

Article

# Sentiment Analysis of Students' Feedback on E-Learning Using a Hybrid Fuzzy Model

Maryam Alzaid<sup>1</sup> and Fethi Fkih<sup>1,2,\*</sup> 

<sup>1</sup> Department of Computer Science, College of Computer, Qassim University, Buraydah 51452, Saudi Arabia; m.alzaid@qu.edu.sa

<sup>2</sup> MARS Research Lab LR 17ES05, University of Sousse, Sousse 4002, Tunisia

\* Correspondence: f.fki@qu.edu.sa

**Abstract:** It is crucial to analyze opinions about the significant shift in education systems around the world, because of the widespread use of e-learning, to gain insight into the state of education today. A particular focus should be placed on the feedback from students regarding the profound changes they experience when using e-learning. In this paper, we propose a model that combines fuzzy logic with bidirectional long short-term memory (BiLSTM) for the sentiment analysis of students' textual feedback on e-learning. We obtained this feedback from students' tweets expressing their opinions about e-learning. There were some ambiguous characteristics in terms of the writing style and language used in the collected feedback. It was written informally and not in adherence to standardized Arabic language writing rules by using the Saudi dialects. The proposed model benefits from the capabilities of the deep neural network BiLSTM to learn and also from the ability of fuzzy logic to handle uncertainties. The proposed models were evaluated using the appropriate evaluation metrics: accuracy, F1-score, precision, and recall. The results showed the effectiveness of our proposed model and that it worked well for analyzing opinions obtained from Arabic texts written in Saudi dialects. The proposed model outperformed the compared models by obtaining an accuracy of 86% and an F1-score of 85%.

**Keywords:** fuzzy logic; sentiment analysis; feature extraction; deep neural networks; deep learning; e-learning



**Citation:** Alzaid, M.; Fkih, F. Sentiment Analysis of Students' Feedback on E-Learning Using a Hybrid Fuzzy Model. *Appl. Sci.* **2023**, *13*, 12956. <https://doi.org/10.3390/app132312956>

Academic Editor: Giacomo Fiumara

Received: 10 October 2023

Revised: 26 November 2023

Accepted: 28 November 2023

Published: 4 December 2023



**Copyright:** © 2023 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

## 1. Introduction

Technology changes rapidly and impacts every aspect of people's day-to-day lives. In line with this, we have seen a dramatic change in the education sector and the widespread use of e-learning. The term e-learning refers to the use of digital resources and internet-enabled devices to deliver learning in synchronous or asynchronous environments [1]. E-learning was officially and extensively employed during the recent COVID-19 pandemic [2]. Today, in most universities and schools around the world, e-learning has become the first alternative when attendance is impossible for any reason. In particular, this procedure is currently being followed in Saudi Arabia. Students have provided varied feedback on the profound changes they face regarding e-learning, which is a different learning experience. Student feedback has grown in popularity and importance in recent years, as student feedback and opinions are valuable sources of information [3]. Generally, the detection and classification of this feedback is receiving great attention from researchers as a novel and important research topic. Previously, the method for analyzing students' opinions was based only on data collected using questionnaires. However, accessing and analyzing students' opinions has become simpler, as students nowadays express their feedback on various social media sites. Thus, many researchers exploit social media sites, such as Twitter, to collect feedback and opinions [4].

Many studies have analyzed student feedback based on the sentiment analysis approach [5], where students' textual feedback is classified according to the sentiment polarity

that it expresses [6]. Feedback analysis using opinion mining or the sentiment analysis approach revolves around the same meaning and objectives. They all fall within fields of study aimed at identifying the sentiments, appraisals, attitudes, and opinions expressed in human-written texts toward various entities, such as issues, events, and topics [7]. They utilize natural language processing (NLP), text mining, and text feature extraction to classify the polarity of a text as positive or negative [8]. Despite the similarities between opinion mining and the sentiment analysis approach, they are slightly different. The sentiment analysis approach identifies words and phrases that convey emotions, while opinion mining is the process of extracting opinions in general on a particular entity [3]. In this study, we employ the sentiment analysis approach to analyze students' feedback about e-learning. Researchers have used various techniques in the literature to correctly predict the sentiments implied in a text. These techniques started with lexicon-based approaches, then evolved to machine learning techniques, and then advanced to deep learning techniques [9].

Regarding the Arabic sentiment analysis, there are recent studies related to the topic and type of dataset used in this study, i.e., e-learning and Twitter-based data. The authors in [4,10] performed sentiment analyses of general Arabic tweets about e-learning, while the researchers in [11,12] analyzed Twitter datasets about e-learning relating to Saudi Arabia. These studies only focused on using traditional machine learning techniques, such as the naive Bayes and random forest techniques, and traditional feature extraction methods, such as the N-Gram and TF-IDF methods. Table 1 provides an overview of the machine learning techniques used in these studies and the methods of feature extraction, as well as information about the dataset language and the type of annotation. Table 1 reveals many gaps related to Arabic Sentiment Analysis. In fact, the most serious gap in this domain is the few numbers of research articles that handle this problem, especially for the Saudi dialects.

**Table 1.** Summary of recent studies related to the topic and type of the dataset in this study.

Ref	Machine Learning Techniques	Feature Extraction Methods	Type of Dataset Annotation	Language of Dataset
[4]	Logistic Regression, Support Vector Machine	Bag of words	N/A	Arabic
[10]	Logistic Regression, K-Nearest Neighbor, Naive Bayes, Multinomial Naive Bayes, Support Vector Machine	N/A	Automatic	Arabic
[11]	Naive Bayes, Random Forest, K-Nearest Neighbor	N/A	Automatic	Arabic in Saudi dialect
[12]	Support Vector Machine, Random Forest, K-Nearest Neighbor, Naive Bayes, Logistic Regression, XGBoost	N-Gram, TF-IDF	Manual	Arabic in Saudi dialect

Machine learning techniques have been enhanced by deep learning techniques, which carry out feature extraction automatically [9]. Therefore, they perform better in NLP tasks, such as text classification. Accordingly, they have shown a great enhancement in the sentiment analysis of texts written in different languages. The authors of [13–15] performed an analysis of the sentiment of Saudi tweets, and they used a particular type of deep learning technique, that is, the recurrent neural network such as the GRU and BiLSTM. Contrastingly, there is inherent ambiguity in the natural language of unstructured data, and it cannot be addressed using only deep learning techniques due to their fully deterministic nature making them unable to reduce uncertainties in data [16]. Thus, deep

learning techniques face challenges because of the vagueness and uncertainties within many written opinions [7]. Furthermore, it can be said that extracting features from Arabic texts and then classifying and analyzing them are challenging tasks; this is due to the lack of contextual information and explicit opinion words in texts, where Arabic texts have a rich morphology, many irregular forms, and a variety of dialects [17,18]. Addressing and managing these uncertainty and ambiguity issues is a challenging and highly critical task when using sentiment analysis techniques [7]. Considering this importance, in this study we aimed to develop a hybrid model for analyzing the sentiment of student feedback, written in Saudi dialectal Arabic, about e-learning. This hybrid model combines fuzzy logic with BiLSTM, which is a type of deep neural network. Although fuzzy logic has been utilized in several ways by previous works to solve the uncertainty problems in textual feedback analysis based on sentiment in terms of the English language, to the best of our knowledge, no clear efforts have been made in terms of the Arabic language. Apparently, the use of fuzzy logic with the deep learning of an Arabic sentiment analysis has not yet been explored. Most efforts have only targeted the use of fuzzy logic with a lexicon-based approach. Biltawi et al. [19] and Rattrout et al. [20] proposed a lexicon-based approach using fuzzy logic to predict the sentiment of texts written in the Arabic language.

The proposed model in this study is a hybridization of fuzzy logic and a deep neural network. The main purpose of the proposed model is to incorporate the advantages of both these techniques, as they complement each other. This hybridization combines the approximate reasoning ability of fuzzy logic with the learning capability of deep neural networks. A combination of deep neural networks and fuzzy logic can produce a more reliable and accurate output, as fuzzy logic enhances the generalization capability of neural networks [16–21].

Many researchers have explored the combining of various deep learning techniques with fuzzy logic. Evidence shows that fuzzy logic can be combined with different types of deep neural networks, such as convolutional neural networks (CNN) and recurrent neural networks. This type of combination has been used successfully in various fields, such as image processing, time-series prediction, and various NLP tasks [22]. Different works are targeted at NLP, such as text summarization [23], sentiment analysis, and opinion mining. We focus on only presenting studies that used this combination for the purpose of sentiment analysis. Attempts to combine fuzzy logic with deep neural networks have produced remarkable results in sentiment analysis [24]. These attempts have often proved that this combination is beneficial.

A system based on sentiment analysis, called the Senti-eSystem, was proposed in [25]. This system is a hybridization of fuzzy logic and BiLSTM, and it was used to examine satisfaction in customer feedback. The proposed system combined BiLSTM and fuzzy logic in a sequential fashion. BiLSTM provides the sentiment polarity from the feedback, and then fuzzy logic is applied to determine customer satisfaction based on the sentiment class. Using the same combination approach, the authors of [26] analyzed the sentiments in aspects of customer reviews about mobile phones. They introduced a system that successively combined LSTM with fuzzy logic to generate sentiment classes for various aspects of mobile phones.

Using a parallel structure, a classifier model that combines fuzzy and deep learning was suggested by the authors of [27]. The classifier model combines CNN and fuzzy logic. The objective of using fuzzy logic was to classify inputs that have two sentiment scores for negativity and positivity into three classes by adding a neutral label. Two Twitter-based datasets were chosen to demonstrate the performance of the model. Several experiments on different aspects were performed to prove the effectiveness of the developed model, and these experiments yielded good results.

Conversely, the authors of [9,24] suggested a hybrid model with a cooperative structure by using deep learning with fuzzy logic at the feature point. The proposed model in [9] integrates the abilities of fuzzy logic with LSTM, and it works by fuzzifying the features to use them as input into LSTM. The authors aimed to predict the sentiments of movie

reviews in an IMDB dataset. Similar to this model, the authors of [24] proposed a hybrid model incorporating fuzzy logic with CNN for text sentiment classification. Both models were able to handle the problems of ambiguity and uncertainty in the addressed data and gain better classification accuracies. These works have given us a good start and improved our vision to develop a useful model using fuzzy logic and a deep neural network for the analysis of the sentiments of students' opinions about e-learning, which are in the form of Arabic texts written in the Saudi dialect, in an informal way.

In Table 2, we summarize the studies discussed in the previous three paragraphs by highlighting the type of deep learning network used in combination with fuzzy logic, the type of dataset and its language, and the accuracy of the performance achieved by the proposed models.

**Table 2.** Summary of studies combining deep learning and fuzzy logic.

Ref	Method	Source of Dataset	Language of Dataset	Best Accuracy Result
[24]	Fuzzy with BiLSTM	Twitter-based dataset	English	92.86%
[25]	Fuzzy with LSTM	Three Amazon review datasets	English	96.93%
[26]	Fuzzy with CNN	Two Twitter-based datasets	English	99.97%
[9]	Fuzzy with LSTM	Movie review dataset	English	88.91%
[19]	Fuzzy with CNN	Two Twitter-based datasets, three movie review datasets	English	78.85%

Based on what we previously discussed in detail, there are no works aimed at using fuzzy logic with deep learning targeting the Arabic language. However, we note that most of the contributions in the field of sentiment analysis about e-learning in Saudi Arabia, which has been widely applied recently, dealt only with analyzing the opinions of the public. Moreover, student feedback, which is considered a vital element of this issue, was not given attention by these studies.

As a result of this study, several main contributions can be highlighted as follows:

- A dataset is built from Twitter, and it includes the opinions of Saudi students about e-learning; these opinions are manually annotated as positive or negative.
- The collected dataset is related to e-learning, which is an important field that researchers in different disciplines are currently studying, so the dataset is helpful for reuse by other research works.
- An efficient hybrid model that combines fuzzy logic with BiLSTM is developed, and it is able to achieve good results. No previous studies have considered using this type of advanced integration in Arabic Sentiment Analysis.
- A comprehensive comparison of the performance of the proposed model with those of baseline models is provided.
- Generally, this study contributes to Arabic NLP tasks in terms of providing labeled data and developing a hybrid model aimed at handling aspects of uncertainty and ambiguity in Arabic texts.

We organized the rest of this paper as follows: Section 2 provides an explanation of the main concepts related to this work. The methodology used in this work is explained in detail in Section 3, and Section 4 provides details about the experiments. Then, Section 5 shows the obtained results. Finally, Section 6 discusses the results, concludes the work, and briefly mentions future work.

## 2. Preliminaries

It is important to present preliminary knowledge of related concepts, as this can help the reader comprehend the proposed model and the ideas behind this study. To achieve this, we first present a general review of deep neural networks, recurrent neural networks,

and the type of recurrent neural networks used in this work. Then, we briefly describe fuzzy logic and its associated terms. Finally, we provide fundamental information about the concept of the combination of fuzzy logic and deep neural networks and their different methods of combination.

### 2.1. Deep Neural Networks

Neural networks are defined as artificial neural networks that mimic the functions of the human brain. Fundamentally, a neural network consists of three main layers: an input layer, an output layer, and a hidden layer between them. Deep learning is a branch derived from machine learning techniques, and neural networks form the basis of deep learning techniques [28]. In fact, increasing the depth of the layers in a neural network to more than three layers results in deep learning techniques or deep neural networks (DNNs). There are various types of DNNs, including convolutional neural networks and recurrent neural networks. Convolutional neural networks and recurrent neural networks are the two primary structures that are typically used, and each has its own different applications.

#### 2.1.1. Recurrent Neural Networks (RNNs)

An RNN is a type of DNN that has an architecture adapted to the processing of sequential data. It works on the principle of recording the previous outputs in memory to generate the subsequent output [29]. A simple RNN faces problems known as gradient vanishing problems when it processes data that have long-term dependencies [30,31]. Thus, RNNs were developed with a more complex architecture type called the long short-term memory network (LSTM) to manage these problems.

The LSTM has some advantages and disadvantages, which are listed as follows:

##### **Advantages:**

1. **Sequential Processing:** LSTMs are well-suited for tasks involving sequential data, making them effective for sentiment analysis, where the order of words in a sentence can be crucial [30].
2. **Capturing Temporal Dependencies:** LSTMs can capture long-term dependencies in sequences, which can be beneficial for understanding the context and sentiment in a sentence [30].
3. **Interpretability:** LSTMs process input sequentially, which can make it easier to interpret the model's decision-making process, as you can trace the flow of information through the time steps [29].
4. **Smaller Datasets:** LSTMs can perform reasonably well with smaller datasets, which is advantageous when labeled sentiment analysis datasets are limited [31].

##### **Disadvantages:**

1. **Limited Parallelization:** LSTMs process sequences sequentially, limiting parallelization during training, which can result in longer training times [29].
2. **Difficulty with Long-Range Dependencies:** While LSTMs are designed to capture long-term dependencies, they may still struggle with very long-range dependencies in sequences [30].

#### 2.1.2. Transformers

The transformer architecture, introduced in the paper [32] by Vaswani et al., has had a profound impact on natural language processing (NLP) and various other machine learning tasks. In fact, the authors proposed a self-attention mechanism that allows the model to weigh different parts of the input sequence differently, capturing long-range dependencies more effectively than traditional recurrent or convolutional architectures.

In the same context, the authors in [33] introduced OpenAI's GPT-2, which is a transformer-based language model that demonstrated the power of generative pre-training. GPT-2 achieved remarkable results in tasks ranging from text completion to text generation.

The work presented in the paper [34] explored the interpretability of transformer models in the context of biomedical text mining. It investigated how attention mechanisms in transformers can be analyzed to gain insights into the model's decision-making process.

As follows, we provide the advantages and the disadvantages of the transformers:

**Advantages:**

1. Attention Mechanism: Transformers, with their attention mechanisms, can capture the global dependencies in the input sequence, allowing them to consider the entire context simultaneously [32].
2. Parallelization: Transformers can efficiently parallelize computations during training, leading to faster training times, especially on hardware that supports parallel processing [32].
3. Transfer Learning: Pre-trained transformer models, such as BERT, can be fine-tuned for sentiment analysis tasks. Transfer learning often leads to improved performance, especially when labeled data is limited [32].
4. Effective for Various Sequence Lengths: Transformers can handle input sequences of varying lengths without the need for padding, which is beneficial for sentiment analysis tasks with variable-length texts [32].

**Disadvantages:**

1. Computational Resources: Transformers, especially large pre-trained models, can be computationally intensive and may require significant resources, both in terms of memory and processing power [33].
2. Interpretability: Transformers may be seen as less interpretable than LSTMs due to their parallel processing and attention mechanisms, making it challenging to trace the flow of information through the model [33].

### 2.1.3. Comparison between LSTM-Based Models and Transformers

Both LSTMs and transformers are popular choices for sentiment analysis, but they have different architectures and characteristics [30]. The choice between an LSTM and transformers for sentiment analysis depends on factors such as the size of the dataset, computational resources, and the specific characteristics of the task. LSTMs are suitable for smaller datasets and tasks where sequential processing is crucial, as in our case where we have used a small dataset of a Saudi dialect where the words' order within the text is very important for the sentiment extraction task. For all these reasons, we have opted for an LSTM-based model. Nevertheless, transformers can be used as an extension for this work if we aim to expand our dataset.

## 2.2. Fuzzy Logic

The fuzzy logic concept was introduced in 1965 by the renowned mathematician Lotfi A. Zadeh [35]. Fuzzy logic is an approach used to describe fuzziness by employing a set of mathematical principles. It is based on the idea that everything can be described by a degree, and it allows for the inclusion of approximate reasoning to handle uncertainty in a subject. In contrast to Boolean logic, fuzzy logic is intended to compute the degree of truth to be between "completely true" and "completely false" or the degree of membership between 0 and 1 [35]. For more clarification, we briefly explain the terms associated with fuzzy logic as follows:

- Fuzzy set: This is set A and is defined by the membership function MA (Equation (1)), and each element x in the set has a certain degree of membership between 0 and 1.

$$MA(x) : X \rightarrow [0, 1] \text{ where } MA(x) = \begin{cases} 1 & \text{if } x \text{ is totally in } A \\ 0 & \text{if } x \text{ is totally not in } A \\ 0 < MA(x) < 1 & \text{if } x \text{ is partly in } A \end{cases} \quad (1)$$

- Membership function: This function computes how each element in the fuzzy sets is mapped to its degree of membership, which is a value from a range within [0,1].

There are several kinds of membership functions, and they are selected depending on the condition of the problem. In general, the most commonly used functions are trapezoidal, Gaussian, and triangular functions.

The following sequential steps are usually followed when implementing fuzzy logic in a real application [35,36]:

- **Fuzzification:** This step uses a membership function to transform a crisp value into a fuzzy value that expresses the degree of membership of an element to different fuzzy sets.
- **Fuzzy inference:** This step applies some of the if-then rules on the results of the membership functions to obtain the fuzzy output.
- **Defuzzification:** This step converts the fuzzy output into a crisp value.

### 2.3. The Fusion of Fuzzy Logic and a Deep Neural Network

There is an advanced concept of a type of fusion represented by the hybridization of fuzzy logic and deep learning [37]. Fuzzy logic can be combined with DNNs to provide an efficient framework for solving various complex problems. This type of combination falls under the category of hybrid techniques, which combine the characteristics of fuzzy logic and DNNs. It comprises the approximate reasoning capability of fuzzy logic and the learning ability of DNNs. It is known that DNNs significantly promote the learning ability of models. Conversely, fuzzy logic has an excellent ability to handle vague and imprecise circumstances [38]. Accordingly, the hybridization of fuzzy logic and DNNs enhances the accuracy of prediction and provides high-level abilities for interpretation and analysis. In fact, fuzzy logic has exhibited remarkable advantages in the context of DNNs and has successfully addressed aspects of ambiguity and uncertainty in various applications of DNNs. Common examples of these applications are seen in the fields of image processing, NLP, time-series prediction, and medical systems [22]. There are three different approaches to the fusion of fuzzy logic and DNNs. They can be combined in either a cooperative, sequential, or parallel structure as described in the subsections below [37–39].

#### 2.3.1. Cooperative Structure

A cooperative structure employs fuzzy logic as an integrated part of the deep learning process. The potential structure of cooperative models can be realized using two approaches. The first approach uses a fuzzy part to generate fuzzy inputs for the DNN. These fuzzy inputs pass through the network and then exit into another fuzzy part. This second fuzzy part defuzzifies these inputs and converts them into a final crisp output. In the second approach, fuzzy logic takes the output given by the DNN, processes it, and then returns it again to the DNN. After that, the DNN produces the final output [37].

#### 2.3.2. Sequential Structure

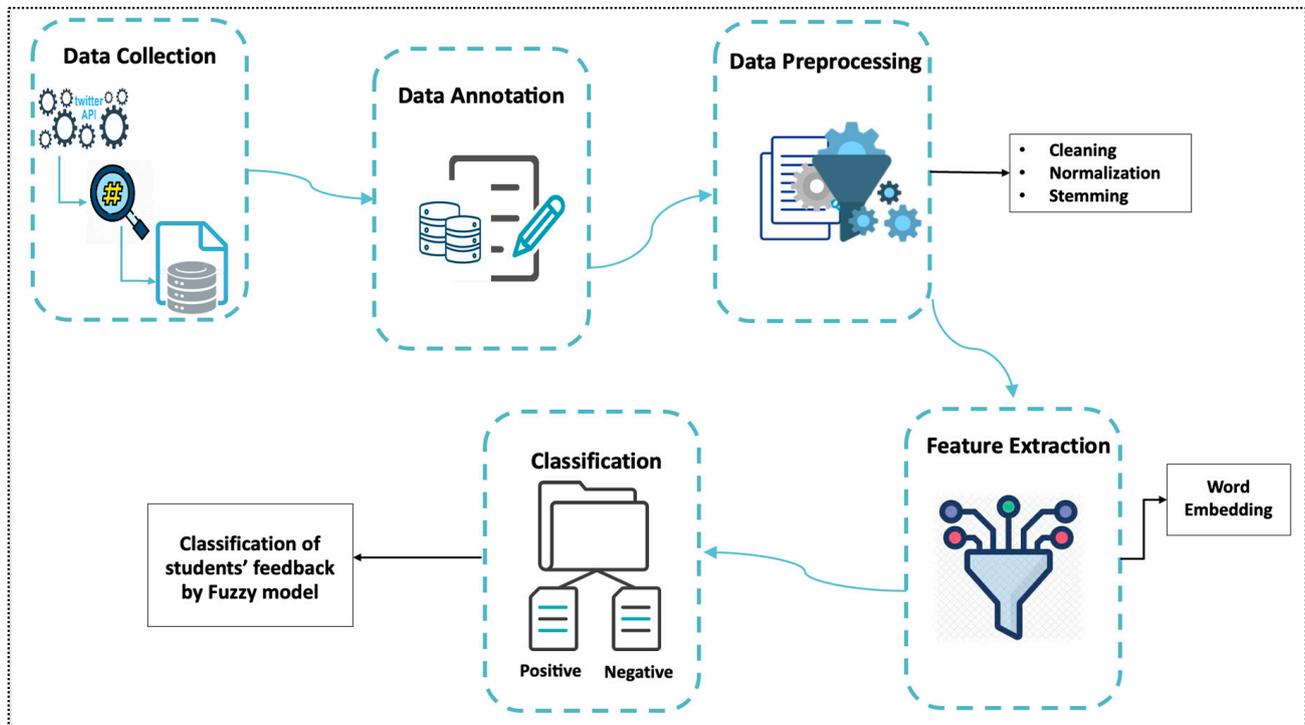
In a sequential structure, the fuzzy logic and the DNN are located in a successive manner. The fuzzy logic may be located before the DNN. Here, fuzzy logic is utilized to preprocess data before they enter the DNN. Alternatively, fuzzy logic is placed after the DNN [38]. This means that the fuzzy logic performs the further processing of the network output. In this way, fuzzy logic is exploited to work as a correction element of outputs or to improve results.

#### 2.3.3. Parallel Structure

Parallel structures are based on a joint learning approach. When the structure is parallelly designed, both the fuzzy logic and the DNN are placed separately from each other. This means that the data are processed independently in fuzzy logic and in the DNN. After that, the outcomes of both are combined to deliver the final output [39].

### 3. Methodology

The methodology followed by the model proposed in this work consists of five steps: data collection, data annotation, data preprocessing, feature extraction, and classification. The entire methodology is represented as a block diagram in Figure 1. The subsections below include a detailed explanation of the methodology's steps.



**Figure 1.** Overall methodology.

#### 3.1. Data Collection

We used Twitter application programming interfaces (APIs) to extract tweets about e-learning related to Saudi Arabian regions. We exploited the time periods when the discussion on the topic was popular among the people in Saudi Arabia such as the COVID-19 pandemic period. We collected tweets using a set of trending Saudi hashtags and keywords related to e-learning, and these are listed in Table 3.

**Table 3.** Hashtags and keywords.

Hashtags and Keywords	English Translation
#التعلم - الإلكتروني	E-learning
التعلم الإلكتروني	
#التعليم - الإلكتروني	E-teaching
التعليم الإلكتروني	
#الدراسة - عن - بعد	Distance learning
الدراسة عن بعد	
#الدراسة أونلاين	Online learning
الدراسة أونلاين	

In total, we collected approximately 5200 tweets that mentioned these hashtags and keywords. Irrelevant tweets that did not express student feedback on e-learning in Saudi Arabia or tweets containing ads and news were excluded. Additionally, duplicate tweets,

retweets, and short tweets containing fewer than three words were removed. This process yielded 3074 tweets out of 5200 tweets.

### 3.2. Data Annotation

The tweets were manually annotated according to two classes following the two-way classification of sentiment. As such, the labels were negative or positive. The annotation process was carried out by two Saudi persons who specialized in computer science and have the ability and experience to understand diverse Saudi dialects.

The tweets were labeled as follows:

- A tweet had a positive label when the student agreed with e-learning by expressing a positive opinion.
- A tweet had a negative label when the student disagreed with e-learning by expressing a negative opinion.

This step resulted in the distribution presented in Figure 2. Additionally, samples of the tweets are presented in Table 4, along with their corresponding labels.

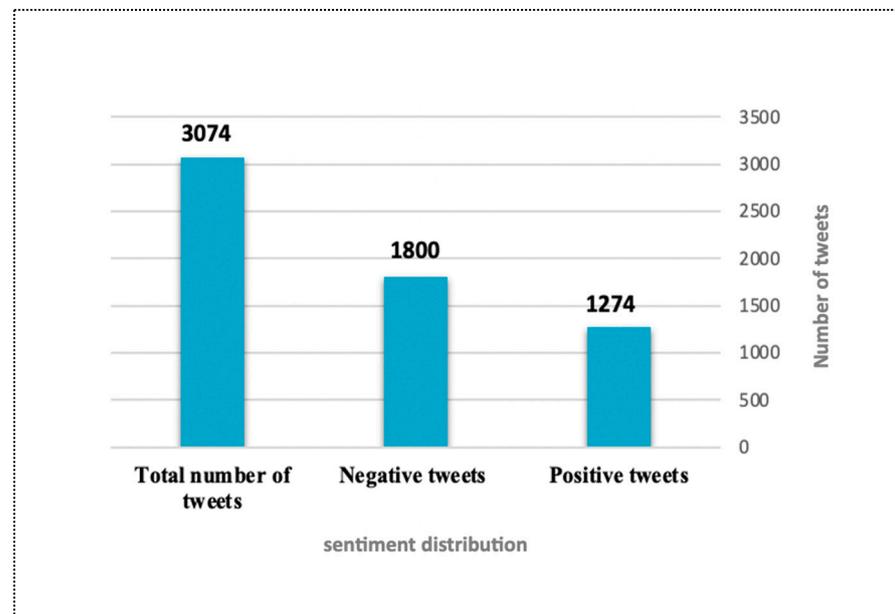


Figure 2. Distribution of sentiment.

Table 4. Samples of the labeled tweets.

Tweet Text	English Translation	Label
الدراسة عن بعد حلوه وممتعه وله الحمد مثارين بكل جد واجتهاد	Distance learning is nice and enjoyable, and thank God, we are continuing with interest.	Positive
دراستنا اونلاين فكرة فاشلة جداً للأسف ضاعت درجاتي	Our online study was a very unsuccessful idea, unfortunately, my grades were lost.	Negative
مستواي تحسن مع استخدام التعلم الالكتروني يكفي سهولة البحث	My studying level has improved with the use of e-learning, it is enough that searching for information has been easy.	Positive
أنا أعترف أنني مليت من الدراسة عن بعد سيئة وجداً متعبة	I admit that I am tired of distance learning, it is bad and very tiring.	Negative

### 3.3. Data Preprocessing

The collected tweets, similar to any other data obtained from online platforms, were in an unstructured format. Moreover, due to the tweets being short and having a limited number of characters, users usually use an informal writing style, including special characters, emojis, numbers, and links. Consequently, Twitter's raw data contain a lot of noise and need to undergo a preprocessing phase before any analysis task [40]. Furthermore, the tweets were written in the Arabic language, which is vibrant and morphologically rich. Given that the collected tweets were Saudi tweets, this means they contain a huge variety of written dialects.

Based on related works and practical considerations, the preprocessing phase of the texts written in the Arabic language should include several essential processing steps: data cleaning, normalization, and stemming [12,41]. They are effective steps that have a positive effect on the overall model's performance. We applied these steps to our dataset, and they are explained below.

#### 3.3.1. Data Cleaning

The data cleaning step included the elimination of unwanted and redundant data to reduce the complexity of the text and to prepare it for subsequent steps. English and Arabic numbers, emojis, URLs, symbols, and punctuation marks were considered unwanted data, so we removed them. Additionally, we removed repeated characters and elongation, and we used a single occurrence instead of duplication. Generally, a dataset that comprises tweets contains other unwanted information, such as hashtags and username mention, so these were also ignored and eliminated from the data. Additionally, we removed the stop words that are often distributed within texts and that were also considered a form of unwanted data.

#### 3.3.2. Normalization

The normalization process converts the various forms of a word within the data into a single standard and common form [10]. In particular, we performed orthographic normalization to unify certain Arabic letters that have more than one shape. Table 5 demonstrates the orthographic normalization that we applied. In addition, the normalization process included the removal of tashkeel, which is a diacritic added to Arabic letters. Generally, the normalization step minimizes text sparsity and its noise, as well as guarantees the consistency of the data.

**Table 5.** Orthographic normalization.

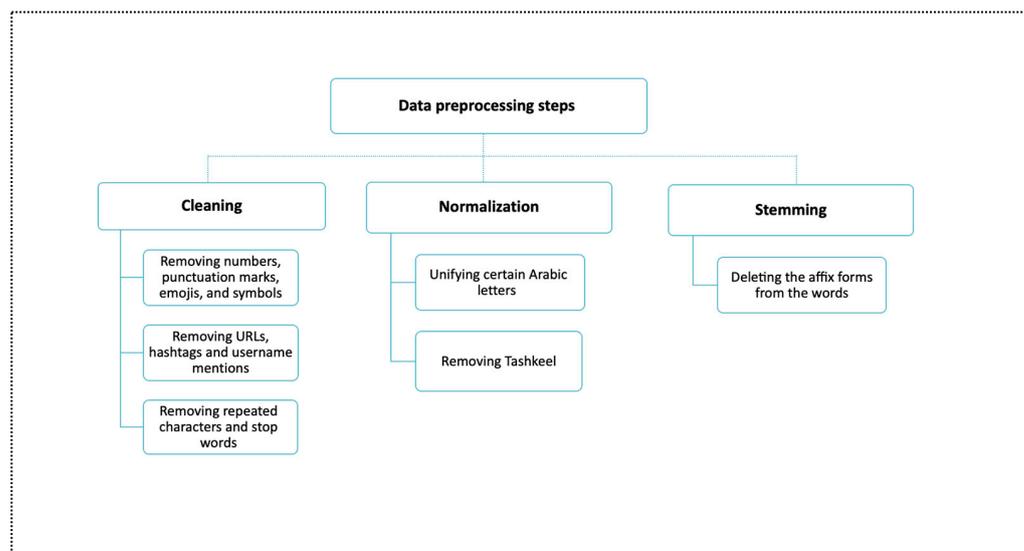
Shape of the Letter	Normalized to
أَ، آ، إِ، إِي	ا
ؤ	و
يَ، يِي	ي
ة	ه

#### 3.3.3. Stemming

Stemming refers to the process of standardizing words within a text by returning each word to its stem form [42]. This is carried out by deleting all the affix forms (i.e., infixes, prefixes, and suffixes) from the words. Consequently, the stemming process greatly contributes to reducing the complexity of texts. Even though stemming may not be necessary for some languages, it is certainly a highly significant step in Arabic preprocessing because the Arabic language has a very rich and complex morphology when compared with nonagglutinative languages. Several types of Arabic stemmers are available, including light stemmers and heavy stemmers. Regarding light stemmers, only prefixes and suffixes are

removed from the words, without returning them to their root. As previously mentioned, the Arabic language in our dataset was written in dialect style. However, a heavy or root stemmer may not work accurately with dialectal words [43]. Previous works that targeted Saudi dialectal Arabic, such as [12,15,44], chose to use light stemmers, as they worked well with the relevant data. Therefore, based on this and the findings of a previous comparative study by the authors of [45], the FARASA stemmer was selected. FARASA is an Arabic light stemmer that was presented by Abdelali et al. in 2016 [46].

In Figure 3, we summarize all the data preprocessing steps described in detail in the previous three subsections.



**Figure 3.** Data preprocessing steps.

### 3.4. Feature Extraction

In the domain of NLP tasks, the step of text feature extraction represents a critical part [47]. Textual data are not computable and cannot be directly processed by learning models. In fact, data must be expressed numerically so that they can be processed using different learning models. Therefore, we must somehow convert the given textual data into numerical values and a computer-readable format. This is implemented in the feature extraction step using various techniques. A meaningful feature can be extracted using these techniques to transform texts into numerical vectors. We used the word embedding method to convert the words in our dataset into numerical vectors.

Word embedding is a technique used to represent words. It works by capturing the relationships between words in textual data. These relationships suggest that words that occur in similar contexts have similar semantics. Therefore, words with similar meanings are represented [28]. To construct this representation, each word is individually mapped into one embedding. This embedding is represented as a  $d$ -dimensional vector with numerical values. We conducted this process by using an embedding layer, which is typically included as part of deep neural network models [7–48]. There are three parameters that must be specified for the embedding layer, namely, the input dimension, the output dimension, and the input length, and they are defined as follows:

- Input dimension: This is the number of all unique words in textual data, usually called the vocabulary size.
- Output dimension: The dimension of the generated vector is determined empirically, and it is usually set from 100 to 300.
- Input length: This is the number of words in each input sequence that has the maximum length.

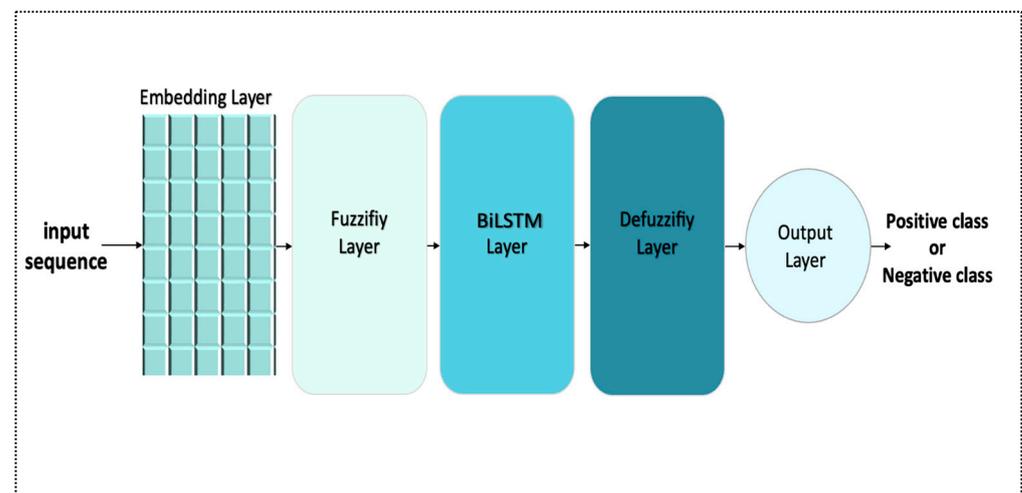
We performed the following sequential steps while preparing the textual data to be input into the embedding layer:

1. We applied tokenization to each input sequence; this is the process of splitting a text sequence into separate words or tokens.
2. We built an indicating dictionary using all the vocabulary from the whole input dataset to be assigned into unique indices. As a result, we obtained a known vocabulary size that defines the input dimension parameter of the embedding layer.
3. We applied the padding method, as the length of each input sequence in the dataset was expected to differ. For consistency, we padded certain additional tokens at the end of each input sequence to unify their lengths to the maximum length. As a result, the lengths of all input sequences were equal to the maximum length that defines the input length parameter of the embedding layer.

Next, these padded sequences with fixed lengths were ready to be passed to the embedding layer. Then, the embedding layer delivered its output as a two-dimensional numerical vector for each word in the input sequence.

### 3.5. Proposed Model

We propose a model that combines fuzzy logic and BiLSTM. This is a hybrid model and falls under the concept of the fusion of fuzzy logic and DNNs. We built our proposed model using an efficient architecture that fully integrates fuzzy logic and a DNN to take advantage of them as much as possible. The proposed structure is a multilayer model combining specific layers depicting fuzzy logic with the already-existing structure of BiLSTM. For the fuzzy logic, we configured two layers, named the fuzzify layer and the defuzzify layer. Each of these two layers performs a specific process of fuzzy logic. These layers were concatenated with three essential layers commonly used when BiLSTM is utilized for text classification. The architecture of the proposed model is shown in Figure 4.



**Figure 4.** The architecture of the proposed model.

It comprises five layers located in the following order: the embedding layer, the fuzzify layer, the BiLSTM layer, the defuzzify layer, and the output layer. Firstly, the input text is mapped into the embedding vectors via the embedding layer, which was explained in detail in the previous section. Then, the vectors are transformed into fuzzy values via the fuzzify layer. These fuzzy values then pass through the BiLSTM layer. The BiLSTM layer is basically two LSTM layers. The output from the BiLSTM layer is passed to the defuzzify layer to convert it into a crisp value. Finally, the output is obtained via a fully connected layer.

The next subsections explain the implementation of the layers related to fuzzy logic: the fuzzify layer and the defuzzify layer.

### 3.5.1. The Fuzzify Layer

The fuzzify layer is located after the embedding layer, so its input is the embedding vectors obtained from the embedding layer. The fuzzify layer processes these vectors into fuzzy values using the membership function. These values define the membership degree of the input vector to a certain fuzzy set. The type of membership function that we used is the Gaussian membership function, which is defined in Equation (2):

$$g(x, c, w) = e^{-\frac{(x-c)^2}{w^2}} \quad (2)$$

where  $x$  is the input vector previously produced by the embedding layer, and the parameters  $c$  and  $w$  denote the center value and the width value of the membership function, respectively. Figure 5 shows how these two values are represented.

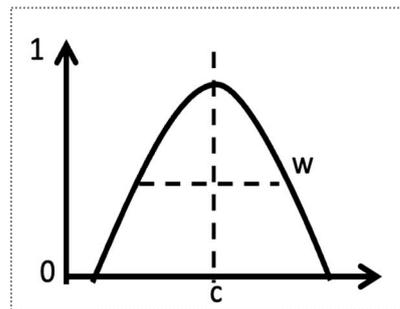


Figure 5. Gaussian membership function.

As defined in Equation (2), it is clear that the Gaussian membership function is determined by the  $c$  and  $w$  parameters. In our model, these parameters are trainable, i.e., the appropriate parameters can be obtained via the training and learning process that occurs in the neural network. Thus, we can obtain suitable shapes for the Gaussian membership function with which the region of each of the fuzzy sets is determined.

### 3.5.2. The Defuzzify Layer

The output obtained from the BiLSTM layer is still considered a fuzzy value that should be converted to a crisp value. Therefore, the defuzzification operation is performed in this layer. The defuzzify layer is trained with a ruleset and by a defuzzification function to produce a crisp output. This function is defined in Equation (3):

$$d(x, r) = \sum_{i=0}^n x_i r_i \quad (3)$$

where  $x$  is the fuzzy value (vector) obtained from the BiLSTM layer, and  $r$  is the ruleset obtained by training and adjusting the connection weights.

## 4. Experiments

This section presents the details of the experimental environment in terms of the hardware and software requirements; the experimental setup, including the implementation and setting of the hyperparameters; and the training procedures of the proposed model.

### 4.1. Hardware and Software Requirements

All the experiments were carried out on a MacBook Pro computer with an Apple M1 chip, an 8-core CPU, and up to 16 GB of memory. For implementation, we used Python 3.11.2, which is a high-level programming language. We wrote and executed the Python code of our experiments using Google Colab pro. The TensorFlow platform was selected to develop our proposed model. We used some of the efficient libraries that work on top of TensorFlow: Keras and scikit-learn [49].

#### 4.2. Implementation and Hyperparameter Setting

First, our dataset underwent the preprocessing steps previously discussed in Section 4.3. This dataset was saved in a file as clean data ready to be used in the experiments. Then, it was prepared for entry into the embedding layer by sequentially applying the tokenization, indexing, and padding processes. Then, each sequence in the dataset went through the embedding layer to extract the features. As previously explained in this paper, this layer requires the defining of three parameters: the input dimension, the input length, and the output dimension. The input dimension refers to the vocabulary size, which, in our dataset, equaled 15,816 words. The input length or the maximum length of each sequence in our dataset equaled a value of 55. We set the value of the output dimension for the embedding vector to a value of 150. The vector that exited the embedding layer passed through three layers: the fuzzify layer, the BiLSTM layer, and the defuzzify layer. Generally, there are several parameters that need to be tuned to compile a model. We set these parameters empirically through trial and error. Table 6 shows the optimal values of the parameters. The number of neurons in the BiLSTM layer was 32. We selected the optimizer ADAM, and we set the learning rate to 0.0001. We chose binary\_crossentropy as a loss function. We applied dropout to regularize the neural networks and to avoid the overfitting problem. Finally, the output layer generated the predicted result using the sigmoid activation function.

**Table 6.** Optimal values of the parameters.

Parameters.	Optimal Value
Number of neurons in BiLSTM	32
Dropout rate	0.5
Optimizer	ADAM
Learning rate	0.0001
Loss function	Binary_crossentropy
Activation function	Sigmoid

#### 4.3. Training Procedures

To evaluate the performance and to verify the efficiency of the proposed model, it needed to be trained using the dataset. Therefore, we conducted several experiments with two different training approaches. In the first type of experiment, we applied the k-fold cross-validation approach to split our data for training and testing. We chose to set the k-fold number to 5. This means that the dataset will be split into five parts in each iteration, four parts used for training and one for testing alternately in each iteration. In the second type of experiment, we used the train/test split approach at three different sizes. Among the whole dataset, we took a subset of data equal to 60%, 70%, or 80% as the size for the training set. Then, the remaining subset of data, that is, 40%, 30%, or 20%, was used for testing.

### 5. Results

In this section, we provide a comprehensive summary of the performance evaluation of the model and the results of its comparison with selected baseline models. Moreover, we present additional results related to the dataset.

#### 5.1. Experimental Results

We present the results for our proposed model based on the conducted experiments that differed in terms of the training approach, as explained in Section 4.3. In Table 7, we summarize the results obtained from the experiments using the four-performance metrics: accuracy, F1-score, recall, and precision [39]. We provide the obtained results of the five-fold cross-validation experiment. Typically, in this type of experiment, the overall performance of the model is evaluated by utilizing the resulting average for each of the performance metrics across the five folds. Moreover, we show the results of the three experiments that used the train/test split approach.

**Table 7.** Results of experiments of the proposed model.

Experiment	Accuracy	F1-Score	Recall	Precision
Five-fold cross-validation	0.840	0.826	0.821	0.832
(60–40%)	0.853	0.845	0.861	0.831
Train/test split	0.852	0.846	0.851	0.842
(70–30%)	0.852	0.846	0.851	0.842
(80–20%)	0.861	0.851	0.870	0.834

As shown in Table 8, the proposed model obtained good accuracy values in all experiments, ranging from 84% to 86%. We note that the highest accuracy, which equaled 86%, occurred in the train/test split experiment with 80% for the training set and 20% for the testing set. In the five-fold cross-validation experiment, the result of the F1-score measure was 83%, while in the other three experiments, it maintained a value equal to 84%. The resulting values of the precision measure were either 83% or 84% in all four experiments. Furthermore, the resulting values of the recall measure were 82% in the five-fold cross-validation experiment. However, in the three train/test split experiments, the recall measure values were 86%, 85%, and 87%.

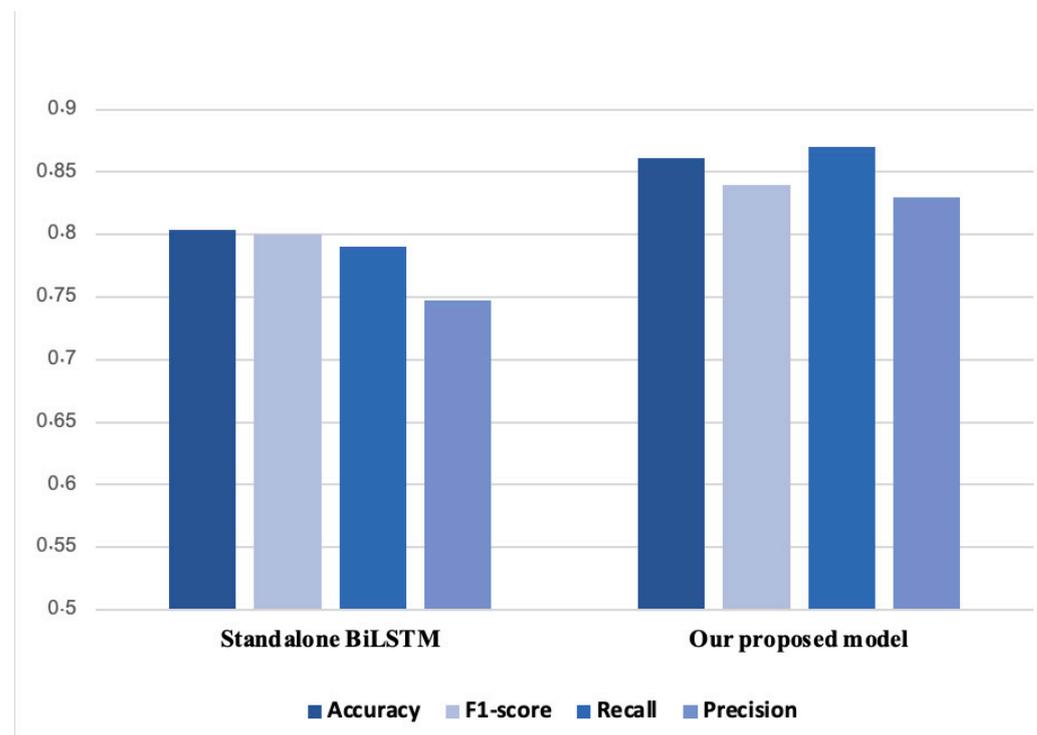
**Table 8.** Comparative results for the proposed model and the standalone BiLSTM.

Model	Accuracy	F1-Score	Recall	Precision
Standalone BiLSTM	0.804	0.767	0.790	0.747
Our Proposed Model	0.861	0.851	0.870	0.834

### 5.2. Comparative Results

The proposed model combines fuzzy logic with the well-known BiLSTM model. In fact, this standalone BiLSTM model has been used in most closely related works, such as [14,15]. Accordingly, we considered the standalone BiLSTM model as a baseline to be compared with our proposed model in order to demonstrate the efficiency of integrating fuzzy logic with DNN models and to show how this integration contributes to achieving the purpose of this study. For a fair comparison, we set the parameters of the standalone BiLSTM to be the same as those of the proposed model. We also first performed several separate experiments on the standalone BiLSTM model using different configurations. These experiments were carried out on the same dataset and using the same train/test split procedure (80–20%), to determine whether better results could be obtained with the other configurations. The experiments showed that the optimal parameters for our proposed model also gave the best results for the standalone BiLSTM. In Table 8 and Figure 6, only the best results for the standalone BiLSTM and our proposed model are shown. The results show that the performance improved with the proposed model compared to the standard BiLSTM, and it is apparent that our proposed model has the highest values in the obtained results. The proposed model outperformed the standalone BiLSTM by almost 6%. The standalone BiLSTM had an accuracy of 80% and an F1-score of 76%, whereas our proposed model had an accuracy of 86% and an F1-score of 84%.

Furthermore, we compared the performance of our proposed model with that of four prevalent and well-known machine learning models: naive Bayes (NB), random forest (RF), k-nearest neighbor (KNN), and logistic regression (LR). These models were used in experiments conducted by [11,12], which can be considered studies that are closely related to our work according to the type of analysis tasks carried out and the types of datasets used. Additionally, we performed a comparison with another prevalent machine learning model, namely, the decision tree (DT) and its fuzzy version (FDT), both of which were utilized in [50]. We have to mention that we have used TF-IDF-based features for learning the traditional machine learning models. Practically, each student's feedback has been converted to a vector of numerical data. Additionally, we carried out our experimental study on the same dataset and using the same train/test split procedure (80–20%).



**Figure 6.** Comparative results for the proposed model and the standalone BiLSTM.

Table 9 and Figure 7 demonstrate the results of all the above-mentioned machine learning models. In addition, the last row in Table 5 and the last column in Figure 6 show the results for the proposed model. The results for the machine learning models were comparable. The NP, RF, and DT performance results gave lower values ranging from 70% to 76% in terms of the accuracy and F1-scores, whereas the performance of the other three models was similar and gave results higher than those previously mentioned, ranging from 77% to 79% in terms of accuracy and from 75% to 78% in terms of F1-scores. However, the results for our model show that it still outperformed the other baseline models.

**Table 9.** Comparative results for the proposed model and machine learning models.

Model	Accuracy	F1-Score	Recall	Precision
<b>NB</b>	0.76	0.75	0.76	0.75
<b>RF</b>	0.76	0.75	0.71	0.80
<b>LR</b>	0.79	0.77	0.74	0.81
<b>KNN</b>	0.78	0.76	0.77	0.76
<b>DT</b>	0.72	0.70	0.70	0.71
<b>FDT</b>	0.77	0.75	0.75	0.76
<b>Our proposed model</b>	0.86	0.85	0.87	0.83

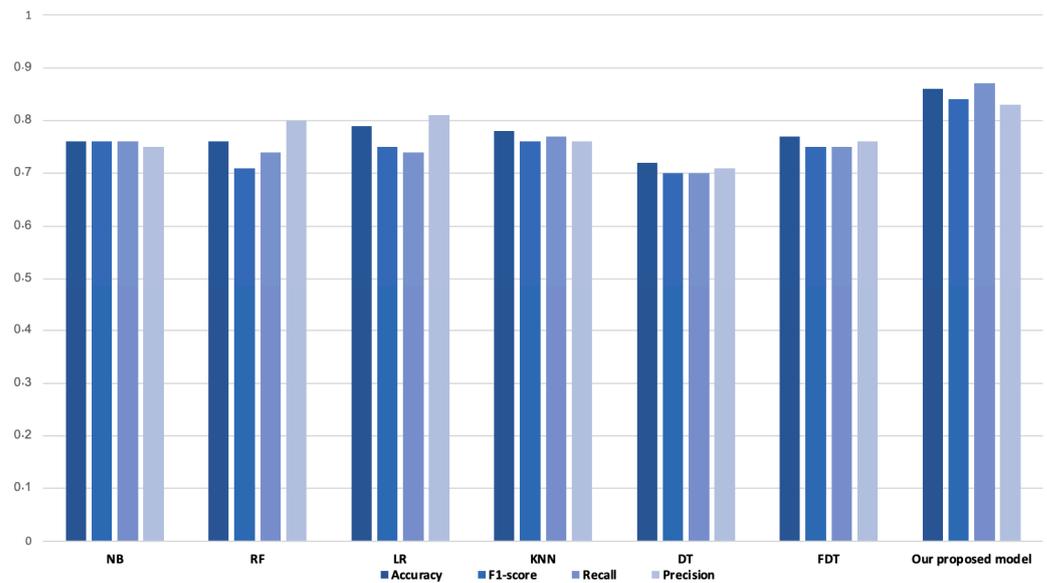


Figure 7. Comparative results for the proposed model and machine learning models.

### 5.3. Additional Results

In the distribution of the sentiment of our dataset, previously mentioned in Section 3.2, we noted that the students’ opinions we collected about e-learning tended to be negative sentiments rather than positive sentiments. Considering this, we attempted to investigate the issues contained within the negative student feedback. Therefore, we generated a word cloud that identified the most frequently words used in the opinions that were classified as being negative. In a word cloud, the words used in texts are visually represented, as shown in Figure 8, with each word having a certain size indicating how often it appears [10]. It can be noted that some words appeared more frequently than other words; they are exam, blackboard, online, difficult, hanging, and problem. Hence, we returned to the negative opinions that contained these words and studied them. It can be said that the students with negative feedback were mainly concerned with three issues regarding their experience of e-learning. These were the exams, blackboard platform, and technical problems.



Figure 8. Word cloud of the most frequent words in the negative opinions.

This type of statistical representation can be useful in helping researchers in this field and other corresponding fields, as well as decision-makers, better recognize the issues

that students are concerned about during their experience of e-learning and then find appropriate solutions [10].

## 6. Discussion

This work combined fuzzy logic with BiLSTM to develop a hybrid model. We used this model to analyze the sentiment of students' feedback about e-learning. This feedback was obtained from tweets posted by students expressing their opinions about e-learning in Saudi Arabia. Inherently, this collected feedback had ambiguous characteristics. We discuss these ambiguous characteristics in the next paragraph.

Normally, expressions of opinions written naturally by humans are texts that contain a lot of vagueness and noise [51–53]. The opinion orientation of personal text differs depending on the context or domain in which it is expressed. Moreover, the language of the feedback is Arabic and written in an informal way without writing standards, using different Saudi dialects. This means that these texts have a rich morphology and orthography [17,18]. The opinion terms are vague in nature and have unclear boundaries, and the diversity of dialects may lead to the meaning of these terms being interpreted differently. Textual data that possess all the above ambiguities impose significant difficulties in processing, feature extraction, and word representation, as well as in classification.

The findings confirm that our model was able to yield good results, and they also reveal that our model outperformed the selected baseline models. There are essential and significant interpretations of these results. The feature extraction process is the backbone impacting the performance of a model. Essentially, better feature extraction leads to better classification and prediction. In fact, the proposed model has an efficient architecture for the processing, feature extraction, and classification of textual data. We added two fuzzy layers, along with BiLSTM. We configured the fuzzify layer directly after the embedding layer that extracts the features. The fuzzify layer is in the middle, between the embedding layer and the BiLSTM layer. This fuzzify layer, with its membership functions, provides an effective additional procedure for extracting features and making them more separable. It reduces the potential vagueness of the data representation resulting from the feature extraction. Then, the output extracted features are more distinguishable for the subsequent layers.

Therefore, we can confirm that the proposed model worked well in addressing uncertainty issues. Thus, it makes an important contribution to analyzing texts written in the Arabic language. Evidently, combining fuzzy logic with the existing BiLSTM model afforded a further enhancement of the feature extraction and text classification, hence obtaining more accurate results. In addition, it provided better performance and allowed us to classify with higher accuracy than the standalone BiLSTM model.

The annotation process for the dataset represented an unexpected challenge. It was costly, consumed time and human effort, and required annotators who had knowledge of the Saudi dialects and experience with the annotation process. Consequently, the annotation process was not completed for the desired size of the dataset. This challenge produced the first limitation of this research, which was the limited size of the data. Therefore, in the future, we plan to investigate these dataset annotation challenges and increase the dataset size by completing the annotation process.

In the future, we intend to carry out an in-depth comparative study between our model and a set of transformers. Moreover, we will investigate the possibility of combining fuzzy models with a suitable transformer. Also, we plan to apply our model to different Arabic datasets.

**Author Contributions:** M.A. and F.F. contributed equally to all parts of this work. All authors have read and agreed to the published version of the manuscript.

**Funding:** This work was funded by the Deanship of Scientific Research at Qassim University, Saudi Arabia, under number (COC-2022-1-2-J-29742), during the academic year 1444 AH/2022 AD.

**Institutional Review Board Statement:** Not applicable.

**Informed Consent Statement:** Not applicable.

**Data Availability Statement:** The data presented in this study are available upon request from the corresponding author. The data are not publicly available due to privacy restrictions.

**Acknowledgments:** The authors gratefully acknowledge Qassim University, represented by the Deanship of Scientific Research, on the financial support for this research under the number (COC-2022-1-2-J-29742) during the academic year 1444 AH/2022 AD.

**Conflicts of Interest:** The authors declare no conflict of interest.

## References

1. Mujahid, M.; Lee, E.; Rustam, F.; Washington, P.B.; Ullah, S.; Reshi, A.A.; Ashraf, I. Sentiment analysis and topic modeling on tweets about online education during COVID-19. *Appl. Sci.* **2021**, *11*, 8438. [[CrossRef](#)]
2. Arambepola, N. Analysing the Tweets about Distance Learning during COVID-19 Pandemic using Sentiment Analysis. In Proceedings of the International Conference on Advances in Computing and Technology (ICACT-2020) Proceedings, Kelaniya, Sri Lanka, November 2020.
3. Kastrati, Z.; Dalipi, F.; Imran, A.S.; Pireva Nuci, K.; Wani, M.A. Sentiment analysis of students' feedback with nlp and deep learning: A systematic mapping study. *Appl. Sci.* **2021**, *11*, 3986. [[CrossRef](#)]
4. Almalki, J. A machine learning-based approach for sentiment analysis on distance learning from Arabic Tweets. *PeerJ Comput. Sci.* **2022**, *8*, e1047. [[CrossRef](#)] [[PubMed](#)]
5. Ulfa, S.; Bringula, R.; Kurniawan, C.; Fadhli, M. Student Feedback on Online Learning by Using Sentiment Analysis: A Literature Review. In Proceedings of the 2020 6th International Conference on Education and Technology, ICET 2020, Malang, Indonesia, 17 October 2020. [[CrossRef](#)]
6. Nasim, Z.; Rajput, Q.; Haider, S. Sentiment Analysis of Student Feedback Using Machine Learning and Lexicon Based Approaches. In Proceedings of the International Conference on Research and Innovation in Information Systems, ICRIS, 2017, Langkawi, Malaysia, 16–17 July 2017. [[CrossRef](#)]
7. Al-Bayati, A.Q.; Al-Araji, A.S.; Ameen, S.H. Arabic Sentiment Analysis (ASA) Using Deep Learning Approach. *J. Eng.* **2020**, *26*, 85–93. [[CrossRef](#)]
8. Subhashini, L.; Li, Y.; Zhang, J.; Atukorale, A.S. Integration of Fuzzy and Deep Learning in Three-Way Decisions. In Proceedings of the IEEE International Conference on Data Mining Workshops, ICDMW, Sorrento, Italy, 17–20 November 2020. [[CrossRef](#)]
9. Bedi, P.; Khurana, P. Sentiment Analysis Using Fuzzy-Deep Learning. In *Lecture Notes in Electrical Engineering*; Springer: Berlin/Heidelberg, Germany, 2020. [[CrossRef](#)]
10. Ali, M.M. Arabic sentiment analysis about online learning to mitigate COVID-19. *J. Intell. Syst.* **2021**, *30*, 524–540. [[CrossRef](#)]
11. Althagafi, A.; Althobaiti, G.; Alhakami, H.; Alsubait, T. Arabic Tweets Sentiment Analysis about Online Learning during COVID-19 in Saudi Arabia. *Int. J. Adv. Comput. Sci. Appl.* **2021**, *12*, 234147349. [[CrossRef](#)]
12. Aljabri, M.; Chrouf, S.M.B.; Alzahrani, N.A.; Alghamdi, L.; Alfehaid, R.; Alqarawi, R.; Alhuthayfi, J.; Alduhailan, N. Sentiment analysis of arabic tweets regarding distance learning in Saudi Arabia during the COVID-19 Pandemic. *Sensors* **2021**, *21*, 5431. [[CrossRef](#)] [[PubMed](#)]
13. Alkhalidi, S.; Alzuabi, S.; Alqahtani, R.; Alshammari, A.; Alyousif, F.; Alboaneen, D.A.; Almelih, M. Twitter Sentiment Analysis on Activities of Saudi General Entertainment Authority. In Proceedings of the ICCAIS 2020—3rd International Conference on Computer Applications and Information Security, Riyadh, Saudi Arabia, 19–21 March 2020. [[CrossRef](#)]
14. Alhuri, L.A.; Aljohani, H.R.; Almutairi, R.M.; Haron, F. Sentiment Analysis of COVID-19 on Saudi Trending Hashtags Using Recurrent Neural Network. In Proceedings of the International Conference on Developments in eSystems Engineering, DeSE, Liverpool, UK, 14–17 December 2020. [[CrossRef](#)]
15. Alqarni, A.; Rahman, A. Arabic Tweets-Based Sentiment Analysis to Investigate the Impact of COVID-19 in KSA: A Deep Learning Approach. *Big Data Cogn. Comput.* **2023**, *7*, 16. [[CrossRef](#)]
16. Deng, Y.; Ren, Z.; Kong, Y.; Bao, F.; Dai, Q. A Hierarchical Fused Fuzzy Deep Neural Network for Data Classification. *IEEE Trans. Fuzzy Syst.* **2017**, *25*, 1006–1012. [[CrossRef](#)]
17. Elfaiik, H.; Nfaoui, E.H. Deep Bidirectional LSTM Network Learning-Based Sentiment Analysis for Arabic Text. *J. Intell. Syst.* **2021**, *30*, 395–412. [[CrossRef](#)]
18. Heikal, M.; Torki, M.; El-Makky, N. Sentiment Analysis of Arabic Tweets Using Deep Learning. *Procedia Comput. Sci.* **2018**, *142*, 114–122. [[CrossRef](#)]
19. Biltawi, M.; Etaawi, W.; Tedmori, S.; Shaout, A. Fuzzy Based Sentiment Classification in the Arabic Language. In *Advances in Intelligent Systems and Computing*; Springer: Berlin/Heidelberg, Germany, 2018. [[CrossRef](#)]
20. Rattrout, A.; Ateeq, A. Sentiment Analysis on Arabic Content in Social Media; ACM International Conference Proceeding Series. In Proceedings of the 3rd International Conference on Future Networks and Distributed Systems, New York, NY, USA, 1–2 July 2019. [[CrossRef](#)]
21. Vidyapeetham, A.V. Fuzzy Based Machine Learning: A Promising Approach. 2012. Available online: [www.csi-india.org](http://www.csi-india.org) (accessed on 1 November 2023).

22. Das, R.; Sen, S.; Maulik, U. A Survey on Fuzzy Deep Neural Networks. *ACM Comput. Surv.* **2020**, *53*, 54. [[CrossRef](#)]
23. Tomer, M.; Kumar, M. Improving Text Summarization Using Ensembled Approach Based on Fuzzy with LSTM. *Arab. J. Sci. Eng.* **2020**, *45*, 10743–10754. [[CrossRef](#)]
24. Nguyen, T.-L.; Kavuri, S.; Lee, M. A fuzzy convolutional neural network for text sentiment analysis. *J. Intell. Fuzzy Syst.* **2018**, *35*, 6025–6034. [[CrossRef](#)]
25. Asghar, M.Z.; Subhan, F.; Ahmad, H.; Khan, W.Z.; Hakak, S.; Gadekallu, T.R.; Alazab, M. Senti-eSystem: A sentiment-based eSystem-using hybridized fuzzy and deep neural network for measuring customer satisfaction. *Softw. Pract. Exp.* **2021**, *51*, 571–594. [[CrossRef](#)]
26. Sivakumar, M.; Uyyala, S.R. Aspect-based sentiment analysis of mobile phone reviews using LSTM and fuzzy logic. *Int. J. Data Sci. Anal.* **2021**, *12*, 355–367. [[CrossRef](#)]
27. Es-Sabery, F.; Hair, A.; Qadir, J.; Sainz-De-Abajo, B.; Garcia-Zapirain, B.; Torre-Diez, I. Sentence-Level Classification Using Parallel Fuzzy Deep Learning Classifier. *IEEE Access* **2021**, *9*, 17943–17985. [[CrossRef](#)]
28. Alhumoud, S.O.; Al Wazrah, A.A. Arabic sentiment analysis using recurrent neural networks: A review. *Artif. Intell. Rev.* **2022**, *55*, 707–748. [[CrossRef](#)]
29. Wahdan, A.; AL Hantoobi, S.; Salloum, S.A.; Shaalan, K. A systematic review of text classification research based on deep learning models in Arabic language. *Int. J. Electr. Comput. Eng. (IJECE)* **2020**, *10*, 6629–6643. [[CrossRef](#)]
30. Seo, S.; Kim, C.; Kim, H.; Mo, K.; Kang, P. Comparative Study of Deep Learning-Based Sentiment Classification. *IEEE Access* **2020**, *8*, 6861–6875. [[CrossRef](#)]
31. Xu, G.; Meng, Y.; Qiu, X.; Yu, Z.; Wu, X. Sentiment analysis of comment texts based on BiLSTM. *IEEE Access* **2019**, *7*, 51522–51532. [[CrossRef](#)]
32. Vaswani, A.; Shazeer, N.; Parmar, N.; Uszkoreit, J.; Jones, L.; Gomez, A.N.; Kaiser, L.; Polosukhin, I. Attention is All You Need. In *Advances in Neural Information Processing Systems*; NIPS Foundation: La Jolla, CA, USA, 2017; pp. 5998–6008.
33. Brown, T.B.; Mann, B.; Ryder, N.; Subbiah, M.; Kaplan, J.; Dhariwal, P.; Neelakantan, A.; Shyam, P.; Sastry, G.; Askell, A.; et al. Language Models Are Few-Shot Learners. In *Proceedings of the 34th International Conference on Neural Information Processing Systems (NIPS'20)*, Virtual, 6–12 December 2020; Curran Associates Inc.: Red Hook, NY, USA, 2020; pp. 1877–1901.
34. Jesse, V.; Ali, M.; Lav, V.; Caiming, X.; Richard, S.; Nazneen, R. BERTology Meets Biology: Interpreting Attention in Protein Language Models. In *Proceedings of the 9th International Conference on Learning Representations, (ICLR)*, Virtual Event, Austria, 3–7 May 2021. [[CrossRef](#)]
35. Zadeh, L.A. Fuzzy sets. *Inf. Control* **1965**, *8*, 338–353. [[CrossRef](#)]
36. Tashtoush, Y.M.; Orabi, D.A.A.A. Tweets Emotion Prediction by Using Fuzzy Logic System. In *Proceedings of the 2019 6th International Conference on Social Networks Analysis, Management and Security, SNAMS*, Granada, Spain, 22–25 October 2019. [[CrossRef](#)]
37. Talpur, N.; Abdulkadir, S.J.; Alhussian, H.; Hasan, M.H.; Aziz, N.; Bamhdi, A. Deep Neuro-Fuzzy System application trends, challenges, and future perspectives: A systematic survey. *Artif. Intell. Rev.* **2023**, *56*, 865–913. [[CrossRef](#)] [[PubMed](#)]
38. Zheng, Y.; Xu, Z.; Wang, X. The Fusion of Deep Learning and Fuzzy Systems: A State-of-the-Art Survey. *IEEE Trans. Fuzzy Syst.* **2022**, *30*, 2783–2799. [[CrossRef](#)]
39. Talpur, N.; Abdulkadir, S.J.; Alhussian, H.; Hasan, H.; Aziz, N.; Bamhdi, A. A comprehensive review of deep neuro-fuzzy system architectures and their optimization methods. *Neural Comput. Appl.* **2022**, *34*, 1837–1875. [[CrossRef](#)]
40. Alqurashi, T. Stance Analysis of Distance Education in the Kingdom of Saudi Arabia during the COVID-19 Pandemic Using Arabic Twitter Data. *Sensors* **2022**, *22*, 1006. [[CrossRef](#)] [[PubMed](#)]
41. Hadwan, M.; Al-Sarem, M.; Saeed, F.; Al-Hagery, M.A. An Improved Sentiment Classification Approach for Measuring User Satisfaction toward Governmental Services' Mobile Apps Using Machine Learning Methods with Feature Engineering and SMOTE Technique. *Appl. Sci.* **2022**, *12*, 5547. [[CrossRef](#)]
42. Oussous, A.; Benjelloun, F.-Z.; Lahcen, A.A.; Belfkih, S. ASA: A framework for Arabic sentiment analysis. *J. Inf. Sci.* **2020**, *46*, 544–559. [[CrossRef](#)]
43. Alassaf, M.; Qamar, A.M. Improving Sentiment Analysis of Arabic Tweets by One-way ANOVA. *J. King Saud Univ. Comput. Inf. Sci.* **2022**, *34*, 2849–2859. [[CrossRef](#)]
44. Bahamdain, A.; Alharbi, Z.H.; Alhammad, M.M.; Alqurashi, T. Analysis of Logistics Service Quality and Customer Satisfaction during COVID-19 Pandemic in Saudi Arabia. *Int. J. Adv. Comput. Sci. Appl.* **2022**, *13*. [[CrossRef](#)]
45. Almazrua, A.; Almazrua, M.; Alkhalifa, H. Comparative Analysis of Nine Arabic Stemmers on Microblog Information Retrieval. In *Proceedings of the 2020 International Conference on Asian Language Processing, IALP 2020*, Kuala Lumpur, Malaysia, 4–6 December 2020. [[CrossRef](#)]
46. Abdelali, A.; Darwish, K.; Durrani, N.; Mubarak, H. Farasa: A Fast and Furious Segmenter for Arabic. In *Proceedings of the NAACL-HLT 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Proceedings of the Demonstrations Session*, San Diego, CA, USA, June 2016. [[CrossRef](#)]
47. Oueslati, O.; Cambria, E.; Ben HajHmida, M.; Ounelli, H. A review of sentiment analysis research in Arabic language. *Futur. Gener. Comput. Syst.* **2020**, *112*, 408–430. [[CrossRef](#)]
48. Abdelminaam, D.S.; Neggaz, N.; Gomaa, I.A.E.; Ismail, F.H.; Elsayy, A.A. ArabicDialects: An efficient framework for arabic dialects opinion mining on twitter using optimized deep neural networks. *IEEE Access* **2021**, *9*, 97079–97099. [[CrossRef](#)]

49. Zahidi, Y.; El Younoussi, Y.; Al-Amrani, Y. A powerful comparison of deep learning frameworks for Arabic sentiment analysis. *Int. J. Electr. Comput. Eng.* **2021**, *11*, 745–752. [[CrossRef](#)]
50. Bahuguna, A.; Yadav, D.; Senapati, A.; Saha, B.N. A unified deep neuro-fuzzy approach for COVID-19 twitter sentiment classification. *J. Intell. Fuzzy Syst.* **2022**, *42*, 4587–4597. [[CrossRef](#)]
51. Liu, H.; Burnap, P.; Alorainy, W.; Williams, M.L. A Fuzzy Approach to Text Classification With Two-Stage Training for Ambiguous Instances. *IEEE Trans. Comput. Soc. Syst.* **2019**, *6*, 227–240. [[CrossRef](#)]
52. Fkih, F.; Moulahi, T.; Alabdulatif, A. Machine Learning Model for Offensive Speech Detection in Online Social Networks Slang Content. *WSEAS Trans. Inf. Sci. Appl.* **2023**, *20*, 7–15. [[CrossRef](#)]
53. Haddad, O.; Fkih, F.; Omri, M.N. Toward a prediction approach based on deep learning in Big Data analytics. *Neural Comput. Appl.* **2023**, *35*, 6043–6063. [[CrossRef](#)]

**Disclaimer/Publisher’s Note:** The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.