Advanced deep learning techniques for sentiment analysis: combining Bi-LSTM, CNN, and attention layers

Asmaa Sami Mirdan [a,1,*], Selim Buyrukoğlu [b,2], Mohammed Rashad Baker [c,3]

[a] College of Computer Science and Information Technology, Kirkuk University, Kirkuk, Iraq
[b] Computer Engineering, Faculty of Engineering, Cankiri Karatekin University, Cankiri, Turkey
[c] Software department, College of Computer Science and Information Technology, Kirkuk University, Kirkuk, Iraq
[1] asmaa.sami50@uokirkuk.edu.iq; [2] sbuyrukoglu@karatekin.edu.tr; [3] mohammed.rashad@uokirkuk.edu.iq
* corresponding author

ARTICLE INFO

Article history
Received November 15, 2024
Revised February 5, 2025
Accepted February 6, 2025
Available online February 28, 2025

Keywords
Sentiment Analysis (SA)
Deep learning
Customer engagement
Product reviews
Bi-LSTM architecture

ABSTRACT

Online platforms enhance customer engagement and provide businesses with valuable data for predictive analysis, critical for strategic sales forecasting and customer relationship management. This study explores in depth the potential of sentiment analysis (SA) to enhance sales forecasting and customer retention for small and large businesses. We collected a large dataset of product review tweets, representing a rich consumer sentiment source. We developed an artificial neural network based on a dataset of product review tweets that captures both positive and negative sentiments. The core of our model is Bi-LSTM (Bidirectional Long Short-Term Memory) architecture, enhanced by an attention mechanism to capture relationships between words and emphasize key terms. Then, a one-dimensional convolutional neural network with 64 filters of size 3x3 is applied, followed by Average_Max_Pooling to reduce the feature map. Finally, two dense layers classify the sentiment as positive or negative. This research provides significant benefits and contributions to sentiment analysis by accurately identifying consumer sentiment in product review tweets. The proposed model that integrated Bi-LSTM with attention mechanism and CNN detects negative sentiment with a precision of 0.97, recall of 0.98, and F1-score of 0.98, allowing companies to address customer concerns, improving satisfaction and brand loyalty proactively. In addition, the proposed model presents a better sentiment classification on average for both positive and negative sentiments, and accuracy (96%) compared to the other baselines. It ensures high-quality input data by reducing noise and inconsistencies in product review tweets. Moreover, the dataset collected in this study serves as a valuable benchmark for future research in sentiment analysis and predictive analytics.

1. Introduction

E-commerce has revolutionized how people shop and has become an essential daily life for many of them [1]. With the rapid growth of e-commerce, understanding user behavior has become crucial for businesses to provide a personalized and seamless experience for their customers. Most of the time, a consumer buys or cancels a product based on reviews alone. Thus, surveys are helpful [2]. However, it can be challenging to sift through hundreds of reviews whenever someone considers purchasing a product [3]. As a result, extracting some relevant information from these reviews would be useful. On the other hand, every business organization relies on an intelligent decision-making analytics system to analyze consumer behavior [4]. This analysis and forecasting can greatly affect an organization's demand-driven supply chain management. Data analysts use various tools, including machine learning (ML) and

data mining techniques, to find hidden patterns in consumer behavior and forecast sales. The latest data mining and artificial intelligence strategy has paved the way to uncover hidden layers because traditional analysis methods cannot match the speed of data generated by current e-commerce sites [5]. Now, everything has changed with Artificial Intelligence (AI). The applications of AI have been widely researched in areas such as business [6], sentiment analysis (SA) [7], prediction of credit card fraud transactions [8], and prediction of colon cancer [6]. Natural Language Processing (NLP) techniques and deep learning algorithms effectively analyze user behavior in e-commerce platforms [9]. Using these techniques, businesses can gain insight into customer preferences, purchasing patterns, and other valuable information that can help them improve marketing strategies, enhance customer satisfaction, and increase revenue. In this paper, we examine the application of NLP techniques and deep learning models in analyzing user behavior in e-commerce platforms and discuss their advantages and limitations, all of which aim to satisfy the customer and keep them coming back to a particular online store. This paper presents a deep learning model based on Bi-LSTM, attention mechanism, and one-dimensional CNN. Then, the feature map is reduced by applying max pooling, and the output is fed into two dense layers to obtain the final output. This novel approach outperforms traditional methods by effectively capturing contextual relationships between words. The research also contributes to a robust preprocessing pipeline that ensures high-quality input data by reducing noise and inconsistencies in product review tweets. Beyond academia, the findings have practical applications in the field of business intelligence, enabling companies of all sizes to automate customer sentiment monitoring, improve marketing strategies, and enhance decision-making. This study bridges the gap between sentiment analysis and strategic business applications, demonstrating how AI-driven insights can drive market competitiveness. The paper is organized as follows: a set of previous studies that dealt with the topic of sentiment analysis will be reviewed, then the details of the proposed methodology in this paper will be reviewed from the pre-processing processes, the structure of the proposed model, and performance metrics. Finally, the results are reviewed and compared with machine learning algorithms, and conclusions and recommendations will be drawn.

2. Literature Review

This section discusses data mining, natural language processing, and artificial intelligence techniques for user engagement in social networks such as Twitter and predicting customer sentiment in online shopping.

In recent years, due to the growth of the Internet space, the sharing of sentiments by users on social networks, blogs, product review pages, and websites for online sales of products and services has increased dramatically [10]. This problem has led to a new field of work in NLP called opinion mining. Opinion mining, also known as SA, analyzes people's feelings, emotions, attitudes, and opinions about products, services, organizations, people, issues, events, topics, and their characteristics. Opinion mining, as one of the most important modern methods in data mining, involves understanding and extracting human sentiments from simple and unstructured text data. Opinion analysis is a sub-branch of text analysis that analyzes and examines written language to extract theoretical and emotional expressions, expressed tendencies, and emotions [11]. Accordingly, opinion polls can be applied in various fields, such as examining customer opinions, business intelligence, marketing promotion, evaluating services and products, and analyzing political and sports opinions, transactions, social media, and interactions to extract trends and opinions.

Sentiment analysis (SA), also known as opinion mining, is a natural language processing technique that identifies and extracts subjective information from text data. The goal is to identify the attitude, opinions, or sentiments expressed in a piece of text, especially to ascertain whether the writer's attitude toward a particular topic is positive, negative, or neutral [12]. The early work in SA focused on document-level classification, categorizing whole documents as expressing positive or negative sentiment [13]. ML methods like support vector machines and naive Bayes were applied to document feature vectors. However, this ignored sentiment towards specific entities and topics. Later research moved

towards more granular aspect-based SA [14]. This involves extracting opinion targets and sentiment polarity towards each target. For example, a smartphone review may express positive sentiment towards screen quality but negative sentiment towards battery life.

Zhou *et al.* [15] presents a novel pre-trained sentiment model called SentiX. The primary goal is to enhance sentiment analysis across domains by addressing overfitting and leveraging invariant sentiment knowledge. SentiX is trained on product review datasets, extracting sentiment features such as emojis, sentiment words, and ratings. The authors propose training tasks at the symbol and sentence levels to capture these features without human annotation. SentiX achieves superior performance over BERT by learning invariant sentiment representations. Notably, the model performs well with only 1% of labeled samples, making it highly effective when the number of data samples is low.

Li *et al.* [16] proposes a novel approach called DualGCN to enhance aspect-based sentiment analysis (ABSA). ABSA is a sentiment classification task that aims to identify the sentiment of a particular aspect in a sentence. Previous approaches using graph convolutional networks (GCNs) based on dependency trees have limitations, such as inaccuracy in parsing and spoken language expressions. The authors address these challenges by introducing a dual model that includes both syntactic and semantic information. The authors propose two modules; the first module, SynGCN, is used to analyze syntactic dependencies and capture structural connections between words, and the second module, SemGCN, applies self-attention to capture semantic connections between words, which is useful for capturing patterns in spoken language. Experiments on three public datasets (restaurant, laptop, and Twitter) show that DualGCN outperforms state-of-the-art methods, proving particularly useful for complex or spoken language-based reviews.

Zhang *et al.* [17] presents a novel approach to Aspect-Based Sentiment Analysis (ABSA) through a generative framework. This paper's generative approach encodes input sentences and natural language tags into a uniform format, facilitating broader and more flexible application across different ASA tasks without task-specific designs. The authors propose two modeling models, one based on annotation style where tags (aspect, opinion, and sentiment) are embedded into the sentence during training. The other is based on extraction style modeling where aspect, opinion, and sentiment terms are treated as targets in a more direct extraction manner. Results show that this approach achieves new performance improvements with significant improvements in complex tasks.

Deep models outperform shallow models in feature extraction, using hidden layers to improve computational efficiency and reduce parameters. LSTMs excel at processing long sentences but are slower than CNNs, which require fewer hyperparameters [18]. Hybrid approaches that combine multiple approaches address the limitations of the individual method, as demonstrated in Osman *et al.* [19], Azevedo *et al.* [20] in Sentiment Analysis Improvements. Ouni *et al.*[18] evaluate three models, CNN, LSTM, and SVM, and explore their feature extraction and text classification capabilities using eight datasets, including tweets and reviews. The methodology includes embedding layers, convolutional layers, and LSTM for classification, with detailed performance metrics for different datasets provided, demonstrating the effectiveness of hybrid models in sentiment analysis.

3. Method

Based on previous studies [21]–[27], the proposed methodology is based on building a deep learning model using Python to detect the underlying sentiments within product reviews. Fig. 1 represents the general structure of the proposed methodology.

3.1. Data Collection

This research focuses on a product review dataset consisting of tweets from Twitter, which were collected and classified using a dictionary-based approach. The dataset used includes 1,048,576 product review tweets written in unstructured natural language. The data was collected through the Twitter API, which provides access to a wide range of data related to tweets, including the tweet's content, the user who posted it, the timestamp, and metadata such as the number of likes, retweets, and replies. However,

a significant challenge with this dataset is the potential bias in sentiment caused by the imbalance between positive and negative data, as there is a much larger amount of positive data than negative, which will need to be addressed for us to train the model on a dataset that accurately reflects the distribution of real-world sentiment so that it can generalize well when analyzing new tweets.

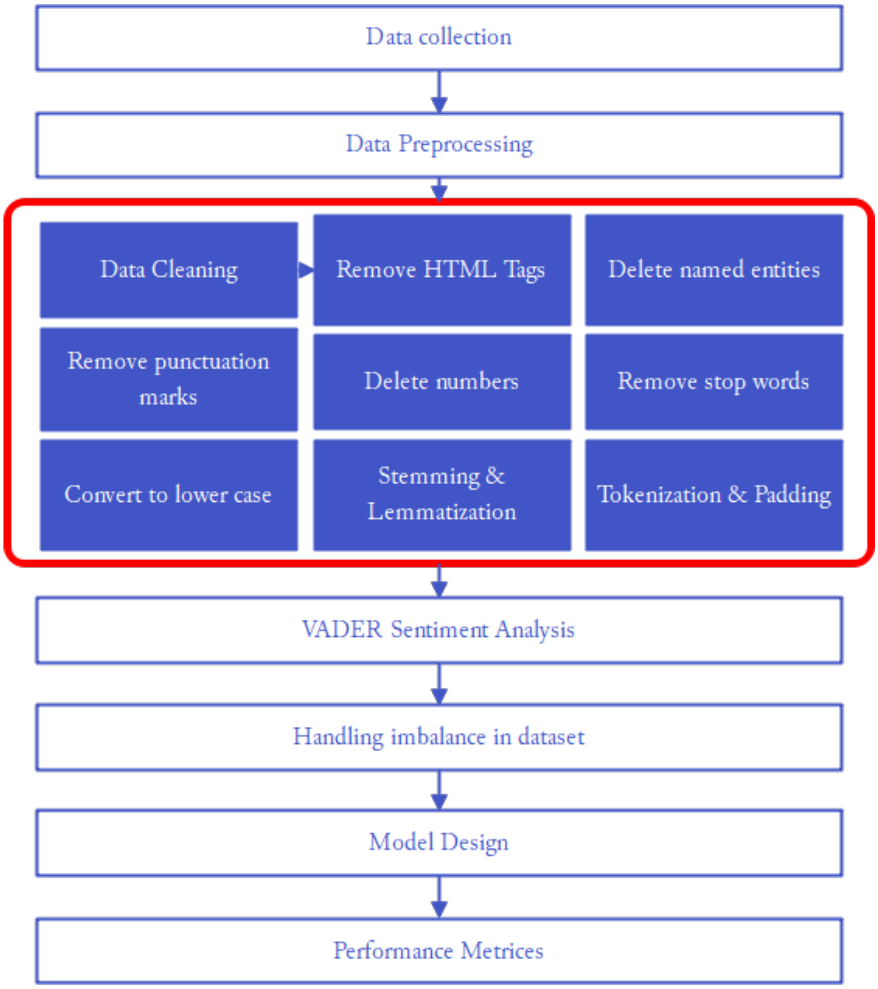

Fig. 1. The general structure of the proposed methodology

3.2. Data Processing

Data processing ensures high-quality input for the proposed models. This phase involves several preprocessing techniques to clean, normalize, and structure textual data, enhancing model accuracy and efficiency.

1. **Data Cleaning**. Social media data, especially tweets from Twitter, often contain many words and characters that are not useful for data analysis [28]. For instance, tweets may include content like "@safemoonjustv?, hilari and educ," which contain irrelevant elements such as "@" and "?". Data cleaning techniques, combined with regular expressions (regex), can identify and eliminate these unnecessary characters, thereby improving the overall quality of the dataset.
2. **Remove HTML tags**. The unstructured text frequently includes much noise, particularly when methods like web or page scraping are employed. HTML tags are commonly found elements that contribute little to the comprehension and analysis of the text. Consequently, we eliminate these unnecessary tags in this section while preserving the relevant textual content across all documents.
3. **Delete Named Entities**. In text documents, certain terms represent specific institutions and carry unique, informative significance. These are known as named entities, referring to real-world objects such as people, places, organizations, and more, which are often identified by proper names. Since

these entities do not contribute meaningful insights for sentiment analysis (SA), we remove them in this section.

4. **Remove Punctuation Marks**. We remove all sets of punctuation marks including [~{|}'_[\]@?<=>;:/.-,+*()'&%$# "!] Because these signs are seen in all sentences, removing them will lead to positive results.
5. **Delete Numbers**. During text preprocessing, we remove the numbers in the text data that are not related to the text analysis and do not lead to the production of meaningful information.
6. **Remove Stop Words**. Stop words are actually words that are commonly used. Words that are meaningless or have no special meaning, especially when semantic features are extracted from the text. These items are usually very frequent in the text, and usually, these words include adjectives, conjunctions, additions, and such. Some examples of stop words include and, the, an, a, and the like. During NLP, there is no tendency for these types of words to occupy space or take up valuable processing time. For this reason, these words can be easily removed in this section. Prepositions, conjunctions, adjectives, slank words, pronouns, and many more words are quite useful. These terms are typically seen combined with the primary word; therefore, it is not distinctive and has no special meaning. A stop word or stop list is a list of words that do not contribute much to analytical content.
7. **Convert to Lower Case.** In this subsection, all the letters in the text data are converted to lowercase letters.
8. **Stemming and Lemmatization.** In every language, words will appear differently according to their role in sentences. However, considering that all of them are made from the same root, they will help us in terms of meaning and concept. Therefore, in many NLP-based methods, we must first find the root of the words. The act of rooting makes it possible to convert different word forms into a single form. With this, the number of features is reduced, the different forms of a word are removed, and the computer can consider different forms of a word as one. The process of returning words to their root form is called the operation of lexical rooting and semantic rooting, so these two methods are usually used to find the root of words. There are different algorithms to perform lexical rooting. Porter's algorithm is very famous in English. According to a series of regular rules, this algorithm can obtain the roots of words with good accuracy. Methods can also do semantic rooting. In this practice, it is necessary to use a dictionary or something similar to obtain the roots of words because the methods of finding semantic roots are generally not regular.
    - Lemmatization: This is a process in NLP where words are reduced to their base or root form, called 'lemma.' It helps in standardizing words to their canonical form, which is linguistically correct. For example, the words "running," "runs," and "ran" would all be converted to their base form "run." This process is crucial in SA, as it allows for the grouping and analysis of similar sentiments expressed through variations of a word. While it doesn't directly handle slang or non-standard terminology, it helps to unify different forms of standard words, thereby enhancing the accuracy of the SA.
    - Stemming: Stemming is a process where words are shortened by removing their prefixes or suffixes. This process aims to reduce a word to its stem or root form, which may not necessarily be a valid word on its own. For instance, stemming could reduce the word influences to the simpler form "influence". Search engines and other text analysis tools often use stemming to improve the efficiency and relevancy of their results
9. **Tokenization and Padding.** Tokenization is breaking down text or a string of characters into smaller units called tokens. Depending on the task context, these tokens can be words, phrases, symbols, or any other meaningful units. Tokenization is fundamental in natural language processing (NLP) and computational linguistics. It is the basis for various text analysis tasks such as text classification, named entity recognition, sentiment analysis, and machine translation. After tokenization, each word is assigned a number. Padding is adding special characters (usually with the value 0) to sequences to make them equal in length. The purpose of this process is that many machine learning models,

especially neural networks, require fixed-size input sequences. The padding ensures that all sequences in the dataset are the same length. For example, let's say we have two sentences: ["The", "quick", "brown"] and ["Fox"]. If the model expects sequences of length 5, the first sequence will be padded to [1, 2, 3, 0, 0], and the second sequence will be padded to [4, 0, 0, 0, 0].

3.3. VADER Sentiment Analysis

VADER, or Valence Aware Dictionary and Sentiment Reasoner, is a lexicon and rule-based SA tool that performed exceptionally well on social media SA according to [29]. A distinctive feature of VADER is that it eschews polarity from the document scoring process, instead providing positive, negative, neutral, and compound scores. An advantage of VADER is that it doesn't require training data, thus enabling us to apply it to previously unseen data.

This system is capable of detecting both the polarity (positive/negative) and intensity (strength) of emotion. It is included in the NLTK (Natural Language Toolkit) package and can be applied to unlabeled text data without any preprocessing. To enhance the accuracy of SA, they assign values to each word based on whether it is positive, negative, or neutral. For instance, on a numeric scale, the positive word 'good' was assigned an emotional weight of 0.52. Adding an intensifier such as 'so' increased the score by 0.61, compared to when only the word 'good' was used. Conversely, the word 'bad' was given a negative score of -0.48, as it is typically associated with negative sentiments. This scoring approach better reflects the emotional weight each word carries. VADER categorized tweets about AI assistants into positive, negative, and neutral groups, and assessed the sentiment of each document matrix. If a tweet contained no positive or negative terms, the matrix would register a score of 'zero'. Additionally, the system could provide a percentage indicating how many of the words used were positive, negative, or neutral.

3.4. Handling Imbalance in Dataset

Imbalanced datasets are a major challenge in AI in general. Imbalance in a dataset occurs when a dataset contains a class with a significantly larger number of samples than other classes. This can lead to a number of problems [30], such as:

- Biased models: In this case, the model is biased towards the majority class, in other words, the model prioritizes the majority class and learns poorly from the minority classes, resulting in poor model performance.
- Unreliable performance metrics: Performance metrics such as accuracy can be unreliable if the data is unbalanced, as they can give high results, but these results do not represent the true performance of the model. The reason for this is that most of the predictions for the majority class will be correct.
- Overfitting: The model may focus on memorizing the characteristics and patterns of the majority class, resulting in poor generalization of the model to the minority class.

Therefore, at this stage, the crucial step involves correcting the defect in the data set. Fig. 2 shows the number of samples for each class.

Handling unbalanced datasets often involves resampling techniques, which are generally divided into two categories: Undersampling and Oversampling. Oversampling is usually preferred over undersampling techniques because undersampling will result in missing a number of samples that may contain valuable information. To address this problem in our dataset, we used the Synthetic Minority Oversampling Technique (SMOTE), an oversampling technique that produces synthetic samples for the minority class [31]. SMOTE helps overcome the imbalance between classes by generating synthetic examples for the minority class, which improves the model's performance on unbalanced datasets. Fig. 3 shows the number of samples for each class after applying the oversampling process using the SMOTE method.

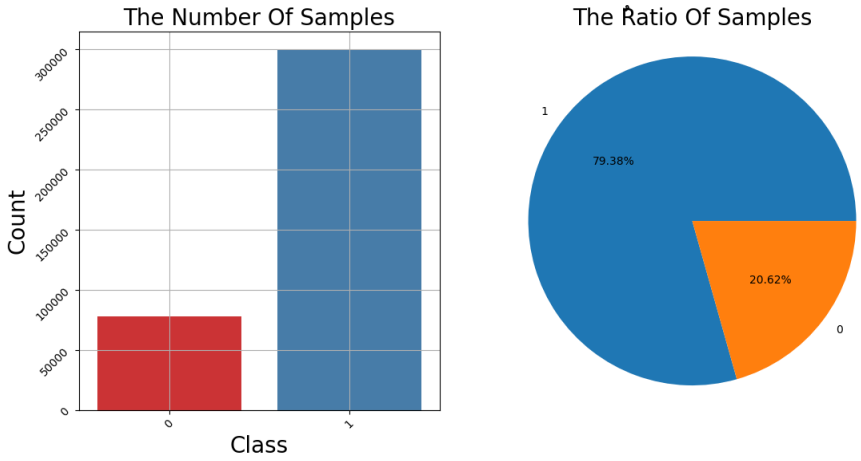

Fig. 2. The number of samples for each class in the dataset

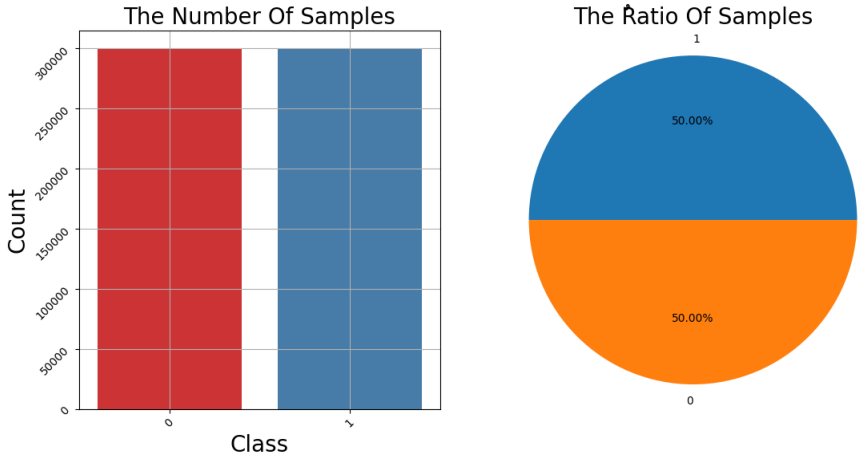

Fig. 3. The number of samples for each class in the dataset after applying the SMOTE

3.5. Model Design

Fig. 4 shows the design of the proposed model. It represents how the layers of the proposed model are stacked. The following is an explanation of the structure of the proposed model:

- Input layer: The model takes input sequences of length 25.
- Embedding layer: The input sequence is embedded into vectors of length 25. This layer converts each word or symbol in the input sequence into a fixed-size vector representation.
- Bidirectional LSTM (Long Short-Term Memory) layer: This layer processes the embedded sequences bidirectionally, meaning it reads them forward and backward. It outputs a series of 50-dimensional vectors.
- Attention layer: This layer computes attention weights on the bidirectional LSTM outputs. It helps the model focus on the most relevant parts of the input sequence when making predictions.
- Convolutional layer (Conv1D): The Conv1D layer applies 64 filters to the output of the attention layer. Each filter has different weights than the other filters, so each filter produces a feature map.
- Global Max Pooling 1D Layer: This layer performs max pooling. Global Max Pooling reduces the dimensionality of the input data. In the case of Global Max Pooling 1D, it works on the dimension along the sequence, collapsing it to a single value for each feature map. Global Max Pooling selects

the maximum value from each feature map, effectively capturing the most prominent feature or activation within each feature map.

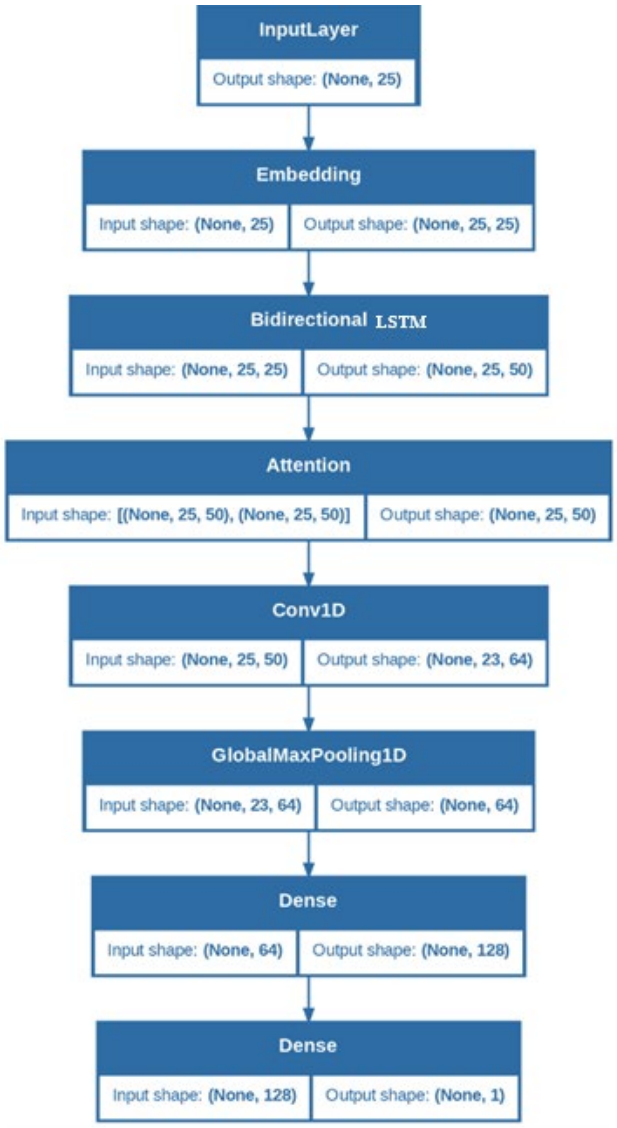

Fig. 4. The design of the proposed model

3.5.1. Embedding layer

While dealing with text data, we need to convert it into numbers before feeding it into any machine-learning model, including neural networks [32]. For simplicity, words can be compared to categorical variables. We use a fast encoder to convert categorical features into numbers. To do this, we generate a unique number for each class, for example, in binary classification, we encode the target class as zero and one. Similarly, if we use a single fast encoder on words in text data, we will have a unique number for each word, which means 10,000 numbers for a vocabulary of 10,000 words. This is not a feasible embedding method as it requires a large storage space for word vectors and reduces the model's efficiency. The embedding layer enables us to convert each word into a fixed-length vector of a specific size. The resulting vector is a dense vector containing real values. The fixed length of word vectors helps us represent words better. In this way, the embedding layer acts as a lookup table. The words are the keys in this table, while the vectors are the values. Embeddings are a very good way to deal with NLP problems because they are able to understand the context of the word so that similar words have similar embeddings [33].

3.5.2. Bidirectional LSTM Layer

To explain the bidirectional LSTM architecture, first explain the unidirectional LSTM architecture. LSTM is a type of recurrent neural network (RNN) architecture. The main component of LSTM networks is the memory cell, which can store information for long periods (Fig. 5).

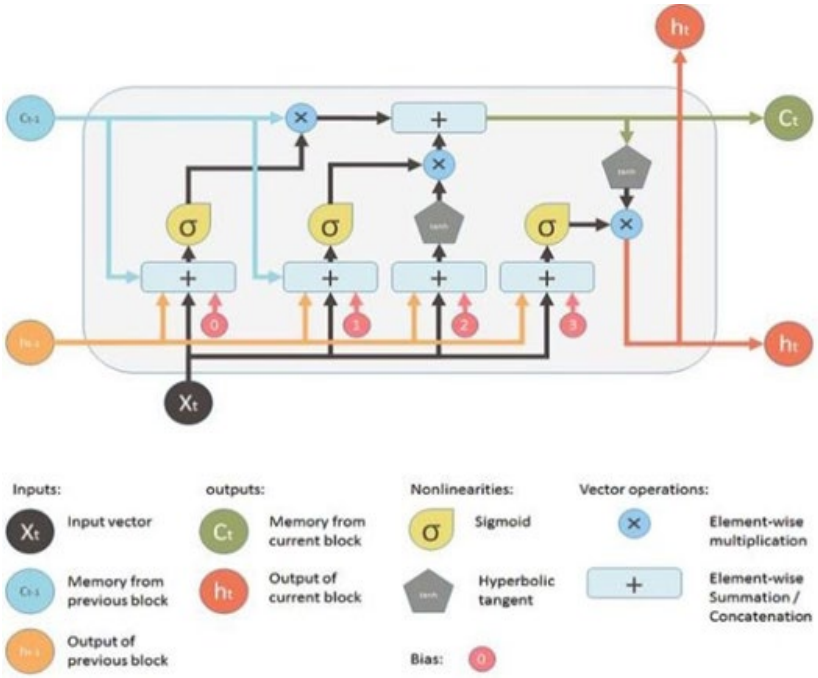

Fig. 5. LSTM cell structure [36]

The cell maintains an internal state that can be updated or modified based on the input data and previous states [34]. This allows LSTMs to remember or forget information over time selectively. An LSTM cell contains three main gates [35]: 1) Forget gate determines the information from the previous state that should be discarded. When data is input, this gate determines the useful information to store and delete and forgets the unimportant information to improve the performance of the recurrent neural network and train it quickly. This gate consists of two inputs; one is the result of the previous node, and the other is the input of the current node after determining the biases and weights of the matrix and applying the sigmoid function to each neuron. The information is saved and remembered according to the values. If the value is 0, the forget gate deletes the information. If the value is 1, the data is important; 2) The input gate determines the new information stored in the current state. Data is entered through this gate, consisting of neurons whose number depends on the amount of data. An activation function called tanh makes the input values in the range [-1,1], which modifies and formats the information before entering it into the next layer; and 3) The output gate determines the output based on the current state. Through this cell, we get the data that interests us from the previous cell after creating a matrix of values in the range [-1,1] using the tanh function, and the sigmoid function determines the final values that will be displayed in the output.

Bidirectional LSTM is a term used for a sequence model with two LSTM layers, one for processing the input in the forward direction and one for processing in the backward direction. The reason behind this approach is that by processing the data in both directions, the model can better understand the sequence relationship (e.g. knowing the next and previous words in a sentence).

3.5.3. Attention layer

Attention mechanisms enhance deep learning models by selectively focusing on important input elements, improving prediction accuracy and computational efficiency. It prioritizes important information and acts as a highlight to improve the model's overall performance. The architecture of the

attention mechanism includes three main components: encoder, attention module, and decoder [37]. Fig. 6 illustrates the architecture of the attention mechanism.

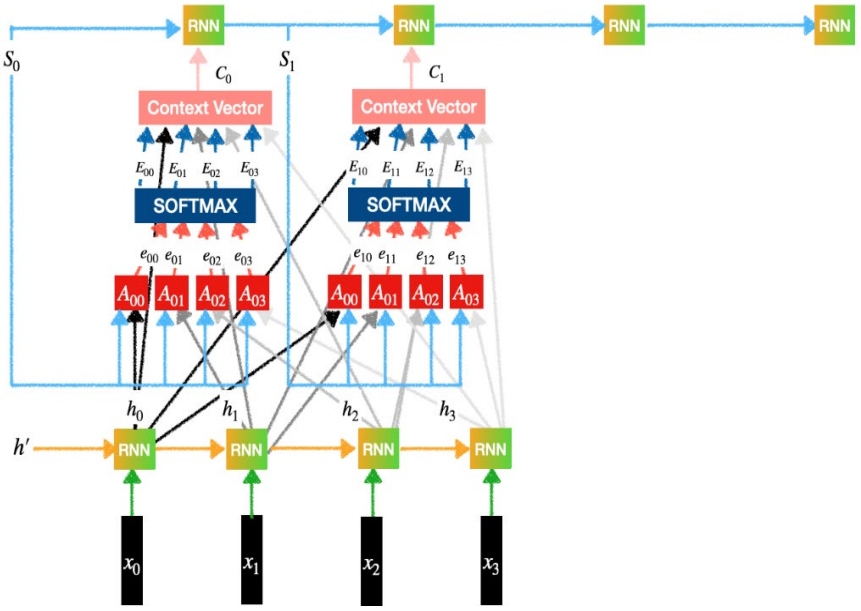

Fig. 6. Structure of the attention mechanism [38]

The encoder applies recurrent neural networks (RNNs) to process the input sequence iteratively. The encoder generates a hidden state at each step containing the data from the previous hidden state and the current input symbol. These hidden states represent the entire input sequence combined [39]. The attention module has 3 sub-parts: feed-forward network, SoftMax activation function, and context vector generation. The context vector is fed to the decoder and the decoder's current hidden state to predict the next symbol in the output sequence [39]. Until the decoder generates the entire output sequence, this process is repeated. We feed these context vectors to the RNNs of the decoder layer individually [40]. The attention mechanism allows the decoder to focus dynamically on different parts of the input sequence based on their importance. As a result, the model can easily handle long input sequences and capture the relationships between different components of the input and output sequence [39].

3.5.4. Convolution Layer (Conv1D)

The one-dimensional (1D) convolutional layer (Conv1D) in deep learning is specifically designed to process one-dimensional (1D) sequence data. The basic operation in the Conv1D layer involves moving a convolutional filter (or kernel) through the input sequence [41]. This filter is a set of learnable weights the network adjusts during training. The convolution operation multiplies the filter values by the original input values in a portion of the sequence. Then it sums the resulting values, summing the results to produce a single output point. This operation is repeated across the entire sequence, resulting in a sequence as output. Fig. 7 shows an example of a one-dimensional convolution operation [41].

A number of filters are applied in the convolutional layer so that the process shown in Fig. 7 is repeated for each filter, resulting in another output, each of which is called a feature map. Thus, the output of the convolutional layer becomes a set of feature maps. The more filters are used in the convolutional layer, the more feature maps are extracted, and therefore, our network is better at identifying patterns within the input data.

3.5.5. Global Max Pooling Layer

This layer is optional in the network design, i.e. its presence is not required, and if it is present, it will come after the convolutional layer and aims to reduce the number of samples or neurons (feature map), as it will shorten each group of input neurons of a certain size to one neuron. This size is

determined by a small window within the network design, and its optimal value is in the case of using a one-dimensional convolutional layer (1x2) because enlarging it may lead to loss of information [42]. Lastly, by applying this process to the output shown in Fig. 7, the result is shown in Fig. 8.

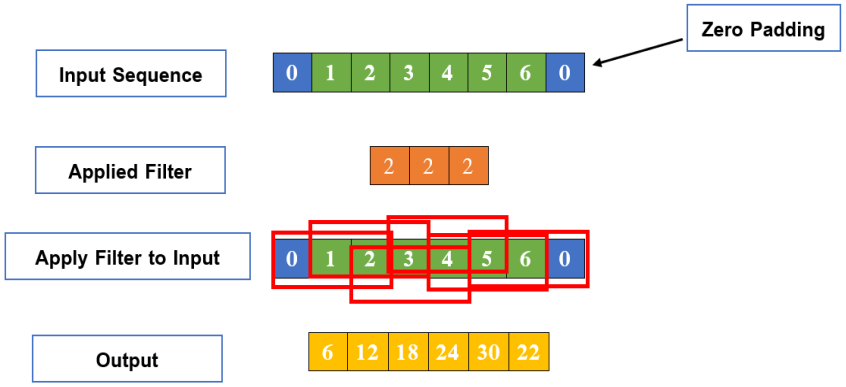

Fig. 7. An example of a one-dimensional convolution process

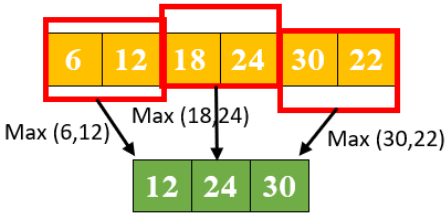

Fig. 8. Max Pooling process.

3.5.6. Dense Layer

A dense layer, also known as a fully connected layer, is a fundamental building block in neural network architectures. In a dense layer, each neuron (or node) is connected to every neuron in the previous layer, hence the term "fully connected." This means that every neuron in the dense layer receives input from every neuron in the previous layer. Each connection between neurons in adjacent layers is associated with a weight parameter. The dense layer computes the weighted sum of the inputs from the previous layer, where the weights are learned during training [43].

3.6. Performance matrices

Within the realm of machine learning, the confusion matrix (CM), also known as an error matrix [44], is a table format used to depict the performance of a model, particularly in supervised learning [45]. The structure of the CM is shown in Fig. 9 , where TP (True Positive) is the model correctly predicts a positive outcome, TN (True Negative) is the model that correctly predicts a negative outcome, FP (False Positive) for the model incorrectly predicts a positive outcome, and FN (False Negative) calculates incorrectly the model predicts a negative outcome.

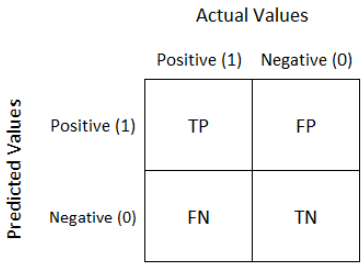

Fig. 9. Confusion matrix architecture.

These parameters are used to calculate key performance metrics such as recall, precision, and accuracy.

$$Recall = \frac{TP}{TP+FN} \tag{1}$$

The above equation can be discovered as "Measures the proportion of actual positives that are correctly identified by the model [46].

$$Precision = \frac{TP}{TP+FP} \tag{2}$$

The above equation can be discovered as "Measures the proportion of positive predictions that are actually correct [47].

The accuracy reflects the overall correctness of the model by measuring the proportion of correct predictions (both positives and negatives) out of all predictions [48]. The equation is given as follows:

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

The F1 Score is a metric that combines precision and recall, providing a balanced evaluation of a model's overall performance. The formula is as follows:

$$F1_{\_Score} = 2x\frac{Precision \times Recall}{Precision+Recall} \tag{4}$$

4. Results and Discussion

A learning curve is a graphical tool that visually represents the progress of a given metric during the training of a machine learning model. These curves depict the learning process mathematically, typically with time or training iterations on the x-axis and performance or error on the y-axis. Learning curves are essential for tracking the evolution of a model during training, allowing for identification of problems and fine-tuning to improve predictive performance. Fig. 10 shows the accuracy and loss curves for both the training and validation phases.

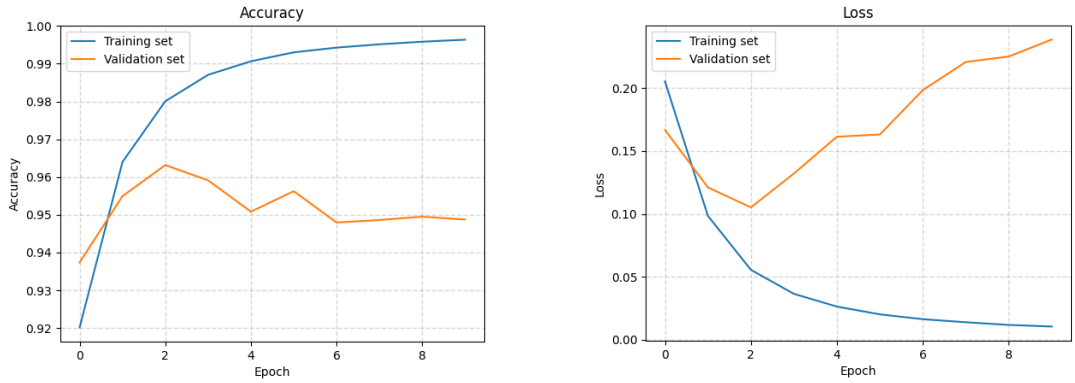

Fig. 10. The accuracy and loss curves for both the training and validation phases

These training results provide valuable insights into the model's performance over the training epochs. We see from Fig. 10 that the training accuracy started at 92.02% and steadily improved, reaching 99.64% by epoch 10. This indicates that the model learned well from the training data, and the training loss decreased correspondingly, from 0.2053 to 0.0105. We also see that the validation accuracy started at 93.74% in epoch 1, improved to a peak of 96.32% by epoch 3, before gradually decreasing to around 94.87% by the end of epoch 10. The validation loss also initially improved from 0.1668 to 0.1052 but increased after epoch 3, indicating that the model's ability to generalize to unseen data began to deteriorate. By the end of the training, the validation loss was 0.2385, indicating that overfitting occurred

after epoch 3. Therefore, to avoid overfitting, we used a set of callbacks, EarlyStopping(), to stop the model when overfitting started, and ModelCheckpoint() was used to keep the model in the epoch that achieved the highest accuracy and lowest loss. The model achieved the best validation accuracy (96.32%) in epoch 3, so this point represents the most effective model to use in practice. Fig. 11 shows the results of the confusion matrix resulting from applying the trained model to the test data.

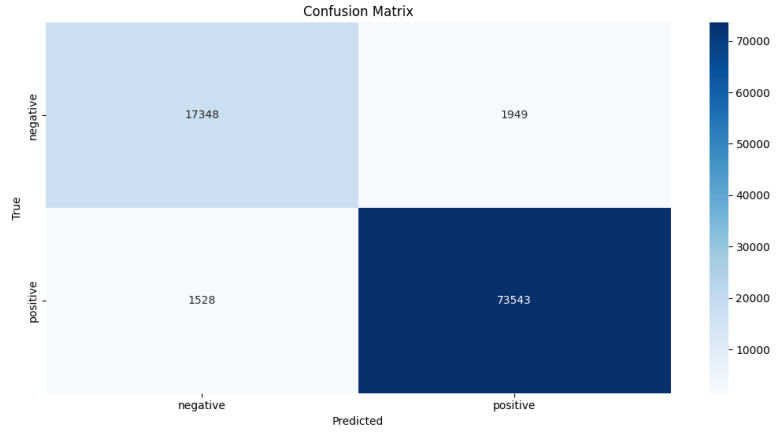

Fig. 11. The results of the confusion matrix resulting from applying the trained model to the test data

The confusion matrix summarizes the results of the predictions made by the model. The elements of the main diagonal represent the correct predictions of the model, and the secondary diagonal elements represent the incorrect predictions of the model. We notice from Fig. 11 that the model correctly predicted 17348 samples of data that the sentiments in them are negative and correctly predicted 73543 samples of data that the sentiments in them are positive. We also notice that the model made mistakes in predicting 1949 samples as it predicted them as positive when they were actually negative and incorrectly predicted 1528 samples as negative when they were positive. Table 1 shows the values of Precision, Recall, and $f1_{\_score}$ for each category.

Table 1. Precision, Recall, and $f1_{\_score}$ for each category of the proposed model

	Precision	Recall	F1_Score	Accuracy
Positive	0.92	0.90	0.91	**96%**
Negative	0.97	0.98	0.98	
Average	**0.945**	**0.94**	**0.945**	

We note from Table 1 that the model achieved a precision value of 0.92 for the positive category, meaning that among all the cases predicted as positive, 92% of them were correctly classified. This indicates that the model performed well in not misclassifying negative cases as positive. The recall value for the positive category is 0.9, meaning that among all the actual positive cases, the model predicted 90% of them correctly. The $f1_{\_score}$ value for the positive category indicates that the model achieves a balanced performance between precision and recall when predicting the positive category.

The model achieved a precision value of 0.97 for the negative category, so the model rarely misclassifies positive cases as negative and thus achieves very high accuracy. The recall value for the negative category is 0.98, meaning that almost all negative cases were correctly identified, which means that the model performs excellently in detecting negative sentiments. The $f1_{\_score}$ value for the negative category is 0.98, meaning that the model achieves a very high balance between precision and recall for the negative category. The model's overall accuracy is 96%, which indicates that the model performs very well in predicting both positive and negative sentiments. Compared to the deep learning models achieved in [48], the results are shown in Table 2.

Table 2. Comparing with deep learning models in [48]

	Precision	Recall	F1_Score	Accuracy
CNN	0.821	0.849	0.835	84.9%
BI-LSTM	0.855	0.871	0.861	87.1%
Proposed Model	**0.945**	**0.94**	**0.945**	**96%**

5. Conclusion

In this work, we performed an automatic tagging and SA method on raw Twitter data related to e-commerce activity. This was done using VADER for sentiment polarity detection and the proposed deep learning model. Before data analysis, the data was cleaned through data deletion, stop word removal, and stemming. Once cleaned, the deep learning model could automatically tag and classify the data. The proposed model demonstrated strong performance with an overall accuracy of 96%, particularly excelling in detecting negative sentiments with high precision (0.97), recall (0.98), and F1 score (0.98). Though slightly lower, the performance on positive sentiments was still effective, with an F1 score of 0.91. However, the model showed signs of overfitting after the third epoch, as the validation loss began to rise while accuracy plateaued. This suggests that while the model can learn well from the training data, its ability to generalize to unseen data could be improved. This research focuses on a product review dataset consisting of tweets from Twitter, which were collected and classified using a dictionary-based approach that identifies tweets based on predefined vocabulary within a dictionary. Therefore, tweets collected using specific predefined keywords may not represent the broader sentiment distribution. This is because sentiments can be expressed differently depending on language, dialect, or cultural context, leading to misclassifications. In future work, the dataset can be expanded to be more specific and consider different languages, dialects, and cultures to make the model more comprehensive and unbiased towards a specific language or culture. The model has 9,275,463 trainable parameters, which is relatively high as the embedding layer contributes the largest number of parameters (9,247,150), indicating a large vocabulary size and dimensionality of the embedding sequence, which may slow down the training. Also, using bidirectional LSTM increases the computational cost of the model. In future work, the embedding size can be reduced by using pre-trained embeddings, which may help reduce the dimensionality. Also, GRU can be used instead of LSTM as GRU has fewer parameters and computes faster. Also, in future works, exploring advanced architectures such as transformers or further tuning hyperparameters might improve both precision and recall for the positive class. Finally, incorporating class weight adjustments could help the model handle imbalanced data more effectively.

Declarations

Author contribution. A S Mirda (Conceptualization, Methodology, Writing - Original Draft) , S Buyrukoğlu (Investigation, Writing - Review & Editing), M R Baker (Supervision, Writing - Review & Editing).

Funding statement. This research received no specific grant from any funding agency in the public, commercial, or not-for-profit sectors.

Conflict of interest. The authors declare no conflict of interest.

Additional information. No additional information is available for this paper.

Data and Software Availability Statements

Data will be made available on request.

References

[1] T. Bhumika, Jyoti, G. Neha, and K. Santosh, "Overview of Electronic commerce (E-commerce)," *i-manager's J. Inf. Technol.*, vol. 11, no. 2, p. 29, 2022, doi: 10.26634/jit.11.2.18955.

[2] S. Baehre, "From Research to Action: Enhancing Net Promoter Score Utilization in Managerial Practice," *Int. J. Mark. Res.*, vol. 66, no. 2–3, pp. 174–181, May 2024, doi: 10.1177/14707853231209893.

[3] D. M. Goldberg and A. S. Abrahams, "Sourcing product innovation intelligence from online reviews," *Decis. Support Syst.*, vol. 157, p. 113751, 2022, doi: 10.1016/j.dss.2022.113751.

[4] C. Burnay, M. Lega, and S. Bouraga, "Business intelligence and cognitive loads: Proposition of a dashboard adoption model," *Data Knowl. Eng.*, vol. 152, p. 102310, Jul. 2024, doi: 10.1016/j.datak.2024.102310.

[5] M. Nowak and M. Pawłowska-Nowak, "Dynamic Pricing Method in the E-Commerce Industry Using Machine Learning," *Appl. Sci.*, vol. 14, no. 24, p. 11668, Dec. 2024, doi: 10.3390/app142411668.

[6] M. R. Baker, E. Z. Mohammed, and K. H. Jihad, "Prediction of Colon Cancer Related Tweets Using Deep Learning Models," in *Lecture Notes in Networks and Systems*, 2023, vol. 646 LNNS, pp. 522–532, doi: 10.1007/978-3-031-27440-4_50.

[7] M. R. Baker, Z. N. Mahmood, and E. H. Shaker, "Ensemble Learning with Supervised Machine Learning Models to Predict Credit Card Fraud Transactions," *Rev. d'Intelligence Artif.*, vol. 36, no. 4, pp. 509–518, 2022, doi: 10.18280/ria.360401.

[8] Y. Song and B. Yoon, "Prediction of infectious diseases using sentiment analysis on social media data," *PLoS One*, vol. 19, no. 9, p. e0309842, Sep. 2024, doi: 10.1371/journal.pone.0309842.

[9] N. L. Rane, S. P. Choudhary, and J. Rane, "Artificial Intelligence, Natural Language Processing, and Machine Learning to Enhance e-Service Quality on e-Commerce Platforms," *Int. J. Artif. Intell. Mach. Learn.*, vol. 4, no. 2, pp. 67–82, Jul. 2024, doi: 10.51483/IJAIML.4.2.2024.67-82.

[10] M. Rodríguez-Ibánez, A. Casánez-Ventura, F. Castejón-Mateos, and P.-M. Cuenca-Jiménez, "A review on sentiment analysis from social media platforms," *Expert Syst. Appl.*, vol. 223, p. 119862, Aug. 2023, doi: 10.1016/j.eswa.2023.119862.

[11] M. S. Md Suhaimin, M. H. Ahmad Hijazi, E. G. Moung, P. N. E. Nohuddin, S. Chua, and F. Coenen, "Social media sentiment analysis and opinion mining in public security: Taxonomy, trend analysis, issues and future directions," *J. King Saud Univ. - Comput. Inf. Sci.*, vol. 35, p. 101776, no. 9, 2023, doi: 10.1016/j.jksuci.2023.101776.

[12] F. Greco, "Sentiment analysis and opinion mining," *Elgar Encycl. Technol. Polit.*, vol. 5, no. 1, pp. 105–108, 2022, doi: 10.4337/9781800374263.sentiment.analysis.

[13] M. S. Islam *et al.*, "'Challenges and future in deep learning for sentiment analysis: a comprehensive review and a proposed novel hybrid approach,'" *Artif. Intell. Rev.*, vol. 57, no. 3, p. 62, Mar. 2024, doi: 10.1007/s10462-023-10651-9.

[14] Y. C. Hua, P. Denny, J. Wicker, and K. Taskova, "A systematic review of aspect-based sentiment analysis: domains, methods, and trends," *Artif. Intell. Rev.*, vol. 57, no. 11, p. 296, Sep. 2024, doi: 10.1007/s10462-024-10906-z.

[15] J. Zhou, J. Tian, R. Wang, Y. Wu, W. Xiao, and L. He, "SentiX: A Sentiment-Aware Pre-Trained Model for Cross-Domain Sentiment Analysis," in *Proceedings of the 28th International Conference on Computational Linguistics*, 2020, pp. 568–579, doi: 10.18653/v1/2020.coling-main.49.

[16] R. Li, H. Chen, F. Feng, Z. Ma, X. Wang, and E. Hovy, "Dual graph convolutional networks for aspect-based sentiment analysis," in *ACL-IJCNLP 2021 - 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing, Proceedings of the Conference*, 2021, pp. 6319–6329, doi: 10.18653/v1/2021.acl-long.494.

[17] W. Zhang, X. Li, Y. Deng, L. Bing, and W. Lam, "Towards Generative Aspect-Based Sentiment Analysis," in *ACL-IJCNLP 2021 - 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing, Proceedings of the Conference*, 2021, vol. 2, pp. 504–510, doi: 10.18653/v1/2021.acl-short.64.

[18] C. Ouni, E. Benmohamed, and H. Ltifi, "Sentiment analysis deep learning model based on a novel hybrid embedding method," *Soc. Netw. Anal. Min.*, vol. 14, no. 1, p. 210, Oct. 2024, doi: 10.1007/s13278-024-01367-x.

[19] T. Osman, H. Khalil, M. Miltan, K. Shaalan, and R. Alfrjani, "Exploiting Functional Discourse Grammar to Enhance Complex Arabic Relation Extraction using a Hybrid Semantic Knowledge Base - Machine Learning Approach," *ACM Trans. Asian Low-Resource Lang. Inf. Process.*, vol. 22, no. 8, pp. 1–30, Aug. 2023, doi: 10.1145/3610581.

[20] B. F. Azevedo, A. M. A. C. Rocha, and A. I. Pereira, "Hybrid approaches to optimization and machine learning methods: a systematic literature review," *Mach. Learn.*, vol. 113, no. 7, pp. 4055–4097, Jul. 2024, doi: 10.1007/s10994-023-06467-x.

[21] H. Khalid, "Modern techniques in detecting, identifying and classifying fruits according to the developed machine learning algorithm," *J. Appl. Res. Technol.*, vol. 22, no. 2, pp. 219–229, 2024, doi: 10.22201/icat.24486736e.2024.22.2.2269.

[22] H. Khalid, "Efficient Image Annotation and Caption System Using Deep Convolutional Neural Networks," *BIO Web Conf.*, vol. 97, no. 3, p. 103, 2024, doi: 10.1051/bioconf/20249700103.

[23] H. J. Alantari, I. S. Currim, Y. Deng, and S. Singh, "An empirical comparison of machine learning methods for text-based sentiment analysis of online consumer reviews," *Int. J. Res. Mark.*, vol. 39, no. 1, pp. 1–19, Mar. 2022, doi: 10.1016/j.ijresmar.2021.10.011.

[24] M. Assaad and G. Shakah, "Optimizing Health Pattern Recognition Particle Swarm Optimization Approach for Enhanced Neural Network Performance(2):76-3. Available from: ," *Cihan Univ. Sci. J.*, vol. 8, no. 2, pp. 76–83, 2024, doi: 10.24086/cuesj.v8n2y2024.pp76-83.

[25] F. Husari and M. Assaad, "Intelligent Handwritten Identification Using Novel Hybrid Convolutional Neural Networks – Long-short-term Memory Architecture," *Cihan Univ. Sci. J.*, vol. 8, no. 2, pp. 99–103, 2024, doi: 10.24086/issn.2519-6979.

[26] H. O. Ahmad and S. U. Umar, "Sentiment Analysis of Financial Textual data Using Machine Learning and Deep Learning Models," *Informatica*, vol. 47, pp. 153–158, 2023, doi: 10.31449/inf.v47i5.4673.

[27] Y. M. Hazzaa and S. U. Umar, "Improving Network Intrusion Detection with Convolutional Neural Networks and Data Balancing Techniques," in *Proceedings of Third International Conference on Computing and Communication Networks*, 2024, pp. 675–687, doi: 10.1007/978-981-97-0892-5_53.

[28] O. Reda and A. Zellou, "Fulmqa: a fuzzy logic-based model for social media data quality assessment," *Soc. Netw. Anal. Min.*, vol. 13, no. 1, p. 150, Nov. 2023, doi: 10.1007/s13278-023-01148-y.

[29] A. H. Alamoodi *et al.*, "Public Sentiment Analysis and Topic Modeling Regarding COVID-19's Three Waves of Total Lockdown: A Case Study on Movement Control Order in Malaysia," *KSII Trans. Internet Inf. Syst.*, vol. 16, no. 7, pp. 2169–2190, 2022, doi: 10.3837/tiis.2022.07.003.

[30] C. Vairetti, J. L. Assadi, and S. Maldonado, "Efficient hybrid oversampling and intelligent undersampling for imbalanced big data classification," *Expert Syst. Appl.*, vol. 246, p. 123149, Jul. 2024, doi: 10.1016/j.eswa.2024.123149.

[31] D. Elreedy, A. F. Atiya, and F. Kamalov, "A theoretical distribution analysis of synthetic minority oversampling technique (SMOTE) for imbalanced learning," *Mach. Learn.*, vol. 113, no. 7, pp. 4903–4923, Jul. 2024, doi: 10.1007/s10994-022-06296-4.

[32] B. V. Sekharreddy, V. N. Thatha, G. U. Kiran, V. Srilakshmi, and S. Sanapala, "A Survey On Text Classification Using Different Machine Learning Approaches," in *Futuristic Trends in Computing Technologies and Data Sciences Volume 3 Book 1*, Iterative International Publisher, Selfypage Developers Pvt Ltd, 2024, pp. 178–186, doi: 10.58532/V3BGCT1P5CH4.

[33] P. L. Rodriguez and A. Spirling, "Word Embeddings: What Works, What Doesn't, and How to Tell the Difference for Applied Research," *J. Polit.*, vol. 84, no. 1, pp. 101–115, Jan. 2022, doi: 10.1086/715162.

[34] A. Boujamza and S. Lissane Elhaq, "Attention-based LSTM for Remaining Useful Life Estimation of Aircraft Engines," *IFAC-PapersOnLine*, vol. 55, no. 12, pp. 450–455, Jan. 2022, doi: 10.1016/j.ifacol.2022.07.353.

[35] S. Nosouhian, F. Nosouhian, and A. Kazemi Khoshouei, "A Review of Recurrent Neural Network Architecture for Sequence Learning: Comparison between LSTM and GRU," *Preprints.org*, no. July. pp. 1–7, Jul. 12, 2021, doi: 10.20944/preprints202107.0252.v1.

[36] N. Deshmukh, "Semi-Supervised Natural Language Processing Approach for Fine-Grained Classification of Medical Reports," in *2019 IEEE MIT Undergraduate Research Technology Conference, URTC 2019*, 2019, pp. 1–4, doi: 10.1109/URTC49097.2019.9660430.

[37] Z. Niu, G. Zhong, and H. Yu, "A review on the attention mechanism of deep learning," *Neurocomputing*, vol. 452, pp. 48–62, Sep. 2021, doi: 10.1016/j.neucom.2021.03.091.

[38] A. Ahmadzade and S. Malekzadeh, "Spell Correction for Azerbaijani Language using Deep Neural Networks," *arXiv Prepr.*, vol. arXiv:2102, pp. 1–5, 2021, [Online]. Available at: https://arxiv.org/abs/2102.03218.

[39] S. Jamshidi *et al.*, "Effective text classification using BERT, MTM LSTM, and DT," *Data Knowl. Eng.*, vol. 151, p. 102306, May 2024, doi: 10.1016/j.datak.2024.102306.

[40] S. Das, A. Tariq, T. Santos, S. S. Kantareddy, and I. Banerjee, "Recurrent Neural Networks (RNNs): Architectures, Training Tricks, and Introduction to Influential Research," in *Neuromethods*, vol. 197, Humana Press Inc., 2023, pp. 117–138, doi: 10.1007/978-1-0716-3195-9_4.

[41] M. K, A. Ramesh, R. G, S. Prem, R. A A, and D. M. P. Gopinath, "1D Convolution approach to human activity recognition using sensor data and comparison with machine learning algorithms," *Int. J. Cogn. Comput. Eng.*, vol. 2, pp. 130–143, 2021, doi: 10.1016/j.ijcce.2021.09.001.

[42] A. Zafar *et al.*, "A Comparison of Pooling Methods for Convolutional Neural Networks," *Appl. Sci.*, vol. 12, no. 17, pp. 1–21, 2022, doi: 10.3390/app12178643.

[43] X. He, X. Wang, Z. Zhou, J. Wu, Z. Yang, and L. Thiele, "On-Device Deep Multi-Task Inference via Multi-Task Zipping," *IEEE Trans. Mob. Comput.*, vol. 22, no. 5, pp. 2878–2891, May 2023, doi: 10.1109/TMC.2021.3124306.

[44] S. Haghighi, M. Jasemi, S. Hessabi, and A. Zolanvari, "PyCM: Multiclass confusion matrix library in Python," *J. Open Source Softw.*, vol. 3, no. 25, p. 729, 2018, doi: 10.21105/joss.00729.

[45] I. Markoulidakis, I. Rallis, I. Georgoulas, G. Kopsiaftis, A. Doulamis, and N. Doulamis, "Multiclass Confusion Matrix Reduction Method and Its Application on Net Promoter Score Classification Problem," *Technologies*, vol. 9, no. 4, 2021, doi: 10.3390/technologies9040081.

[46] K. Roth, L. Pemula, J. Zepeda, B. Scholkopf, T. Brox, and P. Gehler, "Towards Total Recall in Industrial Anomaly Detection," in *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 2022, vol. 2022-June, pp. 14298–14308, doi: 10.1109/CVPR52688.2022.01392.

[47] S. J. Maceachern and N. D. Forkert, "Machine learning for precision medicine," *Genome*, vol. 64, no. 4, pp. 416–425, 2021, doi: 10.1139/gen-2020-0131.

[48] G. Fu *et al.*, "A deep-learning-based approach for fast and robust steel surface defects classification," *Opt. Lasers Eng.*, vol. 121, pp. 397–405, 2019, doi: 10.1016/j.optlaseng.2019.05.005.