Research Article

SOHAG JOURNAL OF SCIENCES

https://doi.org/ 10.21608/sjsci.2025.391898.1302

Decoding Lexical and Semantic Text Complexity: Evaluating Machine and Deep Learning Models for Document Classification across Different Categories

Hameda A. Sennary

Department of Mathematics and Computer Science, Faculty of Science, Aswan University, Aswan, Egypt
**Email: hasinary@sci.aswu.edu.eg*

Received: 2nd Augest 2025 **Revised:** 16th October 2025 **Accepted:** 22nd October 2025
Published online: 24th November 2025

Abstract: Rapid advancements in natural language processing (NLP) have made it possible for robots to comprehend and classify text data more effectively. The efficiency of many machine and deep learning models in automatically categorizing texts into five groups—business, politics, sports, technology, and entertainment—is examined in this study. We used preprocessing approaches to adjust a publically available dataset and compared models such as Long Short-Term Memory (LSTM), Convolutional Neural Networks (CNN), Multilayer Perceptrons (MLP), Support Vector Machines (SVM), Logistic Regression (LR), Multinomial Naïve Bayes (NB), Deep Neural Networks (DNN), K-Nearest Neighbors (KNN), Decision Trees (DT), Random Forests (RF), and Gradient Boosting Classifier (GBC). Our findings demonstrate the powerful prediction power of ensemble methods, as SVM and MLP both achieved high accuracy rates of 97%. Future research on automating text categorization, tackling issues like data noise and complexity, and optimizing models for particular applications would benefit greatly from the findings. This study emphasizes how crucial it is to choose and adjust models in order to improve performance in NLP applications.
Keywords: Text Document Classification, Convolutional Neural Networks, Deep Neural Networks, Machine Learning Algorithms, NLP.

1. Introduction

NLP began in the 1950s with early work in linguistic theory and Machine Translation, relying on rule-based and symbolic methods focused on grammar and syntax [1]. In the 1980s and 1990s, statistical techniques using probabilistic models and large text corpora marked a paradigm shift, significantly improving language understanding. The rise of Machine Learning in the early 2000s introduced models like BERT and GPT-3, along with advanced Deep Learning architectures that dominate today [2-6]. These advancements greatly enhanced automatic text summarization, although challenges persist in understanding complex contexts and addressing data biases [7-11]. Current research in Text Mining focuses on extracting insights from vast databases, including subfields like Information Retrieval (IR), Recommendation Systems, and NLP [12]. As online and social media content grows, Document Classification has become a key research area. Platforms like Facebook analyze messages and interactions to infer user preferences, while Google refines its search algorithms to align results with human intent. YouTube similarly analyzes user comments to filter or recommend videos.

Text Classification, a part of NLP, allows machines to categorize text systematically. It combines elements from Artificial Intelligence, Machine Learning, Computer Engineering, and Information Engineering to enhance language understanding. Also known as Document Categorization, it organizes texts into predefined categories based on their content, drawing from Computer Science, Information Science, and Library Science. While Library Science deals with intellectual classification, Computer Science and Engineering focus on automated methods. Texts with multimedia components, such as photos or stickers, require distinct classification models [13].

Research on Real-Time Text Mining is exploring applications like Relationship Extraction [14], Spam Detection [15, 16], Document Retrieval, Ontology Mapping, Email Classification, Directory Management, Routing, Filtering, and Sentiment Analysis [17], among others [18–20]. These applications can process vast amounts of text data, including the unstructured information on the World Wide Web, such as conference materials, publications, and news, which requires automatic processing due to diverse formats. To improve WWW text organization, advanced learning agents are needed to classify relevant content. With the explosion of electronic documents in the Digital Era, vast data is available but requires efficient Document Organization, which Data Mining, an AI subfield, can facilitate [21]. In large datasets, Association Rule Mining [22] helps uncover valuable relationships for decision-making. The Naïve Bayes Classifier assumes conditional independence of words given the class and uses Maximum A Posteriori estimates but often needs large training datasets to perform well [23, 24]. Alternatively, Genetic Algorithms generate rules randomly and evaluate their fitness. Recurrent Neural Networks (RNNs) and advanced variants like Long Short-

Term Memory (LSTM) are effective for encoding or generating short text sequences, usually limited to a few sentences.

2. Related work

In automatic text classification, terms are considered the most effective units for representing and categorizing text [25]. Since text documents lack the structured format of databases, unstructured data, particularly large free-form text, must be converted into a structured format. Several preprocessing techniques have been proposed to aid in this conversion [26, 27]. After structuring the data, an efficient document representation model is necessary for classification. The Bag of Words (BoW) model, a common approach, represents documents as vectors based on term frequency, but it doesn't preserve the relationships and context of terms, which is crucial for understanding the document's content [28]. Jain and Li [29] introduced a binary representation to address this, but it still faces issues with sparse matrices and high dimensionality. Hotho et al. [30] proposed an ontology-based representation to retain semantic relationships, though constructing ontologies automatically remains challenging. Cavanar (1994) [31] suggested using N-Grams, but determining the optimal length of N-Grams is difficult. Another method [32] uses multi-word terms as vector components, but it requires advanced term extraction algorithms. Lastly, Wei et al. (2008) proposed Latent Semantic Indexing (LSI) as an alternative for improved representation[33].

Document representation is essential for preserving key characteristics. (LSI) focuses on maintaining important features, unlike methods that prioritize discriminative features. To address LSI's limitations, particularly its neglect of local semantic structures, Locality Preserving Indexing was developed [34]. While LPI effectively captures these local structures, it has time and memory efficiency issues [35]. Choudhary and Bhattacharyya (2002) improved document representation by using the Universal Networking Language, converting documents into graph structures with words as nodes and relationships as edges [36]. However, this method is labor-intensive, especially with large document collections, as each document requires its own graph. Once a solid representation is created, classifying documents into predefined categories follows. Several models, including (NB) model [37, 38], (KNN) [39, 40], Centroid Classifier [41], (DT) [42, 43], Rocchio classifier [44], Neural Networks [45], and (SVM) [46], have been developed for this purpose.

Deep learning models have become integral to NLP, particularly in language modeling and text classification. Systematic comparisons of models like Recurrent Neural Networks (RNN), Deep Belief Networks (DBN), and (CNN) have been conducted, such as Liu et al.'s study [47] on their relation to relation classification. Techniques like Naive Bayes (NB) and Support Vector Machines (SVM) use rule-based features [48], and hybrid methods combine SVM, NB, and Conditional Random Fields for dependency tree building [49]. CNNs are used to extract relevant n-grams and capture long-term dependencies, aiding in tasks like time series forecasting [50]. DBNs can extract patterns from high-dimensional feature spaces [51], while RNNs are key in language modeling [52] and forecasting tasks [53]. Chen et al. [54] showed DBNs combined with SVMs improve Chinese text classification, similar to CNNs' role in semantic labeling [55]. RNNs have also proven effective in long-term sentence classification [56, 57]. The Network In Network (NIN) model [58] improves CNNs by using global average-pooling layers and 1x1 convolutional filters. Combining CNNs with (LSTM) networks enhances attention-based tasks, while comparisons of CNN, word2vec [59], GRU, and LSTM for sentiment classification in Russian tweets indicate that GRUs often outperform CNNs and LSTMs [60]. Studies [61, 62] show no clear winner between GRU and LSTM, with hyperparameter tuning being more important than model architecture selection.

The DBWorld Email dataset was divided into "announces of conferences" and "everything else" classes. The performance of (DT), (SVM), (KNN), and (NB) was compared using five-fold cross-validation. Results showed that SVM, with a quadratic kernel, had the highest accuracy (93.8%), despite DT having the longest training time. While SVM was the best performer, other methods also showed promise, indicating potential for convolutional neural networks in future text classification [63]. The authors in [64] introduced W2vRule, a text classification method combining machine learning with the Word2Vec framework. Tested on the Reuters Newswire dataset, W2vRule outperformed traditional methods like NB and DT in terms of accuracy, precision, and recall, especially when hyperparameter tuning was applied, proving it effective for large-scale document classification. Amina Khatun et al. (NB), (DT), (SVM), and (KNN) for classifying SMS messages as "spam" or "ham." After text preprocessing and feature extraction, NB and SVM achieved the highest accuracy rates, 98.66% and 98.02%, respectively[65].

Even though there have been many prior studies on text categorization using machine learning and deep learning models, the most of them have only used short datasets or a small number of categories, demonstrating a great deal of variation in model performance. Additionally, these research have shown that the accuracy of findings is significantly impacted by the choice of classification model and the fine-tuning of its parameters, particularly when dealing with multi-class texts and small semantic variations between words. Therefore, it is crucial to perform a thorough assessment of a wide range of deep and traditional models on a well-balanced dataset that spans several categories in order to determine which models are the most dependable and successful as well as to comprehend the advantages and disadvantages of each in the context of complex text classification. This method makes it possible to focus research efforts on enhancing automated categorization systems' functionality and using them more precisely and successfully in real-world applications.

3. Proposed Methodology

This section describes the suggested approach, illustrating both the proposed strategy and the progression of the machine learning methodology. Three main categories are used to summarize the suggested methodology:

3.1. Data Collection and Splitting:

The proposed framework is trained and evaluated on a publicly available dataset; the text document classification dataset contains 2,225 text samples across five categories of documents. These categories are politics, sports, technology, entertainment, and business, as shown in Fig 1. This dataset can be used for document classification and clustering. It contains two features: text and label. The dataset has 2,225 rows and 2 columns. The text feature includes different categories of text data, while the label feature contains labels for the five categories: 0 (Politics), 1 (Sports), 2 (Technology), 3 (Entertainment), and 4 (Business). Using the 80-20 ratio, the preprocessed data was split into 80% training data and 20% test data. To learn classification rules, the training data is fed into the classification model. Furthermore, using the test data, the classification model is evaluated further [66].

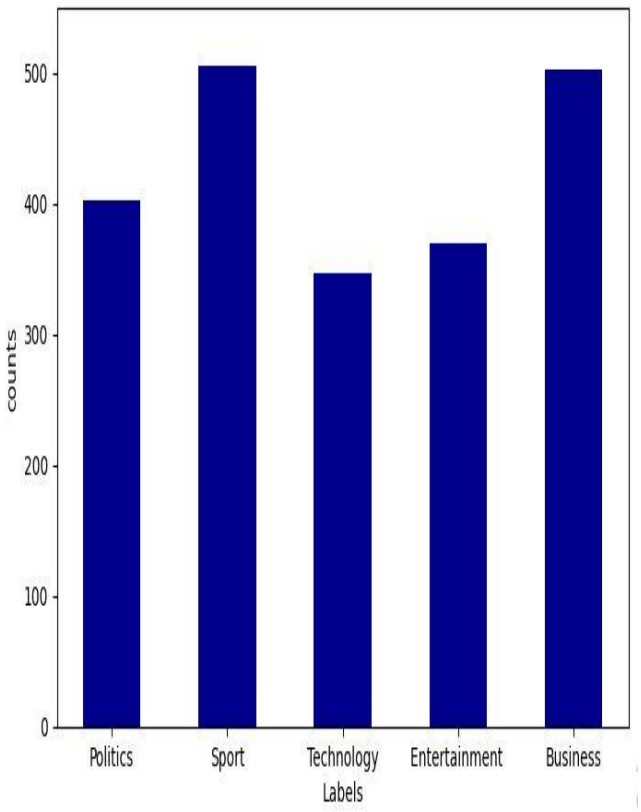


Figure 1: Show the distribution of five categories

3.2. Data Preprocessing

To achieve uniformity, the input text is first converted to lowercase in the preprocessing function for text document classification. This enables the model to treat words with different cases (such as "Text" vs. "text") as equivalent. Then, to get rid of unimportant material that doesn't add to the text's main point, mentions—usually user handles preceded by the "@" symbol—are eliminated. Subsequently eliminates URLs, including complete web addresses that start with "http" and those that start with "www," as they frequently don't have semantic weight for classification. The text is then made simpler for analysis by removing all non-alphanumeric characters, leaving a clean dataset made up solely of letters and integers. This set of transformations helps to prepare the text for effective classification by reducing noise and standardizing the input. Finally, the function combines any multiple spaces into a single space to ensure consistent spacing throughout the text, avoiding issues that may arise from irregular spacing [67].

3.3. Machine and Deep Learning Approach

For each term in the training dataset, first determine its Term Frequency-Inverse Document Frequency (TF-IDF) scores, accounting for both the term's rarity throughout the total document set and its frequency within individual documents. Then, to enable quantitative comparisons across documents in the training set, describe each document as a numerical vector, with the components of the vector being the TF-IDF values assigned to each word. After that, train every model using these TF-IDF-weighted vectors. Lastly, apply the training models to their respective TF-IDF-based representations to produce predictions for the documents in the testing set. Then, evaluate each model's accuracy based on these predictions.

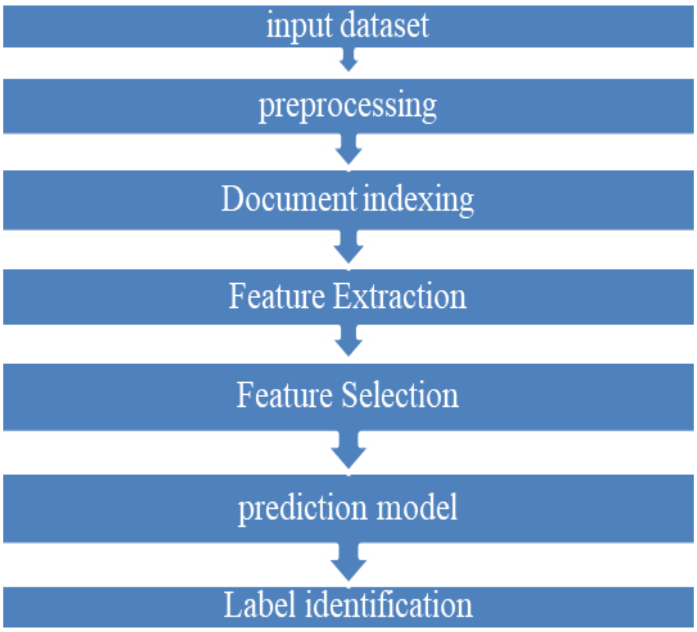
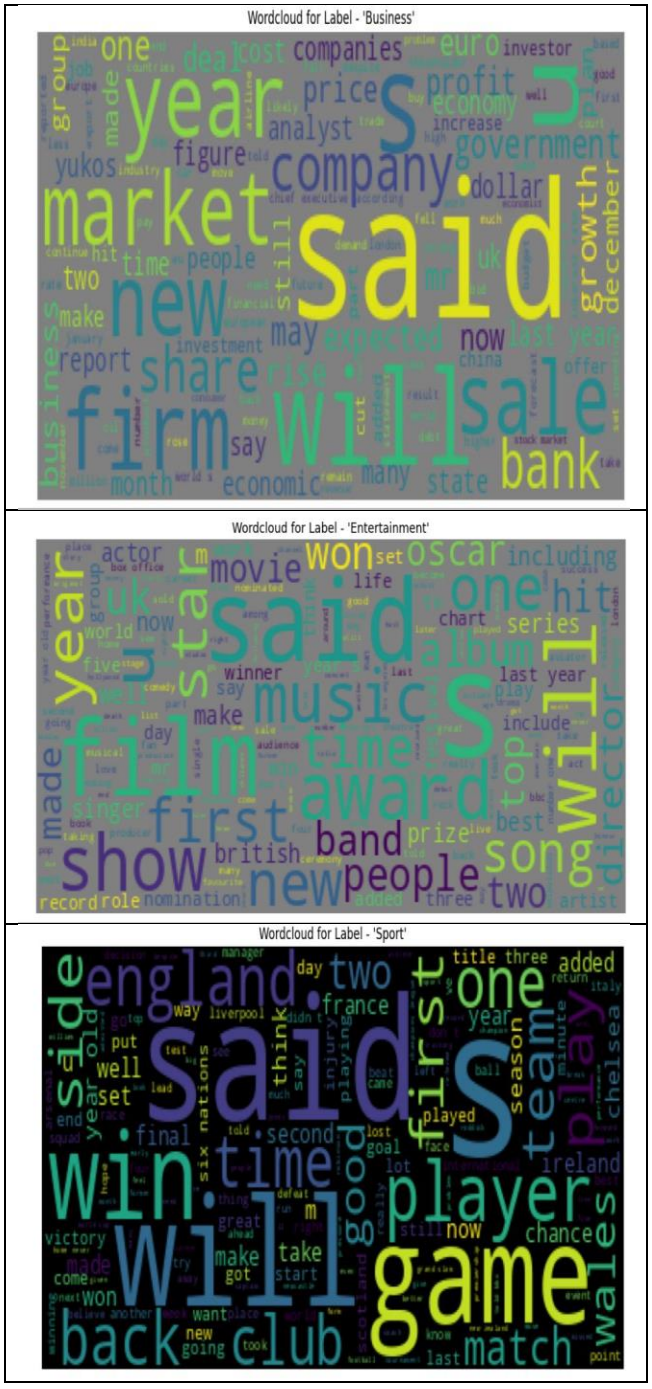


Figure 2: Workflow diagram depicting the essential stages in the document classification process: Input Dataset, Preprocessing, Document Indexing, Feature Extraction, Prediction Model, and Label Identification.

4. Experimental results

An essential tool for classification problems, a word cloud is a visual representation of text data in which the size of each word indicates its frequency or importance within a particular dataset. Each of the five categories—business, sports, politics, technology, and entertainment—can be represented by a different word cloud as shown in Figure 3, when it comes to document classification. For instance, the sports word cloud would emphasize phrases like "team," "game," and "score," while the business word cloud might prominently display words like "finance," "investment," and "market." Likewise,

the political cloud may display words like "election," "policy," and "government," while the technology cloud may display words like "software," "innovation," and "artificial intelligence." Lastly, terms like "movie," "celebrity," and "music" can be the main focus of the entertainment word cloud. By finding patterns and improving document categorization accuracy, the analysis of these word clouds helps train classification algorithms by revealing the most common themes and keywords within each class. Developers can produce more efficient models that differentiate between the distinct textual details of each category by utilizing the traits displayed in these visual representations.



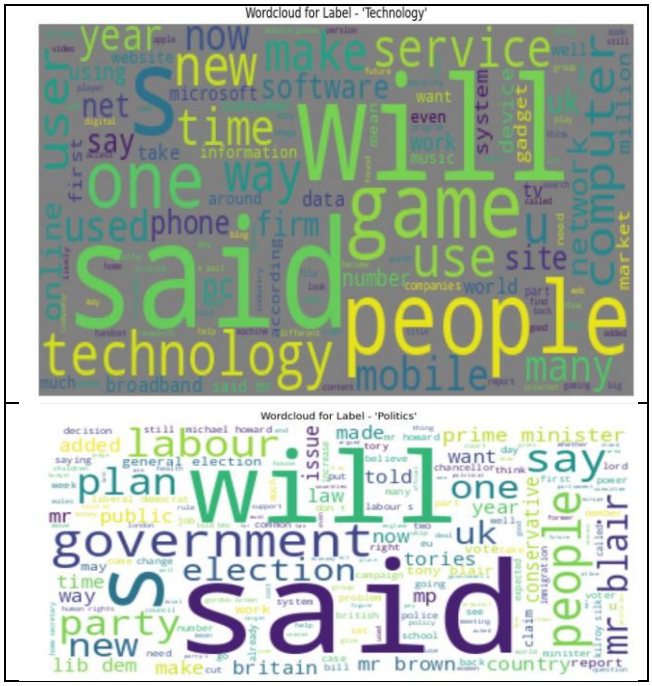


Figure 3: show the Word clouds for document classification categories : The most common terms for eachcategory,business, sports, politics, technology, and entertainment—are represented by word clouds, which aid in identifying important themes and enhancing the accuracy of document classification. By emphasizing specific terms that are pertinent to each category—for example, "team" in sports or "finance" in business—these visuals help train computers.

The distribution of the data is shown graphically in the accompanying figure 4, where the text length is represented by the x-axis and the frequency or count of text lengths inside each bin is represented by the y-axis. Data exploration, visualization, and possible feature engineering chores are made easier by this output, which has great value since it enables data scientists or analysts to rapidly spot trends or skewness in the data, such as whether the majority of text lengths are uniformly distributed, long, or short.

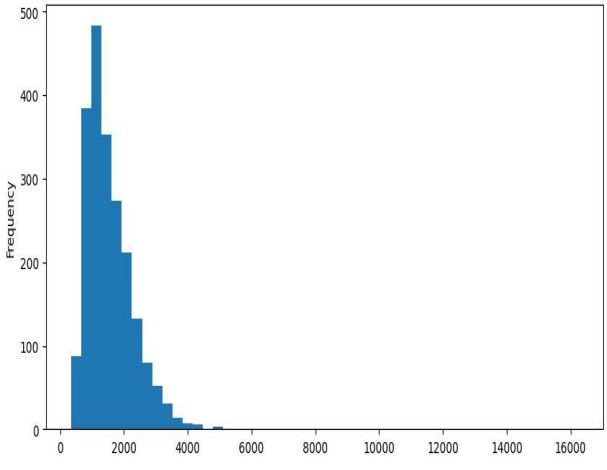


Figure 4: the distribution of text lengths in the dataset.

The ten most common bigrams from the cleaned text of tweets saved in a data frame are displayed visually in Fig. 4. It successfully illustrates the linguistic patterns and themes present in the tweet dataset by charting these bigrams against their corresponding counts in a horizontal bar format. This visualization is essential because it makes it possible to quickly identify frequently used terms and offers insights into popular subjects, public opinion, and important concerns that users are discussing. Better analysis and comprehension of the textual data at a glance are made possible by the use of a predetermined figure size, which guarantees that the output is readable and clear.

Figure 5: depicts the 10 most prevalent bigrams derived from the sanitized tweet dataset. The horizontal bar chart illustrates commonly utilized paired phrases, like "last year," "told BBC," "said Mr.," and "prime minister," offering insights into dominant issues, public debate, and significant themes within the dataset. The lucid and comprehensible format enables rapid recognition of prevailing linguistic trends and domains of public concern.

The 30 most common words in a set of text data are displayed in a bar chart in Fig 6. With the words on the x-axis and their frequency on the y-axis, the chart illustrates the frequency of each word. The term with the highest frequency is on the far left of the chart, while the word with the lowest frequency is on the far right, since the chart is arranged in descending order of frequency. Blue bars of various heights indicate the terms' frequencies, which are listed vertically. Understanding the topic, sentiment, or style of the text can be aided by the chart's visual representation of the most frequently used terms in the text data.

Figure 6 shows the 30 most frequent words in the text collection in a bar chart, it arranged by decreasing frequency. Each blue bar's height reflects the frequency of the term, giving readers a visual summary of the most common words to help them grasp the text's general theme, sentiment, or style.

Figure 7 analyzes the character length of various text categories, including politics, sports, entertainment, technology, and business. The research finds differences in text length by counting the characters in text samples from each category. This can show patterns in audience interest and content engagement across various domains. Comprehending these distinctions is essential since it allows marketers and content producers to efficiently customize their messaging tactics. While longer entries in the technology category would signal a more engaged audience seeking in-depth knowledge, a lesser character count in the sports area might suggest a preference for quick updates. In the end, this type of study aids in improving communication effectiveness within each category and optimizing the delivery of material.

The top figure in Figure 8, shows the loss values in the y-axis, while the x-axis shows the number of epochs in a graph showing loss during text categorization in (CNN) model. The training data curve and the validation data curve are two separate curves on the graph. These curves initially converge at the same spot, showing a comparable loss trajectory as the model gains knowledge from the training set. Both curves decline as epochs go on, suggesting better performance; nevertheless, there is a significant divergence when the validation curve plateaus horizontally and the training curve keeps declining dramatically. This shape's benefit is that it shows learning: the first drop indicates successful learning. In the bottom figure, shows the accuracy values in the y-axis, and the x-axis shows the number of epochs in the graph showing text classification accuracy. The training data curve and the validation data curve are two separate curves on the graph. Both curves initially begin at the same location, indicating a comparable degree of precision. The validation accuracy curve, on the other hand, trails behind the training accuracy curve, which increases noticeably as training goes on. The model's ability to efficiently learn patterns in the training dataset is demonstrated by the steep climb of the training curve, which gives it an advantage when it comes to obtaining high accuracy on known data.

The y-axis shows the loss value, and the x-axis shows the number of epochs in the graph that shows loss for text classification in (DNN) model in Figure 9. At the beginning of the training process, the training and validation loss curves begin at the same place, suggesting that the loss scores of the two datasets are comparable. Both curves have a lower trend as the epochs go on, indicating gains in model performance. The model appears to be successfully learning the training data, though, as the training loss curve declines more abruptly than the validation loss curve. Although the steep decline of the training curve is advantageous because it shows that the model can learn complex patterns, the validation curve's continued height draws attention to the model's possible overfitting, in which it performs well on training data but less well on unseen data. The need for close observation during training to make sure the model generalizes properly and does not become unduly specialized to the training dataset is highlighted by this divergence between the two curves. As a result, the validation curve is an essential measure of the model's resilience and generalization skills, whereas the training curve's drop indicates successful learning. On the other side, the y-axis shows the accuracy values, and the x-axis shows the number of epochs in the graph showing text classification accuracy in (DNN) model. The training and validation accuracy curves initially begin at the same location, indicating an initial performance level that is equal. The model is learning effectively from the training data when the training accuracy curve increases over the course of the epochs and surpasses the validation accuracy curve. This situation has the benefit of boosting training accuracy, which indicates that the model is successfully identifying patterns in the training dataset.

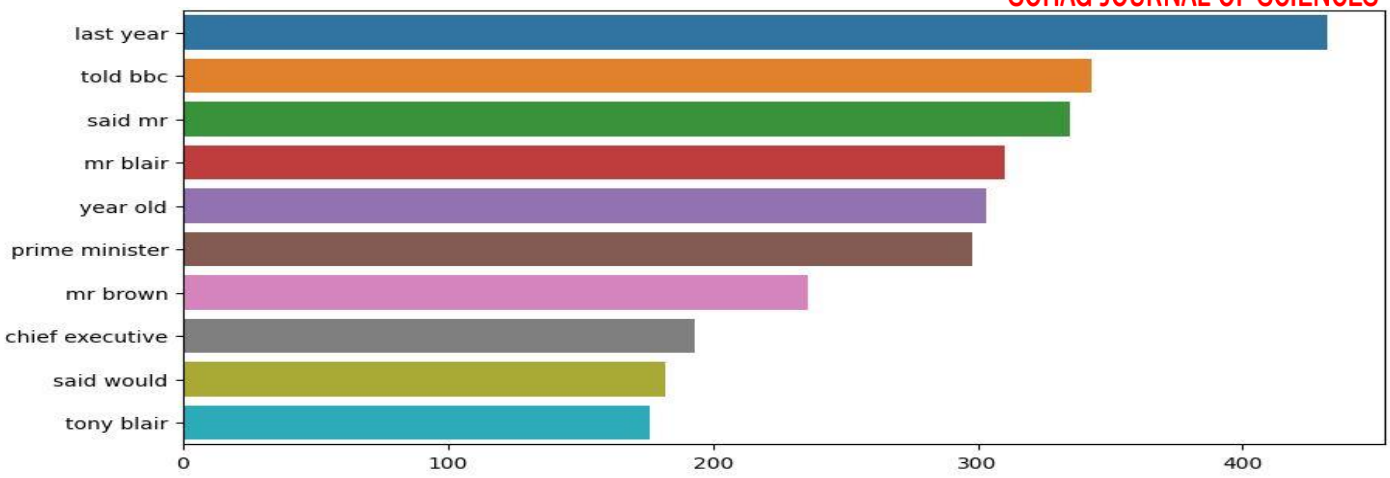


Figure 5: depicts the 10 most prevalent bigrams derived from the sanitized tweet dataset

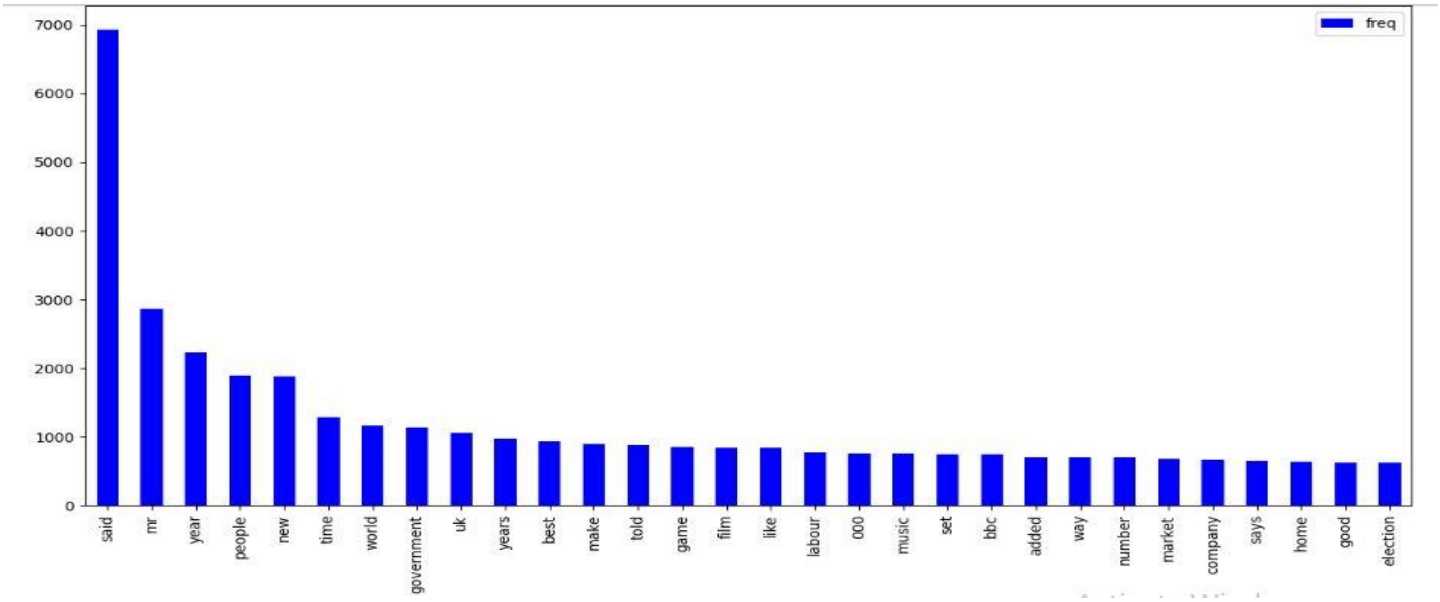


Figure 6: The 30 most frequent words in the text collection are shown in a bar chart in this figure

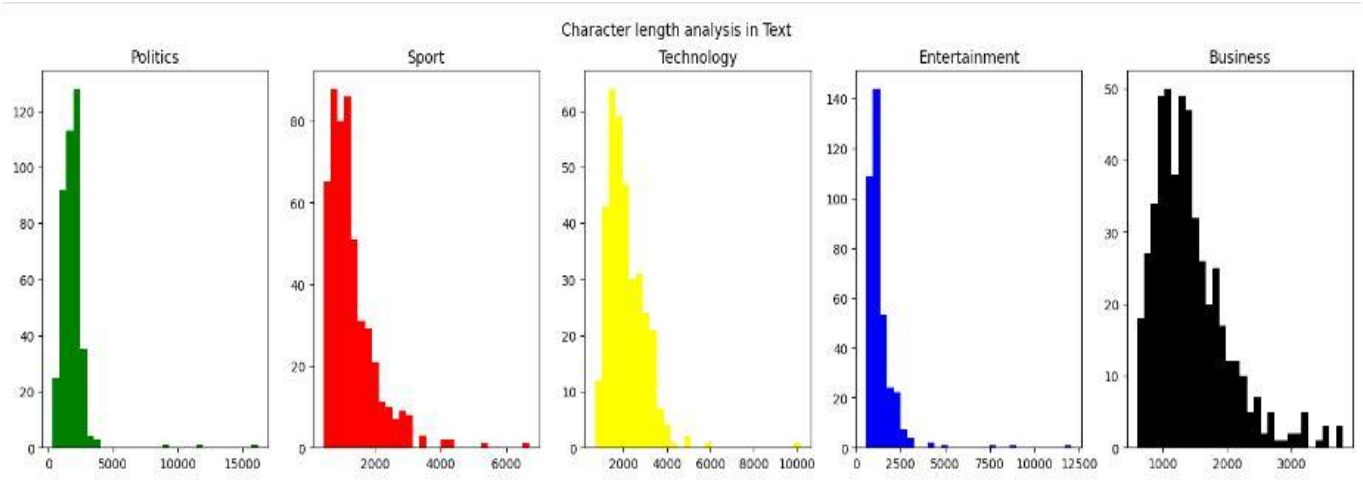


Figure 7: show the audience engagement and content preferences, a comparative analysis of character length across various text categorie

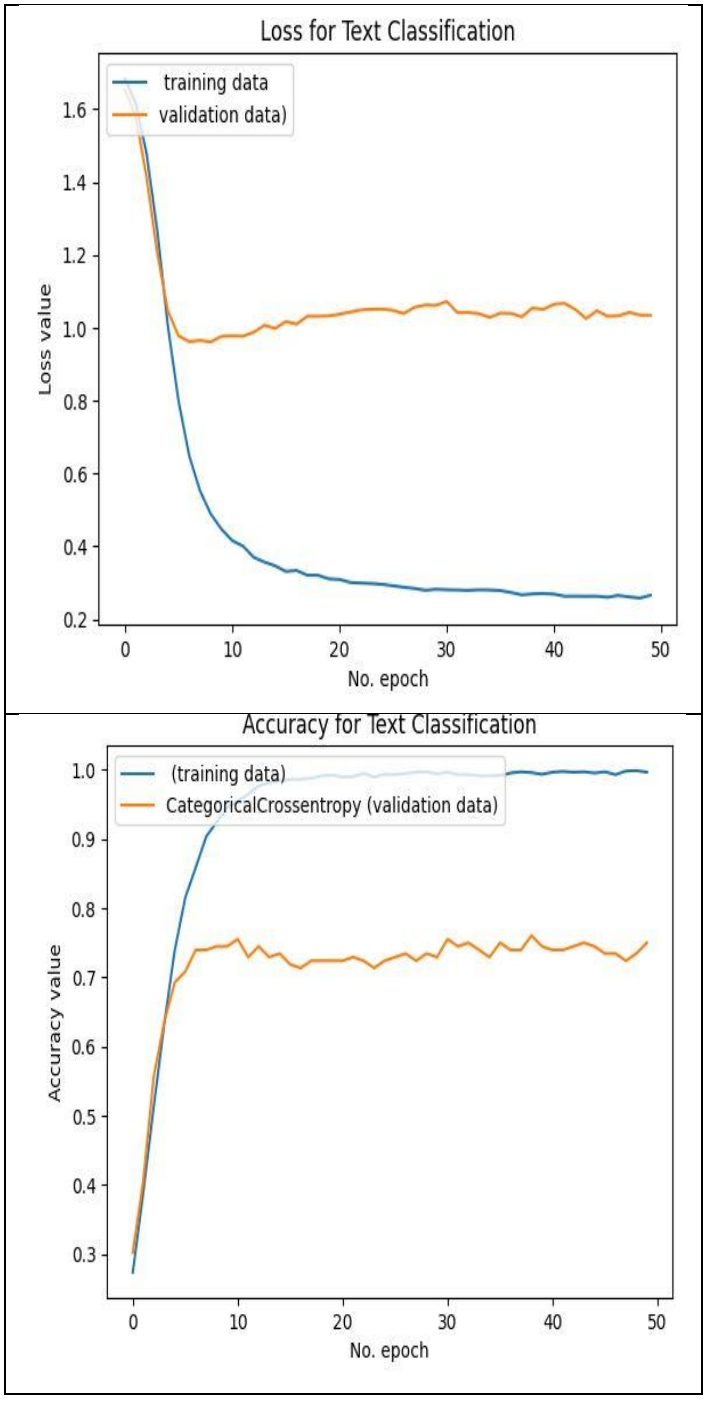


Figure 8: The first graph on the left shows the loss values on the y-axis against epochs on the x-axis.

Both curves initially converge, suggesting comparable learning progress, but as training goes on, they diverge, with the training loss dramatically declining and the validation loss plateauing, suggesting possible overfitting. The accuracy over epochs is shown in the other graph on the right, where the validation accuracy trails behind, demonstrating the model's poor generalization to unknown data, while the training accuracy rises steadily, demonstrating the model's enhanced performance on training data.

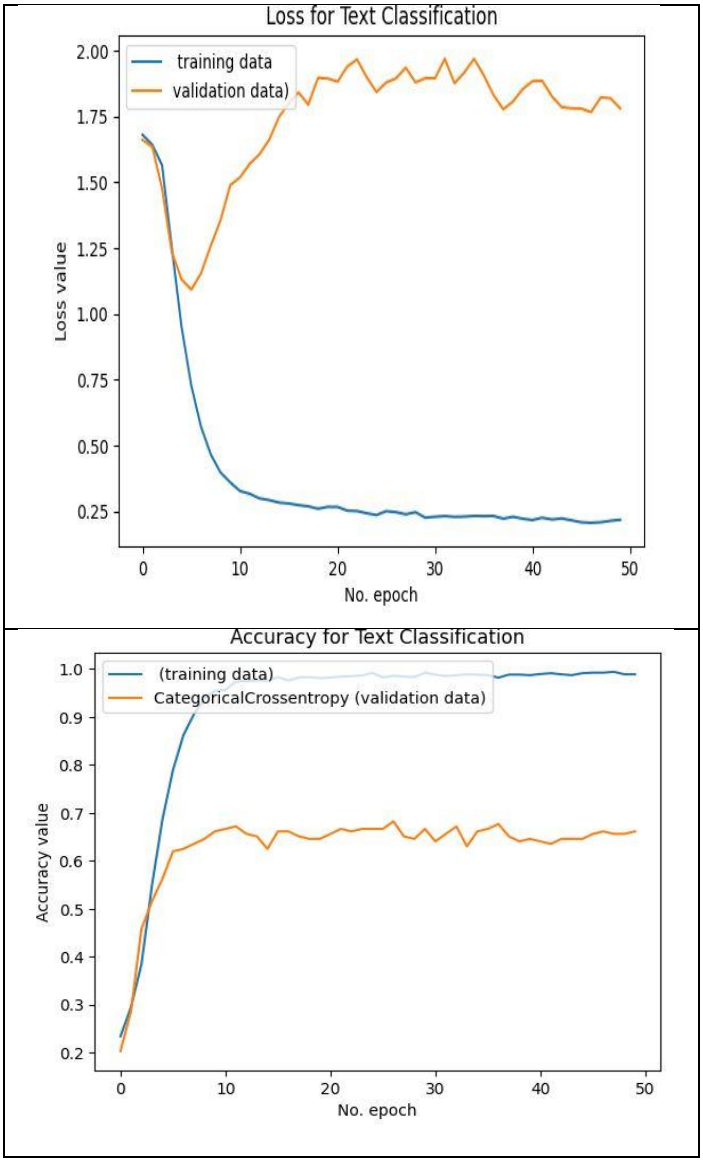
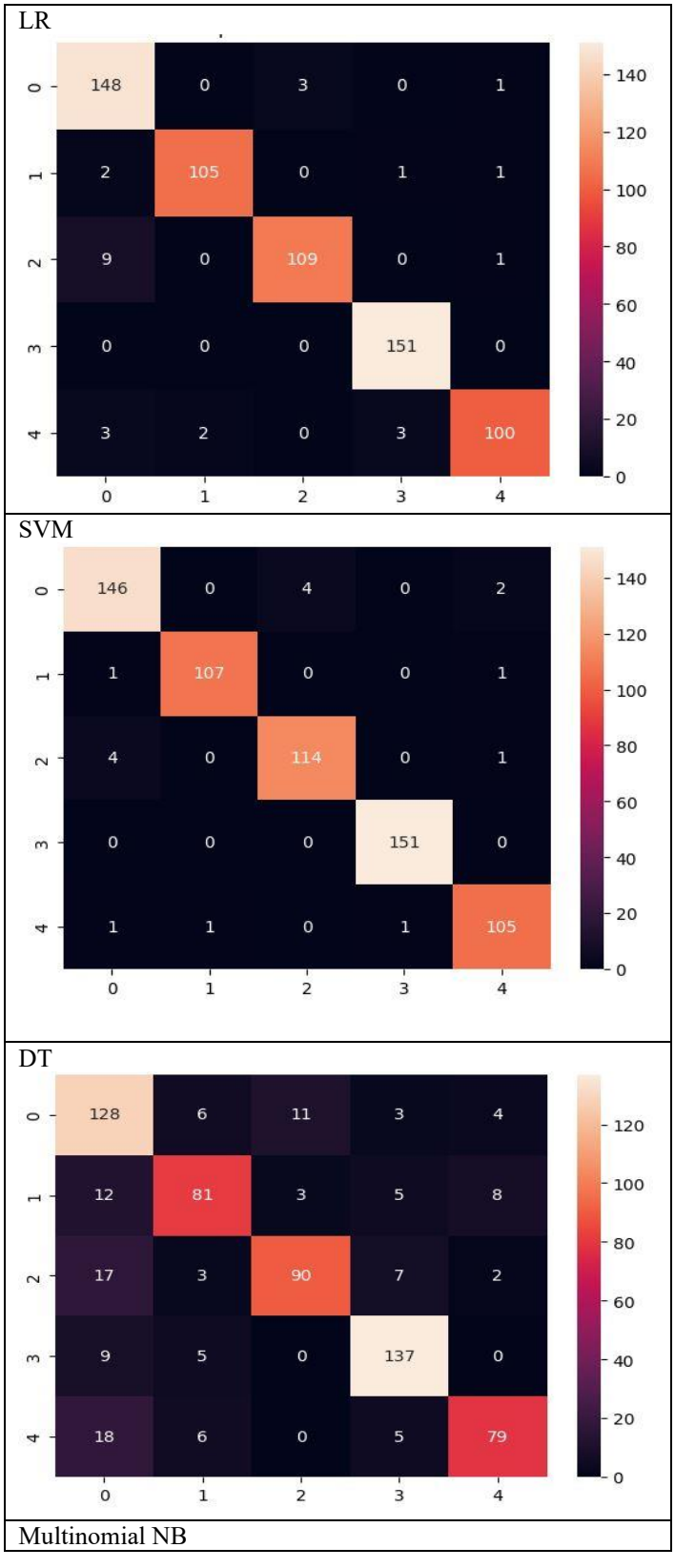


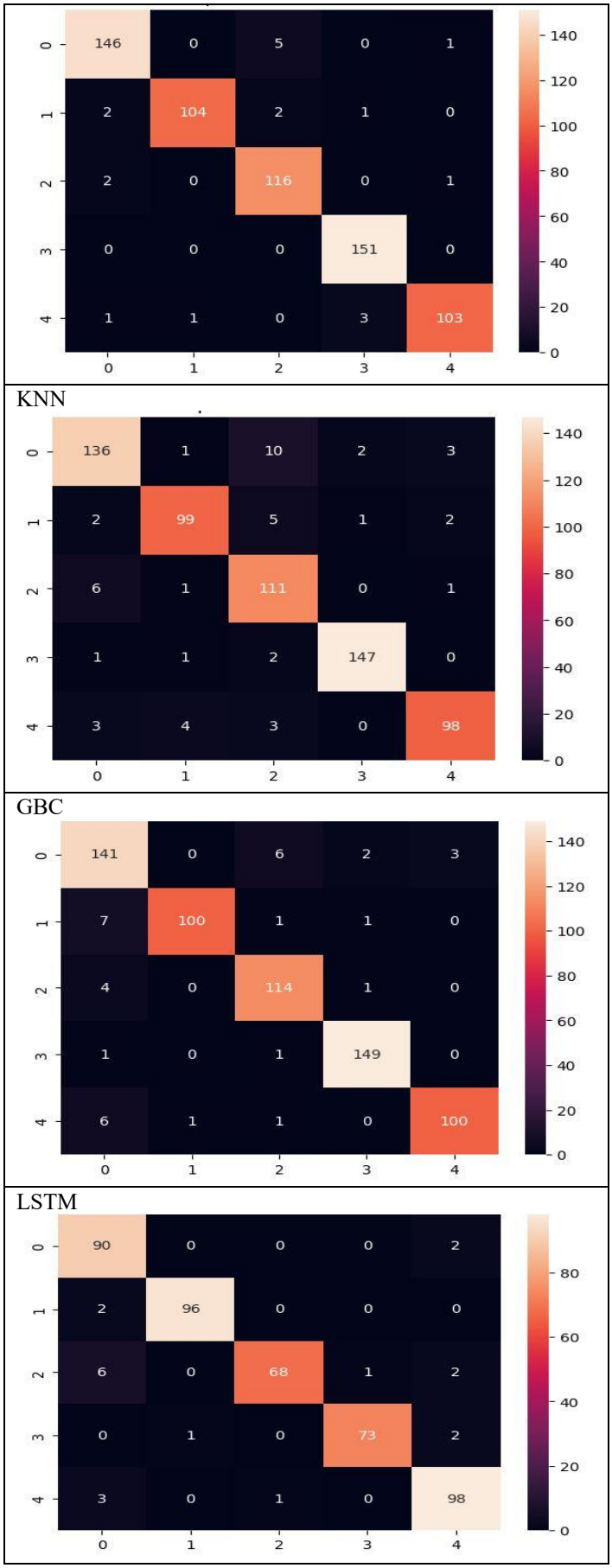
Figure 9: Loss and accuracy for DNN

When classifying data into discrete classes, such sports, technology, entertainment, business, and politics, a confusion matrix is a useful tool in machine learning for assessing how well classification models perform. In this case, the matrix compares true labels with predicted labels to show how well the model separates these five categories. For instance, false positives and false negatives show misclassifications, such as mislabeling a sports item as technology or vice versa, whereas true positives show instances within each category that were correctly classified. Additionally, the matrix offers insightful information about particular aspects of model performance, which aids analysts in optimizing algorithms and raising accuracy across all categories. The confusion matrix makes it easier to see a model's advantages and disadvantages when it comes to handling the intricate world of content classification by displaying the number of categories and the accompanying predictions. In Figure 10. (LR) model's confusion matrix summarizes the model's performance in classifying different

types of data, showing the distribution of predictions across

Five different categories: business, entertainment, politics, sports, and technology. The matrix shows that the model correctly classified 148 instances in the business category, indicating a relatively strong performance in this segment, while it slightly underperformed in the entertainment category, with 105 correct classifications; additionally, the politics category had 109 correct predictions, indicating moderate accuracy; the sports category had the highest number of correct classifications, at 151, indicating robust prediction capabilities in this area; while the technology category had the fewest correct classifications (100), indicating the model's ability to distinguish between categories. 146 data points were correctly classified as Business, 107 as Entertainment, 114 as Politics, 151 as Sports, and 105 as Technology. The remaining data points that were incorrectly classified are dispersed throughout the second confusion matrix for SVM model, which looks to be a table showing the model's classification accuracy across five categories. The following is a breakdown of the numbers in each cell: False Positives (FP) are data points that were mistakenly predicted to belong to another category, True Negatives (TN) are data points that were correctly predicted to not belong to a specific category, FN (False Negatives) are data points that were mistakenly predicted to belong to their own category, and TP (True Positives) are the correct classifications (146, 114, and 151). With counts of 128 for business, 81 for entertainment, 90 for politics, 137 for sport, and 79 for technology, (DT) model demonstrated its relative superiority in classifying sports data, but its effectiveness in entertainment was lower. With counts of 146, 104, 116, 151, and 103, respectively, (NB) model showed a balanced approach and efficacy in detecting all categories, with Sports showing the best results. While (GBC) recorded 141, 100, 114, 149, and 100, indicating a similar pattern of strength in Sports classification, the (KNN) model produced counts of 136, 99, 111, 147, and 98, demonstrating a strong performance, particularly in Sports. However, with scores of 90, 96, 68, 73, and 98, (LSTM) model performed comparatively worse across the board, particularly in the Politics classification. With counts of 147, 104, 113, 151, and 105, (MLP) model demonstrated respectable performance in all categories, especially in sports. (DNN) model, on the other hand, fared noticeably worse across all categories, scoring only 22 for Business, 43 for Entertainment, 26 for Politics, 27 for Sport, and 36 for Technology. While (CNN) demonstrated a noticeable weakness with only 24 for Business, 45 for Entertainment, 30 for Politics, 33 for Sport, and 34 for Technology, (RF) model provided a strong performance with counts of 148, 104, 110, 149, and 100, demonstrating particular strength once more in Sports. The models' overall efficacy varied greatly; Multinomial NB and RF were among the best.





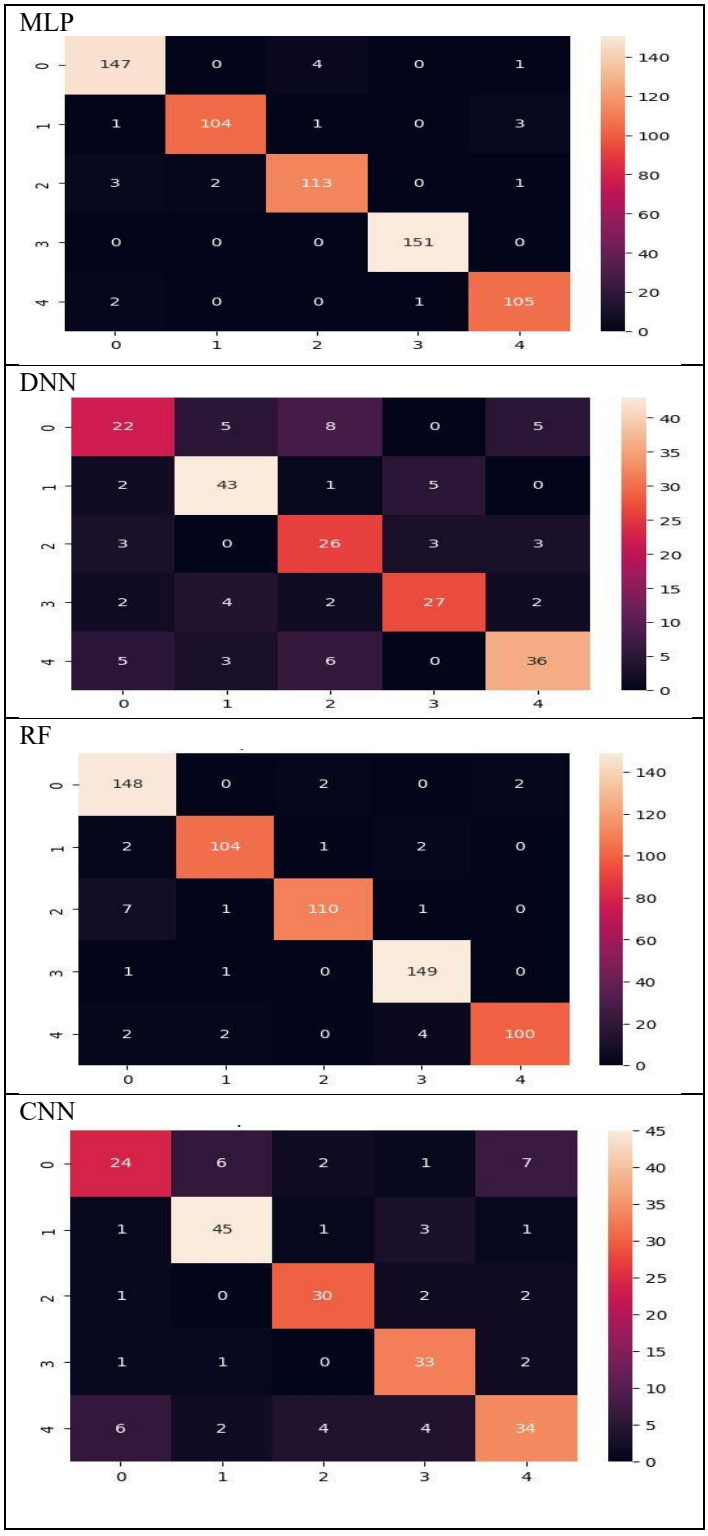


Figure 10: Confusion Matrices for all categories

The performance of various classification models across all categories is depicted in this chart. The most accurate models are RF and Multinomial NB, which perform very well in sports classification. Strong and reliable results are also shown by MLP and LR. On the other hand, CNN and DNN models exhibit poor performance in every category, with a significantly low percentage of accurate classifications. The Confusion matrices show the advantages and disadvantages of each model in terms of category differentiation.

Precision (prec), recall, and F1 scores (F1) for different models in five categories—politics, sports, technology, entertainment, and business—are displayed in Table 1. SVM consistently outperforms the other models, obtaining high scores in every category and primarily exhibiting prec, recall, and F1 values between 0.96 and 1.00. Multinomial NB and MLP also perform well and consistently, with scores that are not far behind SVM's. However, with far poorer precision, recall, and F1 scores—especially in the business category—the (DNN) performs the worst across all categories. In comparison to other models, CNN and (DT) also perform poorly, particularly in domains like technology and business. Based on these criteria, SVM is the most effective model overall for classification, whereas DNN is the least successful.

In Table 2.The models that perform the best are (SVM), (MLP), and (multinomial NB), each of which achieves an accuracy of 97% and prec, recall, and F1 score of 0.97. With overall scores of 0.96, (LR) comes in second, demonstrating a great capacity for prediction. (GBC) performs marginally worse, scoring 0.95 and having a 95% accuracy rate. (KNN), on the other hand, exhibits more reasonable outcomes, with an accuracy of 92%, prec of 0.92, recall of 0.92, and F1 score of 0.93. Similar to LR, (RF) classifier maintains a balanced 0.96 across all measures. (DNN) and (CNN) show a notable decline in performance, with DNN attaining an accuracy of 72% (0.73 prec, 0.72 recall, and F1 score) and CNN obtaining an accuracy of 78% (0.78 across all metrics), we note that the accuracy of both DNN and CNN relatively few, and this is due to the fact that the number of epochs used is only 50, and perhaps if we increase it, we will get better results for both of them.. With an accuracy of 81% (0.81 prec, recall, and F1 score), (DT) likewise performs poorly.

The models are ranked from best training time to shortest training time as follows, based, that is provided: With training times of 2.8371810913085938e-05s and 2.8848648071289062e-05s, respectively, MLP and LR trailed closely behind LSTM, which completed in the quickest amount of time at 2.7894973754882812e-05s. CNN and RF, two more highly competitive models, have training times of 2.86102294921875e-05s and 2.8848648071289062e-05s, respectively. At the same time, Multinomial NB and DT completed their training in 2.9325485229492188e-05s. With training times ranging from 2.956390380859375e-05s to 4.076957702636719e-05s, the remaining models-DNN, GBC, SVM, and KNN—took longer to train and were ranked Lowest in terms of training efficiency.

Table 3 summarizes the mean scores and standard deviations of the machine and deep learning models' performance characteristics. Despite having a relatively low average performance, (SVM) shows significant variability in its outputs, as evidenced by its 29.6 mean and extremely large standard deviation of 302. With a comparable mean of 29.8 and a standard deviation of 285, (LR) indicates stability but weak predictive power. (NB) displays similar performance levels with a mean of 30.1 and a standard deviation of 206. With a high mean of 92.3 and standard deviation of 988, (KNN) stands out as a model that produces predictions that are remarkably consistent. With a mean of 40.1 and a standard deviation of 422, DT exhibits modest performance with notable variability. With a mean of 74.2 and a large standard deviation of 817, (RF) exhibits great average performance but a wide range of output variance. With a mean of 59.6 and a standard deviation of 128 (GBC) exhibits modest performance and variability. (DNN) exhibits significant output variation despite great average performance, as evidenced by its high mean of 91.7 and incredibly enormous standard deviation of 743. With a mean of 95.8 and a standard deviation of 634, (CNN) performs well but exhibits significant variability. The MLP model has moderate to high performance with noticeable output dispersion, as evidenced by its mean of 78.7 and standard deviation of 611. Finally, (LSTM) model performs poorly and is unpredictable, as evidenced by its lowest mean of 17.4 and highest standard deviation of 889. Overall, the data shows a range of model performance levels, with some exhibiting consistent outcomes and others exhibiting notable fluctuation. Each model has unique prediction strengths and limits.

The information presented in Table 4, shows the accuracy of different deep learning and machine learning models as documented in related works and contrasts them with the accuracy attained by a suggested method in three different investigations. Notably, (SVM) performs admirably, outperforming the highest comparable task accuracy of 93.8% with an accuracy of 98.02% in the suggested method. Similarly, with an astounding accuracy of 98.66%, the (Multinomial NB) model fared noticeably better than the earlier research. Although precise comparisons from other research were not given, the suggested (LR) and (MLP) models both demonstrate excellent performance, reaching 96% accuracy. However, the effectiveness of (RF) and (DT) models varies; in one related work, RF achieved 74.2%, whereas in the proposed approach, it achieved 96%. Additionally, the performance of DT increased significantly from 78.1% in related works to 96% in the proposed approach. In the suggested method, (KNN) model performs consistently with 92%, while in related studies; it performs with 85.9% and 91.9%. When compared to related research, the suggested methods demonstrate a notable improvement across nearly all models, suggesting that the new approach may include improved tactics or methodologies that lead to higher accuracy rates.

Table 1: comparison of categorization models' performance in various categories. In the categories of Politics, Sports, Technology, Entertainment, and Business, the table displays the (prec), Recall, and F1 for a variety of machine and deep learning models. With excellent precision and recall levels, SVM and LSTM consistently outperform other models in the majority of categories.

Model	politics			sport			Technology			Entertainment			Business		
	Prec	Recall	F1	Prec	Recall	F1	Prec	Recall	F1	Prec	Recall	F1	Prec	Recall	F1
SVM	0.97	0.96	0.96	0.99	1.00	1.00	0.96	0.97	0.97	0.99	0.98	0.99	0.96	0.96	0.96
MLP	0.96	0.95	0.95	0.99	1.00	1.00	0.95	0.97	0.96	0.98	0.95	0.97	0.96	0.97	0.96
GBC	0.93	0.96	0.94	0.97	0.99	0.98	0.97	0.93	0.95	0.99	0.92	0.95	0.89	0.93	0.91
LR	0.97	0.92	0.94	0.97	1.00	0.99	0.97	0.93	0.95	0.98	0.96	0.97	0.91	0.97	0.94
KNN	0.85	0.93	0.89	0.98	0.97	0.98	0.94	0.91	0.92	0.93	0.91	0.92	0.92	0.89	0.91
multinomialNB	0.94	0.97	0.96	0.97	1.00	0.99	0.98	0.95	0.97	0.99	0.95	0.97	0.97	0.96	0.96
RF	0.97	0.92	0.95	0.96	0.99	0.97	0.98	0.93	0.95	0.96	0.95	0.96	0.93	0.97	0.95
DNN	0.60	0.74	0.67	0.77	0.73	0.75	0.78	0.72	0.75	0.78	0.84	0.81	0.65	0.55	0.59
LSTM	0.99	0.88	0.93	0.99	0.96	0.97	0.94	0.96	0.95	0.99	0.98	0.98	0.89	0.98	0.93
CNN	0.81	0.86	0.83	0.77	0.89	0.82	0.74	0.68	0.71	0.83	0.88	0.86	0.73	0.60	0.66
DT	0.87	0.76	0.81	0.87	0.91	0.89	0.85	0.73	0.79	0.80	0.74	0.77	0.70	0.84	0.76

Table 2: Comparison of training time and model performance: prec, recall, accuracy, F1, and training time for a number of classification models are shown in this table.
The models like CNN and DNN performed poorly, SVM, MLP, and Multinomial NB had the highest accuracy (97%). All models have comparatively low training times, albeit each model's computational efficiencyvaries .

Model	Weighted avg			accuracy	Training time
	Prec	Recall	F1		
SVM	0.97	0.97	0.97	97	3.337860107421875e-05s
MLP	0.97	0.97	0.97	97	2.8371810913085938e-05s
GBC	0.95	0.95	0.95	95	2.9087066650390625e-05s
LR	0.96	0.96	0.96	96	2.8848648071289062e-05s
KNN	0.92	0.92	0.93	92	4.076957702636719e-05s
Multinomial NB	0.97	0.97	0.97	97	2.9325485229492188e-05s
RF	0.96	0.96	0.96	96	2.8848648071289062e-05s
DNN	0.73	0.72	0.72	72	2.956390380859375e-05s
LSTM	0.96	0.96	0.96	96	2.7894973754882812e-05s
CNN	0.78	0.78	0.78	78	2.86102294921875e-05s
DT	0.81	0.81	0.81	81	2.9325485229492188e-05s

Model	Mean	Standard Deviation
SVM	29.6	302
DNN	91.7	743
GBC	59.6	128
LR	29.8	285
KNN	92.3	988
Multinomial NB	30.1	206
RF	74.2	817
DT	40.1	422
LSTM	17.4	889
CNN	95.8	634
MLP	78.7	611

Table 3: show the mean and standard deviation of performance for each classification model.
The mean and standard deviation for each model's performance are shown in the table. CNN has the highest average. KNN, RF, and LSTM show higher standard deviations, indicating greater variability, whereas models like SVM, LR, and Multinomial NB demonstrate more consistent performance with smaller standard deviations.

Table 4: Accuracy comparison of the suggested method and various models in related work.

The accuracy of several models from relevant publications [63], [64], [65], and the suggested method is contrasted in the table. The suggested method exhibits significant gains in accuracy for models like SVM, Multinomial NB, and MLP.

Model	Accuracy of Related Work			Accuracy of the Proposed approach
	[63]	[64]	[65]	
SVM	93.8%		98.02%	97%
LR	-------	----------	---------	96%
,MLP	-------	----------	---------	**97%**
RF	--------	74.2%	---------	**96%**
MultinomialNB	82.8%	74.6%	98.66%	97%
KNN	85.9%		91.9%	**92%**
GBC	------	----------	---------	95%
DT	78.1%	----------	96%	81%
LSTM	---------	--------	---------	96%
CNN	---------	----------	---------	78%
DNN	---------	----------	---------	72%

5. Conclusion

This study shows the advantages and disadvantages of different machine learning and deep learning models while highlighting the changing field of text document classification within (NLP). We have thoroughly examined models like (SVM), (MLP), (CNN), (LSTM) and others by carefully examining a dataset of 2,225 text documents in five categories: business, politics, sports, technology, and entertainment. Our results demonstrate the superior performance of ensemble approaches, particularly the SVM and MLP, which consistently had precision and recall metrics above 0.96 across a variety of categories and attained accuracy rates of 97%. These findings support the efficiency of optimized machine learning algorithms in managing the intricacies of text classification, clearly separating successful models from underperforming ones, like (CNN) and (DNN), which had generalization issues. The study opens the door for further research targeted at improving automated text classification systems by highlighting the significance of model selection adapted to certain classification tasks. Additionally, it tackles common problems in document categorization, like data noise and the difficulties posed by different text formats. The findings also point to the necessity of ongoing model adaptation and optimization in order to better accommodate the complex properties of the textual input being processed. Furthermore, our experimental analyses, which include confusion matrix Evaluations and performance metric assessments, support the notion that traditional machine learning techniques may provide more consistent generalization across a variety of datasets, even though deep learning models like (LSTM) show promise in sequential data handling. This study is an essential resource for academics and professionals who want to use these discoveries to further push the limits of text classification Techniques as NLP technology develops.

Data availability statement:

The data used to support the findings of this study are available from the corresponding author upon request.

Acknowledgments:

The authors are thankful to the editor and reviewers for their Valuable comments towards improving this paper.

References

[1] Z. Zhang, *Applied Computational Engineering*, 57 (2024) 146–152.

[2] K.L. Reddy, P. Shanmukh, C. Kumar, T. Kumar, A. Kumar, P. Kumar, K. Venkatraman, *Fourth International Conference on Advances,* 2024.

[3] A.W. Singh Lo, A. Manish, *Journal of Portfolio Management*, 49 (2023) 201–235.

[4] S. Freyberg, H. Hauser, *Studies in History and Philosophy of Science Part A*, 100 (2023) 1–11.

[5] L. Geiszler, *Studies in History and Philosophy of Science Part A*, 100 (2023) 22–31.

[6]. J. Burton, *Technology in Society*, 75 (2023).

[7] V.K. Finn, *Automatic Documentation and Mathematical Linguistics*, 54 (2020) 140–173.

[8] F. Ansari, *IFAC-PapersOnLine*, 52 (2019) 1597–1602.

[9] E. Baralis, L. Cagliero, S. Jabeen, A. Fiori, S. Shah, *Data Mining and Analysis in the Engineering Field*. 1st Edition, IGI Global, Hershey, PA, USA, 2014.

[10] L. Waardenburg, M. Huysman, *Information and Organization*, 32 (2022) 1-11.

[11] S. Gupta, S. Modgil, A. Kumar, U. Sivarajah, Z. Irani, *International Journal of Production Economics*, 254 (2022) 108642.

[12]R. Valencia-García, F. García-Sánchez, *Computer Standards & Interfaces*, 35 (2013) 415–416.

[13] C. Aggarwal, C. Zhai, *Mining text data*. 1st Edition. Springer, New York, 2013.

[14] D. Zeng, K. Liu, Y. Chen, J. Zhao, *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing* , 2015.

[15]A.H. Wang, *International Conference on Security and Cryptography*, (2010) 142-151.

[16] S. Xie, G. Wang, S. Lin, P.S. Yu, *Proceedings of the 18th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 2012.

[17] A.Y.N., C.P. Richard Socher, Alex Perelygin, Jean Y. Wu, Jason Chuang, Christopher D. Manning, *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing* , 2013.

[18] Y.H. Li, A.K. Jain, *the Computer Journal*, 41 (1998) 537–546.

[19] D. Isa, L.H. Lee, V.P. Kallimani, R. Rajkumar, *IEEE Transactions on Knowledge and Data Engineering*, 20 (2008) 1264–1272.

[20] V. Mitra, C.J. Wang, S. Banerjee, *Applied Soft Computing*, 7 (2007) 908–914.

[21] S. Bird, E. Loper, *Proceedings of the ACL Interactive Poster and Demonstration Sessions*, 2004.

[22] R. Agarwal, H. Mannila, R. Srikant, H. Toivonen, A.Verkamo, *Advances in Knowledge Discovery and Data Mining*. 1st Edition. AAAI Press / MIT Press, Menlo Park, CA, 1996.

[23] M. Chowdhury, F.A. Sohel, P. Naushad, K.S.M. Kamruzzaman, *International Conference on Information Technology*, 2003.

[24] S.M. Kamruzzaman, F. Haider, *international Conference on Electrical & Computer Engineering*, 2004.

[25] L. Rigutini, *Ph.D. Thesis*, Univ. Siena, 2004.

[26] M.F. Porter, *Program: Electronic Library and Information Systems*, 14 (1980) 130–137.

[27] A. Hotho, A. Nürnberger, G. Paaß, *Journal for Language Technology and Computational Linguistics*, 20 (2005) 19–62.

[28]G. Salton, A. Wang, C.S. Yang, *Communications of the ACM*, 18 (1975) 613–620.

[29] Y.H. Li, A.K. Jain, *The Computer Journal*, 41 (1998) 537–546.

[30] A. Hotho, A. Maedche, S. Staab, *workshop "Text Learning: Beyond Supervision*, Morgan Kaufmann, Seattle, WA, USA, 2001.

[31] W. B. Cavnar, *proceedings of the Third Text REtrieval Conference (TREC-3)*, Gaithersburg, Maryland, USA, 1994.

[32] E. Milios, Y. Zhang, B. He, L. Dong, *Pacific Association for Computational linguistics* , Halifax, Nova Scotia, Canada, 2003.

[33] C.P. Wei, C.C. Yang, C.M. Lin, *Decision Support Systems*, 45 (2008) 606–620.

[34] X. He, D. Cai, H. Liu, W.Y. Ma, *Proceedings of the 27th ACM International Conference on Research & Development in Information Retrieval*, Sheffield, South Yorkshire, UK, 2004.

[35] D. Cai, X. He, W.V. Zhang, J. Han, *16th ACM Conference on Information and Knowledge Management (CIKM'07)*, Lisbon, Portugal, 2007.

[36] B. Choudhary, P. Bhattacharyya, *11th International World Wide Web Conference (WWW 2002)*, Honolulu, Hawaii, USA, 2002.

[37] A. McCallum, K. Nigam, *AAAI-98 Workshop on Learning for Text Categorization*, Madison, Wisconsin, USA, 1998.

[38] K.B. Kim, H.C. Rim, D. Yook, H.S. Lim, *7th Pacific Rim International Conference on Artificial Intelligence (PRICAI '02)*, Tokyo, Japan, 2002.

[39] S. Tan, *Expert Systems with Applications*, 30 (2006) 290–298.

[40] E.H. Han, G. Karypis, V. Kumar, *Computer Science and Engineering, University of Minnesota*, Technical Report, 1999.

[41] S. Tan, *Expert Systems with Applications*, 35 (2008) 279–285.

[42] L.M. Wang, X.L. Li, C.H. Cao, S.M. Yuan, *Knowledge-Based Systems*, 19 (2006) 511–515.

[43] D.D. Lewis, M. Ringuette, *3rd Annual Symposium on Document Analysis and Information Retrieval*, Las Vegas, Nevada, USA 1994.

[44] D.D. Lewis, R.E. Schapire, J.P. Callan, R. Papka, *19th International Conference on Research and Development in Information Retrieval (SIGIR '96)*, Zürich, Switzerland, 1996.

[45] E.D. Wiener, J.O. Pedersen, A.S. Weigend, *4th Annual Symposium on Document Analysis and Information Retrieval*, Las Vegas, Nevada, USA, 1995.

[46] T. Joachims, *10th European Conference on Machine Learning (ECML '98)*, Chemnitz, 1998.

[47] N.T. Vu, H. Adel, P. Gupta, H. Schütze, *Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL-HLT 2016)*, San Diego, California, USA , 2016.

[48] J. Silva, L. Coheur, A.C. Mendes, A. Wichert, *Artificial Intelligence Review*, 35 (2011) 137–154.

[49] T. Nakagawa, K. Inui, S. Kurohashi, *Human Language Technologies – NAACL HLT 2010 (The 2010 Annual Conference of the North American Chapter of the Association for Computational Linguistics*, Los Angeles, California, USA, 2010.

[50] R. Ghazali, Z.A. Bakar, Y. Mazwin, M. Hassim, *international Conference on Intelligent Computing*, Springer, 2014.

[51] J. Song, S. Qin, P. Zhang, *2016 IEEE/ACIS 15th International Conference on Computer and Information Science (ICIS)*, Okayama, Japan, 2016.

[52] P. Vincent, *Journal of Machine Learning Research*, 3 (2003) 1137–1155.

[53] R. Ghazali, N.A. Husaini, L.H. Ismail, T. Herawan, Y.M.M. Hassim, *International Joint Conference on Neural Networks (IJCNN)*, IEEE, 2014.

[54] G. Tzortzis, A. Likas, *19th IEEE International Conference on Tools with Artificial Intelligence (ICTAI)*, Patras, Greece, 2007.

[55]R. Collobert, J. Weston, *25th International Conference on Machine Learning (ICML '08)*, Helsinki, Finland, 2008.

[56] M. Zulqarnain, R. Ghazali, M.G. Ghouse, M.F. Mushtaq, *International Journal on Informatics Visualization*, 3 (2019) 377–383.

[57] S. Lai, L. Xu, K. Liu, J. Zhao, *AAAI Conference on Artificial Intelligence (29th AAAI, AAAI '15*), Austin, Texas, 2018.

[58] N. Aloysius, M. Geetha, *IEEE International Conference on Communication and Signal Processing*, Chennai, India, 2017.

[59] T. Mikolov, K. Chen, G. Corrado, J. Dean, *Advances in Neural Information Processing Systems*, Lake Tahoe, Nevada, USA, 2013.

[60] A.K. Arkhipenko, K.I. Kozlov-Ilya, T.J. Integral, S.K. Kirillskorniakov, G.A. Gomzin, T.D. Turdakov, *Computational Linguistics and Intellectual Technologies: Proceedings of the International Conference Dialogue*, Moscow, Russia, 2016.

[61] J. Chung, C. Gulcehre, K. Cho, Y. Bengio, *arXiv preprint* (arXiv:1412.3555, 2014.

[62] A. Dosovitskiy, J.T. Springenberg, T. Brox, *IEEE Conference on Computer Vision and Pattern Recognition (CVPR '15)*, 2015.

[63] Y.M. Tun, P.H. Myint, *International Journal of Computer*, 33 (2019) 19–25.

[64] A.M. Aubaid, A. Mishra, A. Mishra, *International Journal of System Assurance Engineering and Management*, 15 (2024) 5637-5652.

[65] A. Khatun, M.M.H. Matin, M.A. Miah, M.R. Miah, M. d. Robbani, *International Journal of Engineering Science Invention*, 9 (2020) 21-33.

[66] D. Greene, P. Cunningham, *23rd International Conference on Machine Learning (ICML '06)*, ACM, 2006.

[67] P. Bhuvaneshwari, A. N. Rao, *International Journal of Cloud Computing*, 11 (2022) 61-78.