Anti-Islamic Arabic Text Categorization using Text Mining and Sentiment Analysis Techniques

Rawan Abdullah Alraddadi[1]
Department of Computer Science
College of Computer Science and Engineering
Taibah University, Medina, Saudi Arabia

Moulay Ibrahim El-Khalil Ghembaza[2]
Department of Computer Science
College of Engineering and Information Technology
Unaizah Colleges, Qassim, Saudi Arabia

***Abstract*—The aim of this research is to detect and classify websites based on their content if it encourages spreading hate speech toward Islam and Muslims, or Islamophobia using sentiment analysis and web text mining techniques. In this research, a large dataset corpus has been collected, to identify and classify anti-Islamic online contents. Our target is to automatically detect the content of those websites that are hostile to Islam and transmitting extremist ideas against it. The main purpose is to reduce the spread of those webpages that give the wrong idea about Islam. The proper dataset is collected from different sources, and the two datasets for the Arabic language (balanced and unbalanced) have been produced. The framework of the proposed approach has been described. The approach used in this framework is based on supervised Machine Learning (ML) approach using Support Vector Machines (SVM) and Multinomial Naive Bayes (MNB) models as classifiers, and Term Frequency-Inverse Document Frequency (TF-IDF) as feature extraction. Different experiments including word level and tri-gram level on the two datasets have been conducted, and compared the obtained results. The experimental results shows that the supervised ML approach using word level is the finest approach for both datasets that produce high accuracy with 97% applied on the balanced Arabic dataset using SVM algorithm with TF-IDF as feature extraction. Finally, an interactive web-application prototype has been developed and built in order to detect and classify toxic language such as anti-Islamic online text-contents.**

Keywords*—Web Text mining; text classification; Arabic computational linguistics; natural language processing; SVM; MNB; opinion mining; hate speech; toxic language detection

I. Introduction and Background

Islamophobia has escalated in the past decades and this has shown in the real world as well as on online websites. This problem has affected Muslim communities especially those living in non-Muslim foreign countries, or any places containing extremists with anti-Islam ideas. Nowadays, it moved to the Internet where webpages are created specifically to attack the Muslim faith. This problem needs to be addressed and solved immediately. Unfortunately, there is a lack of research that addresses and solves the problems of classification in Arabic language, especially the classification of this type of texts. Therefore, this motivates us to search and explore methods and techniques to deal with this issue. Furthermore, the only paper found which is directly related to the work is the one proposed by Vidgen and Yasseri [1] who build a multi-class classifier for detecting islamophobia hate speech based on Twitter dataset; whereas the objective is to consider formal web contents. Moreover, the authors worked only with English content and not the Arabic content; whereas our objective concerns both Arabic language undertaken in this paper and English language conducted in our previous paper [2].

For these reasons, the need to detect anti-Islamic online content has increased, to prevent people from posting inaccurate or incorrect articles and rumors about Islam in order to prevent any attacks against Islam and Muslims. Such study became even more urgent after the mass killing that took place at the Al-Noor Mosque in New Zealand[1], and other unfortunate events such as the killing of Muslim students in America [3].

Therefore, the aim is to build an anti-Islamic related content analysis framework that classifies the content of webpages into anti-Islamic or not anti-Islamic classes. The main aim is to reduce the spread of those webpages that give the wrong idea about Islam. A framework to classify the content is needed in order to prevent these types of events in the future and stop the spreading of extremist ideas about the Muslim religion.

We focus on the accuracy of the classification, not the speed. Therefore, the effectiveness of the proposed approaches that is looking for lies in the correctness of the detection rather than the rapidity of detection. Moreover, the focus was on the formal language in the process of collecting the datasets instead of the informal language. Accordingly, for the Arabic datasets, the collected text was written in Modern Standard Arabic (MSA), which is the formal language instead of the Arabic dialects that are informal. The particular reason for choosing these types of writing texts is because they are more widespread on the web, and easier to process them uniformly.

The remainder of this paper is structured as follows: section two provides a review of some related work. Section three describes the proposed framework along with the data collection and the various stages of the methodology. In section four the implementation is provided. Section five contains experimental results and discussion followed by section six which illustrates the prototype of the proposed web-application. Finally, section seven concludes the paper with a summary and future work.

[1] "Christchurch shootings: The people killed as they prayed - BBC News." [Online]. Available: https://www.bbc.com/news/world-asia-47593693. [Accessed: 26-Mar-2021].

A. Web Text Mining

Web text mining enables the extraction and the integration of meaningful information from natural language text that exists in different webpages to be used by data mining algorithms [4]. The information can be discovered from distributed and heterogeneous environments. There are different heterogeneous forms for the information: structured information such as databases, unstructured information such as text files and semi-structured information such as XML documents.

Text mining uses various algorithms to convert unstructured text into structured data, so that it can be analyzed. Web text mining is a very useful process as it reduces the effort and time to extract only the meaningful information from large text data sources. Text mining uses Natural Language Processing (NLP) to analyze and understand the meaning of the text content to perform the required task. Text mining has different tasks including text classification, sentiment analysis and other methods such as text summarization and named-entity recognition.

B. Text Classification and Sentiment Analysis

Text classification is the process of classifying a text into binary classes or multi-classes based on different algorithms. Most of the classification systems go into almost four main phases as shown in Fig. 1: preprocessing, feature extraction, classification and evaluation. Preprocessing of the document includes tokenization, stop-words removal, special symbol removal and other preprocessing techniques such as changing all the words in the text to lowercase and replacing the regular expressions. Preprocessing can help in reducing the dataset dimensionality which eventually reduces the time and memory complexity [5].

When dealing with unstructured text, the text must convert into a structured feature to be used in mathematical modeling; and here comes the role of feature extraction. Feature extraction is the process of converting the unstructured text in the dataset into structured features that can be used by the classifier using either word embedding or weighted word technique. There are some common feature extraction techniques such as: Term Frequency-Inverse Document Frequency (TF-IDF), Term Frequency (TF), N-gram, bag of words, Word2Vec, and Global Vectors for Word Representation (GloVe).

Classification phase is considered the most important phase in text classification where the model learns from the training dataset, therefore, the classifier should be chosen carefully. The final phase consists of evaluating the performance of the model using one of the various measures. There are different methods for evaluating the model such as accuracy calculation, F1 Score and Matthews Correlation Coefficient (MCC) [5].

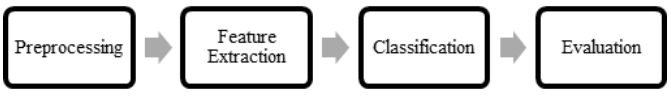

Fig. 1. Text Classification Phases.

Sentiment analysis, also called opinion mining, is an essential and special task associated with text classification where it classifies the text based on the sentimental polarities of the opinions contained in the text.

C. Arabic Computational Linguistics

Arabic language is widely used on the Internet and it is considered the fourth most used language after English, Chinese and Spanish according to the Internet World Stats[2]. Unfortunately, there is a lack of Arabic sentiment resources [6]. Arabic Sentiment Analysis (ASA) is considered a difficult task as it deals with unstructured text including a lot of rhetorical characteristics and implicit meanings, and classifies it as positive or negative documents.

The morphology of the Arabic language is difficult and different from other languages as it starts from right to left, also it does not contain uppercase or lowercase letters. Moreover, the meaning of the word depends on its position in the sentence. Arabic text can be classified into binary classes or multi-classes with the help of NLP to understand the meaning of Arabic text.

Arabic language is a challenging language and it requires good preprocessing techniques to achieve high accuracy. There are different good preprocessing techniques, but the most important ones are stop-words removals, lemmatization and stemming. Furthermore, Arabic language is a rich language that contains a large number of stop-words, where removing them can help to speed the analyses process.

Arabic language also contains many synonyms for the same word that gives the same meaning. Therefore, it is good to use lemmatization and stemming when working with Arabic text to achieve a good accuracy. Moreover, this may require some further preprocessing such as part of speech (POS) tagging, semantic analysis and subjective analysis [7]. Take in consideration that some opinions are expressed using either MSA which is the formal language, or Arabic dialects which are the informal language used in social media, or a combination of both; when preparing the data for analysis [8].

Some issues that ASA faces are in the linguistic and the contextual levels. In the linguistics level, the diacritic marks cause problems. The particular reason for this is that most of the words do not include them, besides the difficulty of the Arabic language morphology. In the contextual level, some negation words appear in the sentence which causes inaccurate results in word polarity. Moreover, some words appear in a domain that changes the polarity depending on the domain in which they are. In addition, some opinions are expressed in a sarcastic way, which causes inaccuracies in the polarity of the word [9].

D. Anti-Islam, Anti-Muslims and Islamophobia

Anti-Islam and anti-Muslims can be expressed as the hatred toward the Islam and Muslims, especially as a political force that promotes terrorism. In particular, it is a criticism of Islam, its actions, its nature, and what it teaches people [10]. Anti-Islam may involve Islamophobia, which is the fear and hatred against the Islamic religion and Muslims.

[2] https://www.internetworldstats.com/stats7.htm

The term Islamophobia has existed around the late 1980s and the early 1990s started in the United Kingdom to express the rejection against the Muslims who live in the West [11]. Some people want to harm and weaken the Islamic religion; therefore, they spread different content that contains wrong information to create confusion and a phobia against the Muslim faith. Some content can contain threatening, blaming, and labelling, which is known as toxic language. Some users write this toxic content without realizing its consequences. Unfortunately, some people label Muslims and Islam based on what they see or hear from the news, people or social media, even if this toxic content is considered as fake news and hate speech, and it has nothing to do with the Islamic teaching.

II. Related Work

A. Arabic Fake News Detection

Detection of Arabic fake news is considered a rough task as the Arabic sentiment resources are limited as well as the corpora and lexicon of Arabic language. Therefore, some researchers [6], [7], [8], [9], [12] and [13] have spent some time trying to solve these issues and implement fake news detection for Arabic texts. Alkhair et al. [12] classified Arabic YouTube comments into rumor and not-rumor. They use YouTube API to collect their dataset which consist of more than 4000 Arabic comments. They built three supervised machine learning algorithms for classification, namely SVM, MNB and Decision Trees. The accuracy of their systems differs according to the topic and the used classifier.

Almerekhi and Elsayed [6] were mainly interested in classifying the Arabic tweets as either automated or manual. They proposed four categories of features: the first category is the formality features where it measures how formal a tweet is based on the emotion, diacritics and elongation. The second category is structural features where it considers the structure of the tweet such as the length of the tweet in terms of the total number of characters, the number of question marks, and the number of exclamation marks. The third category is tweet-specific features where it checks for the data associated with tweets such as retweets, replies, hashtags and URLs contained in the tweets. The last category is temporal features where it focuses on the posting nature such as the activity period on Twitter that checks the time period the account is being used and spreads out the velocity. The dataset was collected from Twitter and consists of 3500 randomly labeled Arabic tweets that contain different dialects including Egyptian, Gulf and other different Arabic dialects. The model has an accuracy up to 92% and has classified 2000 automated Arabic tweets and 1500 manual tweets which shows that most of the Arabic tweets are automatically generated.

Moreover, Penuela [7] has worked on Arabic tweets as well as news headlines. The proposed system uses ML algorithms to classify the Arabic tweets and news headlines into true or deceptive messages. The author used two datasets, the first one consists of 1444 news headlines, 679 of them were true news and 765 are false news; and the second dataset consists of 532 Twitter messages, 259 of them were true tweets and 273 are false tweets. He also used hashtags, user mentions, emojis for the features in Twitter data along with the number of words in the document. Data frame used was for both datasets that contain the bag of words to train the classifier. The obtained results of the F-score of the News and Twitter datasets are 0.70 and 0.77, respectively.

Jardaneh et al. [8] presented a supervised ML model for classifying Arabic tweets based on the credibility of the tweet containing only honest, high-quality news and information. The authors extracted 45 features for each tweet and categorized them into two categories which are content-based features with 26 features extracted from the content and user-based features which are extracted from the profiles of users. The dataset is taken from publicly available dataset named Arabic Corpora for Credibility Analysis that consists of 1862 tweets about the Syrian crisis; divided into two classes: 1051 credible tweets and 810 non-credible tweets and some tweets are excluded because they became unavailable. They used four supervised ML algorithms to compare between them and work with the one that gives the best results; these algorithms are Random Forest, Decision Tree, AdaBoost, and Logistic Regression. The results showed that their system can classify non-credible tweets with an accuracy of 76%.

Bouchlaghem et al. [9] focused on sentiment analysis for MSA on a dataset consisting of Twitter posts. They used several sentiment features including lexicon features, linguistic features and sentence specific features. In addition, they used Tweet specific features as in papers [6] and [7]. They present different supervised ML algorithms for classification including SVM, k-Nearest Neighbor (k-NN), Naive Bayes, Decision Trees, Random Forest. The experimental results showed that the SVM algorithm has a better F-score with 70.64% for classifying Arabic sentiments in Twitter. The second classifier that has an F-score close to the SVM is the Naive Bayes with 70.02%.

Nagoudi et al. [13] implemented two detection methods, the first one for manipulated text detection and the second one for fake news detection. In manipulated text detection, they used two datasets: Arabic Treebank and A New Large-Scale Arabic News Dataset (AraNews) which they collected from different topics and sources. They proposed a method for automatic manipulation of texts and applied it on the AraNews dataset to produce a dataset of manipulated Arabic news. In fake news detection, they used external human-crafted fake news dataset which is a public dataset. They used crowdsourcing to determine the true and the false claims from the title. The experimental results showed that the detection of the manipulated Arabic news achieved good results on Arabic fake news detection where the F-score reached up to 70.06%.

B. Arabic Hate Speech Detection

Faris et al. [14] proposed a deep learning approach to detect and classify hate speech in the Arabic region. Their dataset consists of 3696 tweets without any duplicates and removed any irrelevant tweet. The content of the tweets is related to hate expressions in different topics in the Arabic region. Their approach is based on a word embedding features with a hybrid model of convolutional neural network (CNN) and long short-term memory (LSTM) network. For the preprocessing stage, they have deleted all the non-Arabic characters and stop-words, punctuation, hashtags, numbers, symbols, web addresses and diacritics. Moreover, they have tokenized the Arabic words and

also implemented some of the Arabic normalization techniques such as converting any variant of the Arabic Alif letter إ or أ or آ into ا and any ي into ى. For text vectorization, they have used the Word2Vec word embedding model. Their approach classifies tweets as Hate or Normal in terms of accuracy, precision, recall, and F1 measure. The experimental results showed that the AraVec word embedding approach with the recurrent convolutional networks produced good results with 66.564% accuracy.

Omar et al. [15] proposed a standard Arabic dataset that can be used for hate speech and abuse detection. The dataset was collected from more than one platform including Facebook, Twitter, Instagram, and YouTube. The dataset contains 20,000 posts or comments that were labeled manually by three Arabic annotators into two balanced classes, which are hate and not hate labels. Their preprocessing techniques include removing non-Arabic characters, emoji, or URLs, and additionally removing text containing less than two words because it is not necessary and it increases the dataset size. They have tested the dataset performance using twelve machine learning algorithms including MultinomialNB, LinearSVC, LogisticRegression and Decision Tree, in addition to two deep learning architectures namely the CNN, and the RNN. Their experimental results showed that the Complement NB produced the best result compared to the other ML algorithms with accuracy up to 97.59%, while the accuracy for the deep learning algorithm is 98.70% achieved by RNN which make it the highest performance achieved in both machine learning and deep learning.

Husain [16] proposed two approaches; one is based on the ML approach, and the second one is based on the ensemble ML approach, to detect and classify the offensive Arabic language. The ensemble ML classifier combines the prediction of different ML models in order to produce better performance. He has used three models called bagging, random forest, and AdaBoost. He used different preprocessing techniques including converting the emojis to written text in English language then translating it into the Arabic language, and removed the emoji. In addition, he has normalized some of the Arabic dialects and some of the Arabic letters. Furthermore, he removed numbers, symbols, HTML tags and double spaces. For the feature extraction, he used the TF-IDF on the n-gram of 1-2 words and 2-5 characters. The experimental results showed that for the ML models, the SVM produces the highest F1 score results with 82%; after that comes the logistic regression with 81%. For the ensemble ML models, the bagging produces the highest F1 score results with 88%.

Omar et al. [17] proposed a multi-labeled short Arabic text to classify the content into eleven balanced classes including politics, economics, religion and sports. They have found a relationship between hate speech and the different topics in social media; most of the hate speech is shown in the political posts, followed by sports and some of the economic posts. Their dataset consists of 44000 posts and tweets collected from Facebook and Twitter containing eleven topics. They used common preprocessing techniques such as tokenizing, stemming, removing URL and emojis, in addition to removing the diacritics, Tatweel and removing characters that appear more than one time. For the feature extraction, they have used three techniques N-gram, bag of words and TF-IDF with nine ML algorithms including MultinomialNB, LinearSVC, LogisticRegression and Decision Tree to evaluate the classifier with the best performance. The classifier with the best performance was the LinearSVC classifier with N-gram (1, 2) with accuracy score of 97,92%. Moreover, they have built a dataset for Arabic vulgar speech consisting of 6,000 posts; each comment is manually labeled as hate or non-hate speech.

Vidgen and Yasseri [1] proposed a multi-class classifier to detect and classify Islamophobic text on social media where it classifies the document into weak Islamophobic, strong Islamophobic and non-Islamophobic content. They have collected manually their dataset, which consists of 140 million tweets taken from Twitter. The used input feature is a gloVe word embeddings model that is trained on their collected dataset; they did not mention anything about the used preprocessing techniques. They have tested the model on six different algorithms namely Naïve-Bayes, Random Forests, Logistic Regression, Decision Trees, SVM and Deep Learning. All the algorithms produced good results ranging from 61.23% to 72.17%, but the best was achieved by SVM with 72.17% followed by Deep Learning with 71.14%. For the classification, they have used cross-validation on the SVM model. The model achieved good results with 77.6% accuracy score and 83% balanced accuracy score.

III. Proposed Framework

The framework of the proposed methodology consists of four stages as shown in Fig. 2.

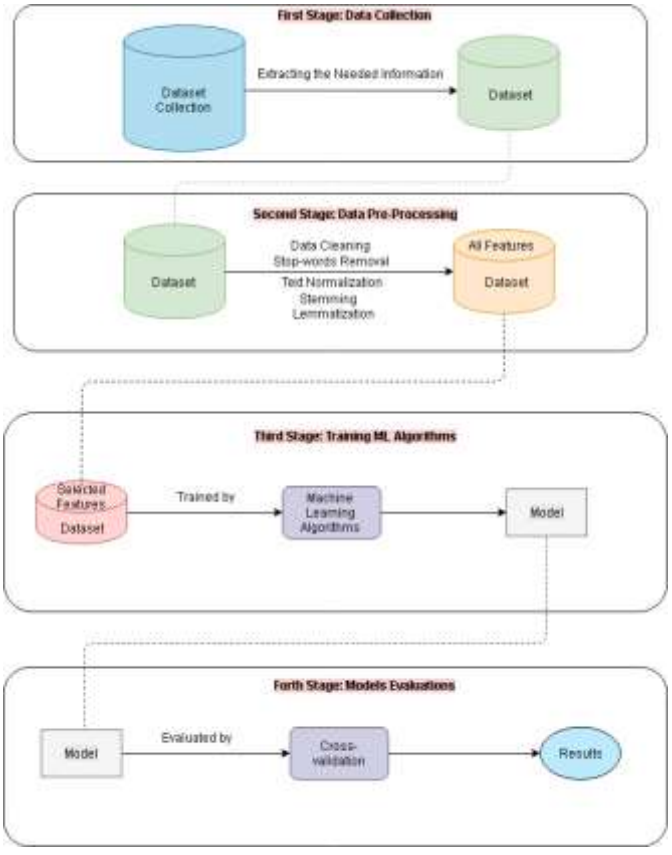

Fig. 2. Framework of the Proposed Approach.

The first stage consists of data collection, where the data is collected from different Arabic sources into two different datasets, each dataset contains an anti-Islam text content and a not anti-Islam text content. The second stage involves data pre-processing, where the data is prepared for processing and selects the features to be used in the next stage. The third stage is the process of training the ML models, where the ML algorithm is provided with the training data to learn from. The last stage is the evaluation of the ML models. The models used in this research are based on a supervised ML approach using SVM and MNB algorithms.

A. Data Collection and Annotations

The ambition is to create a general benchmark dataset containing a huge dataset for anti-Islamic web text content. The collected data were from articles, journals and some of them are from personal blogs. The main reason for choosing these types of data is because the focus was in the formal language used in the academic writing content and not informal language used in social media. The data was gathered from the Internet using Yahoo and Google search engines. Furthermore, the MSA text content was gathered, which is the formal Arabic language instead of the informal Arabic dialects. The collection of the Arabic data started from the end of February 2021 until the mid of April 2021.

The main keywords used to collect data in Arabic language were: معاداة الإسلام، مناهضة الإسلام، الإساءة للرسول، كره الإسلام، محاربة الإسلام، الإنفصالية الإسلاموية، اضطهاد المرأة، الشمولية الإسلاموية والتطرف الإسلاموي.

These keywords helped us to reduce the amount of search, in order to find the desired content, due to the huge number of articles that talk about Islam in good or in bad ways. Two datasets for the Arabic language were produced.

The collected data is organized into an excel spreadsheet using a web-scraping tool called Octoparse. This tool takes the URL of the webpage to extract data from, then selects the target data to be extracted, and runs the scraping to get the data as CSV, Excel, Application Programming Interface (API), or save them to a database. The extracted data contain the title, the content, the URL, the date, and label them as an anti-Islamic content or not. During the process of collecting data, one challenge was the extraction and the retrieval of the blocked webpages containing extremist ideas or false information about Islam from Saudi Arabia search engines.

B. Data Preprocessing

Preprocessing is considered an important stage in the process of preparing the data to be used. This stage includes different techniques such as stop-words removal, normalization, stemming and lemmatization.

Stop-words are a list of the most used words in a language. This list is different for different languages and there are different public stop-word lists that can be used in NLP. Stop-words can be safely removed without changing the meaning of the text. In English language, some stop-words are: the, is, in, on, at, which, and of …, whereas in Arabic language some stop-words are: الذي، إن، أنا، أنت، أنتم and إلى.

Text normalization includes different techniques to end up with a clean corpus that can be used in the classification process. If the text contains numbers, there are different ways to deal with them; either keep them as they are, or remove them using regular expressions, or convert them into words that can be used. Normalization also includes removing punctuations and white spaces, which are the starting and the ending spaces in the text. Furthermore, tokenization is the good way to normalize the text, which is the process of dividing the text into smaller parts known as a token.

Stemming removes the last characters (suffixes) and/or the beginning characters (prefixes) in the word to return the word into its stem or root. This process can lead to incorrect words in the language. This technique is one of the most useful and effective techniques in NLP. Stemming is used in the classification task to reduce the high dimensionality of the document and increase the functioning of the classifier especially in difficult languages such as Arabic language. Some of the Arabic stemming are: Information Science Research Institute Stemmer (ISRI Stemmer), and Arabic light Stemmer (ARLSTem) (both are included in NLTK library).

Lemmatization groups together the inflected forms of the word to be analyzed as a single element, specified by the lemma or the dictionary form of the word. This technique produces more accurate results than stemming technique. The meaning of the text is preserved as it takes into account the context of the words. However, using this technique requires lots of computation and deep knowledge about the morphology of the language.

C. Feature Selection and Classification Process

Feature extraction enables us to convert unstructured text into a structured feature, so that it can be used in the classification process, which requires mathematical modeling for working.

Classification process is considered a critical step in building the right model in text classification where the text can be automatically classified into one or more defined categories. There are different algorithms that can be used in this step but to obtain good results, the size of the dataset should be taken into consideration. If the dataset is large it is best to use deep learning, but if the dataset size is relatively small it is better to use ML algorithms.

One of the most used and accurate ML algorithms is SVM. SVM is one of several supervised learning algorithms used in text classification; the algorithm classifies a given document based on some selected features into one of the pre-labeled categories. The reason for choosing this algorithm is because the dataset consists of almost 9000 data, that is relatively small and hence this algorithm is the most appropriate for this situation. In addition, it needs less data for training the model, which is suitable for the dataset to produce accurate and fast results.

Naive Bayes classifier is also a supervised ML algorithm used for classification; the algorithm uses Bayes' theorem where it computes the conditional probabilities of the occurrence of two events based on the probabilities of each individual event. Naive Bayes has different members such as

Gaussian, Bernoulli and Multinomial, and one of the best members that produce good results is MNB. The reason for choosing this algorithm is because it is the second most suitable algorithm for the datasets (as we will prove it later), this algorithm does not require much computation for classification. In addition, this algorithm works well with small to medium datasets, and it produces accurate results.

D. Model Evaluation

In the model evaluation phase, the model is tested on unseen dataset to evaluate how well the ML model works on these new dataset. The performance of the model can be estimated using two techniques: Holdout and Cross-Validation. In holdout evaluation, the dataset is randomly divided into three subsets: training, validation and testing. Training set is a subset of the dataset used to build the model. Validation set is the subset of the dataset used to evaluate the performance of the model. Testing set is an unseen dataset that can be used to test the future performance of the model.

Regarding cross-validation technique, it divides the dataset into a training set to train the model and an independent predefined set used to evaluate the performance of the model. One of the cross-validation techniques is k-fold cross-validation, where the dataset is divided into k equal size such as 5 or 10 folds. This process is repeated k times, where most of the data are used in the test set. Holdout approach is a simple and fast approach, but it has high variability that causes differences in accuracy. However, cross-validation reduces bias and variance because most of the data are used in the test set.

In the model evaluation, there are various metrics that can be used to measure model performance. Some of the classification metrics are: Classification Accuracy, Confusion Matrix, Logarithmic Loss, Area Under the Curve (AUC) and F-measure.

Classification accuracy is the ratio of all the correct predictions done by the model. Confusion matrix shows the true positive and the true negative (correctly predicted as positive and negative), the false positive and the false negative (incorrectly predicted as positive and negative). Logarithmic loss measures the performance of the model as a probability value between 0 and 1, where the optimal model achieves log loss of 0. AUC is used when a binary classifier can differentiate between the two classes, the curve of the optimal result achieved by the classifier will be along the Y axis and then along the X axis. F-measure known as F-score, it measures the accuracy taking into account the precision and the recall of the test to compute the score.

IV. Implementation

A. General Information about the Datasets

The dataset is the core of any classification model to evaluate the performance of the proposed approach. Therefore, two datasets were collected consisting of different numbers of data. The two datasets contain Arabic text (non-balanced and balanced datasets). The balanced Arabic dataset consists of 6142 articles and 1038 words per article on average. The non-balanced Arabic dataset is made up of long articles containing 8510 articles, and 879 words per article on average. The maximum number of words in the longest article is 6605 words in both datasets; and the minimum number of words in the shortest article is one word in both datasets. The maximum number of words in the anti-Islamic articles is 6108, whereas for the non-anti-Islamic articles is 6605.

Fig. 3 illustrates the information given above about the two datasets, and Fig. 4 illustrates the information given above for only the anti-Islamic contents in the datasets.

B. Data Preparation and Preprocessing

After collecting the data that is related to the topic, some preprocessing techniques were performed to keep only the necessary information. Some of the preprocessing techniques are removing punctuation, removing whitespaces and replacing some characters with others such as phone number with the words number. Moreover, the Arabic stop-words are removed. Removing those words in the dataset will produce a smaller dataset which will help in speeding up the process of classifying the documents.

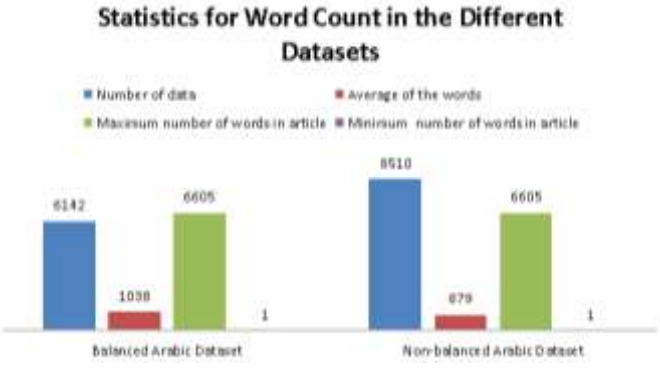

Fig. 3. Statistics about the Two Datasets.

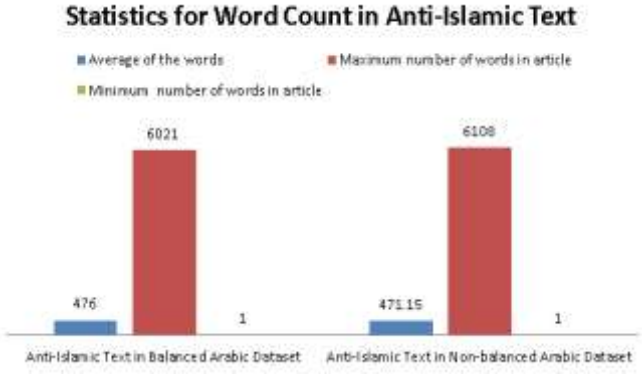

Fig. 4. Statistics about the Anti-Islamic Contents in the Datasets.

Fig. 5 illustrates a sample of the Arabic dataset before the preprocessing phase.

Fig. 5. Arabic Dataset before the Preprocessing Phase.

Furthermore, Arabic stemming is used which return the Arabic words into their root based on some Arabic language rules. However, in some words, when the suffixes and/or the prefixes are removed, the result can produce a word that is not in fact a word in the Arabic language.

For the Arabic language stemming, the ISRI Stemmer is used, which is a rule-based stemmer that stems the word based on some rules to return the word to its root [18].

Fig. 6 shows some preprocessing techniques used to handle the Arabic language text. These techniques include removing Tatweel or Kasheeda, which refers to the elongation character "-" so as to create justification in Arabic language; removing diacritics which are the Harakat, that represent the short vowel marks in Arabic language, the small letters, and the Tashkeel which are the supplementary diacritics used as phonetic guides marks in Arabic language. Moreover, the Arabic punctuations are removed.

```
text = strip_tashkeel(text)
text = strip_diacritics(text)
text = strip_tatweel(text)
```

Fig. 6. Some Arabic Preprocessing Techniques.

C. Feature Selection

We have used TF-IDF with word level, where it is considered the frequency of a single word in the dataset. Moreover, the TF-IDF with N-gram is used, which is a model that depends on the sequence of words with a predefined length N to predict the next word. In the experiment, the tri-gram word-based model is used, where it is considered the frequency of three words in the dataset.

D. Training Process

The non-balanced Arabic dataset was tested on six different algorithms based on the different related work we discussed previously, in order to select the best two classifiers that produce good results based on the dataset. The different classification algorithms we have tested are: Decision Tree, k-NN, Random Forest, Logistic Regression, MNB and SVM classifiers.

Table I shows that the six different algorithms produce good results except the k-NN; with accuracy ranging from 90.601% to 97.274%. However, the SVM classifier outperforms the other five classifiers followed by the MNB classifier. The accuracy of SVM is the highest with 97.274% while the MNB accuracy is 96.193%, which is less than the SVM classifier by 1.081 points.

Due to the results of the above comparison between the six different classification algorithms, the SVM and the MNB algorithms are selected to be used for defining the ML model in order to achieve the goal in detecting and classifying the anti-Islamic content.

The dataset is divided into training and testing sets. The training data is used to train the models. In addition, we used the testing data to make sure that the trained model performs well for the hidden data. The data is split into 70% for training data and 30% for testing data used in the end when the training of the model is completed.

TABLE I. TESTING THE DIFFERENT CLASSIFIERS

Algorithm	Accuracy
k-NN classifier	64.661%
Decision Tree classifier	90.601%
Random Forest classifier	92.105%
Logistic Regression classifier	95.864%
MNB classifier	96.193%
SVM classifier	97.274%

E. Overcoming the Problems of Data Leakage and Harm

We used TF-IDF after splitting the datasets into training and testing sets, to ensure that no information is shared between the two sets. This is considered as a big problem and it is called data leakage, which means that the data in the training and testing are accidentally shared. To overcome data leakage problems, different techniques are used to minimize it during the process of building the model. These techniques include splitting the datasets into training and testing before using TF-IDF, pipeline architectures, ten folds cross-validation and testing the model using unseen validation dataset.

Another problem arises in these types of classification is that sometimes the classifier can cause harm instead of reducing it during the process of classification [19]. This problem can happen when the text contains racial bias or minority populations; in our case, *women* and *hijab* themes are considered kinds of harm. Moreover, this problem can be caused by different problems in the training data, labels or even the resources used in the model [20]. Unfortunately, there are no general solutions for this problem, but the model can be evaluated on different datasets with different topics [21].

V. RESULT, EVALUATION AND DISCUSSION

A. Tri-gram Level TF-IDF

Table II and Table III list the different results when using the tri-gram for the two classifiers on the two different datasets. The observation can show that the results have no significant change. The difference between the results obtained by the two classifiers is one percent. The accuracy obtained using the MNB classifier is 89% compared to 88% obtained with the SVM classifier.

The experimental results show that for the Arabic language, the highest accuracy is achieved by the ML approach, using MNB on a non-balanced Arabic dataset with tri-gram level TF-IDF as feature extraction, with an accuracy of 89%.

TABLE II. RESULTS FOR TRI-GRAM ON NON-BALANCED DATASET

Non-balanced	*Precision*	*Recall*	*F1 score*	*Accuracy*
TF-IDF with SVM	89%	88%	88%	88%
TF-IDF with MNB	90%	89%	89%	89%

TABLE III. RESULTS FOR TRI-GRAM ON BALANCED DATASET

Balanced	*Precision*	*Recall*	*F1 score*	*Accuracy*
TF-IDF with SVM	87%	87%	87%	87%
TF-IDF with MNB	86%	70%	72%	70%

Table IV and Table V list the precision, the recall and the F1 score for the negative articles on the different datasets. The results show that the overall values of the precision and the F1 score concerning the non-balanced datasets achieve the best results compared to the balanced datasets. However, the recall is higher in the balanced datasets compared to the non-balanced one.

TABLE IV. RESULTS FOR NON-BALANCED NEGATIVE ARTICLES DATASET USING TRI-GRAM

Dataset Type	*Non- balanced*	*Precision*	*Recall*	*F1 score*
Arabic dataset (Negative Articles)	TF-IDF with SVM	93%	89%	91%
	TF-IDF with MNB	96%	88%	92%

TABLE V. RESULTS FOR BALANCED NEGATIVE ARTICLES DATASET USING TRI-GRAM

Dataset Type	*Balanced*	*Precision*	*Recall*	*F1 score*
Arabic dataset (Negative Articles)	TF-IDF with SVM	88%	87%	87%
	TF-IDF with MNB	43%	97%	59%

B. Word Level TF-IDF

Fig. 7 shows the confusion matrix using ML model with non-balanced Arabic dataset on word level, whereas Fig. 8 shows the confusion matrix using ML model with non-balanced Arabic dataset on tri-gram level with the same classifier. When TF-IDF is used, the True Positive (TP), which is the number of correct predictions, is 1617 for the ML model on word level with non-balanced Arabic dataset, and on tri-gram level with non-balanced Arabic dataset the number of correct predictions is 1573. For the True Negative (TN), which is the correct predictions for the negative class, the model on word level produces 860, and the model on tri-gram level achieves 717 correct predictions. For the False Positive (FP), the number of predictions in the model on word level is 33, and in the model on tri-gram level is 186. For the False Negative (FN), which is the false prediction of the negative class, the model on word level produces 43, and the model on tri-gram level achieves 113 negative predictions.

Table VI and Table VII list the different results when using the word level for the two classifiers on the two different datasets. The results show that for all the measurements: the precision, the recall, the F1 score and the accuracy are the same for each algorithm. However, the comparisons between the two classifiers (SVM and MNB), on the balanced and a non-balanced Arabic datasets show a small change in results.

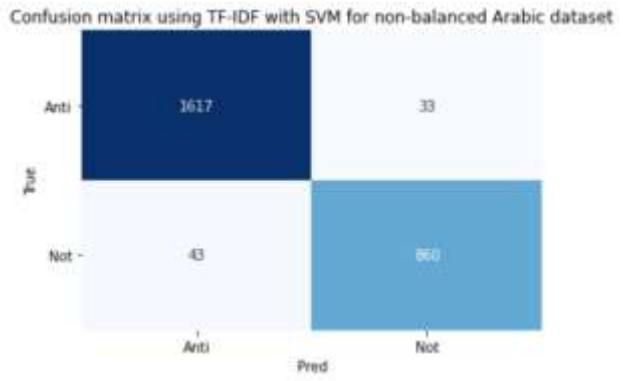

Fig. 7. Confusion Matrix using Word Level TF-IDF with SVM for Non-Balanced Arabic Dataset.

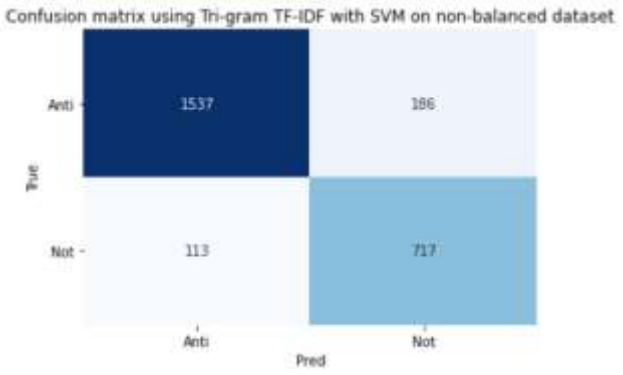

Fig. 8. Confusion Matrix using Tri-Gram Level TF-IDF with SVM for Non-Balanced Arabic Dataset.

TABLE VI. RESULTS FOR WORD LEVEL ON NON-BALANCED DATASET

Non- balanced	*Precision*	*Recall*	*F1 score*	*Accuracy*
TF-IDF with SVM	97%	97%	97%	97%
TF-IDF with MNB	95%	95%	95%	95%

TABLE VII. RESULTS FOR WORD LEVEL ON BALANCED DATASET

Balanced	*Precision*	*Recall*	*F1 score*	*Accuracy*
TF-IDF with SVM	97%	97%	97%	97%
TF-IDF with MNB	83%	75%	73%	75%

Table VIII and Table IX list the precision, the recall and the F1 score for the negative articles on the different datasets using word level. The results show that the overall values of the precision, the recall and the F1 score concerning the non-balanced dataset achieve the best results compared to the balanced dataset.

TABLE VIII. RESULTS FOR NON-BALANCED NEGATIVE ARTICLES DATASET USING WORD LEVEL

Non-balanced	*Precision*	*Recall*	*F1 score*
TF-IDF with SVM	97%	98%	98%
TF-IDF with MNB	98%	94%	96%

TABLE IX. RESULTS FOR BALANCED NEGATIVE ARTICLES DATASET USING WORD LEVEL

Balanced	*Precision*	*Recall*	*F1 score*
TF-IDF with SVM	96%	97%	97%
TF-IDF with MNB	100%	51%	67%

C. Discussion

A detailed description is given about all the experimental results applied to the datasets and achieved by the proposed two classifiers approach using the feature extraction techniques, namely word level TF-IDF and Tri-gram level TF-IDF.

The experimental results using our approach with different datasets (Arabic balanced and Arabic non-balanced), showed that the best algorithm producing high accuracy was SVM with word level TF-IDF as feature extraction. Therefore, almost there is no matter regarding if the datasets are balanced or not except for the tri-gram on a non-balanced dataset.

In addition, the results demonstrated that the SVM was the best classifier in terms of accuracy, and it outperforms the MNB classifier in almost all experiments.

VI. WEB-APPLICATION PROTOTYPE

We have developed and built an interactive web-application prototype using the Streamlit framework in python. In the homepage of the web application (Fig. 9), you can choose between the two proposed datasets. Furthermore, there are two proposed classifier models (SVM or MNB) to choose from. In addition, you also have the ability to choose at the N-gram level (word level or tri-gram level), in order to finally test and predict the category of the entered text if it contains an anti-Islamic content or not.

Fig. 10 illustrates an example of a classification process result. The LIME library is used to explain predictions of a given text. LimeTextExplainer helps in explaining the predictions of a trained model to categorize sentences on any given area. Fig. 10 below shows the result of the entered text as not anti-Islamic content associated with their probability, followed by the LIME explanation.

Fig. 9. Web-Application Homepage.

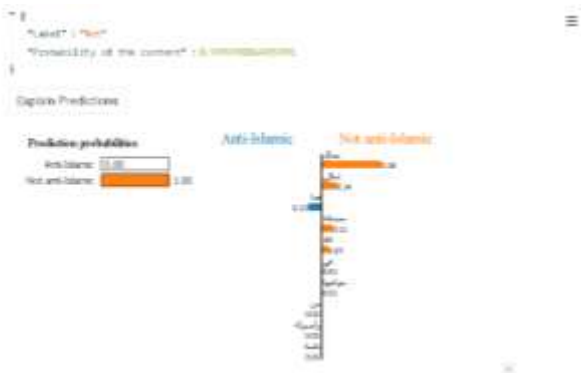

Fig. 10. Results of the Classification Process.

VII. CONCLUSION AND FUTURE WORK

We have proposed an anti-Islamic Arabic text categorization framework using text mining and sentiment analysis techniques. This framework will help us to identify and classify the text content of different webpages into anti-Islamic content or not anti-Islamic content; and to increase awareness toward these kinds of toxic contents that promote hate. Proper datasets have been collected and used in this framework to classify the anti-Islamic web text content; also, the features that can be used for anti-Islamic toxic language texts have been identified.

The models used in this research are based on supervised ML approaches using SVM and MNB algorithms. The experimental results showed that for the datasets, the best algorithm that produced high accuracy with 97% applied on the balanced Arabic dataset using SVM algorithm with word level TF-IDF as feature extraction. In addition, the results demonstrated that the SVM was the best classifier in terms of accuracy, and it outperforms the MNB classifier in almost all experiments.

We have faced different challenges during the process of achieving our goals such as the absence of a dataset that contains anti-Islamic content in Arabic. In addition, a number of webpages that promote hate or spread false information about Islam were blocked, and we were not able to reach them from Saudi Arabia. This slowed down the process of collecting and gathering the data and made it harder to find different webpages that contain this kind of information. Another encountered issue faced in this research was the lack of an efficient Arabic preprocessing library that supports us to accomplish some tasks such as lemmatization.

In the future, more data will be added to the datasets in order to explore the use of a deep learning approach. We propose to implement a translation-based approach to deal with different languages other than Arabic in order to overcome the lack of datasets in the respective language. Furthermore, the ontology will be taken into account to encode the knowledge in this domain into a graph in order to improve the accuracy of the classification. Another research area is to explore different social media contents on which can be collected and accumulate data, to deal with the Arabic dialects, which are informal languages; and compare their contents with the MSA datasets, which contain formal language, and notice what the experiment's results will show.

REFERENCES

[1] B. Vidgen and T. Yasseri, "Detecting weak and strong Islamophobic hate speech on social media," arXiv, pp. 1–6, 2018.

[2] R. A. Alraddadi and M. I. E.K. Ghembaza, "Automatic Detection and Classification of Anti-Islamic Web Text-Contents," 7th EAI International Conference on Interactive Digital Media (EAI ICIDM 2021), Proc., Conf., 30th July-1st August 2021.

[3] K. C. Kavakli and P. M. Kuhn, Dangerous Contenders: Election Monitors, Islamic Opposition Parties, and Terrorism, vol. 74, no. 1. 2020.

[4] R. Agrawal and M. Batra, "A Detailed Study on Text Mining Techniques," Int. J. Soft Comput. Eng., no. 26, pp. 2231–2307, 2013.

[5] K. Kowsari, K. J. Meimandi, M. Heidarysafa, S. Mendu, L. Barnes, and D. Brown, "Text classification algorithms: A survey," Inf., vol. 10, no. 4, pp. 1–68, 2019.

[6] H. Almerekhi and T. Elsayed, "Detecting automatically-generated Arabic tweets," Lect. Notes Comput. Sci. (including Subser. Lect. Notes Artif. Intell. Lect. Notes Bioinformatics), vol. 9460, no. January, pp. 123–134, 2015.

[7] F. Javier Fernaandez-Bravo Penuela, "Deception detection in Arabic tweets and news," CEUR Workshop Proc., vol. 2517, no. December, pp. 122–126, 2019.

[8] G. Jardaneh, H. Abdelhaq, M. Buzz, and D. Johnson, "Classifying Arabic tweets based on credibility using content and user features," 2019 IEEE Jordan Int. Jt. Conf. Electr. Eng. Inf. Technol. JEEIT 2019 - Proc., no. 1, pp. 596–601, 2019.

[9] R. Bouchlaghem, A. Elkhelifi, and R. Faiz, "A machine learning approach for classifying sentiments in Arabic tweets," ACM Int. Conf. Proceeding Ser., vol. 13-15-June, 2016.

[10] C. Froio, "Race, religion, or culture? Framing Islam between racism and neo-racism in the online network of the French far right," Perspect. Polit., vol. 16, no. 3, pp. 696–709, 2018.

[11] F. B. López, "Towards a definition of Islamophobia: Approximations of the early twentieth century," Ethn. Racial Stud., vol. 34, no. 4, pp. 556–573, 2011.

[12] M. Alkhair, K. Meftouh, K. Smaïli, and N. Othman, "An Arabic Corpus of Fake News: Collection, Analysis and Classification," Commun. Comput. Inf. Sci., vol. 1108, pp. 292–302, 2019.

[13] E. M. B. Nagoudi, A. R. Elmadany, M. Abdul-Mageed, T. Alhindi, and H. Cavusoglu, "Machine generation and detection of arabic manipulated and fake news," arXiv, pp. 1–15, 2020.

[14] 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 2020 - Proc. 9th Int. Conf. Pattern Recognit. Appl. Methods, no. March, pp. 453–460, 2020.

[15] A. Omar, T. Mahamoud, and T. Abd-el-hafeez, "Comparative Performance of Machine Learning and Deep Learning Algorithms for Arabic Hate Speech Detection in OSNs," Proc. Int. Conf. Artif. Intell. Comput. Vis., vol. 1, no. 1153, pp. 159–169, 2020.

[16] F. Husain, "Arabic Offensive Language Detection Using Machine Learning and Ensemble Machine Learning Approaches," arXiv preprint arXiv:2005.08946,2020.

[17] A. Omar, A. Omar, T. M. Mahmoud, T. Abd-el-hafeez, and A. Mahfouz, "Multi-label Arabic text classification in Online Social Networks Multi-label Arabic text classification in Online Social Networks," Inf. Syst., vol. 100, no. April, p. 101785, 2021.

[18] Y. A. Alhaj, J. Xiang, D. Zhao, M. A. A. Al-Qaness, M. Abd Elaziz, and A. Dahou, "A Study of the Effects of Stemming Strategies on Arabic Document Classification," IEEE Access, vol. 7, pp. 32664–32671, 2019.

[19] M. Sap, D. Card, S. Gabriel, Y. Choi, and N. A. Smith, "The risk of racial bias in hate speech detection," ACL 2019 - 57th Annu. Meet. Assoc. Comput. Linguist. Proc. Conf., pp. 1668–1678, 2020.

[20] V. Nandi and S. Agrawal, "Sentiment Analysis using Hybrid Approach," Int. Res. J. Eng. Technol., pp. 1621–1627, 2016.

[21] M. Mitchell et al., "Model cards for model reporting," FAT* 2019 - Proc. 2019 Conf. Fairness, Accountability, Transpar., no. Figure 2, pp. 220–229, 2019.