Received: April 14, 2024. Revised: July 16, 2024.

INASS

International Journal of
Intelligent Engineering & Systems

http://www.inass.org/

Using Deep Learning Techniques to Detect Hate and Abusive Language in Arabic Tweets

Heba Al-Jarrah[1] Mohammad Al-Smadi[2] Mahmoud Hammad[1] Fatima Shannaq[3]*

[1]*Jordan University of Science and Technology, Jordan*
[2]*Qatar University, Qatar*
[3]*Amman Arab University, Jordan*
* Corresponding author's Email: f.alshannaq@aau.edu.jo

Abstract: Billions of people spend hours on social media platforms every day. While there are numerous known benefits of social media, hate speech and abusive language on social media platforms have become an increasingly serious social problem affecting individuals and societies' psychological state. Detecting and preventing hate speech and abusive language is a crucial task for healthy and safety digital communication. To overcome this important social problem, we propose four various deep learning and pre-trained models: (i) Ensemble deep learning model, (ii) Multilingual BERT model, (iii) Arabic BERT model, and (iv) ALBERT model to detect hate speech and abusive language in Arabic text. To that end, we utilized the L-HSAB Arabic dataset to build our models, and we utilized the OSACT dataset to evaluate the generalizability of our model's architecture. Our model tackles two classification tasks: a binary classification and a multiclass classification task. Our Arabic Bidirectional Encoder Representations from Transformers (BERT)-based model achieved the best performance on the binary classification task with an F1-score of 90.8%. While our Multilingual BET-based model obtained the best result on the multi-class classification task with an F1-score of 80.0%. Finally, the generalizability experiment of our best-performing mod-el on the binary classification task achieved an F1-score of 90.2% on the OSACT dataset.

Keywords: Arabic BERT, Multilingual BERT, Hate speech.

1. Introduction

All manuscripts must be in English. These guidelines include complete descriptions of the fonts, spacing, and related information for producing your proceedings manuscripts.

Recently, the Internet has become an integral part of our life, and everyone has at least one device connected to the Internet. With the advent of Web 3.0, people can have virtual interaction, personal communication, and a place to express their opinions. Social networking sites like X (formally known as Twitter) and Facebook facilitate sharing content and freely communicate ideas and views among users. Like any other means of communication, the online content can be clean and friendly or obscene and rude, and in some cases may contain hate speech. Hate speech is the use of abusive, offensive, or insulting language towards an individual or a group of people [1].

The United Nations and Human Rights identify the concept of hate speech as any speech, whether written or verbal, that expresses hatred and encourages violence against a person or group based on something like race, religion, or gender towards immigrants or minorities [2]. The motivation of this study is driven by the wide spread of hate speech and its social consequences on individuals as well as societies [3, 4]. History has shown that the consequences of hate speech may be dire; it may create an aggressive society that incites discrimination and inequality, such as the Christchurch Mosque shootings incident which occurred in March of 2019, motivated by white supremacy and Islamophobia

(https://en.wikipedia.org/wiki/Christchurch mosque shootings).

Hate speech detection received considerable attention from researchers in recent years. To the best of our knowledge, few studies detect hate speech and abusive language in the context of the Arabic language. Detecting hate speech in Arabic language is challenging as it contains a lot of morphological, synonymy, and dialects [5]. Thus, some words in certain countries may carry an ordinary meaning, while the same words in other countries may take the meaning of hate.

This study proposes an automatic technique to detect hate and abusive language posted on the X platform (Twitter) in Arabic language using four various deep learning and pre-trained models: (i) Ensemble deep learning model, (ii) Multilingual BERT model, (iii) Arabic BERT model, and (iv) ALBERT model. Moreover, several word embedding techniques were utilized to extract features and fed them into our proposed ensemble deep learning model: wiki-news- 300d-1M, crawl-300d-2M, and cc.ar.300.vec. We evaluated the proposed models using two benchmark datasets: (i) Levantine Hate Speech and Abusive Dataset (L-HSAB) [6], the main dataset used to train our models, and (ii) The Open-Source Arabic Corpora and Processing Tools (OSACT) [7], the secondary dataset used to examine the generalizability of the deep learning architecture of our best- performing model. We conducted two tasks on the primary dataset: (i) Binary classification task to classify tweets into abusive or normal. And (ii) multi-class classification task to distinguish normal, hate, or abusive tweets.

The remainder of this paper is organized as follows: Section 2 sheds light on related work. Section 3 describes the methodology proposed in this research. Our experimental evaluation results are described in Section 4 and discussed in Section 5. Finally, the paper concludes and draw future work in Section 6.

2. Related work

Many researchers have investigated the problem of hate speech and abusive language detection in literature. However, very few of them focused on the Arabic language. Following, we discuss their work classified into three categories based on the model used: (1) traditional machine learning (ML) models presented in Section 2.1, (2) deep learning models presented in Section 2.2, and (3) pre-trained models presented in Section 2.3.

2.1 Traditional machine learning classifiers

Since the spread of hate speech on social media platforms, many researchers used traditional ML methods to classify text and predict hate speech. For instance, [8] collected the dataset from Indonesian Instagram comments. They collected 1,053 comments as a training set and 34 comments were taken as a test set. They used Support Vector Machine (SVM) model to classify the comments into bullying/non-bullying. SVM obtained an accuracy of 79%. [9] participated in the SemEval 2019 task 5 subtask A for the English language and proposed a transfer learning technique from the Universal Sentences Encoder beside the machine learning algorithms such as SVM with RBF kernel for the classification. They also used the SMOTE technique to process imbalanced data. They achieved the first position in subtask A with a result of 65% on the test set. In addition, [10] introduced a method to build an Arabic dataset with no specific dialects or types of offensive. Their dataset is considered the largest Arabic offensive language until this moment, which includes 10,000 tweets. The dataset was collected from the X platform and annotated manually to four labels: offensive, vulgar, hate speech, or clean. For evaluation, they used SVM besides Mazajak embedding. The proposed model achieved a 79.7% F1-score. [11] proposed two models to tackle SemEval 2020 Task 12 subtask A for the Arabic language. The first model is an SVM model beside features selection models like TF-IDF on the word and character n-grams, the second model is the CHBA network. They used an external dataset called WideBot's offensive dataset to handle the imbalanced dataset problem. Their models achieved an F1-score of 86.91% and 88.72% for SVM and CHBA models, respectively.

While [12] introduced the first Arabic dataset for abusive language. The dataset was collected from the X platform and annotated manually into abusive or not abusive. They experimented with three different classical machine learning algorithms namely: SVM, Naïve Base (NB), and Decision Tree (J48), with 5 tweets and 50 features, 5 tweets and 100 features, 10 tweets and 50 features, 10 tweets and 100 features, 15 tweets and 50 features, and 15 tweets and 100 features. The result shows that the best performance was NB classifier with 10 tweets and 100 features with an F1-score of 90%. Similarity, [6] constructed a dataset for abusive and hate speech-language over Arabic text, labeled into three classes: hate, abusive, and normal tweets. The proposed dataset is known as (L-HSAB). They applied NB and SVM models with n-grams and TF representation.

Table 1. Machine learning approaches for detecting offensive, cyberbullying, abusive and hate speech language

Ref.	Dataset Language	Platform	Dataset size	Labels	Classifier	F1-score
[13]	Arabic	Three datasets: 1)X platform, 2)YouTube, 3)Facebook & X platform.	1100 8577 12,000	cyberbullying and non-cyberbullying	KNN Logistic Regression and Linear SVM used for voting.	71.1%, 75.8%, 98.3%
[10]	Arabic	X platform	10,000	offensive, vulgar, hate speech, or clean	SVM	79.7%
[11]	Arabic	X platform	10,000	offensive and non-offensive	SVM	86.9%
[9]	English	X platform	train: 9,000 validate:1,00 test: 3,000	Hate and not hate	SVM with RBF kernel	65.0%
[6]	Arabic	X platform	5,846	Binary: abusive or normal. Multi-class: hate, abusive, normal	NB	74.4%
[8]	Indonesian	Instagram	train: 1053 test: 34	Bullying and non-bullying	SVM	79.0%
[44]	Turkish	X platform and Instagram	900	cyberbullying and no cyberbullying	NB	84.0%
[12]	Arabic	X platform	1,3,00,000	abusive and not abusive	NB	90.0%

The result using NB and SVM was an accuracy of 74.4% and 66.8%, respectively. O¨zel, Sara¸c, Akdemir, and Aksu (2017) collected 900 messages from Instagram and the X platform in Turkish language. They applied different machine learning classifiers like SVM, Decision Tree, Naïve Bayes Multinomial (NBM), and K-Nearest Neighbors (KNN) classifiers. They proved that NBM was the most successful classifier with an F1-score of 84% when feature selection is applied. They also showed that all classifiers improved after feature selection except the decision tree classifier. [13] aimed to compare the performance of single and ensemble machine learning algorithms for detecting offensive language. They conducted experiments on three datasets, two of which were publicly available [14, 15], while the third was created to establish a balanced dataset for use in offensive language detection. Their results indicated that the ensemble machine-learning methodology (voting) outperformed the single-learner machine-learning classifiers, achieving F1-scores of 71.1%, 75.8%, and 98.3% for each of the three datasets used, respectively. Table 1 summarizes the research efforts using ML models with the focus on the Arabic datasets.

Despite the use of traditional machine learning methods to detect offensive language and hate speech in the mentioned studies, several challenges were encountered. Firstly, the diversity and variation in the size and quality of the datasets posed a significant challenge, as seen in the study [8] which relied on a small dataset of Indonesian Instagram comments. Secondly, dealing with imbalanced data was a common issue, necessitating techniques like SMOTE as used in study [9]. Thirdly, the identification of different dialects and languages presented another challenge, particularly in studies focusing on Arabic, such as study [10] which aimed to build the largest Arabic dataset without specific dialects. Fourthly, performance quality varied depending on the classification model and feature extraction techniques used, as in study [11] that utilized SVM and CHBA models. Fifthly, the impact of feature selection and the balance between single-learner models and ensemble methods was evident, with ensemble methods outperforming single models in study [13]. These challenges highlight the need for improved data and techniques to achieve better accuracy in detecting offensive language and hate speech.

2.2 Deep learning classifiers

This section presents the research efforts that utilized deep learning models to detect hate speech. Table 2 summarizes these research efforts. For instance, [16] proposed a stacked model based on BiGRU with a capsule network.

Table 2. Deep learning approaches for offensive, cyberbullying, abusive and hate speech detection

Ref	Dataset Language	Platform	Dataset size	Labels	Classifier	F1-score
[23]	Algerian	Facebook, YouTube and X platform.	N/A	Hate, cyberbullying, and offensive	Bi-GRU	75.8%
[24]	Arabic	X platform	1,1000	none, religious, racial, sexism, or general hate	CNN+LSTM	73.0%
[21]	Arabic	X platform	3,696	hate, normal	LSTM and CNN	71.68%
[20]	Arabic	X platform	8,000	offensive and non-offensive	CNN+LSTM	69.0%
[11]	Arabic		10,000 messages	offensive and non-offensive	CHBA Network	88.72%
[16]	English	X platform	train: 9,000, valid: 1,000 test: 3,000	Hate or not	FastText embedding with Bi-GRU	54.6%
[17]	English	X platform	train: 9,000 valid: 1,000 test: 3,000	hate or not	n-gram with BiLSTM	51.0%
[18]	English	X platform	train: 9,000 valid: 1,000 test: 3,000	hate or not	pre-trained emending (Glove embedding) with LSTM	51.9%
[6]	English	X platform	train: 9,000 valid: 1,000 test: 3,000	hate or not	n-gram with feed-forward neural network	50.0%
[19]	English	X platform	train: 9,000 valid: 1,000 test: 3,000	hate or not	Dense layer	42.0%
[22]	Arabic	X platform	6,000	Hateful and non-hateful	GRU-based RNN	77.0%

Whereas [17] experimented with two models: the first was by using the BiLSTM model with and without attention, depending on Glove embedding as feature extraction, while the second one used Logistic Regression with n-gram as word embedding. The result proved that BiLSTM without attention is the best result with an F1-score of 51%.

In addition, [18] proposed a transfer learning model, in which they used a pre-trained Glove embedding as a word embedding and then fed these weights to the LSTM, dense layer, and finally used sigmoid function in the last layer for making classification. The proposed model achieves an F1-macro score of 51.9% on the testing dataset. [19] used a single multilingual model, a dense layer, and tested on Spanish and English datasets. Briefly, Vista.ue team transformed tweets to a binary array of integer values, then fed them to the dense network. Task 5 of SemEval-2019 was used to evaluate the proposed approach. The proposed model achieved F1-score of 42.0% and 59.4% on the English and Spanish datasets, respectively. [20] investigated and explored the effectiveness of traditional machine learning and deep learning approaches in detecting offensive language in Arabic tweets. The machine learning models include SVM, Random Forest, XGBoost, Extra Trees, Decision Trees, Gradient Boosting, and Logistic Regression. Besides TF-IDF and word embedding as a feature. While in deep learning they investigated CNN, and the RNN model, which includes LSTM and GRU models. They concluded that the combination of CNN and LSTM layers with pre-trained word embedding achieved the best results with an F1-score of 69.0%.

Moreover, [21] introduced a dataset collected from the X platform on different topics in the Arabic language. They used a word2Vec and AraVec word embeddings and then fed the output of the

embeddings as input for stacked deep learning networks consists of LSTM and CNN algorithms. The proposed model achieved an accuracy of 66.564%. Similarly, [22] introduced the first hate speech dataset against religions and religious believes with 6,000 tweets. To annotate the dataset, they launched the CrowdFlower task for Arabic-speaking Middle Eastern people and annotated it as either hateful or not hateful. In addition, they created the first Arabic religious lexicon and used GRU based on the Recurrent Neural Network (RNN) model for evaluation and achieved an F1-score of 77.0%.

[23] proposed a new dataset in Algerian dialect consisting of 14,150 comments collected from different sources such as Facebook, YouTube, and the X platform. The dataset was annotated into three labels: Hate Speech, Cyberbullying, and Offensive. To evaluate the dataset, they applied different machine learning models, such as Random Forest, Naive Bayes, Linear Support Vector, Stochastic Gradient Descent (SGD), and Logistic Regression. They also used different deep learning models, including CNN, LSTM, GRU, Bi-LSTM, and Bi-GRU. They conducted experiments and found that Bi-GRU outperformed other models with an F1-Score of 75.8%.

[24] collected a new dataset from the X platform and labeled it with five hateful classes: none, religious, racial, sexism, or general hate. In their work, they took the SVM as a baseline model against four deep-learning models: LSTM, CNN +LSTM, GRU, and CNN+GRU. The results show that CNN +LSTM outperforms other models with an average F1-score of 73.0%.

Despite the advancements in deep learning for detecting hate speech, several challenges persist in these studies. First, the complexity and diversity of datasets present significant hurdles, as illustrated by [16, 17], who used various embedding techniques and architectures like BiGRU and BiLSTM, yet achieved moderate F1-scores of around 51%. Secondly, the need for extensive computational resources and the complexity of model architectures, such as those using transfer learning models and pre-trained embeddings like in [18], can be prohibitive. Thirdly, the performance inconsistency across different languages and contexts, as seen in [19], where the multilingual model performed variably on English and Spanish datasets, achieving F1-scores of 42.0% and 59.4%, respectively. Additionally, the studies highlight the challenge of achieving high accuracy and F1-scores, with models like CNN and LSTM combinations in [20, 21] showing results around 66.564% to 69.0%. Lastly, specific issues such as annotating datasets accurately, as in [22, 23], and handling different types of hate speech categories, as explored by [24], further complicate the model development and evaluation process. These challenges underscore the need for continued refinement of data, models, and techniques to improve the detection of offensive language and hate speech effectively.

2.3 Pre-trained models

Recently, pre-trained models have played an important role in Natural Language Processing (NLP) tasks and mainly in detecting hate speech and achieved good results comparing to the machine learning and deep learning classifiers. For in-stance, [25] proposed a new pre-trained transformer language model for ArabicBERT model. Their model consists of BERT base and CNN model. The output of embeddings from the last 4 hidden layers in the BERT model is fed as input into several CNN filters and convolution layers. They evaluated their model on Task 12 of SemEval-2020 dataset which included various languages including Arabic, Turkish, and Greek languages. The BERT-CNN model proves its overall performance with an F1-score of 89%, 84%, and 81% for Arabic, Greek, and Turkish, respectively. While [26] investigated the effect of fine-tuning strategies on the performance of the transfer learning approach based on the BERT base model. They evaluated their approach into two different datasets from [27, 28]. Their result proved that the BERT+CNN model outperforms other experiments included (BERT+LSTM, BERT+Nonlinear Layer) with F1-score of 88.0% and 92.0%, respectively for the two datasets.

[29] proposed a transfer learning model consisting of CNN and n-gram (CNN-gram) with the BERT model. They evaluated their model on their dataset which was collected from the X platform in a Roman language called Roman Urdu Hate-Speech and Offensive Language Detection (RUHSOLD). The result showed that their model achieved an F1-score of 90.0%.

[30] investigated the effect of transfer learning from pre-trained models and evaluated it on different Arabic datasets. They used four datasets, one of them was the L-HSAB dataset. However, their study is limited to only binary classifications (offensive or not offensive), so they converted the abusive and hate classes in L-HSAB dataset to offensive languages class. They conducted experiments with 2 models for monolingual Arabic (AraBERT Arabic- base-BERT), and 2 models for multilingual (BERT, XLM-RoBERTa) on all datasets. Their best-performing

model achieved an F1-score of 87.0 using the L-HSAB dataset.

[31] proposed an approach that combines static word embeddings with BERT-based language models. They evaluated their approach using multiple Arabic and English datasets. The L-HSAB dataset was one of the Arabic datasets used. The best proposed model applied on the L-HSAB dataset achieved an F1-score of 72.1% which was a combination of CNN + AraBERT with AraVec-Twitter embedding. They proved that combining word embedding with the BERT language model outperforms the BERT model alone.

[32] used [33] dataset, which is a dataset written in the standard Arabic language and with three different Arabic dialects: Gulf, Egyptian, and Iraqi, In addition, they translated this dataset from Arabic to English for comparative purposes. For evaluation, they used different BERT-based models and concluded that the classical BERT model on the English language achieved the best result with an F1-score of 98%, while AraBERT achieved an F1-score of 95% on the Arabic language.

Despite the promising results achieved by pre-trained models in detecting hate speech, several challenges persist in these studies. Firstly, the complexity and resource intensity of fine-tuning large pre-trained models like BERT and its variants require significant computational power, as highlighted by studies [25, 26]. Secondly, the performance of these models can vary across different languages and dialects, as seen in the varying F1-scores for Arabic, Greek, and Turkish in [25], and the different Arabic dialects in [32]. Thirdly, there are limitations in handling multi-class classifications effectively, as some studies like [30] had to simplify datasets to binary classifications, potentially losing nuanced distinctions between types of offensive language. Additionally, integrating static word embeddings with BERT-based models, as explored by [31], can be complex and requires careful optimization to outperform standalone BERT models. Lastly, translating datasets for comparative purposes, as done in [32], can introduce translation biases and affect the model's performance. These challenges emphasize the need for continuous refinement and optimization of pre-trained models and their fine-tuning strategies to enhance their effectiveness in detecting hate speech across diverse contexts.

While numerous research efforts focus on detecting hate speech in the Arabic language, there is currently no efficient model that demonstrates

Table 3. Pre-trained models for offensive, cyberbullying, abusive and hate speech detection

Ref	Dataset Language	Platform	Dataset size	Labels	Classifier	F1-score
[32]	Arabic and English	YouTube	15,050 YouTube comments	Offensive or inoffensive	BERT based models	Arabic: 95.0% English: 98.0%
[30]	Arabic	X platform	Four datasets. (1) 31,692 (2) 15,050 (3) 5,846 (4) 10,000	offensive and non-offensive	AraBERT	74.0% 87.0% 87.0% 91.0%
[25]	Arabic, Greek, and Turkish	X platform	10,000 tweets for Arabic, 10,288 for Greek and 35,284 tweets for Turkish	Offensive and non-offensive	BERT-CNN model	89.0%, 84.0% and 81.0%
[29]	Roman Urdu	X platform	10,000	Binary: Offensive and Normal Multi-class: Abusive/Offensive, Sexism, Religious Hate, and Profane	BERT+CNN -gram	90.0%
[31]	Arabic and English	X platform	Arabic datasets: (1) 1,800 (2) 10,126 (3) 5,846 (4) 4,000, English datasets (1) 4,618 (2) 14,100 (3) 564 (4) 13,000	_	static word embedding with (AraBERT + CNN)	72.1%
[26]	English	X platform	Two datasets: (1) 16,000 tweets and (2) 6,900 tweets	1- racism, sexism, or neither. 2- racism, sexism, neither, and both	BERT+CNN	88.0% and 92.0%

generalizability and consistently yields effective results across diverse Arabic datasets. Hence, the main contribution of this study is to introduce four state-of-the-art models for detecting hate speech in Arabic language. These models undergo training and evaluation using (L-HSAB), a benchmark of Arabic dataset designed for hate speech analysis. The research focuses on two classification tasks: a binary classification task, which distinguishes between normal or abusive Arabic text, and a multi-class classification task, which classifies Arabic text into one of three categories: normal, hate, or abusive. Moreover, the best-performing model in the binary classification task underwent evaluation using another Arabic dataset, the OSACT dataset, to assess the generalizability of our model's architecture.

3. Methodology

Fig. 1 depicts a high overview of our methodology. It describes the pipeline followed to build our models, starting from data pre-processing, feature extraction, then building the models. Finally, we have evaluated the performance of our models. This section describes the datasets used to train our models, the L-HSAB dataset, and the dataset used to measure the generalizability of our model, the OSACT dataset. Then, we describe the pre-processing techniques used to prepare the Arabic dataset. After that, we describe the feature extraction utilized on the dataset. Then, we describe the various proposed deep learning models. Finally, we discuss the performance metrics used to measure our models.

3.1 Dataset description

Two datasets have been utilized in this study. L-HSAB corpus, the main dataset for building the proposed models, as well as the OSACT corpus, which is used as a secondary dataset to examine the generalizability of our best achieved model.

3.1.1. L-HSAB dataset

L-HSAB is the first benchmark for Arabic Levantine Hate Speech and Abusive Language dataset introduced by [6], collected from the X platform via Twitter API and manually annotated by Levantine native speakers. L-HSAB is publicly available on GitHub (https://github.com/Hala-Mulki/L-HSAB-First-Arabic-Levantine-HateSpeech-Dataset) and presented as tabular data. It contains 5,846 tweets written in the Arabic Levantine dialect, the Arabic dialect widely spoken in Syrian and Lebanese. The dataset is labeled into three classes: hate, abusive, and normal. As shown in Table 4, the dataset is imbalanced as the number of normal instances is 3,650, abusive 1,728 instances, and hate 468 instances.

In this research, we designed models for two classification tasks: multi-class classification task, Task 1, and binary classification task, Task 2.

3.1.1.1. Task 1 (Multi-class classification)

Task 1 of this study is related to classifying Arabic tweets into one of three classes: Normal, Abusive, or Hate. We followed the baseline to split the dataset into train and test sets with a ratio of 80% for training data and 20% for testing data for each class. Based on that, the training data consists of 4,205 tweets and the testing data contains 1,641 tweets. Table 4 shows the classes and the number of tweets in each class for Task 1.

3.1.1.2. Task 2 (Binary classification)

Task 2 intends to classify Arabic tweets as Abusive or Normal. To make the data fits this task, we merged the Hate instances with the Abusive one as recommended by [6]. We used 80% of the data for training while the rest was used for testing. Table 5 shows the structure of the dataset for Task 2.

Table 4. Task 1 Dataset Structure

Class	Train	Test	Overall
Normal	2,626	1,024	3,650
Abusive	1,244	484	1,728
Hate	335	133	468
Overall	4,205	1,641	5,846

Table 5. Task 2 Dataset Structure

Class	Train	Test	Overall
Normal	2,626	1,024	3,650
Abusive	1,579	617	2,197
Overall	4,205	1,641	5,846

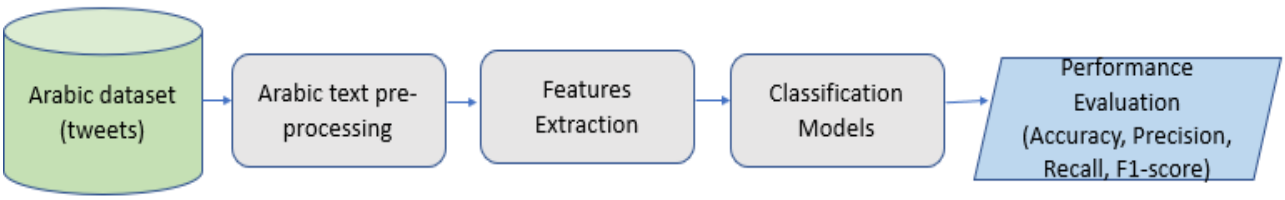

Figure. 1 High overview of our methodology for building hate and abusive language detection models for Arabic text

Table 6. OSACT Dataset Structure

Class	Train	Dev.	Test	Overall
OFF (not offensive)	1,410	179	402	1,991
NOT OFF (offensive)	5,590	821	1,598	8,009
Overall	7,000	1,000	2,000	10,000

3.1.2. OSACT dataset

The Open-Source Arabic Corpora and Processing Tools (OSACT) [7] is a publicly available corpus of Arabic tweets, and previously used in many competitions to detect offensive speech in the realm of Arabic text in social media. It contains 10,000 tweets, manually annotated for offensiveness as OFF, not offensive, or NOT OFF, offensive. The OSACT corpus used in this study to check if we can generalize our best-achieved models and pre-processing pipeline on unseen dataset. Table 6 illustrates the distribution of data in each file.

3.2 Data pre-processing

To prepare the dataset to train and evaluate our models, we have performed the following data pre-processing steps based on the characteristics of the Arabic language since the L-HSAB dataset is written in Arabic slang format:

- Remove Arabic stopping words.
- Remove URLs, special characters, punctuation marks, and numbers.
- Apply normalization: remove repeated characters and diacritics marks include (Fathah, Tanwin Fatha, Damma, Tanwin Damma, Kasra, and Tatwil) for example:
  The word: لاااا (*nooooo*) normalized into: لا (*no*)
  The word: الحياة (a common misspelling of the Arabic word *life*) normalized into: الحياه (the correct spelling of the Arabic word *life*)
  The word: عفواً (a common misspelling of the Arabic word *welcome*) normalized into: عفوا (the correct Arabic word of *welcome*)

As well as the normalization of Arabic short and long vowel characters such as changing the letter ي to ى, and آ, إ, ؤ, ئ to ا.

Moreover, we applied data augmentation using random oversampling technique: datasets in real world may suffer from a bias in their classes, which may affect the performance of the model. Data augmentation helps in reducing the overfitting and increasing the samples in training datasets with low costs; instead of collecting and labeling new samples, data augmentation improves the performance of the model and reduces its overfitting and enhances its model's generalizability [34]. It is worth mentioning that we used oversampling technique only on the binary classification task since our internal experiments, not reported in this research, showed no gain in the F1-score in the multi-class classification problem after oversampling.

3.3 Feature extraction

Word embedding is a widely used technique in many Natural Language Processing (NLP) tasks to extract features from text. We have used word embedding techniques for representing texts in a numeric form [35, 36]. It represents a word in an n-dimensional vector space where n is the number of dimensions. Compared to encoding a word as a unique integer, word embedding allows more information to be encoded into each word. Also, it assists in understanding the contextual, syntactic, and semantic information of each word and the implicit relationships between adjacent words. In this research, we used three different pre-trained word to vector techniques for Arabic language trained using the fastText algorithm. fastText algorithm computes embedding for character n-grams instead of word n-grams. This technique allows the model to capture words with similar meanings but different morphological word structures.

Our first proposed model is an ensemble deep learning model combined with three different word embedding techniques: wiki-news-300d [1], cc.ar.300.vec[2] and crawl-300d-2M.vec[3]. The wiki-news-300d-1M embedding contains one million word vectors trained on Wikipedia in 2017, UMBC web-based corpus, and statm.org news dataset, totaling 16 billion tokens [37]. The cc.ar.300.vec is trained on Common Crawl and Wikipedia, this model was trained using Continuous Bag of Words (CBOW) model with position-weights in dimension of 300 with character n-grams of length 5 and a window of size 5 and 10 negatives [38]. Finally, the crawl-300d- 2M.vec embedding contains 2 million word vectors trained on Common Crawl with 600 billions of tokens [37]. These embeddings encode the

[1] https://dl.fbaipublicfiles.com/fasttext/vectors-english/wiki-news-300d-1M.vec.zip

[2] https://dl.fbaipublicfiles.com/fasttext/vectors-crawl/cc.ar.300.vec.gz

[3] https://dl.fbaipublicfiles.com/fasttext/vectors-english/crawl-300d-2M.vec.zip

terms as 300-dimensional vectors and trained using fastText algorithm.

3.4 Proposed models

In this study, we build four distinct models to detect hate speech and abusive language: (1) an ensemble deep learning model, (2) Arabic-BERT model, (3) Multilingual BERT model, and (4) ALBERT model. Each of these models is discussed in detail in the following subsequent sections.

3.4.1. Ensemble deep learning model

This model consists of a CNN model concatenated with the Bi-LSTM model and a fully connected neural network to build an ensemble model to handle the hate and abusive detection problem. Next, we will discuss the framework regarding the proposed CNN-BiLSTM model.

After the pre-processing steps, the tokenization process was performed to tokenize the tweets before feeding them to our ensemble model.

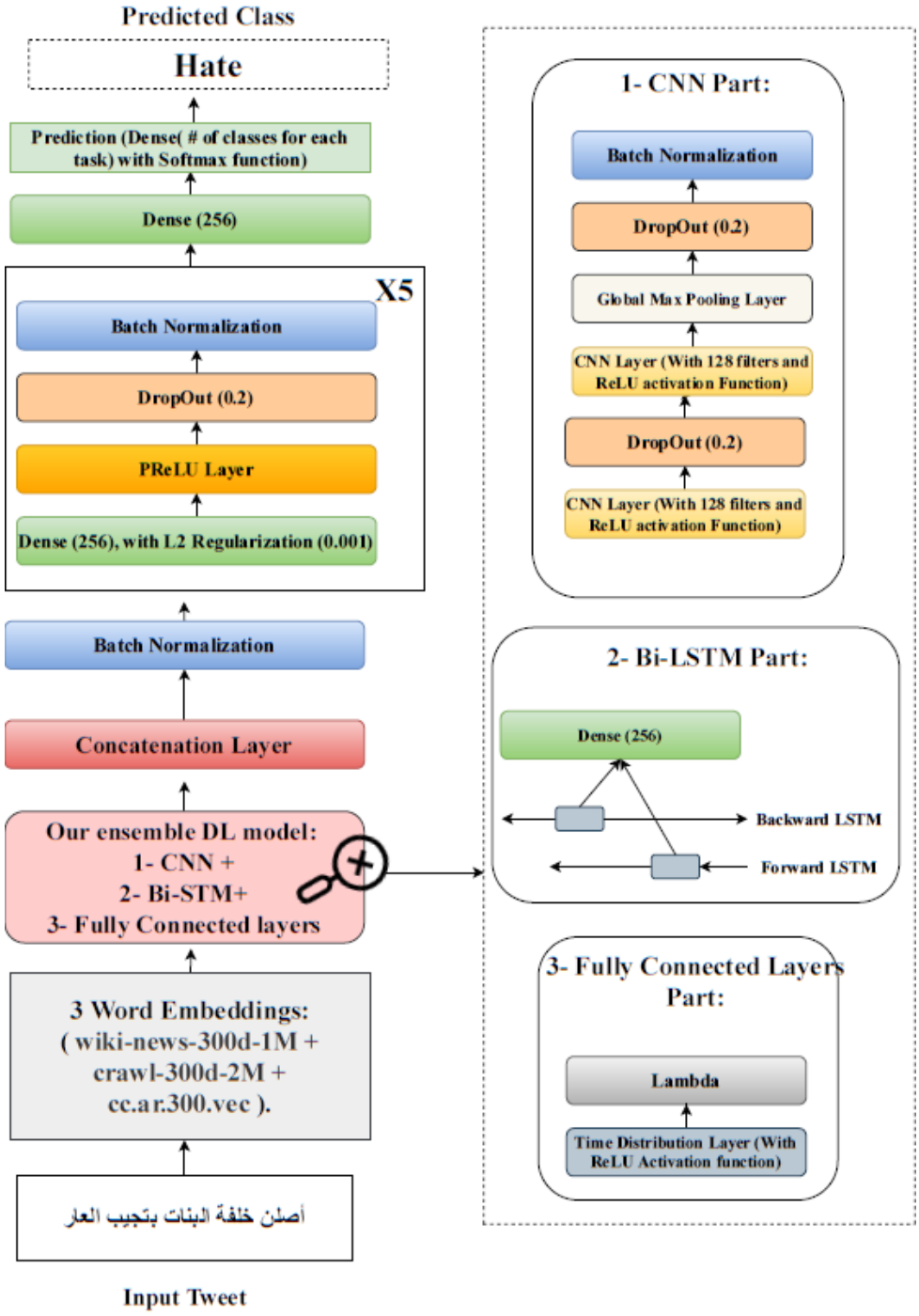

Figure. 2 The Architecture of our Proposed Model (ensemble DL model). The Arabic example in the given tweet means in English “giving birth to girls bring shame”

It was done on spaces and encoded as word-unique integers. A maximum sequence length of size 50 tokens (words) was included in the input to the DL model since the length of the maximum sequences (Tweets) in the dataset is 50 words. If a tweet has less than 50 tokens, it will be padded with 0's at the end of the tweet until the tweet contains 50 tokens.

In this model, three-word pre-trained embedding were used (wiki-news- 300d-1M, crawl-300d-2M and cc.ar.300.vec.). Each pre-trained embedding vector had a dimension of 300. Each pre-trained embedding has been fed into three different DL models. Fig. 2 illustrates the detailed architecture of our proposed ensemble model.

Next, the input tweet goes through 9 pre-trained word embedding layers, and each one of them goes to 3 different DL models. For example, the input tweet goes to the first embedding layer (wiki-news matrices) with 300-dimensions, then it is fed into three DL models; the first DL model is a fully connected layers consisting of time-distributed with dense layer of 256 units and *relu* as activation function and followed by lambda layer with output shape of 256. The second DL model starts with a new embedding layer which fed into the conv1d layer with filters equals to 128, filter length of 5, and *relu* activation function, followed by dropout layer with a dropout probability of 20% to minimize overfitting. Then another conv1d with the same parameters of the previous conv1d layer, followed by a 1-dimensional max-pooling layer, then dropout layer with a dropout probability of 20%, dense of 256 and Batch-Normalization layer. The third DL model works as follows; a new embedding layer goes through the Bidirectional LSTM model with a dropout probability of 20%. The same goes for the remaining two pre-trained word embedding layers (crawl-300d-2M and cc.ar.300.vec.).

Then, these three pre-trained words embeddings concatenated with three DL models as shown in Fig. 2 using concatenation layer. The idea behind using multiple embedding matrices was that more information would be given to the model for every word and capture the syntactic and semantic relations among huge words represented in the vector space.

Finally, the concatenation layer is fed into the Batch Normalization layer followed by the Dense layer of 256 units with L2 Regularization (0.001), followed by *PReLU* Layer, Dropout (0.2), and Batch Normalization layer repeated for five times, then to a Dense layer with 256 units. Then to a final dense layer using (the *Softmax* activation function with number of classes for each task) with loss functions: binary cross-entropy for binary task and categorical cross entropy for the multi-class classification task with the *adamax* optimizer.

Fig. 2 illustrates the architecture of our ensemble DL model (CNN- BiLSTM). The Batch normalization [39], dropout [40], and L2 regularization were applied to avoid one of the most known problems in deep learning models which involves predicting new instances based on the patterns presented to them, using instances of patterns observed and learned during the training process.

3.4.2. Pre-trained models

Given the success of the deep learning transformers in the NLP tasks, we fine-tuned various deep learning transformers. We fine-tuned various BERT-based transformer models for detecting hate and abusive language in Arabic text. The models are Arabic-BERT, Multilingual BERT, and ALBERT models.

3.4.2.1. Arabic-BERT model

We utilized the BERT model released by Google. As proposed in [25], the Arabic-BERT is a pre-trained BERT language model for the Arabic language. Arabic-BERT was released in four different sizes: Large, Base, Medium, and Mini. The difference between them is in the number of hidden layers, attention heads, hidden size, and the hyper parameters. In this study, we fine-tuned the asafaya/bert-large-arabic BERT model obtained from the Huggingface (https://huggingface.co). The BERT-large-Arabic model is one of the Arabic BERT models which is trained on nearly 95 GB of Arabic text from Open Super-large Crawled and Wikipedia sources. The training data includes dialectical words not restricted to Modern Standard Arabic, which is the closest to the dialect of the Arabic language used in the X platform. The proposed model architecture is shown in Fig. 3.

3.4.2.2. Multilingual-BERT model

As proposed in [41], the Multilingual BERT model has been pre-trained on the top 104 languages from the Wikipedia dataset. There are two versions of the Multilingual-BERT, an old version and a new version. In this study, we used the new version which was downloaded from the TensorFlow hub. The main differences between the old and the new version of the multilingual-BERT models are that the new version supports 104 languages rather than 102 languages as in the old version. Moreover, the new version fixes normalization issues in many languages. Therefore, it is highly recommended to use the new

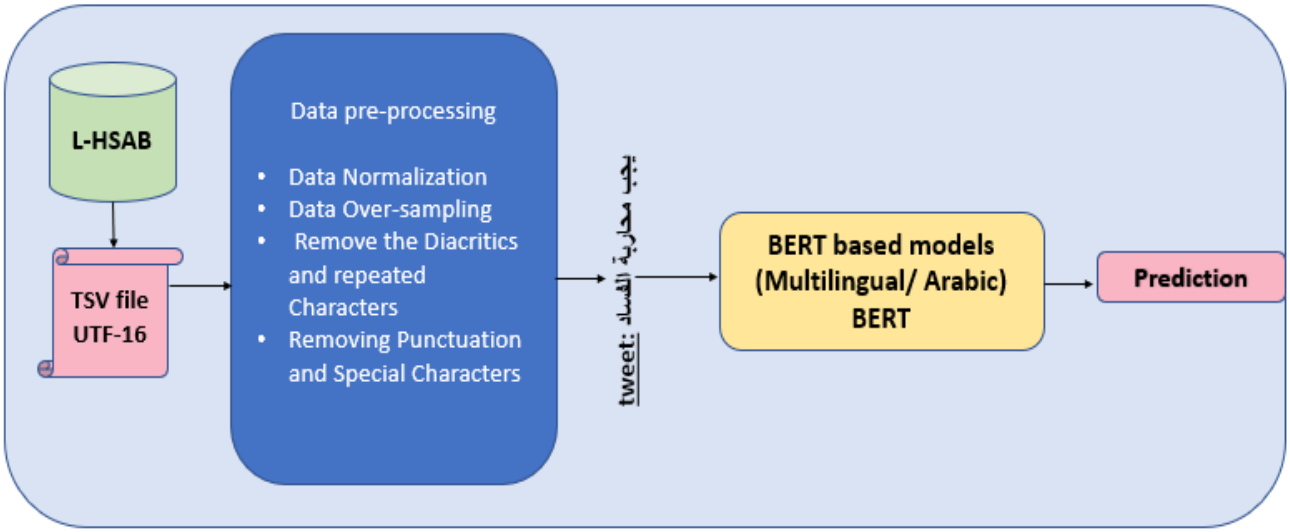

Figure. 3 The architecture of our proposed models which use pre-trained transformers, the Arabic-BERT and the Multilingual-BERT models. The example given in this figure means in English "Corruption should be flighted"

version for non-Latin alphabets languages like Arabic. Fig. 3 shows the architecture of the multilingual-BERT model. The hyper-parameters of the model have been optimized during the training phase with batch size of 32, learning rate of 2e-5, and 15 epochs.

3.4.2.3. ALBERT model

[42] proposed a lite version of the BERT model called the ALBERT model for supervised learning of language representation. ALBERT model is more lite in terms of the number of parameters and faster in terms of data throughput. The fine-tuned ALBERT model retrieved from Huggingface which has many versions. We used the kuisailab/ALBERT- base-Arabic for hate and abusive detection in Arabic tweets.

Table 7 compares between the pre-trained models used in terms of vocabulary size, embedding size, and numbers of parameters.

Table 7. Comparison of the utilized pre-trained models

	Arabic-BERT	Multilingual-BERT	ALBERT
Vocabulary size	3.4B	119K	32K
Embedding size	1024	768	128
No. of parameters	334M	110M	12M

3.5 Performance metrics

To evaluate our four proposed models, we followed two approaches: (1) performance evaluation of the proposed models using various evaluation metrics and (2) measuring the generalizability of the best performing model using the secondary dataset without training or modifying the hyper-parameters.

The performance of the proposed classifiers was evaluated by utilizing different evaluation metrics such as accuracy, precision, recall, and F1 score. Accuracy is the ratio of correctly classified instances over all the correct and the incorrect number of classified instances, calculated by Eq. (1):

$$Accuracy = \frac{TP+TN}{TP+TN+FP+FN} \quad (1)$$

Where TP is the true positive, TN is the true negative, FP is the false positive, and FN is the false negative.

Precision is the ratio of the texts that correctly identified from the positive class over the number of all non- positively classified texts, calculated by Eq. (2):

$$Precision = \frac{TP}{FP+TP} \quad (2)$$

While the Recall indicates how much the learning algorithm can identify the positively classified texts, calculated by Eq. (3):

$$Recall = \frac{TP}{FN+TP} \quad (3)$$

Table 8. The fine-tuned hyper-parameters for both tasks

(a) The fine-tuned hyperparameters for the multilingual BERT model for the multi-class classification task	
Hyperparameter	**Value**
# of epochs	15
Batch size	32
Patience	2
Learning Rates	2e-5
Optimizer	Adam
Max_Seq_Length	128

(b) The fine-tuned hyperparameters for the Arabic BERT model for the binary classification task	
Hyperparameter	**Value**
# of epochs	4
Batch size	8
Patience	2
Learning Rates	4e-6
Optimizer	Adam
auto_weights	True

F1 score or the F score is the weighted mean of precision and recall. It denotes the classifier ability in balancing between the precision and recall particularly in highly imbalanced dataset calculated as seen in Eq. (4):

$$F1 = \frac{2 \cdot precision \cdot recall}{precision + recall} \quad (4)$$

4. Experiments and results

The experiments in this research have conducted in two phases:

- Phase 1: experiments for building and evaluating the proposed models using the L-HSAB dataset.
- Phase 2: experiments for measuring the generalizability of the best-performing model using the secondary dataset, the OSACT dataset.

4.1 Experimental setup

We followed these setups in all our experiments. First, all our experiments have been conducted on a Google Collaboratory (colab) with 12GB of RAM and Hardware accelerator GPU. Second, we split the dataset into 80% training and 20% testing for each class. Third, fine-tuning the pre-trained language models on L-HSAB dataset. Tables 8a and 8b show the hyperparameters after tuning for the best-proposed models, the Multilingual BERT and Arabic BERT models for the multi-class classification task and the binary classification task, respectively. The experiments developed using the transformer library in Python named simple transformers, and the Python library Scikit-Learn for calculating the evaluation metrics.

Table 9. L-HSAB training data distribution before and after the oversampling

Class name	**Before oversampling**	**After oversampling**
Abusive	1,579	2,626
Normal	2,626	2,626

Finally, we applied data augmentation using random oversampling technique only on the binary classification task. Table 9 shows the distribution of the training set before and after oversampling on the binary dataset.

4.2 Building and evaluating the models using the L-HSAB dataset

The following two sections illustrate the process of building our proposed models and evaluating their performance using the L-HSAB dataset for both tasks, the binary classification, and the multi-class classification tasks. Then, comparing the results with the baseline model and other related works in the literature.

4.2.1. Task 1: Multi-class classification

We proposed our ensemble DL model, and the fine-tuned pre-trained language models to classify instances in L-HSAB dataset into Hate, Abusive, or Normal. Then we evaluated the models using the F1-score, precision, recall, and accuracy. Table 10 illustrates the evaluation results of our proposed models for the multi-class classification task. The results show that the multilingual BERT model achieves high results compared to the other proposed models and the baseline model which used the Naive Bayes classifier.

Table 10. Performance evaluation for our proposed models for multi-class classification task compared with the baseline model

Classifier	F1-score (%)	Precision (%)	Recall (%)	Accuracy (%)
Multilingual BERT	**80**	**81**	**79**	**81**
Ensemble DL model	78	80	77	80
ALBERT	74.8	76	74	82
Arabic BERT	74.7	75	74.5	83.67
[6]				
(Baseline and NB)	74.4	86.3	70.8	88.4

Table 11. Comparison of the best proposed model for the multi-class classification task with the other related models in literature

Classifier	F1-score (%)	Precision (%)	Recall (%)	Accuracy (%)
Our best model (Multilingual BERT)	**80**	**81**	**79**	**81**
[6]	74.4	86.3	70.8	88.4
[31]	72.1	-	-	-

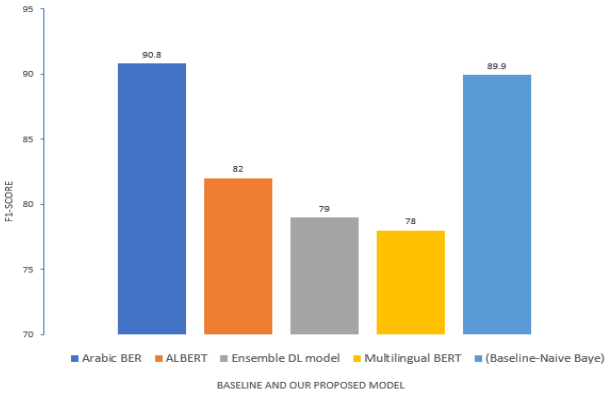

Figure. 4 Task 1 The F1-score of our four models compared with the baseline

Multilingual BERT model obtained 80.0% as F1-score, while the baseline model achieved 74.4%, as shown in Fig. 4.

Furthermore, Table 11 compares the results obtained from the best achieved model for Task 1 with compared with other research efforts presented in the literature. The table shows that our model unfolds a magnificent 5.6% improvement in F1-score compared to the other related work [6], and 7.9% over [31].

4.2.2. Task 2: Binary classification

In this experiment, we use our proposed models to classify tweets in L-HSAB dataset into Abusive or Normal. Table 12 illustrates the evaluation results of our proposed models for the binary classification task. The results show that our Arabic BERT model with oversampling achieves better than the other proposed models and the baseline model. The Arabic BERT model obtained 90.8%, 91.8%, 89.8%, and 92.7% as F1-score, precision, recall, and accuracy, respectively, while the baseline model achieved 89.9%, 90.5%, 89.0%, and 90.3% as F1-score, precision, recall, and accuracy, respectively. Fig. 5 compares the results of our proposed models and the baseline model.

Table 12. Performance evaluation of our proposed models for the binary classification task compared with the baseline model

Classifier	F1-score (%)	Precision (%)	Recall (%)	Accuracy (%)
Arabic BERT	**90.8**	**91.8**	**89.8**	**90.8**
ALBERT	82	82.8	82.48	84
Ensemble DL model	79	81	80	80
Multilingual BERT	78	80.3	77.9	84.5
[6] (Baseline-Naive Bayes)	89.9	90.5	89	90.3

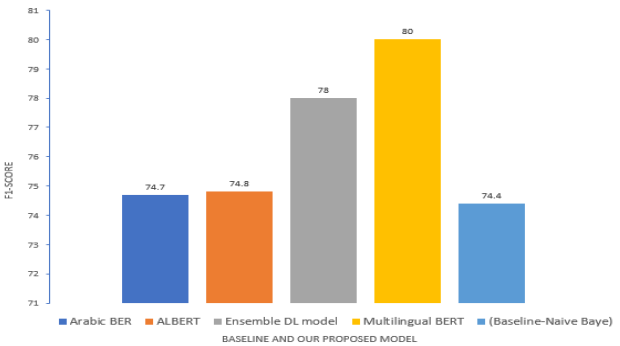

Figure. 5 Task 2 comparison results

Table 13. The performance of our proposed model compared with the other related works for the binary classification task

Classifier	F1-score (%)	Precision (%)	Recall (%)	Accuracy (%)
Our best model (Arabic BERT)	**90.8**	**91.8**	**89.8**	**90.3**
[6]	89.9	90.5	89	90.3
[30]	87	87	87	88

Furthermore, Table 13 compares the results obtained from the best achieved model for Task 2 with the other related works presented in the literature. The table shows that our model obtains better results than the best performing models in the literature. F1-score boosted by 0.9% compared with the work of [6], and 3.8% compared with [30].

4.3 Generalizability experiment

Since the whole idea of building a ML model is to be used on unseen instances, we have conducted another experiment to measure the generalizability of our best-performed binary model, the Arabic BERT, on a different corpus. Table 14 shows the evaluation result of our model on the secondary dataset, the OSACT dataset. The results show that the Arabic BERT model achieves outstanding performance with 90.2%, 90.9%, 89.5%, and 93.8% for F1-score, precision, recall, and accuracy, respectively. Our results are close to the results of the best approach for Subtask A in the OSACT shared task competition, which was submitted by [43], and achieved 90.5% F1-score.

Table 14. The performance of our best-performing binary model on the OSACT dataset

Classifier	F1-score (%)	Precision (%)	Recall (%)	Accuracy (%)
Our binary model (Arabic BERT)	90.2	90.9	89.5	93.8
1st place on OSACT task	90.5	90.1	90.8	93.8

These experimental results proof the generalizability of our best-performing binary model, the Arabic BERT model, although our model is not trained on that dataset.

5. Discussion

As shown in Section 4, the Multilingual BERT was the best model for the multi-class classification task, and the Arabic BERT model achieved the highest score in the binary classification task. Both models outperformed the baseline model which evaluated on the L-HSAB dataset with an enhancement of 5.6% on the first task and 11.8% on the second task, as shown in Figs. 4 and 5.

The distinction results of both pre-trained models, the Multilingual BERT, and the Arabic BERT, is credited to three main factors: (1) essence of pre-trained data, (2) vocabulary size of the pre-trained data, and (3) the dimensionality of our word embedding techniques.

The essence of pre-trained data is an important reason for boosting the performance of our models. The Arabic BERT model was pre-trained on dialectical words, the common dialect on the X platform, and not restricted to Modern Standard Arabic (MSA). While the Multilingual BERT was pre-trained on the top 104 languages including the huge Wikipedia dataset, and the fact that it is highly recommended for languages with non-Latin alphabets like the Arabic language.

However, the vocabulary size of the pre-trained data has a superb impact on the model performance, in which the larger the data size, the more diversity of pre-training distribution. For instance, the Arabic BERT model has a vocabulary size of 3.4 billion words, and the Multilingual BERT trained on a vocabulary size of 119,547 words, whereas the vocabulary size of the ALBERT model is only 32,000 words.

Furthermore, the dimensionality of the word embedding affects the model performance in which larger dimension captures more syntactic and semantic relations among huge words that are represented in the vector space (recall Table 7).

Tables 11 and 13 show that our proposed models outperformed all the previous works for both tasks with an enhancement of 0.9% and 5.6% over the best performing models, i.e. the model proposed by [6].

Although [30] used a pre-trained language model (AraBERT), the reason for the variance in results is because they did not apply pre-processing to the text in addition to differences in fine-tuning the hyper-parameters such as learning rate, batch size, and number of epochs.

Deep learning models are typically implemented around a specific dataset whereas the pre-processing pipeline and the fine-tuned hyper-parameters are chosen after careful analysis of the dataset. Consequently, there is no guarantee that the same pipeline will perform satisfactorily over another dataset. This could be tested by taking the entire deep learning model that has been implemented on one dataset and then running the same pipeline and the same benchmarks on a second dataset. To that end, we have tested our best- performing model on a secondary dataset, the OSACT dataset, and ran the same pipeline to see how well the model generalizes on a secondary dataset. OSACT dataset is binary data that aims to classify the tweets into offensive or non-offensive. The results were impressive, in which the Arabic BERT model with the best results on binary task for the L-HSAB dataset achieved also superior results on the OSACT dataset as summarized in Table 14.

6. Conclusion and future work

Hate speech and abusive language detection is a challenging task, especially for the Arabic language. In this study, we proposed four deep learning models: (1) ensemble model of different neural networks, which are a combination of fully connected layers, a convolution neural network (CNN), and bidirectional long short-term memory (Bi LSTM) with three different word embedding: wiki-news-300d-1M, crawl-300d-2M, and cc.ar.300.vec, (2) multilingual BERT language model, (3) the Arabic-BERT language model, and (4) the ALBERT model that is retrieved from a simple transformer library. The proposed models are trained and evaluated on the L-HSAB dataset. Two classification tasks were targeted in this study: binary classification task, which aims to classify tweets into normal or abusive, and multi-class classification task that categorizes tweets into one of three classes: normal, hate, or abusive.

The results showed that fine-tuning the multilingual and Arabic BERT- based language models outperformed the other models and achieved better performance with F1-score of 80.0% and 90.8% on task 1 and task 2, respectively. Moreover, the best model on the binary task has been evaluated on a secondary dataset, the OSACT dataset, to evaluate its generalizability. The results were outstanding where the Arabic BERT model achieved an F1-score macro average of 90.2%. Our study showed that pre-trained language models are more effective in predicting hate speech in the Arabic context than pre-trained word embedding.

As a future work, we plan to expand the study to include classifying the type of hate, like hate based on race, color, national origin, disability, religion, or orientation. Moreover, we are planning to use Large Language Models LLMs in the future to evaluate their performance in detecting hate speech.

Moreover, diacritics in Arabic are challenging, and some Arabic words may show hate in some contexts while sounding normal in other contexts because of changing their diacritics, such a consequence may cause ambiguity in sentiment analysis workbooks. For instance, the word (نور) has two meanings depending on its diacritics: (نَوَر) is a racial slur used to derogate someone's heritage or origin, and it means (gypsies) in English. Whereas (نوْر) means (light) in English. This example illustrates the complexity of the classification of Arabic hate speech, and there is no built-in library, or pre-trained data to support the diacritics issues. We also plan to overcome such problems in the Arabic language in the future.

Conflicts of Interest

The authors declare no conflict of interest.

Author Contributions

All authors contributed to the paper as follows: forming the research idea: H. Aljarrah, M. Hammad and M. Al-Smadi; study conception and design: H. Aljarrah, M. Hammad, M. Al-Smadi, F. Shannaq; data collection: H. Aljarrah and F. Shannaq; conducting the experiments: H. Aljarrah and F. Shannaq; data analysis and interpretation of the results: M. Hammad and F. Shannaq; draft manuscript preparation: M. Hammad and F. Shannaq and H. Aljarrah; revising manuscript: M. AlSmadi, M. Hammad and F. Shannaq. All authors reviewed the results and approved the final version of the manuscript.

References

[1] C. Nobata, J. Tetreault, A. Thomas, Y. Mehdad, and Y. Chang, "Abusive language detection in online user content", In: *Proc. of the 25th international conference on world wide web*, pp. 145-153, 2016.

[2] A. Guterres et al., "United nations strategy and plan of action on hate speech" [Online Article]: https://www.un.org/en/hate-speech/understanding-hate-speech/what-is-hate-speech

[3] P. Fortuna and S. Nunes, "A survey on automatic detection of hate speech in text",

ACM Computing Surveys (CSUR), Vol. 51, No. 4, pp. 1-30, 2018.

[4] K. Saha, E. Chandrasekharan, and M. De Choudhury, "Prevalence and psychological effects of hateful speech in online college communities", In: *Proc. of the 10th ACM Conference on Web Science*, pp. 255-264, 2019.

[5] A. Farghaly and K. Shaalan, "Arabic natural language processing: Challenges and solutions", *ACM Transactions on Asian Language Information Processing (TALIP)*, Vol. 8, No. 4, pp. 1-22, 2009.

[6] H. Mulki, H. Haddad, C. B. Ali, and H. Alshabani, "L-hsab: a levantine twitter dataset for hate speech and abusive language", In: *Proc. of the Third Workshop on Abusive Language Online*, pp. 111-118, 2019.

[7] F. Husain, "Osact4 shared task on offensive language detection: Intensive preprocessingbased approach", *arXiv preprint arXiv:2005.07297*, 2020.

[8] M. Andriansyah, A. Akbar, A. Ahwan, N. A. Gilani, A. R. Nugraha, R. N. Sari, and R. Senjaya, "Cyberbullying comment classification on indonesian selebgram using support vector machine method", In: *Proc. of 2017 Second International Conference on Informatics and Computing (ICIC)*, IEEE, pp. 1-5, 2017.

[9] V. Indurthi, B. Syed, M. Shrivastava, N. Chakravartula, M. Gupta, and V. Varma, "Fermi at semeval-2019 task 5: Using sentence embeddings to identify hate speech against immigrants and women in twitter", In: *Proc. of the 13th International Workshop on Semantic Evaluation*, pp. 70-74, 2019.

[10] H. Mubarak, A. Rashed, K. Darwish, Y. Samih, and A. Abdelali, "Arabic offensive language on twitter: Analysis and experiments", *arXiv preprint arXiv:2004.02192*, 2020.

[11] Y. Otiefy, A. Abdelmalek, and I. E. Hosary, "Woli at semeval-2020 task 12: Arabic offensive language identification on different twitter datasets", *arXiv preprint arXiv:2009.05456*, 2020.

[12] E. A. Abozinadah, A. V. Mbaziira, and J. Jones, "Detection of abusive accounts with arabic tweets", *Int. J. Knowl. Eng.-IACSIT*, Vol. 1, No. 2, pp. 113-119, 2015.

[13] M. Khairy, T. M. Mahmoud, A. Omar, and T. Abd El-Hafeez, "Comparative perfor- mance of ensemble machine learning for arabic cyberbullying and offensive language detection", *Language Resources and Evaluation*, Vol. 58, No. 2, pp. 695-712, 2024.

[14] K. S. Alam, S. Bhowmik, and P. R. K. Prosun, "Cyberbullying detection: an ensemble based machine learning approach", In: *Proc. of 2021 third international conference on intelligent communication technologies and virtual mobile networks (ICICV)*, pp. 710-715, 2021.

[15] S. Nadali, M. A. A. Murad, N. M. Sharef, A. Mustapha, and S. Shojaee, "A review of cyberbullying detection: An overview", In: *Proc. of 2013 13th International Conference on Intellient Systems Design and Applications*, pp. 325-330, 2013.

[16] Y. Ding, X. Zhou, and X. Zhang, "Ynu_dyx at semeval-2019 task 5: A stacked bigru model based on capsule network in detection of hate", In: *Proc. of the 13th International Workshop on Semantic Evaluation*, pp. 535-539, 2019.

[17] A. Baruah, F. Barbhuiya, and K. Dey, "Abaruah at semeval-2019 task 5: Bi-directional lstm for hate speech detection", In: *Proc. of the 13th International Workshop on Semantic Evaluation*, pp. 371-376, 2019.

[18] A. Montejo-Ráez, S. M. Jiménez-Zafra, M. A. García-Cumbreras, and M. C. Díaz- Galiano, "Sinai-dl at semeval-2019 task 5: Recurrent networks and data augmentation by paraphrasing", In: *Proc. of the 13th International Workshop on Semantic Evaluation*, pp. 480-483, 2019.

[19] K. Raiyani, T. Gonçalves, P. Quaresma, and V. Nogueira, "Vista. ue at semeval-2019 task 5: single multilingual hate speech detection model", In: *Proc. of the 13th International Workshop on Semantic Evaluation*, pp. 524-528, 2019.

[20] A. Abuzayed and T. Elsayed, "Quick and simple approach for detecting hate speech in arabic tweets", In: *Proc. of the 4th Workshop on Open-Source Arabic Corpora and Processing Tools, with a Shared Task on Offensive Language Detection*, pp. 109-114, 2020.

[21] H. Faris, I. Aljarah, M. Habib, and P. A. Castillo, "Hate speech detection using word embedding and deep learning in the arabic language context", *ICPRAM*, pp. 453- 460, 2020.

[22] N. Albadi, M. Kurdi, and S. Mishra, "Are they our brothers? analysis and detection of religious hate speech in the arabic twittersphere", In: *Proc. of 2018 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining (ASONAM)*, pp. 69-76, 2018.

[23] A. C. Mazari and H. Kheddar, "Deep learning-based analysis of algerian dialect dataset targeted hate speech, offensive language and

cyberbullying", *International Journal of Computing and Digital Systems*, 2023.

[24] A. Al-Hassan and H. Al-Dossari, "Detection of hate speech in arabic tweets using deep learning", *Multimedia systems*, Vol. 28, No. 6, pp. 1963-1974, 2022.

[25] A. Safaya, M. Abdullatif, and D. Yuret, "Kuisail at semeval-2020 task 12: Bert-cnn for offensive speech identification in social media", In: *Proc. of the Fourteenth Workshop on Semantic Evaluation*, pp. 2054-2059, 2020.

[26] M. Mozafari, R. Farahbakhsh, and N. Crespi, "A bert-based transfer learning approach for hate speech detection in online social media", In: *Proc. of International Conference on Complex Networks and Their Applications*, pp. 928-940, 2019.

[27] Z. Waseem and D. Hovy, "Hateful symbols or hateful people? predictive features for hate speech detection on twitter", In: *Proc. of the NAACL student research workshop*, pp. 88-93, 2016.

[28] T. Davidson, D. Warmsley, M. Macy, and I. Weber, "Automated hate speech detection and the problem of offensive language", In: *Proc. of the International AAAI Conference on Web and Social Media*, Vol. 11, 2017.

[29] H. Rizwan, M. H. Shakeel, and A. Karim, "Hate-speech and offensive language detection in roman urdu", In: *Proc. of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pp. 2512-2522, 2020.

[30] F. Husain and O. Uzuner, "Transfer learning approach for arabic offensive language detection system-bert-based model", *arXiv preprint arXiv:2102.05708*, 2021.

[31] I. Alghanmi, L. Espinosa-Anke, and S. Schockaert, "Combining bert with static word embeddings for categorizing social media", In: *Proc. of the sixth workshop on noisy user-generated text*, pp. 28-33, 2020.

[32] Z. Boulouard, M. Ouaissa, M. Ouaissa, M. Krichen, M. Almutiq, and K. Gasmi, "Detecting hateful and offensive speech in arabic social media using transfer learning", *Applied Sciences*, Vol. 12, No. 24, p. 12823, 2022.

[33] A. Alakrot, L. Murray, and N. S. Nikolov, "Dataset construction for the detection of anti-social behaviour in online communication in arabic", *Procedia Computer Science*, Vol. 142, pp. 174-181, 2018.

[34] V. Iosifidis and E. Ntoutsi, "Dealing with bias via data augmentation in supervised learn- ing scenarios", *Jo Bates Paul D. Clough Robert Jäschke*, Vol. 24, No. 11, 2018.

[35] M. Li, Q. Lu, Y. Long, and L. Gui, "Inferring affective meanings of words from word embedding", *IEEE Transactions on Affective Computing*, Vol. 8, No. 4, pp. 443-456, 2017.

[36] M. F. Grawe, C. A. Martins, and A. G. Bonfante, "Automated patent classification using word embedding", In: *Proc. of 2017 16th IEEE International Conference on Machine Learning and Applications (ICMLA)*, pp. 408-411, 2017.

[37] T. Mikolov, E. Grave, P. Bojanowski, C. Puhrsch, and A. Joulin, "Advances in pre-training distributed word representations", *arXiv preprint arXiv:1712.09405*, 2017.

[38] E. Grave, P. Bojanowski, P. Gupta, A. Joulin, and T. Mikolov, "Learning word vectors for 157 languages", In: *Proc. of the International Conference on Language Resources and Evaluation (LREC 2018)*, 2018.

[39] S. Ioffe and C. Szegedy, "Batch normalization: Accelerating deep network training by reducing internal covariate shift", In: *Proc. of International conference on machine learning*, pp. 448-456, PMLR, 2015.

[40] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, "Dropout: a simple way to prevent neural networks from overfitting", *The journal of machine learning research*, Vol. 15, No. 1, pp. 1929-1958, 2014.

[41] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "Bert: Pre-training of deep bidirectional transformers for language understanding", *arXiv preprint arXiv:1810.04805*, 2018.

[42] Z. Lan, M. Chen, S. Goodman, K. Gimpel, P. Sharma, and R. Soricut, "Albert: A lite bert for self-supervised learning of language representations", *arXiv preprint arXiv:1909.11942*, 2019.

[43] S. Hassan, Y. Samih, H. Mubarak, A. Abdelali, A. Rashed, and S. A. Chowdhury, "Alt submission for osact shared task on offensive language detection", In: *Proc. of the 4th Workshop on Open-Source Arabic Corpora and Processing Tools, with a Shared Task on Offensive Language Detection*, pp. 61-65, 2020.

[44] S. A. Özel, E. Saraç, S. Akdemir, and H. Aksu, "Detection of cyberbullying on social media messages in turkish", In: *Proc. of 2017 International Conference on Computer Science and Engineering (UBMK)*, pp. 366-370, 2017.