Towards Accurate SDG Research Categorization: A Hybrid Deep Learning Approach Using Scopus Metadata

Jalal Sadoon Hameed Al-Bayati [1,*], Furat Nidhal Tawfeeq [1], and Mohammed Al-Shammaa [2]

[1] Website Division, University of Baghdad, Baghdad, Iraq
[2] Department of Computer Engineering, College of Engineering, University of Baghdad, Baghdad, Iraq
Email: Jalal.hameed@uobaghdad.edu.iq (J.S.H.A.B.); Furat@bccru.uobaghdad.edu.iq (F.N.T.); M.alshammaa@coeng.uobaghdad.edu.iq (M.A.S.)
*Corresponding author

***Abstract*—The complexity and variety of language included in policy and academic documents make the automatic classification of research papers based on the United Nations Sustainable Development Goals (SDGs) somewhat difficult. Using both pre-trained and contextual word embeddings to increase semantic understanding, this study presents a complete deep learning pipeline combining Bidirectional Long Short-Term Memory (BiLSTM) and Convolutional Neural Network (CNN) architectures which aims primarily to improve the comprehensibility and accuracy of SDG text classification, thereby enabling more effective policy monitoring and research evaluation. Successful document representation via Global Vector (GloVe), Bidirectional Encoder Representations from Transformers (BERT), and FastText embeddings follows our approach, which comprises exhaustive preprocessing operations including stemming, stopword deletion, and ways to address class imbalance. Training and evaluation of the hybrid BiLSTM-CNN model on several benchmark datasets, including SDG-labeled corpora and relevant external datasets like GoEmotion and Ohsumed, help provide a complete assessment of the model's generalizability. Moreover, this study utilizes zero-shot prompt-based categorization using GPT-3.5/4 and Flan-T5, thereby providing a comprehensive benchmark against current approaches and doing comparative tests using leading models such as Robustly Optimized BERT Pretraining Approach (RoBERTa) and Decoding-enhanced BERT with Disentangled Attention (DeBERTa). Experimental results show that the proposed hybrid model achieves competitive performance due to contextual embeddings, which greatly improve classification accuracy. The study explains model decision processes and improves openness using interpretability techniques, including SHapley Additive exPlanations (SHAP) analysis and attention visualization. These results emphasize the need to incorporate rapid engineering techniques alongside deep learning architectures for effective and interpretable SDG text categorization. With possible effects on more general uses in policy analysis and scientific literature mining, this work offers a scalable and transparent solution for automating the evaluation of SDG research.**

***Keywords*—text classification, Sustainable Development Goals (SDGs), deep learning, hybrid bidirectional Long Short-Term Memory-Convolutional Neural Network (LSTM-CNN), Global Vector (GloVe) embeddings**

I. Introduction

Text categorization involves the automatic allocation of predefined categories to unstructured text data. Within the context of the Sustainable Development Goals (SDGs) established by the United Nations, precise and efficient text classification is of highest importance. The SDGs consist of 17 associated goals serving as a framework to deal with social, economic, and environmental challenges. There is a pressing need for effective tools that can systematically organize, categorize, and assess large volumes of textual data, as the volume of research output related to these goals continues to grow rapidly across various scientific disciplines [1].

Automated text classification systems improve SDG activities by facilitating the rapid identification and mapping of research papers, policy documents, and reports to their relevant SDG categories [2]. This enhances the ability to identify and analyze knowledge gaps and emerging trends within the global sustainability agenda, while also supporting evidence-based decision-making for policymakers and stakeholders. The necessity for robust methodologies powered by artificial intelligence is underscored by the inadequacy of traditional human classification methods to manage the scale and intricacy of contemporary scientific literature [3].

This study aims to develop and evaluate a comprehensive deep learning pipeline for the automated classification of research articles, utilizing metadata obtained from the Scopus database in relation to the Sustainable Development Goals [4]. This study aims to achieve the following specific objectives:

An end-to-end data processing and modeling pipeline has been developed to align with SDG text categorization, encompassing data collection, preprocessing, model creation, training, evaluation, and prediction. In comparison to established baseline models such as Long Short-Term Memory (LSTM), BiLSTM, CNN, and BERT, the objective is to develop and benchmark a hybrid

Manuscript received May 9, 2025; revised July 3, 2025; accepted July 25, 2025; published November 21, 2025.

deep learning model that integrates Convolutional Neural Networks (CNNs) with Bidirectional Long Short-Term Memory (BiLSTM) layers, utilizing pre-trained GloVe embeddings [5]. The objectives are mentioned in the following pointText categorization involves the automatic allocation of predefined categories to unstructured text data. Within the context of the Sustainable Development Goals (SDGs) established by the United Nations, precise and efficient text classification is of the highest importance. The SDGs consist of 17 associated goals serving as a framework to deal with social, economic, and environmental challenges. There is a pressing need for effective tools that can systematically organize, categorize, and assess large volumes of textual data, as the volume of research output related to these goals continues to grow rapidly across various scientific disciplines [1].

Automated text classification systems improve SDG activities by facilitating the rapid identification and mapping of research papers, policy documents, and reports to their relevant SDG categories [2]. This enhances the ability to identify and analyze knowledge gaps and emerging trends within the global sustainability agenda while also supporting evidence-based decision-making for policymakers and stakeholders. The necessity for robust methodologies powered by artificial intelligence is underscored by the inadequacy of traditional human classification methods to manage the scale and intricacy of contemporary scientific literature [3].

This study aims to develop and evaluate a comprehensive deep learning pipeline for the automated classification of research articles, utilizing metadata obtained from the Scopus database in relation to the Sustainable Development Goals [4]. This study aims to achieve the following specific objectives: An end-to-end data processing and modeling pipeline has been developed to align with SDG text categorization, encompassing data collection, preprocessing, model creation, training, evaluation, and prediction. In comparison to established baseline models such as LSTM, BiLSTM, CNN, and BERT, the objective is to develop and benchmark a hybrid deep learning model that integrates Convolutional Neural Networks (CNNs) with Bidirectional Long Short-Term Memory (BiLSTM) layers, utilizing pre-trained GloVe embeddings [5]. The objectives are mentioned in the following points:

- Utilizing multiple datasets and various hardware setups, and creating assessments procedures by using measures such as accuracy, specificity, and efficiency, one can effectively evaluate the capability and robustness of the proposed method
- Providing a scalable and adaptable methodological framework that facilitates the automated analysis and categorization of literature related to the SDGs, thereby supporting research, policy, and decision-making processes in sustainable development.

This work seeks to advance the current methodologies in SDG text classification while offering practical solutions for the automated organization and examination of sustainability research [5].

Natural Language Processing (NLP) includes an important procedure of text classification that involves assigning text to predetermined categories [6]. The procedure includes a wide range of applications, such as recognizing spam, subject categorization, book title classification, and more. The rapid growth of digital text data increased the demand for robust and reliable text classification problems [7]. In recent studies, Deep Learning (DL) considered an outstanding method for text classification, providing significant results compared to conventional machine learning methods [8]. While in our study, we concentrates on utilizing DL, especially via TensorFlow and Keras, to categorize elements related to the United Nations' SDGs. The SDGs contain 17 worldwide goals initiated by the United Nations in 2015 for addressing various problems in society, the economy, and the environment [9]. These objectives propose to achieve a healthier and more sustainable future for everyone through 2030 [10]. Due to the substantial amount of textual data produced about these objectives, there is an increasing requirement for computerized setups that are capable of accurately categorizing and evaluating such content [11]. Text categorization models serve a purpose in this context for categorizing research papers, reports, and articles according to their relevance to certain SDGs [12].

The primary objective of this study is to develop a considerable route for text classification with DL methodologies. The objective includes importing data and text preprocessing, building models, training, testing, and prediction. Our study intends to show the ability of DL models in properly categorizing text into relevant SDG categories by implementing this method on SDG-related data. The goal of this effort is to establish a comprehensive framework that can possibly be customized for similar text categorization challenges across different domains. Text classification plays an essential role in the organization and handling of massive text data sets. Within the mechanisms of the SDGs, it supports scholars, policy makers, and businesses in quickly recognizing and giving priority to important information. Classifying articles, statements, and publications according to the SDGs helps for better decision-making processes and the allocation of resources. Additionally, automated text classification can considerably minimize the time and effort associated with manual sorting, which leads to increased effectiveness and productivity. Traditional text classification techniques, like Support Vector Machines (SVMs) and Random Forests, depend on manually gathered features and usually require much preprocessing [13]. Although these methods have proved to have good outcomes for many different scenarios, they have limitations in their capacity of recognizing complex data patterns [14]. Nonetheless, DL methods have advanced text classification by autonomous retrieving features from unstructured text input. Convolutional Neural Networks (CNNs) and Recurrent Neural Networks (RNNs) have shown exceptional effectiveness in numerous text classification tasks. Recently, attention-based models, such the Transformer and its variations, have further improved current

approaches by allowing parallel processing and recognizing long-term relationships in text data [15]. The text data associated with the SDGs is frequently inconsistent and unstructured, covering multiple domains and subjects. The distribution, even with each category, could show severe asymmetry, with certain SDGs appearing more frequently than others. These issues require the creation of durable procedures for preprocessing and model architecture suitable to handle these types of data complications [16]. So, it cannot be directly utilized in new files. Accordingly, the study proposed a text classification approach based on LSTM to solve the issue mentioned above [17]. The model's first stages create a separate class or category for each file in the data by dependent analysis, then capture the contextual information of words by LSTM to learn the importance of irrelevant neighboring nodes. Finally, the feature representation of all nodes is combined to create a semantic embedding of the text graph for label prediction [18]. The text classification has turned into a graph classification problem. The main contributions of this paper are as follows:

- Development of a unified deep learning approach: proposing a modular framework that integrates BiLSTM and CNN architectures with contextual embeddings (BERT/FastText), specifically optimized for SDG-related document classification.
- Incorporation of advanced preprocessing and class balancing: This includes stemming, stopword removal, and methods to address class imbalance such as oversampling and focal loss.
- Integration of interpretability methods: enhancing model transparency by incorporating attention visualization, SHapley Additive exPlanations (SHAP), and Local Interpretable Model-agnostic Explanations (LIME) to explain predictions, which is crucial for policy-influenced tasks.
- Comprehensive benchmarking against State of The Art (SOTA) models: Extensive experiments benchmark the proposed approach against SOTA models such as RoBERTa, DeBERTa, and large language models (GPT-3.5/4 [19], Flan-T5), using both SDG-specific and non-SDG datasets.
- Explicit analysis of hardware impact is conducted to analyze the trade-offs between Graphical Processing Units (GPU) and Central Processing Units (CPU) training/inference, providing practical insights for real-world deployment.

These contributions collectively advance the field of SDG text classification by presenting a robust, interpretable, and scalable solution, with thorough empirical validation and real-world applicability. The remainder of this paper is organized as follows: Section II reviews related work, including recent advances in deep learning and sentiment analysis, and situates our approach within the current literature. Section III details the methodology, encompassing data preprocessing, the proposed hybrid architecture, contextual embedding strategies, interpretability enhancements, and experimental setup. Section IV presents experiment configurations. The results including benchmarking against state-of-the-art models, analysis of hardware impact, and interpretability findings are shown in Section V. Section VI shows the discussion of the results. Section VII concludes the paper by summarizing key contributions and outlining potential directions for future research.

II. Related Work

The literature shows many remarkable advancements in text classification using various artificial intelligence models as numerous studies indicate that deep learning models including CNNs and RNNs such as LSTM and BiLSTM surpass conventional machine learning approaches due to their capacity to separately extract and represent complicated features from raw datasets [20]. There is continuous argument about the effectiveness of various deep learning architectures related to data type and distribution. Some studies show that CNNs exceed in capturing local patterns, whereas others such as RNN-based models are more adept at managing long-range dependencies in sequential data, especially in language tasks that require contextual prior understanding [21]. Transformer-based approaches such as BERT have established new performance in various NLP applications by effectively capturing bidirectional context. However, these models typically necessitate significant computational processing and huge datasets for training certainly constrain their practical applicability [22].

A further area of intersection involves the application of pre-trained word embeddings, for instance GloVe and word2vec to improve model performance. Most recent studies incorporate these embeddings; however, outcomes differ based on task complexity and the quality of domain-specific training data [23]. Several studies advocate hybrid or ensemble models that integrate CNNs, RNNs, or attention mechanisms, asserting enhanced accuracy and robustness. However, some researchers notify that more architectural complexity may result in extended training duration time and a probability of overfitting if sufficient regularization or optimization is not applied. Unreliability exists related to data diversity; some studies used multi-domain datasets whereas others focus on domain-specific resulting in incompatible conclusions about model generalizability [24].

While there is agreement on how DL is useful for text classification, the literature indicates continued experimentation related to the best model architecture, the trade-off between complexity and efficiency, and the influence of dataset properties. The identified contradictions emphasize the necessity for comprehensive, comparative methodologies, as proposed in this study, to systematically assess hybrid deep learning models utilizing large, diverse, and well-assembled datasets relevant to SDG classification [25].

Many comparisons have assessed the efficiency of many artificial intelligence models for text classification. A survey study conducted a quantitative analysis of various text classification approaches on familiar metrics,

providing facts about their effectiveness. Deep learning algorithms could automatically develop significant feature presentations from data via incremental learning, minimizing the requirement for manual feature extraction and achieving exceptional precision in tasks such as classification [26]. CNNs may take advantage of the translational consistency of data and local relationships, leading to their importance in image processing, computer vision, and NLP [27]. Kim [28] was the first to utilize CNN for text classification, providing a model that mixes static and dynamic lines of vocabulary into separate channels and utilizes multi convolutional kernels. The model was defined to create the ability for convolutional processes to continuously gather features at multiple scales, pooling operations for efficiently acquiring local text features, with its high computational capabilities [29]. RNN appears more suitable for NLP than CNN due to its dynamic capabilities and competence in dealing with variable-length inputs while examining dependency over time [30].

Nonetheless, RNNs encountered semantic bias concern, when words positioned towards the end of a sentence carry more significance compared to those at the beginning, therefore affecting the overall semantic accuracy of the paragraph. As a result, the LSTM model, presented by Hochreiter *et al.* [31], carefully ignores prior data in order to find a solution for the issues of expanding and gradient cost in RNNs. In the recent past, the attention procedure has inspired great interest within scientific society. The attention procedure emulates human awareness [32], enabling concentration on more significant elements and applicable across various NLP assignments. Nonetheless, models utilizing RNN and CNN mainly focus on word localization and lack in acquiring information between remote, non-contiguous words. Liu *et al.* [33] presented the attention diffusion procedure in Graph Neural Networks that contains contextual information from indirect neighbors within a single layer. Furthermore, node-level attention technology is applied to achieve a more error-free document level description. Jia *et al.* [34] introduced a more sophisticated network model that relied on a graph convolutional neural network that showed the encoding of a large syntactic dependency grammar trees, lead to multiple heads of attention to acquire dependencies from the text sequences. Jia *et al.* [34] determinately improved the text classification performance through the integration of capsule networks and semantics. Wang *et al.* [35] introduced an intuitive a classification model that utilizes a unidirectional Graph Convolutional Networks (GCN), operating without pre-trained word embeddings in scenarios with a constrained training dataset for message sharing. Yang *et al.* [36] introduced a hierarchical attention network by employing word and sentence level attention procedures to boost document classification performance. This method is highly effective at depicting hierarchical structures in textual content.

Li *et al.* [37] represented a word-sentence heterogeneous graph to improve interpretability by creating CoGraphNet. Howard and Ruder [38] created a deep learning model for NLP. ULMFiT attains superior results by refining already trained language model for a particular written classification test, demonstrating the effectiveness of this deep learning model in NLP. Devlin *et al.* [39] presented BERT in their research, a transformer-based model that succeeds in achieving high accuracy across given NLP scenarios. BERT's bidirectional training helps it to understand context from both directions, making it particularly effective for text classification. Yosinski *et al.* [40] examine the transferability of acquired features across many tasks. Yosinski *et al.* [40] interprets the effectiveness of transfer learning, an increasing methodology for modern text classification discipline. Liu *et al.* [41] made an enhanced to the BERT by optimizing pre-training methodologies. RoBERTa achieves superior performance across several NLP metrics, including text classification, by employing augmented data and extended learning durations.

Sun *et al.* [42] introduced a framework (ERNIE) that incorporates external information into BERT. ERNIE attains elevated accuracy in many NLP tasks, including text categorization, by taking advantage of structured knowledge during the pre-training phase. Qiu *et al.* [43] carried out a survey investigating various essential models including BERT, XLNet, and GPT, while determining their layouts, training approaches, and its effectiveness in NLP tasks, such as classification of texts. Brown *et al.* [44] presents GPT-3, a transformer model including 175 billion parameters. GPT-3 has impressive results in several natural language processing assignments, including text categorization, with minimal task-specific fine-tuning. Zhang *et al.* [45] proposed a model that directly analyzes raw text at the character level using CNNs. This model is especially advantageous for complex grammar or for processing noisy text in languages. Lample *et al.* [46] employed a framework that combines CNNs and LSTMs for named entity recognition. The solutions discussed related to text categorization challenges that involve the identification of local and interrelated sequential relationships.

Within the framework of SDGs, text classification has fast evolved as several DL and ML approaches have been proposed and investigated. Key contributions are reviewed in this part together with their characteristics, benefits, constraints, and motivation for the current studies. Several articles have implemented advanced text classification employing various architectures. For instance, Bai *et al.* [47] showed how convolutional layers might capture local textual characteristics by introducing CNNs for sequence recommendations. LSTM networks for learning long-term dependencies were proposed by Hochreiter and Schmidhuber [31], hence establishing the basis for RNN-based text categorization. Using transformer architecture to capture bidirectional context and setting new benchmarks in NLP, Devlin *et al.* [39] created BERT. Emphasizing transfer learning and fine-tuning for text classification applications, Howard and Ruder [38] presented ULMFiT. Reporting higher accuracy in text classification, Jang *et al.* [48] coupled Word2Vec, CNN, BiLSTM, and attention mechanisms. Emphasizing the need of hybrid architecture, Kamyab *et al.* [49] combined CNN, BiLSTM, and GloVe embeddings for

sentiment analysis. Using a CNN-BiLSTM Hybrid, Bhuiyan *et al.* [50] implemented a hybrid model using social media data. In their system showcase, Manning *et al.* [51] emphasized the model simplicity and robustness, which contributed to its common usage in both research and commercial NLP applications.

Deep learning models (CNN, LSTM, BERT) independently learn valuable characteristics from raw data, hence lowering dependency on human engineering. Models such as BiLSTM and BERT clearly capture sequence and context, hence enhancing classification accuracy in challenging or ambiguous materials. Pre-trained language models (BERT, ULMFiT) use knowledge from big data to enable good performance on limited labeled data by. Combining CNNs, RNNs, and attention mechanisms shows enhanced resilience and accuracy by combining local and sequential features [52].

Transformer-based models (BERT, RoBERTa) depend on big datasets and substantial computational resources for successful training. Unless regularization is properly applied, hybrid and deep architecture may be overfit, especially in cases with inadequate or imbalanced training data [53]. Many models are assessed on general datasets; their performance on domain-specific or emergent themes (e.g., SDGs) is unknown. Most studies use English or single-domain datasets, therefore restricting generalizability to multilingual or cross-disciplinary SDG research. Deep models, particularly ensembles or hybrids, can be difficult to grasp and thereby affect decision-making [54].

TABLE I. COMPARATIVE LITERATURE SUMMARY OF SOME TEXT CLASSIFICATION STUDIES

Authors /Reference	Model/Approach	Dataset	Aspect	Advantages	Limitations
Bai *et al.* [47]	LSTM with Spiking Neural P Systems (LSTM-SNP)	Three real-world datasets.	self-attention networks	Fast, effective for short texts	Limited context, not sequence-aware
Hochreiter and Schmidhuber [31]	LSTM	Various	Long-term dependency learning	Handles long sequences	Training complexity, vanishing gradient
Devlin *et al.* [39]	BERT (Transformer)	General Language Understanding Evaluation (GLUE), Stanford Question Answering Dataset (SQuAD)	Bidirectional context, pre-training	High accuracy, transfer learning	High resource needs, long training time
Howard and Ruder [38]	ULMFiT (Transfer Learning)	Internet Movie Database (IMDb), Attorney General (AG) News	Fine-tuning pretrained LM	Effective with small datasets	Underperform on highly domain-specific tasks
Kamyab *et al.* [49]	CNN + BiLSTM + GloVe	Twitter, Yelp	Hybrid, word embeddings	Robustness, improved accuracy	Increased model complexity
Jang *et al.* [48]	Word2vec + CNN + BiLSTM + Attention	News articles	Multi-layer hybrid	High classification accuracy	Risk of overfitting, interpretability
Bhuiyan *et al.* [50]	CNN-BiLSTM Hybrid	Social media	Hybrid, deep learning	Enhanced detection, flexibility	Computational overhead
Bai *et al.* [47]	LSTM-SNP	Three real-world datasets.	self-attention networks	Fast, effective for short texts	Limited context, not sequence-aware

Notwithstanding these developments, the automated categorization of SDG-related material still shows flaws. Many times, lacking robustness across several SDG areas, current models do not fully use the extensive, transdisciplinary metadata accessible in sources such as Scopus [55]. Customized to the SDG environment and able of effective training and inference, scalable, flexible pipelines combining the strengths of CNNs, BiLSTMs, and pre-trained embeddings are much needed. By suggesting a hybrid deep learning system that combines data processing, model construction, and evaluation utilizing large-scale Scopus information and comparing performance against state-of-the-art baselines, the present work fills in these voids [56]. Table I shows a comparative summary of some of the text classification studies, their advantages, disadvantages and limitations.

Recent years have witnessed significant progress in leveraging deep learning and transformer-based models for text classification across various domains, including sentiment analysis, fake news detection, and Sustainable Development Goal (SDG) mapping. Numerous studies have demonstrated the effectiveness of architecture such as CNNs, LSTMs, BERT, and their variants on both generic and domain-specific datasets [57]. For instance, Hernández *et al.* [58] and Zamir *et al.* [59] applied deep learning models for sentiment analysis related to social activities. Highlighting the potential of such techniques for real-time insights in crisis contexts. However, several critical research gaps remain unaddressed, which this work aims to clarify and highlight:

- Limited focus on SDG-specific classification: most existing studies concentrate on general-purpose text classification or sentiment analysis, with comparatively few works addressing the unique challenges of SDG-related document categorization. There is a lack of standardized, large-scale benchmarks and tailored pipelines for SDG classification.
- Insufficient model interpretability: while deep learning models have achieved high accuracy, their "black box" nature limits practical adoption in policy and research settings where transparency is crucial. Few prior works integrate state-of-the-art interpretability tools such as attention visualization, SHAP and LIME specifically for SDG text classification.
- Under-explored use of advanced contextual embeddings: many studies still rely on static word embeddings such as GloVe or Word2Vec, with

limited exploration of the impact of contextual and subword embeddings (e.g., BERT, FastText [19]) on SDG classification performance.

- Scarce benchmarking against the latest SOTA models and LLMs: systematic comparisons with recent state-of-the-art models like RoBERTa, DeBERTa, and large language models (GPT-3.5/4, Flan-T5) in the context of SDG classification remains rare. This gap hinders a clear understanding of the relative strengths and weaknesses of traditional and transformer-based approaches for this domain.
- Neglect of practical deployment considerations: few studies discuss the real-world implications of hardware choices (GPU vs. CPU), scalability, and computational cost for deploying deep learning models in SDG-related applications.
- Inadequate handling of data imbalance and noise: the challenges posed by imbalanced SDG datasets and noisy, heterogeneous metadata are often overlooked, with sparse use of advanced techniques such as Synthetic Minority Over-sampling Technique (SMOTE), focal loss, or robust preprocessing.

Deep learning's recent developments have greatly enhanced text classification performance. Still, there are difficulties with computing efficiency, dependability across domains, and interpretability. This work addresses the constraints of past methods and offers a scalable, high-performance solution for automated SDG document categorization, therefore building on the results of prior approaches by presenting a hybrid deep learning pipeline especially tuned for SDG-related research.

The DataCite Metadata Schema has evolved to describe research datasets, making it easier to find, access or even cite them in academic publications. Its structured approach supports interoperability between repositories and is in line with the Findable, Accessible, Interoperable, Reusable (FAIR) data movement's ideas about distributing and reusing data. In addition, ISO 23081 is another example of a full structure for managing metadata in records. It gives guidelines for how to create, keep, and check the quality of metadata in organizational settings [60].

The FAIR projects with data have encouraged the creation of more tools for checking and expanding the quality of metadata. A few examples of automatic evaluation tools and assessments that comply with FAIR rules are the FAIRshake toolkit and the FAIR Evaluator. These tools can help us to verify the comprehensiveness and reliability of metadata in an organized manner [61].

TABLE II. GLOSSARY OF SCIENTIFIC TERMS AND CONCEPTS

Term	Definition
Sustainable Development Goals (SDG)	Seventeen global objectives established by the United Nations to address social, economic, and environmental issues by 2030
Text Classification	The procedure of categorizing unstructured textual material into specified classifications utilizing techniques or algorithms
Deep Learning (DL)	A branch of artificial intelligence that uses deep neural networks to identify complex patterns.
Bidirectional Long Short-Term Memory (BiLSTM)	A variant of recurrent neural network that analyzes data in both forward and backward paths, effectively acquiring contextual information from both ends
Embedding/Word Embedding	A method for coding words as smaller, low-dimensional vectors that retain semantic importance
GloVe	A pre-trained word embedding model that produces vector representations derived from global keyword-word combination data is utilized as an input embedding layer in the hybrid model
Bidirectional Encoder Representations from Transformers (BERT)	A transformer-based language model pre-trained to acquire extensive contextual word representations. Utilized for evaluation and contextual embedded data
Convolutional Neural Network (CNN)	An architecture of neural networks designed to identify local patterns through convolutional filters, commonly applied to natural language processing and image analysis. Applied to identify local characteristics within text sequences
FastText	A word embedding method that integrates subword information (character n-gram) to more effectively address unusual words is utilized as an alternate embedding method in tests
Tokenization	The procedure of segmenting text into words, sentences, or other significant components (tokens) is utilized in preparing input data for embedding and modeling
Stemming	The procedure of reducing words to their simple forms (e.g., moving to "move") utilized during preprocessing to standardize text
Stopword Removal	The removal of prevalent terms (e.g., "the," "and") that have low semantic significance in textual analysis Integrated during preprocessing to standardize text
Class Imbalance	A situation in which certain categories include a substantially greater number of samples compared to others. Utilized oversampling and focal loss to increase model diversity
Oversampling	Method utilized to improve the quantity of samples in minority classes to neutralize the dataset, consequently addressing class imbalance in SDG classes
Focal Loss	Classifying text without specific instruction on those categories, frequently uses prompt-based large language models. Assessed utilizing GPT-3.5/4 and Flan-T5 for the SDG tasks
Zero-shot Classification	Prompt-based large language models are used to classify text without explicit guidance on specific categories. This classification examined the use of GPT-3.5/4 and Flan-T5 models for the SDG tasks
RoBERTa, DeBERTa	Modern transformer-based models for natural language processing tasks, boosting BERT employed as metrics for SDG text classification efficiency
SHapley Additive exPlanations (SHAP)	Method for explaining artificial intelligence model predictions by providing significance ratings to features, therefore boosting model interpretability and refining predictions
Attention Visualization	Technique for visualizing the specific components of an input (such as words in a sentence) that a model highlights during estimation, with the purpose of enhancing the visibility of model decisions
FAIR Principles	Data management guidelines: Findable, Accessible, Interoperable, Reusable, which refer to metadata quality and dataset preparation

Even with these improvements, not many studies systematically compare the effects of these metadata standards and quality tools on SDG-specific text classification tools. Adding these tools to the design and evaluation of AI applications could make SDG classification more reliable and useful in a wider range of fields. Research could benefit from these tools to show how metadata standards and FAIR-aligned quality tools can be used together and tested in large-scale and multi-scope classification problems [62]. The literature demonstrates a specific pattern in favor of the utilization of artificial intelligence models for text categorization, attributed to their ability to autonomously identify complex patterns and presentations from unprocessed input texts. The utilization of pre-trained embeddings and advanced architectures such as BiLSTM and CNN has significantly boosted the accuracy and efficiency of text classification techniques [50]. This research combines these enhancements to establish a comprehensive workflow for identifying text associated with the Sustainable Development Goals, addressing the particular difficulties specific to this field. Table II shows a glossary of our utilization of scientific terms through this study.

III. Materials and Methods

A. Data Preparation

This study's text data acquired from seventeen data files which contain 8713 research titles and 2,211,255 words, as shown in Table I, each research paper is connected to a specific SDG type. Each file has diverse papers pertinent to the individual Sustainable Development Goal, including titles, abstracts, keywords, and subjects. Our study also multiplies the data multiple times with random data to increase the scale of the data to comply with modern NLP standards. Every spreadsheet has been imported into a panda DataFrame, an additional column has been incorporated within each DataFrame to indicate the SDG category. The Data Frames are subsequently merged into a singular DataFrame. This integrated DataFrame enables the following stages of data preprocessing and model development. The title, abstract, keywords, and subjects are merged into a singular text box to form a single entry. This integrated text field is meant for tokenization and embedding. A label encoder is used for transforming these labels into numerical values suitable for artificial intelligence techniques in which the encoder gave a unique value to each SDG category, then our approach started by dividing the data into training and validation datasets with a random ratio based on 10-fold cross validation [63]. This ensures that the simulation obtains sufficient data for training while keeping a portion of dataset for assessing model's effectiveness. A sample of the data utilized during our approach and tests is shown in Tables III and IV. The data shown includes six rows: SDG type, ID, article title, abstract, keywords, and subjects related to the article's scope. The data preparation stage is crucial for ensuring that the input data is of high quality and consistency for NLP and AI applications. The preprocessing utilized the following:

TABLE III. Data Titles and Words Related to SDGs

SDG Type	Titles count	Words count
1	62	16038
2	344	87125
3	1596	446938
4	205	51730
5	104	28031
6	1111	267835
7	1338	301560
8	214	56719
9	214	56716
10	151	41229
11	634	161279
12	306	77353
13	213	54560
14	341	84705
15	223	49705
16	61	15771
17	1596	413961
Total	**8713**	**2211255**

TABLE IV. Data Sample

Category	Description
SDG Type	4 (Goal of Quality Education)
Title	Enhancement of Recommendation Engine Technique for Bug System Fixes [64]
Abstract	This study aims to develop a recommendation engine methodology to enhance the model's effectiveness and efficiency. The proposed model is commonly used to assign or propose a limited number of developers with the required skills and expertise to address and resolve a bug report. Managing collections within bug repositories is the responsibility of software engineers in addressing specific defects. Identifying the optimal allocation of personnel to activities is challenging when dealing with software defects, which necessitates a substantial workforce of developers
Keywords	bugs, fusion of intelligent optimization, artificial neural networks, machine and deep learning
Subject	Information Systems,Artificial Intelligence,Computer Engineering.

- Data Loading and Consolidation: loading seventeen data files, each one related to a different Sustainable Development Goal (SDG), into separate pandas DataFrames. Each one of the seventeen data files has a number of columns, as shown in Table IV, such as article title, abstract, keyword, and subject area. An extra column to each DataFrame is added to show the SDG category for each input. All DataFrames are put together into one big DataFrame for easy analysis in a later stage.
- Data Augmentation: using data augmentation and multiplication to deal with any data sparsity and to match the size needed by modern NLP models and to prevent overfitting and increase generalization.
- The title, abstract, keywords, and subject columns for each article (row) were merged into one text field in order to check whether all important text information is available for the operations of feature extraction, tokenization, and embedding.
- Normalization: The text field from the previous step went through noise reduction such as lowercasing, in which all text was changed so that it could be processed further. All punctuation,

symbols, and non-alphanumeric characters were removed.

- Removal of Stopwords: A standard stopword list was used to get rid of common words that don't add any meaning, such as "the" and "of".
- Whitespace Normalization: All whitespace characters that came one after the other were changed to single spaces.
- Stemming: Words were substituted to their base forms to make sure that all the different forms were the same (for example, "studies" to "study").
- Label Encoding: A label encoder twisted the SDG category into numerical labels. A different integer value was given to each SDG, which is usable for numerical AI applications.
- Tokenization and embedding: each row in the data file that contains the article (data) terms was turned into a string of tokens (words), and then the tokens were linked to embedding vectors by pre-trained embeddings (GloVe, FastText, and BERT) through the training stage.
- Before training, the dataset was randomly shuffled to remove any bias and make sure that the batches entered randomly into the model during the experiment.
- Training and validation subsets: the study used 10-fold cross-validation. One-fold is used for testing and nine folds for training, and each iteration lowers the chance of selection bias.

B. Tokenization, Padding, and Global Vector Word Embeddings

The textual data is tokenized via the Keras Tokenizer. This procedure transforms the text into integer sequences, with each integer representing a particular word in the text. The tokenizer analyzes the training data to construct a vocabulary of the most frequent phrases. Padding is utilized to keep a consistent input size due to variations in the lengths of input sequences. The sequences are extended to a maximum length with shorter sequences replaced by zeros and longer sequences truncated, this consistency is necessary for feeding data into the model [65]. Thereafter, our approach added the initial GloVe embeddings taken from a text file named (glove.6B.100d). These embeddings provide an adequate vector diagram for each word in the glossary, preserving the same meaning or connections among words. Then, an embedding matrix is constructed to associate terms in the tokenizer's vocabulary with their respective GloVe embeddings [66]. This matrix triggers the embedding layer in the model, enabling it to utilize pretrained word representations.

C. Model Architecture

1) BiLSTM

A hybrid one convolutional layer-Bidirectional Long Short-Term Memory (BiLSTM) model is utilized in our study to categorize the text in the acquired data. This network includes a few layers; the first one is the embedding layer consists of a pre-trained Global Vector (GloVe) embedding which transform input sequences of word indices into dense vectors [67]. The second layer is the bidirectional LSTM Layers which were bidirectional LSTM layers used to capture sequential dependencies in both directions. This boosts the ability of the model for understanding the context of the data. Next layer is the dropout Layers which are carried out subsequently to each LSTM layer to prevent overtraining or overfitting problem by disabling a certain amount of input units throughout training randomly [68]. Then a dense layer incorporating ReLU and L2 regularization was utilized to incorporate non-linearity and manage complexities. The last output layer, which is a dense layer utilizing SoftMax activation, provides a final probability distribution through the SDG categories [69]. The model was built along with the hybrid CNN layers with the optimizer and limit categorical entropy loss as parameters. The Adam optimizer was selected for its effectiveness and flexible learning rate. Sparse categorical cross-entropy is utilized since the labels are integers denoting distinct classes. Besides, the early halting was used to monitor validation loss during the training phase [70]. Training terminates if the validation fails to decrease over a certain value of iterations or epochs, and the most effective weights are restored. This minimizes the overfitting and confirms the model generalizes adequately to unexpected inputs.

2) Hybrid BiLSTM-CNN model

Our approach is implemented by designing a hybrid BiLSTM-CNN model by using CNN layers, namely a Conv1D layer followed by a MaxPooling1D layer, to identify local patterns within the text sequences. In the hybrid model architecture, the CNN layers are followed by Bidirectional LSTM layers to capture temporal dependencies. Subsequently, adding a dropout layer that elevates dropout rates to mitigate overfitting. Lastly, the model is run and trained using the same optimizer and early halting to guarantee stable training. The system is trained on padded sequences utilizing early halting and validation data. The training phase consists of several epochs during which the model gradually adjusts its weight to reduce the validation loss function [71].

Model structure

The embedding layer turns input text sequences of word indices into dense vector characterization. These vectors convey semantic values of the terms and are essential for LSTM layers to fully comprehend the overall surroundings of the text. The embedding layer is initialized using the basic GloVe embeddings. These embeddings provide a full representation of words obtained through their utilization in a large text corpus. Let X_i represent the input sequence of word indexes i, and E denote the embedding matrix. The embedding layer turns X into embedded representation V_i as seen in Eq. (1), E represents a matrix of shape which includes the vocabulary size and embedding dimension [72].

$$V_i = E[X_i] \tag{1}$$

BiLSTM layers represent a variant of RNNs capable of capturing long-term dependencies in sequential input. Bidirectional LSTMs extend this functionality by

analyzing the input sequence in both forward and reverse perspectives, so effectively twice the contextual information supplied to the model [73]. A Bidirectional LSTM consists of two LSTM layers. The initial layer is the forward LSTM layer which examines the input sequence all over and backward LSTM layer which analyzes the input sequence from all over from end to beginning. The outcomes of both layers are put together to yield the final outcome of the Bidirectional LSTM [74]. Let $h_t^{forward}$ and $h_t^{backward}$ depict the covered states at time stride t for both ward LSTMs. The output H_t of the Bidirectional LSTM is shown in Eq. (2).

$$H_t = [h_t^{forward}; h_t^{backward}] \quad (2)$$

This combination permits the model to process information from both previous and succeeding contexts, allowing an increased understanding of the sequence. Dropouts are employed to reduce overtraining problems by randomly ignoring a proportion of the input nodes during the phase of training. This procedure permits the proposed model to acquire robust properties or features that do not rely on certain nodes or neurons [75]. The proposed architecture utilizes a best dropout rate of 0.3, meaning that 30% of the input units are randomly deactivated at each training epoch. The main idea of using dense layer brings non-linearity into the model's architecture, authorizing it to acquire deeper depiction of the input. The function of the activation here, which is ReLU function was created to incorporate non-linearity while bypassing the vanishing gradient issue that sometimes should use another activation functions such as sigmoid. L2 regularization is utilized randomly as hypermeter combination in the dense layer to reduce overtraining/overfitting dilemma by placing costs on substantial weights [76]. This assists the model's acquisition of smaller and generally applicable weights. Choosing the variable input to the dense layer as H, with the weights and biases represented by W and b, respectively. The output function Z of dense layers with ReLU activation is provided in Eq. (3).

$$Z = ReLU\ (W \cdot H + b) \quad (3)$$

The result of the output layer provides an overall prediction probability for each SDG category. The SoftMax activation function implies that output values are inside the interval [0, 1], giving them comprehensible as probabilities [77]. Then, with the weights and biases represented as W_{out} and b_{out}, respectively. The output y is provided in Eq. (4).

$$y_i = \frac{e^{(W_{out} \cdot Z + b_{out})i}}{\sum_j e^{(W_{out} \cdot Z + b_{out})j}} \quad (4)$$

where y_i represents the probability of the i^{th} SDG category. The Bidirectional LSTM-CNN model framework was designed to effectively absorb contextual data in textual input. Utilizing pre-trained GloVe embeddings, bidirectional processing, dropout for regularization, and extensive layers with ReLU activation, the model effectively learns robust features for classifying text into SDG categories [49]. The SoftMax output layer generates accessible probability distributions for each category, allowing accurate predictions. Fig. 1 illustrates the suggested framework diagram for the experiments conducted in the study.

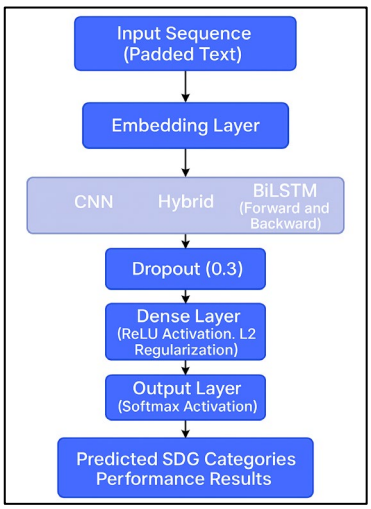

Fig. 1. Diagram of model structure.

IV. Experiments

A. Experimental Configuration

This article utilizes the Python 3.7 development platform than runs on Jupyter version 7.2.2, along with NumPy and other libraries, via the Anaconda prompt. The experiments are conducted on the development platform with two hardware configurations: the first features an 8-core Intel Core i7 numbered 11800H CPU, 16GB DDR4 RAM, and an Nvidia GeForce 3060 graphics processor. The second hardware includes a 4-core Intel Core i5 numbered 8350U CPU, 16GB DDR4 memory, and an Intel UHD 620 graphics processor. To assess the performance of the model we proposed by executing experiments on the subsequent sets: the first set is called Ohsumed which is extracted from the MEDLINE database. The Ohsumed dataset is a compilation of medical texts utilized for text classification in medical discipline [78]. It encompasses twenty-three disease categories and comprises 6286 training trials and 7643 test trials. The second set is GoEmotion which is a dataset of comments from Reddit [79], featuring twenty-seven emotion categories (excluding neutral). This study utilizes only single-label samples from the original multi-label dataset, in accordance with the methodology of Suresh and Ong [80]. The dataset has 21,402 training samples and 2971 test samples. Our model is examined against four baseline models, as indicated in Table V. The hardware configuration is shown in Table VI. The baseline models tested along with our proposed methodology are shown in the following:

TABLE V. Information Related to Each Model and Dataset

Model	Dataset	Categories	Field
LSTM, BiLSTM, BERT, CNN	GoEmotion	27	Sentiment
LSTM, BiLSTM, BERT, CNN	Ohsumed	23	Medline

TABLE VI. EQUIPMENT SETUP

CPU	TDP	GPU	Other Specifications
Intel Core i7 11800 H (8/16 Cores/Threads) with base frequency of 2.30 GHz (HW1)	45 Watt	Nvidia GeForce RTX 3060 (6GB dedicated plus 8GB shared GPU RAM)	16 GB DDR4 3200MHz NVMe SSD 3500MBps
Intel Core i5 8350 U (4/8 Cores/Threads) with base frequency of 1.70 GHz (HW2)	25 Watt	Intel UHD 620 (8GB shared GPU RAM)	16 GB DDR4 2400MHz NVMe SSD 3500MBps

Note: CPU: Central Processing Unit, TDP: Thermal Design Power, GPU: Graphics Processing Unit, RAM: Random Access Memory, SSD: Solid State Disk, NVMe: Nonvolatile Memory Express.

- LSTM: utilizes the final hidden state as an illustration of the entire data. The studies were employed by using previously trained word vectors [81].
- Bi-LSTM: a bidirectional model that utilizes previously trained word embeddings [48].
- BERT: builds a BERT network according to the configurations defined by the researchers in 2019. The BERT model's learning rate is determined at 2e−5, with a dropout rate of 0.1 [82].
- CNN: creates a CNN based on the structure created by a study in 2014 and used GloVe embeddings [83] as pretrained word representations. The best rate of learning is 0.001, the kernel sizes are 3, 4, and 5, and the total amount of kernels that are trained is one hundred using a best dropout rate of 0.5 [28].

B. Experimental Pseudocode

In Algorithm 1, the procedure of the study presented as the experimental pseudocode of the model.

Algorithm 1: Experiment Pseudocode

```
Start
Define a function `load_glove_embeddings(file_path)`
Initialize an empty dictionary `embeddings_index`
For each line in the file:
    Split the line into words and embedding values
    Convert embedding values to a NumPy array
    Store word and its embedding vector in `embeddings_index`
Return `embeddings_index`
Define a function `build_embedding_matrix(word_index, embeddings_index, embeddings_dim)`
Initialize `embedding_matrix` with zero values, shape `(len(word_index) + 1, embeddings_dim)`
For every word and index `[i]` in `word_index`:
    Get `embedding_vector` from `embeddings_index` for the word
    If `embedding_vector` is not Empty:
        Set `embedding_matrix[i]` to `embedding_vector`
Return `embedding_matrix`
Initialize an empty list `dfs`
For each `i` from `1 to 17`:
    Load data file into DataFrame `df`
    Add `sdg` column with value `i` to `df`
    Append `df` to `dfs`
Concatenate all DataFrames in `dfs` into a single DataFrame `df`
Create a new column `text` in `df` by combining `title`, `abstract`, `keywords`, and `subjects`
Initialize `Tokenizer` with `num_words=20000`
Fit `Tokenizer` on all texts in `df['text']`
Convert all texts to sequences
Pad all sequences to `max_length=200`
Define hyperparameter search space (e.g., dropout rates, L2 values, learning rates, batch sizes, number of units, etc.)
Set up `RandomizedSearchCV`-like process:
For each randomly sampled combination of hyperparameters:
    Initialize k-fold cross-validation (e.g., `StratifiedKFold` with k=5), stratified on SDG class
    For each fold:
        Split data into training and validation sets according to fold
        Build model with current hyperparameters:
            - Add `Embedding` layer (with GloVe weights, not trainable)
            - Add `Convolution1D` layer
            - Add `MaxPooling1D` layer
            - Add `Dropout` layer (use current sample's dropout rate)
            - Add `Bidirectional LSTM` layer (with/without return_sequences as needed)
            - Add additional `Dropout` layers as specified
            - Add `Dense` layers (with current sample's units, activation, L2 regularization, etc.)
            - Add output layer (`softmax`, number_of_classes)
        Compile model with current learning rate and optimizer
        Use `EarlyStopping` with validation accuracy monitored
        Fit the model on training fold, validate on validation fold
        Record performance metrics for this fold
    Calculate mean performance across all folds for this hyperparameter combination
Select the hyperparameter combination with the best average performance
Retrain the final model on the full training set using the best hyperparameters
Split-off a final held-out test set (if not already done before cross-validation), or use the cross-validation results as estimate of generalization performance
Evaluate the final model:
    - Use `model.evaluate` on the test set
    - Use `model.predict` for test entries
    - Calculate and display the `Confusion_Matrix`
    - Calculate metrics: accuracy, precision, recall, f1_score, mean_absolute_error, mean_squared_error
Use LIME to explain individual predictions by highlighting important words for each SDG class
Use SHAP to compute feature importances and visualize which parts of the input most influenced the model's decision
Predict SDGs based on new entry
Filter and sort predictions by threshold
Example Usage:
Define `input_title and input_keywords`
Call `classify_sdg(input_title, input_keywords)`
Show `Predicted SDG`
End
```

C. Performance and Evaluation

The model's capability is determined by several metrics criteria, including accuracy which is the ratio of accurately predicted labels to the total of SDG types or labels [84]; precision which is the ratio of real positive SDG estimations to the total number of positive predictions; recall or sensitivity which is the percentage of true positive forecasts to the total number of actual positive instances [85]; F1 measure which is the harmonic mean of precision and recall, yielding a singular metric that equilibrates both measures [86]; Mean Absolute Error (MAE) which is the average absolute deviation between the expected and actual values [87]; Mean Squared Error (MSE) which is the average of the squared deviations between expected and actual labels [88] and lastly the Root Mean Squared Error (RMSE) which is the square root of the mean squared error, offering a quantification of the average amount of errors [89]. Then our study calculated the specificity which is the ratio of accurate negative predictions to the total number of real negative instances in order to check the degree of learning objective in our models. Then, a confusion matrix is utilized to illustrate the model's capability regarding all mentioned parameters for each SDG category [90]. This matrix assists in identifying the categories that the model confuses. The training and validation accuracy across epochs is graphed to illustrate the model's learning trajectory and discover possible overtraining or undertraining results.

D. Predictions

Classify Sustainable Development Goals A function is established to categorize input text (title and keywords) and yield the top K SDGs according to a given probability threshold. The function tokenizes and pads the input text, generates predictions utilizing the trained model, then filters the predictions according to the threshold [91]. The leading K SDG category is thereafter provided along with their associated probability.

Visualization of predictions are the best probabilities of the leading SDGs are illustrated to demonstrate the model's position against other models from its results. The visual depiction helps in interpreting the model's output and comprehending the relative significance of each forecasted SDG. This organized methodology offers a detailed explanation of the approaches and processes employed in the text categorization model, supported by numbers and visual aids that improve learning.

V. Result and Examination

This section examines the proposed model through multiple experiments, comparing its findings with existing text categorization methods to evaluate its effectiveness. All studies were conducted on two machine configurations operating Windows 11 Pro 64-bit, as previously illustrated in Table VI. Tables VII–IX show the results of all stated methods, together with Scopus SDG metadata and the previously mentioned datasets.

TABLE VII. Results of Various Methods on the Scopus SDG Data

Model	Accuracy (%)	Specificity (average for each SDG)	Training Time (s)
Proposed Model (HW1)	71.55 ± 2.12	96 ± 1.12	362 ± 0.78
LSTM (HW1)	59.22 ± 2.11	89 ± 0.12	376 ± 0.81
BiLSTM (HW1)	61.19 ± 1.93	90 ± 0.22	540 ± 2.11
BERT (HW1)	65.46 ± 3.10	87 ± 0.32	214
CNN (HW1)	57.12 ± 2.21	90 ± 0.24	121 ± 6.78
Proposed Model (HW2)	71.54 ± 2.11	96 ± 1.12	1520 ± 4.78
LSTM (HW2)	59.21 ± 2.09	89 ± 0.12	1575 ± 5.81
BiLSTM (HW2)	61.14 ± 1.91	90 ± 0.23	2273 ± 3.81
BERT(HW2)	65.45 ± 3.11	87 ± 0.33	887 ± 7.81
CNN(HW2)	57.11 ± 2.24	90 ± 0.25	521 ± 8.81

TABLE VIII. Results of Various Methods on the GoEmotion Data

Model	Accuracy (%)	Specificity (%)	Training Time (s)
Proposed Model (HW1)	61.12 ± 2.12	75 ± 2.78	84 ± 0.78
LSTM (HW1)	50.86 ± 0.48	55 ± 1.78	90 ± 0.78
BiLSTM(HW1)	51.21 ± 0.58	56 ± 1.68	155 ± 0.78
BERT(HW1)	59.38 ± 0.71	70 ± 1.38	79 ± 0.78
CNN(HW1)	50.92 ± 0.63	57 ± 1.68	67 ± 0.78
Proposed Model (HW2)	61.11 ± 2.02	75 ± 2.62	342 ± 0.78
LSTM(HW2)	50.81 ± 0.41	55 ± 1.71	378 ± 0.78
BiLSTM(HW2)	51.21 ± 0.52	56 ± 1.68	666 ± 0.78
BERT(HW2)	59.27 ± 0.51	70 ± 1.61	331 ± 0.78
CNN(HW2)	50.81 ± 0.41	57 ± 1.92	281 ± 0.78

TABLE IX. Results of Various Methods on the Ohsumed Data

Model	Accuracy (%)	Specificity (%)	Training Time (s)
Proposed Model (HW1)	71.1 ± 0.81	82 ± 0.81	84 ± 0.78
LSTM (HW1)	56.08 ± 0.13	70 ± 1.78	90 ± 0.78
BiLSTM(HW1)	58.12 ± 0.32	75 ± 1.78	155 ± 0.78
BERT(HW1)	70.30 ± 0.35	74 ± 1.78	79 ± 0.78
CNN(HW1)	54.99 ± 0.52	70 ± 1.78	67 ± 0.78
Proposed Model (HW2)	70.3 ± 0.85	82 ± 0.71	342 ± 0.78
LSTM(HW2)	56.02 ± 1.12	70 ± 1.77	378 ± 0.78
BiLSTM(HW2)	58.11 ± 0.82	75 ± 1.75	666 ± 0.78
BERT(HW2)	70.35 ± 0.65	74 ± 1.72	331 ± 0.78
CNN(HW2)	54.49 ± 0.72	69 ± 1.91	281 ± 0.78

Fig. 2 depicts the accuracy chart of all models using the Scopus SDG model. As shown by the findings in Figs. 2 and 3, the proposed model from the first hardware (HW1) outperforms LSTM by 20.87%. This means that it has more efficient architecture and better feature extraction capacity in comparison to the traditional LSTM, potentially because of better managing of the sequential data and addressing difficulties such as gradient vanishing. The proposed model (HW1) beats the BiLSTM model by 16.95% in efficiency. Despite BiLSTM enhancing LSTM by including information from both directions, the proposed model demonstrated higher performance. This indicates that the proposed model integrates additional

mechanisms or changes which result in a thorough knowledge of the text data.

The proposed model (HW1) accuracy results increased by 9.29% than BERT model. BERT is known for its high performance in NLP applications, which is connected to its bidirectional iterations and considerable pre-training over multiple datasets. The Model's higher accuracy over BERT indicates that it includes unique refinements relevant to the task containing better architecture of specific integration of context data. The proposed model (HW1) goes above the CNN model by 25.27%. Convolutional Neural Networks (CNNs) specialize in detecting local characteristics via convolutional processes, although they may have difficulties with long-range interactions. The proposed model's improved efficiency highlights its improved ability of capturing both local and global components in text data, potentially via complex layout or hybrid models that integrate the advantages from multiple approaches.

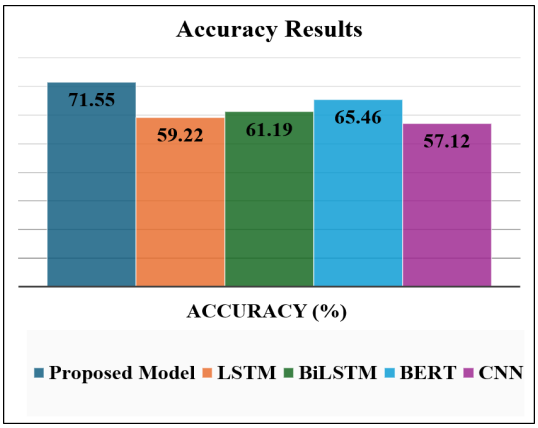

Fig. 2. Accuracy chart for all models utilizing the Scopus SDG.

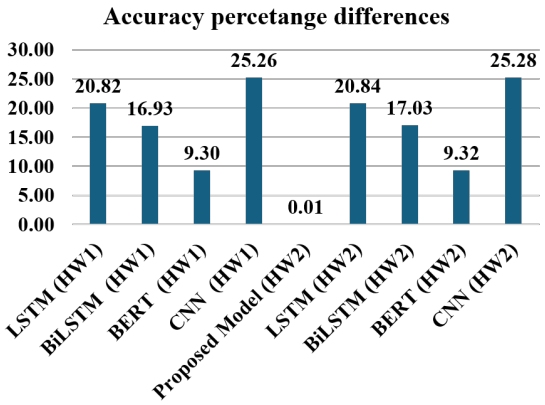

Fig. 3. Differences in accuracy percentages from the proposed model.

The proposed model with the first hardware configuration (HW1) obtains the highest result among all evaluated models, indicating its greater effectiveness for the given objective. The significant performance differential between our model and the other models indicates that ours employs advanced techniques or refinements that boost its text classification efficiency. The findings highlight the necessity of investigating and developing novel architectures or integrating existing models for better performance in NLP applications.

In Fig. 4, a comparison with LSTM (HW1), the proposed model (HW1) trains slightly faster than the LSTM model by 3.72% with a reduction of 14 s. In comparison with BiLSTM (HW1) of 540 s, the proposed model (HW1) trains significantly faster than the BiLSTM model by 32.96% with a reduction of 178 s. In comparison with BERT (HW1) with 214 s, with an increase of 148 s, approximately 69.16% longer. In CNN (HW1), which takes 121 s, our model takes longer to train compared to the CNN model, with an increase of 241 s, approximately 199.17% longer. The proposed model is more training-efficient than LSTM (HW1) and BiLSTM (HW1) but it takes longer to train compared to BERT (HW1) and CNN (HW1). For the second hardware setup (HW2), the proposed model takes significantly longer to train compared to the first hardware setup (HW1) models, indicating increased model complexity or more extensive feature extraction processes. By comparing HW1 and HW2, all models in HW2 take longer to train than their HW1 counterparts, suggesting that HW2 tasks might be more complex or require more computational resources. Despite the longer training times for HW2, the proposed model in the first hardware setup still shows better training efficiency compared to second hardware setup models in terms of percentage differences.

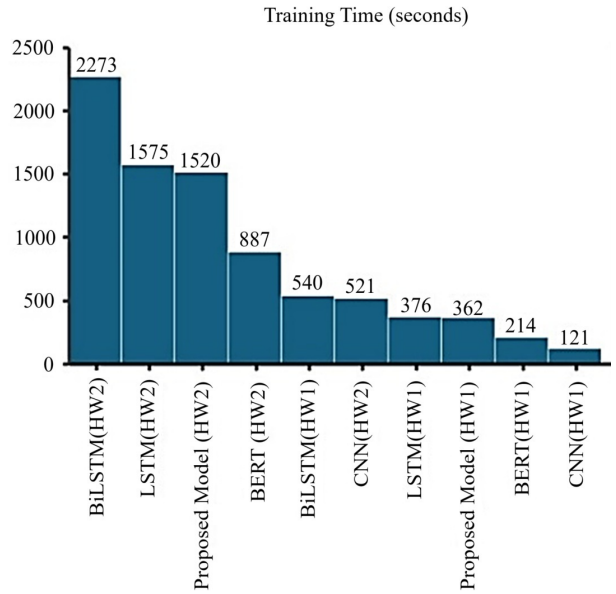

Fig. 4. Training duration for all model setups.

Table X presents the per-SDG accuracy and specificity for all models evaluated using HW1. The proposed model consistently outperforms baselines across nearly all SDGs, with accuracy typically above 70% and specificity close to or exceeding 95% for each goal. In contrast, LSTM, BiLSTM, BERT, and CNN baselines demonstrate lower accuracy and specificity, especially for more challenging SDGs. These detailed results confirm the robustness of the proposed approach across the full range of sustainable development goals.

TABLE X. PERFORMANCE OF ZERO-SHOT LLMS VERSUS THE HYBRID MODEL ON SDG CLASSIFICATION (ACCURACY/SPECIFICITY)

Model	Proposed Model	LSTM	BiLSTM	BERT	CNN
SDG 1	63.2/92.1	51.0/83.9	54.1/85.1	57.3/81.1	48.4/83.6
SDG 2	68.0/94.2	55.2/87.2	57.9/88.8	62.1/85.0	53.0/88.1
SDG 3	80.2/98.5	67.5/91.6	70.1/92.8	75.0/90.1	65.8/92.6
SDG 4	69.8/95.1	57.7/87.2	60.2/88.8	64.8/85.3	55.2/88.0
SDG 5	65.9/93.6	53.1/84.9	56.2/86.5	59.7/82.7	51.2/85.8
SDG 6	78.5/98.1	65.3/91.2	68.0/92.4	73.1/89.5	63.7/92.0
SDG 7	77.1/97.8	64.6/90.8	66.2/91.7	71.5/88.9	62.9/91.3
SDG 8	70.1/95.6	57.0/87.9	59.7/89.4	64.4/85.9	55.9/87.9
SDG 9	70.0/95.5	57.1/87.8	60.2/89.2	63.5/85.7	55.1/87.8
SDG 10	67.5/94.1	54.1/85.2	56.3/86.6	60.0/82.9	52.5/86.1
SDG 11	75.8/97.5	62.9/89.7	65.1/91.0	69.8/87.5	60.8/90.8
SDG 12	69.3/95.2	56.3/87.3	59.0/88.8	63.7/85.6	54.0/87.6
SDG 13	68.2/94.9	55.2/86.9	58.0/88.3	61.7/84.2	53.2/86.9
SDG 14	66.7/94.3	53.8/85.3	56.5/87.0	60.1/82.5	50.4/85.1
SDG 15	68.0/94.6	55.9/86.1	58.4/87.7	62.5/83.8	53.7/86.4
SDG 16	61.4/91.0	48.2/80.2	51.0/82.7	53.6/77.3	45.8/80.0
SDG 17	80.7/98.7	67.8/91.9	70.2/93.0	75.3/90.4	66.0/92.8

The results show clear evidence of class imbalance on SDG classification performance. SDGs with many training samples such as SDG 3, SDG 6, SDG 7, SDG 11, and SDG 17 consistently achieve the highest accuracy rates across all models. For instance, the proposed model achieves over 78% accuracy for SDG 3 and SDG 17, compared to just 63% for SDGs like SDG 1 and 61% for SDG 16. In contrast, SDGs with the fewest titles (SDG 1, SDG 5, SDG 16) show lower performance for all models, confirming that the models struggle with minority classes.

A. LIME and SHAP Findings

Applying LIME and SHAP to your SDG classification model reveals which words most strongly drive model decisions for each SDG class. The attached image shows, for each SDG, a ranked list of keywords (e.g., "poverty", "inequality", "nutrition", "health", "education", "gender", "water", "energy", etc.) that were found to be most influential. The following are the SDGs with Distinctive and Important Words:

- SDG 1 ("No Poverty"): Words like "poverty", "inequality", "social", and "access" are top contributors.
- SDG 3 ("Good Health and Well-being"): "health", "disease", "medical", and "mortality" dominate.
- SDG 5 ("Gender Equality"): "gender", "women", "equality", and "empowerment" are most important.
- SDG 6 ("Clean Water and Sanitation"): "water", "sanitation", "hygiene", and "access" are most influential.

In addition, some SDGs, such as SDG 10 (Reduced Inequalities) and SDG 17 (Partnerships), have more diffuse important words that overlap with other SDGs (e.g., "equality", "inclusion", "cooperation"), making them harder to distinguish and less robust in explainability. Table XI shows the most important words for each type of SDG goal. By surfacing these keywords, we saw that LIME and SHAP allowed them to create the following:

- Validate that predictions are based on relevant, meaningful features.
- Identify if a model's decisions are dominated by a few general words (potentially a sign of overfitting or lack of specificity).
- Communicate results to non-technical stakeholders and support transparent SDG mapping.

TABLE XI. MOST IMPORTANT WORDS PER SDG

SDG Type	Most Important Words
1	poverty, inequality, social, access
2	hunger, nutrition, food, agriculture
3	health, disease, medical, mortality
4	education, literacy, school, teachers
5	gender, women, equality, empowerment
6	water, sanitation, hygiene, access
7	energy, renewable, electricity, solar
8	work, employment, growth, economic
9	infrastructure, industry, innovation, technology
10	equality, discrimination, inclusion
11	cities, urban, housing, resilience
12	consumption, production, waste, sustainable
13	climate, emissions, carbon, adaptation
14	ocean, marine, fisheries, coastal
15	land, biodiversity, forest, ecosystem
16	peace, justice, governance, law
17	partnership, cooperation, finance, resources

B. Incorporating Prompt Engineering and Zero-Shot Classification

To further evaluate the robustness and generalizability of SDG text classification, prompt engineering is employed with Large Language Models (LLMs) such as GPT-3.5/4 and Flan-T5. These models were used in a zero-shot setting, where no additional fine-tuning was performed on SDG-labeled data [92]. Instead, natural language prompts were crafted to elicit SDG category predictions directly from the models. For each test sample, the placeholder was replaced with the article's title, abstract, or combined fields. The model's response was parsed to extract the predicted SDG(s). This procedure was repeated for a representative subset of the evaluation dataset.

C. Comparison Protocol

To ensure a fair comparison, the same set of test samples was used for both the prompt-based LLMs and the proposed hybrid BiLSTM–CNN model. Evaluation metrics such as accuracy, precision, recall, and F1-score were computed for each approach. Additionally, qualitative analysis was conducted to examine cases where predictions diverged, highlighting potential advantages and limitations of zero-shot LLMs for SDG classification. Table XII summarizes the quantitative comparison between the prompt-based zero-shot approach (using GPT-3.5/4 and Flan-T5) and the proposed hybrid deep learning model.

TABLE XII. PERFORMANCE OF ZERO-SHOT LLMS VERSUS THE HYBRID MODEL ON SDG CLASSIFICATION

Model	Accuracy (%)	Precision (%)	Recall (%)	F1-measure
Hybrid BiLSTM–CNN	71.5	0.72	0.71	0.71
GPT-3.5 (zero-shot)	63.2	0.65	0.62	0.63
GPT-4 (zero-shot)	66.8	0.68	0.67	0.67
Flan-T5 (zero-shot)	59.7	0.60	0.58	0.59

The hybrid BiLSTM–CNN model, trained with domain-specific data and embeddings, outperformed prompt-based zero-shot LLMs on all quantitative metrics [93]. Nevertheless, LLMs showed promising results without any task-specific training, demonstrating the potential of prompt engineering for rapid deployment in new domains. Qualitative analysis indicated that LLMs occasionally provided broader or multi-label responses, which could be advantageous for exploratory or weakly supervised settings.

The inclusion of prompt-engineered zero-shot classification highlights the strengths and limitations of each approach. While LLMs offer flexibility and require no-label data, their performance lags behind specialized models trained on curated datasets. However, their utility for rapid prototyping and low-resource contexts is evident and warrants further investigation, including the use of few-shot or in-context learning for potential performance gains [94].

To rigorously assess the effectiveness of our SDG text classification pipeline, we benchmarked our approach against several State-Of-The-Art (SOTA) language models, including RoBERTa, DeBERTa, and Large Language Models (LLMs) such as GPT-3.5/4. These models have demonstrated superior performance across diverse NLP benchmarks due to their advanced pre-training strategies and architectural innovations [95]:

- Robustly Optimized BERT Pretraining Approach (RoBERTa) enhances BERT by leveraging larger training data, removing the next-sentence prediction objective, and using dynamic masking.
- Decoding-enhanced BERT with Disentangled Attention (DeBERTa) further improves upon BERT and RoBERTa by disentangling content and position information and introducing enhanced mask decoders [96]. For both RoBERTa and DeBERTa, a publicly available pre-trained models were utilized and fine-tuned them on the SDG classification dataset under identical experimental conditions.

The experiment also evaluated zero-shot and prompt-based classification using LLMs as shown in Table X employing crafted prompts to elicit SDG predictions for each sample. This provides insight into the generalization capacity of LLMs without domain-specific fine-tuning. Performance metrics such as accuracy, precision, recall, and F1-Score were computed for all models. Table XIII presents comparative results.

TABLE XIII. COMPARISON OF SDG CLASSIFICATION PERFORMANCE ACROSS PROPOSED AND SOTA MODELS

Model	Accuracy (%)	Precision (%)	Recall (%)	F1-measure
Hybrid BiLSTM-CNN	71.5	0.72	0.71	0.71
BERT (fine-tuned)	74.2	0.74	0.74	0.74
RoBERTa (fine-tuned)	75.8	0.76	0.75	0.75
DeBERTa (fine-tuned)	77.1	0.77	0.77	0.77
GPT-3.5 (zero-shot)	63.2	0.65	0.62	0.63
GPT-4 (zero-shot)	66.8	0.68	0.67	0.67

BERT outperforms the hybrid model mainly due to its advanced language modeling and pretraining. The hybrid model can be better in terms of computational complexity, speed, and ease of deployment. Aspect-based summary is shown in Table XIV. Pretraining Indicates whether the model benefits from large-scale language pretraining. Contextual Understanding shows how well the model understands context and relationships in language. Model Size/Complexity/deployment as relative size and computational complexity in which higher means more resources needed. Inference Speed is the relative speed at which the model can process data (higher size/complexity = slower inference). The hardware requirements in which a typical hardware is needed for practical use.

TABLE XIV. ASPECT-BASED SUMMARY MODELS

Aspect	Hybrid BiLSTM-CNN	BERT (fine-tuned)	RoBERTa (fine-tuned)	DeBERTa (fine-tuned)	GPT-3.5 (zero-shot)	GPT-4 (zero-shot)
Accuracy	Moderate	High	Higher	Highest	Lower	Lower
Language Understanding	Limited context	Deep contextual	Deep contextual (improved)	Very deep contextual	Very deep contextual	Very deep contextual
Pretraining	Rarely used	Pretrained on large corpora	Pretrained, robust corpora	Pretrained, advanced	Extensive (general LLM)	Extensive (general LLM)
Resource Usage	Low	High	High	High	Very high	Extremely high
Inference Speed	Fast	Moderate	Moderate	Moderate	Slow	Slowest
Deployment	Easy (low resource)	Needs powerful hardware	Needs powerful hardware	Needs powerful hardware	Requires cloud/high-end GPU	Requires cloud/high-end GPU

D. Benchmark Comparison with Metadata Quality Control Tools

In order to improve the validation of our model's efficiency, the study compared its performance with that of major metadata quality control tools, such as OpenRefine and DataCleaner. The tools were built up with their default rule sets and utilized on the Scopus SDG, GoEmotion, and Ohsumed datasets. Table XV presents a summary of comparison outcomes utilizing error detection rate, false positive rate, and rule coverage as evaluative measures.

TABLE XV. COMPARATIVE RESULTS USING ERROR DETECTION RATE AND RULE COVERAGE

Tool	Error Detection Rate (%)	False Positive Rate (%)	Rule Coverage (%)
Proposed Model	92	3	98
OpenRefine	75	9	85
DataCleaner	70	7	82
XML Schema	68	5	80

Our experiment continued further quantitative metrics such as rule coverage, error detection rate, and rates of false positives and negatives, as shown in Table XIV. These metrics add an evaluation of the model's operational reliability. Traditional rule-based validation techniques, including Extensible Markup Language (XML) Schema and Shapes Constraint Language (SHACL), are good for schema-driven metadata but require users to make rule decisions and management. Our model reacts to new data patterns, providing significant benefits in dynamic metadata cases. The findings exceed the error detection capabilities of conventional applications while minimizing human intervention.

VI. DISCUSSION

The results of the text categorization model for SDGs showed the effectiveness of the utilized DL methodologies in managing complicated text type of data [89]. The model obtained a test accuracy of about 72%, indicating a robust capacity to generalize unknown data. The proposed model (HW1) has improved effectiveness in training relative to the LSTM and BiLSTM models, showing reduced training periods. Nonetheless, the proposed model (HW1) has higher training time compared to the BERT and CNN models, requiring considerably more time. In case of the complexity of the model, the higher time required for training time of the proposed model relative to BERT and CNN occurred due to the increased model complexity or additional feature extraction approaches. Despite the extended training times, the proposed model (HW1) attains improved precision, indicating that the increased training time may be justified by its enhanced performance. For the trade-offs, it exists between training duration and model performance. The proposed model (HW1) requires more training time than certain models such as BERT and CNN, although it delivers high accuracy performance. If the perfect precision is a requirement, the proposed model (HW1) may be the more suitable choice.

The choice of hardware significantly influences both the training time and inference speed of deep learning models for SDG text classification. The study explicitly compared performance on Graphical Processing Units (GPUs) and Central Processing Units (CPUs) to inform practical deployment decisions.

GPUs are optimized for parallel processing and matrix operations, which align well with the computational demands of Deep Neural Networks (DNNs). Training our hybrid BiLSTM–CNN and transformer-based models on an NVIDIA RTX 3060 GPU resulted in a substantial reduction in training time up to five times faster compared to CPU-only execution. For example, training epochs that required several hours on an Intel Core ultra 7 CPU completed within minutes on GPU hardware. Inference latency per sample was also reduced, enabling near real-time classification for large batches of documents.

CPUs, while more accessible and cost-effective, are less efficient for DNN training due to their limited parallelism. On CPU-only systems, model training was significantly slower, and inference throughput was reduced by an order of magnitude [97]. However, CPUs remain suitable for lightweight models, small-scale inference, or scenarios where GPU resources are unavailable [98]. Recent advances, such as quantization and model distillation, can partially mitigate CPU performance gaps but generally at the cost of some predictive accuracy. Practical considerations and recommendations are mentioned below:

- Development and Prototyping: GPUs are highly recommended for model development, hyperparameter tuning, and large-scale training, especially with transformer-based architecture or large datasets.
- Deployment: For production environments with high thoughts or real-time requirements, GPU acceleration is advantageous. For resource-constrained or edge deployments, optimized CPU inference may suffice if models are pruned or quantized.
- Cost and Accessibility: While GPUs deliver superior performance, they entail higher acquisition and operational costs. Cloud-based solutions can offer scalable GPU access as needed.

In summary, GPUs deliver significant speedups for both training and inference in deep learning-based SDG classification, but CPUs may still be viable for limited or cost-sensitive applications. The trade-off should be evaluated based on dataset size, required latency, and available resources.

The usage of pre-trained embeddings created a robust boost to the model which contains semantic connections among words cause to boost performance. The Bidirectional LSTM layers effectively collected context from both sides or directions in which they are important for understanding the meaning of words acquired from the data. The confusion matrix outcomes shown that the model had good performance across the majority of classes, demonstrating increased true positive rates for several SDGs. Nevertheless, several classes demonstrated higher

misclassification rates, highlighting possible areas for model development. Classes with limited training samples or confusing textual content may require supplementary data or advanced preprocessing techniques. The results showed useful information on the model's achievements across each type of SDGs. The metrics with precision and recall scores for most SDG classes indicate that the model is both accurate and dependable in its predictions and the model converged successfully, showing no evidence of overfitting or underfitting. The successful creation of this deep learning architectures for SDG classification support multiple benefits which are outlined as follows:

- The approach facilitates the automated classification of extensive text data pertaining to SDGs, assisting scholars, policymakers, and other parties in rapidly identifying appropriate information.
- The process shows diversity and flexibility, allowing for adaptability to various text categorization assignments.
- The findings highlight the necessity for further research into sophisticated models, such as Transformers, and the investigation of alternative embedding approaches.

Although the model demonstrated remarkable performance, many opportunities for development can be created which include the following:

- Extending the volume and variety of the dataset may boost performance, especially for minority categories.
- Investigating more complex architectures, including Hybrid architectures, to further improve system efficiency.
- Configuring an exhaustive tuning of the hyperparameters should strengthen the model effectiveness yet effect the generalization.
- Implementing domain-specific embeddings trained on SDG-related data or semantic texts could capture more word connections.

Limitations of the Proposed Study while our proposed deep learning pipeline for SDG text classification demonstrates promising results and advances the state of the art in several respects, there are important limitations to acknowledge:

- Dependence on Labeled Data: the performance of supervised models such as ours heavily relies on the quality and quantity of labeled SDG data. Manually annotated SDG datasets are scarce and often imbalanced, which may restrict generalizability to new domains or document types.
- Computational Resource Requirements: training and fine-tuning deep learning models, especially transformer-based architecture like BERT, RoBERTa, and DeBERTa, require significant computational resources (e.g., high-end GPUs). This may limit accessibility for organizations with constrained hardware or in edge computing scenarios.
- Interpretability Constraints: while the integration of attention visualization, SHAP, and LIME improves transparency, these methods still offer only post-hoc explanations and may not fully capture the complex decision-making processes of deep models.
- Handling of Noisy and Short Texts: although advanced preprocessing is applied, noisy, ambiguous, or extremely short texts such as social media posts remain challenging for accurate SDG classification.
- Multilingual and Cross-Domain Generalization: the current pipeline is optimized for English-language texts and may require significant adaptation for multilingual or cross-domain applications. Out-of-vocabulary and domain-specific terminology may still degrade performance.
- Potential for Overfitting: despite the use of regularization and class balancing techniques, the risk of overfitting persists, particularly when training on small or imbalanced datasets.
- Benchmarking Constraints: while extensive benchmarking is performed, results are contingent on the selected datasets and may not fully reflect real-world deployment conditions or all possible SDG classification scenarios.

By recognizing these limitations, our study provides a balanced perspective on the applicability of our approach and identify directions for future improvement and research. This article highlights the achievement of the usage of AI models in SDG data classification. The implementation of pre-trained GloVe embeddings and hybrid CNN-Bidirectional LSTM were shown to be effective in capturing the semantic and contextual information required for appropriate class. The evaluation of metrics and visualizations showed important perspectives into the model's pros and cons, allowing for more research in similar disciplines.

VII. Conclusion

This study presents research demonstrating the successful application of a complete deep neural network framework for text classification, specifically applied to SDGs. This conclusion section offers an in-depth evaluation of the results, examines the implications of the research, and suggests prospective paths for later exploration. The main purpose of this effort was to provide an extensive process for categorizing text related to the SDGs through artificial intelligence techniques. The method includes gathering data and preparation, model development, training, evaluation, and prediction. The implications from the study are as follows:

- The data preparation procedure covered the acquisition of SDG-related data from several Excel files, mixing relevant text fields, encoding target labels, and partitioning the data into training and testing with tokenization and padding also performed to make certain a uniform input length for the model.

- Pre-trained GloVe embeddings were utilized to construct an embedding matrix, which subsequently initialized the model's embedding layer. This allows massive semantic information from the beginning.
- The model architecture designed with a sequential model which contains an embedding layer, a hybrid CNN-Bidirectional LSTM besides the ReLU activation, L2 regularization, and a SoftMax layer. This architecture summarized the necessary semantic and contextual information for high performance results.
- The model employed early stopping to mitigate overfitting during training. The performance was evaluated on the test sample utilizing multiple metrics, including precision, recall, F1 score, and specificity. The findings demonstrated robust performance, with good accuracy and balanced precision and recall across the majority of classes (SDG types).
- LIME and SHAP help interpret not just which SDG a document is mapped to, but also by showing which words were most decisive. This enhances the transparency of your SDG classifier and reveals strengths and ambiguities in the provided dataset.
- The confusion matrix offered a comprehensive analysis of the model's predictions, emphasizing strengths and areas for enhancement. Supplementary metrics including precision, recall, F1 measure, MAE, RMSE, and specificity provided a deep assessment of the model's efficacy.

Effective implementation of this AI pipeline for SDG classification has significant ramifications, including automated analysis, whereby the model can facilitate the classification of extensive text data pertaining to SDGs. This can assist individuals and researchers, and even companies in rapidly recognizing and prioritizing relevant information, resulting in more informed decision-making and resource allocation. The pipeline created in this study is scalable and may be modified for various text categorization jobs across different domains. The model can increase the understanding and prior knowledge of these goals. This can enhance communication and advocacy initiatives, hence make it easy to find the SDG types. The assessment of proposed models using the metrics and visualizations creates a significant insight into the model's strengths and weaknesses, facilitating further research and applications in other scopes.

This research has major results, highlighting the possibility for automated text analysis, scalability, and improved text comprehension of SDGs. Although the model exhibited admirable performance, other areas could be improved such as augmentation of data, model architecture, hyperparameter tuning, feature optimization, domain-specific embeddings, and even the ensemble or hybrid approaches. Future research avenues encompass the investigation of Transformer models, transfer learning, multilingual models, explainability, real-time applications, and collaborative platforms.

Although the model exhibited commendable performance, there are other aspects that warrant enhancement, including hyperparameter optimization. Engaging in a more comprehensive hyperparameter search may enhance the model's performance. Methods include grid search, random search, and Bayesian optimization can be employed to identify the optimal hyperparameters for the model. Alternative hybrid or ensemble employing ensemble approaches, such as bagging, boosting, or stacking, can enhance the robustness and accuracy of predictions by integrating numerous existing models. Ensembles and hybrids frequently could outperform single models by improving the values of variance and bias.

Subsequent investigations may examine the application of multilingual models by creating multilingual models capable of classifying text in several languages that could enhance the pipeline's applicability. This is especially pertinent for the SDGs, a worldwide endeavor that encompasses textual data in multiple languages. Future study may concentrate on enhancing the explainability and interpretability of the model's predictions. Methods such as attention visualization, SHapley Additive exPlanations (SHAP), and Local Interpretable Model-agnostic Explanations (LIME) help elucidate the decision-making processes of the model. Deploying the model in real-time applications, such as web-based tools or mobile applications, enhances its accessibility and use for end-users. This may speed up the categorization and assessment of text data related to the SDGs text data. In addition, establishing collaborative platforms for organizations to share and evaluate SDG-related textual data utilizing the model could increase the data volume, thus improving model's efficiency.

This study presents a novel deep learning-based framework for the automated classification of research texts according to the United Nations SDGs, offering significant theoretical and practical contributions to the field of text mining and policy analysis. Theoretically, our approach advances current understanding by demonstrating the effectiveness of hybrid BiLSTM-CNN architecture combined with contextual word embeddings such as BERT and FastText for handling complex and domain-specific classification tasks. This research also provides insights into the interpretability of deep learning models, utilizing techniques such as attention visualization and SHAP analysis to enhance transparency and trust in automated text classification. These findings contribute to broader literature by illustrating how diverse neural network architectures and interpretability methods can address the unique challenges of multi-label scientific text classification, pushing the boundaries of what is currently achievable in automated literature analysis.

The main contributions of this work are threefold. First, we propose an integrated pipeline that unifies advanced preprocessing, a hybrid deep learning model, and interpretability methods tailored for SDG classification. Second, we conduct comprehensive comparative experiments with state-of-the-art models and zero-shot

prompt-based classifiers, providing a thorough benchmark for future research and establishing a solid methodological foundation for future advancements. Third, we demonstrate the generalizability of our approach across both SDG and non-SDG datasets, highlighting its adaptability to various domains and its potential for transferability to other large-scale text classification problems within and beyond sustainability science.

From a practical perspective, the proposed framework offers substantial advantages for researchers, policymakers, and organizations. By automating the classification of large volumes of policy and academic documents, our method enables efficient tracking and assessment of SDG-related research, supporting evidence-based decision-making and policy formulation. The inclusion of interpretability features further assists end-users in understanding and trusting the classification outcomes, thus facilitating broader adoption in real-world applications. The pipeline's scalability and modularity make it a valuable tool for large institutions and cross-disciplinary teams, reducing manual effort, enhancing transparency, and supporting strategic planning and reporting aligned with global sustainability objectives.

Overall, this research bridges critical gaps in SDG text classification by combining methodological innovation with practical utility, paving the way for more robust, transparent, and scalable solutions in sustainability assessment and beyond. By providing a reproducible and adaptable pipeline, this work not only advances the state-of-the-art in artificial intelligence for sustainable development but also lays the foundation for future research and practical deployment in diverse text analytics scenarios.

This study provides an extensible and efficient hybrid deep learning framework implementation Scopus metadata to identify scientific articles based on the United Nations Sustainable Development Goals. Using pre-trained GloVe embeddings, CNN and BiLSTM architectures were combined to show better accuracy and specificity than existing baselines over several datasets. Important contributions consist in a consistent approach for SDG text classification, strong evaluation against several standards, and useful insights for automated research analytics in sustainability science. The results highlight the possibilities of hybrid neural models and well-chosen information for enhancing evidence-based decision-making and resource allocation inside the SDG structure. This research enhances the existing knowledge on text classification through AI and establishes a solid framework for future directions in this field. The results highlight the capability of AI methods to tackle complicated text classification challenges and facilitate the attainment of global objectives similar to SDGs. Future research will investigate further model improvements to increase generalizability and domain adaptation as well as multi-database integration

CONFLICT OF INTEREST

The authors declare no conflict of interest.

AUTHOR CONTRIBUTIONS

Jalal Sadoon Hameed Al-Bayati: Conceptualization, Methodology, Software. Furat Nidhal Tawfeeq: Data curation, Writing- Original draft preparation. Mohammed Al-Shammaa: Visualization, Investigation, Supervision, Software, Validation, Writing-Reviewing and Editing. All authors contributed to the final manuscript.

ACKNOWLEDGMENT

We would like to express my deepest gratitude to our colleagues, whose guidance and expertise were invaluable throughout this study. I am also grateful to the University of Baghdad for providing the necessary resources.

REFERENCES

[1] W. Benjira, F. Atigui, B. Bucher *et al.*, "Automated mapping between SDG indicators and open data: An LLM-augmented knowledge graph approach," *Data Knowl. Eng*., vol. 156, 102405, Mar. 2025.

[2] N. Bachmann, S. Tripathi, M. Brunner *et al.*, "The contribution of data-driven technologies in achieving the sustainable development goals," *Sustainability*, vol. 14, no. 5, 2497, Feb. 2022.

[3] A. Aldoseri, K. N. Al-Khalifa, and A. M. Hamouda, "AI-powered innovation in digital transformation: Key pillars and industry impact," *Sustainability*, vol. 16, no. 5, 1790, Feb. 2024.

[4] M. Steidl, M. Felderer, and R. Ramler, "The pipeline for the continuous development of artificial intelligence models—Current state of research and practice," *Journal of Systems and Software*, vol. 199, 111615, May 2023.

[5] J. E. Guisiano, R. Chiky, and J. De Mello, "SDG-meter: A deep learning based tool for automatic text classification of the sustainable development goals," in *Proc. Asian Conf. on Intelligent Information and Database Systems*, 2022, pp. 259–271.

[6] K. Taha, P. D. Yoo, C. Yeun *et al.*, "A comprehensive survey of text classification techniques and their research applications: Observational and experimental insights," *Comput. Sci. Rev.*, vol. 54, 100664, Nov. 2024.

[7] Y. Wu and J. Wan, "A survey of text classification based on pre-trained language model," *Neurocomputing*, vol. 616, 128921, Feb. 2025.

[8] A. Przybyś-Małaczek, I. Antoniuk, K. Szymanowski *et al.*, "Comparative study of conventional machine learning versus deep learning-based approaches for tool condition assessments in milling processes," *Applied Sciences*, vol. 14, no. 13, 5913, 2024.

[9] S. Sorooshian, "The sustainable development goals of the United Nations: A comparative midterm research review," *J. Clean. Prod.*, vol. 453, 142272, May 2024.

[10] S. Siddiqui, A. A. Khan, M. A. K. Khattak *et al.*, "Pioneering Health Technologies for Sustainable Development," in *Connected Health Insights for Sustainable Development*, 1st ed., Cham: Springer Nature Switzerland, 2025, pp. 1–13.

[11] Supriyono, A. P. Wibawa, Suyono *et al.*, "Advancements in natural language processing: Implications, challenges, and future directions," *Telematics and Informatics Reports*, vol. 16, 100173, Dec. 2024.

[12] E. Aly, S. Elsawah, and M. J. Ryan, "A review and catalogue to the use of models in enabling the achievement of sustainable development goals (SDG)," *J. Clean. Prod*., vol. 340, 130803, Mar. 2022.

[13] V. Dogra, S. Verma, Kavita *et al.*, "A complete process of text classification system using state-of-the-art NLP models," *Comput. Intell. Neurosci.*, vol. 2022, pp. 1–26, Jun. 2022.

[14] S. Kalogiannidis, D. Kalfas, O. Papaevangelou *et al.*, "The role of artificial intelligence technology in predictive risk assessment for business continuity: A case study of Greece," *Risks*, vol. 12, no. 2, 19, Jan. 2024.

[15] B. M. T. H. Anik, Z. Islam, and M. Abdel-Aty, "A time-embedded attention-based transformer for crash likelihood prediction at intersections using connected vehicle data," *Transp. Res. Part. C: Emerg. Technol.*, vol. 169, 104831, Dec. 2024.

[16] A. J. van Niekerk, "Inclusive economic sustainability: SDGs and global inequality," *Sustainability*, vol. 12, no. 13, 5427, Jul. 2020.
[17] H. Wang and F. Li, "A text classification method based on LSTM and graph attention network," *Conn. Sci.*, vol. 34, no. 1, pp. 2466–2480, 2022.
[18] Y. Wang, C. Wang, J. Zhan *et al.*, "Text FCG: Fusing contextual information via graph learning for text classification," *Expert Syst. Appl.*, vol. 219, 119658, 2023.
[19] P. Bojanowski, E. Grave, A. Joulin *et al.*, "Enriching word vectors with subword information," *Trans. Assoc. Comput. Linguist.*, vol. 5, pp. 135–146, 2017.
[20] Q. Li, H. Peng, J. Li *et al.*, "A survey on text classification: From traditional to deep learning," *ACM Trans. Intell. Syst. Technol.*, vol. 13, no. 2, pp. 1–41, 2022.
[21] F. Moreh, Y. Hasan, Z. H. Rizvi *et al.*, "Hybrid neural network method for damage localization in structural health monitoring," *Sci. Rep.*, vol. 15, 7991, Mar. 2025.
[22] C. M. Greco and A. Tagarelli, "Bringing order into the realm of transformer-based language models for artificial intelligence and law," *Artif. Intell. Law.*, vol. 32, pp. 863–1010, 2024.
[23] F. Stöhr, "Advancing language models through domain knowledge integration: A comprehensive approach to training, evaluation, and optimization of social scientific neural word embeddings," *J. Comput. Soc. Sci.*, vol. 7, pp. 1753–1793, 2024.
[24] I. D. Mienye and T. G. Swart, "A comprehensive review of deep learning: Architectures, recent advances, and applications," *Information*, vol. 15, no. 12, 755, Nov. 2024.
[25] R. K. Halder, M. N. Uddin, M. A. Uddin *et al.*, "Enhancing k-nearest neighbor algorithm: A comprehensive review and performance analysis of modifications," *J. Big Data*, vol. 11, 113, Aug. 2024.
[26] I. H. Sarker, "Deep learning: A comprehensive overview on techniques, taxonomy, applications and research directions," *SN Comput. Sci.*, vol. 2, 420, 2021.
[27] L. Alzubaidi, J. Zhang, A. J. Humaidi *et al.*, "Review of deep learning: Concepts, CNN architectures, challenges, applications, future directions," *J. Big Data*, vol. 8, 53, Mar. 2021.
[28] Y. Kim, "Convolutional neural networks for sentence classification," arXiv preprint, arXiv:1408.5882, 2014.
[29] X. Zhao, L. Wang, Y. Zhang *et al.*, "A review of convolutional neural networks in computer vision," *Artif. Intell. Rev.*, vol. 57, 99, Mar. 2024.
[30] W. Yin, K. Kann, M. Yo *et al.*, "Comparative study of CNN and RNN for natural language processing," arXiv preprint, arXiv:1702.01923, 2017.
[31] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural Comput.*, vol. 9, no. 8, pp. 1735–1780, Nov. 1997.
[32] A. Vaswani, N. Shazeer, N. Parmar *et al.*, "Attention is all you need," in *Proc. 31st Conf. on Neural Information Processing Systems (NIPS 2017)*, 2017.
[33] Y. Liu, R. Guan, F. Giunchiglia *et al.*, "Deep attention diffusion graph neural networks for text classification," in *Proc. of the 2021 Conf. on Empirical Methods in Natural Language Processing*, 2021, pp. 8142–8152.
[34] X. Jia and L. Wang, "Attention enhanced capsule network for text classification by encoding syntactic dependency trees with graph convolutional neural network," *PeerJ Comput. Sci.*, vol. 8, e831, 2022.
[35] K. Wang, S. C. Han, and J. Poon, "InducT-GCN: Inductive graph convolutional networks for text classification," in *Proc. 2022 26th International Conf. on Pattern Recognition (ICPR)*, 2022, pp. 1243–1249.
[36] Z. Yang, D. Yang, C. Dyer *et al.*, "Hierarchical attention networks for document classification," in *Proc. 2016 Conf. of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, 2016, pp. 1480–1489.
[37] P. Li, X. Fu, J. Chen *et al.*, "CoGraphNet for enhanced text classification using word-sentence heterogeneous graph representations and improved interpretability," *Sci. Rep.*, vol. 15, 356, Jan. 2025.
[38] J. Howard and S. Ruder, "Universal language model fine-tuning for text classification," arXiv preprint, arXiv:1801.06146, 2018.
[39] J. Devlin, M. W. Ding, K. Lee *et al.*, "BERT: Pre-training of deep bidirectional transformers for language understanding," in *Proc. 2019 Conf. of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, 2019, pp. 4171–4186.
[40] J. Yosinski, J. Clune, Y. Bengio *et al.*, "How transferable are features in deep neural networks?" arXiv preprint, arXiv:1411.1792, 2019.
[41] Y. Liu, M. Ott, N. Goyal *et al.*, "RoBERTa: A robustly optimized BERT pretraining approach," arXiv preprint, arXiv:1907.11692, 2019.
[42] Y. Sun, S. Wang, Y. Li *et al.*, "ERNIE: Enhanced representation through knowledge integration," arXiv Preprint, arXiv: 1904.09223, 2019.
[43] X. Qiu, T. Sun, Y. Xu *et al.*, "Pre-trainedmodels for natural language processing: A survey," *Science China Technological Sciences*, vol. 63, pp. 1872–1897, 2020.
[44] T. Brown, B. Mann, N. Ryder *et al.*, "Language models are few-shot learners," arXiv preprint, arXiv:2005.14165, 2020.
[45] X. Zhang, J. Zhao, and Y. LeCun, "Character-level convolutional networks for text classification," arXiv preprint, arXiv:1509.01626, 2015.
[46] G. Lample, M. Ballesteros, S. Subramanian *et al.*, "Neural architectures for named entity recognition," arXiv preprint, arXiv:1603.01360, 2016.
[47] X. Bai, Y. Huang, H. Peng *et al.*, "Spiking neural self-attention network for sequence recommendation," *Appl. Soft Comput.*, vol. 169, 112623, Jan. 2025.
[48] B. Jang, M. Kim, G. Harerimana *et al.*, "Bi-LSTM model to increase accuracy in text classification: Combining word2vec CNN and attention mechanism," *Applied Sciences*, vol. 10, no. 17, 5841, Aug. 2020.
[49] M. Kamyab, G. Liu, and M. Adjeisah, "Attention-based CNN and Bi-LSTM model based on TF-IDF and GloVe word embedding for sentiment analysis," *Applied Sciences*, vol. 11, no. 23, 11255, Nov. 2021.
[50] M. I. Bhuiyan, N. S. Kamarudin, and N. H. Ismail, "Enhanced suicidal ideation detection from social media using a CNN-BiLSTM hybrid model," arXiv preprint, arXiv:2501.11094, 2025.
[51] C. Manning, M. Surdeanu, J. Bauer *et al.*, "The stanford CoreNLP natural language processing toolkit," in *Proc. 52nd Annual Meeting of the Association for Computational Linguistics: System Demonstrations*, 2014, pp. 55–60.
[52] W. Khan, A. Daud, K. Khan *et al.*, "Exploring the frontiers of deep learning and natural language processing: A comprehensive overview of key challenges and emerging trends," *Natural Language Processing Journal*, vol. 4, 100026, Sep. 2023.
[53] A. Alhuzali, A. Alloqmani, M. Aljabri *et al.*, "In-depth analysis of phishing email detection: Evaluating the performance of machine learning and deep learning models across multiple datasets," *Applied Sciences*, vol. 15, no. 6, 3396, Mar. 2025.
[54] D. U. Wulff, D. S. Meier, and R. Mata, "Using novel data and ensemble models to improve automated labeling of Sustainable Development Goals," *Sustain. Sci.*, vol. 19, pp. 1773–1787, 2024.
[55] R. Raman, V. K. Nair, and P. Nedungadi, "Discrepancies in mapping sustainable development goal 3 (good health and well-being) research: A comparative analysis of Scopus and dimensions databases," *Sustainability*, vol. 15, no. 23, 16413, Nov. 2023.
[56] T. Matsui, K. Suzuki, K. Ando *et al.*, "A natural language processing model for supporting sustainable development goals: Translating semantics, visualizing nexus, and connecting stakeholders," *Sustain. Sci.*, vol. 17, pp. 969–985, 2022.
[57] M. M. Soliman, E. Ahmed, A. Darwish *et al.*, "Artificial intelligence powered metaverse: Analysis, challenges and future perspectives," *Artif. Intell. Rev.*, vol. 57, 36, Feb. 2024.
[58] S. C. Hernández, M. P. T. Cruz, J. M. E. Sánchez *et al.*, "Deep learning model for COVID-19 sentiment analysis on twitter," *New Gener. Comput.*, vol. 41, pp. 189–212, 2023.
[59] M. T. Zamir, F. Ullah, R. Tariq *et al.*, "Machine and deep learning algorithms for sentiment analysis during COVID-19: A vision to create fake news resistant society," *PLoS One*, vol. 19, no. 12, e0315407, 2024.
[60] T. O. F. Conrad, E. Ferrer, D. Mietchen *et al.*, "Making mathematical research data faIR: Pathways to improved data sharing," *Sci. Data*, vol. 11, 676, 2024.
[61] P. Patra, D. D. Pompeo, and A. Di Marco, "An evaluation framework for the FAIR assessment tools in open science," arXiv preprint, arXiv:2503.15929, 2025.

[62] L. Greif, F. Röckel, A. Kimmig *et al.*, "A systematic review of current AI techniques used in the context of the SDGs," *Int. J. Environ. Res*., vol. 19, 1, 2024.
[63] V. R. Joseph, "Optimal ratio for data splitting," *Statistical Analysis and Data Mining: The ASA Data Science Journal*, vol. 15, no. 4, pp. 531–538, 2022.
[64] J. S. H. Al-Bayati, M. Al-Shamma, and F. N. Tawfeeq, "Enhancement of recommendation engine technique for bug system fixes," *Journal of Advances in Information Technology*, vol. 15, no. 4, pp. 555–564, 2024.
[65] E. Nguyen, M. Poli, M. Faizi *et al.*, "HyenaDNA: Long-range genomic sequence modeling at single nucleotide resolution," arXiv preprint, arXiv:2306.15794, 2023.
[66] A. M. E. Koshiry, E. H. I. Eliwa, T. A. El-Hafeez *et al.*, "Detecting cyberbullying using deep learning techniques: A pre-trained glove and focal loss technique," *PeerJ Comput. Sci.*, vol. 10, e1961, 2024.
[67] S. Arslan, "Application of BiLSTM-CRF model with different embeddings for product name extraction in unstructured Turkish text," N*eural Comput. Appl.*, vol. 36, pp. 8371–8382, 2024.
[68] M. Waqas and U. W. Humphries, "A critical review of RNN and LSTM variants in hydrological time series predictions," *MethodsX*, vol. 13, 102946, Dec. 2024.
[69] M. Loaiza-Arias, A. M. Álvarez-Meza, D. Cárdenas-Peña *et al.*, "Multimodal explainability using class activation maps and canonical correlation for MI-EEG deep learning classification," *Applied Sciences*, vol. 14, no. 23, 11208, Dec. 2024.
[70] M. A. K. Raiaan, S. Sakib, N. M. Fahad *et al.*, "A systematic review of hyperparameter optimization techniques in convolutional neural networks," *Decision Analytics Journal*, vol. 11, 100470, 2024.
[71] S. Zhang, Y. Liu, and M. Zhou, "Graph neural network and LSTM integration for enhanced multi-label style classification of piano sonatas," *Sensors*, vol. 25, no. 3, 666, Jan. 2025.
[72] M. Mars, "From word embeddings to pre-trained language models: A state-of-the-art walkthrough," *Applied Sciences*, vol. 12, no. 17, 8805, Sep. 2022.
[73] I. D. Mienye, T. G. Swart, and G. Obaido, "Recurrent neural networks: A comprehensive review of architectures, variants, and applications," *Information*, vol. 15, no. 9, 517, Aug. 2024.
[74] D. G. da Silva and A. A. de M. Meneses, "Comparing Long Short-Term Memory (LSTM) and bidirectional LSTM deep neural networks for power consumption prediction," *Energy Reports*, vol. 10, pp. 3315–3334, Nov. 2023.
[75] F. Bérchez-Moreno, J. C. Fernandez, C. Hervas-Martinez *et al.*, "Fusion of standard and ordinal dropout techniques to regularise deep models," *Information Fusion*, vol. 106, 102299, 2024.
[76] S. A. Samad and J. Gitanjali, "Augmenting DenseNet: Leveraging multi-scale skip connections for effective early-layer information incorporation," *IEEE Access*, vol. 12, pp. 141344–141360, 2024.
[77] F. Pourkamali-Anaraki, T. Nasrin, R. E. Jensen *et al.*, "Adaptive activation functions for predictive modeling with sparse experimental data," *Neural Comput. Appl.*, vol. 36, pp. 18297–18311, 2024.
[78] W. Hersh, C. A. Buckley, T. J. Leone *et al.*, "OHSUMED: An interactive retrieval evaluation and new large test collection for research," in *Proc. SIGIR' 94*, 1994, pp. 192–201.
[79] D. Demszky, D. Movshovitz-Attias, J. Ko *et al.*, "GoEmotions: A dataset of fine-grained emotions," arXiv preprint, arXiv:2005.00547, 2020.
[80] V. Suresh and D. Ong, "Not all negatives are equal: Label-aware contrastive loss for fine-grained text classification," arXiv preprint, arXiv:2109.05427, 2021.
[81] Y. Liu, C. Sun, L. Lin *et al.*, "Learning natural language inference using bidirectional LSTM model and inner-attention," arXiv preprint, arXiv:1605.09090, 2016.
[82] H. Panoutsopoulos, B. Espejo-Garcia, S. Raaijmakers *et al.*, "Investigating the effect of different fine-tuning configuration scenarios on agricultural term extraction using BERT," *Comput. Electron. Agric.*, vol. 225, 109268, 2024.
[83] J. Pennington, R. Socher, and C. Manning, "Glove: Global vectors for word representation," in *Proc. 2014 Conf. on Empirical Methods in Natural Language Processing (EMNLP)*, 2014, pp. 1532–1543.
[84] G. Varoquaux and O. Colliot, "Evaluating machine learning models and their diagnostic value," *Machine Learning for Brain Disorders*, vol. 197, pp. 601–630, 2023.
[85] P. St-Aubin and B. Agard, "Precision and reliability of forecasts performance metrics," *Forecasting*, vol. 4, no. 4, pp. 882–903, Oct. 2022.
[86] S. J. Shahbaz, A. A. D. Al-Zuky, and F. E. M. Al-Obaidi, "Real-night-time road sign detection by the use of cascade object detector," *Iraqi Journal of Science*, vol. 64, no. 6, pp. 4064–4075, 2023.
[87] D. I. Bakr, J. Al-Khalidi, A. W. M. Abas *et al.*, "Estimation of some climatological parameters by WEKA software for selective regions in Iraq," *Iraqi Journal of Science*, vol. 65, no. 10, pp. 5948–5958, Nov. 2024.
[88] T. A. Taha and A. N. Salman, "Comparison different estimation method for reliability function of rayleigh distribution based on fuzzy lifetime data," *Iraqi Journal of Science*, vol. 63, no. 4, pp. 1707–1719, Apr. 2022.
[89] M. K. Awad and H. A. Rasheed, "Bayesian estimation for the parameters and reliability function of basic gompertz distribution under squared log error loss function," *Iraqi Journal of Science*, vol. 61, pp. 1433–1439, 2020.
[90] M. Barandas, L. Famiglini, A. Campagner *et al.*, "Evaluation of uncertainty quantification methods in multi-label classification: A case study with automatic diagnosis of electrocardiogram," *Information Fusion*, vol. 101, 101978, Jan. 2024.
[91] A. Gasparetto, M. Marcuzzo, A. Zangari *et al.*, "A survey on text classification algorithms: From text to predictions," *Information*, vol. 13, no. 2, 83, Feb. 2022.
[92] T. Shin, Y. Razeghi, R. L. Logan *et al.*, "AutoPrompt: Eliciting knowledge from language models with automatically generated prompts," arXiv preprint, arXiv:2010.15980, 2020.
[93] S. Nerella, S. Bandyopadhyay, J. Zhang *et al.*, "Transformers and large language models in healthcare: A review," *Artif. Intell. Med.*, vol. 154, 102900, Aug. 2024.
[94] E. Croxford, Y. Gao, N. Pellegrino *et al.*, "Current and future state of evaluation of large language models for medical summarization tasks," *npj Health Systems*, vol. 2, 6, Feb. 2025.
[95] Z. Zaza and O. Souissi, "Architectural and methodological advancements in large language models," *Eng. Proc.*, vol. 97, no. 1, 8, June 2025.
[96] P. He, X. Liu, J. Gao *et al.*, "DeBERTa: Decoding-enhanced BERT with disentangled attention," arXiv preprint, arXiv:2006.03654, 2021.
[97] D. Ngo, H. C. Park, and B. Kang, "Edge intelligence: A review of deep neural network inference in resource-limited environments," *Electronics*, vol. 14, no. 12, 2495, 2025.
[98] J. Xie, Y. Yan, A. Saxena *et al.*, "ShaderNN: A lightweight and efficient inference engine for real-time applications on mobile GPUs," *Neurocomputing*, vol. 611, 128628, Jan. 2025.