ_g) was the gate perceptron, and the key variable (tanh) was then online a ractivity value. For every iterative process t, the following formula isused to generate Gaussian Error Linear Unit (GELU) referred by Fattah etal.(2023) activation solutions in order to compute the stack of unique outcomes, denoted a sht

In equation 8. In every condition, c_t ,the cells are linked. Moreover, in every condition, c_t, the cells are interconnected, contributing to the holistic functioning of the network architecture consisting of 19 layers.

c_t=f_t∗C_{t-1}+i_t∗c_t --------------(5)

h_t=o_t∗GELUc_t-----------------(6)

As a function of the coordinates of its whole output, the LSTM value of the cascaded structures is represented by Yns=[$hr - n$,...,$hr - i$]. It is just necessary to estimate the last output sequence attribute, hr-i, while thinking about problems with overall performance assessment.

The formula used to calculate SentivarLSTM involves a multi-stepprocess: beginning with tokenization and preprocessing of input text to extract individual tokens and clean irrelevant information. Following this, Part-of-Speech (PoS) tagging assigns grammatical categories to tokens, crucial for contextual understanding. Weighting PoS tags based on their relevance in sentiment determination is conducted next, followed by identification of opinion words using resources like WordNet. Sentiment scores are then assigned to tokens, considering both PoS tag weights and the presence of opinion words.

Let S denote the sentiment score assigned to a token x_i, represent the Part-of-Speech tag of token x_i, $\omega_i$indicate the presence of opinion words in x_i, and $\alpha(PoS_i)$ denote the weight assigned to the PoS tag PoS_i.

The sentiment score S for token $x_i$is computed as:

$S=\alpha(PoS_i)\times\omega_i$

Where:

- (PoS_i) is the weight assigned to the Part-of-Speech tagPoS_i, determined based on its relevance in sentiment determination.
- $\omega_i$isabinaryvalueindicatingthepresence(1)orabsence(0)ofopinionwordsintokenx_i
.

Algorithm3:SentiVarLSTM

Input:

The input consists of textual data extracted from student sentiment analysis datasets.

Begin:

1. DatasetPre-processing:
   - Tokenization: Split the text in to individual words or tokens.

     $token{=}text.split()$

   - Lower casing: Conver tall tokens to lowercase to ensure consistency.

     $Lowercase_{tokens}{=}[token.lower()fortokenintokens]$

   - Stop words removal: Remove common stop words that do not contribute much to sentiment analysis.
   - Text cleaning operations: Perform additional text cleaning operations such as removing punctuation, special characters, and irrelevant symbols.
2. PoSTagging with SentiVarLSTM:
   - Use an LSTM (LongShort-TermMemory) neural network to perform PoS tagging on the pre-processed text.

     $ft{=}(W_f.[h_{t-1},x_t]{+}b_f)$

   - The SentiVarLSTM model learns to assign appropriate PoStagsto each token in the text, capturing contextual information.

     $i_t{=}(W_i.[h_{t-1},x_t]{+}b_i)$
3. Weight Calculation:
   - Calculate weights for each PoStag based on its importance or relevance in determining sentiment.
   - The weights can be assigned manually or learned from a labeled sentiment dataset.
4. Opinion Word Calculation using WordNet:
   - Utilize WordNet, a lexical database, to identify and extract opinion words from the dataset.
   - Word Net provides a semantic network of words and their relationships, allowing the identification of words that express opinions or sentiments.
5. Sentiment Analysis:
   - Assign sentiment scores to each to ken in the pre-processed text based on the calculated weights and presence of opinion words.
   - Convolution operation:z = $W_x{+}b$,where z is the output,W is the weight matrix, x is the input, and b is the bias
   - Model architecture consistsof19 layers

- Aggregating the sentiment scores for all tokens can yield an overall sentiment score for the text or document.

  $overall_{sentiment}=\sum sentiment_{scores}$

6. Normalize SentimentScores:
   - Normalize the sentiment scores to a common scale, such as arrange of -1 to 1or 0 to 1, for better interpretability and comparison.

$$normalized_{sentiment}=\frac{overall_{sentiment}-\min_{sentiment}}{\max_{sentiment}-\min_{sentiment}}$$

Output:

SentimentScores: The output of the SentiVarLSTM algorithm is sentiment scores assigned to each token in the pre-processed text.

IV. Resultsand discussion

In this section, this research presents the results and engages in a comprehensive discussion of the findings derived from the application of SentiVarLSTM in student sentiment analysis. The results and discussion section presents the performance evaluation of various sentiment analysis algorithms, including Word2Vec, TF-IDF, LSTM, Tokenizing LSTM, and the proposed SentivarLSTM model, across three different educational datasets. The outcomes of the model's performance, including sentiment scores and normalized values, will be analyzed and interpreted, shedding light on the effectiveness and nuances of the proposed methodology.

PerformanceMetrics

1. Accuracy: The fraction of samples with the right classification out of all samples. Mathematically:

$$Accuracy=\frac{(TP+TN)}{(TP+FP+TN+FN)}- \quad (23)$$

2. Precision: Ratio of student sentiment analysis samples with accurate identification to total student sentiment analysis samples with accurate identification. Mathematically:

$$Precision = \frac{TP}{TP+FP} \quad (24)$$

3. Recall (also known as sensitivity or true positive rate): The proportion of correctly classified student sentiment analysis samples out of the total number of actual student sentiment analysis samples. Mathematically:

$$Recall = \frac{TP}{TP+FN} \quad (25)$$

4. F1score:A middle ground between accuracy and memory that strikes a harmonic mean. Mathematically:

$F1score = 2 * Precision * Recall / (Precision + Recall)$ ----------------------------------- (26)

Table1:classification performance metrics comparison

		Algorithms	Accuracy	Precision	Recall	F-measure
Existing methods	**Dataset 1**	**Word2Vec**	0.501	0.504	0.505	0.504
		TF-IDF	0.405	0.405	0.408	0.407
		LSTM	0.27	0.28	0.28	0.30
		Tokenizing LSTM	0.28	0.28	0.29	0.31
Proposed Method		**SentivarLSTM**	0.9791	0.9792	0.9794	0.9798
Existing methods	**Dataset 2**	**Word2Vec**	0.521	0.534	0.539	0.541
		TF-IDF	0.415	0.410	0.409	0.409
		LSTM	0.31	0.33	0.34	0.31
		Tokenizing LSTM	0.32	0.33	0.31	0.32
Proposed Method		**SentivarLSTM**	0.9821	0.991	0.9831	0.9867
Existing methods	**Dataset 3**	**Word2Vec**	0.523	0.537	0.541	0.543
		TF-IDF	0.418	0.413	0.412	0.412
		LSTM	0.33	0.35	0.33	0.33
		Tokenizing LSTM	0.34	0.35	0.33	0.33

Proposed method		**SentivarLSTM**	0.9823	0.992	0.9833	0.9869

The table 1 shows evaluation results across three datasets showcase the performance of various sentiment analysis algorithms, including Word2Vec,TF-IDF,LSTM,TokenizingLSTM, and the proposed Sentivar LSTMmethod. A cross all datasets, SentivarLSTM consistently

Outperforms existing methods interms of accuracy, precision, recall, and F-measure.InDataset1, SentivarLSTM achieves an accuracy of 0.9791, precision of 0.9792, recall of 0.9794, and F-measure of 0.9798, indicating its robustness in sentiment analysis compared to other algorithms. Similarly, in Dataset 2 and Dataset 3, SentivarLSTM demonstrates superior performance, with accuracyvaluesof0.9821and0.9823,precision values of 0.991and 0.991, recall values of 0.9831 and 0.9833, and F-measure values of 0.9867 and 0.9869, respectively. These results underscore the effectiveness of the proposed SentivarLSTM model in accurately analyzing sentiment in student-generated textual data across diverse datasets, high lighting its potential as a valuable tool for educators and researchers in understanding student feedback dynamics.

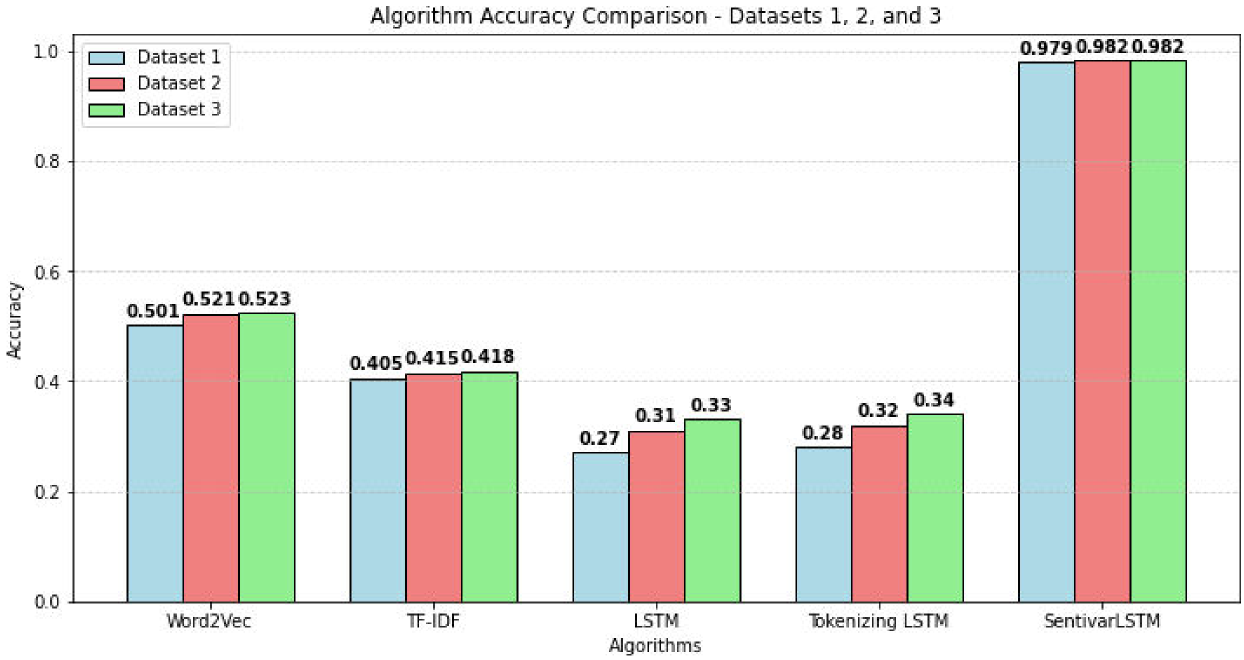

Figure3:Accuracy comparison chart

A chart comparing accuracy is shown in figure3. Algorithms are shown on the x-axis, while accuracy of sentivarLSTM scores are shown on the y-axis.

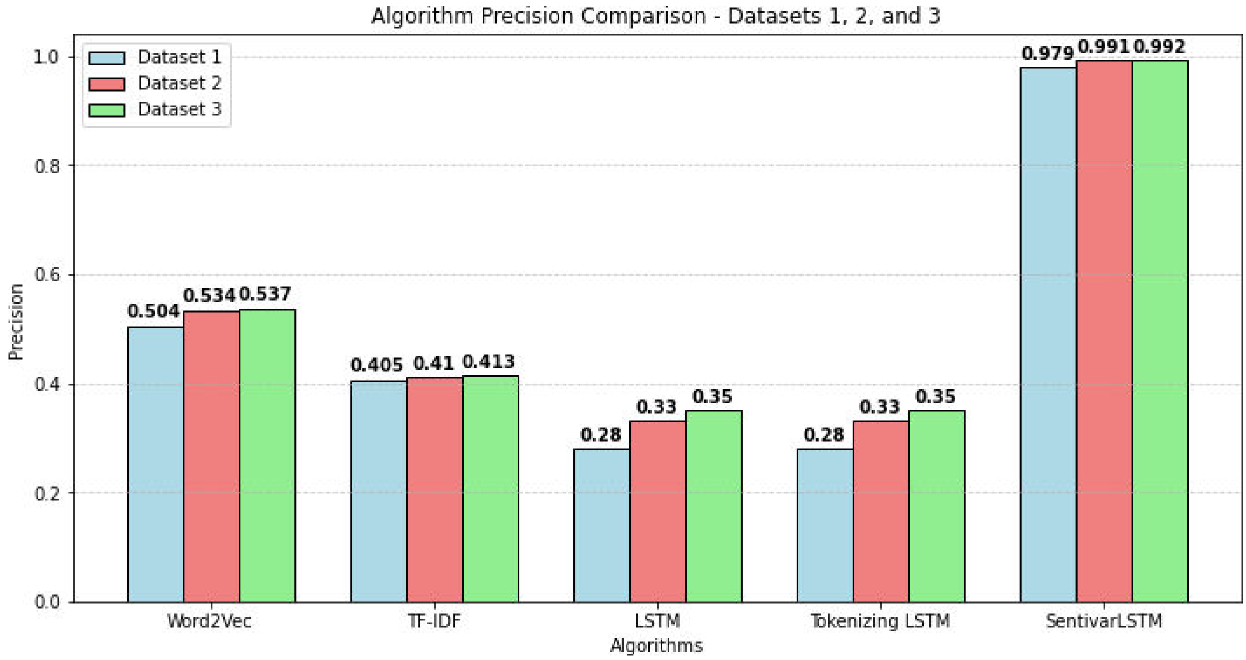

Figure4:Precision comparison chart

Figure4 displays a chart comparing precision. Precision values are shown on the y-axis and algorithms are shown on the x-axis.

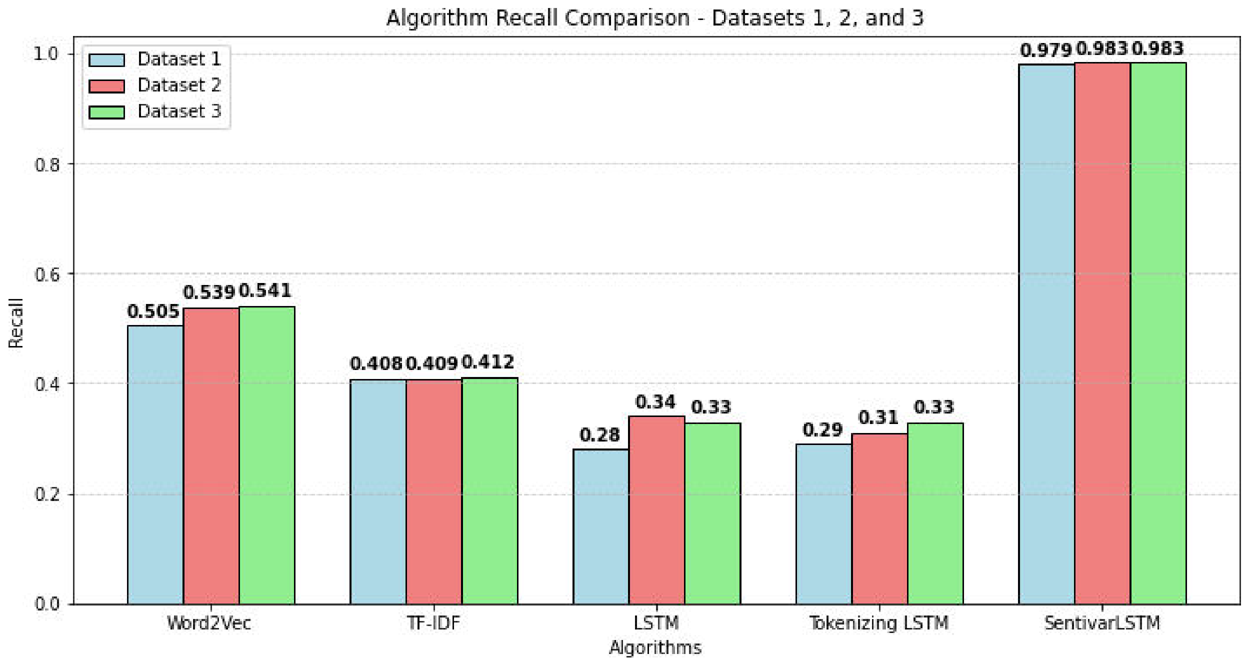

Figure5:Recall comparison chart

There is are call comparison table in figure5.Ononeside, this paper has the algorithm, and on the other, this paper has the recall values.

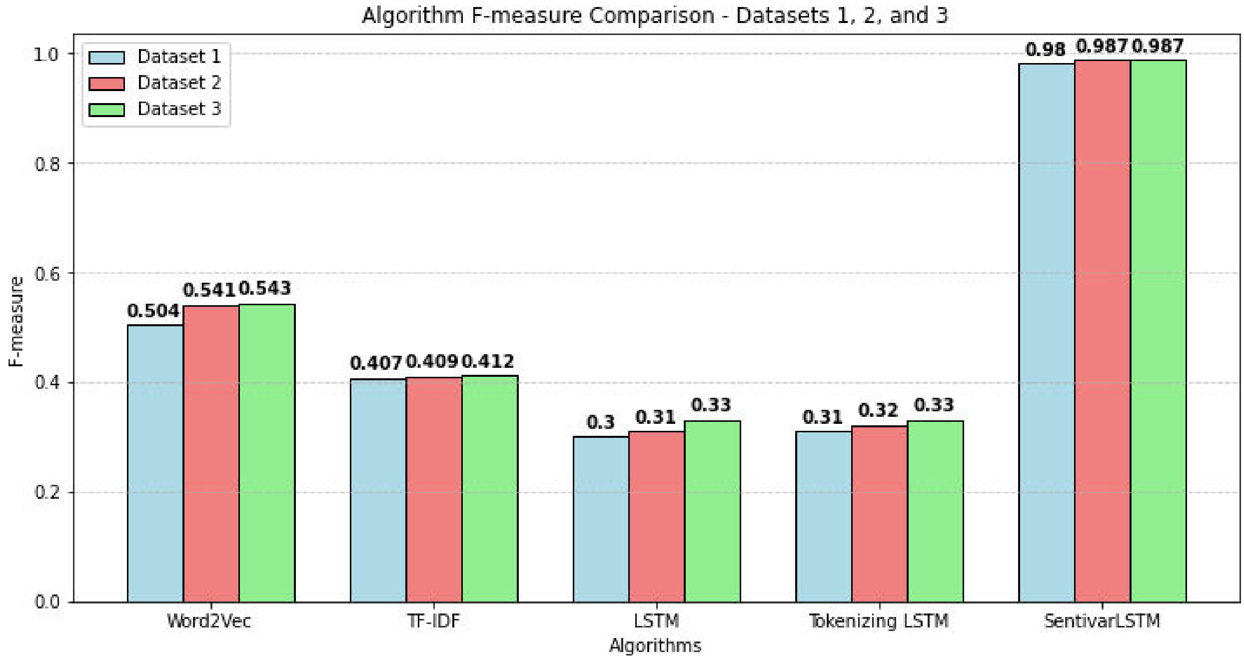

Figure6:F-measure comparison chart

Figure6 displays a chart comparing f-measures.This graph displays algorithms on the x-axis and f-measure values on the y-axis.

V. Conclusion

In conclusion, the SentiVarLSTM model represents a significant advancement in sentiment analysis for student-generated textual data. By addressing the challenges posed by data complexity, volume, and inconsistencies, SentiVarLSTM offers a robust framework tailored specifically for analyzing student sentiments with precision and accuracy. Leveraging sophisticated techniques such as Sentiment Variation Analysis and Part-of-Speech tagging, combined with the power of LSTM neural networks, SentiVarLSTM provides valuable insights into the sentiment dynamics of student feedback. Moreover, the utilization of three diverse educational datasets underscores the versatility and effectiveness of the proposed approach across different contexts. In Dataset 1, SentivarLSTM achieves an accuracy of 0.9791, precision of 0.9792, recall of 0.9794, and F-measure of 0.9798 in Dataset 2 and Dataset 3, SentivarLSTM demonstrates superior performance, with accuracy values of 0.9821 and 0.9823, precision values of 0.991 and 0.991, recall values of 0.9831 and 0.9833, and F-measure values of 0.9867 and 0.9869,respectively. As educators and researchers seek for a better understanding of student experiences and performance, SentiVarLSTM is a vital instrument for increasing educational outcomes and driving continuous improvement in the learning environment. In the future, SentiVarLSTM could be further developed to enhance its adaptability to diverse educational domains and languages, integrate multimodal data sources for a more comprehensive analysis, explore real-time student sentiment analysis techniques, and incorporate domain-specific knowledge through collaboration with educators and stakeholders, ultimately driving continuous improvement in educational outcomes.

VI. REFERENCE

1. Abid, F., Li, C., & Alam, M. (2020). Multi-source social media data sentiment analysis using bidirectional recurrent convolutional neural networks. Computer Communications. doi:10.1016/j.comcom.2020.04.002
2. Agarwal, B., Nayak, R., Mittal, N., & Patnaik, S. (Eds.). (2020). Deep Learning-Based Approaches for Sentiment Analysis. Algorithms for Intelligent Systems. doi:10.1007/978-981-15-1216-2
3. An, H., & Moon, N. (2019). Design of recommendation system for tourist spot using sentiment analysis based on CNN-LSTM. Journal of Ambient Intelligence and Humanized Computing. doi:10.1007/s12652-019-01521-w
4. Demotte, P., Senevirathne, L., Karunanayake, B., Munasinghe, U., & Ranathunga, S. (2020). Sentiment Analysis of Sinhala News Comments using Sentence-State LSTM Networks. 2020 Moratuwa Engineering Research Conference (MERCon). doi:10.1109/mercon50084.2020.9185327
5. Edara,D.C.,Vanukuri,L.P.,Sistla,V.,&Kolli,V. K.K.(2019). Sentiment analysis and text categorization of cancer medical records with LSTM. Journal of Ambient Intelligence and Humanized Computing. doi:10.1007/s12652-019-01399-8
6. Gandhi,U.D.,MalarvizhiKumar,P.,ChandraBabu,G.,&Karthick,G.(2021). Sentiment Analysison Twitter Data by Using Convolutional Neural Network(CNN)andLongShort Term Memory (LSTM). Wireless Personal Communications. doi:10.1007/s11277-021- 08580-3
7. Guo,Y.(2020).Stock Price Prediction Based on LSTM Neural Network: the Effectiveness of News Sentiment Analysis.2020 2nd International Conference on Economic Management and Model Engineering (ICEMME). doi:10.1109/icemme51517.2020.00206
8. Gupta, I., & Joshi, N. (2021).Feature-Based Twitter Sentiment Analysis With Improved Negation Handling. IEEE Transactions on Computational Social Systems, 8(4), 917–927. doi:10.1109/tcss.2021.3069413
9. Hemanth, J., Bestak, R., & Chen, J. I.-Z. (Eds.). (2021).Intelligent Data Communication Technologies and Internet of Things. Lecture Notes on Data Engineering and Communications Technologies. doi:10.1007/978-981-15-9509-7
10. Jin, Z., Yang, Y., & Liu, Y. (2019). Stock closing price prediction based on sentiment analysisandLSTM.NeuralComputingandApplications.doi:10.1007/s00521-019-04504- 2
11. Kaiser, M. S., Bandyopadhyay, A., Mahmud, M., & Ray, K. (Eds.). (2021). Proceedings of International Conference on Trends in Computational and Cognitive Engineering. Advances in Intelligent Systems and Computing. doi:10.1007/978-981-33-4673-4
12. Nath Saha, B., & Senapati, A. (2020). Long Short Term Memory (LSTM) based Deep Learning for Sentiment Analysis of English and Spanish Data. 2020 International Conference on Computational Performance Evaluation (ComPE). doi:10.1109/compe49325.2020.9200054
13. Ombabi, A. H., Ouarda, W., & Alimi, A. M. (2020). Deep learning CNN–LSTM framework for Arabic sentiment analysis using textual information shared in social networks. Social Network Analysis and Mining, 10(1). doi:10.1007/s13278-020-00668-1
14. Paolanti, M., Mancini, A., Frontoni, E., Felicetti, A., Marinelli, L., Marcheggiani, E., & Pierdicca, R. (2021).Tourism destination management using sentiment analysis and geo-location information: a deep learning approach. Information Technology & Tourism, 23(2),

241–264. doi:10.1007/s40558-021-00196-4

15. Sangeetha, K., & Prabha, D. (2020). Sentiment analysis of student feedback using multi- head attention fusion model of word and context embedding for LSTM. Journal of Ambient Intelligence and Humanized Computing, 12(3), 4117–4126.doi:10.1007/s12652-020- 01791-9
16. Senthil Kumar, N. K., & Malarvizhi, N. (2020). Bi-directional LSTM–CNN Combined method for Sentiment Analysis in Part of Speech Tagging (PoS). International Journal of Speech Technology. doi:10.1007/s10772-020-09716-9
17. Shilpa, P. C., Shereen, R., Jacob, S., &Vinod, P. (2021). SentimentAnalysis Using Deep Learning. 2021 Third International Conference on Intelligent Communication Technologies and Virtual Mobile Networks (ICICV). doi:10.1109/icicv50876.2021.9388382
18. Singh, P. K., Panigrahi, B. K., Suryadevara, N. K., Sharma, S. K., & Singh, A. P. (Eds.). (2020).ProceedingsofICETIT2019.LectureNotesinElectrical Engineering. doi:10.1007/978-3-030-30577-2
19. Usama, M., Ahmad, B., Song, E., Hossain, M. S., Alrashoud, M., & Muhammad, G. (2020).Attention-based sentiment analysis using convolutional and recurrent neural network. Future Generation Computer Systems. doi:10.1016/j.future.2020.07.022
20. Venkatesh, Hegde,S.U.,S,Z.A.,&Y,N.(2021).HybridCNN-LSTMModelwithGloVe Word Vector for Sentiment Analysis on Football Specific Tweets. 2021 International Conference on Advances in Electrical, Computing, Communication and Sustainable Technologies (ICAECT). doi:10.1109/icaect49130.2021.9392516
21. C.Li,G.ZhanandZ.Li, "News Text Classification Based on Improved Bi-LSTM-CNN," 2018 9th International Conference on Information Technology in Medicine and Education (ITME), Hangzhou, China, 2018, pp. 890-893, doi: 10.1109/ITME.2018.00199.
22. Al-Obaydy,Wasseem N.I brahem, HalaA.Hashim,Y.A.Najm,andAhmedAdeebJalal. "Document classification using term frequency-inverse document frequency and K-means clustering."Indonesian Journal of Electrical Engineering and Computer Science 27,no. 3 (2022): 1517-1524.
23. Shaik, Amjan, B. Aruna Devi, R. Baskaran, Satish Bojjawar, P. Vidyullatha, and Prasanalakshmi Balaji. "Recurrent neural network with emperor penguin-based Salp swarm(RNN-EPS2)algorithm for emoji based sentiment analysis." Multimedia Tools and Applications (2023): 1-20.
24. Fattah,Fathun Karim, Pritom Mojumder, Azmol Ahmed Fuad, and MohiuddinAhmad."A Comparative Analysis Between LSTM and LSTM Ensembled Strategy for Short-term Load Forecasting."*Authorea Preprints* (2023).
25. Umar, Musa Abba, Penuel Medan, and Zainab Aliyu Atiku. "Students' industrial work experience scheme (SIWES): exploring higher institution student's participation and sense of belonging." Higher Education, Skills and Work-Based Learning (2024).