IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received 2 January 2025, accepted 29 January 2025, date of publication 4 February 2025, date of current version 21 February 2025.

Digital Object Identifier 10.1109/ACCESS.2025.3538621

RESEARCH ARTICLE

Dynamic Text Augmentation for Robust Sentiment Analysis: Enhancing Model Performance With EDA and Multi-Channel CNN

KOMANG WAHYU TRISNA[1,2], JINJIE HUANG[1,3], YUANJIAN CHEN[1], AND I GEDE JULIANA EKA PUTRA[2], (Member, IEEE)

[1]School of Computer Science and Technology, Harbin University of Science and Technology, Harbin 150080, China
[2]Department of Informatic Engineering, Primakara University, Denpasar, Bali 80226, Indonesia
[3]Key Laboratory of Advanced Manufacturing and Intelligence Technology, Harbin University of Science and Technology, Harbin 150080, China

Corresponding author: Jinjie Huang (jjhuangps@163.com)

ABSTRACT The rapid growth of social media has revolutionized the way individuals share and access opinions, generating vast amounts of textual data containing diverse sentiments. Sentiment analysis enables organizations to derive valuable insights from this data, facilitating improved decision-making. However, the limited availability of labeled datasets and challenges related to data imbalance often hinder the development of robust sentiment analysis models. In this study, we propose a novel framework dynamic Easy Data Augmentation (EDA) technique with a Multi-Channel Convolutional Neural Network (MCNN) to address these issues. The EDA techniques including synonym replacement (SR), random insertion (RI), random deletion (RD), and random swap (RS), enrich the training data by introducing diversity, thereby enhancing model robustness. The MCNN architecture effectively captures local patterns in text using multiple filter sizes, enabling comprehensive feature extraction with low computational overhead. Furthermore, the integration of Word2Vec embeddings ensures meaningful text representation, leveraging context-independent word relationships for improved sentiment analysis. Experimental evaluations demonstrate the effectiveness of the proposed approach, highlighting significant improvements in F1-score compared to baseline models. The findings underscore the potential of dynamic text augmentation and efficient deep learning architectures to overcome the challenges of limited training data, offering a scalable solution for sentiment analysis in real-world applications. Additionally, the proposed model achieves an average performance improvement of 11.34% over previous models.

INDEX TERMS Convolutional neural networks, deep learning, sentiment analysis, text augmentation, word embedding.

I. INTRODUCTION

The rapid proliferation of social media has transformed how individuals share and obtain opinions on various topics, including events, products, and services [1], [2]. User feedback on these platforms often reflects personal experiences, encompassing both positive and negative sentiments. Detecting adverse user sentiments is particularly crucial for organizations, as these insights can drive improvements in products and services, ultimately enhancing customer satisfaction and profitability [3]. In contemporary times, individuals are no longer confined to seeking advice from acquaintances when making purchasing decisions regarding consumer products, due to the vast availability of user reviews and discussions on public online forums. As a result, businesses may not need to conduct surveys, opinion pools, or focus group to gather customer insights, due to the abundance of publicity accessible information. However, the process of identifying and navigating opinion websites on the Internet, as well as extracting relevant information

The associate editor coordinating the review of this manuscript and approving it for publication was Maria Chiara Caschera.

from them, remains challenging due to the sheer volume of such platforms. Each website typically hosts large amounts of subjective content, including lengthy blog posts and forum discussions, which can be difficult to interpret. This makes it difficult for the average reader to efficiently identify pertinent websites and distill the opinions expressed within them. Consequently, there is a pressing need for automated sentiment analysis systems [4].

Sentiment analysis primarily focuses on evaluating opinions [5], concentrating on the automated examination of subjective text to assess the emotional tone and attitude of consumers [6]. As a key area of research within Natural Language Processing (NLP), sentiment analysis aims to determine the polarity of textual data, categorizing sentiments as positive, negative, or sometimes neutral [7]. This technique is widely employed by businesses to gain insights from social media comments, product reviews, and various textual data sources. The ability to accurately identify sentiment is crucial for both individual decision-making and organizations or governmental bodies. A precise understanding of public attitudes toward policies, products, and organizations offers significant advantages to these entities, enhancing decision support systems and aiding strategic decisions [8]. Initially, sentiment analysis relied on shallow models trained with carefully crafted features for effective polarity classification [9].

Recent advances in sentiment analysis have shifted towards deep learning techniques [10], offering significant advantages over traditional machine learning approaches, such as Naïve Bayes [11], Support Vector Machines (SVM) [12], and Decision Trees [13]. While traditional algorithms rely on manually crafted features, requiring significant domain expertise [14], deep learning eliminates this dependency by learning features automatically from raw data. This capability, along with its ability to capture intricate relationships between words and sentences, makes deep learning ideal for NLP tasks like sentiment analysis [15], [16]. However, these models require substantial computational resources and large volumes of training data, posing efficiency and interpretability challenges.

Deep learning models, such as Convolutional Neural Networks (CNNs), Long Short-Term Memory (LSTM), and Gated Recurrent Units (GRUs), have demonstrated remarkable capabilities in handling textual data for sentiment analysis. Among these, CNNs are particularly advantageous for their computational efficiency and ability to process local features in text. However, the efficacy of these models heavily depends on the quality and diversity of training data. Despite the abundance of user-generated content, acquiring high-quality labeled data remains a challenge, as data scarcity, imbalance, and redundancy can adversely impact model performance.

To address these issues, text augmentation techniques, such as Easy Data Augmentation (EDA) [17], have been introduced. These methods enhance the diversity and quantity of training data by applying transformations like synonym replacement, random insertion, random deletion, and others, which enable models to generalize better to unseen data. Without augmentation, deep learning models risk overfitting or underperforming due to insufficient or homogeneous training datasets. Thus, integrating data augmentation into sentiment analysis workflows is essential for exploiting the full potential of deep learning models while addressing the challenges of limited and imbalanced datasets.

In this study, we propose a sentiment analysis framework that integrates Dynamic EDA with a Multi-Channel Convolutional Neural Network (MCNN) to boost performance. The MCNN architecture leverages Word2Vec embeddings to capture semantic relationships, ensuring computational efficiency and effective contextual representation. By combining these components, the model aims to address limitations associated with traditional sentiment analysis, such as imbalanced data and insufficient generalization. This research investigates the effectiveness of individual EDA techniques, synonym replacement (SR), random insertion (RI), random deletion (RD), and random swap (RS) on model performance. The proposed approach is designed to improve sentiment analysis accuracy while maintaining scalability, making it suitable for real-world applications where data diversity and efficiency are critical. Additionally, the model aims to determine which EDA technique is most appropriate by proposing a dynamic EDA model. Our contribution as follows:

1. We apply Easy Data Augmentation (EDA) techniques to enrich the training data with new examples that preserve the original meaning. We dynamically design the model to select the augmentation method used for each review.
2. We combine both the original dataset and the augmented data as input for sentiment classification. This integration enriches the feature space and provides a more comprehensive representation of the text, which can potentially improve model performance and sentiment prediction accuracy.
3. We develop a Convolutional Neural Network (CNN) model using a multi-channel architecture for sentiment classification. This approach allows the model to extract richer features from the input text by utilizing multiple convolutional filters with different kernel sizes, enhancing its ability to capture diverse patterns and context in the data.
4. We conduct a thorough analysis of the proposed model across different scenarios, including the impact of using various data augmentation techniques and different word embedding models. This analysis aims to evaluate how each augmentation method and embedding type affects the model's performance, providing valuable insights into their contributions to sentiment classification.

The remainder of this paper is structured as follows: Section II provides a comprehensive review of related work, Section III details the methodologies employed and the proposed model, Section IV discusses the experimental setup

and results, and Section V concludes with a summary of findings and potential directions for future research.

II. RELATED WORKS

In this section, we discuss various theories and previous research related to sentiment analysis. This review covers approaches using convolutional neural networks and text augmentation techniques which have been applied to improve model performance in sentiment analysis. These studies provide a foundation for developing our proposed model and serve as a reference for understanding its contributions and differences from existing approaches.

A. SENTIMENT ANALYSIS BASED ON CNN

Recently, sentiment analysis has become a prominent area of research in NLP tasks. Numerous studies have focused on improving the accuracy of binary sentiment analysis tasks. Sentiment analysis involves identifying the emotional tone of a text or review, categorizing it as positive, negative, or neutral. This process integrates machine learning, statistical methods, and NLP to determine the sentiment orientation of textual data. In its early stages, sentiment analysis relied heavily on supervised machine learning algorithms for classification and clustering tasks [18].

In recent years, deep learning has gained prominence in sentiment analysis, building on its success in various application domains. Advancements in this field have focused on leveraging word embeddings and exploring deep learning models for classification and clustering. Word embeddings, a key aspect of NLP, provide a structured representation of text, enabling deep learning models to generate precise predictions. These embeddings capture not only the literal meaning of words but also their contextual relationships and semantic similarities. By representing words as dense vectors, word embeddings enhance deep learning models' ability to analyze language, capturing both semantic and syntactic nuances for better understanding of complex patterns.

Yoon and Kim [19] proposed a multi-channel lexicon-based model combining convolutional neural networks (CNNs) with bidirectional LSTM (biLSTM) for sentiment classification. Their model relies on lexicon-derived sentiment orientation rules specific to the domain. CNNs, as advanced neural network architectures, are adept at identifying complex features across various data types, including text and images. Traditionally used in computer vision tasks such as image recognition [20], object detection [21], and segmentation [22], CNNs' integration into NLP showcases their versatility and effectiveness in analyzing textual data.

Although initially developed for image processing, CNNs have demonstrated exceptional capabilities in text classification tasks. CNNs are characterized by their multi-layered feed-forward architecture, which includes convolutional, pooling, and fully connected layers. The convolutional layer performs automatic feature extraction by connecting specific portions of the input to preceding layers, forming feature maps. The pooling layer reduces computational complexity by down-sampling the feature maps, commonly using max-pooling to retain the most significant information. The final fully connected layer integrates the extracted features to produce the classification output.

The application of CNNs to NLP tasks was pioneered by Collobert et al. [23] in 2011, who introduced CNNs for part-of-speech tagging. Kim [24] extended their application to sentiment classification, achieving notable accuracy. Innovations such as the Dynamic CNN (DCNN) by Kalchbrenner et al. [25], which incorporated Dynamic K-Max pooling, and Seq-CNN by Johnson and Zhang [26], which combined CNNs with Bag-of-Words (BOW), further demonstrated the versatility of CNNs. Rezaeinia et al. [27] later enhanced CNN models with enriched embeddings incorporating lexical, positional, and syntactic features, achieving remarkable results on short-text datasets.

Given the efficiency of CNNs, particularly their relatively low parameter requirements compared to other deep learning models, this study explores the use of a multi-channel CNN. The model is combined with context-independent word embeddings for sentiment analysis of online reviews. To address the challenges of data scarcity and model overfitting, the study incorporates text augmentation techniques, including synonym replacement (SR), random insertion (RI), random deletion (RD), and random swap (RS). These text augmentation strategies introduce diversity into the training data, enabling the model to learn more generalized patterns and improve its performance on unseen data. By leveraging CNNs' ability to extract localized patterns from text and enhancing the training dataset with text augmentation, the proposed approach aims to provide a robust, scalable solution for sentiment analysis tasks in real-world applications. This methodology not only reinforces the utility of CNNs in NLP but also demonstrates the critical role of data augmentation in improving model robustness and accuracy. The relevant literature reviews on CNN for sentiment analysis are briefly summarized in Table 1.

B. TEXT AUGMENTATION

Text augmentation plays a pivotal role in enhancing the performance of sentiment analysis models, particularly when addressing challenges such as limited labeled data and overfitting. By generating meaningful variations of existing text samples, this technique introduces diversity into the training data, ultimately improving model generalization and robustness. Text augmentation involves expanding a text dataset by applying various transformations to raw text data, which helps improve both the quality and diversity of the dataset. This technique is crucial for NLP applications, as it addresses issues like class imbalance and overfitting. It has proven effective in a range of NLP tasks, including text classification, sentiment analysis, and text summarization.

A more specific and widely used form of text augmentation is EDA, which involves simple, yet effective techniques such as synonym replacement, random insertion, random deletion,

TABLE 1. Summary of related works on CNN for sentiment analysis.

Study	Method	Key Findings	Challenges/ Limitation
Yoon et al. [19]	Multi-channel CNN with biLSTM	Lexicon-based sentiment orientation improves classification.	Domain-specific lexicon limits generalization.
Collobert et al. [23]	CNN for POS tagging	Introduced CNNs for NLP tasks like POS tagging.	Focused mainly on POS tagging, not sentiment.
Kim [24]	CNN for sentiment classification	CNNs achieved high accuracy for sentiment tasks.	Limited scope; not much generalization across domains.
Kalchbrenner et al. [25]	Dynamic CNN (DCNN) with Dynamic K-Max pooling	Dynamic K-Max pooling improves performance.	Requires large datasets for effective training.
Johnson and Zhang [26]	Seq-CNN with Bag-of-Words (BOW)	Combining CNN with Bag-of-Words (BOW) improved accuracy.	BOW limits context capture for complex texts.
Rezaeinia et al. [27]	Enriched CNN with lexical, positional, and syntactic features	Improved performance with lexical, positional, and syntactic features.	Complex feature integration may increase model complexity.

and random swapping. EDA has gained popularity due to its simplicity and ability to generate meaningful variations of the input text without requiring extensive computational resources. EDA techniques have been shown to significantly enhance model performance by increasing the diversity of the training data, thereby improving the model's ability to generalize to new, unseen examples. While data augmentation has been extensively used in computer vision, its application in NLP has posed unique challenges. Unlike visual data, where transformations such as flipping or rotation are relatively simple, applying augmentation to text data is more complex due to the inherent nuances and context-dependency of language. For example, synonym replacement or word swapping must preserve the original meaning of the text, making the augmentation process more delicate and context-sensitive. Nonetheless, the use of techniques like EDA has proven effective in addressing data scarcity and improving the robustness of NLP models.

By increasing the volume of training data, text augmentation can significantly enhance model performance by allowing the model to generalize better across varied input scenarios. An effective augmentation strategy ensures that the generated data maintains a balance: it must be diverse enough to introduce variability without deviating excessively from the original dataset. Striking this balance is critical, as overly similar augmented data can lead to redundancy, while overly dissimilar data may disrupt the model's ability to learn meaningful patterns, potentially leading to overfitting or reduced performance. Consequently, careful design and implementation of text augmentation strategies are vital for achieving optimal results in NLP tasks.

Text augmentation has been widely studied to improve the performance of sentiment analysis tasks and text classification. Nair et al. [28] evaluated the impact of different text augmentation techniques on the performance of DistilBERT, a lightweight transformer-based model. Their study utilized datasets of varying sizes: small (500 samples), medium (5564 samples), and large (43,934 samples). The techniques included synonym augmentation, contextual word embeddings, and back translation. The results indicated that the effectiveness of augmentation varied with dataset size, with small datasets showing up to a 20% performance improvement, while large datasets showed only a marginal improvement of 2%. This highlights the critical role of dataset size in determining the utility of augmentation techniques.

Zhang and Ran [29] introduced a novel contrastive learning framework that integrates external linguistic knowledge and adaptive augmentation. The study addressed two main limitations: the underutilization of linguistic knowledge in contrastive learning and the inefficiency of static augmentation methods. They proposed a WordNet-based positive and negative sampling strategy and a dynamic augmentation policy that adapts based on alignment and uniformity metrics. The experiments demonstrated superior performance across multiple public datasets compared to state-of-the-art methods.

Peng et al. [30] developed a self-attention-based method with prompt learning for short-text classification, particularly in Mandarin. The approach tackled challenges such as data sparsity, annotation scarcity, and class imbalance. It employed a template-based text augmentation mechanism to embed labels and transform single-task classification into multi-task learning. Experiments on datasets like CHNSenticorp and SST-2 showed significant improvements over baseline methods. Yang et al. [31] proposed a graph-based model, CGA2TC, combining contrastive learning with adaptive augmentation strategies. Their model addressed noise and suboptimal corpus-level graph structures by constructing text graphs based on word co-occurrence and document-word relationships. The adaptive strategy emphasized essential connections while mitigating noise. Their approach outperformed traditional graph models on benchmark datasets, highlighting the potential of contrastive augmentation in graph-based text classification.

Feng et al. [32] introduced Tailored Text Augmentation (TTA), designed to enhance discriminative word coverage and reduce overfitting. TTA employed probabilistic word sampling for synonym replacement based on relevance to sentiment, along with masking irrelevant words to address overfitting. The results demonstrated significant performance gains over six baseline methods in both synthetic and real-world sentiment datasets, such as public sentiment on COVID-19 policies. Bencke and Moreira [33] explored text augmentation strategies for text classification in smart cities.

They adapted techniques like back translation, paraphrasing, and generative text modeling for Portuguese datasets. The study showed significant F1-score improvements (up to 43%) in limited data scenarios. However, some methods faced challenges in maintaining semantic consistency, emphasizing the need for augmentation methods sensitive to context. Luo et al. [34] proposed a framework combining SeqGAN with sentence compression for sentiment analysis. The method leveraged LSTM-based compression and sentiment lexicons to preserve critical sentiment-related information in long texts. A data-screening mechanism further refined the augmented samples. The experiments demonstrated that this approach significantly enhanced sentiment classification performance, offering high diversity and semantic retention.

Jia et al. [35], proposed the TACLR (Text Augmentation Contrastive Learning Representation) model, which combines text augmentation techniques with contrastive learning for text classification tasks. Xiang et al [36], introduced the PLSDA (POS-Focused Lexical Substitution for Data Augmentation) model, which is based on part-of-speech (POS)-focused lexical substitution to enhance the performance of machine learning algorithms in sentiment analysis. POS information is used to identify words to be replaced and generate semantically relevant substitutes through various semantic-based substitution and sampling strategies. The relevant literature reviews on text augmentation are briefly summarized in Table 2.

TABLE 2. Summary of related works on text augmentation.

Study	Method	Key Findings	Challenges/ Limitation
Nair et al. [28]	Text augmentation on DistilBERT	Augmentation improved performance, especially on small datasets (20%).	Effectiveness varied with dataset size (marginal on larger datasets).
Zhang and Ran [29]	Contrastive learning with adaptive augmentation	Dynamic policy outperformed existing methods.	Static augmentation is inefficient.
Peng et al. [30]	Self-attention with prompt learning	Significant improvements in short-text classification.	Focused on Mandarin; may not generalize.
Yang et al. [31]	Graph-based contrastive learning	CGA2TC outperformed traditional graph models.	Computationally intensive, needs large datasets.
Feng et al. [32]	Tailored Text Augmentation (TTA)	Improved performance via probabilistic word sampling.	Relevance-based sampling may not apply universally.
Bencke and Moreira [33]	Back translation, paraphrasing, generative models	Up to 43% F1-score improvement on limited Portuguese data.	Some methods lacked semantic consistency.
Luo et al. [34]	SeqGAN + sentence compression	Enhanced sentiment classification, preserving critical sentiment info.	Challenges with data screening and long-text compression.
Jia et al. [35]	TACLR model combining augmentation and contrastive learning	Improved text classification tasks.	Requires fine-tuning for optimal results.
Xiang et al [36]	POS-focused lexical substitution for augmentation	Enhanced sentiment analysis via POS-based substitution.	Depends on accurate POS tagging, noisy data issues.

III. METHODOLOGY

This section provides a detailed explanation of the proposed model. The architecture of the proposed model is illustrated in Fig. 1. The description of each stage will be elaborated in the subsequent sections. In general, the proposed model begins with the original review data, which undergoes a text cleaning or pre-processing step. The pre-processed data is then subjected to an augmentation process. This augmentation is performed dynamically, where each review undergoes a unique augmentation process. The algorithm for the process of selecting augmentation techniques for each review can be seen in Algorithm 1. To further enrich the training dataset, the augmented data is combined with the pre-processed data. As a result, the final dataset used comprises clean data that has been enhanced through augmentation. The final dataset is then processed by an embedding layer, which prepares it as input for the MCNN. The classification process is subsequently performed using this input.

A. DATA PREPROCESSING

Text preprocessing is a critical and foundational step in natural language processing (NLP) tasks, aimed at transforming raw text into a format that can be effectively analyzed. In this research, several key steps are employed to clean and prepare the textual data for further examination. First, all non-alphabetic characters, such as punctuation marks and special symbols, are removed systematically to eliminate irrelevant elements that could distort the analysis. Additionally, single characters, which often represent noise in the text, are filtered out to prevent them from affecting the quality of the data. Any instances of multiple consecutive spaces are also collapsed into a single space, ensuring that the text maintains a consistent structure. Converting the entire text to lowercase standardizes the representation of words, regardless of their original casing, which contributes to the consistency and reliability of subsequent processes.

A crucial step in text preprocessing is tokenization, which breaks down the text into individual words or tokens, enabling easier manipulation and analysis. Following this, stopwords, commonly used words that offer little semantic value, such as ''the'' or ''and'' are removed, reducing noise and improving the efficiency of the analysis by focusing on more meaningful terms. Once these steps are completed, the remaining tokens are reassembled into a clean, coherent string, which is now ready for further modeling or analysis. These preprocessing techniques not only improve the quality of the textual data

Algorithm 1 Dynamic EDA

Input: R_p: *Original pre-processes text instance*
y: *Corresponding labels for the original text*
A: *Set of augmentation function*
n_{aug}: *Number of augmented instances to create for each original text instance*
Output: D_c: *Combined dataset containing both original and augmented text instance*

Initialize empty list:
- $R_{all} = [\ ]$ // *List to store all original and augmented text*
- $y_{all} = [\ ]$ // *List to store all corresponding labels*

For each R_{pi} *in the original dataset* R_p *do:*
  For each augmentation index $j = 1n_{aug}$ *do:*
  Randomly select an augmentation function F_a *from set* $AF_a \in A$
  Apply the selected augmentation function to the original text $R_{p,i}$:
- $R_a = F_a\left(R_{p,i}\right)$ // *Augmented text*

  Append $R_{p,i}$ *and its corresponding label* y_i *to* R_{all} *and* y_{all} :
- *Append* $R_{p,i}$ *to* R_{all}
- *Append* y_i *to* y_{all}

  Append the augmented text R_a *and its corresponding label* y_i *to* R_{all} *and* y_{all} :
- *Append* R_a *to* R_{all}
- *Append* y_i *to* y_{all}

  Combine original and augmented texts and their labels to form the enriched dataset D_c
  Return the enriched dataset D_c :
- $D_c = (R_{all}, y_{all})$

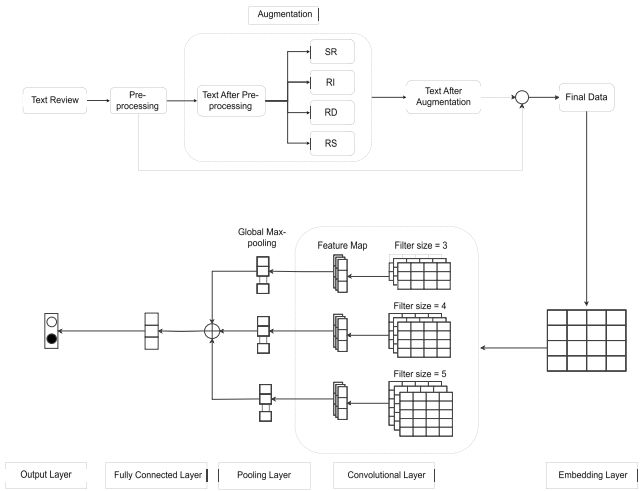

FIGURE 1. Architecture of proposed model.

but also form a vital foundation for the creation of powerful and effective NLP models across a variety of applications, such as sentiment analysis, text classification, and machine translation. By ensuring that the data is clean and structured, preprocessing maximizes the potential for accurate and insightful results in any NLP task.

Raw text often includes unnecessary elements such as punctuation, numbers, and common words (stopwords) that do not provide meaningful sentiment information. The goal of the pre-processing step is to refine the text by eliminating these irrelevant components, thereby preparing it for model training. Consider a dataset D containing a review R, where each review R is represented as a sequence of words $R = \{w_1, w_2, \ldots, w_n\}$, with n representing the length of the sequence. In this case, w_i denotes the individual words within review R. The main objective of pre-processing is to filter this word sequence R, retaining only the words that are relevant and contribute meaningful information for sentiment classification. The outcome of this process is a cleaner text representation, where stopwords and irrelevant characters have been removed. The resulting set of words after pre-processing is denoted as R_p, where:

$$R_p = \{w_i | w_i \in R \text{ and } w_i \notin Stopwords\} \tag{1}$$

B. TEXT AUGMENTATION PROCESS

Text augmentation is a technique used to expand and enhance the training dataset by generating different variations of the existing textual data. The primary goal of augmentation is to increase the diversity of the dataset, thereby improving the model's generalization capabilities and making it more robust. In this study, four augmentation techniques are employed inspired by studies [17] and [37]: Synonym Replacement (SR), Random Insertion (RI), Random Deletion (RD), and Random Swap (RS). Each of these methods introduces slight modifications to the original text, ensuring that the semantic meaning remains intact while introducing sufficient variety to the training data.

The recommended parameters for implementing EDA, as provided in the original study [17], are outlined in Table 3.

TABLE 3. Parameter recommendation.

N_{train}	α	n_{aug}
500	0.005	16
2000	0.005	8
5000	0.1	4
More	0.1	4

These parameters include:

- N_{train} : The size of the original training dataset before augmentation, which serves as a reference for evaluating the impact of augmentation.
- α (alpha): Controls the proportion of words modified during augmentation. For instance, an α value of 0.1 modifies 10% of the words in a sentence.
- n_{aug} : Determines the number of augmented sentences generated per original sentence. For example, if n_{aug}= 4, four variations are created for each sentence, significantly expanding the dataset.

For datasets with 50,000 samples, the recommended parameters are $\alpha = 0.1$ and n_{aug}= 4, ensuring a balanced augmentation process that enriches the dataset without compromising semantic integrity. In our proposed model, if R_p represents a pre-processed review, the augmentation process for each of these techniques can be described as follows:

1) SYNONYM REPLACEMENT

Synonym replacement is an augmentation method aimed at increasing the variability of the training dataset by generating modified versions of the original text. This technique involves identifying specific words within the text that can be swapped with their synonyms, thus creating semantic variations without changing the overall meaning of the text. The process begins by selecting words that have appropriate synonyms. These synonyms are retrieved from a lexical database like WordNet, which offers a broad collection of synonyms that maintain the text's contextual integrity. Afterward, a random selection of these words is chosen for substitution to ensure diversity in the augmented text.

If R_p represents the original text and w_i denotes a word in the sequence that is replaced by its synonym s_i, the resulting augmented text, R_{syn}, is expressed as $R_{syn}{=}R^{'}$ where $w_i{\rightarrow}s_i$. For multiple replacements, this process is represented as:

$$R_{syn} = R_p \text{with } \{w_i \rightarrow s_i, w_2 \rightarrow s_2, \ldots, w_n \rightarrow s_n\} \quad (2)$$

This technique enhances the dataset by introducing nuanced changes in vocabulary and sentence structure, improving the model's ability to generalize and adapt to a wider range of language patterns and expressions.

2) RANDOM INSERTION

Random insertion is a data augmentation strategy aimed at enhancing text diversity by introducing additional words at random positions within the text. This approach increases variability and exposes the model to different word arrangements, thereby improving its resilience to various linguistic structures. The process involves identifying words from the original text that hold meaningful contributions to the overall sentence. These selected words are then reinserted at arbitrary locations, generating a modified version of the text that remains semantically similar to the original.

In the proposed model, this insertion process is configured to occur twice, with a parameter value set to $n = 2$. This implies that two words from the original text are randomly chosen and reintroduced at different positions in the sentence, adding variation to the dataset. Formally, the process can be expressed as $R_{ins}{=}\ R{\cup}w_j$ at position p, where w_j represents the inserted word placed at the selected position p. By leveraging random insertion, this technique enriches the dataset, enabling the model to learn from a broader range of word sequences. This diversity enhances the model's ability to generalize effectively across varied text inputs, contributing to improved performance in sentiment analysis tasks.

3) RANDOM DELETION

Random deletion is designed to reduce the length of text by selectively removing words based on a predetermined probability. This technique generates variability by producing shorter versions of the original text, enabling the model to adapt to inputs with diverse lengths and structures. The process begins by defining a deletion probability, denoted as p, which governs the likelihood of removing each word. In this study, the value of p is set to 0.2, indicating a 20% chance of deletion for any given word. Words that contribute minimally to the overall context or sentiment are then randomly excluded according to this probability, resulting in a condensed version of the text. Formally, if w_k represents a word subject to deletion, the augmented text R_{del} can be expressed as

$$R_{del} = R_p\{w_k | probability\,(w_k) < p\} \quad (3)$$

By streamlining the text while maintaining its core meaning, random deletion creates a more concise dataset. This approach enhances the model's ability to generalize by exposing it to varying sentence structures and lengths, ultimately improving its robustness across diverse linguistic patterns.

4) RANDOM SWAP

Random Swap is designed to modify the order of words within a text by randomly swapping their positions. This technique introduces minor adjustments to the word sequence, exposing the model to variations in sentence structure while preserving the overall meaning. The process begins by selecting random indices within the text. Once identified, the words at these indices are exchanged, resulting in a rearranged version of the original text.

In this study, the swapping operation is configured to occur twice, meaning two pairs of words are interchanged for each text review. Formally, if w_m and w_n are the words chosen for swapping, the resulting augmented text R_{swap} can be expressed as:

$$R_{swap} = R_p \text{ with } w_m \leftrightarrow w_n \quad (4)$$

By incorporating random swaps, this technique enriches the training dataset with subtle variations in word order, enabling the model to focus on semantic meaning rather than rigid syntax. This approach enhances the model's capacity to generalize across diverse sentence structures, thereby improving its robustness and adaptability to various linguistic patterns.

In each step of the augmentation process, the model dynamically selects one function from the four available augmentation techniques. Let A represent the set of these augmentation functions, defined as $A = \{R_{syn}, R_{ins}, R_{del}, R_{swap}\}$, corresponding to SR, RI, RD, and RS, respectively. The augmentation process is formulated as the random selection of an augmentation function F_a from the set A, applied to the original text R_p. This process can be expressed as $R_a{=}F_a\,(R_p)$, where $F_a \in \boldsymbol{A}$ denotes the selected augmentation function, and R_a represents the resulting augmented text. The use of random selection across multiple augmentation methods introduces a dynamic and varied range of transformations into the dataset. This strategy broadens the model's exposure to diverse text variations, enhancing its ability to generalize across different linguistic structures. Once the augmentation process is completed, the original pre-processed text R_p is combined with the augmented text R_a to create an enriched dataset, referred to as D_c. This combined dataset integrates both the original and augmented instances, providing a more

comprehensive range of examples for training. Mathematically, the enriched dataset D_c can be represented as follows:

$$D_c = \left\{ \begin{array}{l} \left(R_{p,1}, y_1\right), \left(R_{p,2}, y_2\right), \ldots, \left(R_{p,N}, y_n\right), \\ \left(R_{a,1}, y_1\right), \left(R_{a,2}, y_2\right), \ldots, \left(R_{a,M}, y_M\right) \end{array} \right\} \quad (5)$$

Here, $R_{p,i}$ denotes the original text samples with their corresponding labels y_i, while $R_{a,i}$ represents the augmented text instances with the same labels. By combining these, the dataset becomes a richer resource for model training, supporting better learning and adaptability. In this equation, each pair $\left(R_{p,i}, y_i\right)$ represents an original text instance R_p with its corresponding label y_i, while each pair $\left(R_{a,i}, y_i\right)$ corresponds to an augmented text instance R_a with the same label y_i. By incorporating both the original instances R_p and the augmented instances R_a, the dataset D_c expands in size and variability. This augmentation enables the model to learn from a broader spectrum of syntactic and lexical variations, thereby improving its ability to generalize. The enhanced dataset exposes the model to both the natural structure of the original data and the augmented variations, which simulate potential real-world language diversity. Here, N represents the number of original text instances, M represents the number of augmented instances, and y_i denotes the label corresponding to each text instance R_i. This combination of data is achieved by concatenating the original and augmented text instances along with their associated labels. The process is formally expressed as follows:

$$R_{all} = \left[R_{p,1}, R_{p,2}, \ldots, R_{p,N}\right] \oplus \left[R_{a,1}, R_{a,2}, \ldots, R_{a,M}\right] \quad (6)$$

$$y_{all} = \left[y_1, y_2, \ldots, y_N\right] \oplus \left[y_1, y_2, \ldots, y_M\right] \quad (7)$$

In this context, $\oplus$ represents the concatenation operation applied to both the text instances and their labels. By performing this operation, the original and augmented text instances are merged into a single dataset, R_{all}, accompanied by their respective labels, y_{all}. This process ensures that the combined dataset D_c provides a richer and more diverse training resource, ultimately supporting the development of a more robust and adaptable model.

C. EMBEDDING LAYER

This study utilizes Word2Vec as the embedding layer, a technique that converts words into fixed-length numerical vectors. Specifically, we employ Word2Vec embeddings with a dimensionality of $d = 300$, meaning each word is encoded as a 300-dimensional vector. For a given vocabulary $V = \left\{w_1, w_2, \ldots, w_{|V|}\right\}$ and an embedding matrix W, each word w_i is represented by an embedding vector $e\left(w_i\right) \in \mathbb{R}^{300}$, as expressed by the following equation:

$$W \in \mathbb{R}^{|V| \times d}, e\left(w_i\right) = W\left[i, :\right] \quad (8)$$

In this framework, the embedding matrix W provides a unique vector representation for each word in the vocabulary. Accordingly, the embedding for a sentence $R = \{w_1, w_2, \ldots, w_n\}$ is constructed as:

$$E\left(R\right) = \left\{e\left(w_1\right), e\left(w_2\right), \ldots, e\left(w_n\right)\right\} \quad (9)$$

This structure enables the model to leverage dense vector representations of words, capturing the semantic relationships within the text. The use of 300-dimensional embeddings facilitates effective encoding of each word in the vocabulary, empowering the model to identify and differentiate subtle patterns in language, thereby enhancing its performance during training.

D. MULTI-CHANNEL CNN ARCHITECTURE

Convolutional Neural Networks (CNNs) are employed to extract textual features by applying convolutional filters of various sizes. In Multi-Channel Convolutional Neural Networks (MCNNs), multiple filters with differing dimensions are utilized to identify features across diverse scales, enabling the model to recognize patterns at varying levels of detail. The MCNN framework operates through a series of distinct stages:

1) CONVOLUTIONAL LAYER

Each filter f slides across the text embeddings to extract localized features. In this study, filters of size 3 are utilized, enabling the network to identify trigrams or small word groupings within the text. The convolution operation at a specific position i is computed as follows:

$$\left(f * E\left(R\right)\right)_i = \sum_{j=1}^{k} f_i \cdot E\left(R\right)\left[i + j - 1\right] \quad (10)$$

where:

- i denotes the position on the convolutional feature map, indicating where the result of the convolution is placed.
- k is the filter size, set to 3 in this study, representing a filter spanning three words.
- j represents each element of the filter f, applied sequentially during the operation.
- f is the convolutional filter with dimensions $k \times d$, where d is the embedding dimension.
- $E\left(R\right)\left[i + j - 1\right]$ refers to the embedding vector of the $(i + j - 1)$-th word.
- $f_i \cdot E\left(R\right)\left[i + j - 1\right]$ indicates the scalar product between the filter f and the word embedding at that position.

This operation captures valuable patterns within localized regions of the text by combining word embeddings based on the filter's size. By employing filters of varying sizes in the MCNN, the model effectively learns features from different n-gram structures, enhancing its capability to discern intricate syntactic and semantic relationships in textual data.

2) GLOBALMAX POOLING LAYER

Following the convolution process, the pooling layer is employed to extract the most prominent features from each feature map. This study utilizes GlobalMax Pooling, chosen for its ability to preserve the most significant value from the entire feature map output by the convolutional layer, regardless of the feature map's size. This method effectively compresses a 2D matrix into a single scalar per feature map,

capturing the strongest activation that represents the most critical feature across the text. GlobalMax Pooling operates by selecting the maximum value from the convolution operation $f * E(R)$, mathematically defined as:

$$p = \max(f * E(R)) \tag{11}$$

where:

- p is the maximum value extracted from the entire feature map generated by the convolutional layer.

Unlike localized pooling methods, such as sliding-window pooling, which focus on specific regions of the feature map, GlobalMax Pooling evaluates the entire feature map to identify the most salient feature globally. By condensing each feature map into its highest activation value, GlobalMax Pooling emphasizes the strongest patterns, which often correspond to essential characteristics in the text. This dimensionality reduction simplifies the representation while enhancing the model's focus on informative features, improving its ability to generalize across diverse linguistic inputs.

3) CONCATENATION LAYER

The features obtained from different filters are merged to create a richer overall feature representation, resulting in a comprehensive vector for further analysis. This concatenation process can be represented as:

$$F = Concatenate(p_1, p_2, \ldots, p_n) \tag{12}$$

where:

- F is the resulting concatenated feature vector, which combines all the individual feature maps.
- $p_1, p_2, \ldots, p_n$ are the outputs from the Global Max Pooling operation applied to the feature maps generated by each convolutional filter.

The concatenation function merges each p_i vector into a larger vector F, thereby improving the model's ability to capture various aspects of the text's features.

4) OUTPUT LAYER

At the final stage of the proposed model architecture, following the convolutional, pooling, and concatenation layers, an output layer is used for sentiment classification.

a: DENSE LAYER

After combining features from multiple filters, a comprehensive feature vector F is obtained that encapsulates the overall text representation. This vector F is then passed through a Dense (Fully Connected) layer, where the operation is represented as:

$$z = W \cdot F + b \tag{13}$$

where:

- W denotes the weights in the Dense layer,
- F is the feature vector from the pooling and concatenation steps,
- b is the bias term of the Dense layer,
- z represents the linear output from this layer.

b: SIGMOID ACTIVATION

For binary sentiment classification (positive or negative), the sigmoid activation function is applied:

$$\hat{y} = \frac{1}{1 + e^{-z}} \tag{14}$$

where $\hat{y}$ is the predicted probability of the positive class. The final output of this layer reflects the model's confidence in the sentiment class. By setting a threshold of 0.5 for the sigmoid output, the model classifies the input as 'positive' if the output exceeds 0.5, and as 'negative' if it is below 0.5. This probabilistic approach provides a clear decision boundary, making the sentiment classification easy to interpret while also allowing flexibility to adjust the threshold as needed.

IV. EXPERIMENTAL RESULT AND DISCUSSION

This section provides a detailed discussion of the results and analysis of the study, as well as the datasets utilized during the research process. Several scenarios are presented to demonstrate the superior performance of the proposed model. First, the evaluation is conducted using various deep learning approaches. Second, the impact of applying individual augmentation models to the entire dataset is examined. Third, the effect of employing different embedding models is analyzed.

A. EXPERIMENT SETUP

The experiment was conducted on Google Colaboratory. The implementation was executed using Python version 3.10.12, alongside version 2.17.0 for developing and training the model. The keras framework, version 3.4.1, was used for building the neural network architecture. For word embeddings, we employed a pre-training Word2Vec model that was trained on Google News dataset, which was accessed via the Gensim library version 4.3.3. For pre-processing text data, we utilized stop words and the WordNet lexical database from the NLTK library version 3.8.1, to ensure text inputs we filtered and standardized. The resources helped refine the text inputs by removing irrelevant words and establishing synonym-based connections that enhance the model's understanding of contextual meanings. Additionally, Pandas version 2.2.2 and Numpy version 1.26.4 were used for data manipulation, while Matplotlib version 3.7.1 and Seaborn version 0.13.2 were employed for visualizing the results, including training history, confusion matrices and ROC curves. The entire environment was optimized for efficiency and reproducibility with all relevant libraries specified for transparency.

B. DATASET

The IMDB dataset, also known as the Internet Movie Database dataset, is extensively used in the fields of data science and computational linguistics. It comprises a vast collection of movie-related data, including film details and user-generated reviews. In recent years, this dataset has been widely utilized for sentiment analysis tasks. The IMDB dataset is particularly favored for sentiment analysis due to

its large number of user reviews, each labeled with either a positive or negative sentiment. It consists of 50,000 movie reviews for text analysis [5]. Reviews are categorized as either positive or negative based on the IMDb rating system. The dataset contains full-length reviews with two primary columns: one for the review text and one for the sentiment label. We convert the sentiment labels into a binary classification, assigning 'positive' a value of 1 and 'negative' a value of 0. This transformation facilitates the processing of data for our proposed model.

C. STATISTIC OF DATASET BEFORE AND AFTER AUGMENTATION PROCESS

The original IMDB dataset contains 50,000 entries, each comprising a movie review with binary sentiment labels: positive or negative. For this research, we divided the dataset into 80% for training and 20% for testing, resulting in 40,000 training samples and 10,000 test samples. Table 4 provides detailed statistics of the data both before and after augmentation, highlighting the distribution and key characteristics that were altered through the augmentation process.

TABLE 4. The results of using different word embedding models.

Data Types	Total Training Data	Total Token	Minimum Token	Maximum Token
Original Data	40000	4873850	4	1427
After Augmentation	56000	6745723	4	1427

Out of the 50,000 original entries, it is observed that before augmentation, the training dataset consisted of 40,000 samples with a total of 4,873,850 tokens. The shortest entry contained 4 tokens, while the longest had 1,427 tokens. After the augmentation process, the training data increased by about 40%, resulting in 56,000 samples. Consequently, the total token count rose by 38.41%, reaching 6,745,723 tokens. Importantly, the minimum and maximum token counts remained unchanged before and after augmentation. This stability is attributed to the nature of the augmentation methods used, which modify the text content without significantly affecting the overall length of the entries. Therefore, while augmentation introduces variability in content, it does not alter the length structure of the longest and shortest reviews. Overall, the MCNN model proved to be the most balanced and robust, making it a strong choice for sentiment analysis tasks that require nuanced sentiment detection.

D. RESULT ON DIFFERENT TYPES OF DEEP LEARNING MODELS

This section compares the proposed model with several deep learning models for sentiment classification, evaluating performance with and without data augmentation using metrics like accuracy, precision, recall, and F1-score. In the non-augmented experiments, the biGRU model achieved the highest accuracy (88.7%) and a balanced precision-recall (88.9% and 88.65%, respectively), with an F1-score of 88.77%. The GRU model had a strong precision of 91.37%, but a lower recall (84.08%), resulting in an F1-score of 87.58%. The LSTM model showed an accuracy of 87.34%, with high recall (89.42%) but slightly lower precision (86.01%), leading to an F1-score of 87.68%. The CNN model achieved an accuracy of 87.64%, with well-balanced precision and recall (87.77% and 87.7%, respectively). The MCNN model, achieving 88.57% accuracy and 88.65% F1-score, benefiting from its multi-channel structure that captures a wider range of features. The experimental results with and without augmentation for each deep learning model are detailed in Table 5.

TABLE 5. Comparison of results with previous models.

Model	Accuracy (%)	Precision (%)	Recall (%)	F1 (%)
Without Augmentation				
LSTM	87,34	86,01	89,42	87,68
GRU	87,98	91,37	84,08	87,58
biLSTM	86,12	84,78	88,31	86,51
biGRU	88,7	88,9	88,65	88,77
CNN	87,64	87,77	87,7	87,73
MCNN	88,57	88,73	88,57	88,65
With Augmentation				
LSTM +Aug	88,16	89,04	87,24	88,13
GRU +Aug	88,43	88,22	88,91	88,56
biLSTM +Aug	85,06	82,46	89,36	85,77
biGRU+ Aug	87,64	88,5	86,74	87,61
CNN +Aug	86,72	84,55	93,39	87,64
MCNN+ Aug	88,65	85,77	92,68	89,09

In experiments with data augmentation, some models showed improvements while others did not consistently benefit. The LSTM+Aug model achieved an accuracy of 88.16% and a precision of 89.04%, though recall slightly decreased to 87.24%. This led to an F1-score of 88.13%, reflecting strong overall performance. The GRU+Aug model showed favorable results, with accuracy at 88.43%, precision at 88.22%, and recall rising to 88.91%, resulting in an F1-score of 88.56%. In contrast, the biLSTM+Aug model saw a drop in accuracy to 85.06%, despite an increase in recall to 89.36%, but with a reduced precision of 82.46%, suggesting more false positives. The biGRU+Aug model also showed decreased performance, with accuracy dropping to 87.4% and F1-score declining to 87.61%. The CNN+Aug model saw a significant improvement in recall (93.39%), but precision fell to 84.55%, leading to an F1-score of 87.64%. Finally, the MCNN+Aug model outperformed all other models with an accuracy of 88.65% and an F1-score of 89.09%, maintaining a balanced

precision of 85.77% and recall of 92.68%. This demonstrates that data augmentation can have varying effects on different models, with MCNN+Aug standing out for achieving both high accuracy and a balanced performance, making it particularly suitable for sentiment classification.

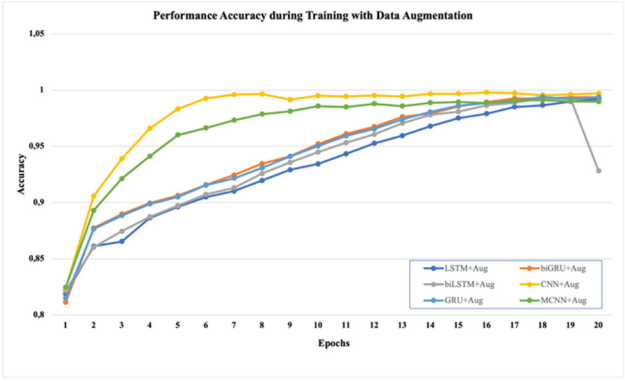

FIGURE 2. Accuracy comparison among deep learning models.

Fig. 2 illustrates the training accuracy across various deep learning models. All models show rapid accuracy improvements during the initial epochs, stabilizing after 5–10 epochs as they focus on learning more complex patterns. CNN with data augmentation achieves near-perfect accuracy quickly, demonstrating its ability to capture fundamental patterns efficiently. The proposed MCNN also performs well, with accuracy slightly below CNN by the end of training, indicating its effectiveness in extracting complex features.

RNN-based models (GRU, biGRU, LSTM and biLSTM) show slower accuracy gains, with accuracy remaining below 0.95 after 20 epochs, despite data augmentation. The biLSTM+Aug model experiences a sharp accuracy drop near the end of training, highlighting overfitting and the need for better regularization or parameter tuning.

While CNN slightly outperforms MCNN during training, Table 3 shows that MCNN surpasses CNN in test accuracy and F1-score. This is likely because CNN, being simpler, quickly optimizes for training data but risks overfitting. In contrast, MCNN's multi-channel architecture, with filters of varying sizes, captures a broader range of features and maintains better generalization, leading to superior test performance.

E. INFLUENCE OF DIFFERENT AUGMENTATION TECHNIQUES

This section provides a comparison of the proposed model with various data augmentation techniques. All Aug corresponds to the proposed model incorporating dynamic all augmentation methods. The results for each augmentation technique are presented in Table 6.

The evaluation results in the table reveal that each augmentation technique uniquely influences the model's performance in sentiment classification. SR and RS achieve a well-balanced performance between precision and recall,

TABLE 6. Result comparison among data augmentation techniques.

Augmentation Method	Accuracy (%)	Precision (%)	Recall (%)	F1(%)
SR	88,68	87,78	90,08	88,91
RI	88,51	88,5	88,73	88,61
RD	87,31	92,57	81,35	86,6
RS	88,48	86,23	91,8	88,93
All Aug	88,65	85,77	92,68	89,09

with F1-scores of 88.91% and 88.93%, respectively. This indicates that both methods enhance the model's ability to accurately detect sentiment without compromising either metric. On the other hand, RD achieves the highest precision (92.57%) among all methods but at the cost of lower recall (81.35%), suggesting that RD is more conservative in identifying positive sentiment, often missing some positive instances. In contrast, the All-Augmentation approach, which process of dynamic augmentation, delivers the highest recall (92.68%) and the best overall F1-score (89.09%). This demonstrates that integrating dynamic augmentation techniques enriches the dataset's diversity, improving the model's sensitivity to a wider range of sentiment patterns. Overall, the All-Augmentation method proves to be the most effective, offering superior performance and a balanced focus on both precision and recall, making it particularly suitable for sentiment classification tasks.

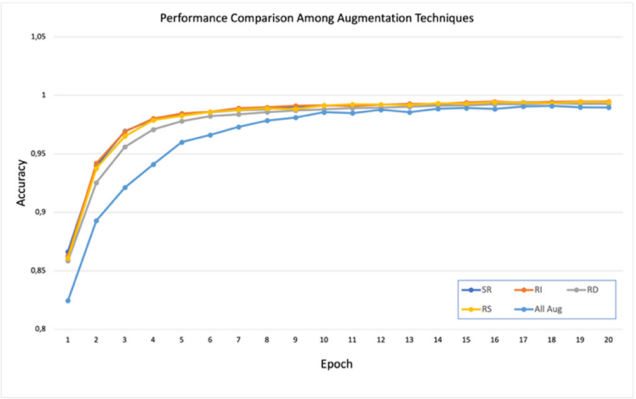

FIGURE 3. Performance comparison among augmentation techniques.

Figure 3 presents the accuracy performance of individual augmentation techniques, illustrating their effectiveness in enhancing model training. After the fifth epoch, all techniques display a more stable increase in accuracy, with values converging near the tenth epoch. Among these, the All Augmentation and RS techniques consistently achieve slightly higher accuracy, indicating their superior ability to generate diverse and meaningful variations of the training data. The All Augmentation method, in particular, stands out due to its combination-based approach, which introduces a broader range of data variations. This diversity enables the model to identify and generalize sentiment patterns more effectively,

especially during the earlier stages of training. These results underscore the importance of selecting augmentation techniques that align with the dataset's complexity and the desired model performance. Furthermore, the findings suggest that employing more sophisticated techniques like All Augmentation can lead to notable accuracy improvements, offering valuable guidance for the design and optimization of sentiment classification models.

F. INFLUENCE OF VARIOUS WORD EMBEDDING MODELS AND COMPARISON WITH PREVIOUS MODELS

To assess the influence of word embeddings on data augmentation and the MCNN model, a comparative analysis was performed using various embedding techniques. This evaluation included context-independent embeddings like Word2Vec, GloVe, and FastText, alongside context-dependent embeddings such as BERT and GPT-2, ensuring a thorough examination. By incorporating embeddings from these distinct categories, the study aimed to evaluate their respective contributions to sentiment classification performance. This analysis not only emphasizes the strengths of traditional context-independent embeddings but also investigates the added value of transformer-based embeddings like BERT and GPT-2, which offer dynamic contextual comprehension.

TABLE 7. The results of using different word embedding models.

Embedding Model	Accuracy (%)	Precision (%)	Recall (%)	F1(%)
Word2Vec	88.65	85.77	92.68	89.09
GloVe	87.43	83.82	93.01	88.18
fastText	87.28	89.1	85.18	87.09
Bert	87.73	89	86	88
GPT2	86.12	87	86	86

The results outlined in Table 7 emphasize that the choice of word embedding significantly influences the performance of the MCNN model for sentiment classification when utilizing augmentation techniques. Among the evaluated embeddings, Word2Vec emerged as the most effective, achieving the highest accuracy (88.65%) and F1-score (89.09%). This superior performance can be attributed to its predictive, context-independant methodology, which effectively captures semantic relationships critical for sentiment analysis. GloVe, while slightly trailing Word2Vec in accuracy (87.43%) and F1-score (88.18%), demonstrated the highest recall (93.01%), indicating its strength in detecting positive and negative sentiments, albeit with a tendency toward higher false positives. FastText stood out for its precision (89.1%) but lagged in recall (85.18%), suggesting a preference for accurate category prediction at the expense of missing some relevant instances.

BERT showed balanced performance across metrics, with an F1-score (88%) comparable to Word2Vec and GloVe. Its ability to interpret complex contexts is notable, but the high dimensionality of its embeddings may pose integration challenges with MCNN, requiring further optimization. Conversely, GPT-2 achieved the lowest performance, with an accuracy of 86.12% and an F1-score of 86%. This limitation likely stems from GPT-2's design as a generative model, making its embeddings less suitable for classification tasks compared to those specifically tailored for semantic analysis. Interestingly, transformer-based embeddings like BERT and GPT-2, while competitive, did not outperform traditional context-independent embeddings such as Word2Vec, GloVe, and FastText. This indicates that MCNN may not fully capitalize on the dynamic contextual information captured by transformer embeddings, which are better suited to architectures capable of handling sequential dependencies, such as RNNs or Transformers. EDA contributed to the diversity of the training data, allowing the model to learn distinctive patterns associated with each embedding. Nevertheless, the findings suggest that simpler embeddings like Word2Vec remain more efficient under the current experimental setup compared to more complex transformer-based options. This underscores the importance of aligning embedding selection with model architecture to achieve optimal results. To provide a comparison of the proposed model with several previously developed models, especially those with similar research using data augmentation in sentiment classification models, we present a comparison with previous models in Table 8.

TABLE 8. Comparison of results with previous models.

Model	Accuracy (%)	Precision (%)	Recall (%)	F1(%)
CGA2TC [31]	77.8	-	-	-
TACLR [35]	79.9	-	-	-
PLSDA [36]	82.0	-	-	-
Ours	88.65	85.77	92.68	89.09

From the table 8, it can be observed that the proposed model outperforms previous similar models. The average improvement, when compared to the three prior models, is 11.34%. There are limited studies that focus on similar models using the same dataset as employed in this research. Therefore, we sought studies utilizing similar domains with comparable models. Among the previous models we compared, CGA2TC and TACLR used the Movie Review dataset [39], while PLSDA used the same dataset as the proposed model, namely the IMDB dataset.

V. CONCLUSION

This study introduces a novel approach for sentiment analysis on English product reviews, developed using data augmentation techniques, Convolutional Neural Networks (CNNs), and context-independent word embeddings. The proposed approach enhanced classification performance by leveraging the strengths of each model component. Easy Data Augmentation (EDA) techniques were employed to improve training data diversity through synonym replacement, random insertion, random deletion, and random swapping, while

preserving semantic meaning. These techniques are designed to reduce overfitting risks and enhance model generalization. The proposed model incorporates a dynamic augmentation process, where each product review undergoes a distinct augmentation procedure.

This augmentation strategy demonstrated significant performance improvements compared to models without augmentation, as evidenced by its application across various deep learning architectures. Word2Vec embeddings were utilized to represent words in a continuous vector space, enabling the model to capture semantic relationships between words more effectively than traditional methods like Bag-of-Words. Experimental results showed that context-independent word embeddings, particularly Word2Vec, are well-suited for data augmentation and multi-channel architectures. The CNN-based deep learning model was designed with a multi-channel architecture, leveraging multiple kernel sizes in the convolutional layers. This design allows the model to effectively capture both local patterns and critical context within the text. The proposed model, employing dynamic augmentation, achieved a commendable performance with an accuracy of 88.65% and an F1-Score of 89.09%. Although the proposed model demonstrated strong performance across various scenarios, future research is recommended to further enhance its capabilities.

Studies could explore datasets from diverse domains, including product reviews and social media posts, and incorporate additional augmentation techniques such as back-translation. Moreover, exploring Generative Adversarial Networks (GANs) for generating synthetic data could provide a means to enhance the augmentation process by generating more realistic and diverse samples. Additionally, exploring transformer-based models like BERT could offer potential improvements in performance. Furthermore, evaluating the model's real-time response time and processing capabilities is suggested as a promising area for future work, particularly for applications requiring real-time sentiment analysis.

REFERENCES

[1] P. Chauhan, N. Sharma, and G. Sikka, "The emergence of social media data and sentiment analysis in election prediction," *J. Ambient Intell. Humanized Comput.*, vol. 12, no. 2, pp. 2601–2627, Feb. 2021, doi: 10.1007/s12652-020-02423-y.

[2] R. K. Behera, M. Jena, S. K. Rath, and S. Misra, "Co-LSTM: Convolutional LSTM model for sentiment analysis in social big data," *Inf. Process. Manage.*, vol. 58, no. 1, Jan. 2021, Art. no. 102435, doi: 10.1016/j.ipm.2020.102435.

[3] L. Yang, Y. Li, J. Wang, and R. S. Sherratt, "Sentiment analysis for e-commerce product reviews in Chinese based on sentiment lexicon and deep learning," *IEEE Access*, vol. 8, pp. 23522–23530, 2020, doi: 10.1109/ACCESS.2020.2969854.

[4] K. W. Trisna, J. Huang, H. Liang, and E. M. Dharma, "Fusion text representations to enhance contextual meaning in sentiment classification," *Appl. Sci.*, vol. 14, no. 22, p. 10420, Nov. 2024, doi: 10.3390/app142210420.

[5] T. Shaik, X. Tao, C. Dann, H. Xie, Y. Li, and L. Galligan, "Sentiment analysis and opinion mining on educational data: A survey," *Natural Lang. Process. J.*, vol. 2, Mar. 2023, Art. no. 100003, doi: 10.1016/j.nlp.2022.100003.

[6] D. Zeng, Y. Dai, F. Li, J. Wang, and A. K. Sangaiah, "Aspect based sentiment analysis by a linguistically regularized CNN with gated mechanism," *J. Intell. Fuzzy Syst.*, vol. 36, no. 5, pp. 3971–3980, May 2019, doi: 10.3233/jifs-169958.

[7] K. W. Trisna and H. J. Jie, "Deep learning approach for aspect-based sentiment classification: A comparative review," *Appl. Artif. Intell.*, vol. 36, no. 1, Dec. 2022, Art. no. 2014186, doi: 10.1080/08839514.2021.2014186.

[8] A. Onan, "Sentiment analysis on product reviews based on weighted word embeddings and deep neural networks," *Concurrency Comput., Pract. Exper.*, vol. 33, no. 23, Dec. 2021, Art. no. e5909, doi: 10.1002/cpe.5909.

[9] I. Chaturvedi, E. Ragusa, P. Gastaldo, R. Zunino, and E. Cambria, "Bayesian network based extreme learning machine for subjectivity detection," *J. Franklin Inst.*, vol. 355, no. 4, pp. 1780–1797, Mar. 2018, doi: 10.1016/j.jfranklin.2017.06.007.

[10] K. W. Trisna, J. Huang, H. Lei, and E. Muntina, "From context-independent embedding to transformer: Exploring sentiment classification in online reviews with deep learning approaches," *J. Theor. Appl. Inf. Technol.*, vol. 102, no. 19, pp. 6980–7003, Sep. 2024.

[11] F. Xu, Z. Pan, and R. Xia, "E-commerce product review sentiment classification based on a naïve Bayes continuous learning framework," *Inf. Process. Manage.*, vol. 57, no. 5, Sep. 2020, Art. no. 102221, doi: 10.1016/j.ipm.2020.102221.

[12] A. Borg and M. Boldt, "Using VADER sentiment and SVM for predicting customer response sentiment," *Expert Syst. Appl.*, vol. 162, Dec. 2020, Art. no. 113746, doi: 10.1016/j.eswa.2020.113746.

[13] M. Syamala and N. Nalini, "A filter based improved decision tree sentiment classification model for real-time Amazon product review data," *Int. J. Intell. Eng. Syst.*, vol. 13, no. 1, pp. 191–202, Feb. 2020, doi: 10.22266/ijies2020.0229.18.

[14] L. Wang, X.-K. Wang, J.-J. Peng, and J.-Q. Wang, "The differences in hotel selection among various types of travellers: A comparative analysis with a useful bounded rationality behavioural decision support model," *Tourism Manage.*, vol. 76, Feb. 2020, Art. no. 103961, doi: 10.1016/j.tourman.2019.103961.

[15] L. Dang, C. Wang, H. Han, and Y.-E. Hou, "A hybrid BiLSTM-ATT model for improved accuracy sentiment analysis," in *Proc. IEEE 24th Int. Conf. High Perform. Comput. Commun.; 8th Int. Conf. Data Sci. Syst.; 20th Int. Conf. Smart City; 8th Int. Conf. Dependability Sensor, Cloud Big Data Syst. Appl.*, Hainan, China, Dec. 2022, pp. 2182–2188, doi: 10.1109/HPCC-DSS-SMARTCITY-DEPENDSYS57074.2022.00323.

[16] R. M. Samant, M. R. Bachute, S. Gite, and K. Kotecha, "Framework for deep learning-based language models using multi-task learning in natural language understanding: A systematic literature review and future directions," *IEEE Access*, vol. 10, pp. 17078–17097, 2022, doi: 10.1109/ACCESS.2022.3149798.

[17] J. Wei and K. Zou, "EDA: Easy data augmentation techniques for boosting performance on text classification tasks," 2019, *arXiv:1901.11196*.

[18] B. AlBadani, R. Shi, and J. Dong, "A novel machine learning approach for sentiment analysis on Twitter incorporating the universal language model fine-tuning and SVM," *Appl. Syst. Innov.*, vol. 5, no. 1, p. 13, Jan. 2022, doi: 10.3390/asi5010013.

[19] J. Yoon and H. Kim, "Multi-channel lexicon integrated CNN-BiLSTM models for sentiment analysis," in *Proc. Conf. Comput. Linguistics Speech Process.*, 2017, pp. 244–253.

[20] J. Xiong, D. Yu, S. Liu, L. Shu, X. Wang, and Z. Liu, "A review of plant phenotypic image recognition technology based on deep learning," *Electronics*, vol. 10, no. 1, p. 81, Jan. 2021, doi: 10.3390/electronics10010081.

[21] H. Fujiyoshi, T. Hirakawa, and T. Yamashita, "Deep learning-based image recognition for autonomous driving," *IATSS Res.*, vol. 43, no. 4, pp. 244–252, Dec. 2019, doi: 10.1016/j.iatssr.2019.11.008.

[22] F. Yuan, Z. Zhang, and Z. Fang, "An effective CNN and transformer complementary network for medical image segmentation," *Pattern Recognit.*, vol. 136, Apr. 2023, Art. no. 109228, doi: 10.1016/j.patcog.2022.109228.

[23] R. Collobert, J. Weston, L. Bottou, M. Karlen, K. Kavukcuoglu, and P. Kuksa, "Natural language processing (almost) from scratch," 2011, *arXiv:1103.0398*.

[24] Y. Kim, "Convolutional neural networks for sentence classification," 2014, *arXiv:1408.5882*.

[25] N. Kalchbrenner, E. Grefenstette, and P. Blunsom, "A convolutional neural network for modelling sentences," 2014, *arXiv:1404.2188*.

[26] R. Johnson and T. Zhang, "Effective use of word order for text categorization with convolutional neural networks," 2014, *arXiv:1412.1058*.

[27] S. M. Rezaeinia, R. Rahmani, A. Ghodsi, and H. Veisi, "Sentiment analysis based on improved pre-trained word embeddings," *Expert Syst. Appl.*, vol. 117, pp. 139–147, Mar. 2019, doi: 10.1016/j.eswa.2018.08.044.
[28] A. R. Nair, R. P. Singh, D. Gupta, and P. Kumar, "Evaluating the impact of text data augmentation on text classification tasks using DistilBERT," *Proc. Comput. Sci.*, vol. 235, pp. 102–111, Mar. 2024, doi: 10.1016/j.procs.2024.04.013.
[29] S. Zhang and N. Ran, "Contrastive learning based on linguistic knowledge and adaptive augmentation for text classification," *Knowl.-Based Syst.*, vol. 300, Sep. 2024, Art. no. 112189, doi: 10.1016/j.knosys.2024.112189.
[30] B. Peng, K. Han, L. Zhong, S. Wu, and T. Zhang, "A head-to-head attention with prompt text augmentation for text classification," *Neurocomputing*, vol. 595, Aug. 2024, Art. no. 127815, doi: 10.1016/j.neucom.2024.127815.
[31] Y. Yang, R. Miao, Y. Wang, and X. Wang, "Contrastive graph convolutional networks with adaptive augmentation for text classification," *Inf. Process. Manage.*, vol. 59, no. 4, Jul. 2022, Art. no. 102946, doi: 10.1016/j.ipm.2022.102946.
[32] Z. Feng, H. Zhou, Z. Zhu, and K. Mao, "Tailored text augmentation for sentiment analysis," *Expert Syst. Appl.*, vol. 205, Nov. 2022, Art. no. 117605, doi: 10.1016/j.eswa.2022.117605.
[33] L. Bencke and V. P. Moreira, "Data augmentation strategies to improve text classification: A use case in smart cities," *Lang. Resour. Eval.*, vol. 58, no. 2, pp. 659–694, Jun. 2024, doi: 10.1007/s10579-023-09685-w.
[34] J. Luo, M. Bouazizi, and T. Ohtsuki, "Data augmentation for sentiment analysis using sentence compression-based SeqGAN with data screening," *IEEE Access*, vol. 9, pp. 99922–99931, 2021, doi: 10.1109/ACCESS.2021.3094023.
[35] O. Jia, H. Huang, J. Ren, L. Xie, and Y. Xiao, "Contrastive learning with text augmentation for text classification," *Appl. Intell.*, vol. 53, no. 16, pp. 19522–19531, Aug. 2023, doi: 10.1007/s10489-023-04453-3.
[36] R. Xiang, E. Chersoni, Q. Lu, C. Huang, W. Li, and Y. Long, "Lexical data augmentation for sentiment analysis," *J. Assoc. Inf. Sci. Technol.*, vol. 72, no. 11, pp. 1432–1447, Nov. 2021, doi: 10.1002/asi.24493.
[37] A. S. Karnyoto, C. Sun, B. Liu, and X. Wang, "Augmentation and heterogeneous graph neural network for AAAI2021-COVID-19 fake news detection," *Int. J. Mach. Learn. Cybern.*, vol. 13, no. 7, pp. 2033–2043, Jul. 2022, doi: 10.1007/s13042-021-01503-5.
[38] A. L. Maas, R. E. Daly, P. T. Pham, D. Huang, A. Y. Ng, and C. Potts, "Learning word vectors for sentiment analysis," in *Proc. 49th Annu. Meeting Assoc. Comput. Linguistics, Hum. Lang. Technol.*, Portland, OR, USA, D. Lin, Y. Matsumoto, and R. Mihalcea, Eds., Jun. 2011, pp. 142–150. Accessed: Oct. 23, 2024. [Online]. Available: https://aclanthology.org/P11-1015
[39] B. Pang and L. Lee, "Seeing stars: Exploiting class relationships for sentiment categorization with respect to rating scales," in *Proc. 43rd Annu. Meeting Assoc. Comput. Linguistics*, Ann Arbor, MI, USA, 2005, pp. 115–124, doi: 10.3115/1219840.1219855.

KOMANG WAHYU TRISNA received the B.Eng. degree in informatics engineering from Universitas Atma Jaya Yogyakarta, Yogyakarta, Indonesia, in 2007, and the M.Eng. degree in information technology from Gadjah Mada University, Yogyakarta, in 2015. She is currently pursuing the Ph.D. degree with the School of Computer Science, Harbin University of Science and Technology, Harbin, China. Her research interest includes natural language processing, sentiment analysis, and deep learning.

JINJIE HUANG received the B.S. and M.S. degrees in automatic control from Harbin University of Science and Technology, Harbin, China, in 1990 and 1997, respectively, and the Ph.D. degree in control theory and control engineering from Harbin Institute of Technology, China, in 2004. He is currently a Professor with the School of Automation, Harbin University of Science and Technology. His research interests include natural language processing, pattern recognition, artificial intelligence, and complex system control.

YUANJIAN CHEN was born in Shandong, Zaozhuang, China. He received the B.S. degree from Shandong University of Science and Technology, Shandong, China, and the M.S. degree from Shandong Normal University, Shandong. He is currently pursuing the Ph.D. degree with the School of Computer Science and Technology, Harbin University of Science and Technology, China. His research interests include audio signal processing, sound understanding, and machine learning.

I GEDE JULIANA EKA PUTRA (Member, IEEE) received the B.Eng. and M.Eng. degrees in electrical engineering from Udayana University, Bali, Indonesia. He is currently a Lecturer with the Department of Informatics, Primakara University, Bali. His research interests include innovation and technology management, smart villages, technology adoption, and computer networks.