Computer Systems Science and Engineering

Tech Science Press

DOI: 10.32604/csse.2024.045066

ARTICLE

Analyzing COVID-19 Discourse on Twitter: Text Clustering and Classification Models for Public Health Surveillance

Pakorn Santakij[1], Samai Srisuay[2,*] and Pongporn Punpeng[1]

[1]Department of Information Technology, Lampang Rajabhat University, Lampang, 52100, Thailand

[2]Department of Computer Science, Lampang Rajabhat University, Lampang, 52100, Thailand

*Corresponding Author: Samai Srisuay. Email: samai@lpru.ac.th

Received: 16 August 2023 Accepted: 26 January 2024 Published: 20 May 2024

ABSTRACT

Social media has revolutionized the dissemination of real-life information, serving as a robust platform for sharing life events. Twitter, characterized by its brevity and continuous flow of posts, has emerged as a crucial source for public health surveillance, offering valuable insights into public reactions during the COVID-19 pandemic. This study aims to leverage a range of machine learning techniques to extract pivotal themes and facilitate text classification on a dataset of COVID-19 outbreak-related tweets. Diverse topic modeling approaches have been employed to extract pertinent themes and subsequently form a dataset for training text classification models. An assessment of coherence metrics revealed that the Gibbs Sampling Dirichlet Mixture Model (GSDMM), which utilizes trigram and bag-of-words (BOW) feature extraction, outperformed Non-negative Matrix Factorization (NMF), Latent Dirichlet Allocation (LDA), and a hybrid strategy involving Bidirectional Encoder Representations from Transformers (BERT) combined with LDA and K-means to pinpoint significant themes within the dataset. Among the models assessed for text clustering, the utilization of LDA, either as a clustering model or for feature extraction combined with BERT for K-means, resulted in higher coherence scores, consistent with human ratings, signifying their efficacy. In particular, LDA, notably in conjunction with trigram representation and BOW, demonstrated superior performance. This underscores the suitability of LDA for conducting topic modeling, given its proficiency in capturing intricate textual relationships. In the context of text classification, models such as Linear Support Vector Classification (LSVC), Long Short-Term Memory (LSTM), Bidirectional Long Short-Term Memory (BiLSTM), Convolutional Neural Network with BiLSTM (CNN-BiLSTM), and BERT have shown outstanding performance, achieving accuracy and weighted F1-Score scores exceeding 80%. These results significantly surpassed other models, such as Multinomial Naive Bayes (MNB), Linear Support Vector Machine (LSVM), and Logistic Regression (LR), which achieved scores in the range of 60 to 70 percent.

KEYWORDS

Topic modeling; text classification; twitter; feature extraction; social media

1 Introduction

The COVID-19 pandemic is caused by the Severe Acute Respiratory Syndrome-Coronavirus-2 (SARS-CoV-2) virus, which was first discovered in Wuhan, China, and has rapidly spread to

many countries around the world. The World Health Organization (WHO) declared the outbreak a pandemic on March 11, 2020 [1]. The pandemic has continued to spread in many waves due to the emergence of numerous variants, some of which have caused severe outbreaks. One example is the B.1.1.529 variant of the SARS-CoV-2 virus, which was first identified in Gauteng province, South Africa, and has been designated as the Omicron variant by the WHO. The Omicron variant has spread rapidly, surpassing the previously dominant Delta variant in many countries, thereby sparking significant concern and discussion on social media.

According to a report by Recode [2], online posts concerning Omicron has exhibited a prevalence six times greater than that of Delta during the equivalent timeframe. Comments by users on social media regarding the widespread outbreak of various strains of COVID-19 have prompted an operational meeting on natural language processing [3]. The meeting emphasized the potential for natural language processing (NLP) to respond to the ongoing epidemic by collecting scientific literature for data analysis of social media and disseminating natural language datasets concerning COVID-19 topics. Textual exchanges on social media platforms like Twitter and Facebook can be grouped for topic modeling, providing continuous tracking and evaluation of epidemic development [4].

Twitter is a widely used online social media platform. Rathore et al. demonstrated that Twitter is the most popular online social media platform when compared to others, owing to its diverse range of applications. Data from Twitter can be easily collected and analyzed by using keywords or hashtags. The ease of data collection using APIs makes deep analysis of Twitter data more convenient, which is often lacking in other platforms. Many studies have used Twitter data to analyze the early stages of the COVID-19 pandemic, with data ranging from hundreds of thousands to millions of tweets. These studies have provided insights into how Twitter users react to the pandemic and their concerns during the initial phases of the outbreak [1].

An approach to utilizing NLP in Twitter data analysis is to create an unsupervised model for topic modeling. Topic modeling is considered as one of the fundamental tasks in applying machine learning techniques. This research focuses on studying literature related to topic modeling. Several methods have proven effective in analyzing long texts, but they yield varying results when applied to short texts, particularly user-generated content (UGC) on online social media platforms. UGC presents specific challenges, such as incorrect spelling, slang usage, data sparsity, and the co-occurrence of infrequent words. To address these challenges, research has been conducted to compare the performance of different topic modeling techniques, including Latent Semantic Analysis (LSA), LDA, NMF, Principal Component Analysis (PCA), and Random Projection (RP) [5,6], using metrics such as Recall, Precision, F-Score, and Topic Coherence. Evaluation results indicate that LDA and NMF consistently demonstrate strong performance in topic modeling [7,8]. Furthermore, in reviewing the field of short text modeling, additional experiments have been conducted. Yong Chen and colleagues conducted a series of experiments to compare basic LDA and NMF with different settings on public short text datasets, finding that NMF tends to outperform LDA. This finding is consistent with the work of Zoya et al., who conducted experiments to cluster the topics of Urdu tweet text using LSA, PLSA, LDA, and NMF. They observed that NMF outperformed the other methods when used in conjunction with term frequency-inverse document frequency (TF-IDF) and bigram on the dataset. However, LDA provided the best results when used to cluster topics in a dataset that combined short text with pseudo documents [9].

In the context of COVID-19-related datasets, Mifrah et al. conducted experiments to classify topics using NMF and LDA, comparing their performance with the C_v measure. Their findings indicated that LDA outperformed NMF in terms of topic coherence scores [10]. Weisser et al. also

conducted experiments on topic classification using a COVID-19 outbreak dataset from Twitter. They used GSDMM and GPM, specifically designed for sparse text data, and compared them with LDA. Their results revealed that both GSDMM and GPM were more effective at generating topics than LDA [11]. Ridhwan et al. utilized LDA to determine the suitable number of topics for GSDMM as a parameter, which resulted in improved topic classification compared to using LDA alone [1].

In topic modeling tasks that emphasize feature extraction, Subakti et al. used BERT, a DNN model for data representation, and compared it to using TF-IDF with multiple clustering models. The results showed that BERT outperformed TF-IDF, even when used with several clustering models such as K-means [12]. Atagun et al. conducted experiments using BERT+LDA to generate vector representations for topic modeling, which yielded superior results compared to TF-IDF [13]. In addition, Sethia et al. conducted experiments on topic classification using the 20-Newsgroup dataset, employing a Hybrid BERT and LDA model. They used BERT and LDA for vector representation along with K-means clustering for topic modeling and compared this approach to other word embedding techniques, such as Word2Vec and Doc2Vec. The findings demonstrated that the proposed method achieved the highest NMI scores [14]. Furthermore, Lande et al. presented a topic modeling framework for extracting topics from a Twitter dataset related to COVID-19 in India. They used a BERT-based word embedding technique and applied Uniform Manifold Approximation and Projection (UMAP) for dimensionality reduction and clustering via HDBSCAN. This framework was compared to GSDMM and LDA, and the experimental results revealed superior performance in terms of topic coherence [15].

In addition to topic modeling, text classification is an ongoing process that requires learning from labeled datasets to apply the classifier to new types of text on Twitter. Various models are used for text classification, including traditional, DNN, and hybrid models. For instance, González-Carvajal et al. compared BERT with traditional models such as LR, Linear SVC, and Multinomial MB for text classification on the IMDB dataset. Their study demonstrated that BERT outperformed the traditional models, highlighting the potential of advanced models like BERT in enhancing text classification accuracy [16]. Benítez-Andrades et al. conducted experiments involving the classification of racism and xenophobia texts from Twitter using CNN, LSTM, BERT, and BETO, a BERT-Based Spanish pre-trained model. The findings showed that BETO performed the best in their experiment [17]. Additionally, a comparison between BERT and LSTM for text classification in a small dataset revealed that LSTM slightly outperformed BERT. It is important to note that these results depend on various factors, including the dataset and hyperparameter tuning [18]. Mohd et al. used CNN and CNN-LSTM and found that traditional models with BOW or TF-IDF as vector representation were not suitable for datasets with data sparsity. In their experiment, CNN-LSTM performed better [19]. A hybrid CNN-BiLSTM attention ensemble was proposed and compared with various traditional and DNN models. The results indicated that the proposed model provided better results in multiclass text classification task [20,21]. Meanwhile, Alhaj et al. [22] conducted text classification experiments for the Arabic language using both traditional models including MNB, LR, SVC, LSVM and DNN. Surprisingly, their evaluation found that traditional models, such as SVC and LSVM, outperformed the DNN. Furthermore, optimizing models to capture slight variations in topics may lead to skewed results in specific directions. This underscores the unreliability of relying solely on a single topic model, emphasizing the necessity of comparing diverse algorithms [23].

Therefore, based on the issues mentioned above, this paper presents a study that utilizes NLP techniques to achieve beneficial outcomes. This research applied the various techniques mentioned earlier for two main purposes. The first purpose was to create a topic modeling model for classifying topics mentioned by Twitter users regarding the spread of COVID-19, utilizing a dataset of comments

on Twitter related to the outbreak of COVID-19. The second purpose of the experiment was to find an optimal model for classifying text that references COVID-19, achieved by training it with a labeled dataset obtained from the topic modeling model created in the initial experiment. The benefit of this experiment lies in its potential to apply these models for monitoring the outbreak of COVID-19 or other diseases caused by various coronaviruses that may emerge in the future through social media.

2 Materials and Methods

The experimental process involves the following steps: First, the dataset is prepared by selecting relevant data and cleaning it for analysis. Next, topic models are created using three popular techniques and a hybrid technique namely NMF, LDA, GSDMM, and BERT-LDA-K-means. The performance of these models is compared using topic coherence metrics. The best-performing model is selected to label the dataset for training several traditional and Deep Neural Network (DNN) text classification models, which include MNB, LSVM, LSVC, LR, LSTM, BiLSTM, CNN-BiLSTM, and BERT. Finally, the performance of these models is evaluated using standard evaluation metrics such as precision, recall, and F1-Score. An overview of the experimental design is presented in Fig. 1, followed by details of the methods and materials used in this study.

2.1 Collection of Tweets

This research utilized a dataset obtained from Twitter, in accordance with Twitter's terms of service and privacy policy for user data [24]. Twitter designates tweets as public data, and, as of the time of writing this article, the use of this data for experimentation is considered compliant with both policies. The data were collected using the Python library 'snscrape' [1] to retrieve tweets discussing COVID-19. The tweets were limited to those written in English and retrieved between November 01, 2022 and July 07, 2023–a period highly relevant to the Omicron outbreak, with a substantial volume of comments on Twitter [2]. The data retrieval area was centered on Bangkok with a radius of 1500 km, as specified by the user's location profile [25]. The dataset consisted of 120,344 tweets, with tweet lengths ranging from 1 to 52 words and an average of 12 words. Prior to using the dataset to train models, it underwent data preprocessing steps.

2.2 Data Preprocessing

The collected dataset underwent rigorous preprocessing prior to model training. This process involved removing stop words, URLs, tags, and irrelevant content from tweets before lemmatization. The lemmatized tweets were then tokenized into words or tokens to maintain semantic context. Both bigram and trigram were employed to preserve the intended meaning, preventing phrases from splitting into unigram tokens. These preprocessing techniques were integrated to enhance the efficiency of the model in the experiment.

The experiment focused on topic modeling and text clustering for categorization. Topic modeling relies on the distribution of content words, which were identified through string matching achieved by lemmatizing their forms, ensuring consistency across documents. Lemmatization plays a vital role in training word vectors, preventing disruptions caused by irrelevant inflections such as plurals or present tense forms. Notably, only function words that had a minimal impact on meaning were removed during preprocessing, preserving the textual essence.

In this study, data preprocessing techniques were applied to all topic modeling methods except BERT, which utilizes self-attention mechanisms. These preprocessing steps were instrumental in ensuring the quality and relevance of the data used for the analysis.

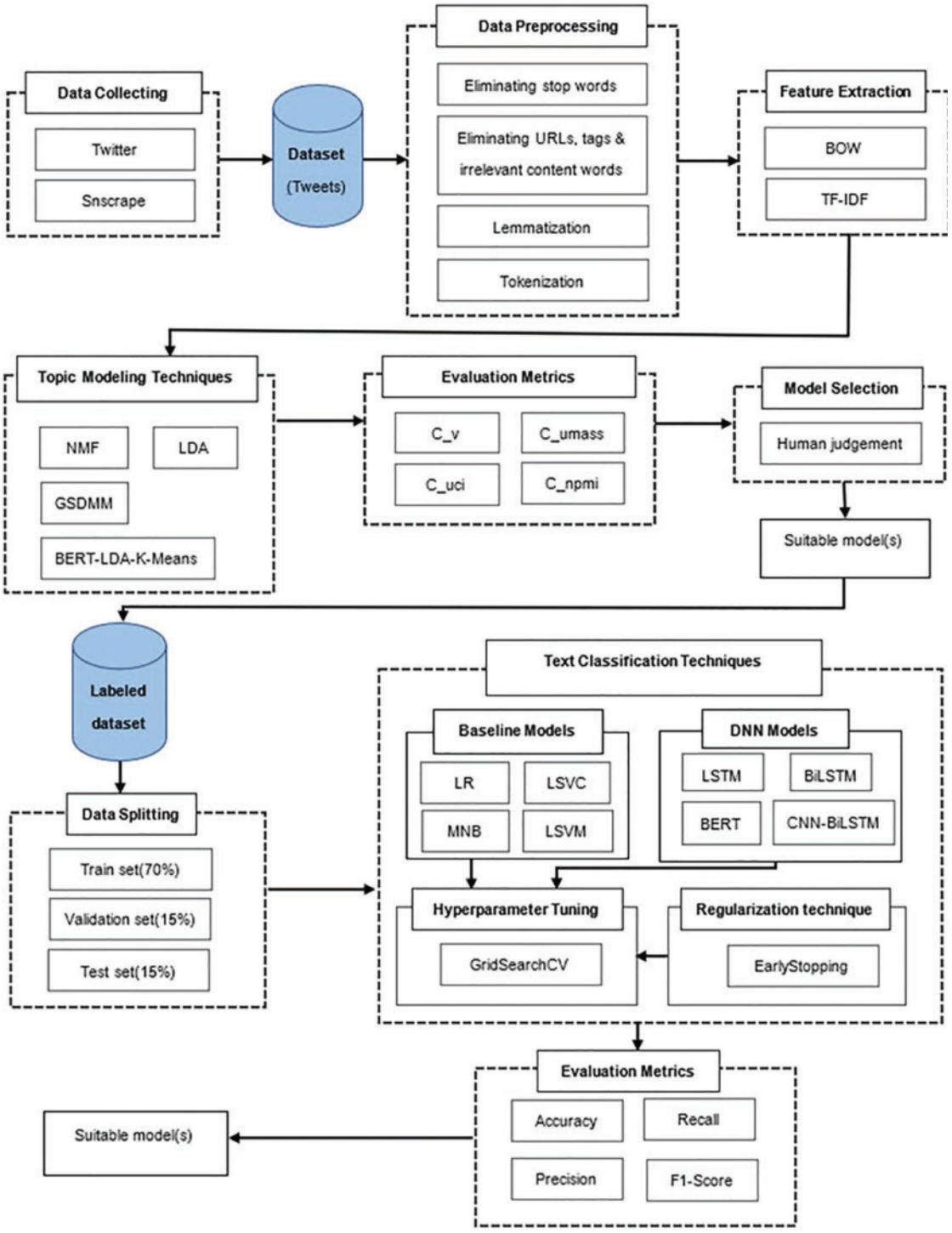

Figure 1: An overview of the experimental design

2.3 Feature Extraction Techniques

Given the short text nature of the dataset, this research employed various feature extraction techniques while taking into account the context of the words. These techniques included the use of bigram and trigram tokenization and the application of BOW [26–28] and TF-IDF [22,29,30] weighting to enhance the features. Additionally, DNNs like BERT were utilized to extract features, along with LDA [13].

2.4 Topic Model Techniques

To identify the topics of the tweets in the dataset, this study employed established methodologies for effective short-text topic modeling [7,8,14]. Fine-tuning of these models involved adjusting hyperparameters, guided by analogous research in the field. The experiment encompassed the evaluation and comparison of four selected models, outlined as follows:

2.4.1 NMF

NMF is an unsupervised algorithm for topic modeling that falls under the category of multivariate analysis and linear algebra. It excels in extracting significant terms that repeat within a corpus. The corpus comprises various documents, with terms collectively forming the content, which can be extracted as different topics embedded within each document. It is assumed that documents with similarities also share similar topics and frequency distribution of words. The NMF method involves the factorization of the document-term matrix V, where each element is non-negative. This factorization results in the product of two lower rank matrices, namely the document-topic matrix W and the topic-term matrix H, such that V is approximately equal to W $\times$ H. This is formally expressed as:

$$V \approx WH \tag{1}$$

In matrix V, vectors are arranged in dimensions n $\times$ m, where m represents the number of word tokens for each of m terms in n documents. This matrix is then factorized into two matrices, W and H, which have dimensions n $\times$ r and r $\times$ m, respectively. Each row in W represents a document and comprises the probabilities of r topics, while each column represents a topic or a semantic feature recurring throughout n documents. Matrix H, representing topics and terms, indicates the number of word tokens for each of the m terms in r topics [31,32].

The implementation of NMF models was conduct by setting their hyperparameters based on recommendations found in [27,31,32]. NMF, incorporating bigram and trigram features along with either bag-of-words or TF-IDF representation, was employed for topic modeling. The range for topic extraction range, spanning from 2 to 30, was determined following the precedents set in topic modeling research, particularly from the work of Ridhwan et al. [1]. Subsequently, NMF was applied to the preprocessed data to factorize the matrix into a document-topic matrix and a topic-word matrix. The interpretation of topics involved examining the top 20 words associated with each topic and assigning a label based on these words. Finally, the quality of the topics was evaluated using metrics such as coherence score and human evaluation.

The NMF models were configured with specific hyperparameters for optimal performance. These parameters included ‘n_components’ (K), defining the number of topics to be extracted (set to k = 30), ‘beta_loss’ determining the distance measure within the objective function (specified as ‘frobenius’), and the ‘solver’ attribute choosing the optimization method (‘mu’). Additionally, ‘max_iter’ was set to impose a limit on the maximum iterations before convergence (established at 1000), while the

initialization methods for the W and H matrices were set using the 'nndsvda' approach. The models also incorporated 'alpha' as a multiplier for the regularization term (with a value of 5e-5) and employed 'l1_ratio' to define the type of regularization, allowing for pure L2 (0), pure L1 (1), or a blend of both (ranging from 0 to 1). Specifically, the experiments conducted with the NMF models utilized a 'l1_ratio' value of 0.5.

2.4.2 LDA

LDA is a three-level Hierarchical Bayesian Model in which each document in the corpus is a mixture of topics. Each topic is a probability distribution over the words, and finally, each word in the document is attributed to a particular topic with probability given by the distribution. The process of LDA is described through plate notation, as shown in Fig. 2.

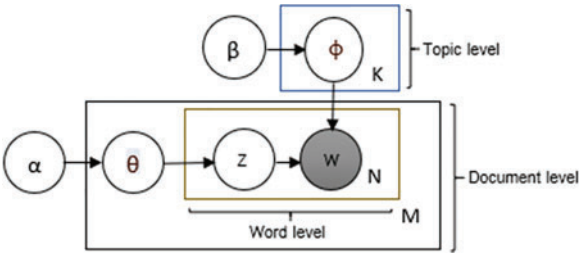

Figure 2: Plate notation representing the LDA model [33]

Fig. 2 represents the LDA Model with plate notation. The boxes are "plates" representing replicates. The outer plate represents documents, while the inner plate represents the repeated choice of topics and words within a document. The many variable names are defined as follows: α is the parameter of the Dirichlet prior on the per-document topic distributions, β is the parameter of the Dirichlet prior on the per-topic word distribution. M denotes the number of documents. N is number of words in a given document (document i has N_i words). θ is a joint distribution of a topic mixture, θ_i is the topic distribution for document i. ϕ_k is the word distribution for topic k where K denotes the number of topics. Z is the latent topic assigned to a word, where Z_{ij} is the topic for the j-th word in document i. Finally, W is the identity of the vocabulary set composed of 'N' words in all documents, where W_{ij} is the specific word.

In this study, LDA with bigram and trigram features and bag-of-words or TF-IDF representation was employed for topic modeling. LDA models were utilized to analyze our dataset, exploring the topic structure and underlying themes in the corpus. The number of topics ranged from 2 to 30, derived from similar studies [8,10]. In LDA, the assignment of hyperparameters to the Dirichlet distributions governing document-topic and topic-word relationships follows either a symmetric or asymmetric approach.

LDA models in topic modeling commonly utilize symmetric configurations by default. In such setups, the alpha and beta parameters play pivotal roles. Alpha determines document-topic density, with higher values leading to documents encompassing a broader range of topics and lower values indicating fewer topics within documents. Conversely, beta governs topic-word density, where higher values imply that topics consist of more words from the corpus, and lower values suggest fewer words per topic. In this study, alpha and beta were symmetrically set through uniform assignment. For the symmetric configuration, alpha was set to 0.1 to achieve similar topic proportions, and beta was set to 0.01 for uniform word distributions across topics.

In contrast, asymmetric distributions allow for customized configurations tailored to specific documents or topics. Higher alpha values result in more specific topic distributions per document, while higher beta values lead to more focused word distributions per topic. In our implementation, symmetric alpha and beta values were uniformly assigned as [0.01,0.03,0.1], and asymmetric beta was set to [0.01,0.02,0.03,0.01].

2.4.3 GSDMM

Additionally, GSDMM utilized bigram and trigram features along with a bag-of-words representation for the purpose of topic modeling. GSDMM is a short-text clustering model that essentially modifies LDA and assumes that a document 'd' in a corpus, consisting of 'D' documents (where d = 1,..., D), is generated by a mixture model and is only about one topic. The GSDMM algorithm is based on the "Movie Group Process" analogy, which explains its basic principle. In this analogy, a group of students represents a set of documents, each with their favorite movies as words. The students are randomly assigned to K tables, and the professor instructs them to shuffle tables with two goals in mind: first, to find a table with more students, and second, to choose a table where their movie interests align with those at the table. This process is repeated until a plateau is reached where the number of clusters remains constant.

The important factors affecting clustering are setting the hyperparameter values of α and β. α controls the cluster creation process, specifically, the number of tables that are removed when they become empty, indirectly affecting the number of clusters formed. A higher value of α means that fewer tables will be removed, resulting in a smaller number of clusters with larger sizes. In contrast, a lower value of α means that more tables will be removed, leading to a larger number of smaller clusters. β determines the degree to which a data point is assigned to a cluster based on the similarity between the data point and the cluster. A lower value of beta results in a greater tendency for a data point to be assigned to a cluster that is more similar to it, rather than to a more popular cluster. Conversely, a higher value of beta leads to a greater tendency for a data point to be assigned to a more popular cluster, regardless of its similarity to that cluster. In this study, the GSDMM were applied to the dataset by setting the hyperparameters $\alpha = 0.1$ and $\beta = 0.1$ as recommended in [11] and the number of iterations for GSDMM was set to 40 due to the algorithm's efficiency in convergence, as mentioned in [2]. The topic range (from 2 to 30 topics) was defined similarly to the previously used LDA model.

2.4.4 BERT-LDA-K-Means

BERT is a pre-trained language model that uses deep learning techniques to understand the meaning and context of words in natural language text. Identifying topics using bag-of-words information (LDA) is effective when texts are coherent and contain frequent words. However, when texts are incoherent in terms of word choice or sentence meaning, additional contextual information is required to comprehensively capture the intended meaning. While K-means clustering is a simple and efficient unsupervised machine learning algorithm that can partition a dataset into K clusters based on similarities between data points, its performance depends on several factors, such as the choice of K, initialization of cluster centroids, and distance metric used to measure the similarity between data points. The algorithm assumes that the clusters have spherical shapes and similar sizes, which may not always be the case in real-world datasets with complex shapes and non-uniform sizes.

To address these limitations, integrating LDA, BERT, and K-means clustering in a combination of bag-of-words and contextual information can preserve semantic details and facilitate the creation of contextual topic identification. This approach allows for the identification of topics in a more

comprehensive and nuanced manner, overcoming the constraints associated with using bag-of-words or K-means clustering alone [12,13].

To conduct the experiments, several techniques were applied to identify topics in the corpus. Firstly, LDA was used to assign a probabilistic topic vector to each document. Secondly, BERT was employed to generate sentence embedding vectors, capturing the meaning and context of the text. The LDA and BERT vectors were then concatenated with a weight hyperparameter, balancing the relative importance of information from each source. Next, an autoencoder was utilized to learn a lower-dimensional latent space representation of the concatenated vector, assuming that the concatenated vector would have a manifold shape in the high-dimensional space. Finally, K-means clustering was performed on the latent space representations to obtain the topics. Overall, this approach combines the strengths of LDA and BERT while also leveraging the power of autoencoders and clustering to identify topics in the corpus.

2.4.5 Propose Topic Modeling Models

In this research, different techniques with various feature representations and parameter settings are explored to identify the most effective approach for the dataset. The models used in this experiment include both feature-based and hybrid models. The Feature-Based models consist of NMF, LDA, and GSDMM models with bigram and trigram Bag-of-Words and TF-IDF feature vectors, respectively. These models are named as follows: NMF-Bi-BOW, NMF-Tri-BOW, NMF-Bi-TFIDF, NMF-Tri-TFIDF, LDA-Bi-BOW, LDA-Tri-BOW, LDA-Bi-TFIDF, LDA-Tri-TFIDF, GSDMM-Bi-BOW, and GSDMM-Tri-BOW. In addition, the Hybrid models include BERT-LDA-K-means. The aim is to determine the optimal technique for extracting topics from the dataset by comparing the results of these models.

2.5 Topic Modeling Evaluation Metrics

Topic coherence measures are used to evaluate the quality of topics by quantifying the degree of semantic similarity among the top-scoring words within a topic [34]. These metrics aid in distinguishing between topics that are semantically meaningful and those that are simple statistical artifacts. Four distinct coherence measures, namely, C_v, C_npmi, C_umass, and C_uci, were employed to assess the coherence of topics based on different criteria. These measures provide unique ways to evaluate topic coherence. Here, we provide a brief overview of the different coherence measures and how they are calculated [9]: C_v uses a sliding window approach with a one-set segmentation of the top words, along with normalized pointwise mutual information (NPMI) and cosine similarity as indirect confirmation measures. C_uci utilizes a sliding window approach and calculates the pointwise mutual information (PMI) of all the word pairs within the given top words. C_umass uses document co-occurrence counts and a one-preceding segmentation along with logarithmic conditional probability as the confirmation measure. Finally, C_npmi is an improved version of C_uci coherence that uses NPMI instead of PMI. Comparing the results from different coherence measures to provide a more comprehensive evaluation of the coherence of topics can help in selecting the most appropriate measure for a particular dataset. As each coherence measure evaluates coherence based on different criteria, choosing the most suitable measure for a specific task is crucial.

2.6 Model Selection

To select the appropriate model, two factors were taken into consideration: topic coherence scores for each model and number of topics. The aim was to identify models with high coherence scores across

all four metrics while ensuring that the number of topics was not excessive. Having too many topics can lead to overfitting, making interpretation difficult for humans.

For each set of generated topics, the top 15 words from each topic were presented to five human raters with expertise in tweet-related terminology. They were instructed to rate the coherence and interpretability of each topic on a scale of 1–5, where 1 represented the lowest and 5 the highest rating. Average rating were calculated for each topic across all raters. Subsequently, Cohen's Kappa was employed to assess the obtained scores for inter-rater agreement [9]. This measure quantifies agreement beyond chance for two or multiple raters by comparing observed agreement to chance agreement, using the formula:

$$K = \frac{P_0 - P_e}{1 - P_e} \tag{2}$$

where:

- P_0 is the observed agreement.
- P_e is the expected agreement by chance.

This statistic extends to assess agreement among multiple parties, providing a reliable measure of consistency in evaluations. The model and the corresponding number of topics that received the highest average ratings were used to cluster and label the dataset.

2.7 Data Splitting

The classification process involved training the models using a labeled dataset. The labeled dataset was divided into three independent sets: a training set, a testing set, and a validation set, with proportions of 70%, 15%, and 15%, respectively. The training set was utilized for model training, while the validation set was employed to prevent bias during hyperparameter tuning. Finally, the test set was reserved for unbiased evaluation of the final models. An examination of the dataset reveals an imbalanced distribution, as shown in Fig. 4. To address this imbalance, a subset of the dataset was randomly sampled for experimentation using Stratified Sampling. This technique is used to obtain representative samples from a population by dividing them into homogeneous subcategories known as strata and then randomly sampling the data from each stratum. Stratified Sampling reduces bias in sample selection while preserving the proportional distribution of the training and testing datasets from the original dataset.

2.8 Text Classification Techniques

The objective of the experiment was to identify a suitable model for multiclass classification in this task. Several models were leveraged in the experiment to compare their performance, selected from those mentioned in the introduction, using traditional machine learning models with a strong track record in text classification as a baseline. Additionally, deep neural networks were selected for performance comparison [19]. The traditional models employed in this study included MNB, LSVC, LSVM, and LR. These models were implemented using the scikit-learn library, with MNB utilizing the MultinomialNB implementation, LSVC employing LinearSVC, and LSVM and LR constructed using SGDClassifier. To enhance their performance, various feature extraction techniques were explored, including the use of a counter vectorizer with different n-grams and TF-IDF representations [20,26]. The hyperparameters for each model were fine-tuned using GridSearchCV [35] to obtain optimal configurations. For further analysis, DNN models, including LSTM, BiLSTM, a variant of the LSTM architecture, and a hybrid CNN+BiLSTM [21], were used. Finally, their performance was compared

with that of BERT, a transformer-based neural network. To control the training process, prevent overfitting, and enhance the model's generalization performance, early stopping was experimented with [17]. Further details of the DNN models are provided below:

2.8.1 LSTM

The LSTM model is structured with four layers [36–38]. The first layer is an embedding layer for the matrix of word vectors, followed by an LSTM layer with 128 LSTM units. Two dense layers are incorporated into the model: the first dense layer, consisting of a feed-forward neural network with 64 units, interprets the LSTM output, and the second dense layer, comprising 16 units, is responsible for producing a final output for the classification of 16 classes. The activation function used in the first dense layer is ReLU, while the final dense layer employs Softmax for multiclass classification.

2.8.2 BiLSTM

The BiLSTM model consists of two LSTM layers, one processing the sequence in the forward direction, and the other processing it in the backward direction [36–38]. In addition, a four-layer BiLSTM model was implemented [38] with the following architecture: an embedding layer for the matrix of word vectors, a BiLSTM layer featuring 128 BiLSTM units, and dropout set at 0.25 to prevent overfitting. Two dense layers were used, with the first serving as a feed-forward neural network with 64 units to interpret the LSTM output, and the second dense layer, consisting of 16 units, was responsible for producing the final output for the classification of 16 classes. The first dense layer utilized the ReLU activation function, whereas the Softmax activation function was used in the final dense layer for multiclass classification.

2.8.3 CNN-BiLSTM

A combination of CNN and BiLSTM is used for multiclass classification. First, a CNN is used to extract features from the corpus by passing the data through the CNN layers to learn essential features from the input. The CNN output consists of a sequence of feature vectors. In the second step, the sequence of feature vectors obtained from the CNN is fed into the BiLSTM layer [20], which learns both features and their temporal dependencies. Finally, the output of the BiLSTM layer is forwarded through a fully connected layer with a softmax activation function to perform multiclass classification. The CNN-BiLSTM model is structured with five layers, commencing with an embedding layer that handles a matrix of word vectors. This is followed by a convolutional layer with a kernel size ranging from $[3 \times 3]$ to $[5 \times 5]$. Subsequently, a BiLSTM layer with 128 BiLSTM units is employed. Two dense layers are incorporated into the model, with the first being a feed-forward neural network featuring 64 units, responsible for interpreting the BiLSTM output. The second dense layer consists of 16 units and produced the final output for classifying 16 classes. The ReLU activation function is used in the CNN layer and the first dense layer, while Softmax activation is employed in the final dense layer for multiclass classification.

2.8.4 BERT

Among the various methods available for multiclass text classification in English, the bert-base-uncased model of BERT, which is a pretrained model specifically designed for this task, was utilized. The model underwent fine-tuned using the labeled dataset to adapt it to a specific classification task. The fine-tuning process began with the pre-processing the text documents by tokenization, which involves splitting text into individual tokens and converting them into numerical representations suitable for the input of the BERT model. For this purpose, the bert-base-uncased model was

employed. Following pre-processing, the BERT model was trained on the preprocessed dataset using a cross-entropy loss function, known for its suitability in multiclass classification tasks. The cross-entropy loss measures the dissimilarity between predicted and true labels. To update the model parameters during training, an AdamW optimizer with various learning rates was utilized to optimal performance [17]. In the training process, a batch size ranging from 32 to 64 was employed. The batch size determines the number of samples processed during each iteration of the training algorithm. The fine-tuning approach involving BERT, the use of cross-entropy loss, AdamW optimizer, and the optimization of batch size collectively contributed to the effective training of the model, ensuring accurate multiclass text classification.

2.9 Regularization Technique

Early stopping is employed as a regularization technique in models to mitigate overfitting and improve generalization performance. It entails monitoring the model's performance on a separate validation dataset during training and terminating the training process when the model's performance plateaus or begins to decline. By employing early stopping, the goal was to prevent overfitting by discontinuing the training before the model becomes excessively specialized for the training data, which could lead to suboptimal performance on unseen data [17,39].

To implement early stopping for LSVM and LR, the early stopping variable was activated in accordance with the specifications provided in Table 1. The stopping criterion was based on validation scores. However, MNB and LSVC models lack predefined stopping criteria. Consequently, exhaustive training and evaluation of the models were conducted to consider all the hyperparameter combinations within the defined grid search space. A portion of the training data was set aside to create a validation dataset for DNN models. After each training epoch, the performance of the model was assessed by monitoring the validation loss. The training process was stopped when the control metrics of the validation set indicated a potential performance decline.

Table 1: The grid search space for the hyperparameters

Model	Hyperameters and grid search space
MNB [22]	ngram_range: [(1, 1), (1, 2), (1, 3)], use_idf: [True, False], fit_prior: [True, False], alpha: [1e-2, 1e-1, 1e0, 1e1]
LSVC [22]	ngram_range: [(1, 1), (1, 2), (1, 3)], use_idf: [True, False], loss: ['hinge', 'squared_hinge'], penalty: ['l2'], multi_class: ['ovr', 'crammer_singer'], fit_intercept: [True, False], random_state: [42], max_iter: [900,1000,1100]
LSVM [22]	ngram_range: [(1, 1), (1, 2), (1, 3)], use_idf: [True, False], loss: ['hinge'], penalty: ['l2', 'l1'], alpha: [1e-5, 1e-4, 1e-3, 1e-2], early_stopping: [True], max_iter: [1000, 1500], random_state: [42]
LR [22]	ngram_range: [(1, 1), (1, 2), (1, 3)], use_idf: [True, False], loss: ['log'], penalty: ['l2', 'l1'], alpha: [1e-5, 1e-4, 1e-3, 1e-2], early_stopping: [True], max_iter: [1000, 1500], random_state: [42]

(Continued)

Table 1 (continued)

Model	Hyperameters and grid search space
LSTM [17,18,36,38]	input_lenght: 250, learning_rate: [1e-5, 1e-4, 1e-3, 1e-2, 1e-1], optimizer: Adam,batch_size: [32, 64], epochs: 8, loss: categorical_crossentropy
BiLSTM [21]	input_lenght: 250, learning_rate: [1e-5, 1e-4, 1e-3, 1e-2, 1e-1], optimizer: Adam, batch_size: [32, 64], epochs: 8, loss: categorical_crossentropy, dropout: 0.25
CNN-BiLSTM [19,21,28,30]	kernel_size: [3x3, 5x5], input_lenght: 250, learning_rate: [1e-5, 1e-4, 1e-3, 1e-2, 1e-1], optimizer: Adam,batch_size: [32, 64], epochs: 8, loss: categorical_crossentropy, dropout: 0.25
BERT [16,40]	lr: [3e-5, 5e-5, 1e-4, 3e-4], eps: 1e-8, max_lenght: 64, optimizer: AdamW, batch_size: [32, 64], epochs: 8

2.10 Hyperparameter Tuning for Text Classification Models

To ensure optimal model results, we employed GridSearchCV [35,39] for hyperparameter tuning. GridSearchCV systematically explores a user-specified parameter grid to identify the best hyperparameter combination for a given model. It generates and evaluates a set of hyperparameters by training the model with each combination on the training data and assessing performance on a validation set.

The output of GridSearchCV provides the hyperparameter combination that maximizes the model's performance. To conduct hyperparameter tuning, a range of values for each hyperparameter was defined, as detailed in Table 1.

Various approaches specific to each model were used to perform hyperparameter tuning. For MNB, LSVM and LR models, different learning rates were tested on a logarithmic scale, starting from the default values recommended by scikit-learn. In the case of LSTM-based models, the Adam optimizer was employed and experimented with various learning rates, following guidance from Keras, on a logarithmic scale.

The AdamW optimizer was employed for the BERT model, and different learning rates were tested based on the original BERT model by Devlin et al. [40]. To evaluate the performance of each model, they were trained using each learning rate. The performances of the models were assessed using a validation set. The optimal learning rate was selected using the early stopping technique, which allowed us to monitor model performance and stop training when no significant improvement was observed.

2.11 Text Classification Evaluation Metrics

The text classifiers used in the experiment were evaluated with four evaluation metrics: Precision, Recall, Accuracy, and Weighted Average F1-Score. Precision is the ratio of true positives (TP) to the total number of predicted positives (TP + FP), measuring the proportion of actual positives among the instances predicted as positive. Recall is the ratio of true positives (TP) to the total number of actual positives (TP + FN), measuring the proportion of actual positives that were correctly identified by the model. Accuracy is the ratio of the total number of correctly classified instances (TP + TN) to the total number of instances. The Weighted Average F1-Score is a variation of the F1-Score, calculated as the harmonic mean of precision and recall, which addresses the class imbalance in the dataset by

assigning a weight to each class based on the number of instances in that class. Here are the equations for Accuracy, Precision, Recall, and Weighted Average F1-Score:

$$Accuracy = \frac{TP + TN}{TP + FN + TN + FP} \tag{3}$$

$$Precision = \frac{TP}{TP + FP} \tag{4}$$

$$Recall = \frac{TP}{TP + FN} \tag{5}$$

$$F1 - Score = \frac{2\,Precision + Recall}{Precision + Recall} \tag{6}$$

$$Weighted\,F1 - Score = \sum_{i=1}^{N} w_i \cdot F1 - Score_i \tag{7}$$

where F1-Score$_i$ represents the F1-Score of the i-th class, and w_i is the weight assigned to that class.

The F1-Score is employed in multiclass classification tasks due to its effectiveness in balancing precision and recall, particularly in cases of imbalanced class distributions. It calculates the harmonic mean of the precision and recall. To address the dataset's imbalance, a weighted average F-Score was chosen as the evaluation metric in the experiment.

2.12 Hardware and Software Utilized in the Experiments

To perform tweet pre-processing and implement machine learning methodologies, a Jupyter notebook utilizing Python 3.6 was employed. The computational tasks were executed on a system with an Intel(R) Core(TM) i5-1135G7 @ 2.40 GHz processor, 6.00 GB of RAM, and an NVIDIA GeForce RTX 3060 6 GB graphics card.

3 Results and Discussion

3.1 Topic Modeling

In the context of topic modeling experiments, our study trained and compared the performance of NMF, LDA, GSDMM, and BERT-LDA-K-means models. These models were deployed with varied word embedding and feature extraction techniques, trained across a range of topics (k) from 2 to 30. The evaluation of model quality was conducted using four metrics, and the results are depicted in Fig. 3.

The results presented in Fig. 3 encapsulate a comprehensive assessment of text clustering models, scrutinizing multiple facets to unravel their performance nuances. Initially, exploring diverse word embeddings and feature extraction methods across these models yielded noteworthy observations. Subsequently, the evaluation of NMF, LDA, GSDMM, and BERT-LDA-K-means models across distinct coherence scores contributed significant insights into their coherence and capacity for generating topics. Moreover, a comparative analysis based on varying topic counts delved into the scalability and interpretability of these models. Leveraging Cohen's Kappa scores enabled an understanding of optimal topic numbers, emphasizing inter-rater agreement. Additionally, an investigation into running costs shed light on the comparable performance among text clustering models.

The exploration of various word embeddings and feature extraction methods across different models provided insightful observations:

- NMF-Bi-BOW and NMF-Tri-BOW: Show a consistent increase in coherences up to a certain number of topics, implying their ability to capture meaningful relations in bigram and trigram word embeddings using Bag-of-Words representation. However, the coherence tends to stabilize or decrease slightly as the number of topics increases.
- NMF-Bi-TFIDF and NMF-Tri-TFIDF: Display varied performance. Bi-TFIDF shows relatively higher and more consistent coherences compared to Tri-TFIDF across different topic counts, indicating that TF-IDF might assist in capturing more distinctive features in bigram over trigram.
- LDA-Bi-BOW and LDA-Tri-BOW: Demonstrate an increasing trend in coherence scores with an increase in the number of topics, particularly with trigram embeddings, suggesting a more coherent representation of topics.
- LDA-Bi-TFIDF and LDA-Tri-TFIDF: Both display a mixed performance, showing fluctuations in coherences across different topic counts. However, TF-IDF based bigrams seem to offer better stability and slightly higher coherence scores.
- GSDMM-Bi-BOW and GSDMM-Tri-BOW: Exhibit moderate performance, with fluctuations in coherence scores across different numbers of topics. Both bigram and trigram representations via BOW show similar coherence trends.

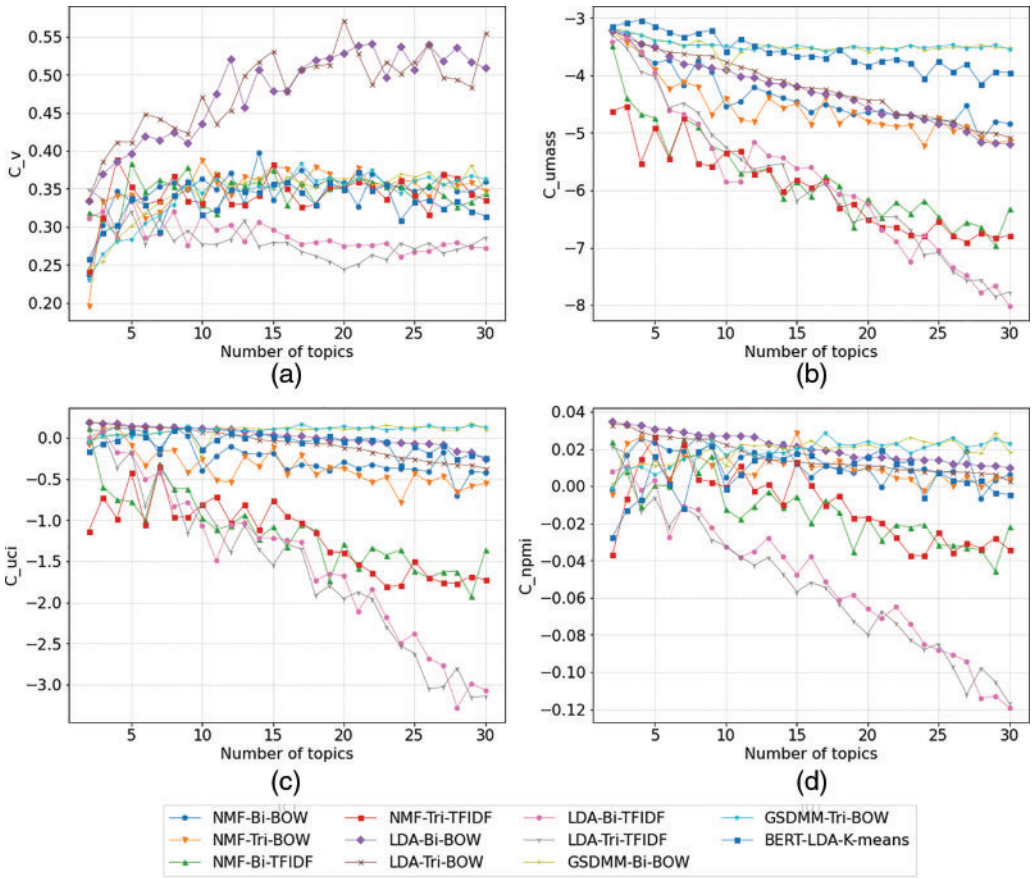

Figure 3: Comparison of topic coherence scores for topic clustering models

In summary, BOW consistently outperforms TF-IDF, particularly with trigram representations, showcasing enhanced stability and higher coherence scores. While trigram generally offer richer context, their effectiveness varies across models, with bigram displaying more consistent coherence scores in certain scenarios.

The assessment of NMF, LDA, GSDMM, and BERT-LDA_KMeans across four coherence scores (C_v, C_umass, C_uci, and C_npmi) offers valuable insights into these models' performance regarding topic coherence.

NMF, utilizing NMF, demonstrated moderate C_v coherence, suggesting reasonable word co-occurrence within topics. Its relatively high C_umass score indicated good topical coherence based on document co-occurrence. The model also performed well in C_uci, highlighting its ability to create coherent topics in terms of document co-occurrence and exclusivity. However, the comparatively lower C_npmi score hinted at a discrepancy in evaluating exclusivity using normalized pointwise mutual information.

LDA, employing LDA, displayed robust performance across C_v coherence and C_umass score, indicating strong word co-occurrence within topics and coherence based on document co-occurrence, respectively. Similar to NMF, it excelled in C_uci, signifying its capacity to generate exclusive and coherent topics. Yet, akin to NMF, LDA showcased a slightly lower C_npmi score, indicating potential disparities in assessing exclusivity through normalized pointwise mutual information.

GSDMM, employing GSDMM, exhibited a moderate C_v coherence and a relatively high C_umass score, suggesting reasonable word co-occurrence within topics and good topical coherence based on document co-occurrence, respectively. The model also demonstrated reasonable performance in C_uci, indicating its ability to create coherent and exclusive topics. However, like NMF and LDA, its C_npmi score was relatively lower, hinting at a potential challenge in evaluating exclusivity.

BERT-LDA-K-means displayed robust C_v coherence and C_umass scores, suggesting strong word co-occurrence within topics and coherence based on document co-occurrence. It performed well in C_uci, emphasizing its capacity to create exclusive and coherent topics. Nevertheless, akin to other models, it showed a slightly lower C_npmi score, potentially indicating challenges in assessing exclusivity through normalized pointwise mutual information.

In summary, while all models showcased strengths across different coherence scores, discrepancies in assessing exclusivity through C_npmi were apparent, suggesting a limitation in evaluating topic coherence solely through normalized pointwise mutual information. Notably, NMF and GSDMM exhibited more variability across coherence scores compared to LDA and BERT-LDA_KMeans. LDA and BERT-LDA-K-means demonstrated consistent strong performance across C_v, C_umass, and C_uci, highlighting their effectiveness in generating coherent topics across various coherence metrics. This comprehensive analysis underscores the importance of a balanced evaluation approach considering multiple coherence metrics to thoroughly assess topic quality.

A comparison among NMF, LDA, GSDMM, and BERT-LDA-K-means based on the number of topics sheds light on their scalability and interpretability in generating topics.

NMF, utilizing NMF, demonstrated consistent yet diminishing performance as the number of topics increased. While it exhibited reasonable coherence and interpretability with fewer topics, maintaining quality became a challenge with the expansion of topics.

In contrast, LDA displayed robust scalability, maintaining high coherence and exclusivity across various topic numbers. Its ability to retain quality topics even with higher topic counts underscored its stability in generating meaningful and interpretable topics.

GSDMM, employing GSDMM, showcased moderate scalability. While maintaining reasonable coherence with fewer topics, it struggled to sustain quality and coherence as the number of topics increased, resembling the challenges observed in NMF.

BERT-LDA-K-means emerged as a model with remarkable scalability, maintaining high coherence and exclusivity across various topic counts. Similar to LDA, it consistently generated interpretable topics across a wide range of topic numbers, making it a robust choice in terms of scalability and interpretability.

Both LDA and BERT-LDA-K-means outperformed NMF and GSDMM in scalability and maintaining coherence across a broader range of topic numbers. NMF and GSDMM faced challenges in maintaining coherence and interpretability as the number of topics increased, indicating limitations in scalability for these models. The ability of BERT-LDA-K-means to sustain high-quality topics across various topic counts stands out, emphasizing the significance of considering scalability and interpretability when selecting a topic modeling approach.

The subsequent step involved selecting a model and determining the optimal number of topics for human interpretability. Model selection was based on identifying the number of topics that performed well across all four metrics, within a range of 5–20 topics to avoid overfitting and ensure comprehensibility. Each model, along with its respective high-coherence topic counts and top 20 relevant terms, underwent human rating for selection. Cohen's Kappa scores were then used to finalize the choice of models with the most optimal topic numbers. Table 2 displays these chosen models with their respective topic counts.

Table 2: Cohen's Kappa for judge's agreement on text clustering models and respective topics

Model	Topics generated	Kappa
LDA-Tri-BOW	15	0.82
LDA-Bi-BOW	12	0.79
BERT-LDA-K-means	12	0.72
LDA-Tri-BOW	10	0.70
BERT-LDA-K-means	9	0.65
LDA-Tri-BOW	6	0.63
BERT-LDA-K-means	11	0.60

Table 2 displays the evaluation of Cohen's Kappa scores, shedding light on determining the optimal number of topics for text clustering. Observing the Kappa scores across various models and their respective topic numbers reveals valuable insights. For instance, higher Kappa scores, such as 0.82 for LDA-Tri-BOW with 15 topics and 0.79 for LDA-Bi-BOW with 12 topics, signify robust inter-rater agreement. This emphasizes the significance of selecting topic numbers that foster consensus among raters, ensuring more coherent clusters.

However, contrasting scores across different topic numbers within the same model (e.g., LDA-Tri-BOW at 15 topics *vs.* 10 topics) indicate a nuanced relationship. While increased topic numbers might enhance agreement initially, a higher number of topics may lead to decreased consensus among raters, as seen with lower Kappa scores for smaller topic counts.

The variability in Kappa scores for models such as BERT-LDA-K-means further highlights the sensitivity of models to the number of topics generated. This underscores the importance of meticulous

topic number selection, aiming to strike a balance between coherence, interpretability, and inter-rater reliability.

Based on the highest rating score obtained, the LDA-Tri-BOW model with 15 topics was selected as a suitable model for the experiment. The model was used to assign the best label to the given topics. The method takes one parameter, 'threshold,' which is the minimum probability required for a label to be considered the best label for a given tweet. If the probability of the top-scoring topic is greater than or equal to the threshold, it is returned as the best label for the tweet. Otherwise, the tweet is labeled as "other". After labeling, the entire dataset can be distributed according to the amount for each topic, as shown in Fig. 4.

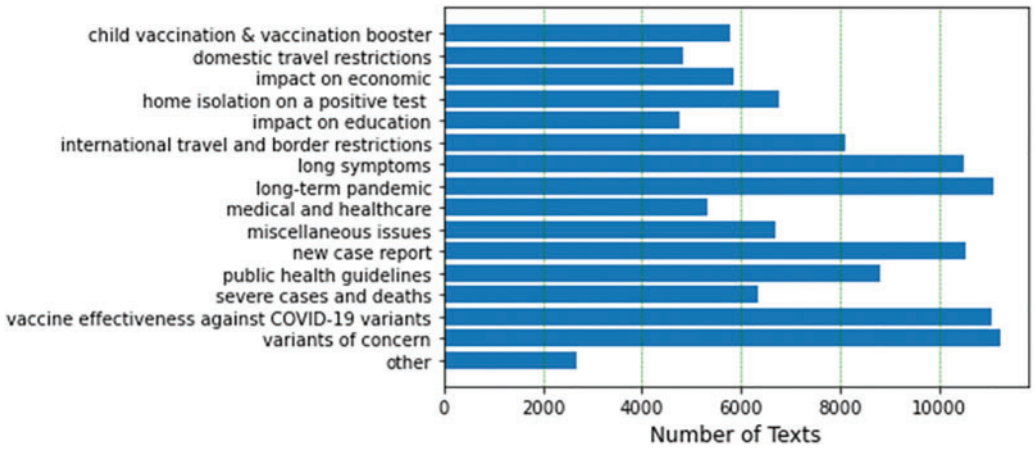

Figure 4: Displays the distribution of COVID-19-related tweets in Thailand and its neighboring regions on November 01, 2021, and July 07, 2022

Finally, the training time for models was a crucial consideration in conducting the experiment. Table 3 illustrates the time consumption of the clustering models utilized in this study.

Table 3 presents the evaluation of average iteration training times across diverse topic modeling algorithms, unveiling valuable insights:

- Bigram *vs.* Trigram: Average execution times per iteration consistently indicated similar computational costs for both bigram and trigram representations within NMF, LDA, and GSDMM models. Minimal variance across these models underscored their comparable computational demands.
- Bag-of-Words (BOW) *vs.* TF-IDF: TF-IDF representation showed marginally higher average execution times per iteration across NMF, LDA, and GSDMM variations, implying a slightly heavier computational load compared to BOW in these topic modeling algorithms.
- Model Performance Insights: Traditional models like NMF, LDA, and GSDMM demonstrated moderate to high computational efficiency, with NMF displaying lower execution times per iteration than LDA and GSDMM across various representations. However, the BERT-LDA-K-means model exhibited substantially higher computational costs, highlighting its resource-intensive nature compared to traditional models.

In summary, this analysis emphasizes comparable performance between bigram and trigram representations, slight computational disparities between BOW and TF-IDF, and varying computational demands among NMF, LDA, GSDMM, and the resource-intensive BERT-LDA-K-means model.

Table 3: Average iteration execution times (in seconds) for topic clustering models

Model	Model variation	Average execution time per iteration
NMF-based models [7–10]	NMF-Bi-BOW	70.32 s/it
	NMF-Tri-BOW	71.11 s/it
	NMF-Bi-TFIDF	96.53 s/it
	NMF-Tri-TFIDF	98.04 s/it
LDA-based models [7–10]	LDA-Bi-BOW	187.32 s/it
	LDA-Tri-BOW	244.75 s/it
	LDA-Bi-TFIDF	290.05 s/it
	LDA-Tri-TFIDF	310.15 s/it
GSDMM-based models [11]	GSDMM-Bi-BOW	998.36 s/it
	GSDMM-Tri-BOW	1102.08 s/it
BERT-LDA-based models [12–14]	BERT-LDA-K-means	7381.82 s/it

The proposed clustering models face several challenges, including predefining the number of clusters and ensuring uniform cluster sizes. Hierarchical clustering has emerged as a viable solution, eliminating the need for predetermined cluster numbers. In future research, the aim will be to address the difficulty in distinguishing certain topics using hierarchical clustering. This approach involves leveraging topic modeling to identify key topics within the corpus and clustering tweet texts based on the similarity of their topic distributions. One method entails representing each document as a topic distribution and utilizing these representations as inputs for clustering algorithms.

3.2 Text Classification

The process of text classification starts with training traditional models by tuning their hyperparameters to obtain the best parameters for each model, as shown in Table 4. Subsequently, these models are trained on the dataset, and their performance is evaluated. The experimental results are illustrated in Fig. 5.

As shown in Fig. 5, the LSVC model outperforms the other models significantly, while the remaining three models exhibit similar performances. The subsequent step involved conducting experiments with LSTM, BiLSTM, CNN-LSTM, and BERT. The process commenced with hyperparameter tuning, as detailed in Table 1, which led to the identification of optimal hyperparameters for each DNN model, as presented in Table 5. The training process involved a gradual increase in the number of epochs while utilizing EarlyStopping [15], which monitors the validation loss during training and halts the process when no further improvements are observed, thus mitigating overfitting. It is worth noting that the validation loss rate stabilized after just five training epochs. The results of the experiments are shown in Figs. 6–8.

The learning curve for training and validation is shown in Fig. 6. As the model iteratively improved its fit to the training data, the training loss gradually decreased. A corresponding reduction in the validation loss was observed alongside the training loss. The training process continued until the validation loss reached a point of stabilization, as shown in Fig. 6a. At this juncture, the EarlyStopping

mechanism was employed to halt training prematurely and prevent overfitting. Throughout the training of all four models, performance evaluation was carried out by assessing accuracy and weighted F1-Score. Fig. 7 shows that BiLSTM achieves the highest classification performance when trained for five epochs.

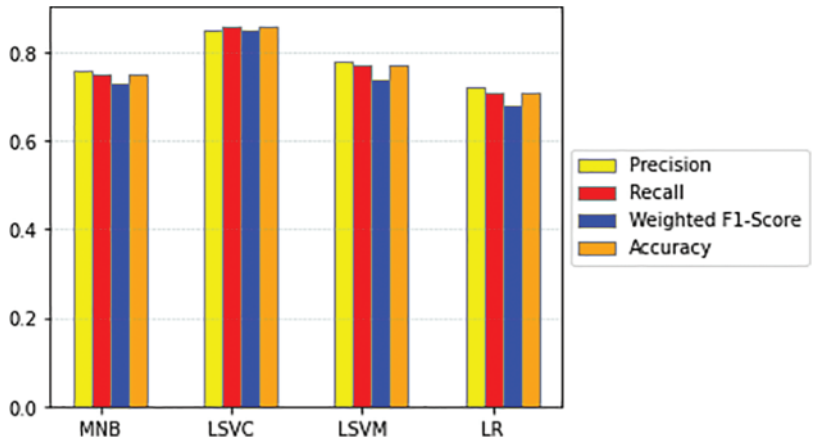

Figure 5: Comparison of precision, recall, weighted F1-Score and accuracy for MNB, LSVC, LSVM, and LR models

Table 4: Four text classification models and their optimal hyperparameters

Model	Hyperameters
MNB	Alpha: 0.5, fit_prior: False, use_idf : False, ngram_range : (1, 1)
LSVC	fit_intercept: False, loss: 'hinge', max_iter: 900, multi_class: 'crammer_singer', random_state: 42, use_idf: True, ngram_range : (1, 1)
LSVM	Alpha: 0.0001, early_stopping: True, loss: 'hinge', max_iter: 1000, penalty: 'l2', random_state: 42, use_idf: True, ngram_range: (1, 1)
LR	Alpha: 0.0001, early_stopping: True, loss: 'log', max_iter: 1000, penalty: 'l1', random_state: 42, use_idf: False, ngram_range: (1, 1)

Table 5: DNN algorithms and their corresponding optimal hyperparameters

Model	Hyperameters
LSTM	lr: 2e-3, batch_size: 64, epoch: 5
BiLSTM	lr: 2e-3, batch_size: 64, dropout: .25, epoch: 5
CNN-BiLSTM	filters: 32, kernel_size: 3, lr: 2e-3, batch_size: 64, dropout: .25, epoch:5
BERT	lr: 1e-5, eps:1e-8, max_length: 64, batch_size: 32, epoch: 5

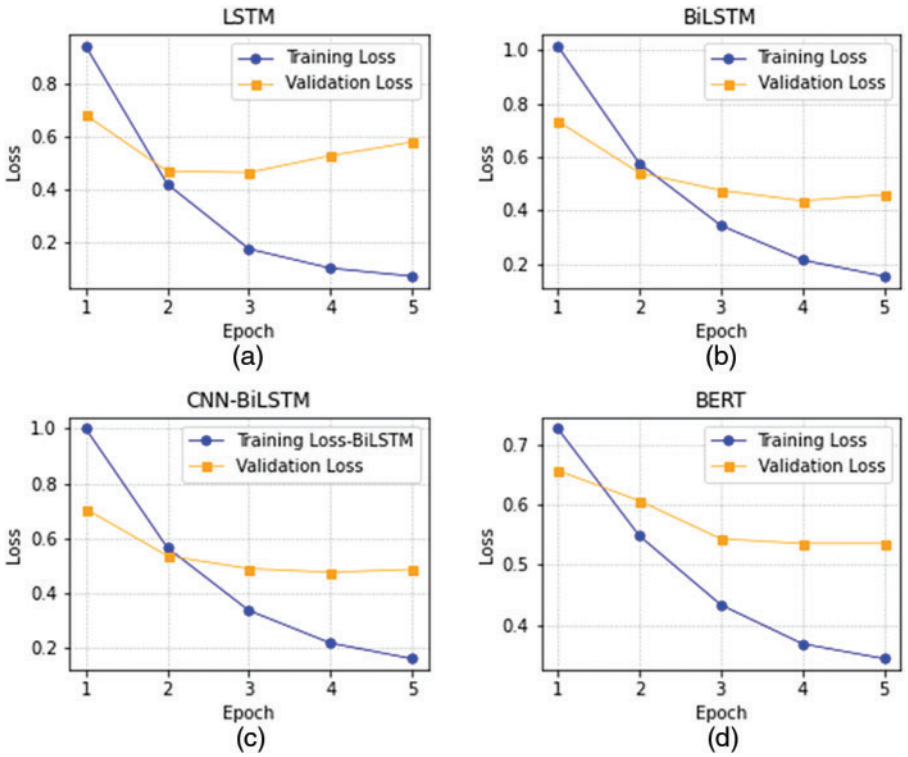

Figure 6: Comparison of training loss and validation loss for LSTM, BiLSTM, CNN-BiLSTM, and BERT models

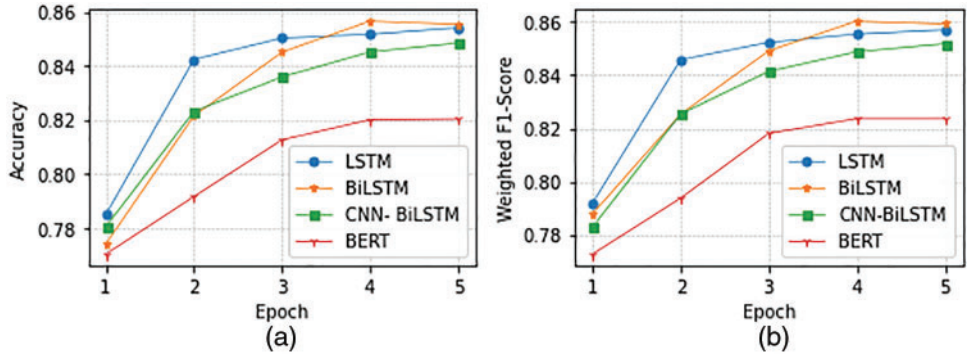

Figure 7: Comparison of accuracy & weighted F1-Score for LSTM, BiLSTM, CNN-BiLSTM, and BERT models

After conducting experiments with all the models presented, the accuracy and weighted F1-Score were compared to identify the best-performing model, as shown in Fig. 8.

We meticulously examined different models and configurations, comparing both traditional and deep learning methods. Our analysis indicated that the performance of these models depended on various factors, including the dataset's nature, feature extraction techniques, and hyperparameter settings. For traditional models, we found that the choice of feature extraction methods and the incorporation of IDF (Inverse Document Frequency) played a crucial role. Some models, like MNB

and LR, performed well without IDF, while others, such as LSVC and LSVM, showed improved performance with IDF. LSVC consistently outperformed other traditional models, demonstrating its reliability in our context.

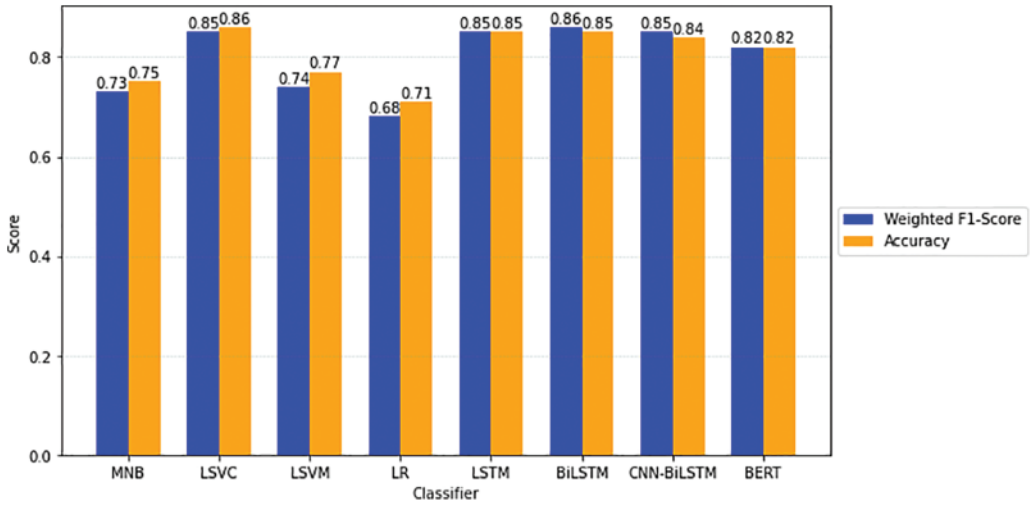

Figure 8: Comparison of accuracy & weighted F1-Score for MNB, LSVC, LSVM, LR, LSTM, BiLSTM, CNN-BiLSTM, and BERT models

In the case of DNN models, we experimented with different architectures, learning rates, kernel sizes, batch sizes, and epochs. Surprisingly, while DNN models, in general, exhibited better performance, LSVC showed inconsistent results. Notably, BiLSTM outperformed our expectations among the DNN models, highlighting its effectiveness in capturing complex patterns within the data. However, the performance variations observed across models can be attributed to the intricacies of the dataset, model architectures, and hyperparameter settings. Our intention is to conduct further experiments with diverse architectures like RoBERTa or XLNet, and explore ensemble methods to enhance the model's overall performance. These ongoing investigations will provide a more comprehensive understanding of the observed results and ensure the robustness of our conclusions.

This study provides valuable insights into public health and its related industries. By employing advanced Natural Language Processing (NLP) techniques, the proposed models effectively analyzed COVID-19-related discussions on social media platforms such as Twitter. These models offer real-time monitoring capabilities, enabling swift identification of emerging trends and potential outbreaks. Industries, especially healthcare and crisis management, can utilize models for proactive decision-making and timely interventions. Moreover, this research contributes to enhancing the efficiency of public health surveillance by harnessing social media data and ensuring a rapid response to evolving health situations. The adaptable nature of our models makes them applicable not only to COVID-19, but also to future disease outbreaks, empowering industries to stay ahead in managing health crises. This research bridges the gap between technology and public health, facilitating informed actions and enhancing overall societal well-being.

4 Conclusion

This study demonstrates the effectiveness of machine learning and deep learning models in topic extraction and classification from a corpus of COVID-19 discourse on Twitter. The experiment used a

dataset collected during the Omicron outbreak and addressed two primary objectives: topic extraction and text classification, with the aim of identifying the most suitable model for the dataset.

In the first part of the experiment, we compared the performance of various models, including LDA, NMF, GSDMM, and BERT-LDA-K-means, for topic extraction from the dataset. The second part involved comparing the performance of the DNN models and traditional ML models for text classification, using the labeled dataset obtained from the previous topic extraction experiment. The results of the experiment highlight the effectiveness of the LDA and BERT-LDA-K-means algorithm in constructing a topic-modeling model for extracting COVID-19-related topics from the Twitter data. Particularly in this specific task, by utilizing LDA with trigram and BOW features, the model achieved satisfactory performance that aligned well with human understanding and interpretation.

In terms of text classification, the LSVC and BiLSTM models demonstrated superior performance, closely matching other DNN models. These findings suggest that the proposed models can be applied as preventive tools for monitoring and tracking the COVID-19 pandemic through social media platforms. While the overall assessment indicates superior performance of models within the DNN group compared to traditional models, it remains speculation whether these models would exhibit equally fitting performance when applied to specific tasks. This observation prompts further investigation to ascertain their suitability and effectiveness in specialized domains. It serves as a directional guideline for future research endeavors in this field.

Acknowledgement: The authors acknowledge the contribution and the support of the Department of Information Technology and Computer Science at Lampang Rajabhat University (LPRU).

Funding Statement: The authors received no specific funding for this study.

Author Contributions: The authors confirm contribution to the paper as follows: study conception and design: Pakorn Santakij; data collection: Pakorn Santakij, Samai Srisuay; analysis and interpretation of results: Pakorn Santakij, Samai Srisuay, Pongporn Punpeng; draft manuscript preparation: Pakorn Santakij, Pongporn Punpeng. All authors reviewed the results and approved the final version of the manuscript.

Availability of Data and Materials: Due to the nature of this research, participants of this study did not agree to have their data shared publicly; therefore, supporting data is not available.

Conflicts of Interest: The authors declare that they have no conflicts of interest to report regarding the present study.

References

[1] K. M. Ridhwan and C. A. Hargreaves, "Leveraging twitter data to understand public sentiment for the COVID-19 outbreak in Singapore," *Int. J. Inf. Manag. Data Insights*, vol. 1, no. 2, pp. 100021, 2021. doi: 10.1016/j.jjimei.2021.100021.

[2] S. Ghaffary, "Omicron is outpacing delta on social media, too," 2021. Vox. [Online]. Available: https://www.vox.com/recode/22850677/omicron-delta-variant-covid-social-media (accessed on 05/01/2022).

[3] K. Verspoor *et al.*, "Introduction to the 1st workshop on natural language processing for COVID-19 at ACL 2020," in *Proc. ACL*, 2020.

[4] M. M. Agüero-Torales, D. Vilares, and A. G. López-Herrera, "Discovering topics in twitter about the COVID-19 outbreak in Spain," *Procesamiento de Lenguaje Natural*, vol. 66, pp. 177–190, 2021.

[5] L. Leng, J. Zhang, G. Chen, M. K. Khan, and K. Alghathbar, "Two-directional two-dimensional random projection and its variations for face and palmprint recognition," in *Proc. ICCSA*, Santander, Spain, 2011, pp. 458–470.
[6] L. Leng, S. Zhang, X. Bi, and M. K. Khan, "Two-dimensional cancelable biometric scheme," in *2012 Int. Conf. Wavelet Anal. Pattern Recognit. (ICWAPR)*, Xi'an, China, 2012, pp. 164–169.
[7] R. Albalawi, T. H. Yeap, and M. Benyoucef, "Using topic modeling methods for short-text data: A comparative analysis," *Front. Artif. Intell.*, vol. 3, pp. 42, 2020. doi: 10.3389/frai.2020.00042.
[8] H. M. Alash and G. A. Al-Sultany, "Improve topic modeling algorithms based on twitter hashtags," *J. Phys. Conf. Ser.*, vol. 1660, no. 1, pp. 12100, 2020. doi: 10.1088/1742-6596/1660/1/012100.
[9] S. L. Zoya, F. Shafait, and R. Latif, "Analyzing LDA and NMF topic models for Urdu tweets via automatic labeling," *IEEE Access*, vol. 9, pp. 127531–127547, 2021. doi: 10.1109/ACCESS.2021.3112620.
[10] S. Mifrah and E. H. Benlahmar, "Topic modeling coherence: A comparative study between LDA and NMF models using COVID'19 corpus," *Int. J. Adv. Trends Comput. Sci. Eng.*, vol. 9, no. 4, pp. 5756–5761, 2020.
[11] C. Weisser *et al.*, "Pseudo-document simulation for comparing LDA, GSDMM and GPM topic models on short and sparse text using twitter data," *Comput. Stat.*, vol. 38, no. 2, pp. 647–674, 2023. doi: 10.1007/s00180-022-01246-z.
[12] A. Subakt, H. Murf, and N. Hariadi, "The performance of BERT as data representation of text clustering," *J. Big Data*, vol. 9, no. 1, pp. 1–21, 2022. doi: 10.1186/s40537-022-00564-9.
[13] E. Atagün, B. Hartoka, and A. Albayrak, "Topic modeling using LDA and BERT techniques: Teknofest example," in *Proc. UBMK*, Ankara, Turkey, 2021, pp. 660–664.
[14] K. Sethia, M. Saxena, M. Goyal, and R. K. Yadav, "Framework for topic modeling using BERT, LDA and K-Means," in *Proc. ICACITE*, Greater Noida, India, 2022, pp. 2204–2208.
[15] J. Lande, A. Pillay, and R. Chandra, "Deep learning for COVID-19 topic modelling via twitter: Alpha, delta and omicron," 2023. arXiv preprint arXiv:2303.00135.
[16] E. C. Garrido-Merchan, R. Gozalo-Brizuela, and S. Gonzalez-Carvajal, "Comparing BERT against traditional machine learning model in text classification," *J. Comput. Cogn.Eng.*, vol. 2, no. 4, pp. 352–356, 2023. doi: 10.47852/bonviewJCCE3202838.
[17] J. A. Benítez-Andrades, Á. González-Jiménez, Á. López-Brea, J. Aveleira-Mata, J. M. Alija-Pérez and M. T. García-Ordás, "Detecting racism and xenophobia using deep learning models on twitter data: CNN, LSTM and BERT," *PeerJ Comput. Sci.*, vol. 8, pp. e906, 2022. doi: 10.7717/peerj-cs.906.
[18] A. Ezen-Can, "A comparison of LSTM and BERT for small corpus," arXiv preprint arXiv:2009.05451, 2020.
[19] N. Mohd, H. Singhdev, and D. Upadhyay, "Text classfication using CNN and CNN-LSTM," *Webology*, vol. 18, no. 4, pp. 2440–2446, 2021.
[20] A. Kulkarni, A. Hengle, and R. Udyawar, "An attention ensemble approach for efficient text classification of Indian languages," in *Proc. ICON*, Patna, India, 2020, pp. 40–46.
[21] F. Sun and N. Chu, "Text sentiment analysis based on CNN-BiLSTM-attention model," in *Proc. ICRIS*, Sanya, China, 2020, pp. 749–752.
[22] Y. A. Alhaj, A. Dahou, M. A. A. Al-qaness, and L. Abualigah, "A novel text classification technique using improved particle swarm optimization: A case study of Arabic language," *Future Internet*, vol. 14, no. 7, pp. 194, 2022. doi: 10.3390/fi14070194.
[23] R. Egger and J. Yu, "A topic modeling comparison between LDA, NMF, Top2Vec, and BERTopic to demystify twitter posts," *Front. Sociol.*, vol. 7, pp. 886498, 2022.
[24] Twitter, "Developer agreement and policy," *Twitter*, 2021. [Online]. Available: https://developer.twitter.com/en/developer-terms/agreement-and-policy (accessed on 06/10/2021).
[25] H. Lyu, L. Chen, Y. Wang, and J. Luo, "Sense and sensibility: Characterizing social media users regarding the use of controversial terms for COVID-19," *IEEE Trans. Big Data*, vol. 7, no. 6, pp. 952–960, 2021. doi: 10.1109/TBDATA.2020.2996401.
[26] Y. Didi, A. Walha, and A. Wali, "COVID-19 tweets classification based on a hybrid word embedding method," *Big Data Cogn. Comput.*, vol. 6, no. 2, pp. 58, 2022. doi: 10.3390/bdcc6020058.

[27] K. Svensson and J. Blad, "Exploring NMF and LDA topic models of Swedish news articles," M.S. thesis, Uppsala University, Sweden, 2020.
[28] A. M. Alayba and V. Palade, "Leveraging Arabic sentiment classification using an enhanced CNN-LSTM approach and effective Arabic text preparation," *J. King Saud Univ.-Comput. Inf. Sci.*, vol. 34, no. 10, pp. 9710–9722, 2022. doi: 10.1016/j.jksuci.2021.12.004.
[29] S. Amin, M. I. Uddin, D. H. alSaeed, A. Khan, and M. Adnan, "Early detection of seasonal outbreaks from twitter data using machine learning approaches," *Complex*, vol. 2021, pp. 5520366, 2021. doi: 10.1155/2021/5520366.
[30] H. Zhou, "Research of text classification based on TF-IDF and CNN-LSTM," *J. Phys. Conf. Ser.*, vol. 2171, pp. 12021, 2022. doi: 10.1088/1742-6596/2171/1/012021.
[31] Y. Chen, H. Zhang, R. Liu, Z. Ye, and J. Lin, "Experimental explorations on short text topic mining between LDA and NMF based schemes," *Knowl. Based Syst.*, vol. 163, pp. 1–13, 2019. doi: 10.1016/j.knosys.2018.08.011.
[32] R. Egger, "Topic modelling: Modelling hidden semantic structures in textual data," *Applied Data Science in Tourism. Tourism on the Verge*, Springer, Cham. pp. 375–403, 2022. doi: 10.1007/978-3-030-88389-8_18.
[33] D. M. Blei, A. Y. Ng, and M. I. Jordan, "Latent Dirichlet allocation," *J. Mach. Learn. Res.*, vol. 3, pp. 993–1022, 2003.
[34] S. H. Mohammed and S. Al-augby, "LSA & LDA topic modeling classification: Comparison study on e-books," *Indones. J. Electr. Eng. Comput. Sci.*, vol. 19, no. 1, pp. 353–362, 2020. doi: 10.11591/ijeecs.v19.i1.pp%25p.
[35] M. Ahmed, M. S. Hossain, R. U. Islam, and K. Andersson, "Explainable text classification model for COVID-19 fake news detection," *J. Internet Serv. Inf. Secur.*, vol. 12, no. 2, pp. 51–69, 2022. doi: 10.22667/JISIS.2022.05.31.051.
[36] Z. Cui, R. Ke, Z. Pu, and Y. Wang, "Stacked bidirectional and unidirectional LSTM recurrent neural network for forecasting network-wide traffic state with missing values," *Transp. Res. Part C: Emerg. Technol.*, vol. 118, pp. 102674, 2020. doi: 10.1016/j.trc.2020.102674.
[37] M. A. Haq, A. K. Jilani, and P. Prabu, "Deep learning based modeling of groundwater storage change," *Comput. Mater. Contin.*, vol. 70, no. 3, pp. 4599–4617, 2022. doi: 10.32604/cmc.2022.020495.
[38] M. A. Haq, "CDLSTM: A novel model for climate change forecasting," *Comput. Mater. Contin.*, vol. 71, no. 2, pp. 2363–2381, 2022. doi: 10.32604/cmc.2022.023059.
[39] M. A. Haq and M. A. R. Khan, "DNNBoT: Deep neural network-based botnet detection and classification," *Comput. Mater. Contin.*, vol. 71, no. 1, pp. 1729–1750, 2022. doi: 10.32604/cmc.2022.020938.
[40] J. Devlin, M. W. Chang, K. Lee, and K. Toutanova, "BERT: Pre-training of deep bidirectional transformers for language understanding," in *Proc. NAACL-HLT*, Minnesota, USA, 2019, pp. 4171–4186.