electronics

MDPI

Article

ABMM: Arabic BERT-Mini Model for Hate-Speech Detection on Social Media

Malik Almaliki [1], Abdulqader M. Almars [1], Ibrahim Gad [2] and El-Sayed Atlam [1,2,*]

[1] College of Computer Science and Engineering, Taibah University, Yanbu 966144, Saudi Arabia
[2] Faculty of Science, Tanta University, Tanta 31527, Egypt
* Correspondence: satlam@taibahu.edu.sa

Abstract: Hate speech towards a group or an individual based on their perceived identity, such as ethnicity, religion, or nationality, is widely and rapidly spreading on social media platforms. This causes harmful impacts on users of these platforms and the quality of online shared content. Fortunately, researchers have developed different machine learning algorithms to automatically detect hate speech on social media platforms. However, most of these algorithms focus on the detection of hate speech that appears in English. There is a lack of studies on the detection of hate speech in Arabic due to the language's complex nature. This paper aims to address this issue by proposing an effective approach for detecting Arabic hate speech on social media platforms, namely Twitter. Therefore, this paper introduces the Arabic BERT-Mini Model (ABMM) to identify hate speech on social media. More specifically, the bidirectional encoder representations from transformers (BERT) model was employed to analyze data collected from Twitter and classify the results into three categories: normal, abuse, and hate speech. In order to evaluate our model and state-of-the-art approaches, we conducted a series of experiments on Twitter data. In comparison with previous works on Arabic hate-speech detection, the ABMM model shows very promising results with an accuracy score of 0.986 compared to the other models.

Keywords: deep learning; BERT Models; Arabic Twitter; machine learning; hate speech

Citation: Almaliki, M.; Almars, A.M.; Gad, I.; Atlam, E.-S. ABMM: Arabic BERT-Mini Model for Hate-Speech Detection on Social Media. *Electronics* **2023**, *12*, 1048. https://doi.org/10.3390/electronics12041048

Academic Editor: Arkaitz Zubiaga

Received: 19 January 2023
Revised: 13 February 2023
Accepted: 13 February 2023
Published: 20 February 2023

1. Introduction

Social media platforms, such as WhatsApp, Facebook, and Twitter, are widely used for exchanging and creating content. They provide users with a convenient and easy way to share information quickly and efficiently, making them a valuable source of information [1–3]. However, social media platforms can also be a means for disseminating offensive and harmful content. The propagation of unpleasant and harmful content on social media can have a significant damaging influence on the experience of users as well as the overall quality of online shared content [4]. Hate speech is an example of such harmful content which can be defined as speech that attacks or incites hatred against someone or something based on their perceived identity, such as ethnicity, religion, nationality, or sexual orientation [5–8]. According to a recent study conducted by the Anti-Defamation League (ADL), 41% of Americans had experienced online hate and harassment [9].

Therefore, building technologies that can automatically detect hate speech has become extremely critical. Fortunately, researchers in the fields of computer science and machine learning have developed algorithms that can automatically identify hate speech on social media platforms. These algorithms can help to mitigate the spread of this type of harmful content on these platforms. However, most of these algorithms focused on the detection of hate speech that appears in English [10–14]; there is a lack of studies on the detection of Arabic hate speech due to the language's complex nature. Prior studies on Arabic social media content have mostly concentrated on either recognizing vulgar or obscene language [15] or on the detection of hate speech that can be distinguished from

it. The Arabic language is the main language in 6 of the top 11 countries with the highest social hostilities index, which evaluates crimes motivated in part by religion or race. This highlights the importance of addressing hate speech in Arabic, as this type of content can have serious negative consequences in communities [16].

The variety and complexity of Arabic morphology present certain difficulties for Arabic NLP researchers to detect hate speech on social media [17]. Dialectal Arabic is more frequently used in casual situations (e.g., social media platforms) than Modern Standard Arabic. Various dialects of Arabic exist within and between countries as well as among regions within the same country. There are no established grammar or spelling rules for dialectal Arabic, in contrast with Modern Standard Arabic [17]. It is common for similar-looking words to have different meanings in different dialects, which makes the language more ambiguous in general. For instance, the Arabic term "عافية" Afia in the Maghrebi Arabic language means "fire", whereas in Gulf Arabic it implies "health". The fact that Arabic has far fewer resources than English makes it more difficult. An Arabic hate vocabulary is one of the tools that is lacking, and it can be highly helpful in studies on cyber hate detection.

Furthermore, there are a number of difficulties in identifying hate and abusive speech on social media. Finding common patterns and trends in data is difficult due to the vast amount of diverse content that is uploaded to social media networks. Additionally, user-generated social network data includes noisy content that presents technological difficulties for text mining and linguistic analysis, such as incorrect grammar, misspelled words, internet slang, abbreviations, word lengthening, and text written in multi-lingual scripts. Finally, social network policies usually prohibit users from publishing any unethical or unlawful content. Due to this, users post information that seems legitimate but very subtly escalates to the extremes of hate speech. As a result, it is difficult to develop tools that can detect hate speech automatically.

This research investigates and develops techniques to automatically extract hate speech from Arabic tweets. To achieve that, the Arabic BERT-Mini Model (ABMM) is presented. We use the BERT model to analyze Twitter data and categorize the results into three categories: normal, abuse, and hate speech. Our model is evaluated by analyzing 9352 Arabic tweets and categorizing them into the above categories. To evaluate the effectiveness of our model, we compare its performance to several state-of-the-art models in the field. The results demonstrate that ABMM produced up to 98% accuracy.

This work makes the following contributions:

- As far as we are aware, this paper is among a small number of research efforts that have focused on addressing the issue of identifying hate speech on Arabic social media, specifically Twitter.
- This paper introduces a novel model called ABMM for identifying hate speech on social media.
- For the purpose of hate detection, the first Arabic lexicon of hate terms and an Arabic dataset are established, and these resources are made public to stimulate further research.
- The analysis of the ABMM language model on the hate speech categorization task demonstrates an improvement over the state-of-the-art models.
- ABMM is useful for monitoring and tracking hate speech on Arabic social media platforms by decision-makers, such as governments.

This work is organized based on the following sections. Section 2 discusses the related work. Section 3 demonstrates the proposed ABMM model. Section 4 discusses the experimental results and discussion, along with potential applications. Section 6 presents the research's findings, conclusions, and future directions.

2. Related Work

There have been several machine learning models proposed for identifying hate speech on social media platforms and other online communities. The topic of hate speech

in English-language social media content has been studied in great detail. In [11], the authors proposed a supervised method for identifying hate speech on Twitter. According to their findings, the use of supervised classifiers was found to be more effective in binary classification tasks than ternary classifiers. Burnap and Williams [18] have created a further binary classifier that uses a labeled dataset to distinguish between hateful and non-hateful tweets.

The textual characteristics of a message can be helpful in detecting hate speech. Using textual information from a user's tweets prior to them declaring support or opposition to ISIS, Magdy et al.'s classifier predicts whether a user supports ISIS or is against it [19]. Spatial and temporal features have also been used to identify hate speech. Jihadist et al. developed a model for identifying hate speech contents based on linguistic and temporal factors [20]. A method was developed by Mubarak et al. [15,21] for automatically building and growing a list of vocabulary words, which would subsequently be used to identify offensive tweets.

There has been some interest in using deep learning models to detect hate speech on social media platforms. Character n-grams are more accurate predictive variables for identifying racist and sexist tweets than word n-grams, according to Waseem and Hovy's theory. The researchers found that adding location information decreased performance, while adding gender as an additional variable only slightly improved it. Another research applied an LSTM-based classifier based on gradient-boosted decision trees (GBDTs) to detect hate speech. Compared to N-gram-based classifiers, this model outperformed them significantly [11].

Advanced models such as BERT have attracted the attention of scholars and practitioners [22–24]. BERT-large and BERT-base are two BERT models that were first presented by Devlin et al. [23] for automatically detecting hate speech in English. The proposed models were pre-trained based on quite substantial internet-extracted corpora. This results in enormous memory footprints and high computing demands. The proposed models make an effort to remedy some of the old models' flaws by enhancing either performance [25] or inference speed [22].

BERT models were also used to pre-train the Arabic language. As an example, Devlin et al. created a multilingual model that covers more than 100 languages, including Arabic [23]. According to Antoun et al., a BERT-based model named Arabert is pre-trained for Arabic content [14]. Around 24 terabytes of text were used for the model's pre-training. Similar to this, Abdul-Mageed et al. [19] trained an Arabic BERT model they called MARBERT using one billion tweets. Even though these models have been used to classify Arabic text, it is unclear if one is more effective than the other at detecting hate speech, or if the training process has affected their effectiveness.

A stacking BERT-based model for Arabic sentiment analysis was presented by Hasna et al. [26]. Transformer-based models were recently regarded as the most advanced model for several languages because of their excellent performance in sentiment analysis. However, Arabic sentiment analysis still needs to be more accurate. In this study, we used various BERT models to offer a stacking architecture for Arabic sentiment analysis. By combining various small, freely accessible datasets, we also produced a sizable Arabic sentiment analysis dataset. Experimental results show that the suggested approach is more accurate in classification than a single-model architecture. Muhammad et al. [27] suggested BERT semi-supervised learning of Arabic dialects.

The popularity of BERT led to more models supporting additional languages, including Arabic. BERT Models for Arabic Text Classification: A Systematic Review were proposed by [28]. Researchers and practitioners are paying more and more attention to bidirectional encoder representations from transformers (BERT), which has emerged as a crucial method for processing natural language. This method is successful for a variety of reasons, including its ability to predict words from context. It also has the ability to be pre-trained using a great deal of plain text data available online.

However, the uses of BERT models to classify Arabic text are currently limited to a few examples, despite several trials for detecting hate speech in Arabic. The purpose of this paper is to start filling in this gap by bringing together several Arabic BERT models that are used for text classification. The effectiveness of other machine learning models and deep learning models is also examined. We also discovered from the literature that there is still room for further research on hate-speech detection in Arabic. Our paper introduces a new model called ABMM for detecting speech on social media.

3. Proposed Methodology

This study proposes an effective approach called the Arabic BERT-Mini Model (ABMM) for detecting Arabic hate speech on social media platforms, namely Twitter. The structure of this model is described and summarized in Figure 1, which gives a high-level view of the system. The following subsections will discuss more illustrations of the model.

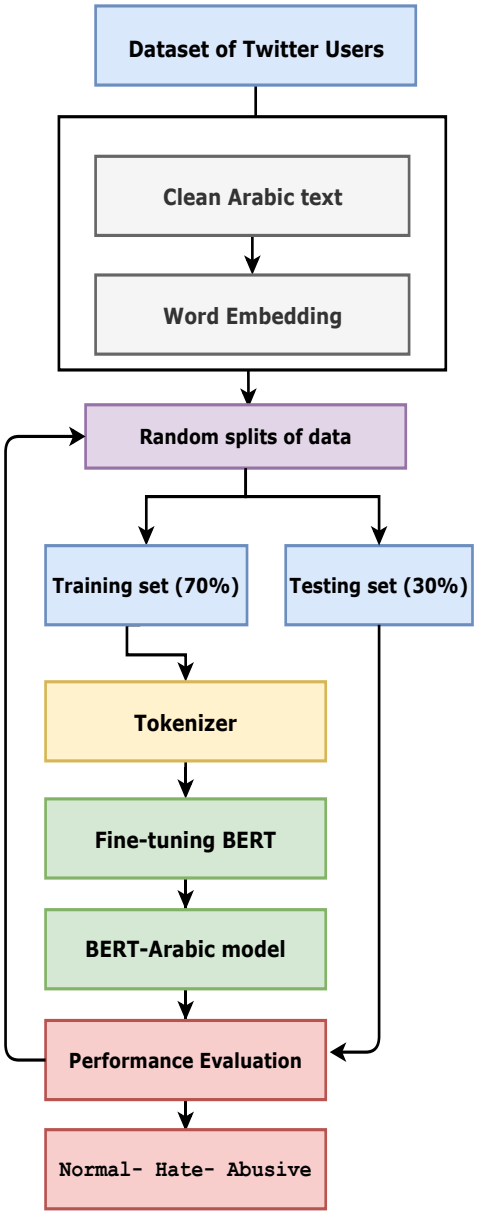


Figure 1. Architecture of the proposed framework.

3.1. Tweets Pre-Processing

A number of pre-processing operations have been carried out in order to establish some degree of structure and consistency in the tweets. The following steps are included in these processes:

1. Deleting Arabic punctuation using an existing list of Arabic punctuation in the NLTK library.
2. Arabic text normalization: In the Arabic language, we have different variations for representing some letters which are: (a) Letter (Alef) "أ" (which has the forms "أ إ آ ا") all these four letters are normalized to become one letter which is "ا". (b) Letter (Alef Maqsora) "ى" (which can be mistaken and written as "ي"). It will be normalized to ي. (c) Letter (Taa Marbouta) "ة" has been normalized to "ه". (d) The Arabic dash utilized to expand the word has been removed (e.g., "الـــها" to "الها").
3. Removal of the general structure of tweets such as @username, URLs, and hashtags.

The manual annotation process was carried out on the data set by a number of specialists. In order to eliminate any potential annotator bias, the labeled data were reviewed by two different volunteers. The volunteers were provided with a manual that would assist them in differentiating the various hate groups. In spite of this, correctly identifying the type of hatred being discussed is not a simple task, because the context and the user's intentions can make a significant difference. The final stage provides a new column that contains the categories of tweets. This column is condensed down to a single character for the sake of simplicity.

Word embeddings are used to accurately represent text in the corpus. This technique offers a more effective way of representing the text that is contained within the corpus when compared to more traditional methods of vectorization. Before the input corpus can be utilized with Keras embedding, it should be tokenized and encoded as integers. In order to accomplish this goal, we make use of the tokenizer that can be found in the Keras library. To train the deep learning (DL) model, the list of tokenized words of varying lengths should be converted into a sequence of an exact length. To complete the conversion, the character labels in the label column are converted to integers for compatibility with the embedding layer.

3.2. Arabic BERT-Mini Model (ABMM)

Transformers work like neural networks (NNs) with encoders and decoders. The encoder–decoder NN model uses a bidirectional LSTM encoder and an attention mechanism decoder to generate word embeddings. An interesting thing about transformer-based models is that they are entirely based on attention mechanisms and do not need any connections that go back and forth. Since its first introduction in 2014, the transformer model has shown results for tasks including text generation, machine translation, sentence paraphrasing, and language modeling. The transformer-based architecture outperforms recurrent neural networks (RNN) and their extensions with LSTM [29–31] and GRU in NLP tasks [32].

A language representation paradigm called BERT was released in 2018 by Jacob Devlin and his colleagues from Google [28]. Since it was first presented, it has rapidly evolved into a standard that is utilized across the field in natural language processing research. BERT was developed to be a bidirectional model for predicting words in both the left and right contexts. This is in contrast to existing models of language representation, which capture the context in a unidirectional manner. BERT was created as an unsupervised model that was trained using plain text from the web in the majority of languages. This model edition can be used to produce predictions in a variety of NLP-related activities (text classification) [33].

Bidirectional encoder representations from transformers are abbreviated as BERT. The two primary BERT implementation techniques are feature extraction and fine-tuning. The

BERT model's architecture is retained during feature extraction, meaning that its parameters are "frozen". To complete a job, features are taken from the pre-trained BERT model and then passed to the classifier model. By putting more layers into the initial BERT design, the parameters of the model are fine-tuned. The model is trained using these newly developed layers for the downstream tasks [34].

The BERT encoder is composed of a multilayer bidirectional transformer (as shown in Figure 2). A bidirectional representation is trained by conditioning both the right and left backgrounds in every layer of the representation. The BERT method is used to identify the vector representation of each token in a text. Depending on the model architecture, there are four different pre-trained versions of the original BERT. These versions are referred to as BERT-mini, BERT-medium, Bert-Base, and Bert-Large [32]. Table 1 presents the different versions of the large, base, medium, and mini-size BERT Models [35].

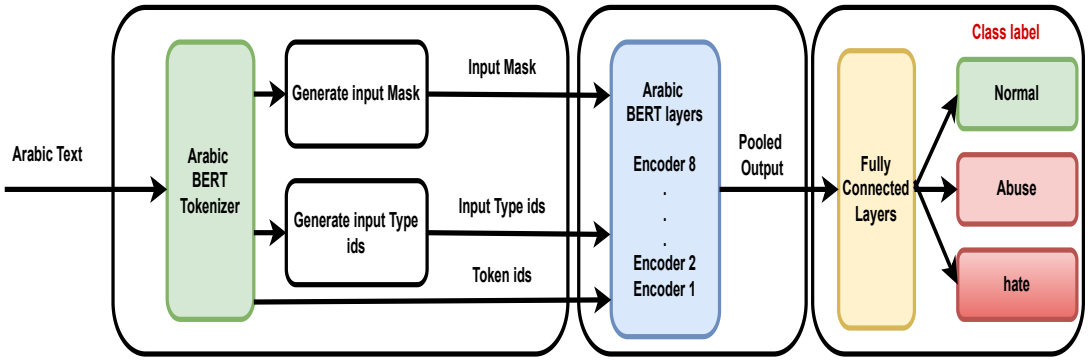


Figure 2. Architecture of the Arabic BERT-mini model.

Table 1. The description of the large, base, medium, and mini-size Arabic BERT Models. Adapted with permission from Ref. [35]. 2023, Springer Nature Switzerland AG.

	Mini	Medium	Base	Large
# Hidden layers	4	8	12	24
Hidden size	256	512	768	1024
# Attention heads	4	8	12	16
# Parameters	11 M	42 M	110 M	340 M

In this paper, mini-BERT is used to train our model depending on the size of the collected dataset. The model has been optimized for Arabic in various iterations. Similarly, the architecture of the basic BERT model is identical to that of the mini-BERT. However, it was pre-trained on 8.2 billion tokens that totaled 95 gigabytes of text and were taken from Wikipedia and other Arabic sources [36]. Additionally, the model was pre-trained on 11 million tweets written in various dialects of Arabic.

Figure 2 illustrates the visual representation of the proposed architecture for the Arabic BERT-mini model. The architecture consists of three main blocks. The first block explains how we performed text pre-processing by breaking down the sentence into tokens using an Arabic BERT tokenizer. This step returns the input mask, type IDs, and token IDs. The second block shows the layers of the Arabic BERT model. The BERT has only eight encoders. A representation vector of $512 \times 4 \times 128$ size with 16 batch sizes is generated from the output of the last four hidden layers. In the pooling operation, the output is concatenated and flattened, and a dense layer is later applied. The third block describes the classifier, which uses the dropout layer, the fully connected layer, and the softmax function to label the data into three classes: normal, hateful, and abusive.

3.3. BERT Embedding Layer

In this study, BERT has been pre-trained on two distinct but related NLP tasks that take advantage of the bidirectional capability: next sentence prediction and masked language modeling. When studying languages related to human languages, BERT may attain high levels of accuracy by dealing with uncertainty, which is the most challenging component of natural language analysis. In contrast to the traditional word2vec embedding layer, which gives static context-independent word vectors, the BERT layers provide dynamic context-dependent word embeddings by taking the full sentence as input and extracting data from the entire sentence. Unlike previous models, BERT tokenizes inputs first, then adds extra tokens at the beginning [CLS] and end [SEP]. Then, BERT models utilize a self-attention mechanism that allows them to process multiple tokens at once, which is why special embedded tokens are required for the next sentence prediction challenge. Transformer encoders read the complete phrase sequence at once, rather than sequentially from left to right or right to left. This makes it bidirectional, yet non-directional is more correct. Based on its surroundings (to the left and right of a word), the model is able to determine a word's context.

3.4. Arabic Tokenizer

The Arabic language is hard to tokenize because its morphology is so rich and complicated. Typically, the definition of a token is a sequence of one or more characters that is both preceded and followed by a space. This concept is useful for languages that do not have agglutinative forms, such as English. Since Arabic tokenization is a pre-processing step, numerous systems have integrated it. Different levels of an Arabic tokenizer are possible to design, and these levels are determined by the scope of the linguistic study. Tokenization models are as follows: (1) a guesser for Tokenization. (2) Tokenization and morphological analysis. (3) Tokenization based on morphological structure [37]. This paper uses WordPiece tokenization throughout its training process for BERT [35]. It denotes that a single word can be divided into multiple different sub-words. Because the vectorBERT ascribed to a word is determined by the context in which it is used, the same word may have numerous vectors.

4. Experimental Results

The performance of the proposed model on Arabic hate speech dataset is evaluated in this section. The dataset that was used to evaluate the Bert model is described in this section. Then, we compare the proposed model's performance against other approaches using accuracy, recall, precision, and the F1 score.

4.1. The Arabic Hate-Speech (AHS) Dataset Description

In this experiment, the Twitter platform served as the primary source of data collection. Since deep learning methods are used, these models usually need a large corpus to train them and obtain results that are meaningful and useful. In this section, we will discuss the steps involved in creating and organizing the corpus.

Any researcher can investigate and create solutions utilizing the public tweets provided by the Twitter API platform. It has endpoints that can be used for a variety of purposes, such as the Twitter Streaming API, which is utilized to fetch data that is occurring in real-time, and the Twitter Search API, which is employed to fetch earlier tweets. In order to use these APIs researchers need to register for Twitter developer accounts [38].

A Python package called Tweepy was used to collect tweets, and a cursor was used to search for tweets. We have compiled a list of hashtags that trigger and attract hateful content. It is impossible to achieve a balance between the number of hate tweets and those that are not hateful, and this ideal does not reflect the reality of social media. We simply chose a list of hashtags that are guaranteed to contain both content that promotes hatred and content that does not promote hatred in order to maintain an authentic real-life

environment. Figure 3 demonstrates a few instances of hate speech that can be found on Twitter. These tweets are directed at specific individuals or groups.

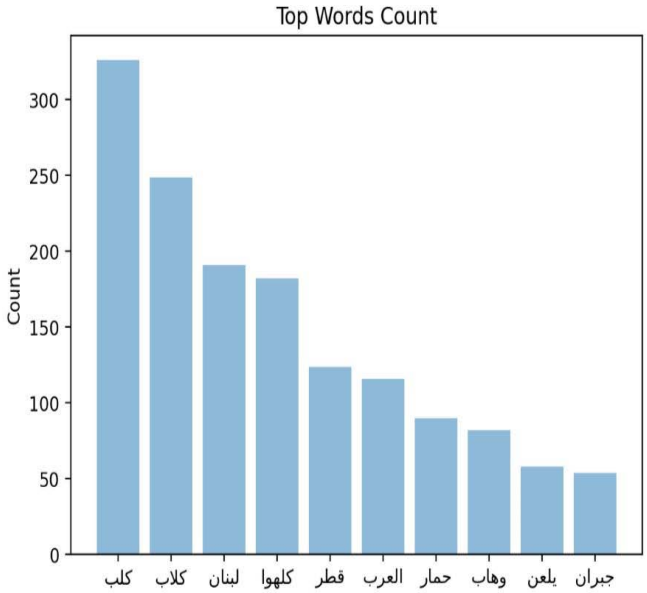


Figure 3. Examples of most hate words in dataset. (The translation of the Arabic words are: كلب: Dog, كلاب: Dogs, لبنان: Lebanon, كلهوا: All, قطر: Qatar, العرب: Arabs, حمار: Donkey, وهاب: Wahab, يلعن: Damned, جيران: Neighbors).

The dataset contains three forms of speech (normal, abusive, and hate speech). Previous studies on Arabic hate-speech recognition concentrated mostly on binary hate categorization. By integrating the prior hate speech features with local Arab culture, we explained what constitutes each of the distinct categories of hate speech in Table 2. After combining all of the tweets that were produced by the different hashtags into a single corpus, we then shuffled the tweets within that corpus. We were able to collect a corpus of 9352 tweets and label them. Figure 4 indicates that the majority of tweets fall into the "Normal" category, while the minority fall into the "Hate" category.

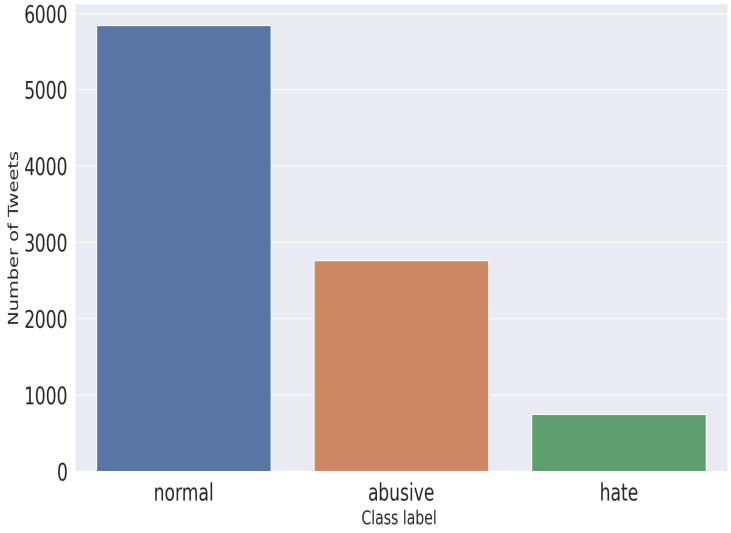


Figure 4. Tweets in each category.

Table 2. Statistics of the datasets and labels.

	Category	Number of Tweets
0	Normal	5840
1	Abusive	2764
2	Hate	748

4.2. Evaluation Metrics

A variety of metrics are used to compare the performance and effectiveness of all classification models. In this study, precision (Equation (1)), recall (Equation (2)), F1 scores (Equation (3)), and accuracy (Equation (4)) are used to evaluate the performance of the models.

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

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

$$F - measure = 2 \cdot \frac{precision \times recall}{precision + recall} \tag{3}$$

$$Accuracy = \frac{TP + TN}{TP + FP + TN + FN} \tag{4}$$

4.3. Experimental Setting

In this study, machine learning techniques are utilized to study Arabic hate speech and multiclass classification. The machine learning models that are used in this paper were chosen due to their nonlinearity and ability to learn from the collected data. To predict Arabic hate speech, standard machine learning models such as support vector machine (SVM), decision trees (DT), k-nearest neighbors (KNN), naive Bayes (NB), random forest, and gradient boosting are used. For comparison, two deep-learning models were chosen: LSTM and CNN-LSTM. These baselines enable us to see how much better the proposed model performs compared to the intended baselines. We have built the baseline models using SciKit-learn and Tensorflow in Python for an unexplored problem (Arabic hate speech and multiclass classification).

All of the models used in the experiments are trained with the exact compilation and training settings. The dataset is divided by using the train_test_split() function from SciKit-learn as shown in Table 3. In order to achieve a more efficient learning process, the training procedure in Keras must be configured by providing the necessary parameters before using the compile method to train the chosen classifiers. The compilation function should always include the following parameters: Since the classification problem is multi-class categorical, the loss function we choose is "sparse categorical cross-entropy".

Table 3. The details of training and testing of the dataset.

All	Training	Testing
100%	70%	30%
9352	7481	1871

The Adam algorithm, which is a well-known adaptive learning rate optimization algorithm, is only one of several techniques that can be implemented to make the learning process as efficient as possible. Experiments are carried out with 5, 10, and 15 epochs, and the results of 10 epochs are satisfactory because more epochs could lead to overfitting.

The baseline models are evaluated based on ten-fold cross-validation, which is carried out on the training set. The training data is divided into ten portions using the ten-fold approach, and one of those parts is chosen to be the test data. The model will be trained on the remaining nine parts in the following steps, and their aggregate performance will be compared to the test set results. In order to accomplish this task, the cross-validation function in SciKit-Learn is a powerful tool for optimizing and tuning the parameters of machine-learning models.

4.4. Results

This study uses several state-of-the-art methods to conduct a comparative study. Table 4 shows the classification report for the linear SVM method. It can be observed that the linear SVM method achieved better results compared with traditional machine learning models with an accuracy rate of 96% on average. However, as we have indicated previously, because we are dealing with the problem of an unbalanced dataset, we will base our evaluation of the models on the recall and accuracy metrics, which are shown in Table 4. We can see from the results that linear SVM is able to differentiate tweets that do not include hate speech, as evidenced by the high recall that was produced as a consequence of the analysis. However, SVM is unable to distinguish between hate speech classes (it has a small recall in the other classes). According to the findings, SVM is not particularly effective when used on multi-class datasets that are not well balanced, but it does a good job in situations involving binary classification. The values of accuracy, precision, and recall are 96%, 96%, and 96%, respectively. In comparison with other models, the decision tree and random forest classifiers scored the lowest in terms of accuracy, precision, and recall.

Table 4. The classification report for Linear SVC.

	Precision	Recall	F1-Score	Support
Normal	0.93	0.95	0.94	549
Hate	0.95	0.83	0.88	139
Abusive	0.97	0.98	0.98	1183
Accuracy		0.96		1871
Macro avg	0.95	0.92	0.93	1871
Weighted avg	0.96	0.96	0.96	1871

Two deep learning techniques have also been applied to demonstrate their performance in the Arabic hate speech multi-classification problem. Table 5 demonstrates the results produced by the LSTM model. The results of Table 5 show more consistency in accuracy, precision, and recall. Additionally, the recall is higher in this case, which shows that the LTSM model was strong enough to recognize a tweet containing Arabic hate speech. Moreover, as indicated by the high recall (95.1%), hate speech tweets were clearly easier for the model to identify. Furthermore, it is clear that the LSTM model outperformed several classical machine learning algorithms in the identification of tweets that did not contain hate speech. Figure 5 shows the accuracy curve of the LSTM model. As illustrated in Figure 5, the test accuracy generally increases as the epoch size increases as well. The train accuracy curve, on the other hand, outperforms the test accuracy curve. In classification problems, the confusion matrix is employed as a performance measure for machine learning models.

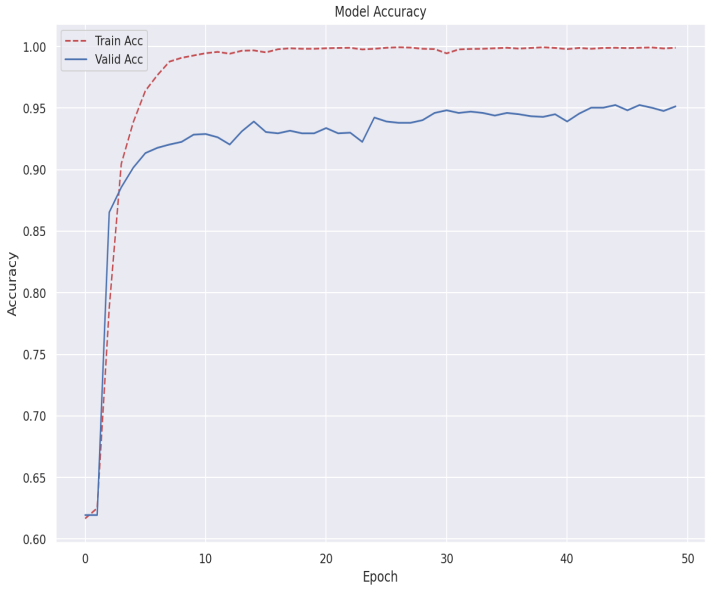


Figure 5. Accuracy curve for the LSTM model.

Table 5. The results of training and testing for the LSTM model.

Model	Training Accuracy	Testing Accuracy	Precision	Recall
LSTM	99.9%	95.1%	95.1%	95.1%

A hybrid deep learning model named CNN-LTSM was also employed for hate speech detection [38]. However, the model achieved the lowest results compared with the proposed model and LSTM in terms of accuracy, precision, and recall. The fine-tuned BERT model was compared to nine algorithms, as shown in Table 6. Table 6 shows the experiment results of CNN-LTSM in comparison with other classifiers.

Table 6. Overall metrics for each model.

Classifier	N-Gram	Accuracy	Precision	Recall
ABMM	**3**	**0.986**	**0.986**	**0.986**
LSTM	3	0.951	0.951	0.951
CNN+LTSM	3	0.75	0.72	0.75
Linear SVC	3	0.960	0.960	0.960
SVC	3	0.953	0.953	0.953
Multinomial NB	3	0.951	0.951	0.951
Bernoulli NB	3	0.936	0.937	0.936
K-Nearest Neighbors	3	0.934	0.934	0.934
SGD	3	0.676	0.675	0.676
Decision Tree	3	0.668	0.682	0.668
Random Forest	3	0.611	0.373	0.611

Compared with the machine learning models, the proposed ABMM model achieved the highest scores. The ABMM model classification report is presented in Table 7. In general, the ABMM model is effective; the recall of all hate-speech classes has improved in a noticeable way, which amounts to an increase in recall that is 3% higher.

Table 7. Classification report of the ABMM model.

	Precision	Recall	F1-Score	Support
Normal	0.99	0.99	0.99	578
Hate	0.98	0.99	0.99	281
Abusive	0.97	0.91	0.94	77
Accuracy		0.99		936
Macro avg	0.98	0.97	0.97	936
Weighted avg	0.99	0.99	0.99	936

From Table 6, the values with the highest precision and recall are marked in bold. It is clear that the proposed BERT-mini model outperforms all the state-of-the-art models in terms of the performance of the three hate classes. This may be credited to BERT's enhanced ability to handle multiclass classification over other classifiers. This demonstrates BERT's robustness in capturing and classifying multiple classes.

The AUC (area under the curve) ROC (receiver operating characteristics) is applied to determine the effectiveness of the multi-class classifier as well as to display its performance. This is one of the most essential performance assessment measures to use when assessing the efficacy of any classification model. It provides an indication of the level at which the model is capable of differentiating between various classes. The higher AUC, the more accurately the model is able to predict that class. Figure 6 shows the AU ROC curve using the BERT-mini model. From this figure, it is clear that the ROC curve for all classes is close to the upper left corner. For example, the area under the curve for class 0 (normal) is 0.99.

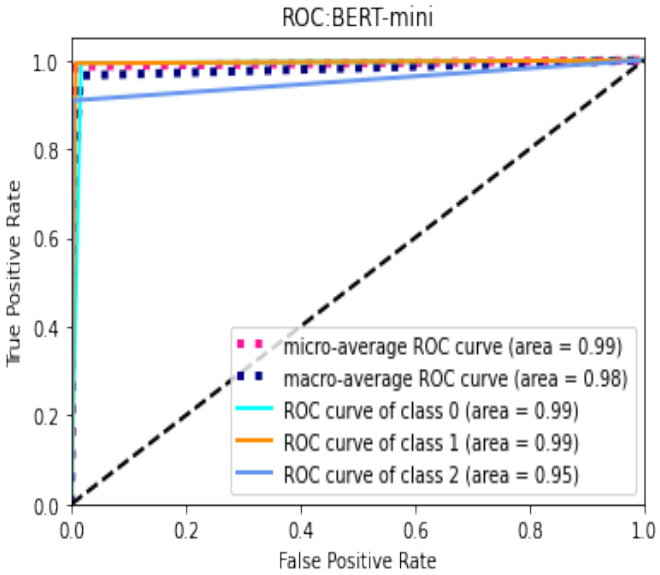
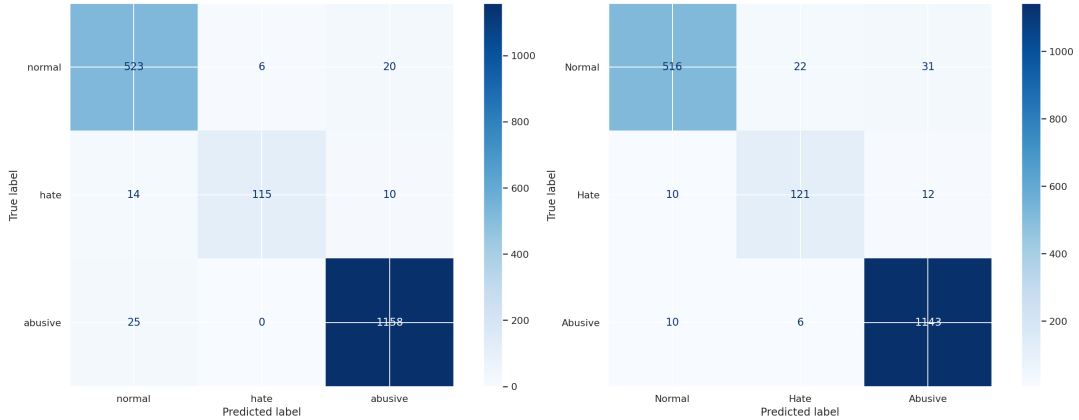
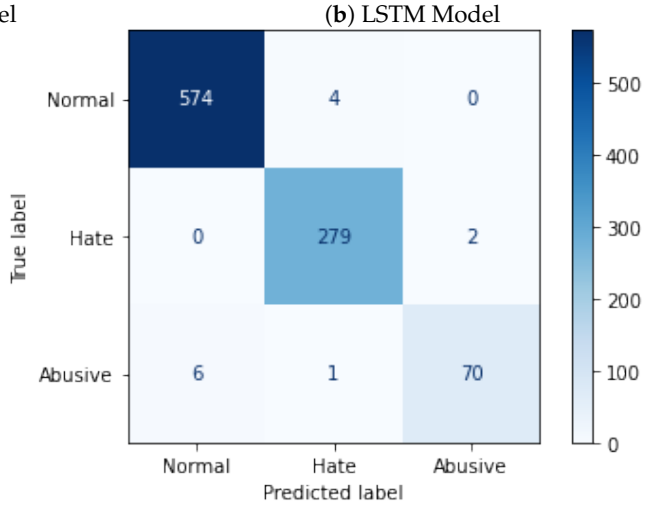


Figure 6. ROC curve using ABMM model.

Figure 7 shows the confusion matrix of the top models. It is a performance metric for evaluating the effectiveness of classification algorithms where the output can be two or more classes. In the confusion matrix, there are four possible combinations of predicted and actual values. It is used to calculate recall, specificity, precision, and accuracy, as well as AUC-ROC curves.



(**a**) Linear SVC Model (**b**) LSTM Model



(**c**) ABMM Model

Figure 7. Confusion matrices for the top models.

5. Discussion

The above results show that ABMM model outperforms traditional approaches. The proposed model achieved an accuracy score equal to 0.986 for hate-speech detection. In order to evaluate ABMM models, we used the Keras "Evaluate" function, which returns the model's loss and metrics values. In terms of recall of the three hatred classes (see Table 6), the BERT-mini model surpasses the LSTM model, which can be attributed to BERT's superior capacity to handle multi-class categorization over LSTM. The LSTM, on the other hand, was more effective than traditional machine learning at identifying tweets that were free of hate speech. This proves BERT's capability to capture and identify multiple classes. There is a higher degree of precision, recall, and consistency in F1 scores with the BERT-mini model. Moreover, the recall, in this case, is higher, which shows that the BERT-mini model is capable of detecting Arabic hate speech in a tweet.

Another comparison with the state-of-the-art models illustrates the efficacy of the proposed model. Table 8 shows a comparison between the proposed ABMM model and the current state-of-the-art AraBERT, CNN+LTSM, and DT models. The highest recall data in Table 8 is highlighted in bold, indicating that the suggested deep learning ABMM model has a reasonably high recall. However, in terms of recall of the three hatred classes, the ABMM model surpasses the AraBERT and CNN+LTSM models, which can be attributed to BERT's ability to handle multi-class categorization over CNN+LTSM. To sum up, in this study, we discovered that applying the BERT model to Arabic text is useful and efficient, and can help with hate-speech detection.

Table 8. The comparative study with state-of-the-art methods.

Models	Accuracy	Precision	Recall	F1-Score
ABMM	**0.986**	**0.986**	**0.986**	**0.986**
AraBERT [39]	0.96	0.95	0.96	0.95
CNN+LTSM [38]	0.75	0.72	0.75	0.73
DT [40]	0.5619	0.56	0.56	0.56

Despite the advantages of the proposed model, it has two main limitations. The first limitation is that it does not consider additional features, such as emoji descriptions in text. Adding such features can improve the model's performance in distinguishing hate speech. We plan to extend our model to include emojis in the near future. A second limitation of the model is that it does not provide an interpretation of classification decisions. As a result, it is difficult for humans to interpret the reasons behind the model's decision. Explainable AI could be integrated into the model to provide explanations for the results.

6. Conclusions

Hate speech can be harmful to individuals and cause mental pain, as well as damage online communities by establishing a hostile environment. In this paper, we developed a new model called the Arabic BERT-Mini Model (ABMM) that can accurately detect Arabic hate speech on Twitter. First, word embedding techniques are used to create word representations. Then, the BERT model was used to recognize and categorize hate speech on Twitter into three main classes. A comparison of the proposed models with traditional machine learning models and state-of-the-art models was conducted. Based on the results of the comparison, the BERT model performed better than the baseline in multi-classifying hate classes. In future work, the researchers will be able to add data from another platform, such as Facebook, which is the most popular platform in the Middle East, to increase the size of the dataset and improve the training of neural networks. In addition, we will be able to experiment with a variety of word representation methods, including using the AraVec project for representing the text in the future. The standard memory and CPU of our system prevented us from implementing additional deep learning layers. Future research could focus on better hardware and experimental settings. This would be possible by utilizing a powerful GPU.

Author Contributions: Conceptualization, I.G., E.-S.A. and A.M.A.; methodology, I.G. and A.M.A.; software, A.M.A.; validation, E.-S.A., M.A. and A.M.A.; formal analysis, I.G.; investigation, A.M.A.; resources, E.-S.A.; data curation, M.A. and A.M.A.; writing—original draft preparation, M.A.; writing—review and editing, A.M.A., I.G. and M.A.; visualization, I.G.; supervision, E.-S.A.; project administration, M.A. All authors have read and agreed to the published version of the manuscript.

Funding: This research has received no external funding.

Institutional Review Board Statement: Not applicable.

Data Availability Statement: The data that support the findings of this study are available from the corresponding author upon reasonable request.

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

References

1. Chen, X.; Sin, S. 'Misinformation? What of it?' Motivations and individual differences in misinformation sharing on social media. *Proc. Am. Soc. Inf. Sci. Technol.* **2013**, *50*, 1–4. [CrossRef]
2. Müller, K.; Schwarz, C. Fanning the Flames of Hate: Social Media and Hate Crime. *J. Eur. Econ. Assoc.* **2020**, *19*, 2131–2167. [CrossRef]
3. Almars, A.M.; Almaliki, M.; Noor, T.H.; Alwateer, M.M.; Atlam, E. HANN: Hybrid Attention Neural Network for Detecting Covid-19 Related Rumors. *IEEE Access* **2022**, *10*, 12334–12344. [CrossRef]

4. Nobata, C.; Tetreault, J.; Thomas, A.; Mehdad, Y.; Chang, Y. Abusive Language Detection in Online User Content. In Proceedings of the 25th International Conference on World Wide Web, Montreal, QC, Canada, 11–15 April 2016. [CrossRef]
5. Waseem, Z.; Hovy, D. Hateful Symbols or Hateful People? Predictive Features for Hate Speech Detection on Twitter. In Proceedings of the NAACL-HLT, San Diego, CA, USA, 12–17 June 2016; pp. 88–93.
6. Davidson, T.; Warmsley, D.; Macy, M.; Weber, I. HAutomated Hate Speech Detection and the Problem of Offensive Language. In Proceedings of the International AAAI Conference on Web and Social Media, Montreal, QC, Canada, 15–18 July 2017; pp. 88–93.
7. Fortuna, P.; Nunes, S. A Survey on Automatic Detection of Hate Speech in Text. *ACM Comput. Surv.* **2018**, *51*, 1–30. [CrossRef]
8. Sharma, S.; Agrawal, S.; Shrivastava, M. Degree based classification of harmful speech using twitter data. *arXiv* **2018**, arXiv:1806.04197.
9. Almars, A.M. Attention-based Bi-LSTM model for Arabic depression classification. *Comput. Mater. Contin.* **2022**, *71*, 3091–3106. [CrossRef]
10. Djuric, N.; Zhou, J.; Morris, R.; Grbovic, M.; Radosavljevic, V.; Bhamidipati, N. Hate Speech Detection with Comment Embeddings. In Proceedings of the 24th International Conference on World Wide Web, Florence, Italy, 18–22 May 2015; pp. 1–6. [CrossRef]
11. Badjatiya, P.; Gupta, S.; Gupta, M.; Varma, V. Deep Learning for Hate Speech Detection in Tweets. In Proceedings of the 26th International Conference on World Wide Web Companion, Perth, Australia, 3–7 April 2017; pp. 1–6. [CrossRef]
12. Gitari, N.D.; Zhang, Z.; Damien, H.; Long, J. A Lexicon-based Approach for Hate Speech Detection. *Int. J. Multimed. Ubiquitous Eng.* **2015**, *10*, 215–230. [CrossRef]
13. Silva, L.; Mondal, M.; Correa, D.; Benevenuto, F.; Weber, I. Analyzing the Targets of Hate in Online Social Media. *Proc. Int. AAAI Conf. Web Soc. Media* **2021**, *10*, 687–690. [CrossRef]
14. Kwok, I.; Wang, Y. Locate the Hate: Detecting Tweets against Blacks. *Proc. AAAI Conf. Artif. Intell.* **2013**, *27*, 1621–1622. [CrossRef]
15. Mubarak, H.; Darwish, K.; Magdy, W. Abusive Language Detection on Arabic Social Media. In Proceedings of the First Workshop on Abusive Language Online, Vancouver, BC, Canada, 4 August 2017; Association for Computational Linguistics: Vancouver, BC, Canada, 2017; pp. 1–6. [CrossRef]
16. Mohammad, A.S. Mother tongue versus Arabic: The post-independence Eritrean language policy debate. *J. Multiling. Multicult. Dev.* **2016**, *37*, 523–535. .:10.1080/01434632.2015.1080715. [CrossRef]
17. Darwish, K.; Magdy, W.; Mourad, A. Language Processing for Arabic Microblog Retrieval. In Proceedings of the 21st ACM International Conference on Information and Knowledge Management, Maui, HI, USA, 29 October–2 November 2012; Association for Computing Machinery: New York, NY, USA, 2012; pp. 2427–2430. [CrossRef]
18. Burnap, P.; Williams, M.L. Cyber Hate Speech on Twitter: An Application of Machine Classification and Statistical Modeling for Policy and Decision Making. *Policy Internet* **2015**, *7*, 223–242. [CrossRef]
19. Magdy, W.; Darwish, K.; Weber, I. #FailedRevolutions: Using Twitter to study the antecedents of ISIS support. *First Monday* **2016**. [CrossRef]
20. Kaati, L.; Omer, E.; Prucha, N.; Shrestha, A. Detecting Multipliers of Jihadism on Twitter. In Proceedings of the 2015 IEEE International Conference on Data Mining Workshop (ICDMW), Atlantic City, NJ, USA, 14–17 November 2015; pp. 1–6. [CrossRef]
21. Atlam, E.S.; Fuketa, M.; Morita, K.; Aoe, J.-i. Similarity measurement using term negative weight and its application to word similarity. *Inf. Process. Manag.* **2000**, *36*, 717–736. [CrossRef]
22. Sanh, V.; Debut, L.; Chaumond, J.; Wolf, T. DistilBERT, a distilled version of BERT: Smaller, faster, cheaper and lighter. *arXiv* **2019**, arXiv:1910.01108.
23. Devlin, J.; Chang, M.; Lee, K.; Toutanova, K. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. *arXiv* **2018**, arXiv:1810.04805.
24. Bonifazi, G.; Corradini, E.; Ursino, D.; Virgili, L. New Approaches to Extract Information From Posts on COVID-19 Published on Reddit. *Int. J. Inf. Technol. Decis. Mak.* **2022**, *21*, 1385–1431. [CrossRef]
25. Liu, Y.; Ott, M.; Goyal, N.; Du, J.; Joshi, M.; Chen, D.; Levy, O.; Lewis, M.; Zettlemoyer, L.; Stoyanov, V. Roberta: A robustly optimized bert pretraining approach. *arXiv* **2019**, arXiv:1907.11692.
26. Chouikhi, H.; Chniter, H.; Jarray, F. Stacking BERT based Models for Arabic Sentiment Analysis. In Proceedings of the 13th International Joint Conference on Knowledge Discovery, Knowledge Engineering and Knowledge Management, Valletta, Malta, 25–27 October 2021; pp. 1–6. [CrossRef]
27. Zhang, C.; Abdul-Mageed, M. No Army, No Navy: BERT Semi-Supervised Learning of Arabic Dialects. In Proceedings of the Fourth Arabic Natural Language Processing Workshop, Florence, Italy, 28 July–2 August 2019 ; pp. 1–6. [CrossRef]
28. Alammary, A.S. BERT Models for Arabic Text Classification: A Systematic Review. *Appl. Sci.* **2022**, *12*, 5720. [CrossRef]
29. Malki, Z.; Atlam, E.S.; Hassanien, A.E.; Dagnew, G.; Elhosseini, M.A.; Gad, I. Association between weather data and COVID-19 pandemic predicting mortality rate: Machine learning approaches. *Chaos Solitons Fractals* **2020**, *138*, 110137. [CrossRef]
30. Malki, Z.; Atlam, E.; Dagnew, G.; Alzighaibi, A.R.; Ghada, E.; Gad, I. Bidirectional Residual LSTM-based Human Activity Recognition. *Comput. Inf. Sci.* **2020**, *13*, 40. [CrossRef]
31. Malki, Z.; Atlam, E.; Ewis, A.; Dagnew, G.; Reda, A.; Elmarhomy, G.; Elhosseini, M.A.; Hassanien, A.E.; Gad, I. ARIMA Models for Predicting the End of COVID-19 Pandemic and the Risk of a Second Rebound. *Neural Comput. Appl.* **2020**, *33*, 2929–2948. [CrossRef] [PubMed]

32. Saidi, R.; Jarray, F.; Mansour, M. A BERT Based Approach for Arabic POS Tagging. In Proceedings of the Advances in Computational Intelligence, 16th International Work-Conference on Artificial Neural Networks, Online, 16–18 June 2021; pp. 311–321. [CrossRef]
33. Alshalan, R.; Al-Khalifa, H. A Deep Learning Approach for Automatic Hate Speech Detection in the Saudi Twittersphere. *Appl. Sci.* **2020**, *10*, 8614. [CrossRef]
34. Kamath, U.; Graham, K.L.; Emara, W. Bidirectional Encoder Representations from Transformers (BERT). In *Transformers for Machine Learning*; Chapman and Hall/CRC: Boca Raton, FL, USA, 2022; pp. 43–70. [CrossRef]
35. Chouikhi, H.; Chniter, H.; Jarray, F. Arabic Sentiment Analysis Using BERT Model. In *Advances in Computational Collective Intelligence. ICCCI 2021*; Springer: Cham, Switzerland, 2021; pp. 621–632. [CrossRef]
36. Al-Twairesh, N. The Evolution of Language Models Applied to Emotion Analysis of Arabic Tweets. *Information* **2021**, *12*, 84. [CrossRef]
37. Attia, M.A. Arabic tokenization system. In Proceedings of the 2007 Workshop on Computational Approaches to Semitic Languages Common Issues and Resources, Prague, Czech Republic, 28–29 June 2007. [CrossRef]
38. Al-Hassan, A.; Al-Dossari, H. Detection of hate speech in Arabic tweets using deep learning. In *Multimedia Systems*; Springer Nature: Cham, Switzerland, 2021. [CrossRef]
39. Boulouard, Z.; Ouaissa, M.; Ouaissa, M.; Krichen, M.; Almutiq, M.; Gasmi, K. Detecting Hateful and Offensive Speech in Arabic Social Media Using Transfer Learning. *Appl. Sci.* **2022**, *12*, 12823. [CrossRef]
40. Anezi, F.Y.A. Arabic Hate Speech Detection Using Deep Recurrent Neural Networks. *Appl. Sci.* **2022**, *12*, 6010. [CrossRef]