A Novel Machine Learning-based Framework for Detecting Religious Arabic Hatred Speech in Social Networks

Mahmoud Masadeh[1], Hanumanthappa Jayappa Davanager[2] and Abdullah Y. Muaad[3]
Computer Engineering Department, Yarmouk University, Irbid 21163, Jordan[1]
Department of Studies in Computer Science, Mysore University, Manasagangothri
Mysore 570006, India[2,3]
Sana'a Community College, Sana'a 5695, Yemen[3]
Corresponding Author[3]

***Abstract*—Social media platforms generate a huge amount of data every day. However, liberty of speech through these networks could easily help in spreading hatred. Hate speech is a severe concern endangering the cohesion and structure of civil societies. With the increase in hate and sarcasm among the people who contact others over the internet in this era, there is a dire need for utilizing artificial intelligence (AI) technology innovation that would face this problem. The rampant spread of hate can dangerously break society and severely damage marginalized people or groups. Thus, the identification of hate speech is essential and becoming more challenging, where the recognition of hate speech on time is crucial in stopping its dissemination. The capacity of the Arabic morphology and the scarcity of resources for the Arabic language makes the task of distinguishing hate speech even more demanding. For fast identification of Arabic hate speech in social network comments, this work presents a comprehensive framework with eight machine learning (ML) and deep learning (DL) algorithms, namely Gradient Boosting (GB), K-Nearest Neighbor (K-NN), Logistic Regression (LR), Naive Bayes (NB), Passive Aggressive Classifier (PAC), Support Vector Machine (SVM), Ara-BERT, and BERT-AJGT are implemented. Two representation techniques have been used in the proposed framework in order to extract features: a bag of words followed by BERT-based context text representations. Based on the result and discussion part, context text representation techniques with Ara-BERT and BERT-AJGT outperform all other ML models and related work with accuracy equal to 79% for both models.**

***Keywords*—Machine learning; Arabic language; hatred detection; social network; classification algorithm**

I. Introduction

Low-resource languages, e.g., Arabic, Hindi, and Urdu, do not have a considerable amount of data for training and building conversational Artificial Intelligence (AI) systems. The Arabic language is the authorized language for 22 Arabic countries with roughly more than 422 million aboriginal speakers [1]. Additionally, it is a religious language spoken by more than 1.5 billion Muslims. There are three main types of it: *i*) Classical Arabic which is the language of the Holy Quran, *ii*) Modern Standard Arabic (MSA) which is used by academia, and *iii*) Dialectical Arabic which differs between regions since it is used for daily life networking [2]. Thus, the Arabic language, with a large number of speakers worldwide, is challenging task when we work with AI systems. Moreover, the Arabic language is identified as the 4^{th} in the usage on the internet [3]. However, the sophistication of the Arabic language makes the automatic identification of Arabic hate speech a complex task. Dialectal Arabic doesn't have formal grammar or spelling regulations. Moreover, spelled words can have different importance based on various dialects, which augments the vagueness of the language [4].

Sociable applications, e.g., Facebook, YouTube, and Twitter, are generating an extensive quantity of data which is considered a valuable goldmine for researchers. Social media-generated data helps in recognizing unlawful behavior, restricting potential hurt, and maintaining residents safe [2]. Some users utilize the wild adoption of online social networks to spread radical and biased statements that diffuse hate speech. *Sentiment analysis* (SA), i.e., opinion mining, analyses individuals' thoughts, attitudes, emotions, and opinions towards an entity, e.g., person, object, or service. The SA is implemented at various levels of granularity [5]: *i*) *document-level*: each text fragment is considered as a component with an opinion towards a single object. It aims to categorize a review as positive, negative, or neutral, *ii*) *sentence-level*: aims to extract opinions for a smaller text which is more challenging than SA for a document, and *iii*) *aspect-level*: determines the major features of a belief while focusing on aspect extracting and feeling categorization of aspects. The approaches of semantic analysis could be supervised, unsupervised, or hybrid. For *unsupervised* methods, numerous sentiment phrases are required to reveal the semantic orientation of texts. Thus, lexicon-based approaches are used. However, supervised techniques rely mostly on utilizing data mining tools to build a learning algorithm on a collection of tagged information. A prime application of SA is hate speech detection through online social network [6].

Hate speech is any class of inappropriate language, e.g., insults, slurs, threats, encouraging violence, and impolite language, that targets individuals or groups based on typical attributes such as nationality, religion, ideology, disability, social class, or gender. Hate speech includes racism, misogyny, religious discrimination, and abusive speech. *Racism* implies hate speech that attacks people based on their skin color, race, origin, class, or nationality [7]. *Misogyny* is the hate speech that targets females, i.e., women or girls [8]. Religious bias is hatred vocabulary towards somebody based on their beliefs, faiths, practices, or even the deficiency of religious faiths.

The *abusive speech* represents disrespectful, rude, or criticizing speech to hurt or deliver harmful sentiments.

Hate speech (HS) detection is a branch of offensive language detection. There is an increasing studies for abusive/HS detection for English language. However, it is still very limited for Arabic dialects due to the scarcity of the publicly obtainable resources required for abusive/HS detection in Arabic social media texts. The authors of [9] declared that the harmful online content on social media can be grouped into various categories including: Vicious, Vulgar, Offensive, Violent, Adult content, Terrorism and Spiritual hate speech.

This work targets *religious hate speech* (RHS) that could be insulting, abusive, or hateful. RHS aims to instigate hate, intolerance, or roughness toward people because of their religious faiths. The recent immense usage of social networks mandates applying different *text processing* on such cyberspace. The remarkable amount of generated data requires applying new monitoring tasks such as cyberbullying recognition [10], hate speech detection [6], irony identification [11], and discovery of offensive language [12]. Accordingly, battling hate speech mandates generating and elucidating a considerable amount of data for automatic hatred speech identification by building artificial intelligence-based models, i.e., ML and DL [13].

Lately, detecting abusive and hateful speech has gained increasing attraction from investigators in NLP and computational social sciences societies. Thus, detecting abusive speech and hate speech is essential for online safety. lately, various studies indicated that the existence of hate speech may be related to hate crimes [8]. Therefore, this work aims to enhance the detection of offensive language and hate speech on Arabic text. Detecting religious hate speech in any language, including Arabic, has different challenges, including : 1) the gigantic volume of the data generated over social networks makes it difficult to locate typical patterns and trends in the data, 2) noise may exist in the data, e.g., inaccurate grammar, misspelled phrases, Internet slang, abbreviations, lengthening of words, and multi-lingual scripts, 3) the comments being written in poorly text, and including paralinguistic signs, e.g., emoticons, and hashtags. Moreover, hate detection is a context-dependent task, and it is still missing a consense of what is forming hate speech due to the different cultures, customs and traditions, and 4) since the social networks prevent posting illegal content, users post information that looks authentic and simple but quietly causes a hate speech. Thus, building a tool for the automatic detection of hate speech would be complex [6].

Social platforms, e.g., Twitter and Facebook began battling online hate speech by explaining procedures that limit the use of violent and dehumanizing languages [14]. Moreover, various Arabic countries, where their users of social media sites are adding Arabic content, modified their laws to combat cybercrimes including hate speech. For example, Jordan added a new cybercrime laws [15] that defines hate speech as any action, writing, or speech planned to cause and raise ethical conflict or call for violence and provocation to fighting between the diverse segments of the nation. Regarding the Arabic language, there is a clear shortage in the conducted research for hate speech on online social networks. Thus, artificial intelligence, data mining, and machine learning techniques could be utilized to efficiently perform more research and experiments on hate speech detection which constitutes a fertile resource for investigation. This work aims to design a prototype for the automatic identification of abusive and hate speech using various ML and DL techniques with a standard data set.

The remaining sections are organized as follows. Section II presents preliminaries necessary to understand the context of the work. Section III highlights some of the important related work. The various aspects of the proposed methodology are explained in Section IV. Section V introduces the experimental setup and analysis. The obtained results and their explanation are discussed in Section VI. Finally, Section VII concludes the paper with future directions.

II. Preliminaries

A. Natural Language Processing (NLP)

Natural Language Processing is a major component of Artificial Intelligence (AI). It enables robots to analyse and comprehend human language, enabling them to carry out repetitive activities without human intervention. Machines can analyse and comprehend human language through a process known as NLP. NLP-based approaches process a considerable amount of data to obtain useful knowledge. For that different data mining and machine learning approaches are used. Thus, text pre-processing should be applied in order to prepare text for further processing such as representation features engineering that are required to extract features and pass it to ML approaches. For example, pre-processing could include text tokenization, and stop-word removal.

B. Machine Learning (ML) Algorithms

ML is used in various applications, e.g., healthcare [16], hardware design [17], quality control [18], and NLP, where this work targets NLP application. Information is an organised collection of discrete pieces of data, and it conceals the whole spectrum of representational patterns. The machine's primary objective is to extract patterns that reflect a certain event. If the machine is able to recognise these patterns, then machine learning has taken place. It demonstrate that by adding fresh data or information, where the computer can make accurate predictions. The authors of [19] have mentioned that the advancements in machine learning especially deep learning enable us to design algorithms that use real-world information to make decisions that seem subjective. As shown in Section IV-B, there are different methods to prepare text for further processing. *Text tokenization*, which is also called text segmentation or lexical analysis, groups the text into tokens/words separated by space. Stop-words such as articles (e.g., a, an, the), conjunctions (e.g., and, but, if), and prepositions (e.g., in, at, on) [20], do not represent a specific meaning. Thus, they should be eliminated. Features in ML are essentially numerical attributes. However, the data may not contain numerical attributes, such as in sentiment analysis. Thus, various types of features (e.g., word, character, so on) are converted into numerical features where such operation is called representation and choosing from them which make ML working properly is called *feature engineering (feature selection and feature extraction)*.

C. Hate Speech

Recently, the broad usability of smartphones and the high availability of internet access increased the number of users on social media. Moreover, the rapid growth of social media has made it practically unattainable to manually monitor and inspect the massive amount of messages published online every day. Also, social media witnessed a substantial increase in hate and abusive speech, which is a severe problem worldwide that threatens the solidarity of civil communities. Therefore, automatic detection for hate speech, utilizing various classification techniques, is required to filter such harmful content. Twitter is one of the most importing social media platform which is ubiquitous, informal, and unstructured at the same time. Tweets usually have abbreviations, acronyms, spelling errors, and non-ideal punctuation so designing a model to handle this will be an interesting topic for future work.

D. Transfer Learning (TL)

ML still has some constraints for specific real-world domains. For example, the requirement of having a tremendous amount of training data which have a distribution similar to the testing data could be difficult to satisfy [21]. Thus, semi-supervised learning could be utilized due to the shortage of labeled data. However, for a small amount of unlabeled data, the build model would be defective. Therefore, transfer learning is a promising procedure for such systems. Transfer learning (TL) is a branch of machine learning (ML) which aims to improve the performance of target learners on specific fields by transferring the knowledge possessed in separate but connected source domains [21]. Thus, constructing target learners will have a reduced dependency on a large number of target-domain data. In ML models, knowledge is not retained or accumulated, where learning is performed without considering past learned knowledge in other tasks. However, in transfer learning, the learning process can be faster, more accurate, and require less training data. TL can be classified into: 1) homogeneous where the disciplines are of the identical feature space, 2) heterogeneous where the disciplines have diverse feature spaces.

E. Data Oversampling and Underselling (Re-Sampling)

With the tremendous increase in the size of the generated data in various applications, there is a lack of equality in the labeled data. However, various ML techniques assume equal distribution for the target classes which is not always a realistic assumption. Such class imbalance problems will have a good accuracy while other evaluation metrics including precision, recall, F1-score, and ROC (Receiver Operating Characteristics) score, will not have enough scores. As shown in Fig. 1, Re-sampling including under-sampling or oversampling could be used to resolve the problem of an imbalanced data set. Under-sampling reduces the amount of the majority target samples. On the other hand, oversampling raises the quantity of minority class instances by yielding new instances or reproducing some instances [22].

Fig. 1. Undersampling vs Oversampling [22].

III. Related Work

Various researches have been conducted to detect hate speech as a wide notion with different types in the English language. Many proposed works performed hate speech detection as a binary classification problem and considered a broad concept such as detecting bullying and derogatory language. In [23], the authors presented an original technique to detect hatred speech in English tweets. For that, they utilized three models, i.e., logistic regression (LR), XGBoost classifier (XGB), and support vector machine (SVM). The obtained performance showed competitive results compared to standard stacking, base classifiers, and majority voting techniques. The authors of [24] determined and discussed challenges encountered by online automatic techniques for hate speech detection in text. The limited availability of the data, sensitivity in language, and the exact definition of what forms of hate speech are well-known challenges. They proposed a SVM technique with high performance while the decisions are easier to interpret than neural methods. However, the used datasets did not include Arabic text.

In [25], the authors used different machine learning algorithms for the automatic identification of hate speech in tweets written in the Indonesian language. Their results showed that the Multinomial Naive Bayes algorithm has the most promising results with a value of 71.2% and 93.2% for accuracy and recall, respectively. The authors of [2] researched the capability of deep learning based on Convolutional Neural Networks (CNN), CNN-long short-term memory networks (CNN-LSTM), and bidirectional LSTM (BiLSTM-CNN) to automatically detect hateful content posted on social media. For that, they used the ArHS dataset with 9833 tweets, which is believed to be the largest Arabic dataset with hate speech content.

The authors of [14] aimed to identify Cyber hate speech within the Arabic content of Twitter where they used various NLP and ML techniques. In [26], the authors used Twitter to construct an Arabic text detection hate speech model. They use this knowledge to analyze a dataset of 11 thousand tweets. They apply the Term Frequency — Inverse Document Frequency (TF-IDF) words representation to the SVM model. Finally, they presented four deep learning models that can notice and classify Arabic hate speech on Twitter into several types.

In [27], the authors were the first who addressed the problem of recognizing speech encouraging religious hatred in the Arabic Twitter. Thus, they were able to detect messages that use provocative sectarian speech to promote hatred and violence against people based on their religious beliefs. They found that a simple Recurrent Neural Network (RNN) architecture with Gated Recurrent Units (GRU) can adequately detect religious hate speech. The used data set is available online at [28]. The authors of [29] presented the foremost publicly-available Levantine Hate Speech and Abusive (L-

HSAB) Twitter dataset. It is intended to be a benchmark dataset for automatic detection of online Levantine harmful contents. The dataset, which is a available at [30], includes 5,846 tweets that could be of Normal class, Abusive, or Hate speech.

Considerable work has been investigated for hatred speech detection in the English language. However, rare work has targeted the detection of hate speech in the Arabic language. The majority of the Arabic research targeted web pages and search engines, while a few targeted comments on social networks. In this work, we target the Arabia language and use the data set of [28]. Thus, our constructed models would be mainly compared with [27].

IV. Proposed Methodology

The proposed architecture for Arabic hate speech detection is showing in Fig. 2. It includes the subsequent major steps: collection of labelled text document/tweet, text preprocessing, text representation and feature extraction, building of classification models (learning), and Relearning (testing) and classification process.

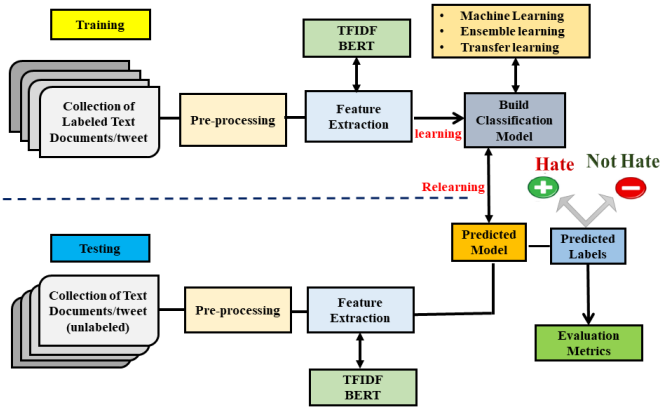


Fig. 2. The Proposed Architecture for Hate Speech Detection Model.

A. Data set: Collection of Labelled Text Document/Tweet

In this work we used the data set which was collected by [27] and it is available online at [28]. The data set contains 6164 Arabic tweets and concentrates on the four most typical sacred religions in the Middle East, which are Islam (93.0%), Christianity (3.7%), Judaism (1.6%), and Atheism (0.6%). Originally, the training data set contains 5,569 examples, while the testing data set contains 567 document. The data works for binary classification with two hateful and non hateful classes represented by 1 and 0 respectively. Since data re-sampling is be utilized to settle the issue of an imbalanced data set, we performed re-sampling technique. According, the model built with data oversampling is called *Classifier*-Over while the model built with data under sampling is called *Classifier*-Under, where *Classifier* could be any of the six models we used.

B. Data Pre-Processing

Text pre-processing includes various techniques that prepare text for further processing. Pre-processing aims to remove the unwanted words from the text, e.g., punctuation, slang, and stop words. Usually, we have to deal with various preprocessing techniques and a combination of them, including:

1) Tokenization: Tokenization is the activity of splitting text into terms, phrases, symbols or additional important elements, called tokens [31]. The obtained elements can be single items (1-gram) or a series of n words (n-gram). Items can be phonemes, syllables, letters, words or even sentences.

2) Stopwords Removal: Our work targets Arabic text. Thus, for pre-processing stage, we first remove the non-Arabic text. Every non-Arabic character is replaced with a whitespace character. Moreover, we remove *stopwords*, which appear frequently in the text and are not important for text classification, e.g., لكن ، عن ، على ، في ، أو ، مع . A list of the most frequently used Arabic stop words is available at [20]. Approximately, 20%–30% of the total words in a record are stopwords, that is, terms that can be removed as they are redundant without any semantic value [32]. The traditional approach for extracting stopwords includes a pre-filled list, containing all words that are semantically irrelevant to a specific language. This technique is a static. On the other hand, the stopwords are recognized online and not specified previously for the dynamic technique. The features are specified based on their importance. Similar to the removal of stopwords, this work eliminates the punctuation and digits from the Arabic text.

3) Stemming and Lemmatization: In the Arabic language, various words could be generated from the basic/root word. For example, the words لعبة ، يلعب، ملعب، لاعب، لاعبة are derived from the word لعب. Thus, the stemming operation is applied to reduce the words into their stems. Stemming algorithms can be categorized into three classes: truncation, statistical and mixed techniques. This work conducts Light Stemming for Arabic words to reduce words to their stems. Light stemming withdraws common affixes from words without declining them to their stems. The main idea is that numerous word variants do not have identical meanings although they are developed from the same root. Light stemming aims to improve feature drop while keeping the words' meanings. It removes some specified prefixes and suffixes from the word instead of removing the original root. Lemmatization is a pre-processing approach similar to stemming; the purpose is to decrease the morphological forms of a word to its lemma.

There are many approaches proposed for stemming Arabic words, e.g., light stemming, morphological analysis, statistical-based stemming, and N-grams. Some approaches are language-independent while other approaches are language-dependent. Statistical approaches are language-dependent. Thus, can be tailored for Arabic. Light stemming does not reduce the word into a three-letter stem. However, it just expels the prefixes and suffixes and can achieve good information retrieval without morphological studies.

C. Text Representation/ Feature Engineering

The feature selection procedure allows selecting some of the initial feature set, removing the attributes with little predictive capability. For example, wrapper methods in WEKA, execute an investigation over the potential subsets of the initial feature set, assessing the implementation of a classifier over

each one. However, wrapper methods are unusable for large problems. Thus, they are discarded in text classification. On the other hand, filter methods are independent of the classifier with a less computational expense. Filters applied before using the feature selection metric incorporates the removal of infrequent words and overly common words. There are various techniques to convert string data into numerical data such as Bag of words (BoW), Term Frequency — Inverse Document Frequency (TFIDF), Word2Vec, and Bidirectional Encoder Representations from Transformers (BERT). In the following section, some of these techniques will be explained [33][13].

1) Bag of Words (BoW): BoW is a textual representation method suitable for classification models, where the text is viewed as a set of words without considering its syntax or semantics. BoW reveals whether a word is present in the document or not, where the order of the words in the document is insignificant [34]. While constructing the BoW the list of stop words is excluded since they appear frequently with little of useful information. The performance of various ML methods we built utilizing BoW was poor due to the loss of semantic and syntactic information between words. Thus, we used other representation techniques that can handle semantics and syntactic in order to increase performance.

2) Term Frequency — Inverse Document Frequency (TF-IDF): Term Frequency (TF) is a well-known textual representation model which is similar to the BoW technique. However, TF relies on the recurrence of the term in a provided text, while BoW depends on its presence. TF is the frequency of any *term* in a given *document*, which is expressed as given in Equation 1. However, words that are *common* in every document, such as *articles*, *conjunctions* , and *prepositions* rank low because they don't express much to the document. Therefore, we use Inverse Document Frequency (IDF) to reduce the significance of phrases that occur very often in the document collection and improve the importance of phrases that occur infrequently. IDF is constant per corpus and accounts for the ratio of documents that include that specific *term*. It is expressed as given in Equation 2. TF-IDF is a statistical standard to assess how much a phrase is related to a manuscript in a set of documents, i.e., corpus. TF-IDF is computed by multiplying TF by IDF. TF-IDF is regarded as a simple procedure for text classification. Thus, the TF-IDF is developed during model training and then utilized for the test set.

$$TF = \frac{Number\ of\ times\ a\ term\ appear\ in\ the\ document}{Total\ number\ of\ terms\ in\ the\ document} \quad (1)$$

$$IDF = Log_{10}\frac{Total\ number\ of\ Documents}{Number\ of\ documents\ that\ includes\ the\ term} \quad (2)$$

3) Word2vec: Word2vec is a word-embedding technique. It is useful in constructing guidance engines and making sense of sequential data [35]. Word2vec is a prediction-based approach built based on a persistent bag-of-words (CBOW) and a skip-gram (SG). These measures utilize small neural networks (NN) to realize the mapping of words to a point in a vector space. To train the word2vec, the number of the embedding dimensions is set between 50 and 500 while the length of the context window is set between 5 and 10 [36].

4) Bidirectional Encoder Representations from Transformers (BERT): BERT is a contextualized word representation model founded on a multilayer bi-directional transformer-encoder, where the transformer neural network uses parallel attention layers rather than sequential recurrence [37]. The authors of [37] introduced **BERT** (Bidirectional Encoder Representations from Transformers), where the proposed framework includes two phases: (1) pre-training: the model is prepared on unlabeled data over various pre-training tasks, and (2) fine-tuning: the model is initialized with the pre-trained parameters. Then, all of the parameters are fine-tuned using labeled data from the downstream tasks. Thus, deep bidirectional architectures of BERT allow the same pre-trained model to successfully embark on a broad set of natural language processing tasks. In this work, we used TF-IDF and Bidirectional Encoder Representations from Transformers for the Arabic language called (AraBERT).

D. Building of Classification Models (Learning)

We utilized the six ML-based models described next which are Gradient Boosting (GB), K-Nearest Neighbor (K-NN), Logistic Regression (LR), Naive Bayes (NB), Passive Aggressive Classifier (PAC), Support Vector Machine (SVM). Thus, we built the classifiers for these models. The default data partitioning was 80/20 where 80% of the data are used for building the classification model and the remaining 20% are used for model testing. For additional investigation, we built the same models/classifiers for 70/30 data partitioning as well as 90/10. Thus, we have a total of 18 configurations. Moreover, we used transfer learning and build a model based on Ara-BERT and another model called AJGT-BERT. The evaluation of the 20 models/classifiers we built based on different classification metrics is explained in Section VI.

1) Gradient Boosting: Boosting algorithm is an ensemble learning algorithm utilizing learning theory [38]. Thus, an group of weak classifiers with low classification accuracy are used to build a strong classifier with higher accuracy. The training procedure of expanding algorithm is incremental, i.e., develops a new classifier in each iteration. Thus, the classifier ranks all instances to evaluate the importance of each instance. Then, the importance of the earlier samples with misclassification are improved. Finally, a stable and better performance classification model is obtained. The gradient boosting (GB) algorithm [38] is an amended algorithm based on the classic boosting algorithm, where it shows better learning ability. Like boosting, GB builds the model with an iterative design, but the model is extended with an optimized loss function. Gradient Boosting is a method drawing awareness for its prediction quickness and accuracy, particularly with extensive and complicated data. Building a GB model start by creating a single leaf rather than a tree or a stump. This leaf symbolizes an starting prediction for the class of all instances. Like AdaBoost, Gradient Boosting build a fixed sized tree based on previous tree's errors where each tree can be larger than a stump. GB scales all the trees by the same amount.

2) K-Nearest Neighbor (K-NN): The k-nearest neighbors (KNN) algorithm is a straightforward, easy-to-implement supervised ML algorithm that is applicable for both classification

and regression [33]. K-NN supposes the likeness between the recent and the known cases. Then, place the recent case into the class that is most identical to the available classes. K-NN algorithm keeps all the available data and categorizes a new instance based on the similarity. K-NN algorithm does not make any hypothesis on underlying data. Thus, it is a non-parametric algorithm [23]. KNN is a lazy learner algorithm because it holds the dataset and it achieves an activity on the dataset at classification, i.e., does not memorize from the training set immediately. However, as the number of independent variables increases the algorithm gets incredibly slower.

3) Logistic Regression (LR): It is comparable to linear regression. However, it expects if a value is True or False rather that predicting a continuous value. Linear regression fits a curve to the data while LR fit an "S" shaped logistic function [23]. Logistic regression can perform on both continuous and discrete data. Thus, its ability to predict the probability and classify new samples makes it a popular ML method, where it is referred to as a probabilistic classifier since it predicts the probability of an output. Usually, logistic regression is used for classification. In linear regression, we fit the line between the data using "least squares". However, the concept of "residual" does not apply to LR where the concept of "maximum likelihood" is rather used.

4) Naive Bayes (NB): Naive Bayes classifier relies on Bayes Theorem, which works on conditional probability. The conditional probability is the likelihood that something will happen, given that something else has already occurred [39] [40]. The formula for calculating the conditional probability is given in Equation 3, where H is the hypothesis and E is the evidence.

$$P(H|E) = \frac{P(E|H) \times P(H)}{P(E)} \tag{3}$$

For a set of labeled training data, NB evaluates different model parameters, e.g., the likelihood of each class label to appear. Then, predict the class for any given test data based on its probability to be assigned for different classes. The maximum probability determines the predicted class [23].

5) Passive Aggressive Classifier (PAC): This is an online ML algorithm, where it responds as passive for correct classifications and as aggressive for any miscalculation. In PAC we train a system incrementally by providing it samples sequentially, i.e., individually or in small groups called mini batches [33] [41]. The primary principle of this algorithm is that it notices data, learns from the data, and discards it without the need of storing the data. However, in batch learning the entire training dataset is used at once. Thus, PAC is suitable for systems that acquire data in a steady stream such as news and social media [42]. When the prediction is correct, we keep the model without any changes since the data in the example was not enough to change the model. Thus, it is called Passive. However, for incorrect prediction we introduce some changes to the model that could correct it. Thus, it is called Aggressive. PAC algorithm proved its effectiveness for online learning to solve various real-world problems [43].

6) Support Vector Machine (SVM): SVM is a supervised classifier. For a set of labeled training data, SVM realizes a hyperplane that distinctly classifies the data points while maximizing the margin between the data instances and the hyperplane itself [13] [44]. Then, the class of test data is determined based on the realized hyperplane [42].

7) AraBERT: Based on BERT [37], the authors of [45] presented **AraBERT** (transformer-based Model for Arabic Language Understanding), where they pre-trained BERT, especially for the Arabic language aiming to achieve the same success as BERT. The authors of [45] used the original configuration of BERT which has 12 encoder blocks, 768 hidden dimensions, 12 attention heads, 512 maximum sequence length, and a total of 110M parameters. Then, to better fit the Arabic language, they introduced additional preprocessing before model's pre-training. To avoid information loss, they maintained words with Latin characters so they can cite named entities and scientific phrases in their original language. Thus, after eliminating duplicate sentences, the final size of the pre-training data sets was 70 million sentences.

8) AJGT-BERT: The authors of [46] introduced an Arabic tweets corpus documented in Jordanian dialect and Modern Standard Arabic (MSA) annotated for sentiment analyses. The generated AJGT corpus consists of 1,800 tweets with 900 classified as positives and the remaing 900 are negatives. The Arabic Jordanian General Tweets (AJGT) data sets is publicly available online at [46].

V. Experimental Analysis

A 2 $\times$ 2 matrix, which is called a confusion matrix, is created to visually illustrate the performance of a binary supervised learning problem. Table I shows the confusion matrix for Arabic hate speech detection. It includes four classes, which are true positive, true negative, false positive, and false negative. In this work, **True Positive** (TP) indicates that the comment is actually hate speech and correctly classified as hate speech. **True Negative** (TN) indicates that the comment is non-hate speech and correctly classified as non-hate speech. **False Positive** (FP) means that the comment is actually non-hate speech but incorrectly classified as hate speech, and **False Negative** (FN) describes the comment that is actually hate speech but incorrectly classified as non-hate speech. For any classification model, we aim to maximize the value of TP and TN and minimize the value of FP and FN.

TABLE I. Confusion Matrix of Arabic Hate Speech Detection

	Actual Hate Speech	**Actual Non-Hate Speech**
Predicted Hate Speech	True Positive(TP)	False Positive(FP)
Predicted Non-Hate Speech	False Negative (FN)	True Negative(TN)

A. Implementation Environment

To accomplish all investigations in this work, we utilized a PC with the following details: Intel R © Core(TM) i7-6850 K processor with 8 GB RAM and 3.360 GHz frequency. Regarding the software, we have used Python 3.8.0 programming with Anaconda [Jupyter notebook] for ML and Colab for transfer learning models. We used various libraries such as NumPy, Pandas, Sci-kit-learn TensorFlow, and Keras.

B. Evaluation Metrics

As given in Equation 4, **accuracy** denotes the number of rightly classified data samples over the total number of data samples. However, for an unbalanced dataset, where positive and negative classes have a different number of instances, the accuracy is not suitable to evaluate the model. **Precision** (positive predictive value) as defined in Equation 5, should be 1 for a perfect classifier while the value of FP is zero. **Recall** which is known as sensitivity or true positive rate is defined as given in Equation 6. For a perfect classifier, recall should be 1 while the value of FN is zero. For an ideal classifier, both precision and recall are 1. **F1-score** is a metric that depends on both precision and recall and is defined as given in Equation 7. F1-score becomes 1 only when precision and recall are both 1. So, F1-score is the harmonic mean of precision and recall and it is a better measure than accuracy [41] [44].

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

$$Precision = \frac{TP}{TP + FP} \tag{5}$$

$$Recall/Sensitivity = \frac{TP}{TP + FN} \tag{6}$$

$$F1\ Score = 2 \times \frac{Precision \times Recall}{Precision + Recall} \tag{7}$$

VI. Discussion and Results

In this section, we will explain the result of various machine learning models which we have used in this study. We have designed different models to detect and classify religious Arabic hate speech based on various methods, e.g., data partitioning, re-sampling and transfer learning. For that, we have divided our data (train/test) for three scenarios. The data is partitioned into (70/30), (80/20), and (90/10). For each partitioning, we use the original data sets in addition to the oversampling and under-sampling techniques. The best-obtained classification performance in partitioning scenarios was for (80/20), where a detailed explanation is given in Section VI-A. Then, Section VI-B explains the classification performance for 70/30 data partitioning while Section VI-C is dedicated to 90/10 data partitioning. The proposed models were evaluated on a testing data set related to religious hate speech in Arabic text [28]. In order to enhance the performance of the classifiers that we build for six ML algorithms, we have extended this implementation to include transfer learning methods. The transfer learning models called Ara-BERT and AJGT-BERT. The comparison of the performances of all models have been done in terms of accuracy, recall, precision, and F1 score using Arabic hate speech data set.

A. Hate Speech Detection for (80/20) Data Partitioning

Fig. 3 shows the various obtained classification metrics based on various models, i.e., GB, K-NN, LR, NB, PAC, SVM, Ara-BERT, and BERT-AJGT. Clearly, the Ara-BERT model achieves the best classification metrics followed by AJGT-BERT while the KNN classifier has the lowest metrics. A detailed explanation for each metric is given next.

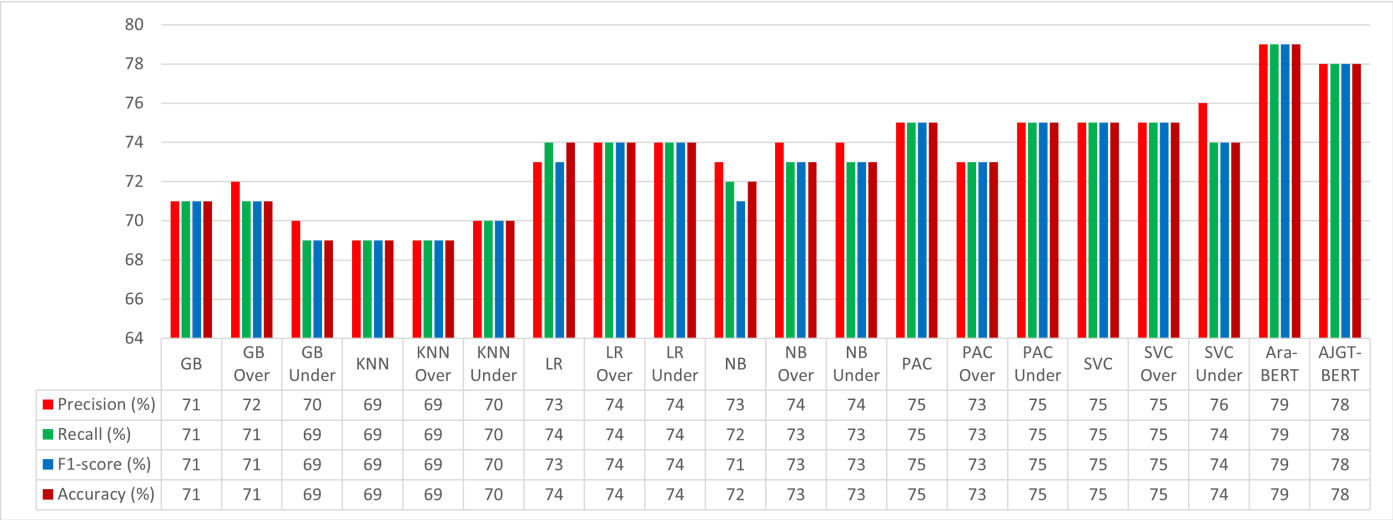
The obtained **precision** for the original data without re-sampling based on 8 classifiers is shown in Fig. 3. Both PAC and SVC has a precision of 75%. Moreover, the transfer-based classifiers, i.e., **Ara-BERT** and AJGT-BERT, have the highest precision with **79%** and 78%, respectively. KNN classifier has the lowest precision of 69%. The obtained **Recall** based on 8 classifiers without re-sampling are very similar to the obtained precision. The transfer-based classifiers, i.e., **Ara-BERT** and AJGT-BERT, have the highest Recall with **79%** and 78%, respectively, while KNN classifier has the lowest precision of 69%. Moreover, both PAC and SVC have a Recall of 75%. As given in Equation 7, **F1-Score** depends on both precision and recall. Regarding the obtained F1-Score for the various classifiers, the transfer-based classifiers, i.e., **Ara-BERT** and AJGT-BERT, have the highest F1-Score with **79%** and 78%, respectively, while KNN classifier has the lowest F1-Score of 69%. When evaluating the **accuracy** we notice that it is very close to F1-Score of the various classifiers. Transfer-based classifiers are the highest while KNN classifier has the lowest accuracy, i.e., the accuracy of **Ara-BERT** model is **79%**. Next, we are going to explain the classification metrics with data over-sampling and under-sampling.

Hate Speech Detection for (80/20) Data Partitioning with Oversampling

With oversampling, we apply oversampling technique by increase the minority of samples to be same to majority like 2196 to 3650. Then, we used 80% of the data to construct the classification model and the rest 20% for testing the model/classifier. The obtained Precision, Recall, F1-Score and Accuracy of the various classifiers are shown in Fig. 3 where the name of a specific classifier is indicated as *Classifier*-Over. The various metrics for the Gradient Boosting and K-Nearest Neighbor (K-NN) classifiers with data oversampling remains the same as the original data set. However, with over-sampling the LR and NB classifiers have a 1% to 2% improvement of Precision and F-Score. Data oversampling reduces the various metrics of PAC by 2%. However, SVM based classifier is unaffected by data oversampling. Transfer-based classifiers are unaffected by oversampling. Thus, **Ara-BERT** has a value of **79%** for Precision, Recall, F1-Score and Accuracy.

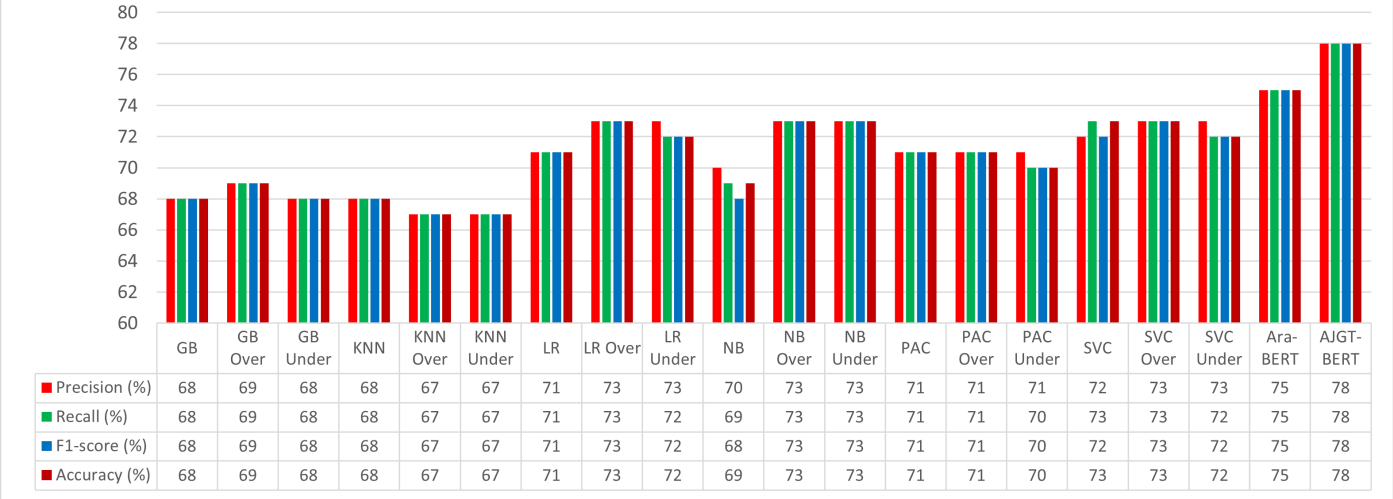
Hate Speech Detection for (80/20) Data Partitioning with Under-Sampling

With under-sampling, we decreased the number of normal data from 3650 to 2196 to be equal to the hate speech. Then, we employed 80% of the data to construct the model and the remaining 20% for testing the model/classifier. The obtained Precision, Recall, F1-Score and Accuracy of the various classifiers are shown in Fig. 3 where the name of a specific classifier is indicated as *Classifier*-Under-sampling. With under-sampling, the various metrics of GB classifier are



	GB	GB Over	GB Under	KNN	KNN Over	KNN Under	LR	LR Over	LR Under	NB	NB Over	NB Under	PAC	PAC Over	PAC Under	SVC	SVC Over	SVC Under	Ara-BERT	AJGT-BERT
Precision (%)	71	72	70	69	69	70	73	74	74	73	74	74	75	73	75	75	75	76	79	78
Recall (%)	71	71	69	69	69	70	74	74	74	72	73	73	75	73	75	75	75	74	79	78
F1-score (%)	71	71	69	69	69	70	73	74	74	71	73	73	75	73	75	75	75	74	79	78
Accuracy (%)	71	71	69	69	69	70	74	74	74	72	73	73	75	73	75	75	75	74	79	78

Fig. 3. Various Classification Metrics for 80/20 Data Partition with Different Models.



	GB	GB Over	GB Under	KNN	KNN Over	KNN Under	LR	LR Over	LR Under	NB	NB Over	NB Under	PAC	PAC Over	PAC Under	SVC	SVC Over	SVC Under	Ara-BERT	AJGT-BERT
Precision (%)	68	69	68	68	67	67	71	73	73	70	73	73	71	71	71	72	73	73	75	78
Recall (%)	68	69	68	68	67	67	71	73	72	69	73	73	71	71	70	73	73	72	75	78
F1-score (%)	68	69	68	68	67	67	71	73	72	68	73	73	71	71	70	72	73	72	75	78
Accuracy (%)	68	69	68	68	67	67	71	73	72	69	73	73	71	71	70	73	73	72	75	78

Fig. 4. Various Classification Metrics for 70/30 Data Partition with Different Models.

reduced by 1% while they are increased by 1% for the KNN classifier. Similarly, the Precision and F1-Score of the LR classifier are enhanced by 1% while its recall and accuracy are unchanged. All metrics of the NB classifier are increased by 1% for under-sampling while they remain the same for the PAC classifier. The precision of SVC is increased by 1% to reach 76% while its recall, F1-Score and Accuracy are reduced by 1% to become 74% for all of them. Transfer-based classifiers are unaffected by under-sampling. Thus, **Ara-BERT** has a value of **79%** for Precision, Recall, F1-Score and Accuracy while the AJGT-BERT classifier has a value of 78% for the same metrics.

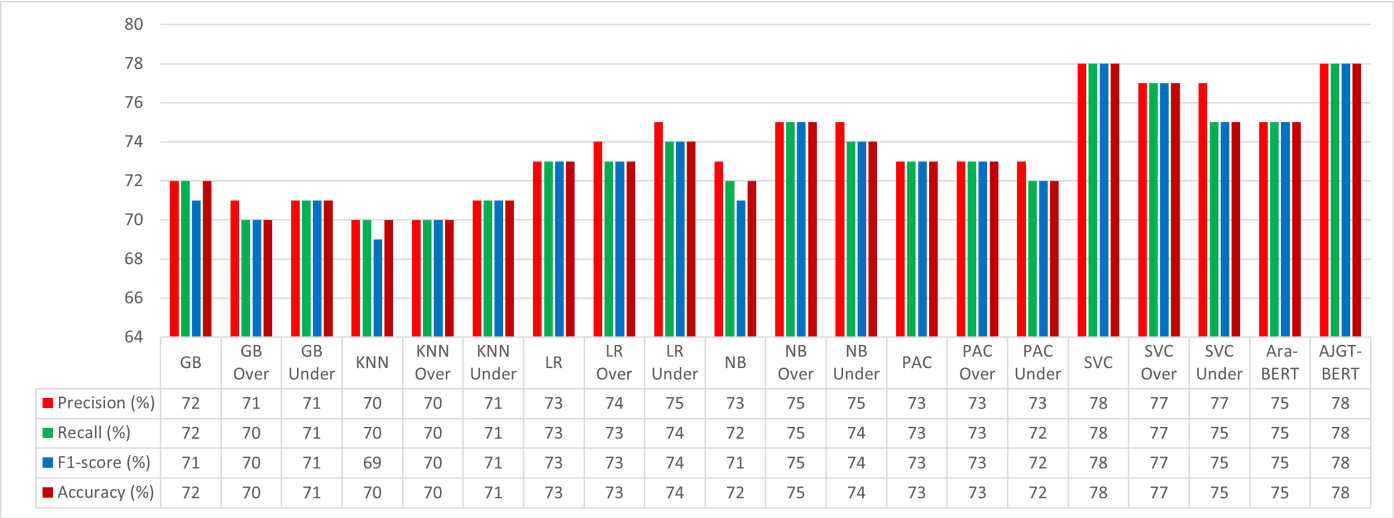
Based on the shown result for 80/20 data partitioning, the Ara-BERT models archives the best evaluation metrics with 79% for the precision, recall, F1-Score and accuracy. Data re-sampling introduced a 1% to 2% improvement where such insignificant gain is due to the original distribution of the training data.

B. Hate Speech Detection for (70/30) Data Partitioning

Fig. 4 shows the various classification metrics for the different models with 70/30 data partitioning. AJGT-BERT classifier has the best metrics of 78% (for precision, recall, F1-Score and accuracy) while Ara-BERT classifier has a value of 75%. There are various classifiers that achieves 73% for all classification metrics including, LR-Over, NB-Over, NB-Under, and SVC-Over. We notice that sometimes data re-sampling introduces a minor improvement of 1% to 2% in few classifiers.

C. Hate Speech Detection for (90/10) Data Partitioning

Fig. 5 shows the various classification metrics for the different models with 90/10 data partitioning. AJGT-BERT and SVC have the highest performance with 78% (for precision, recall, F1-Score and accuracy) followed by SVC with oversampling (SVC-Over) with 77% performance. Ara-BERT, SVC-Over,



	GB	GB Over	GB Under	KNN	KNN Over	KNN Under	LR	LR Over	LR Under	NB	NB Over	NB Under	PAC	PAC Over	PAC Under	SVC	SVC Over	SVC Under	Ara-BERT	AJGT-BERT
Precision (%)	72	71	71	70	70	71	73	74	75	73	75	75	73	73	73	78	77	77	75	78
Recall (%)	72	70	71	70	70	71	73	73	74	72	75	74	73	73	72	78	77	75	75	78
F1-score (%)	71	70	71	69	70	71	73	73	74	71	75	74	73	73	72	78	77	75	75	78
Accuracy (%)	72	70	71	70	70	71	73	73	74	72	75	74	73	73	72	78	77	75	75	78

Fig. 5. Various Classification Metrics for 90/10 Data Partition with Different Models.

TABLE II. COMPARISON OF CLASSIFICATION RESULTS OVER [28] DATASET

	Precision	Recall	F1-Score	Accuracy
Related work [27]	76	78	77	79
Ara-BERT with 70/30	75	75	75	75
AJGT-BERT with 70/30	**78**	**78**	**78**	78
Ara-BERT with 80/20	**79**	**79**	**79**	**79**
AJGT-BERT with 80/20	**78**	**78**	**78**	78
Ara-BERT with 90/10	75	75	75	75
AJGT-BERT with 90/10	**78**	**78**	**78**	78
SVC with 90/10	**78**	**78**	**78**	78

and NB-Over classifiers have 75% performance while NB-Under and LR-Under are 74%.

Table II shows a comparison of classification results over [28] where we used their dataset. The table compares various models we build with the related work [27]. Clearly, we were able to build models similar to or better than the related work. In most cases, transfer learning based models have the highest precision, recall, F1-Score, and accuracy, while other ML models have a very similar metrics

VII. CONCLUSION

Hate speech is one of the major problems at this time, especially with the increasing number of users on social media. At the same time, an increasing number of crimes became a serious concern that threatens the cohesiveness and structure of civilian societies. Therefore, this work presents an efficient framework to detect Arabic hate speech based on the content of social networks. We utilize various ML and DL models to perform an efficient classification of users' comments. Based on the content of this work, the classes are hate speech or normal. The proposed framework has six ML algorithms and two DL, which are Gradient Boosting, K-Nearest Neighbor, Logistic Regression, Naive Bayes, Passive Aggressive Classifier, Support Vector Machine, Ara-BERT, and BERT-AJGT. For wider investigation, we utilized various scenarios of data partitioning, re-sampling techniques, and transfer learning. We were able to successfully have various classification models with better results in terms of precision, recall, F1-Score, and accuracy compared to the most relevant related work. For future work, we aim to create huge and benchmark data sets. Moreover, working with the mixed language problem, multi-model and data augmentation can be interesting topics for future work on this topic, especially for the Arabic language. In future work, the classification can be expanded to cover many classes such as racism, misogyny, religious discrimination and so on.

REFERENCES

[1] H. Butt, M. R. Raza, M. J. Ramzan, M. J. Ali, and M. Haris, "Attention-based CNN-RNN Arabic text recognition from natural scene images," *Forecasting*, vol. 3, no. 3, pp. 520–540, 2021.

[2] R. Duwairi, A. Hayajneh, and M. Quwaider, "A deep learning framework for automatic detection of hate speech embedded in arabic tweets," *Arabian Journal for Science and Engineering*, vol. 46, no. 4, pp. 4001–4014, 2021.

[3] I. Guellil, H. Saâdane, F. Azouaou, B. Gueni, and D. Nouvel, "Arabic natural language processing: An overview," *Journal of King Saud University-Computer and Information Sciences*, vol. 33, no. 5, pp. 497–507, 2021.

[4] K. Darwish, W. Magdy, and A. Mourad, "Language processing for arabic microblog retrieval," in *21st ACM international conference on Information and knowledge management*, 2012, pp. 2427–2430.

[5] J. Chen, Y. Chen, Y. He, Y. Xu, S. Zhao, and Y. Zhang, "A classified feature representation three-way decision model for sentiment analysis," *Applied Intelligence*, vol. 52, no. 7, pp. 7995–8007, 2022.

[6] G. Kovács, P. Alonso, and R. Saini, "Challenges of hate speech detection in social media," *SN Computer Science*, vol. 2, no. 2, pp. 1–15, 2021.

[7] A. Matamoros-Fernández and J. Farkas, "Racism, hate speech, and social media: A systematic review and critique," *Television & New Media*, vol. 22, no. 2, pp. 205–224, 2021.

[8] K. Barker and O. Jurasz, "Online misogyny as a hate crime:# timesup," in *Misogyny as Hate Crime*. Routledge, 2021, pp. 79–98.

[9] A. Al-Hassan and H. Al-Dossari, "Detection of hate speech in social networks: a survey on multilingual corpus," in *6th international conference on computer science and information technology*, vol. 10, 2019.

[10] D. Sultan, A. Suliman, A. Toktarova, B. Omarov, S. Mamikov, and G. Beissenova, "Cyberbullying Detection and Prevention: Data Mining in Social Media," in *International Conference on Cloud Computing, Data Science & Engineering*. IEEE, 2021, pp. 338–342.

[11] S. U. Maheswari and S. Dhenakaran, "Analysis of Approaches for Irony Detection in Tweets for Online Products," in *Innovations in Computational Intelligence and Computer Vision*. Springer, 2022, pp. 141–151.

[12] I. Kwok and Y. Wang, "Locate the hate: Detecting tweets against blacks," in *Twenty-seventh AAAI conference on artificial intelligence*, 2013.

[13] A. Y. Muaad, H. J. Davanagere, D. Guru, J. Benifa, C. Chola, H. Al-Salman, A. H. Gumaei, and M. A. Al-antari, "Arabic document classification: Performance investigation of preprocessing and representation techniques," *Mathematical Problems in Engineering*, vol. 2022, 2022.

[14] I. Aljarah, M. Habib, N. Hijazi, H. Faris, R. Qaddoura, B. Hammo, M. Abushariah, and M. Alfawareh, "Intelligent detection of hate speech in arabic social network: A machine learning approach," *Journal of Information Science*, vol. 47, no. 4, pp. 483–501, 2021.

[15] "Jordanian Ministry of Justice," last accessed September 14, 2022. [Online]. Available: http://www.moj.gov.jo/EchoBusV3.0/SystemAssets/5d38ea27-5819-443e-a380-b65c7e1f5b56.pdf

[16] M. Masadeh, A. Masadeh, O. Alshorman, F. Khasawneh, and M. Masadeh, "An efficient machine learning-based covid-19 identification utilizing chest x-ray images," *IAES International Journal of Artificial Intelligence*, pp. 356–366, 2022.

[17] M. Masadeh, O. Hasan, and S. Tahar, "Machine-learning-based self-tunable design of approximate computing," *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, vol. 29, no. 4, pp. 800–813, 2021.

[18] ——, "Machine learning-based self-compensating approximate computing," in *2020 IEEE International Systems Conference (SysCon)*. IEEE, pp. 1–6.

[19] I. Goodfellow, Y. Bengio, and A. Courville, *Deep learning*. MIT press, 2016.

[20] R. NL. Stopword lists. [Online]. Available: https://www.ranks.nl/stopwords/arabic

[21] F. Zhuang, Z. Qi, K. Duan, D. Xi, Y. Zhu, H. Zhu, H. Xiong, and Q. He, "A Comprehensive Survey on Transfer Learning," *Proceedings of the IEEE*, vol. 109, no. 1, pp. 43–76, 2021.

[22] R. Mohammed, J. Rawashdeh, and M. Abdullah, "Machine learning with oversampling and undersampling techniques: Overview study and experimental results," in *11th International Conference on Information and Communication Systems (ICICS)*, 2020, pp. 243–248.

[23] M. K. A. Aljero and N. Dimililer, "A Novel Stacked Ensemble for Hate Speech Recognition," *Applied Sciences*, vol. 11, no. 24, p. 11684, 2021.

[24] S. MacAvaney, H.-R. Yao, E. Yang, K. Russell, N. Goharian, and O. Frieder, "Hate speech detection: Challenges and solutions," *PLOS ONE*, vol. 14, no. 8, pp. 1–16, 08 2019.

[25] T. Putri, S. Sriadhi, R. Sari, R. Rahmadani, and H. Hutahaean, "A comparison of classification algorithms for hate speech detection," in *IOP Conference Series: Materials Science and Engineering*, vol. 830, no. 3. IOP Publishing, 2020, p. 032006.

[26] A. Al-Hassan and H. Al-Dossari, "Detection of hate speech in arabic tweets using deep learning," *Multimedia Systems*, pp. 1–12, 2021.

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

[28] "Religious Hate Speech Detection for Arabic Tweets," last accessed September 14, 2022. [Online]. Available: https://github.com/nuhaalbadi/Arabic_hatespeech

[29] H. Mulki, H. Haddad, C. B. Ali, and H. Alshabani, "L-HSAB: A Levantine Twitter Dataset for Hate Speech and Abusive Language," in *Proceedings of the third workshop on abusive language online*, 2019, pp. 111–118.

[30] Hala-Mulki. First-arabic-levantine-hatespeech-dataset. [Online]. Available: https://github.com/Hala-Mulki/L-HSAB-First-Arabic-Levantine-HateSpeech-Dataset/blob/master/Dataset/L-HSAB

[31] M. O. Hegazi, Y. Al-Dossari, A. Al-Yahy, A. Al-Sumari, and A. Hilal, "Preprocessing Arabic text on social media," *Heliyon*, vol. 7, no. 2, p. e06191, 2021.

[32] T. Kanan, B. Hawashin, S. Alzubi, E. Almaita, A. Alkhatib, K. A. Maria, and M. Elbes, "Improving arabic text classification using p-stemmer," *Recent Advances in Computer Science and Communications (Formerly: Recent Patents on Computer Science)*, vol. 15, no. 3, pp. 404–411, 2022.

[33] A. Y. Muaad, H. Jayappa Davanagere, J. Benifa, A. Alabrah, M. A. Naji Saif, D. Pushpa, M. A. Al-Antari, and T. M. Alfakih, "Artificial intelligence-based approach for misogyny and sarcasm detection from arabic texts," *Computational Intelligence and Neuroscience*, vol. 2022, 2022.

[34] J. Salminen, M. Hopf, S. A. Chowdhury, S.-g. Jung, H. Almerekhi, and B. J. Jansen, "Developing an online hate classifier for multiple social media platforms," *Human-centric Computing and Information Sciences*, vol. 10, no. 1, pp. 1–34, 2020.

[35] T. Mikolov, K. Chen, G. Corrado, and J. Dean, "Efficient estimation of word representations in vector space," *arXiv preprint arXiv:1301.3781*, 2013.

[36] F. K. Khattak, S. Jeblee, C. Pou-Prom, M. Abdalla, C. Meaney, and F. Rudzicz, "A survey of word embeddings for clinical text," *Journal of Biomedical Informatics*, vol. 100, p. 100057, 2019.

[37] 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.

[38] L. Mason, J. Baxter, P. Bartlett, and M. Frean, "Boosting algorithms as gradient descent," *Advances in neural information processing systems*, vol. 12, 1999.

[39] M. Masadeh, A. Aoun, O. Hasan, and S. Tahar, "Decision tree-based adaptive approximate accelerators for enhanced quality," in *International Systems Conference (SysCon)*. IEEE, 2020, pp. 1–5.

[40] A. Elouardighi, M. Maghfour, H. Hammia, and F.-z. Aazi, "A machine learning approach for sentiment analysis in the standard or dialectal arabic facebook comments," in *3rd International Conference of Cloud Computing Technologies and Applications (CloudTech)*, 2017, pp. 1–8.

[41] A. Y. Muaad, G. H. Kumar, J. Hanumanthappa, J. B. Benifa, M. N. Mourya, C. Chola, M. Pramodha, and R. Bhairava, "An effective approach for arabic document classification using machine learning," *Global Transitions Proceedings*, vol. 3, no. 1, pp. 267–271, 2022.

[42] K. Nagashri and J. Sangeetha, "Fake news detection using passive-aggressive classifier and other machine learning algorithms," in *Advances in Computing and Network Communications*. Springer, 2021, pp. 221–233.

[43] J. Lu, P. Zhao, and S. C. Hoi, "Online Passive-Aggressive Active Learning," *Machine Learning*, vol. 103, no. 2, pp. 141–183, 2016.

[44] A. Y. Muaad, H. J. Davanagere, M. A. Al-antari, J. B. Benifa, and C. Chola, "Ai-based misogyny detection from arabic levantine twitter tweets," in *Computer Sciences & Mathematics Forum*, vol. 2, no. 1. MDPI, 2021, p. 15.

[45] W. Antoun, F. Baly, and H. M. Hajj, "AraBERT: Transformer-based Model for Arabic Language Understanding," *CoRR*, vol. abs/2003.00104, 2020.

[46] K. M. Alomari, H. M. ElSherif, and K. Shaalan, "Arabic tweets sentimental analysis using machine learning," in *International Conference on Industrial, Engineering and Other Applications of Applied Intelligent Systems*. Springer, 2017, pp. 602–610.