FRAMEWORK FOR ANALYSING SENTIMENTS OF EDUCATIONAL STAKEHOLDERS ON NEP 2020 BY USING HYBRID MACHINE LEARNING TECHNIQUES

Ms. Purnima Kumari Srivastava, Faculty of Computing and Information Technology, Usha Martin University, Ranchi, Jharkhand.
Dr. Sharmistha Roy, Faculty of Computing and Information Technology, Usha Martin University, Ranchi, Jharkhand.

ABSTRACT
The National Education Policy (NEP) 2020 marks a significant reform in India's education system, impacting diverse stakeholders, including students, parents, teachers, and policymakers. Understanding the sentiments of these stakeholders is critical for effective implementation and adaptation. Sentiment analysis is the process of analysing, identifying and categorizing our opinion, thoughts, feelings, which may be expressed in piece of text, pictures, emojis and voice, and has become increasingly vital in understanding student's behavior and teaching methodology which are adopted by teachers' and also opinions of other educational stakeholders. Traditional machine learning algorithms, such as Naïve Bayes(NB), Support Vector machine(SVM). Random Forest(RF), Decision Tree(DT) and Logistic Regression(LR)etc. have been widely used to analyse sentiments. Although, they work well but they face challenges in handling complex linguistic nuances, such as sarcasm and context dependence, which can reduce their accuracy. To address these limitations, this research paper proposed a model "LDN-LSTM" (Linguistically Driven Neural Long Short-Term Memory) that enhances sentiment analysis by leveraging linguistic nuances and addressing the limitations of conventional models in handling complex, context-dependent language structures. The proposed model is compared with traditional machine learning models to measure the accuracy and efficiency of the proposed approach. This can help policymakers, educators and researchers to gauge reception of NEP -2020 and make data driven decisions.
Keywords: Sentiment Analysis, Machine Learning, NEP 2020, LSTM

I.Introduction

1.1. Introduction to Sentiment Analysis: In recent years, blogs, review writing, tweets etc are popularly used by people to express their feelings, opinion etc. Sentiment analysis is one of the emerging processes which focusses on identifying and categorizing opinions, thoughts, feelings which are expressed in textual format, images, emoji. The primary goal of sentiment analysis is to classify the emotions of the people in the emotions of people in positive, negative or in natural category.

1.2. Introduction to Machine Learning: Machine Learning is the subfield of artificial Intelligence that aims on developing algorithms and models by which computer machine learn from and make decision based on the data. Machine learning system identify patterns in data, make predictions, classifying information into several category and take actions based on new data.

1.2.1. Basic term in Machine Learning:

- Algorithms: Algorithms are sets of rules and instructions that system follow and make decisions or predictions. Some common algorithms are: Support Vector Machine (SVM), RF(Random Forest), DT(Decision Tree), Naïve bayes(NB) etc.
- Training and testing: Machine learning Models are trained using a dataset. During training, the models learn to recognize patterns recognize patterns in the data. When machine trained, the separate dataset is used for testing and for evaluating performance.
- Features and Labels:
  - Features: The input variables or attributes used by models to make pattern.
  - Labels: the output or target variable model aims to predict.

- Underfitting and Overfitting: Underfitting occurs when a model is too simple to capture the underlying patterns in the data. Overfitting occurs when a model is too complex and captures not just the underlying patterns but also the noise in the training data.
- Supervised, Unsupervised and Re-enforcement learning: Various Machine learning algorithm defined data items by using predefined class label or without class label.

1.3. Introduction to Deep Learning: Deep learning is a method in artificial Intelligence that teaches computers to process the data in such a way that human neural network architecture to learn the hidden patterns and relationships in the data set.

1.3.1. Features of Deep learning:

i. Multiple Layers (Deep Architecture)

- Feature Hierarchy: Deep learning models consist of multiple layers of neurons (artificial nodes) that progressively extract higher-level features from raw data. Each layer learns increasingly abstract representations, allowing the model to capture intricate patterns in the data.
- Depth: The term "deep" in deep learning refers to the large number of layers (sometimes hundreds or more) that can be stacked in a network, enabling it to learn complex functions.

ii. Automatic Feature Extraction

- Learning from Raw Data: Unlike traditional machine learning models that require manual feature engineering, deep learning models can automatically learn to extract features from raw data. For instance, in image recognition, a deep neural network might automatically learn to detect edges, shapes, and eventually more complex structures like faces or objects.
- End-to-End Learning: Deep learning models often operate in an end-to-end fashion, meaning they take raw input data (like pixels in an image or words in a sentence) and directly map it to an output (like a label or prediction) without the need for intermediate steps of feature extraction.

iii. Nonlinear Transformations

- Activation Functions: Deep learning models use nonlinear activation functions (such as ReLU, sigmoid, or tanh) to introduce nonlinearity into the network. This allows the model to learn complex mappings between inputs and outputs, enabling it to solve problems that are not linearly separable.

iv. High Capacity for Learning

- Large Parameter Space: Deep neural networks have a vast number of parameters, which gives them a high capacity to learn from data. This makes them capable of fitting very complex functions and modelling sophisticated patterns in the data.
- Flexibility: The large number of parameters also allows deep learning models to adapt to a wide range of tasks, from image and speech recognition to natural language processing and game playing.

v. Scalability

- Large Datasets: Deep learning thrives on large amounts of data. The more data available, the better the model can learn. This scalability is one reason why deep learning has become so powerful with the advent of big data.
- Hardware Efficiency: Deep learning models are highly parallelizable, which makes them well-suited for modern hardware like GPUs and TPUs, allowing for efficient training on massive datasets.

vi. Generalization

- Transfer Learning: Deep learning models can often generalize well across tasks through transfer learning, where a model pre-trained on one task is fine-tuned on a different, but related, task. This capability leverages knowledge learned from a large dataset to improve performance on a smaller dataset.

vii. Representation Learning

- Hierarchical Representations: Deep learning models excel at representation learning, where they automatically learn hierarchical representations of data. For example, in natural language processing, different layers might learn to represent letters, words, sentences, and semantic meaning.

- Unsupervised Learning: Deep learning models can also learn representations in an unsupervised manner, meaning they can discover structure in the data without explicit labels.

viii. Complex Task Handling

- Sequential Data: Deep learning models like RNNs, LSTMs, and Transformers are well-suited for handling sequential data, such as time series, speech, or text, where the order of inputs matters.
- Multi-Task Learning: Deep learning models can be designed to perform multiple related tasks simultaneously, leveraging shared representations to improve overall performance.

ix. Regularization Techniques

- Preventing Overfitting: Given the high capacity of deep learning models, they are prone to overfitting. Regularization techniques like dropout, L2 regularization, and data augmentation are commonly used to mitigate this risk and improve generalization.

x. Innovative Architectures

- Convolutional Neural Networks (CNNs): Specially designed for spatial data like images, CNNs use convolutional layers to detect local patterns and are highly effective in tasks like image recognition.
- Recurrent Neural Networks (RNNs): Designed for sequential data, RNNs and their variants (LSTMs, GRUs) maintain a memory of previous inputs, making them ideal for tasks like language modelling and time series forecasting.
- Transformers: A more recent architecture, Transformers have become the foundation of many state-of-the-art models in natural language processing (e.g., BERT, GPT). They use self-attention mechanisms to capture long-range dependencies in data.

1.4. Introduction to Proposed LDN-LSTM(Linguistically Driven Neural Long Short-Term Memory**):**

Proposed model combines different machine learning techniques, Deep Learning models, or frameworks to leverage their individual strengths and mitigate their weaknesses. The goal is to create a more robust, accurate, and efficient solution for complex problems that may not be adequately addressed by a single model or technique for analysing Sentiments towards NEP 2020 by different Educational Stakeholders. In Proposed Hybrid Machine Learning Approach, we have used Logistic Regression, Decision Tree, Naïve bayes and LSTM approach to find more accuracy in analysing sentiments.

The LDN-LSTM model integrates linguistic insights into its architecture, enabling it to better handle the complexities of natural language:

- **Dependency Parsing**: By understanding syntactic relationships between words, the model focuses on relevant terms and ignores irrelevant ones. For example, in the sentence *“The implementation is not as bad as expected”*, dependency parsing helps recognize that the overall sentiment is positive despite the presence of negative words.
- **Part-of-Speech (POS) Tagging**: POS tags help the model differentiate between adjectives, nouns, and verbs, allowing it to understand context better. For instance, adjectives like *“good”* or *“poor”* carry more sentiment weight than verbs like *“is”*.

Industrial Engineering Journal
ISSN: 0970-2555
Volume : 54, Issue 3, No.1, March : 2025

II.Literature Review:

Table 1: Comparative study of various research work related to sentiment analysis on LSTM(Long Short Term Memory) along with advantages and disadvantages

Title	Introduction	Methodology	Result	Advantages	Disadvantages
Sentiment Analysis of Students evaluation of teaching based on Bi-LSTM algorithm[11].	This research analyse the emotional tendency of students	Dataset-IMDB-50000 film review, students' teaching evaluation-7956, Algorithm-Bi-LSTM, LSTM, RNN	LSTM is better than RNN. Bi-LSTM accuracy-84.33%	Comparative study of LSTM and Bi-LSTM	Pre-processing is not done
Sentiment Analysis of students's comment using Long - Short Term Model[4].	Analyse the sentiments of student's by their comments	Dataset-3000 students feedback(2017-2018) LSTM used	Accuracy Trained dataset +ve-92% -ve-79%	Has prospective to overcome several flows in traditional method e.g. bags-of-word, n-gram, NB and SVM	Not suitable for multi-lingual for SA
Sentiment Analysis using LSTM[10]	This paper evaluate sentiments on movie review	Dataset-IMDB movie review	Accuracy - 86.85%	Shows that LSTM can handle long term dependencies very effectively	Domain specific
Emotion recognition for Education using Sentiment Analysis[1].	Sentiment recognition on textual phrases written in Spanish	CNN-LSTM used. Also KNN, Multinomial NB, Bernoulli NB etc. used	Accuracy Bernoulli NB-76.77% Multinomial NB-75.31% KNN-68.46% CNN+ LSTM-88.26%	Helps to detect Sentiment in Educational context.	Not suitable for voice recognition, facial expression and brain signal

Sentiment Analysis of Students' feedback with NLP and Deep Learning: A systematic Mapping study[5].	Related to SA of Publication, literature review	Used systematic mapping as the research methodology for reviewing the strucure	LSTM-CNN accuracy- 90% above	Helps to analyse the paper according to publishers and time period	Limited dataset
Sentiment Analysis of Animated Online Education Texts Using Long Short-Term Memory Networks in the Context of the Internet of Things[6].	describes a sophisticated system that leverages IoT technology to collect multimodal data from students engaged in animated online education	Dataset Collected from sensory organ, motion sensor, smart pen. Bi-LSTM used	Accuracy Bi-LSTM-93.92%	Provides valuable insights and guidance for intelligent development of educational field	Diversity in dataset is not found, Limited dataset
Text Based Sentiment Analysis using LSTM[7].	Related to textual sentiment analysis	Dataset-IMDB Movie Review LSTM Model used	Accuracy LSTM-85%	Textual Review can be easily classified	Limited Variety of dataset
Sentiment Analysis using Neural Network and LSTM[2].	SA on twitter dataset by Deep learning [LSTM]	Dataset-Kaggle(1.6 million tweets are analyzed), LSTM, CNN , Simple Neural Network models are used.	Accuracy LSTM-87% CNN-82% Simple Neural Network- 81%	Comparative Analysis of models done on performance evaluation metric	No use of REST API and web crawling-based solution
Sentiment Analysis using simplified Long -Short Term memory recurrent Neural Network[3].	Proposed slim version of LSTM	Dataset-CrowdFlower from Everyone Library- LSTM and Bi-LSTM model used	Negative Sentiment is more accurately predicted than positive Sentiments.	It seems that many parameter can effect the working of LSTM	Conceptual work

			LSTM is ore accurate than LSTM.		
Sentiment Analysis using a multinomial LR-LSTM Model[8].	Proposed a stack multinomial LR-LSTM model for classification of tweets	Dataset- twitter Sentiment dataset- 162,980 tweets, Model- KNN, NB, LR, DT AdaBoost, RF, CNN, LSTM	Default sentiment accuracy CNN+ LSTM- 87% Ml Model with TextBlob- MLR- LSTM - 88%	Comparative Performance Analysis	Domain specific
An efficient Sentiment Analysis methodology based on Long-Short Term Memory networks[9].	Proposed a Model(APSO -LSTM) in which skip-gram architecture is used, LSTM , ANN and SVM is used	Dataset- Amazon dataset, Trip advisor data set, Demonetization data set, Book review dataset, LSTM Model	Accuracy- APSO- LSTM Amazon Product Review - 96.80% Trip advisor -97.8% Demonetization- 93.2% Book Review Dataset - 95.2%	Diverse dataset used	------------

III.Framework of Analyzing Sentiments towards NEP 2020

3.1. Dataset: The dataset is taken from Kaggle(https://www.kaggle.com/datasets/rishabh6377/india-national-educationpolicy2020-tweets-dataset). The dataset contains tweets which are from 31 July 2020 to 12 August 2020, on New Education policy 2020. This is written in English language. This dataset contains 18240(0 to 18239) tweets. The dataset has seven attribute- #(Serial_No), Author_ID, Date_of_Tweet, Tweets, Likes_on_Tweets, User_handle, Tweet_Link.

3.2. Machine Learning Algorithms

- **Logistic Regression**: Logistic regression is commonly utilized for classification tasks, particularly when dealing with categorical data variables[12]. It is most suitable where the output is a categorical variable. Unlike linear regression, which produces values ranging from negative infinity to positive infinity, logistic regression models the probability of a response variable that lies within the range of 0 and 1[12]. To achieve this transformation, a mapping function is required to constrain the values between 0 and 1. The core of this mapping process in logistic regression is the sigmoid function, which has an S-shaped curve and outputs values strictly between 0 and 1[12]. An example of this is illustrated in Figure 1[15]. This is known as logit function. This is mathematically represented as:

$Logit(x)= 1/(1+e^{-x})$ ....$eq^n(1)$

Where x is the independent variable and e is the euler number. The purpose of logit function is to map any real number to 0 and 1.

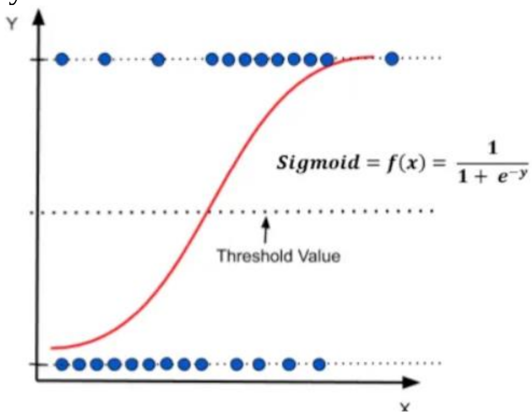

Figure 1: Logistic Regression

Logistic regression is primarily designed for binary classification. However, it can be extended to handle multiple classes through a technique known as multinomial logistic regression[12]. For instance, if there are three classes—A, B, and C—multinomial logistic regression formulates three separate classification problems: distinguishing class A from not class A, class B from not class B, and class C from not class C. These three models are evaluated simultaneously to determine the class with the highest probability relative to the others, enabling accurate classification[12].

Following steps involved in finding the result in python:

  - Data Pre-processing
  - Fitting logistic regression to training set
  - Result prediction
  - Creation of confusion matrix for finding accuracy
  - Visualizing test set result

- **Decision Tree**: Decision trees are among the most widely used supervised predictive learning models, known for their high accuracy and consistency in classifying data instances [12]. These models perform inductive inference by drawing general conclusions from observed examples. The algorithm works by recursively dividing the input space into smaller regions based on specific features. At each node of the tree, a decision is made depending on the value of a feature. This process continues until a predefined stopping criterion is satisfied, such as reaching a maximum tree depth or having nodes that contain instances belonging to a single class. Decision trees are capable of handling both categorical and continuous target variables [12].The decision-making process is visually represented in a tree-like structure, where internal nodes denote decisions based on particular features, branches represent the outcomes of those decisions, and leaf nodes signify final predictions or classifications, as illustrated in Figure 2[14].

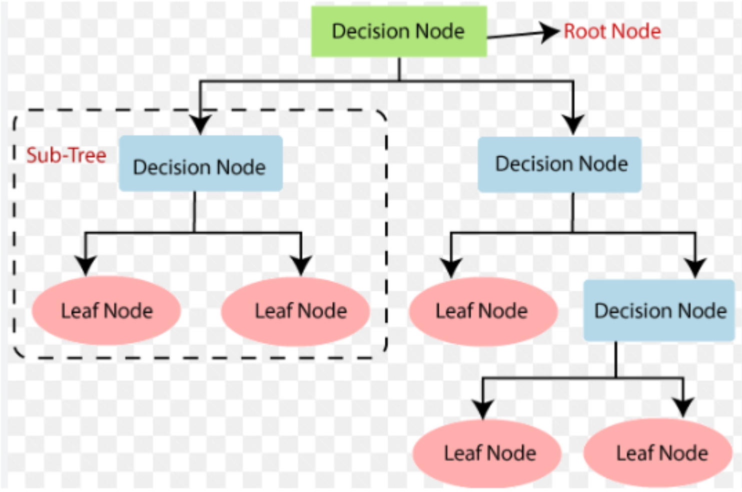

Figure 2: Decision Tree

Following steps are used to create a decision tree learning model in python using library scikit-learn:

- Importing libraries
- Preprocessing and splitting the dataset into features(X) and target variable(Y)
- Splitting the dataset into training and testing sets
- Instantiating a DecisionTreeClassifier and fitted it to the training data
- Trained model used to make prediction on testing data.
- Model Evaluation
- Confusion matrix creation for checking accuracy of model

- **Naïve Bayes**: Naïve Bayes is a widely used machine learning algorithm for classification tasks. Bernoulli Naïve Bayes estimates the likelihood of each feature occurring given a specific class and combines this with the prior probability of each class to make predictions based on Bayes' theorem. It operates under the assumption that features are independent of one another, which simplifies computations (Figure 3)[13]. This algorithm is particularly effective in applications such as text classification and sentiment analysis.

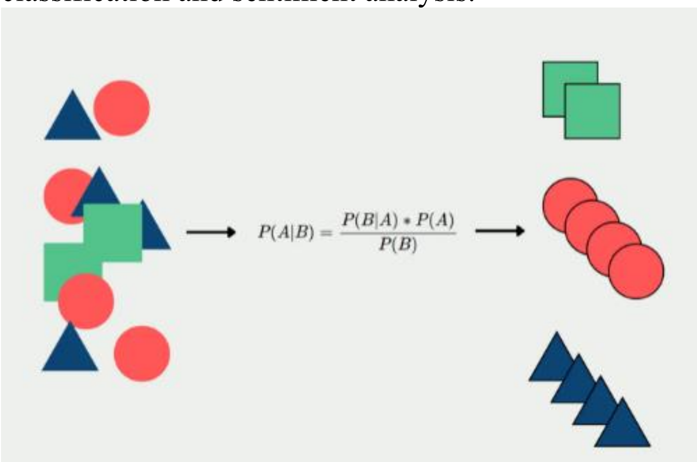

Figure 3. Naïve Bayes Algorithm

Given a Class C_k and an input vector $\boldsymbol{x}=(x_1,x_2,\ldots.x_n)$, the prediction for the class is determined by the following formula:

$$P(C_k|\boldsymbol{x})=\frac{P(Ck)\prod_{i=1}^{n}P(xi|Ck)}{P(x)} \quad \text{.....................eq}^{n}(2)$$

where:

- $P(C_k|\boldsymbol{x})$ is the posterior probability of class C_k given the input vector x,

Industrial Engineering Journal
ISSN: 0970-2555
Volume : 54, Issue 3, No.1, March : 2025

- $P(C_k)$ is the prior probability of class C_k,
- $P(\mathbf{x}_i|C_k)$ is the probability of feature x_i given class C_k
- $P(x)$ is the probability of observing the input vector x.

Following steps are used in classification of NEP-2020 by Bernoulli's naïve bayes in Python:

- Importing necessary libraries like scikit-learn and pandas
- Encoding categorical variable and splitting dataset into features(X) and (Y)
- Splitting the dataset into training and testing sets
- Instantiating a Naïve Bayes classifier and fitted it to the training data
- Trained model used to make prediction on testing data.
- Model Evaluation
- Confusion matrix creation for checking accuracy of model

3.3 Framework for evaluating Sentiments by MLTs

Following Figure-4 show the Framework of Sentiment Analysis for NEP 2020 by using Machine learning Classifiers

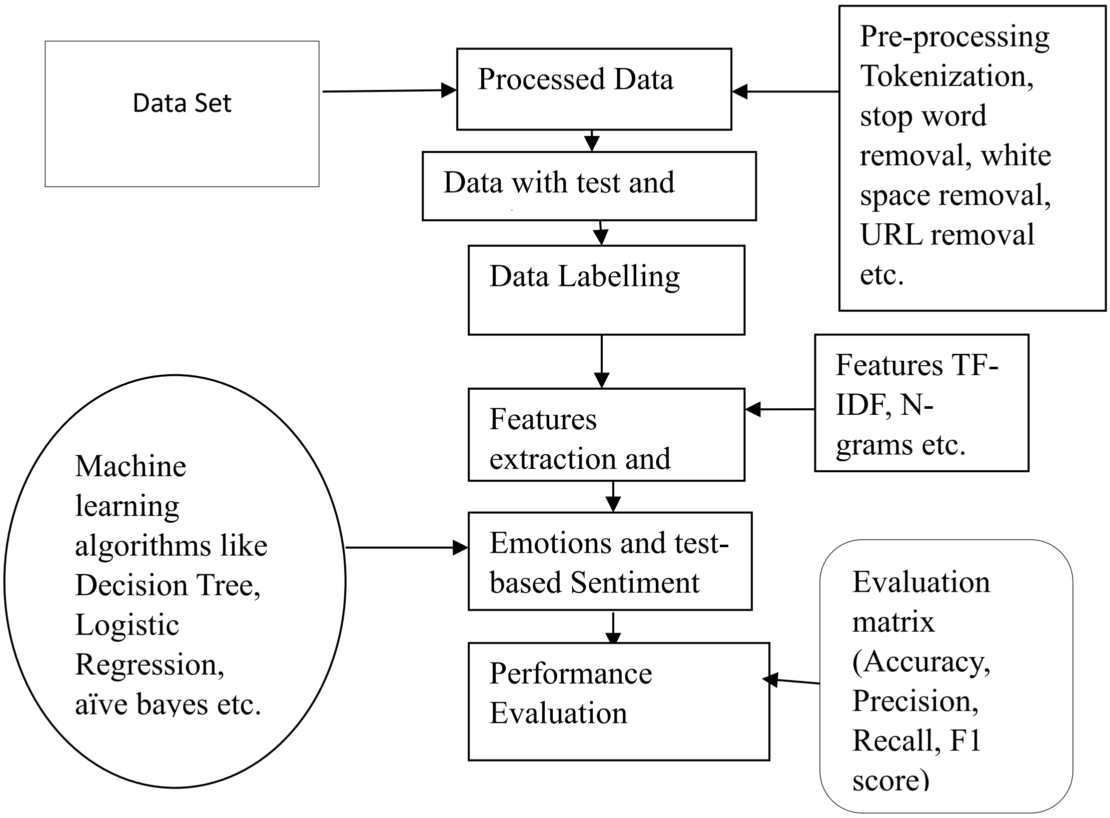

Figure 4. Framework for Sentiment Analysis by ML algorithms

Steps for Analyzing Sentiments on New Education Policy 2020

i. Data Collection: The dataset is taken from Kaggle(https://www.kaggle.com/datasets/rishabh6377/india-national-education-policy2020-tweets-dataset). The data set has seven attributes: #(Serial_No) Author_ID, Date_of_Tweet, Tweets, Likes_on_Tweets, User_handle, Tweet_Link.

ii. Data Preprocessing:

- Text Cleaning: Removing irrelevant characters, punctuation, and special symbols.
- Tokenization: Splitting text into individual words or tokens.
- Stop words Removal: Eliminating common words that do not carry significant meaning.
- Lemmatization or stemming: Reducing words to their base or root form.

iii. Labeling: Manually creating label a subset of the data set to create a ground for sentiment analysis. The labeled data will be used to trained and evaluate machine learning models.
iv. Feature Extraction: Translation of the text data into numeric representations understandable by machine learning algorithms will be performed. This is typically done by some methods like TF-IDF (Term-Frequency-Inverse Document Frequency).
v. Model Selection: Different Machine Learning algorithms for sentiment analysis, such as Bernoulli' Naïve Bayes, Decision Tree and Logistic Regression will be chosen for classifying sentiments.
vi. Model training: Training of selected model will be done.
vii. Testing data model and Performing Sentiment Analysis: testing on rest of the dataset will be performed on trained models to predict sentiment scores (e.g. positive, negative, neutral) for each text or document.
viii. Evaluation: Assess the performance of sentiment analysis model(s) using evaluation metrics like accuracy, precision, recall and F1 –score etc.
ix. Interpretation : Finally, Analyse the performance of different machine learning algorithm in sentiment analysis on NEP-2020.

3.4. Deep Learning Algorithm: Following Deep Learning algorithm is used to analyze sentiments on NEP 2020.

- Long-Short Term Memory (LSTM): .302LSTM are particularly well-suited for tasks involving sequential data, such as sentiment analysis. Sentiment analysis involves determining the sentiment or emotion expressed in a piece of text, and LSTMs are effective at capturing the context and dependencies between words in a sentence, which is crucial for understanding sentiment. As we know Sentiment often depends on long-term dependencies in text. For example, in the sentence "I don't like this book," the sentiment depends on the word "don't," which is far from the word "like." LSTMs are designed to remember these long-term dependencies through their memory cells. LSTMs can process text word by word, maintaining an understanding of context across the sequence. LSTMs use a set of gates (input gate, forget gate, and output gate) to control the flow of information, allowing them to selectively remember or forget information. This is crucial in sentiment analysis where not all words contribute equally to the sentiment of a sentence.

➢ Components of an LSTM Model: the component of LSTM is shown in Figure 5[16].

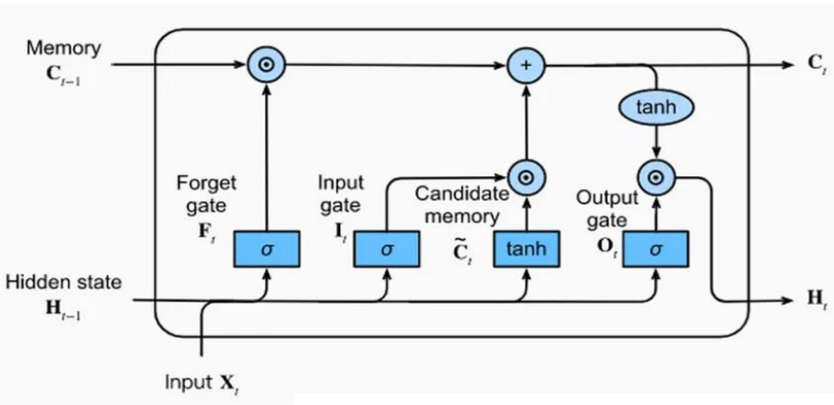

Figure 5: Working of LSTM model

○ Memory Cell:The core component of an LSTM is the memory cell, which is responsible for maintaining information over time. The cell state is like a conveyor belt that runs through the entire chain, with only some minor linear interactions. It has the ability to carry information across many time steps, which is crucial for capturing long-term dependencies.

○ Gates:LSTMs use three gates to control the flow of information in and out of the memory cell:

* Forget Gate: Decides what information should be thrown away or kept in the cell state. It takes the previous hidden state and the current input as inputs, passes them through a sigmoid function, and outputs a number between 0 and 1 for each number in the cell state.

* Input Gate: Decides what new information should be stored in the cell state. It also has two parts: the input gate layer, which decides which values to update, and the candidate layer, which creates a vector of new candidate values.
* Output Gate: Decides what part of the cell state should be output. The output is based on the cell state but is a filtered version that only includes the parts that are decided to be output.
* Cell State Update: The cell state is updated using the forget gate and the input gate. The updated cell state is then used to compute the hidden state which is passed on to the next time step and also as the output of the current time step.

➢ LSTM Model Architecture

When we build an LSTM model, typically, we would stack LSTM cells together and may add additional layers such as:

- Embedding Layer (if working with text data): Converts words to dense vectors of fixed size.
- LSTM Layers: One or more LSTM layers stacked on top of each other.
- Dense Layer: A fully connected layer to interpret the features learned by the LSTM layer(s).
- Output Layer: Depending on the task, it could be a sigmoid layer for binary classification or a softmax layer for multi-class classification.

➢ Advantages of LSTM:

- Captures long-term dependencies better than traditional RNNs.
- Effective in dealing with sequential data like text.

➢ Limitations of LSTM:

- LSTMs can be computationally expensive and slower to train compared to simpler models like traditional RNNs or even CNNs.
- May require a large amount of data to train effectively.

In summary, LSTMs are a powerful tool for sentiment analysis due to their ability to capture complex dependencies in text, making them a popular choice in natural language processing tasks.

3.5 Sentiment Analysis on NEP 2020 by LSTM:

Following are the steps to analyze sentiments on NEP 2020 by Using LSTM for Sentiment Analysis(Figure 6)

Industrial Engineering Journal
ISSN: 0970-2555
Volume : 54, Issue 3, No.1, March : 2025

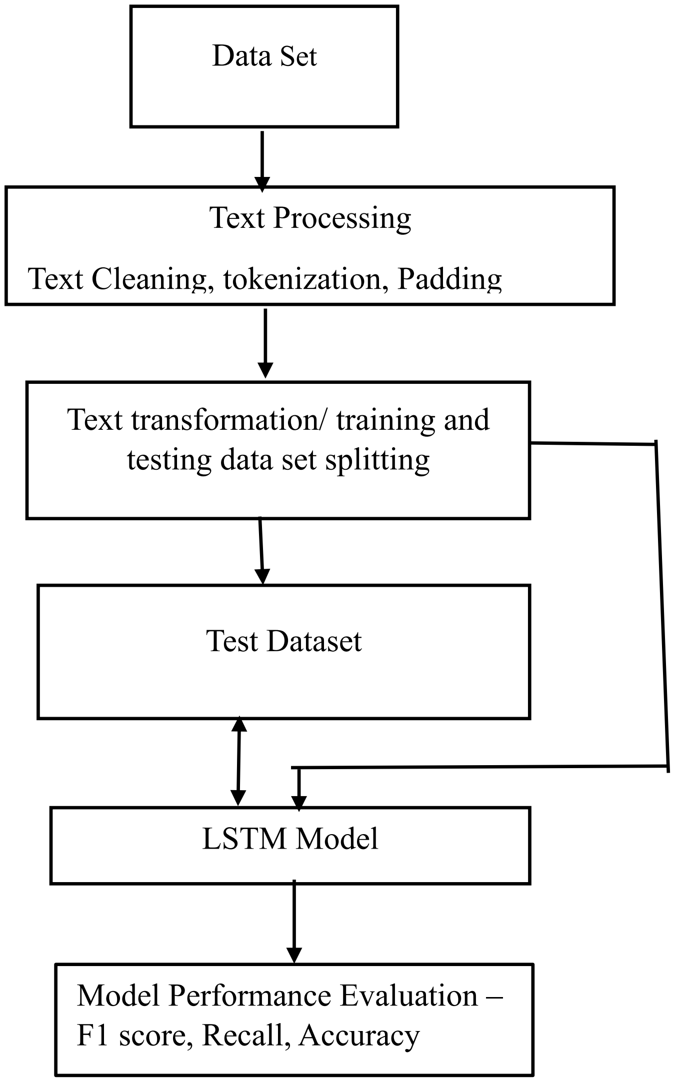

Figure 6: Steps in LSTM Model Analysis

i. DataSet: The dataset is taken from Kaggle(https://www.kaggle.com/datasets/rishabh6377/india-national-education-policy2020-tweets-dataset).

ii. Data Preparation (Data Preprocessing):

- Text Cleaning: Remove unnecessary elements such as special characters, numbers, HTML tags, and stopwords from the text.
- Tokenization: Break down the text into individual words or tokens.
- Padding: Ensure that all sequences are of the same length by padding shorter sequences with zeros or truncating longer ones.
- Text to Sequences: Convert words into numerical sequences (using tools like the Tokenizer from Keras or CountVectorizer/TF-IDF Vectorizer).
- Word Embeddings: Represent words in a dense vector space using pre-trained embeddings like GloVe, Word2Vec, or train your own embeddings with the model.

iii. Train-Test Split:Splitting the dataset into training, validation, and test sets. Typically, an 80/20 or 70/30 split is used for training and testing, respectively. In our Research 70% dataset is taken as training and 30% dataset taken as testing.

iv. LSTM Model Design

- Input Layer: This is where the pre-processed text sequences are input into the network.

Industrial Engineering Journal
ISSN: 0970-2555
Volume : 54, Issue 3, No.1, March : 2025

- o Embedding Layer: Use this layer to convert the input tokens into dense vectors of fixed size (embedding layer can be initialized with pre-trained embeddings or trained from scratch).
- o LSTM Layer(s): Add one or more LSTM layers that will capture the sequential dependencies and context in the text.
- o Dense Layer(s): After the LSTM, add fully connected (Dense) layers. The final Dense layer should have the appropriate number of neurons corresponding to the number of sentiment classes (e.g., 1 neuron with a sigmoid activation for binary classification, or multiple neurons with softmax for multi-class).
- o Activation Functions: Use ReLU or Tanh for intermediate layers and Sigmoid or Softmax for the output layer.

v. Compile the Model
- o Loss Function: Use categorical_crossentropy for multi-class classification.
- o Optimizer: Use optimizers like Adam or RMSprop to minimize the loss function.

vi. Train the Model: Train the model on the training set using the validation set to monitor overfitting.Use callbacks like EarlyStopping to stop training when the model stops improving on the validation set.

vii. Evaluate the Model: Evaluate the model's performance on the test dataset. Check metrics like accuracy, precision, recall, and F1-score to assess how well the model generalizes to unseen data.

viii. Fine-tuning and Hyperparameter Optimization
- o Adjusting hyperparameters: Experiment with the number of LSTM layers, number of neurons, learning rate, batch size, and epochs.
- o Regularization: Use techniques like dropout, L2 regularization, or batch normalization to prevent overfitting.

ix. Make Predictions: Use the trained LSTM model to predict the sentiment of new or unseen text data.

3.6. Proposed Hybrid Machine Learning Approach (LDN-LSTM): This model, combines traditional machine learning approach and deep learning technique. Here we use Logistic Regression , Decision Tree and Naïve Bayes Machine Learning Approach and LSTM, the Deep Learning Approach.

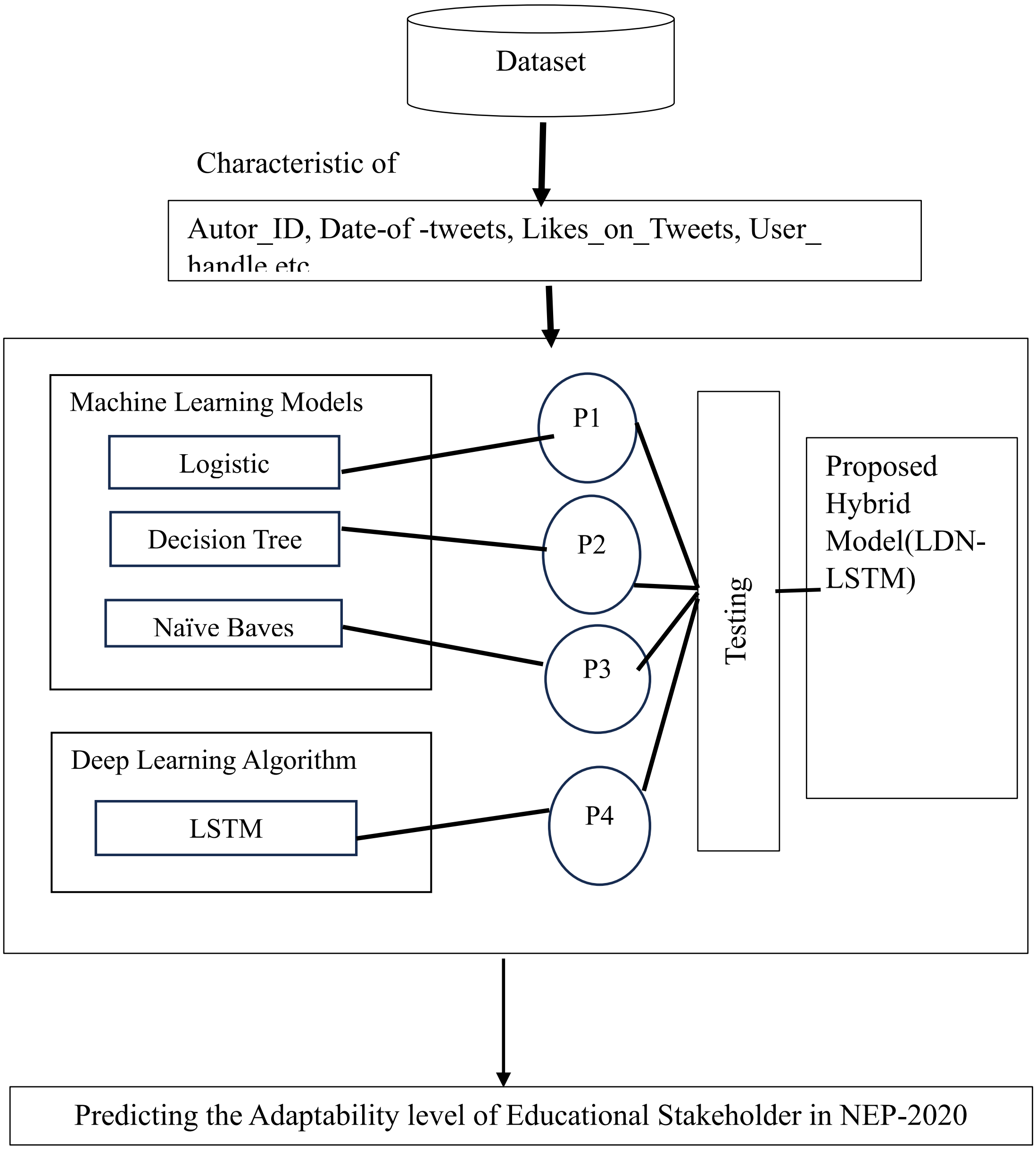

Figure 7 : Proposed Hybrid Machine Learning Models

The proposed model was trained and evaluated using LSTM and other classifiers like logostic regression, Decision Tree and naïve Bayes. We use ReLUfor Dense Layer activation and softmax for output layer, with Dense layer units tailored to the dataset's sizeas shown in Figure 7. LSTM parameters are given in following Table 2. Overfitting is crucial in Deep learning model, training arising when input layer outnumber output layer or when neuron count in hidden layer, deviates from a set of thresholds. Prior setting of these parameters areessential before training. To counter over-fitting, we used a dropout layer. Furthermore, sufficient training data is needed, with 70% of the dataset being allocated for training and 30% for testing.

Table 2: LDN-LSTM Parameter and Value

Parameter	Values
Epoch	500
Batch size	32
No.of Layers	3
Loss Function	MSE
Learning rate	0.001
Dropout Rate	0.5
No. of hidden unit	20
Training	70%
Testing	30%

IV.Result and Discussion

4.1 Performance Metrices of Models: the performance of model is evaluated on following Performance metrices:

- **Accuracy**: It measures the ratio of correctly predicted instances to the total instances in the dataset.

$$\text{Accuracy}=\frac{TP+TN}{TP+TN+FP+FN}$$

Where TP, TN, FP and FN are True Positive, True Negative, False Positive and False Negative Respectively.

- **Precision**: It helps in evaluating the exactness or quality of the positive predictions made by the model. A higher precision value indicates that there are fewer false positives, meaning the model is more accurate when it predicts an instance as positive.

$$\text{Precision}=\frac{TP}{TP+FP}$$

Where TP and FP are True Positive and False Positive respectively.

- **Recall:** It measures the proportion of true positive results among the actual positive instances in the dataset. A higher recall value indicates that the model can successfully identify more relevant instances in the dataset.

$$\text{Recall}=\frac{TP}{TP+FN}$$

Where TP and FN are True Positive and False Negative respectively.

- **F1-Score**: This metric combines both precision and recall value. It provides a balance between two measures and is particularly useful when we seek a balance between precision and recall. The F1-Score considers both false positives and false negatives and is especially valuable when the class distribution is uneven.

$$\text{F1-score}=\frac{2*(Precision*Recall)}{Precision+Recall}$$

- **Macro Average F1-Score**: It calculates the average F1-Score for each class and then computes the mean, giving equal weight to each class, regardless of its size.
- **Macro Average Precision**: It calculates the average precision for each class and then computes the mean, giving equal weight to each class, regardless of its size.
- **Macro Average Recall**: It calculates the average Recall for each class and then computes the mean, giving equal weight to each class, regardless of its size.

Table 3 to Table 12 shows the confusion matrix of Naïve Bayes, Logistic Regression, Decision Tree , LSTM and LDN-LSTM.

Industrial Engineering Journal
ISSN: 0970-2555
Volume : 54, Issue 3, No.1, March : 2025

Table 3: Confusion matrix of Naive Bayes (Training Set)

N=12768	Positive	Neutral	Negative
Positive	1038	89	472
Neutral	27	2384	561
Negative	1127	4527	6081

Table 4: Confusion matrix of Naive-Bayes (Testing Set)

N=5472	Positive	Neutral	Negative
Positive	533	44	235
Neutral	13	1176	348
Negative	27	50	3046

Table 5 : Confusion matrix of Logistic Regression(Training set)

N=12768	Positive	Neutral	Negative
Positive	1881	113	280
Neutral	12	2810	362
Negative	34	150	7126

Table 6 : Confusion matrix of Logistic Regression (Testing Set)

N=5472	Positive	Neutral	Negative
Positive	605	63	144
Neutral	7	1380	150
Negative	16	57	3050

Table 7: Confusion matrix of Decision Tree (Training Set)

N=12768	Positive	Neutral	Negative
Positive	1901	117	239
Neutral	15	2830	369
Negative	30	154	7113

Table 8: Confusion Matrix of Decision Tree (Testing Set)

N=5472	Positive	Neutral	Negative
Positive	710	35	67
Neutral	32	1455	50
Negative	59	74	2990

Industrial Engineering Journal
ISSN: 0970-2555
Volume : 54, Issue 3, No.1, March : 2025

Table 9: Confusion matrix of LSTM (Training Set)

N=12768	Positive	Neutral	Negative
Positive	2061	175	248
Neutral	18	2845	210
Negative	21	50	7140

Table 10: Confusion Matrix of LSTM (Testing Set)

N=5472	Positive	Neutral	Negative
Positive	721	30	61
Neutral	26	1541	20
Negative	22	40	3011

Table11: Confusion Matrix of LDN-LSTM(Training Set)

N=12768	Positive	Neutral	Negative
Positive	2068	170	149
Neutral	13	2851	139
Negative	16	131	7231

Table 12: Confusion Matrix of LDN-LSTM(Testing Set)

N=5472	Positive	Neutral	Negative
Positive	801	24	58
Neutral	24	1377	66
Negative	28	32	3062

Industrial Engineering Journal
ISSN: 0970-2555
Volume : 54, Issue 3, No.1, March : 2025

Table 13: Comparison of statistical parameters for Bernoulli's Naïve-bayes, Logistic Regression ,Decision Tree, LSTM and LND-LSTM, class-wise.

Performance Metrices	Class Name	Bernoulli Naïve Bayes		Logistic Regression		Decision Tree		LSTM		LDN-LSTM	
		Training Set	Testing Set	Training Set	Testing Set	Training Set	Testing Set	Training Set	Testing Set	Training Set	Testing Set
Precision	Positive	0.640	0.650	0.827	0.745	0.842	0.874	0.829	0.887	0.866	0.907
	Neutral	0.80	0.765	0.882	0.897	0.880	0.842	0.925	0.971	0.949	0.980
	Negative	0.741	0.975	0.974	0.976	0.974	0.957	0.990	0.979	0.980	0.961
Recall	Positive	0.921	0.930	0.976	0.963	0.976	0.886	0.980	0.937	0.986	0.939
	Neutral	0.526	0.925	0.914	0.920	0.9126	0.930	0.926	0.956	0.904	0.960
	Negative	0.854	0.839	0.917	0.912	0.912	0.962	0.939	0.973	0.961	0.961
F1-Score	Positive	0.755	0.765	0.895	0.840	0.904	0.879	0.897	0.911	0.922	0.922
	Neutral	0.634	0.837	0.897	0.908	0.896	0.883	0.925	0.963	0.925	0.970
	Negative	0.793	0.901	0.945	0.943	0.947	0.959	0.963	0.975	0.970	0.970
Accuracy	Positive	0.949	0.941	0.965	0.979	0.968	0.964	0.963	0.928	0.972	0.976
	Neutral	0.786	0.916	0.950	0.949	0.948	0.965	0.964	0.932	0.964	0.974
	Negative	0.753	0.879	0.935	0.932	0.937	0.954	0.953	0.928	0.965	0.967

Table 14: comparisons of Naive bayes, Logistic Regression and Decision, LSTM and LND-LSTM Tree Model based on different Performance metrics

Performance Metrics	Bernoulli's Naïve-Bayes		Logistic Regression		Decision Tree		LSTM		LND-LSTM	
	Training Set	Testing Set	Training Set	Testing set	Training Set	Testing Set	Training Set	Testing Set	Training Set	Testing Set
Macro Average Precision	0.727	0.796	0.894	0.872	0.898	0.882	0.914	0.945	0.931	0.949
Macro Average Recall	0.767	0.898	0.935	0.933	0.936	0.926	0.948	0.955	0.950	0.953
Macro Average F1-Score	0.727	0.834	0.912	0.892	0.915	0.90	0.928	0.949	0.939	0.954
Classifier Accuracy	0.744	0.868	0.925	0.920	0.927	0.942	0.943	0.920	0.951	0.959

The dataset underwent training and testing with a diverse set of Machine Learning algorithms for prediction and analysis. These algorithms include Logistic Regression, Decision Tree, Naïve Bayes, and LSTM. The objective was to predict the adaptability level of educational stakeholders in the context of NEP 2020. A proposed model LDN-LSTM , A hybrid model is introduced with the aim of analysing sentiments of educational stakeholders on NEP 2020. In our assessment of model performance, we utilized four evaluation metrics: Precision, Recall, Accuracy, and F1- Score as shown in Table 13. Table 13 shows class-wise accuracy, precision, recall and accuracy.The results demonstrate that the LDN-LSTM, hybrid model proposed in this study exhibited the highest levels of accuracy of 95.90 %, recall of 95.30%, precision of 94.90%, and F1-score 95.40%. Subsequently, the LSTM, Decision Tree and Logistic Regression classifier shows better performance than Naïve Bayes. Table 14 shows the macro average precision, macro average recall, macro average F1-score and overall accuracy.

Accuracy of 95.90%, Precision 94.90% indicates the ability of a model to make accurate positive predictions as shown in Figure 8. Proposed LDN-LSTM (hybrid model), LSTM, Logistic Regression and Decision Tree all exhibit high precision values. This suggests that these models are highly reliable in correctly analysing sentiments and adaptability level without generating many false positive predictions. Recall, also known as sensitivity, assesses the model's ability to correctly identify all positive instances. Across all models, there is consistently high recall, with values of 95.30%. This signifies that the models are effective in capturing the majority of educational stakeholders' adaptability level within the dataset. The F1-Score is a balanced metric that considers both precision and recall. It provides a harmonic mean of these two metrics and is particularly useful when dealing with imbalanced datasets. All models, achieved good F1-Scores of 83 % or higher, indicating strong overall performance.

V. Conclusion

In this Paper, sentiment analysis on National Education Policy (NEP)-2020 is interpreted.by different classifiers, we have analyzed the performance of different machine learning models and Deep Learning models on implementation of NEP in educational sectors. The Naïve bayes shows 86.80% accuracy, Logistic Regression shows 92.00% accuracy, Decision tree shows 94.20% accuracy, LSTM shows 92.00% accuracy. The highest accuracy shown by our proposed classifier (LDN-LSTM) with accuracy 95.50%, as shown in Figure 8.

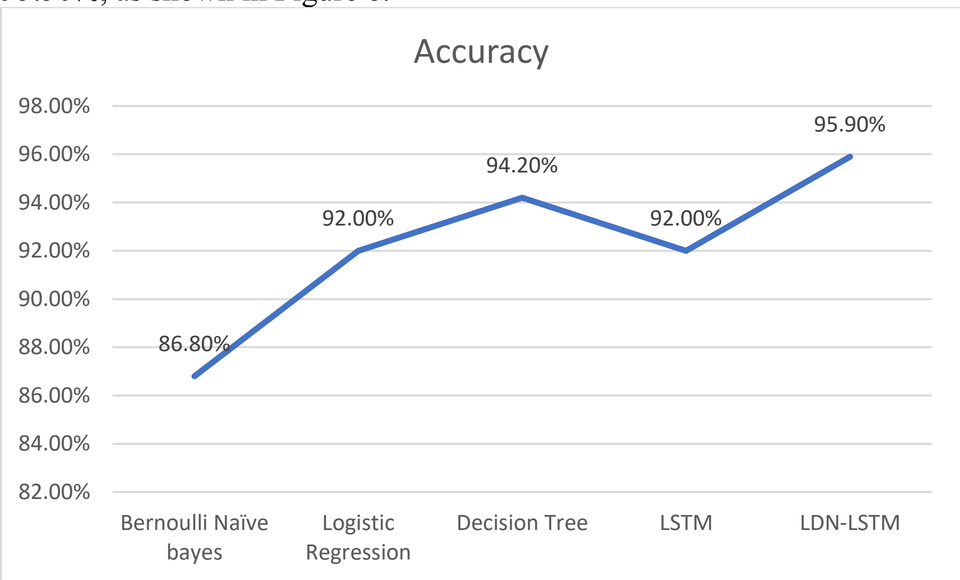

Figure 8: Classifier's Accuracy

References:

[1] Barron-Estrada María Lucia; Zatarain-Cabada Ramón; Oramas-Bustillos Raúl. Emotion Recognition for Education using Sentiment Analysis, Research in Computing Science 148(5), 2019, ISSN 1870-4069.

[2] Chandra Akana.; Srinivas Mouli Venkata; Satyanarayana Ch.; Divakar Ch.; Sirisha Katikireddy Phani.;Sentiment Analysis using Neural Network and LSTM, IOP Conf. Series: Materials Science and Engineering 1074 (2021).

[3] Gopalakrishnan Karthik.;Salem Fathi M. Sentiment Analysis Using Simplified Long Short-term Memory Recurrent Neural Networks, Published in arXiv.org 8 May 2020, Computer Science

[4] Kandhro Irfan Ali.; Wasi Shaukat; Kumar Kamlesh ; Rind Malook .;Ameen Muhammad.; Sentiment Analysis of Students' Comment using Long-Short Term Model, Indian Journal of Science and Technology, ISSN (Print) : 0974-6846, Vol 12(8), DOI: 10.17485/ijst/2019/v12i8/141741, February 2019

[5] Kastrati Z.; Dalipi F.; Imran A.S.;Pireva Nuci K.; Wani M.A. Sentiment Analysis of Students' Feedback with NLP and Deep Learning: A Systematic Mapping Study. Appl. Sci. 2021, 11, 3986. https://doi.org/10.3390/ app11093986

[6] Mao Jun.;Qian zhe,;Lucas terry. Sentiment Analysis of Animated Online Education Texts Using Long Short-Term Memory Networks in the Context of the Internet of Things, IEEE Open Access, Vol 10, 2023.

[7] Murthy G. S. N.; Allu Shanmukha Rao.; Andhavarapu Bhargavi.; Bagadi Mounika.; Belusonti Mounika.Text based Sentiment Analysis using LSTM, International Journal of Engineering Research & Technology (IJERT) ,ISSN: 2278-0181 , Vol. 9 Issue 05, May-2020.

[8] Rani Seema.;Bhagwan Jai.; Kumar Sanjeev Chaba Yogesh.;Godara Sunila,; Sindhu Sumit. Sentiment Analysis using a Multinomial LR-LSTM Model, International Journal of Intelligent systems and Applications in engineering, ISSN:2147-67992

[9] Shobana J.; Murali M.; An effcient sentiment analysis methodology based on long short-term memory networks", Complex & Intelligent Systems (2021) 7:2485–2501

[10] Tholusuri Ashok;Anumala Manish;Malapolu Bhagyaraj;Jaya Lakshmi G., Sentiment Analysis using LSTM, International Journal of Engineering and Advanced Technology (IJEAT) ISSN: 2249-8958 (Online), Volume-8 Issue-6S3, September 2019.

[11] Yuan Hongli; Alexander Herhandez A.Sentiments Analysis of student evaluation of Teaching Based on Bi-LSTM algorithm, IEEE 15th International Conference on Advanced Infocomm Technology", 2023, DOI: 10.1109/ICAIT59485.2023.10367267

[12] Sridhar S.;Vijayalakshmi M.; Machine learning, Oxford Higher Education(2021).

[13] https://databasecamp.de/en/ml/naive-bayes-algorithm

[14] https://www.javatpoint.com/machine-learning-decision-tree- classification-algorithm

[15] https://medium.com/@MudSnail/the-importance-of-logistic-regression-in-image-classification-1966d07e7a0c

[16] https://medium.com/@ottaviocalzone/an-intuitive-explanation-of-lstm-a035eb6ab42c