International Journal on Recent and Innovation Trends in Computing and Communication
ISSN: 2321-8169 Volume: 11 Issue: 9
***Article Received:** 05 July 2023 **Revised:** 25 August 2023 **Accepted:** 15 September 2023*

Feature Selection based Sentiment Analysis on US Airline Twitter Data

[1] Shiramshetty Gouthami, [2] Prof. Nagaratna P. Hegde
[1] Research Scholar, Computer Science and Engineering, Osmania University, Hyderabad, India.
[1] gouthami.shiramshetty@gmail.com
[2] Professor, Computer Science and Engineering ,Vasavi College of Engineering, Hyderabad, India
[2] nagaratnaph@staff.vce.ac.in

Abstract: Review document emotions are classified using sentiment analysis. Researchers grade features to remove non-informative and noisy attributes with low grades to improve classification accuracy. This paper utilizes six different NLP models to predict user sentiments based on Twitter reviews about airlines, using the Twitter US Airline Sentiment dataset. The best-performing models from both machine learning (K-nearest neighbor, Random forest, and Multinomial Naive Bayes) and deep learning (Artificial Neural Networks, LSTM, and Bidirectional LSTM with Glove embeddings) were implemented through Anaconda and Google Colab platforms. This paper introduces a new type of feature dimensionality technique termed "inquiry extension grade (IEG)," inspired by the inquiry extension term weighting technique. Additionally, we modified the traditional TF-IDF method, referred to as "improved TF-IIDF (IFFIDF)," specifically tailored for processing unbalanced text collections. To assess the effectiveness of the proposed methods, a series of simulations were conducted. The results indicate that the combination of IEG-ITFIDF Vectorization and Bi-LSTM with Glove embeddings yielded the best accuracy of 94.26% in sentiment classification for the Twitter US Airline Sentiment dataset.

Keywords- . Bi-LSTM; sentiment analysis; machine learning: Feature selection ; attention mechanism

I. INTRODUCTION

Social media has become an integral part of modern human life, with people sharing their daily activities, emotions, and day-to-day lifestyle on several social media platforms [1]. Among these platforms, Twitter stands out as the most widely used for social media surveillance, capturing about 64% of users. Twitter data holds immense value for gathering and analyzing user perspectives, including those related to the airline industry. The airline industry recognizes the importance of staying updated and connected with customers, especially when travel resumes after the pandemic. Traditional customer feedback methods are common but time-intensive [2]. To address this, sentiment analysis has emerged as a crucial approach to swiftly understanding user input and opinions. Since the airline business has changed so much in the previous decade, thousands of travelers express their daily journeys [3]. These real-time feedbacks, from positive ones with pictures of clouds and staff to complaints about service issues like missing baggage, delayed flights, and IT system failures, provide valuable insights for passengers in their decision-making process.

Additionally, this feedback allows airline management teams and staff to analyze situations promptly and take necessary actions to improve services and enhance passenger experiences. Twitter data, in particular, serves as a valuable resource for collecting user tweets and conducting sentiment analysis [4, 5]. Numerous research studies have examined sentiment analysis utilizing diverse data preparation, feature selection, and classification methods [6-10]. Some feature extraction methods remove irrelevant or unneeded features to improve classification performance [11]. Review emotions are classified using supervised learning [12, 13].

The paper introduces a new feature selection method called "inquiry extension grade (IEG)," specifically tailored to reduce the dimensionality of the feature space in sentiment analysis problems. The goal is to examine whether these feature selection methods can effectively reduce feature sizes and enhance sentiment classification accuracy across different document domains, languages, and classifiers. Additionally, the paper proposes an enriched version of TF-IDF, named "TF-IADF," designed to handle imbalanced data distribution, often found in internet media reports. Three other term weighting schemes based on the same idea are also presented to improve sentiment analysis performance under imbalanced conditions..

The remaining sections of the paper are organized as follows: Section 2 reviews related work on sentiment analysis; Section 3 introduces the methods used in the study, including the newly proposed feature selection method (IEG). Section 4 details experimental settings, datasets, performance measures, and testing results. Finally, Section 5 concludes the paper, summarizing the findings and implications of the research.

Article Received: *05 July 2023* ***Revised:*** *25 August 2023* ***Accepted:*** *15 September 2023*

II. RELATED WORK

This section provides a thorough examination of earlier work on the subject of sentiment analysis as well as the models currently in use. In addition, three fundamental methods are presented: word embedding, convolutional neural networks (CNN), and bidirectional long short-term memory.

In [14], the authors investigated three machine learning algorithms to conduct sentiment analysis on a dataset acquired from US airline Twitter data via Kaggle, including decision trees, support vector machines (SVM), and neural networks. Their analysis was carried out using ML methods. An evaluation of the efficacy of several ML algorithms revealed that the neural network-based technique achieved the highest accuracy, coming in at 75.99%.

In [15], the authors aimed to increase sentiment analysis accuracy by employing several classification algorithms. They used precision, recall, f1-score, micro average, macro average, and accuracy to evaluate the performance of the various classifiers. In addition, they introduced a novel ensemble bagging technique for the various classifiers. They also computed the classification accuracy and the average predictions produced by the classifiers to ensure the results were high quality. Regarding accuracy, bagging classifiers performed much better than non-bagging classifiers, according to the data.

In [16], the authors constructed a voting classifier (VC) to conduct sentiment analysis within companies. The VC used LR and a "stochastic gradient descent classifier "to make a final prediction and a voting system based on soft votes. Furthermore, the impact of several feature extraction methods on classification accuracy was explored: TF, TF-IDF, and word2vec. The dataset was also used to assess the effectiveness of LSTM. According to the data, the suggested VC outperformed other classifiers, achieving accuracies of 0.789 and 0.791 when TF and TF-IDF feature extraction was performed. Nonetheless, the LSTM model performed less well compared to the machine learning classifiers.

In [17], the authors presented Binary Cuckoo Search as a binary version of cuckoo search for optimal feature selection in sentiment analysis of textual online content. Their study used supervised learning methodologies such as support vector machines (SVM) with the traditional tf-idf model and the novel feature optimization approach. Based on the accuracy of the results, it was concluded that the proposed binary cuckoo search for feature selection optimization outperformed simple supervised algorithms utilizing the standard tf-idf score.

In [18], the authors presented the HMRFLR model, a hybrid model. This model combined many ML classifiers. This hybrid model assessed tweets about US airlines, which were classified as positive, neutral, or negative based on how positive, neutral, or negative they were. Their objective is to establish the amount of satisfaction that customers now have with airlines. Individual accuracy scores for Logistic Regression and Random Forest were 79.1% and 76.87%, respectively, but the hybrid model had an improved accuracy score of 88.16%.

In [19], the authors presented DL efficiently combined various word embeddings for multi-class SA of tweets associated with six important airlines in the United States. The methodology employed CNN pre-processing methods, as well as tweet-cleaning algorithms, in addition to DNN raw data extraction. They assessed positive, negative, and neutral tweet interpretations in conjunction with a three-class dataset and precision evaluation. The data demonstrated that the suggested model outperformed its predecessors, making it a more reliable tool for sentiment analysis.

Regarding term frequency (TF), it considers a local document's phrase count to assess whether terms with a higher frequency are more significant than others. However, it needs to understand the frequency of record collecting and may be unable to distinguish between significant and irrelevant recordings. To improve the discriminative ability of text classification words, the inverse document frequency (IDF) was suggested to be employed and linked to the collection frequency. The document frequency (DF) measure, which reflects document count, consists of a term.

III. THE PROPOSED WORK

This section presents the architecture of our novel model, Bi-LSTM-SA. The overall structure is illustrated in Figure 1, encompassing several essential components, including data processing, feature vectorization, and classification.

Dataset

US airline customer sentiment comprises 14,640 rows and 15 columns. This study analyses the first two columns of this table. The first column contains sentiment labels, while the second column has passenger text reviews. 9,178 rows are negative, 3,099 are neutral, and 2,363 are positive, negative, and neutral sentiments are labeled. The proposed model uses 9,516 and 5,124 data rows for training and testing. Kaggle (https://www.kaggle.com/welkin10) and Crowd Flower (https://www.data.world/crowdflower/) host the US airline sentiment dataset. https://www.data.world/ has these resources.

Table 1 shows US airline attitudes by dataset [20].

Airlines	**Negative**	**Sentiment Neutral**	**Positive**
American	1960	463	336
Delta	955	723	544
Southwest	1186	664	570
US airways	2263	381	269
United	2633	697	492

***Article Received:** 05 July 2023 **Revised:** 25 August 2023 **Accepted:** 15 September 2023*

Virgin America	181	171	152

Data Preprocessing

The first part of the experiment, known as "data preprocessing," aims to separate the text data properties that are most significant to sentiment prediction. Textual data frequently include errors in spelling, POS labeling, slang, exclamation marks, acronyms, punctuation marks, and other linguistic features. The primary purpose of preprocessing is to remove redundant or irrelevant text data, as well as conflicts, distortions, and contradictions. Depending on the circumstances, text data can be obvious, distorted, inconsistent, or confusing. As a result, text data preparation is critical to analyze similarities and show them in a form suitable for subsequent analysis. The present study uses data preprocessing to remove stop words and airline-specific data. Still, critical phrases like "nor," "not," "no," and so on, which are likely to convey negative attitudes. This was done for the results to be generalized. Afterward, the textual data underwent stemming and lemmatization processes.

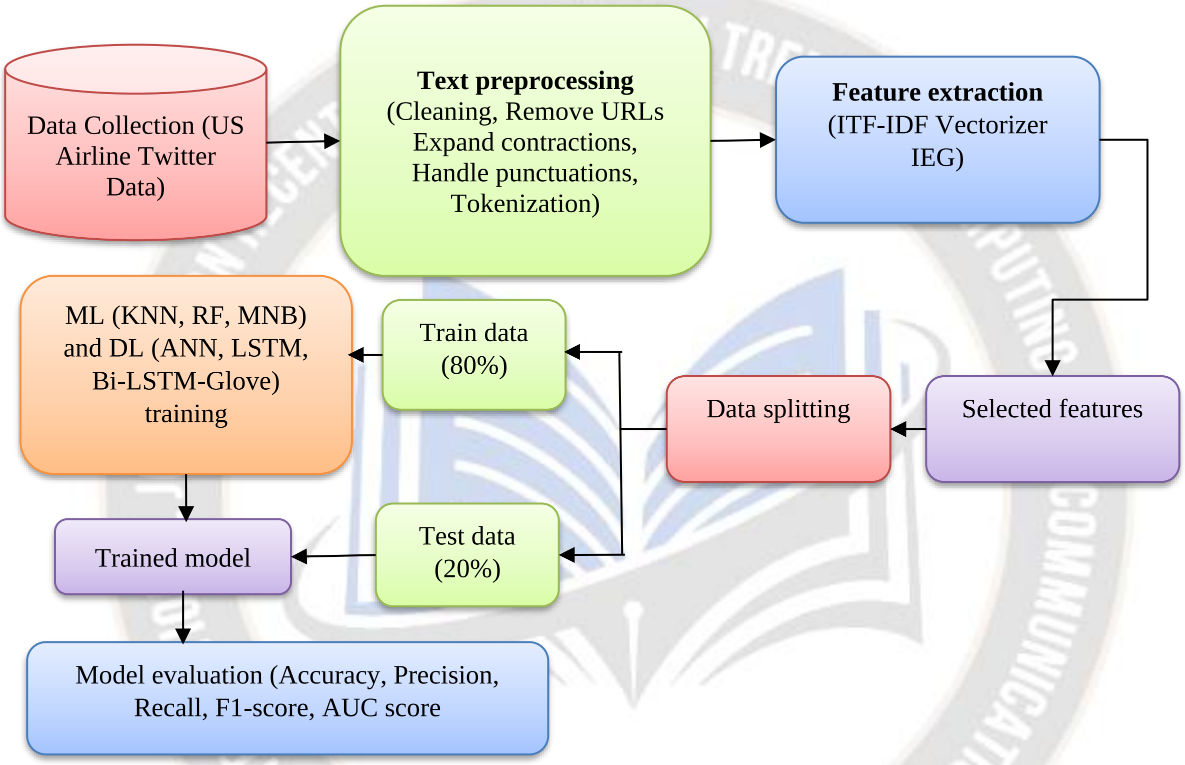

Data Collection (US Airline Twitter Data)
Text preprocessing (Cleaning, Remove URLs Expand contractions, Handle punctuations, Tokenization)
Feature extraction (ITF-IDF Vectorizer IEG)
Selected features
Data splitting
Train data (80%)
Test data (20%)
ML (KNN, RF, MNB) and DL (ANN, LSTM, Bi-LSTM-Glove) training
Trained model
Model evaluation (Accuracy, Precision, Recall, F1-score, AUC score

Figure 1: The overall structure flow of the methodology

Additionally, regular expressions (re) were used to handle word repetition and perform data cleaning, including removing usernames, punctuation marks, HTML, emojis, and URLs. Furthermore, all the text data were converted to lowercase to ensure consistency. Then, text tokenizing into specific, resulting in 12,041 unique tokens extracted from the US airline sentiment text dataset.

Feature selection

This is important for the classification of text for SA [21], which evaluates the importance of features using specific measures, allowing the removal of non-informative features while preserving the most relevant ones, thereby improving model performance. This paper explores different FS techniques, such as Bag-of-Words (BoW), Word2Vec (W3Vec), and our newly proposed Inquiry Extension Grade (IEG), in order to compare their effectiveness for sentiment analysis.

Proposed ITF IDF:

The IDF value of a particular term can be calculated using the following formula:

$$IDF(t, d, D) = log\frac{|D|}{DF(t, D)} \tag{1}$$

In equation (2), DF(*t*, *D*) denotes the document frequency value of the term *"t"* in the corpus *"D."* The symbol *"N"* in equation (2) represents the total number of documents in corpus "*D*." To prevent the issue of infinite values in extreme cases, the formula is sometimes optimized, as illustrated in the following:

$$IDF(t, d, D) = log\frac{|D| + 1}{DF(t, D) + 1} \tag{2}$$

The traditional form of TF-IDF can be represented as:

$$TF - IDF(t, d, D) = TF(t, d) * IDF(t, d, D) \tag{3}$$

In equation (3), *TF − IDF(t, d, D*) is the term *t* weight of document *d* in corpus *D*, while *TF(t, d)* is its *TF* value.

The IDF formula, presented in equations (1) and (2), illustrates that terms from greater sizes will be allocated lower values than terms from other categories when the corpus is not highly balanced. Even while some low-document-frequency terms are meaningless, their IDF values are far higher than others, which is inaccurate. We focus on the deviation since a term's discriminative capacity is limited when its DF value deviates from the corpus average. Fixing the previous problems required adding *ADF* to the collection frequency factor. This study's corpus's average *DF* value is *DF*, while term t's *ADF* value in document *D* is *ADF(t, D*). Equations (4) and (5) calculate them for *n* terms:

$$DF = \frac{\sum DF(t, D)}{n} \tag{4}$$

$$A_{DF}(t, D) = \frac{(DF(t, D) - DF)^2}{n} \tag{5}$$

ADF extends DF, hence replacing DF with ADF in the calculation optimises IDF. A new collection frequency formula is:

$$IADF(t, D) = log\frac{\lfloor D \rfloor + 1}{A_{DF}(t, D) + 1} \tag{6}$$

In most conditions, the IDF method works; we must alter it for extreme cases. Another unique formula, equation (7), uses ADF to minimize the weight of terms with very high or low DF values.

$$IADF^*(t, D) = log\frac{[D] + 1}{DF(t, D) + 1} * \frac{1}{log(A_{DF}(t, D) + 1)} \tag{7}$$

As mentioned earlier, the two ADF-based methods have shown potential for enhancing the TC (Text Classification) performance. However, there are limitations arising from the variance, where excessively large or small sizes lead to relatively small or large variances. Such extreme values for terms can noticeably impact the TC performance. To address this, we optimized further by normalizing the ADF to mitigate the impact of extreme term values. Initially, *ADF*(*t*, *D*) was modified following equation (8), and subsequently, a normalization formula was applied as depicted in equation (10):

$$A'_{DF}(t, D) = log\frac{1}{(A_{DF}(t, D) + 1)} + 1 \tag{8}$$

$$A'_{DF}(t, D) = \frac{A'_{DF}(t, D) - min\left(A'_{DF}(t, D)\right)}{max\left(A'_{DF}(t, D)\right) - min\left(A'_{DF}(t, D)\right)} \tag{9}$$

Based on ADF″ , Two additional novel formulas, represented by equations (11) and (12), have been devised. These formulas incorporate an optional weight proportion α (with a default value of 1) to adjust the significance of ADF″ in various scenarios.

$$IADF_{norm}(t, D) = log\frac{\lfloor D \rfloor + 1}{A'_{DF}(t, D) + 1} \tag{10}$$

$$IADF^{+}_{norm}(t, D) = log\frac{\lfloor D \rfloor + 1}{DF(t, D) + 1} \tag{11}$$

Four novel word weighting strategies are illustrated in equations (12)–(15):

$$TF - IADF(t.d, D) = TF(t, d) * IADF(t, D) \tag{12}$$

$$TF - IADF^*(t.d, D) = TF(t, d) * IADF^*(t, D) \tag{13}$$

$$TF - IADF_{norm}(t.d, D) = TF(t, d) * IADF_{norm}(t, D) \tag{14}$$

$$TF - IADF_{norm}(t.d, D) = TF(t, d) * IADF^{+}_{norm}(t, D) \tag{15}$$

This model replaces the IDF component of the TF-IDF approach with four additional calculation formulas. Four unique term weighting strategies improve uneven text processing.

Inquiry Extension Grade (IEG)

This feature selection approach is based on information retrieval (IR) inquiry expansion methods. This approach finds more relevant documents for an inquiry that gains new terms to find more relevant documents, and the extended inquiry uses terms from relevant documents. This approach rates terms from relevant documents for an inquiry, designated I. The highest-scoring terms extend the initial inquiry, improving information retrieval accuracy. The information retrieval system receives the initial inquiry, I, and returns relevant documents. Subsequently, the user marks the relevant documents, and all relevant terms are retrieved and evaluated using a scoring algorithm [22]. We chose the top k terms with the highest scores as the most valuable ones to extend the inquiry. The resulting extended inquiry, I', comprises the original terms along with the k new terms and their corresponding scores. This expanded inquiry, I', is then submitted to the information retrieval system, which returns additional relevant documents related to the initial query, I. We compute the score f using Equation 10. This scoring algorithm helps to identify the most significant terms and includes them in the extended inquiry, thereby improving the precision of the information retrieval process.

Multinomial Naive Bayes:

Sentiment analysis uses this simple but efficient probabilistic learning approach. The Bayes theorem asserts that the likelihood of an event (e.g., a text being positive) given certain evidence (e.g., the words in the text) is proportional to the probability of the evidence times the prior probability of the event. This uses words as evidence and text sentiment as the event. The event's prior probability is the text's positivity or negativity, derived using a training dataset. Counting how often each word appears in positive and negative sentences in the training dataset estimates this likelihood. Multinomial Naive Bayes sentiment analysis mathematical model:

$P(sentiment \mid words) = P(words \mid sentiment) * P(sentiment)$

where *sentiment* is the sentiment of the text (e.g., positive or negative)

words is the set of words in the text

$P(words \mid$ sentiment) is the probability of the words appearing in the text given that the sentiment is sentiment

$P(sentiment)$ is the prior probability of the sentiment

ML Models for Sentiment Analysis

Multinomial Naive Bayes: We performed estimating the probability of words appearing in a text based on a particular sentiment using the training dataset. This entails counting the occurrences of each word in both positive and negative texts. The prior probability of sentiment is determined by dividing the number of positive texts by the total number of texts in the training dataset. After training, the model can predict text sentiment. During the prediction process, the model calculates the probability of each sentiment for the new text and then selects the sentiment with the highest probability as the output.

Random Forest classifier: Random Forest uses ensemble learning to train numerous decision trees. The majority class predicted by individual trees determines the random forest's categorization result. In regression problems, the random forest delivers the average prediction of the individual trees. Mathematically, the random forest classifier is:

$$RF(a) = T\sum Tx(a) \qquad (16)$$

Where $RF(a)$ is the prediction of the random forest classifier for a new data point a.

T is the number of trees in the random forest.

$Tx(a)$ is the prediction of the xth tree in the random forest for the new data point a.

The prediction of the random forest classifier is the majority vote of the predictions of the individual trees. The random forest classifier predicts the class that is most common among the predictions of the individual trees.

K-Nearest Neighbors (KNN): KNN, often known as k-nearest neighbors, is a simple but effective classification and regression algorithm. It finds the k most similar cases in the training set to a new instance and predicts its label based on its k nearest neighbors. Converting a review or other text into a feature vector for sentiment analysis using KNN. Positive words, negative words, sentiment scores, and more may be included. The KNN method uses the labels of the k nearest neighbors to classify the text as positive, negative, or neutral.

• The algorithm calculates the Euclidean or Manhattan distance between the new instance and each instance in the training set.

• The *k* closest examples are the nearest neighbors.

• The *k* nearest neighbors' labels predict the new instance's label.

Adjusting *k*, a hyperparameter improves KNN algorithm performance. A smaller *k* number emphasizes the nearest neighbor labels, whereas a larger *k* value gives them more weight. Experimentation determines the best *k* value for the dataset and the task.

Deep Models for Sentiment Analysis

Artificial Neural Network (ANN): A straightforward yet effective architecture for numerous machine learning tasks involves an artificial neural network (ANN) comprising only dense layers and two dropout layers. The dense layers enable the ANN to grasp intricate relationships between input and output data, while the dropout layers serve to prevent overfitting. These dropout layers function by randomly deactivating (setting to zero) a subset of neurons within the dense layers during training. This forces the ANN to rely on the remaining neurons, thus avoiding excessive dependence on any specific set of features. The ANN consists of three dense layers, each containing 128, 64, and 10 neurons, respectively. ReLU serves as the activation function for each layer, which is a non-linear function facilitating the ANN's learning of complex relationships between input and output data.

Long Short-Term Memory (LSTM): This particular type of recurrent neural network (RNN) is highly suitable for sentiment analysis. RNNs excel at learning long-term dependencies in sequential data, which makes them particularly effective for tasks like understanding the sentiment of a piece of text. Among RNNs, Long Short-Term Memory networks (LSTMs) have a unique architecture that enables them to retain information from previous time steps. This characteristic is crucial for sentiment analysis, as the sentiment of a text can often be influenced by the words that precede it. For example, the word "but" can indicate a change in sentiment. LSTMs have demonstrated remarkable effectiveness in sentiment analysis. The steps involved in using LSTM for sentiment analysis are as follows:

Preprocess the text data.

· Create a vocabulary.

· Encode the text data.

***Article Received:** 05 July 2023 **Revised:** 25 August 2023 **Accepted:** 15 September 2023*

· Build the LSTM model, which includes specifying the number of LSTM layers, the number of hidden units in each layer, and the activation function.

· Train the LSTM model by feeding the encoded text data into it and adjusting its parameters to predict the sentiment of the text accurately.

· Evaluate the LSTM model by testing it on a held-out dataset and measuring its accuracy.

Bi-LSTM model with GloVe

The Bi-LSTM model with GloVe is a deep-learning architecture suitable for sentiment analysis. It comprises two major components:

GloVe word embeddings: GloVe is a technique used to represent words as vectors, effectively capturing both their semantic and syntactic relationships. This involves training a word-to-vector model on a vast text corpus. The resultant vectors are then utilized to represent words in the Bi-LSTM model.

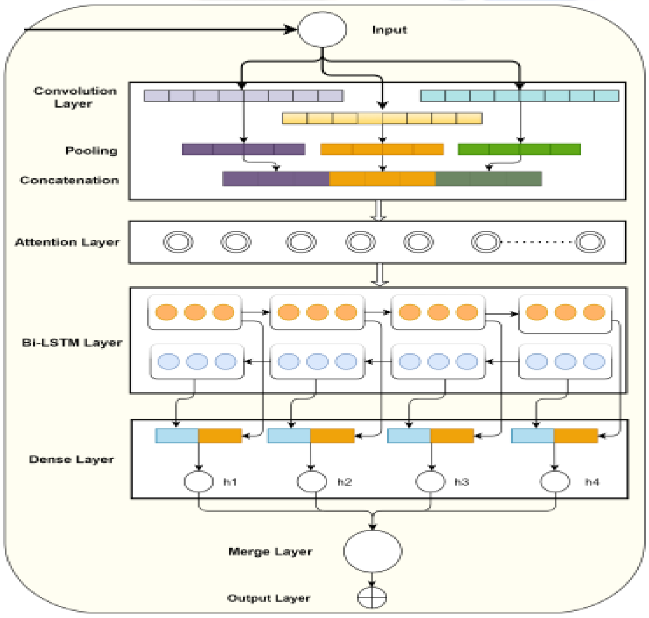


Figure 2: Bi-LSTM model with GloVe

Bidirectional LSTM: LSTM, a variant of recurrent neural networks, is adept at handling sequential data. The Bi-LSTM model employs two LSTM units—one reading. This dual directionality enables the model to grasp both the forward and backward context of a word, a crucial aspect for sentiment analysis, as depicted in Figure 2.

Attention Based Deep Layers

We fed the weighted word depicts into three-layer CNN networks and fed each tweet into a convolutional layer. Convolutional word vector matrix calculation in which a weighted F_n matrix is $w \epsilon R^{t \times m}$ used to represent local and inherent features. We select the word vector t from the F_n matrix.

$$h_i - f\left(V_{i:j+t-1} \times W[i] + b_i\right) \qquad (17)$$

where f is the nonlinearity activation function, W denites the matrix weight, b denotes bias,. Later it minimises dataset dimensions and important fetures are extracted as in Eq.(18)

$$p_i = Max[h_i] \qquad (18)$$

where the $p_i \epsilon R^{n-t+1/2}$

The feature map after max-pooling layers contains CNN layer features. While max-pooling helps extract important features, it does not focus on semantics and polarity importance. An attention mechanism emphasizes the value of each CNN-generated feature to overcome this constraint. This attention mechanism allows us to give greater weight to certain features based on their significance for the task at hand:

$$A_i = \frac{exp(p_i)}{\sum_i expp_i} \qquad (19)$$

The max-pooling attention equation is shown above, where A_i is the produced attention.

To further enhance the feature context, we apply a Bidirectional LSTM (Bi-LSTM) network to the output of the attention scores. This Bi-LSTM processes the feature map sequentially, generating final features. The CNN feature context pi and attention scores Ai determine the final feature map. To ensure both forward and backward features are taken into account, we utilize Bidirectional LSTM, which processes the features in parallel and concatenates the hidden states from both forward and backward LSTMs at each position. Equation (20) represents the forward LSTM, while equation (21) represents the backward LSTM.

$$\vec{h}_{t_{lstm}} = \overrightarrow{LSTM}(c_{t-1}, h_{t-1}, A_i) \qquad (20)$$

$$\overleftarrow{h}_{t_{lstm}} = \overleftarrow{LSTM}(c_{t-1}, h_{t-1}, A_i) \qquad (21)$$

The hidden states and memory cell are represented by h_{t-1} and c_{t-1}. However, reflect the LSTM function's prior states. LSTM networks use attention score A_i as input vectors. As stated in Equation (22) we concatenate the forward and backward context to annotate each input vector.

$$h_{t_{lstm}} = LSTM\left[\vec{h}_{t_{lstm}}, \overleftarrow{h}_{t_{lstm}}\right] \qquad (22)$$

$h_{t_{lstm}}$ denotes the concatenating output. The feature extracted is represented as $[\vec{h}_{t_{lstm}}, \overleftarrow{h}_{t_{lstm}}]$. By considering the forward and backward contexts simultaneously, the bidirectional network captures complete information. A fully connected dense layer converts this bidirectional information into a high-level sentiment representation and predicts text sentiment polarity.

$$h_i = Relu(w_i h_p + b_i) \qquad (23)$$

Using the Bi-LSTM network, w_i and b_i represent the learned parameters θ, h_i represents the acquired features, and h_p is the created feature map. The output layer is responsible for sentiment classification using the merged feature layer. For binary datasets, a sigmoid classifier is used, whereas for multiclass datasets, a Softmax classifier is employed. We use cross entropy as the loss function to measure the difference between the predicted and the original sentiment.

IV. RESULTS AND DISCUSSIONS

The experiments in this study were performed on a Windows 11, 64-bit system with an Intel Core i5 8th generation processor. The techniques were implemented in Python using Scikit-learn version 0.22.2 post1 and sklearn version 10.0.6 for machine learning and deep learning models. Validation metrics including accuracy, precision, recall, and F1 score were derived using equations (24)–(27) to evaluate classification performance on the datasets:

$$Accuracy = \frac{TN + TP}{TN + FP + TP + FN} \qquad (24)$$

$$Precision = \frac{TP}{TP + FP} \qquad (25)$$

$$Recall = \frac{TP}{TP + FN} \qquad (26)$$

$$F1\ score = 2 * \frac{Precision * Recall}{Precision + Recall} \qquad (27)$$

Averaging evaluation metrics can examine multiclass classification performance. This is often done with microaverage and macroaverage. We calculate micro-F1 and macro-F1 scores to evaluate experimental methods in our study. Equations (28) and (29) define macro-F1 and micro-F1.

$$macro - F1 = \frac{1}{m}\sum_{i=1}^{m} F1 \qquad (28)$$

$$micro = F1 = \frac{2 * \sum_{i=1}^{m} TP}{2 * \sum_{i=1}^{m} TP + \sum_{i=1}^{m} FP + \sum_{i=1}^{m} FN} \qquad (29)$$

Table 2 lists the Conv Bidirectional-LSTM model's hyperparameters with its uodated values.

Table 2: Hyperparameter setting.

Parameters	**Values**
Filter	128
Pool size	3
Dropout	0.3
Kernel size	5
Batch size	128
Bi-LSTM output size	512
Embedding dimensions	GloVe = 300
Optimizer	Adam
Learning rate	0.001
Loss function	Categorical_crossentropy

The KNN Classification method does not specify feature significance. This is because KNN needs to learn the significance of each feature directly. Instead, it predicts based on the distance between a new data point and its k closest neighbors. Figure 3 depicts the value of "bad customer service" and "seat pay" compared to other features.

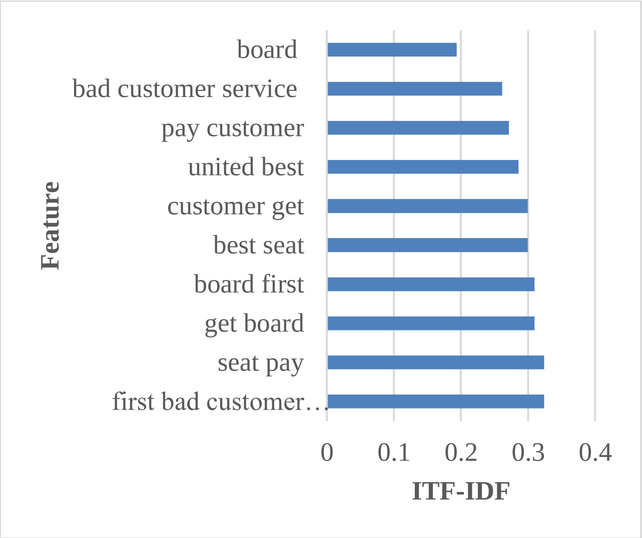
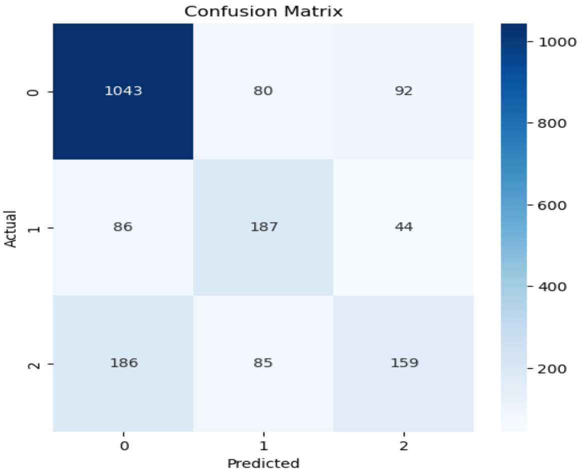
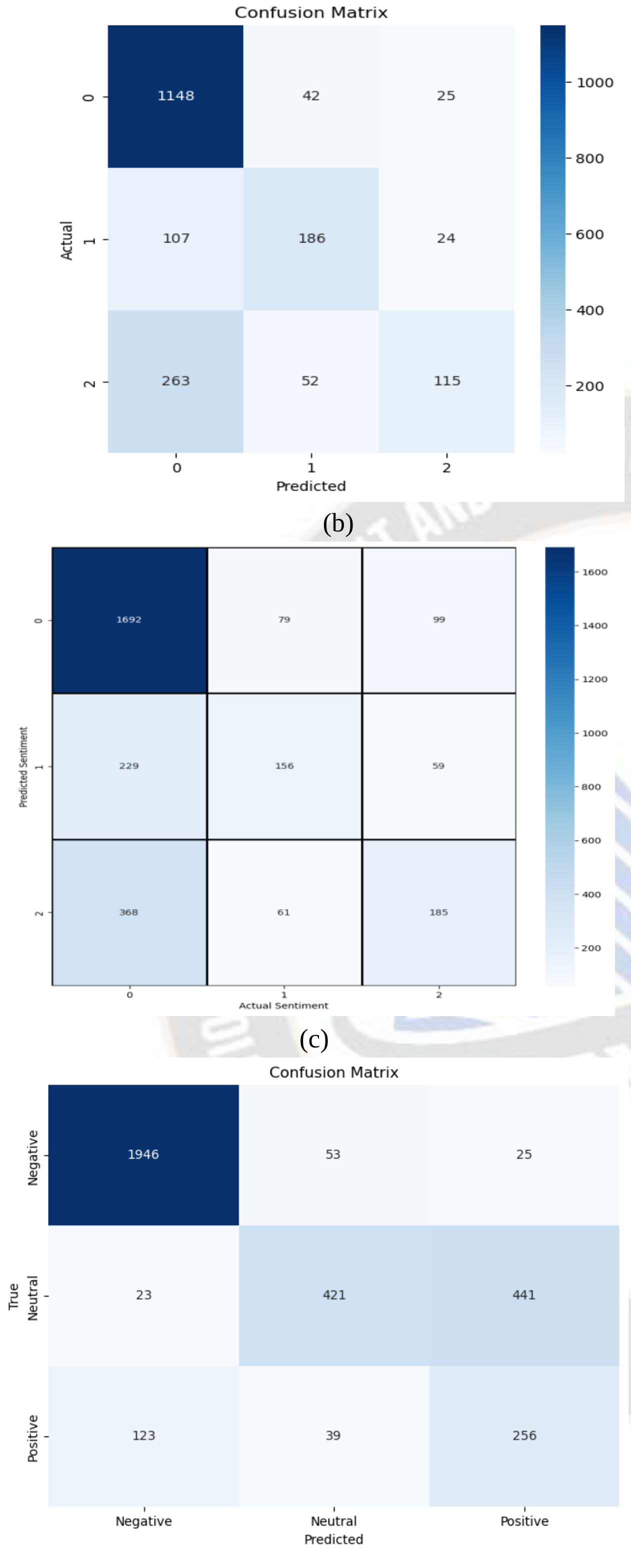


Figure 3: Displaying top ITF-IDF features

Figure 4 shows the confusion matrix of KNN in which IEG-ITFIDF outperforms other three vectorizations.



(a)



(b)

(c)

(d)

Figure 4: Confusion matrix of KNN (a) BOW vectorizer (b) IEG-ITFIDF Vectorization (c) Avg Word2Vec-KNN (d) TF-IDF Weighted Word2Vec

Table 3 lists BOW, W2V-AVG, W2V-TFIDF, and IEG-ITFIDF vectorization with K-nearest neighbor for computing Area under curve (AUC) score and F1-score.

Table 3: Different vectozations based K-nearest neighbour

Category	AUC Score	F1-Score
BOW	0.85	0.82
W2V-AVG	0.86	0.81
W2V-TFIDF	0.86	0.83
IEG-ITFIDF	**0.89**	**0.88**

Figure 5 shows the performance comparison of four vectorization models of KNN in which IEG-ITFIDF has the highest AUC score of 0.89 and F1-score of 0.88 related to the other three vectorizations.

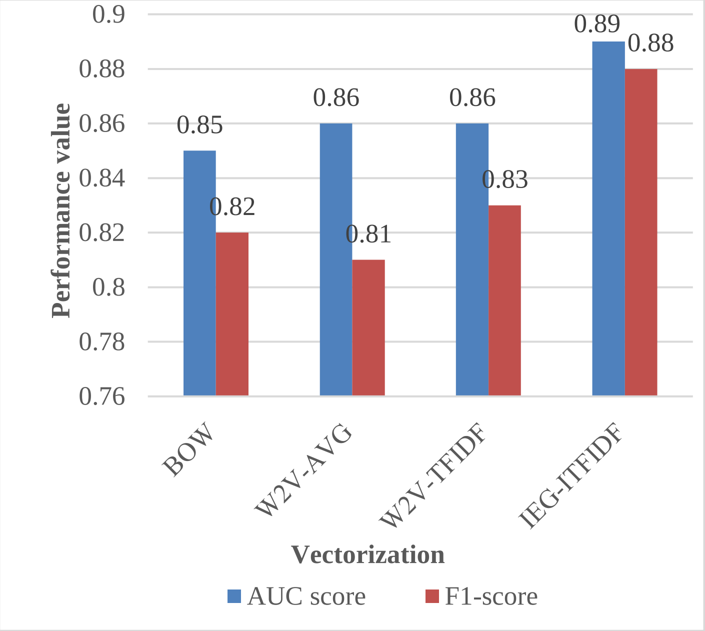
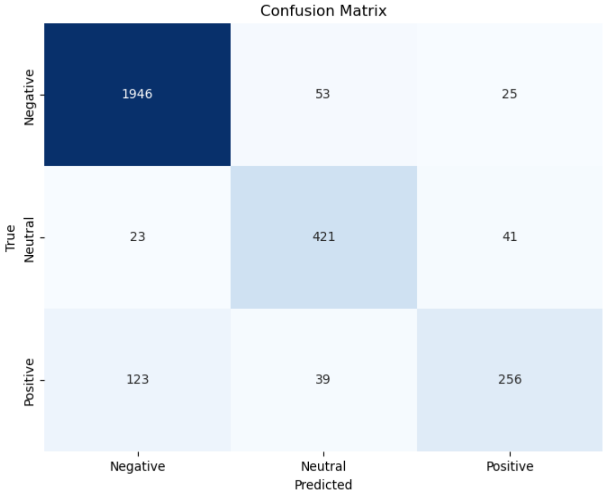


Figure 5: Performance comparison of four vectorization models of KNN

Figure 6 shows the confusion matrix of RF in which IEG-ITFIDF outperforms other three vectorizations.



Article Received: *05 July 2023* ***Revised:*** *25 August 2023* ***Accepted:*** *15 September 2023*

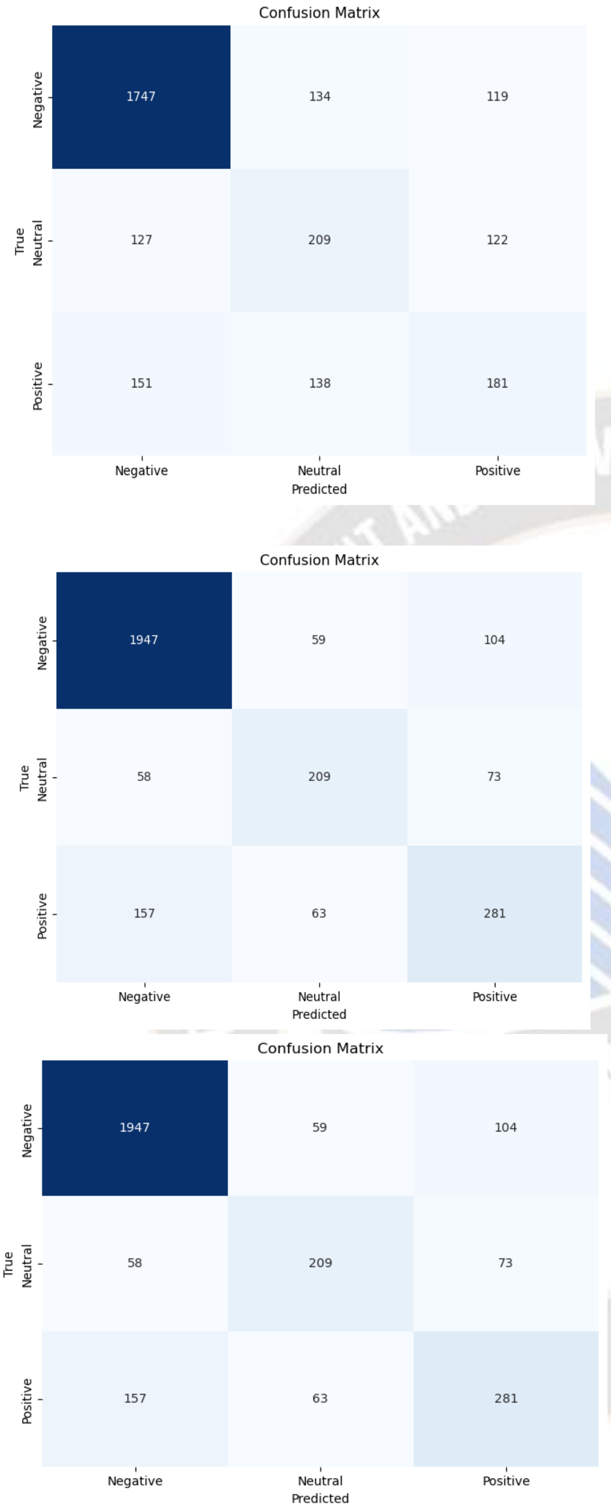


Figure 6: Confusion matrix of RF (a) BOW vectorizer (b) IEG-ITFIDF Vectorization (c) Avg Word2Vec-KNN (d) TF-IDF Weighted Word2Vec

Table 4 lists BOW, W2V-AVG, W2V-TFIDF, and IEG-ITFIDF vectorization with RF for computing Area under curve (AUC) score and F1-score.

Table .4: Different vectozations based Random forest

Category	**AUC Score**	**F1-Score**
BOW	0.92	0.70
W2V-AVG	0.88	0.66
W2V-TFIDF	0.88	0.65
IEG-ITFIDF	**0.92**	**0.71**

Figure 7 shows the performance comparison of four vectorization models of RF in which IEG-ITFIDF has the highest AUC score of 0.92 and F1-score of 0.71 related to the other three vectorizations.

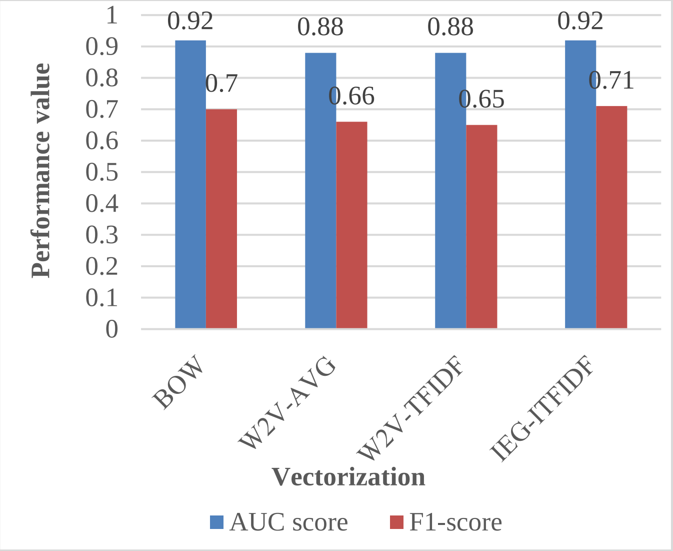
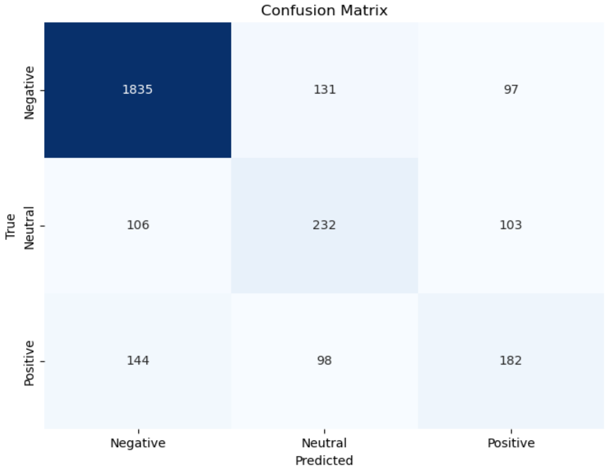
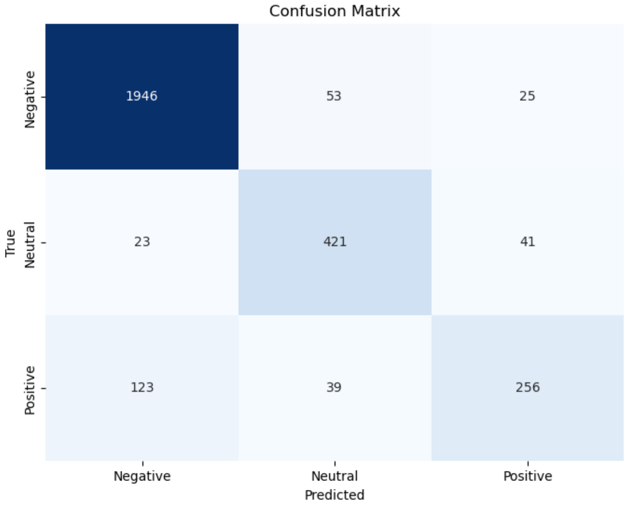


Figure 7: Performance comparison of four vectorization models of RF

Figure 8 shows the confusion matrix of RF in which IEG-ITFIDF outperforms other one vectorization.



(a)



(b)

Figure 8: Confusion Matrix of MNB (a) BOW vectorizer (b) IEG-ITFIDF Vectorization

Table 5 lists BOW and IEG-ITFIDF vectorization with MNB for computing Area under curve (AUC) score and F1-score.

Table 5: Different vectozations based MNB

Category	**AUC Score**	**F1-Score**
BOW	0.95	0.69
IEG-ITFIDF	**0.95**	**0.72**

Figure 9 shows the performance comparison of two vectorization models of MNB in which IEG-ITFIDF has the highest AUC score of 0.95 and F1-score of 0.72 related to the BOW vectorization.

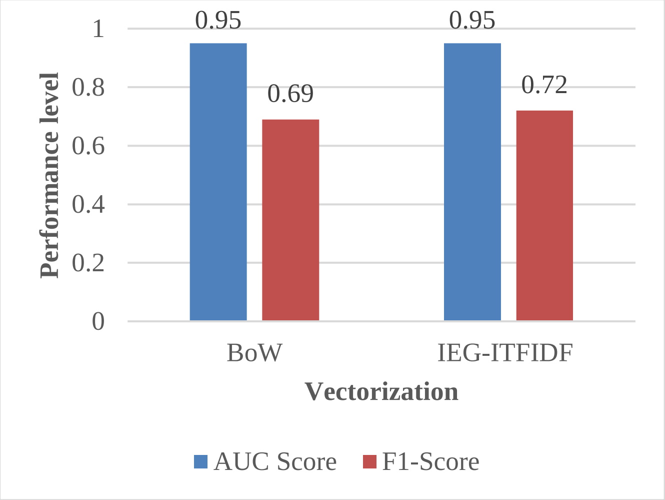


Figure .9: Performance comparison of four vectorization models of MNB

Figure 10 shows the confusion matrix of ANN with IEG-ITFIDF that correctly classified 1999 of negative, 451 of neutral and 287 of positive tweets respectively.

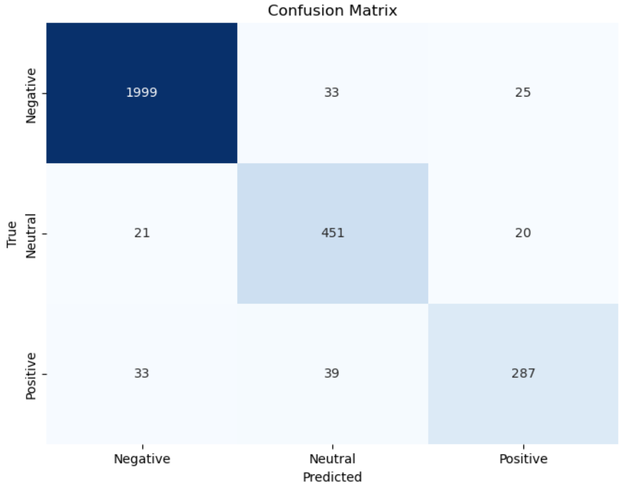
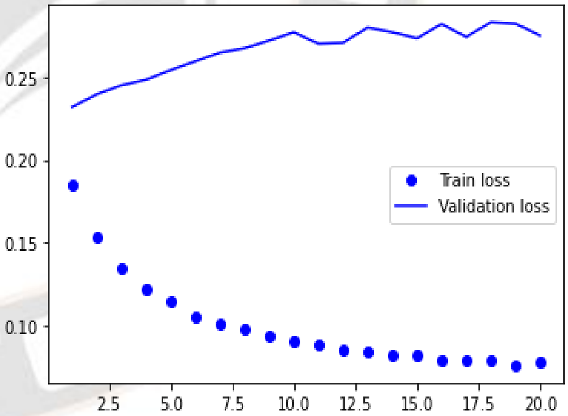
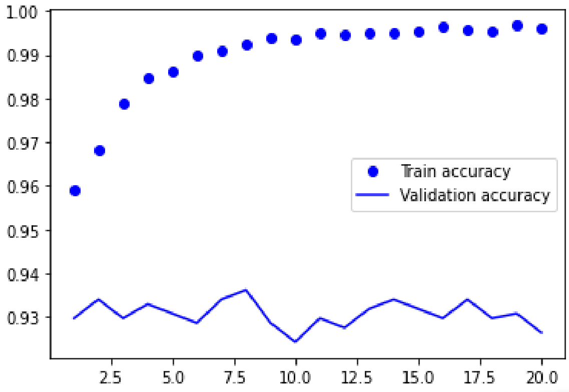


Figure 10: Confusion Matrix of Bi-LSTM

As illustrated in Figure 11 (b), the accuracy of the US airline training data begins at 0.958 and progressively increases to 0.995 at epoch 10, remaining constant for all following epochs. Furthermore, the validation data accuracy swings between 0.928 and 0.938 every epoch increment. Simultaneously, it remained stable after the 20th epoch.



(a)



(b)

Figure 11: Performance comparison of ANN model's (a) loss (b) accuracy

For simplicity, Table .6 summarize the proposed model's accuracy with the aforementioned models. It depicts that the proposed IEG-ITFIDF Vectorization- Bi-LSTM with Glove model successfully attained the highest accuracy of 94.26% with our structured US airline dataset.

Table .6: Summary of the proposed model's accuracy

Model	**Accuracy in %**
IEG-ITFIDF Vectorization-KNN	89%
IEG-ITFIDF Vectorization-RF	86%
IEG-ITFIDF Vectorization- MNB	90
IEG-ITFIDF Vectorization- ANN	92
IEG-ITFIDF Vectorization- LSTM	93
IEG-ITFIDF Vectorization- Bi-LSTM with Glove	**94.26%**

Overfitting can reduce deep learning models' accuracy and performance. Figure 12 shows the validation data's correctness, which starts at 0.814 and rises to 0.935 at epoch 10, then stays constant. However, training data accuracy ranges from 0.60 to 0.952 as epochs rise. Stabilises after 15th epoch. Figure 13 shows the loss value, indicating that our model prevented overfitting on the US airline Twitter dataset. As epochs increase, our model loss in both training and validation sets decreases and stabilises. In Figure 13, the validation data loss starts at 0.5106, drops to 0.1764 during 10 epochs, and then stabilises.

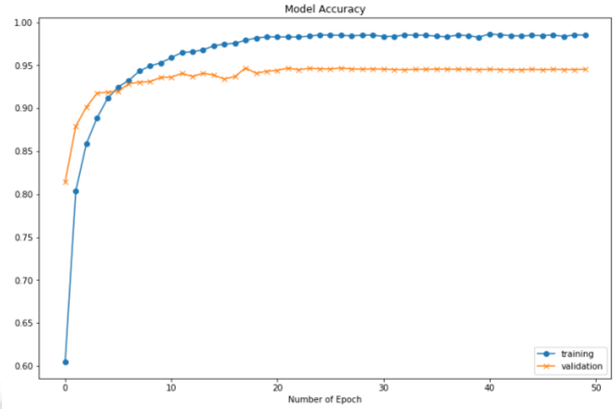


Figure 12: Performance comparison of model's accuracy

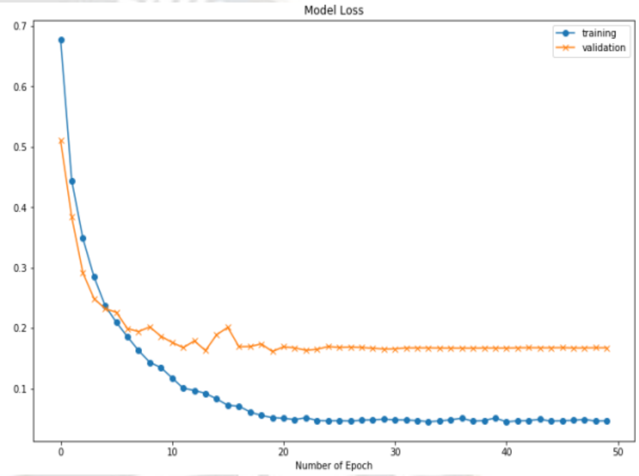


Figure 13: Loss rate change on training and validation data

V. CONCLUSION

This paper introduces an improved TF-IDF with novel phrase weighting strategies to handle sentiment analysis' imbalanced dataset distribution. We used Bag of Words, ITFIDF, and word2Vec vectorization methods to use textual information efficiently. We tested hyperparameter-tuned machine-learning algorithms. The Multinomial Naive Bayes Model achieved 89% accuracy and 0.95 AUC. The KNN and Random Forest Models also had 85%–87% accuracies and AUC ratings of 0.92. Our deep learning models started with Artificial Neural Networks with Dense Layers and two Dropout Layers. After hyperparameter tuning with Keras Tuner, we found that regularisation and dropout layers improved our predictions, leading to test and validation set accuracies of 92-93%. Embedding Layers, LSTMs, Dense, and Dropout Layers were added to improve accuracy. We methodically tuned hyperparameters for each layer, achieving 98% accuracy on the test set and 93% accuracy on the validation set in 4 cycles. We presented Embedding Layers, Bidirectional LSTM, Dense and Dropout Layers, and Keras Tuner hyperparameter tuning for each layer. This sophisticated model obtained 97.23% test set accuracy and 94.26% validation set accuracy in 4 iterations.

These results indicate our sentiment analysis method's efficacy and potential for practical applications.

REFERENCES

[1] Tubergen, Frank & Cinjee, Tobias & Menshikova, Anastasia & Veldkamp, Joran. (2021). Online activity of mosques and Muslims in the Netherlands: A study of Facebook, Instagram, YouTube and Twitter. PloS one. 16. e0254881. 10.1371/journal.pone.0254881.

[2] Singh, Neelam & Upreti, Manisha. (2022). HMRFLR: A Hybrid Model for Sentiment Analysis of Social Media Surveillance on Airlines. 10.21203/rs.3.rs-2012451/v1.

[3] Hasib, Khan & Habib, Md. Ahsan & Towhid, Nurul Akter & Showrov, Md. Imran Hossain. (2021). A Novel Deep Learning based Sentiment Analysis of Twitter Data for US Airline Service. 10.1109/ICICT4SD50815.2021.9396879.

[4] Nanath, Krishnadas & Joy, Geethu. (2021). Leveraging Twitter data to analyze the virality of Covid-19 tweets: a text mining approach. Behaviour & Information Technology. 42. 1-19. 10.1080/0144929X.2021.1941259.

[5] Kim, Annice & Miano, Thomas & Chew, Rob & Eggers, Matthew & Nonnemaker, James. (2017). Classification of Twitter Users Who Tweet About E-Cigarettes. JMIR Public Health and Surveillance. 3. e63. 10.2196/publichealth.8060.

[6] Saglam, Fatih & Sever, Hayri & Genç, Burkay. (2016). Developing Turkish sentiment lexicon for sentiment analysis using online news media. 1-5. 10.1109/AICCSA.2016.7945670.

[7] Alkan, Bilal & Karakus, Leyla & Direkci, Bekir. (2022). Generating a sentiment dictionary in R and dictionary-based sentiment analysis in Turkish texts. Digital Scholarship in the Humanities. 38. 10.1093/llc/fqac093.

[8] Almekhlafi, Muneer A.s & Salah, Saleh. (2023). Hybrid Filter-Genetic Feature Selection Method For Arabic Sentiment Analysis. Thamar University Journal of Natural & Applied Sciences. 8. 10.59167/tujnas.v8i1.1487.

[9] Kanakkahewa, K. & Mohotti, W. & Subashini, Shashikala. (2023). PoS tag-based Attention for Feature Selection in Sentiment Analysis. 10.21203/rs.3.rs-3151544/v1.

[10] Parlar, Tuba & Özel, Selma. (2016). A new feature selection method for sentiment analysis of Turkish reviews. 1-6. 10.1109/INISTA.2016.7571833.

[11] Fattah MA (2017) A novel statistical feature selection approach for text categorization. J Inf Process Syst 13:1397–1409. https://doi.org/10.3745/JIPS.02.0076

[12] Mackie, Iain & Chatterjee, Shubham & Dalton, Jeffrey. (2023). Generative Relevance Feedback with Large Language Models.

[13] Saputra, Rio & Khoirudin, Khoirudin. (2022). Sistem Informasi Perpustakaan Menggunakan Metode Rocchio Relevance Feedback Berbasis Web. Information Science and Library. 3. 89. 10.26623/jisl.v3i2.5987.

[14] Ravi Kumar, G. & Kongara, Venkata Sheshanna & Babu G, Anjan. (2021). Sentiment Analysis for Airline Tweets Utilizing Machine Learning Techniques. 10.1007/978-3-030-49795-8_75.

[15] Et.al, M.Veera. (2021). Collaborative Classification Approach for Airline Tweets Using Sentiment Analysis. Turkish Journal of Computer and Mathematics Education (TURCOMAT). 12. 3597-3603. 10.17762/turcomat.v12i3.1639.

[16] Rustam, Furqan & Ashraf, Imran & Id, † & Mehmood, Arif & Ullah, Dr. Saleem & Choi, Gyu Sang & Khan, Yar. (2019). Tweets Classification on the Base of Sentiments for US Airline Companies. Entropy. 21. 10.3390/e21111078.

[17] Kumar, Akshi & Jaiswal, Arunima & Garg, Shikhar & Verma, Shobhit & Kumar, Siddhant. (2019). Sentiment Analysis Using Cuckoo Search for Optimized Feature Selection on Kaggle Tweets. International Journal of Information Retrieval Research. 9. 1-15. 10.4018/IJIRR.2019010101.

[18] Singh, Neelam & Upreti, Manisha. (2022). HMRFLR: A Hybrid Model for Sentiment Analysis of Social Media Surveillance on Airlines. 10.21203/rs.3.rs-2012451/v1.

[19] Hasib, Khan & Habib, Md. Ahsan & Towhid, Nurul Akter & Showrov, Md. Imran Hossain. (2021). A Novel Deep Learning based Sentiment Analysis of Twitter Data for US Airline Service. 10.1109/ICICT4SD50815.2021.9396879.

[20] M. Gupta, R. Kumar, H. Walia, and G. Kaur, "Airlines based twitter sentiment analysis using deep learning," in Proceedings of the 2021 5th International Conference on Information Systems and Computer Networks (ISCON), October 2021.

[21] Guyon I, Elisseeff A (2003) An introduction to variable and feature selection. J Mach Learn Res 3:1157–1182. https:// doi.org/10.1016/j.aca.2011.07.027.

[22] Harman D (1992) Relevance feedback revisited. In: Proceedings of the 15th annual international ACM SIGIR conference on Research and development in information retrieval—SIGIR'92. ACM Press, New York, pp 1–10

[23] Nguyen, H.T.; Nguyen, M.L. An ensemble method with sentiment features and clustering support. Neurocomputing 2019, 370, 155–165