(Online) ISSN 2744-1741
Defense and Security Studies *Original Research*
Vol. 7, No. 1, 2026, pp.106-122
https://doi.org/10.37868/dss.v7.id316

A robust feature engineering approach for Arabic extremist content detection in social media

Ahmed Salman Ibraheem[*1]
[1]Department of Cyber Security, Imam Alkadhum College (IKC), Iraq

*Corresponding author E-mail: ahmed100@iku.edu.iq

Received: Feb. 19, 2026
Revised: Mar. 10, 20265
Accepted: Mar. 11, 2026
Online: Mar. 12, 2026

Abstract

In this paper, we propose an end-to-end machine learning system to analyze sentiment polarity in extremist (terrorism-related) Arabic content with novel designed features concentrating on linguistic discourse, properties and changes of such contents. We constructed three corpora (V1, V2 and V3) from Arabic tweets; which have been pre-processed by using various linguistic techniques: Normal stemmer, root pattern, Light Stemming. We have employed various machines' learning algorithms such as SVMs, NB and KNN with BOW and ngr am models to retrieve features. Our large scale comparative analysis based on a real-dataset benchmark chose linear SVM and Uni-gram model in conjunction with Term Frequency-inverse document Frequency (TF-IDF) as the preferable choice. Our approach achieved better accuracy for extremist sentiment detection and greater Recall in V1 (81.097%) and V2 (81.707%) compared to this setup. These ones were superior to other combination of SVM kernels along with the KNN algorithm that also was very competitive. Our findings outperformed the already established approach (Kanan & Fox) for classifying extremist Arabic texts (our BEA as an average achieved accuracy rate higher than their 78.00% but using P-Stemmer and SVM). The precision-recall and ROC AUC values for SVM settings also reinforced the performance, and high scores reflected its ability to handle complex features of Arabic like syllabic lengthening and diacritics. The present study demonstrates the potential applicability of this approach to enhanced supporting extremism detection analysis in Arabic textual data, and may offer a clearer perspective for those concerned on security, education and policy making domains.

Keywords: Arabic extremist content, KNN, machine learning, N-grams, SVM, TF-IDF

1. Introduction

advances rapidly, having direct effects that reflects from traditional to digital and vice versa in life of the people, so there is a need of systems which can analyze data in real time manner for Decision Making over some specific problem. One such problem is the identification of extremist content, which requires substantial time and effort to fully perceive and deal with a variety of topics or situations [1]. The increasing use of social media

platforms has increased distribution of extremist propaganda, requiring strong automated tools for detection and analysis [2]. The goal of this work is to fill an existing gap in extremism detection under analysis and opinion mining, where little attention has been given to the problem so far, especially regarding Arabic texts which exhibit greater linguistic challenges by being morphologically rich and include more potential scare languages than English [3]. It presents a new Arabic sentiment lexicon specifically designed for detecting and analyzing extremist content, which facilitates our understanding of how extremism is framed in the Arab online community [4, 5].

The main aims of this study are illustrated in Figure 1, which present Machine learning techniques to tackle Arabic extremism accurately using T F and T F- ID F as features and also examine the impact for setting different values of n [6]. The present study targets techniques Support Vector Machine (SVM) [7], NaÃƒÆ'-ve Bayes (NB) and K–Nearest Neighbors (KNN) [8] that possesses individual responsibilities in the classification of the sentiments with regards to extremist ideas.

We choose SVM [9] as our classifier because of its discriminative property and the ability to directly learn the mapping between features in input datamining space to target, without relying on whatever underlying probability distribution of these data. Rather, NB is employed to insure the jpd between input and output i.e. how data is distributed [10].

We select K-nearest neighbors (KNN) [11] because it has the nature of instance-based and lazy learning, where the prediction or classification is based on similarity to the nearest neighbor(s) in feature space. This multifaceted approach is intended to enhance the precision of existing sentiment analysis techniques, and make a significant contribution to security and digital communication research.

Recent researches have shown that employing n-grams and TF-IDF to classifying Arabic text is a very effective technique by obtaining accurate results with respect to extremist content detection [12]. However, there are problems that stem from richness in Arabic morphology such as stemming and diacritics normalization, which is widely believed to have great impact on the performance of the classifiers as in Figure 1 [13]. Additionally, the ethical implications of analyzing extremist content must be carefully considered to avoid biases and ensure responsible research practices [14].

We must stand against Muslims in Sweden and demolish and burn all mosques and kill Muslims. The detection of this post is extremism	يجب الوقوف ضد المسلمون في السويد وهدم جميع المساجد وحرقها وقتل جميع المسلمون

Figure 1. Example of this study

The contributions of this research work can be stated as follows:

1. This research work focusses on improving Arabic text classification by leveraging TF-IDF with varied N-gram sequences, enhancing extremist detection.
2. We have constructed a balanced dataset with rigorous annotation and ethical considerations, ensuring reliable and responsible extremist content classification.
3. We have presented a comparative analysis of SVM, Naïve Bayes, and KNN demonstrated that SVM with a linear kernel achieved the highest accuracy.

The structure of this paper is as follows: Section 2 outlines the research background. Section 3 reviews related literature. Our proposed method is detailed in Section 4. Section 5 discusses experimental setup and results. Lastly, Section 6 concludes the paper and outlines future research directions.

2. Research background

The threat of extremism to academic institutions and national security represents a significant global concern [15]. This paper aims to identify articles that exhibit distinct polarities. Figure 2 provides an illustration of a brief article pertinent to our study. Given a set of articles denoted by A, the objective is to discern the author's perspective or other relevant information, as illustrated in Equation (1).

$$OpInion = f(P) \quad (1)$$

Here, the function used to determine P is denoted as f, where P represents the polarity. Suppose $A = \{A_i | \ A_i \text{ is an article}, i \in \mathbb{Z}^+\}$ and $L = \left\{\{L_j | L_j \text{ is label of article}\right\}$. L creates a partition on A such that $A_i \in L_j$ for some j, when $A_i \in L_j$ referred to A_i by A_i^j. Every article A_i is segmented into sentences, expressed as $A_i = \{s_1^i, s_2^i, s_3^i, \cdots, s_n^i\}$. Each sentence S_k^i, where $1 \leq k \leq n$, is further broken down into words $w_{k,r}^i$, with each word $w_{k,r}$ belonging to sentence k in article i and having a word count of r. The words comprising each of the sentences in the article are presented in Equation (2):

The prices of the neighbouring area of influence in its vicinity, one of its neighbouring areas, one of its neighbouring areas, one of the neighbouring areas of influence, and its stability in an adjacent area.	يا صديقي النفوذ الإيراني في المنطقة تجاوز ما احدثته معاهدة سايكس بيكو من حيث إن المعاهدة قسمت المنطقة إلى دول، ولكن النفوذ الإيراني فتت الدولة الواحدة وقام با أحياء كل الاثنيات والطائفية والنعرات الإقليمية نحو احتراب رتب موت كل عمليات التنمية في الوطن العربى وتذابح طائفي غير مسبوق

Figure 2. Example of a short article about this study

$$F_i = \bigcup w_{k,r}^i \, , where \; w_{k,r}^i \in A_i^j \quad (2)$$

In this study, extremism and non-extremism were selected as the two labels, where F_j is the label of the article and denotes the limited set of terms used to construct the vector applicable in this study. The subsequent section will discuss issues related to polarity.

3. Related works

The use of machine learning to identify Islamic extremism in Arabic text Some very advanced techniques are required for the analysis of these texts so as to deal with the linguistic complexity of Arabic. In this paper, we review the particular application of a robust feature extraction method such as Term-Frequency (TF) and TF-IDF, plus diverse n-gram sequences in machine learning-based systems for recognizing and exploring Arabic extremist texts. The following are some of the related studies to our study.

3.1. Term frequency and inverse document frequency (TF-IDF)

TF-IDF has widely been used a feature extraction technique in text classification [16]. It represents how important a word is in a document compared to other documents and minimize the value of frequent terms, increasing the im portance of uncommon ones. Alraddadi and Ghembaza [17] investigated categorization of anti-Islamic Arabic texts, in their research, by mean of text mining and sentiment analysis. It used methods such as TF-IDF and sentiment scoring of social media posts and comments. The results showed that the classification of negative sentiments towards Islam is done in an efficient way such as distinguishing between neutral and islamo-phobic was good accuracy..

3.2. N-Gram models

Single word analysis may lack the context in N-grams, which are n item sequences of a given sample of text. The unigram to trigram order n-gram model has been used over different n range for size of context window [18]. Rekik et al. [19] proposed a recursive approach of detecting the profile of an extremist in a social network. They unearth the extremist accounts time and again, by analyzing any suspect contacts or dangerous messages. They employed the textual analysis producing toxic vocabulary, with resources for itemset mining, N-grams and violence degree. The effectiveness of the method in detection for extremism detection in Twitter was confirmed with credible discrimination results..

3.3. Combination of TF-IDF with N-Grams

The representation of TF-IDF with n-grams allows to have a more sophisticated description of the text, treated according to context. Al-Harbi and Kamsina [20] combined TF-IDF and n-gram as features into a machine learning to identify the sentiment polarity of tweets on terrorism. The labeled dataset involved tweets that were

labelled as “positive”, “negative” or “neutral” in the context of terrorism. The performance of the classifier appeared to be in identifying different sentiments well; moreover it also performed reasonably with respect to retrieving terrorist tweets with a relatively high precision and recall. For example, Alshahrani [12] proposed a classification of Arabic extremist Web content and he applied NLP tools (e.g: TF-IDF, NGram) to analyze data set on collected Arabic web pages as well as forum posts. Approaches focus instead on finding linguistic characteristics of the Arabic language and extremist jargon. Our experimental findings demonstrate high level of descriptiveness and support the effectiveness of our proposed techniques in identifying extremist narrative in Arabic web space. The study of Asif et al. [21] employed TF-IDF and n-grams to perform sentiment analysis on extremism in social networks on a dataset where text sources come from multiple social platforms. These methodologies were used for feature extraction that will help in extremist sentiment recognition and classification. The accuracy of these approaches was demonstrated in the results and confirmed in their ability to distinguish extremist from nonextremist content.

In addition, limited work has been done to examine the performance of different machine learning models on TF-IDF and n-gram data. Furthermore, the influences of the n-gram sizes on classifier are significant. The larger n-grams (as explained in literature) can retain more context but might be noisier and resource dependent. There is however trade-off as in the work of Elgammal [22], where he combined models based on TF-IDF and n-gram to categorize sentiment from a Twitter corpus during Hajj. The attempted method was to label tweets as positive, negative or neutral for sentiment intensity of participants. The proposed method derives emotion based on the integration of these methods and differentiates emotional responses for Hajj, where positive emotions are mostly prevalent. [13], [23].

4. Proposed method

In this section, we present our methodology: to explore various machine learning methodologies in order to solve our specific problem. Our approach can be divided into several steps. We first select the corpus, Best feature extraction method (what are they),., What is a word embedding? and the machine learning model that gave the most accurate predictions. To close the gap, we are pruning our approach by kicking out a reduced number of less desirable configurations after experiments and analysis. Finally, we compare various R-U methods for our task. This systematical procedure enables an automatic way for well testing and validating our solutions.

In our work, the dataset was collected and constructed carefully to be representative of several extremist Arabic text. For our study, we manually crafted the database that consists of 44,007 tweets from Twitter; we refer as Extremism Arabic Post Dataset (EAPD). This dataset has been used and collected by us presented in [24] and we follow the same method used for data collection and annotation followed in [25, 26]. To ensure that the dataset was representative of various types of extremist narratives, tweets and posts were collected using a set of keywords (known to be associated with extremist content from previous literature or fact-to-face meeting) as well as hashtags. In addition negative samples (text not extremist related) were extracted from Arabic News Channels, research pages and normal conversation to make the classification task more diverse.

An automated as well as manual annotation design-guideline was used for annotating the data. A pre-trained classifier was employed to support primary classification, which was later reviewed by human annotators {domain experts: Arabic native speakers with expertise in extremist studies} over two and half months. During the labelling phase, humans annotators were given each instance whether Extreme content is believed to be present or not in it after [24] expressive guidelines. Linguistic markers, sentiment and context inidcating extremist ideology were the selection criterea for the labels. To ensure the accuracy of prediction, several annotators annotated the samples independently and Cohen’s Kappa score was provided to evaluate the reliability. These posts were read manually to see if they met the criteria of our proposed approach, resulting in two groups: extremist and non-extremist. We also worked with oversampling technique in our approach to cope up with class imbalances, as it could cause the models to lean towards majority classes,also for Improving Model performance on minority classes. Additionally, we manually curated the extremist/nonextremist labels

to provide a sanitary data set for training and evaluating the ML models. Table1 shows the distribution of these groups. Regarding ethical considerations for analyzing extremist contents, we followed the guidelines of ethics provided by [12, 24].

Table 1. The details of dataset specifications

Post label	Post N. in Original	Post N. after oversampling	Arabic meaning
Extremism	21502	22505	متطرف
Non-Extremism	22505	22505	غير متطرف

It is crucial to note that any inaccuracies in the labelling process could significantly degrade the quality of the dataset, undermining the reliability of our analysis. We then constructed three different versions of the corpus (V1, V2, and V3) as shown in Table 2, each employing varying degrees of preprocessing techniques:

- V1 (Pre-processing): This version involved basic preprocessing techniques, excluding stemming, to clean the actual data.
- V2 (Root stemming): Built by applying the ISRI root stemmer to V1 to further refine the text.
- V3 (Light stemming): Developed by applying a light stemming process to V2, aiming for a balance between root stemming and minimal text alteration.

Table 2. Corpus version details.

Version	Corpus name	Description
V1	Pre-processing	This version was created by doing pre-processing on actual data.
V2	Root stemming	Created by using the ISRI root stemmer on V1.
V3	Light stemming	This version was created by using a light stemmer on V2.

The preprocessing steps, as illustrated in Figure 3, are designed to remove extraneous information commonly found in online text, such as scripts, HTML tags, links, and punctuation [23].

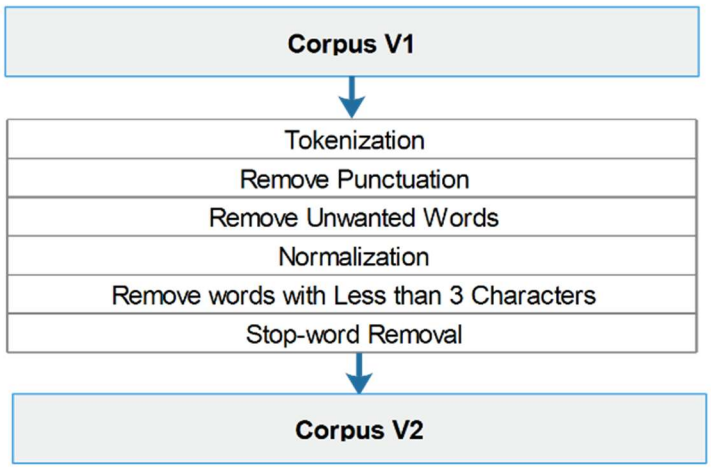

Figure 3. Pre-processing steps

This intensive pre-processing and normalization are necessary for guaranteeing the quality and uniformity of data going into our machine learning models, which in turn leads to more reliable sentiment analysis and classification tasks. In our work, we tokenize the text into sentences and words. Due to the complexity in Arabic script, some special punctuation and orthographic symbols are present [3], we apply the specific regular expressions (Table 3) to preprocess our text data..

In the Preprocessing step, Arabic text normalization is applied by stripping diacritics and normalizing typography in order to standardize the dataset. One example of these typographical aspects is the Tweel for stretching Arabic text. The normalization procedure is explained in Figure 4, where one can view the diacritics and certain letter forms being set to standard..

Table 3. Regular expression breakdown to refine our text data

Regular expression	Outcome
[a-zA-Z] +	Eliminate English Alphabetical symbols
[0-9] +	Eliminate English Numerical digits
[0-9] +	Eliminate Arabic Numerical digits
#$%|@^~(&*)+	Eliminate Miscellaneous characters

Figure 4. Arabic diacritics standardization

In Arabic typography, the Tweel (تطويل) is utilized to extend text by increasing the spacing between words or characters, represented by the script (ـــــ). For example, "الامـــــــــلاح" ,"الامـــلاح", and "الاملاح" all normalize to "الاملاح". Addressing such variations is crucial for maintaining data consistency and enhancing the analysis's accuracy.

Normalization stage 3 involves the substitution of special Arabic characters with their most frequent corresponding character to make text standard across treatments. This is summarized in Table 4, which presents a number of normalization rules that were used to normalize Arabic characters so orthographic differences do not interfere the analysis. The analysis involves elimination of Arabic stop words by comparing the text with two separate lists of stop words.

Table 4. The letter normalization of the Arabic language.

Original version	Normalized version	Illustration
إأآا	ا	ألاملاح "salts"
ى	ي	قوى "powerful"
ؤ	ء	اداؤه "his performance"
ئ	ء	يتلائم "suits"
ة	ه	ثورية " groundbreaking "

The linguistic complexity of Arabic, i.e., large vocabulary and complex syntactical rules, requires the development of stemming accordingly [27,28]. The ISRI stemming approach (employed here for corpus V2) reduces words to roots according to specific rules. A variant (the Arabic light stemmer) -this is a description can be found in table 5- that update these rules to better capture the subtleties of Arabic, such as 'waw' and certain prefixes and suffixes. The flowchart in Figure 5 summarizes the procedure that was followed using the text in V2 to arrive at V4, here also taking advantage of both root-stemming and light stemming leading to corpus refinement there further.

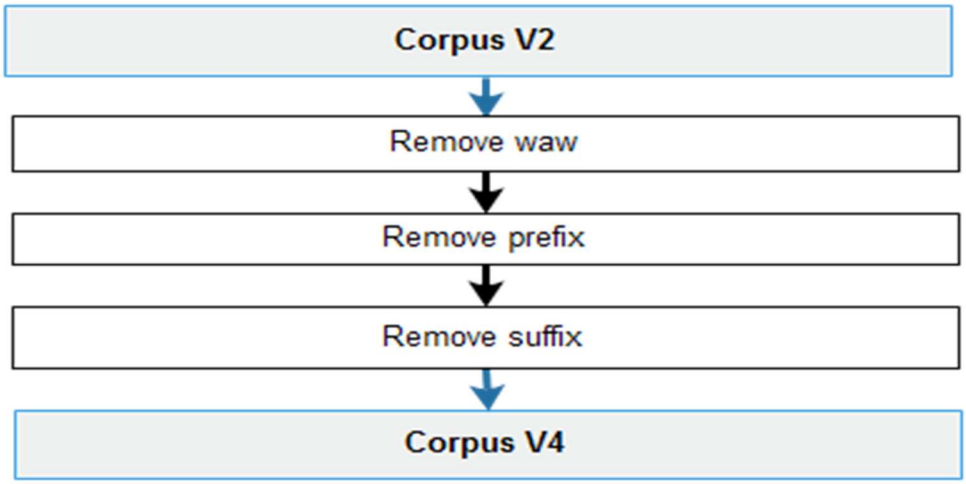

Figure 5. The process of conversion corpus version 2 to corpus version 4

Table 5. Examples of modifications in the Arabic light stemmer include handling the WAW prefix and suffix

Original	Become	Power (kW)	
Waw	w >= 4	1	و "and"
Prefix	p>=6	3	وال ولل، بال، كال، "Kal, pal, and for, and the"
	p>=5	2	ال لل، "For, the"
Suffix	s>=6	3	تين كمل، تان، همل، تمل، "act, action, tension, complete, fig"
	s>=5	2	هم,ما , وا ,ني ,كن , تم , ها يا، نا، هن، كم، تن، ين، ان، ات، ون، "We, you, you, we, you, you, how many, they, we, you, you, they"

In this study, we employ the Bag of Words (BOW) model and with three n-gram specifications: Uni-ngram, Bi-ngram, and Tri-ngram. These setups enable the formation of feature vectors that represent document based on term frequency and term significance as proposed by TF and TF-IDF (see [24, 29]).

The TF-IDF weighing scheme is based on the term frequency (TF) but also takes into account how common each word is in general. The TF-IDF of term is calculated as given in 5 where accounts for the frequency of word types (t). and its inverse document frequency across a set of documents. TF-IDF algorithm is also shown algorithm 1.

For each document, the BOW model captures the occurrence of n-grams, which are then weighted by their TF-IDF scores to form a vector representation of the document. This vectorization process is crucial for the subsequent machine learning analysis. The BOW approach represents the articles in a vector format, where the weights assigned to each term correspond to the frequency of its occurrence within the text. Using Equation (3), we can determine the weight vector in Equation (4).

$$t_i^j = \begin{bmatrix} d & T_1 & T_2 & \cdots & T_m & l \\ A_1 & w_1^1 & w_2^1 & \cdots & w_k^1 & l_1 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ A_i & w_1^i & w_2^i & \cdots & w_k^i & l_i \end{bmatrix} \quad (3)$$

$$(tf - idf)_{t,d} = tf_{t,d} * idf_t \quad (4)$$

Table 6 illustrates the distribution of features extracted using different n-gram techniques across various corpus versions used for training. The dataset was partitioned with 70% dedicated to the training phase and the remaining 30% reserved for evaluation, ensuring a balanced approach to model training and performance assessment.

Table 6. Feature extraction using three grams for training set.

Corpus	Uni-ngram	Bi-ngram	Tri-ngram
V1	63594	255642	254454
V2	38622	237391	280685
V3	12892	199923	277317

Algorithm 1. TF-IDF Algorithm

Input: Collection of training and testing data samples Tr = set of training samples x_i, l_j where $i = 1, 2, 3, \ldots, n$ and $j = 1, 2, 3,$ comprising training samples and respective class labels Z = set of e test samples, denoted as z_i where $i = 1, 2, 3, \ldots, e$

$Y \leftarrow \emptyset$, an empty set

Analyze the training data Tr

For each $z_i \in Z$, perform:
(a) $P\ l_j \leftarrow$ compute article's class using (2)
(b) $P\ x_i|l_j \leftarrow$ calculate likelihood for each class based on the model
(c) $y \leftarrow$ predicted label by applying (2) on z_i, considering (a) and (b)
(d) $Y \leftarrow Y \cup y$, adding y to the set Y
Output: Set of predicted class labels Y = set of predicted labels yi where $i = 1, 2, 3, \ldots, e$ - the test samples in Z with their corresponding predicted class labels.

5. Experimental results

In this study, various machine learning algorithms and feature extraction methods such as TF and TF-IDF were utilized. The setup parameters of the baseline models were configured as described in their original papers. To determine the most effective combination of feature extraction method and corpus for machine learning, algorithm comparisons were essential.

The experiments were conducted using Python 3.7.4 and PyCharm IDE 2024.2.3. Various required libraries, including NumPy, NLTK, Scikit-learn, and Tweepy, were utilized for the implementation and experimental configurations. The experiments were performed on a personal system equipped with an Intel Core i7 CPU, Windows 10 operating system, and 32 GB of RAM. Three experiments were conducted using a carefully selected benchmark from a real-dataset machine learning repository. The EAPD dataset was divided into two segments: 70% of the data was allocated for training, and the remaining 30% was set aside for testing. This partitioning was crucial for evaluating the efficacy of the proposed strategy. Table 7 reveals that as the number of syllables in a word increases from Bi-ngram to Tri-ngram, and so forth, there is a proportional increase in the vector's size.

Table 7. Various implementations of the naïve Bayes algorithm leveraging TF-IDF.

Naïve Bayes model	N-Gram model	(Corpus)			Vote
		V1	V2	V3	
Multi-nomial	Uni-ngram	78.658	83.536	77.439	V2
	BI-Ngram	77.439	78.048	80.487	V3
	Tri-ngram	58.536	60.975	60.975	V2 & V3
Berno-ulli	Uni-ngram	78.658	81.707	78.048	V2
	BI-Ngram	72.560	72.560	73.170	V3
	Tri-ngram	56.097	56.707	59.146	V3
Comp-lement	Uni-ngram	79.268	82.926	79.268	V2
	BI-Ngram	78.658	79.268	78.658	V2
	Tri-ngram	59.756	61.585	62.195	V3
Gauss-ian	Uni-ngram	70.731	71.951	72.560	V3
	BI-Ngram	78.048	75.609	75	V1
	Tri-ngram	64.024	67.073	67.682	V3
Best gram/ corpus		Uni-ngram (3)	Uni-ngram (3)	Uni-ngram (2) & BI-Ngram (2)	V3(7)

The experiments were performed on three different corpuses: V1, V2, and V3. According to Table 7, the highest performance was achieved with corpus V3 utilizing the TF-IDF feature extraction method and various Naive Bayes models, achieving a score of 7. Corpus V2 scored 5 with the same feature extraction method, while Corpus V1 scored 1. The precision-recall metrics for different Naive Bayes models using TF-IDF on corpus V3 are displayed in Figure 6. The highest-class ROC for extremism and non-extremism was 0.88, while the lowest class ROC for both classes in Tri-n-gram was 0.69.

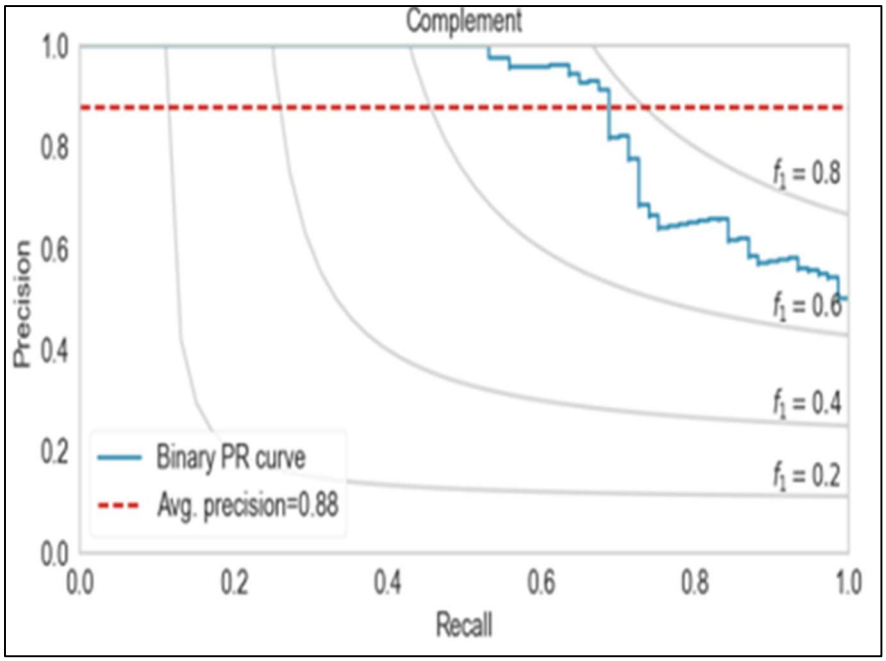

Figure 6. Precision-recall graph for different NB models using TF-IDF on corpus V3

Table 8 indicates that the best performing text corpus was V3 when using the Uni-n-gram feature representation in combination with Naive Bayes classification models. Specifically, the complement and multinomial models had average precision-recall values of 0.88, whereas the Gaussian model recorded a value of 0.64. The Bernoulli model had an average precision recall of 0.86. Among these, the Complement Naive Bayes model for Uni-n-gram provided the highest accuracy in this experimental study.

Table 8. Different naive bayes using TF-IDF

SVM	N-gram model	(Corpus)			Vote
		V1	V2	V3	
RBF	Uni-ngram	46.951	46.951	46.951	-
	Bi-ngram	46.951	46.951	46.951	-
	Tri-ngram	46.951	46.951	46.951	-
Polynomial	Uni-ngram	46.951	46.951	46.951	-
	Bi-ngram	46.951	46.951	46.951	-
	Tri-ngram	46.951	46.951	46.951	-
Sigmoid	Uni-ngram	46.951	46.951	46.951	-
	Bi-ngram	46.951	46.951	46.951	-
	Tri-ngram	46.951	46.951	46.951	-
Linear	Uni-ngram	81.097	81.707	79.268	V1 & V2
	Bi-ngram	75	75	75.609	V3
	Tri-ngram	62.804	65.853	66.463	V3
Best gram/ corpus		Uni-ngram (1)	Uni-ngram (1)	Uni-ngram (1)	V3 (2)

The Receiver Operating Characteristic Area Under the Curve (ROC AUC) graph, depicted in Figure 7, visualizes the precision and effectiveness of the models. This plot illustrates the balance between the True Positive Rate (TPR) on the vertical axis and the False Positive Rate (FPR) on the horizontal axis.

The ideal position in the top-left corner of the figure represents the optimal point that can be achieved by the ROC: zero FPR and one TPR. Assessing this trade-off allows for evaluating the model configurations that achieve the highest level of accuracy and performance.

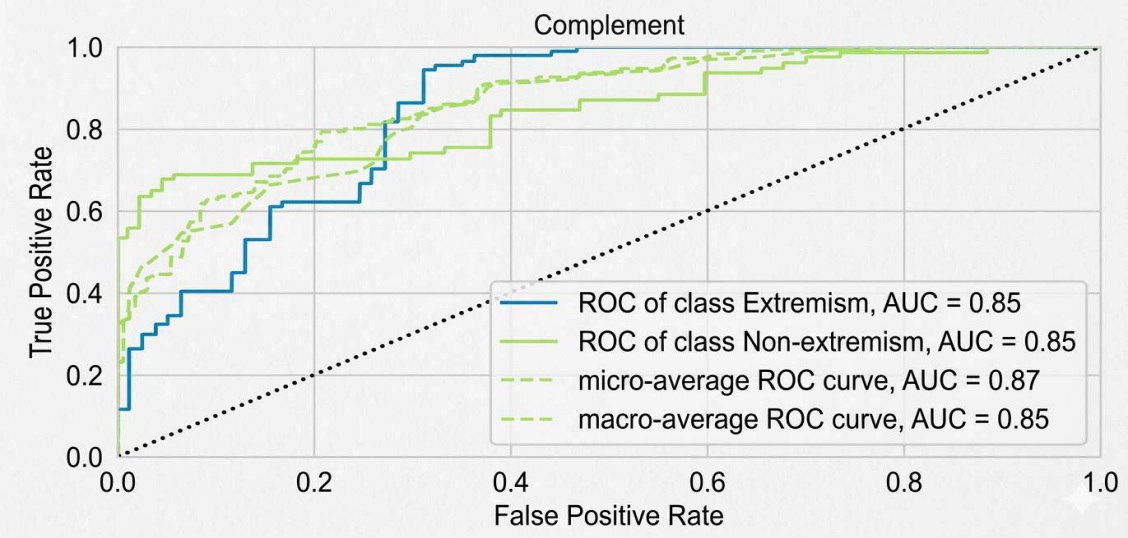

Figure 7. Results of ROC AUC curve for different NB models on corpus V3

Figure 7 also shows the ROC and AUC curves used to determine the optimal curve for the two groups in the Arabic extremism datasets. The results revealed that both the complement and multinomial models achieved an AUC score of 0.85 across the extremism and non-extremism categories. Bernoulli's model achieved an AUC of 0.83 for both classes. The Gaussian model performed the lowest in all classes with a ROC of 0.72, whereas the highest was achieved by both the complement and multinomial models with a ROC of 0.85. Table 9 displays different Naive Bayes models that utilized TF feature extraction with three grams.

Table 9. Different SVM models employing the TF-IDF technique.

NB model	N-gram model	(Corpus)			Vote
		V1	V2	V3	
Multi-nomial	Uni-ngram	80.487	83.536	78.048	V2
	Bi-ngram	75	76.829	76.829	V2 & V3
	T-Ngram	60.365	60.975	60.365	V2
Berno-ulli	Uni-ngram	78.658	81.707	78.048	V2
	Bi-ngram	72.560	72.560	73.170	V3
	T-Ngram	56.097	56.707	59.146	V3
Comp-lement	Uni-ngram	81.097	83.536	78.048	V2
	Bi-ngram	74.390	76.219	76.219	V2 & V3
	T-Ngram	59.756	60.975	59.756	V2
Gauss-ian	Uni-ngram	72.560	75	70.121	V2
	Bi-ngram	79.878	78.048	76.829	V1
	T-Ngram	64.024	67.073	67.682	V3
Best gram/ corpus		Uni-ngram (3)	Uni-ngram (3)	Uni-ngram (3)	V2 (8)

The results detailed in Table 8 underscore that corpus V2 demonstrated superior performance when combined with TF and various NB models, outperforming other corpora by a significant margin. Corpus V3 achieved a notable score of 5, while Corpus V1 lagged with a score of just 1. The complement and multinomial algorithms particularly showed high accuracy with both TF-IDF and TF feature extraction methods, with corpora V2 and V3 outperforming V1, especially when employing Uni-ngrams.

Table 10 reveals that the RBF, polynomial, and sigmoid kernels achieved the same accuracy of 46.951%. The linear kernel with the Uni-ngram feature achieved the highest accuracy in corpus V2, recording a score of 81.097%. The results across Tables 9 and 10 demonstrate promising performance of corpus V2 when used with Support Vector Machine (SVM) models.

Figure 8 illustrates that the Uni-ngram feature consistently performed the best compared to other n-grams. Figure 9 shows the F-score for corpus V2 using a linear kernel, ranging from 0.65 to 0.77 when paired with TF-

IDF. Figure 10 displays the precision-recall plot for corpus V2 using Euclidean and Minkowski distances with TF-IDF.

Table 9 highlights the performance variations across different kernels and three corpora using TF-IDF. The linear kernel with Uni-ngram scored the highest accuracies of 81.097% and 81.707% in corpora V1 and V2, respectively. In contrast, corpus V3's Bi-ngram achieved an accuracy of 75.609%, which was higher than the Tri-ngram's accuracy of 66.463%.

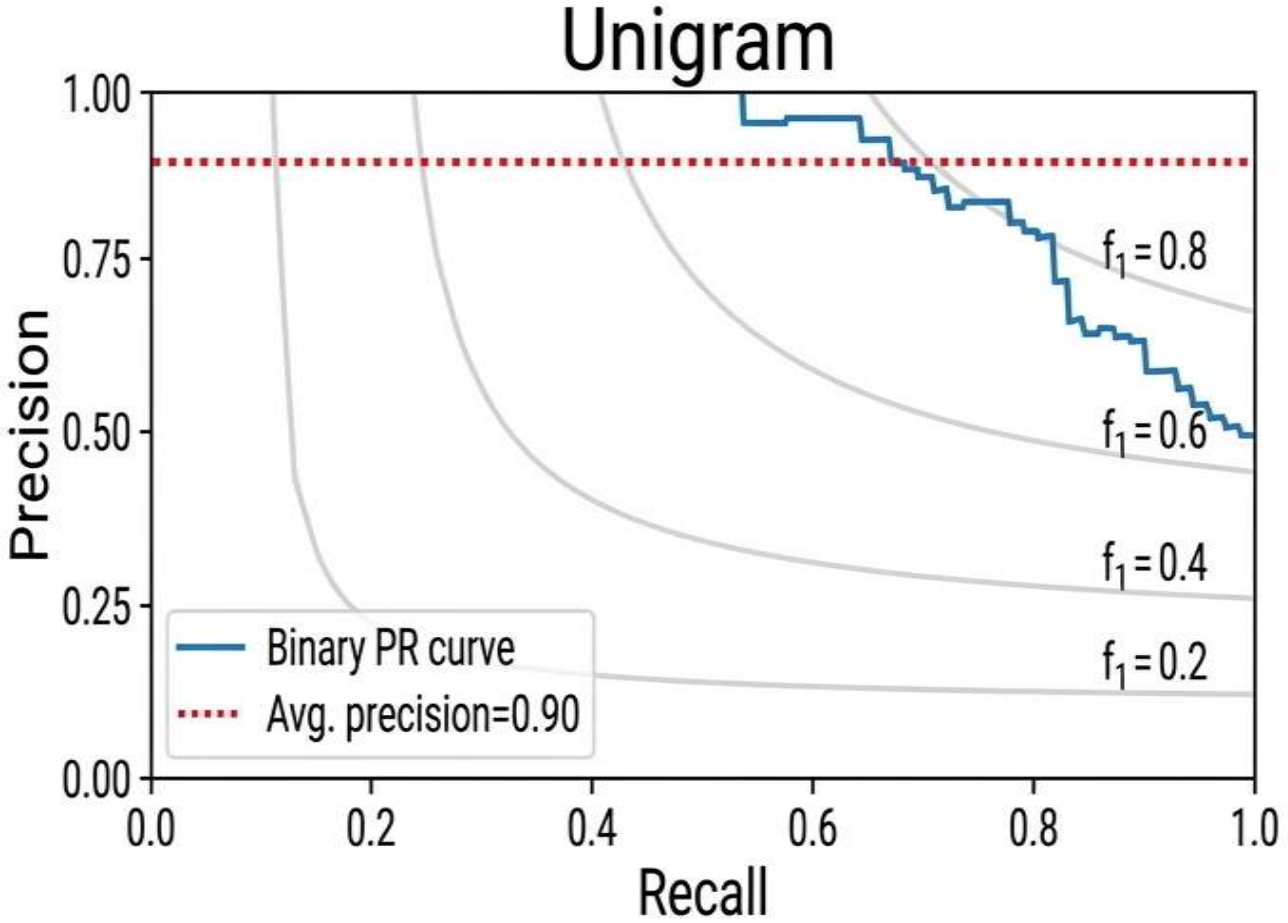

Figure 8. Precision-recall graph of linear kernel with TF-IDF on corpus V2

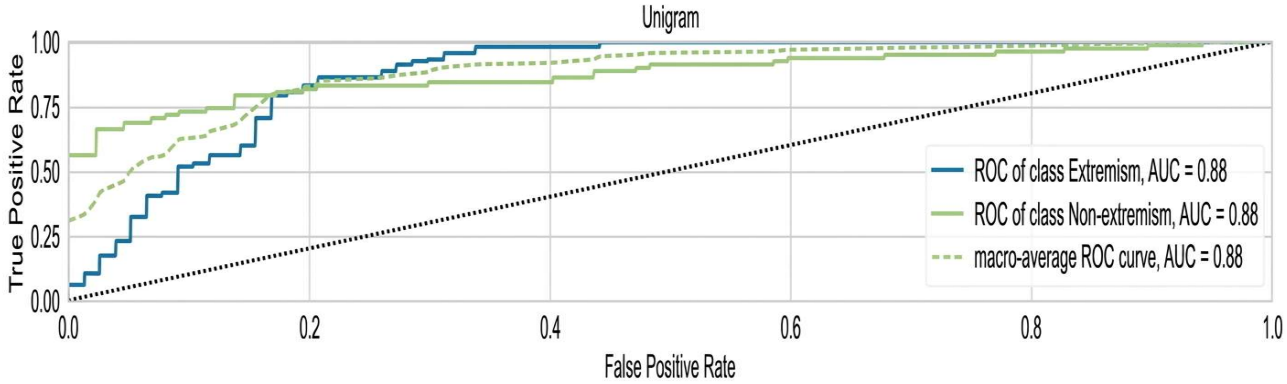

Figure 9. ROC AUC curve of linear kernel using TF-IDF with three grams

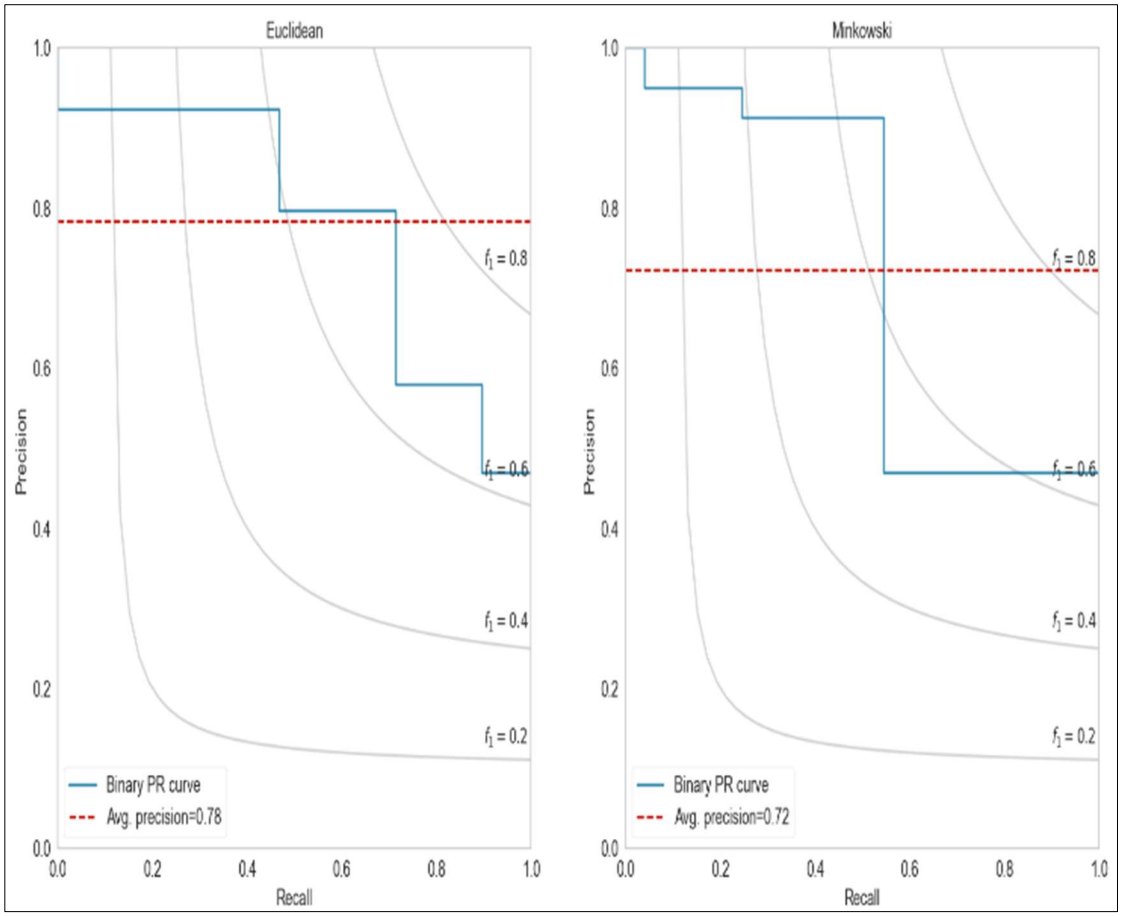

Figure 10. Precision-recall graph of Euclidean and Minkowski with TF-IDF on corpus V2

Table 10. Different SVM models employing the TF-IDF technique.

SVM	N-gram model	Corpus			Vote
		V1	V2	V3	
RBF	Uni-ngram	46.951	46.951	46.951	-
	Bi-ngram	46.951	46.951	46.951	-
	Tri-ngram	46.951	46.951	46.951	-
Polynomial	Uni-ngram	46.951	46.951	46.951	-
	Bi-ngram	46.951	46.951	46.951	-
	Tri-ngram	46.951	46.951	46.951	-
Sigmoid	Uni-ngram	46.951	46.951	46.951	-
	Bi-ngram	46.951	46.951	46.951	-
	Tri-ngram	46.951	46.951	46.951	-
Linear	Uni-ngram	78.658	81.097	74.390	V2
	Bi-ngram	72.560	72.560	71.951	V1 & V2
	Tri-ngram	59.756	60.365	60.365	V2 & V3
Best gram/ corpus		Uni-ngram (1)	Uni-ngram (1)	Uni-ngram (1)	V2 (3)

Table 11 shows the performances of the others feature extractions also based on different kernels to SVM algorithm that can be configured through a shorts list. The Nearest Neighbor Algorithm (that uses the combination of Euclidean and Minkowski distance metrics) was the most stable.

Analysis Results To validate the generalization of our model, we tested its performance on testing set and it could be seen from The results in Table 12 that with TF-IDF extracted features, the KNN Classifier produced satisfactory results by using only a subset of 3-K. Majority Voting KNN is one of the methods for estimation of classifier performance. As shown in Table 11, for the accuracy, the Euclidean distance clearly outperformed

the Minkowski distance. Corpus V2 had the highest accuracy using Euclidean distance metrics for uni-ngram (78.048%) and bi-ngram (76.219) yielding top performance with respect to all corpora and grams..

Table 11. KNN uses TF-IDF feature extraction.

KNN	Grams	Corpus			Vote
		V1	V2	V3	
Euclidean	Uni-ngram	73.170	78.048	73.170	V2
	Bi-ngram	73.170	76.219	73.780	V2
	Tri-ngram	53.048	53.048	53.048	-
Minkowski	Uni-ngram	59.146	64.024	65.853	V3
	Bi-ngram	53.048	53.048	53.048	-
	Tri-ngram	53.048	53.048	53.048	-
Best gram/ corpus		Uni-ngram (2)	Uni-ngram (2)	Uni-ngram (2)	V2 (2)

Table 12. KNN using TF-IDF feature extraction.

KNN	Grams	Corpus			Vote
		V1	V2	V3	
Euclidean	Uni-ngram	76.829	76.829	76.829	-
	Bi-ngram	70.121	70.731	68.292	V2
	Tri-ngram	53.048	53.048	53.048	-
Minkowski	Uni-ngram	75.609	75.609	76.219	V3
	Bi-ngram	70.121	70.731	67.682	V2
	Tri-ngram	53.048	53.048	53.048	-
Best gram/ corpus		Uni-ngram (2)	Uni-ngram (2)	Uni-ngram (2)	V2 (2)

Table 12 shows i.e. the number of votes obtained in KNN Voting using TF-IDF feature extraction, V2 (corpus) got two votes with one vote on V3 and zero on V1. This table 12 presents the performance of the KNN algorithm while using TF feature extraction and (respectively) Euclidean and Minkowski distance measure in operation.

As can be seen in Tables 11 and 12, K-NN consistently provided accurate predictions for the testing data set. Figure 8, the scaling candidate evaluation method transferred satisfactorily in the testing step and average precision-recall were 0.72 - 0.78 for corpus V2. 1 with the distance uses of Euclidean and Minkowski.

For those features, precision-recall area was 0.78 on both extremes and without-extremes for Euclidean distance and it surpass the Minkowski distance one. Corpus V2 was better than even the light-stemming corpus V3, which resulted in a lower score compared to corpora V1 and V3. For both-classes KNN, 3-k and Euclidean distance was better than Minkowski as shown in Figure 10.

Additionally, Figure 11 (from top to bottom) present the ROC AUC analysis of Euclidean and Minkowski distance between all corpora by using TF-IDF as feature weighting technique.

The confusion matrices for Corpus V1 and Corpus V2 are given in Figure 12 and Figure 13, respectively. They present a more fine-grained analysis of the classification performance that provides the number of extremist and non-extremist texts (correctly or not) classified.

Table 13 depicts a comparison of our study with a baseline method [34]. Our study outperformed Kanan & Fox [34] in extremist Arabic text classification, achieving accuracy rates of 81.097% and 81.707% using TF-IDF with varied N-gram sequences, compared to their 78.00% with P-Stemmer and SVM.

This demonstrates the advantage of our approach in capturing contextual information more effectively than stemming-based methods. The results highlight the importance of robust feature extraction for Arabic text classification, emphasizing that TF-IDF with N-grams enhances detection accuracy.

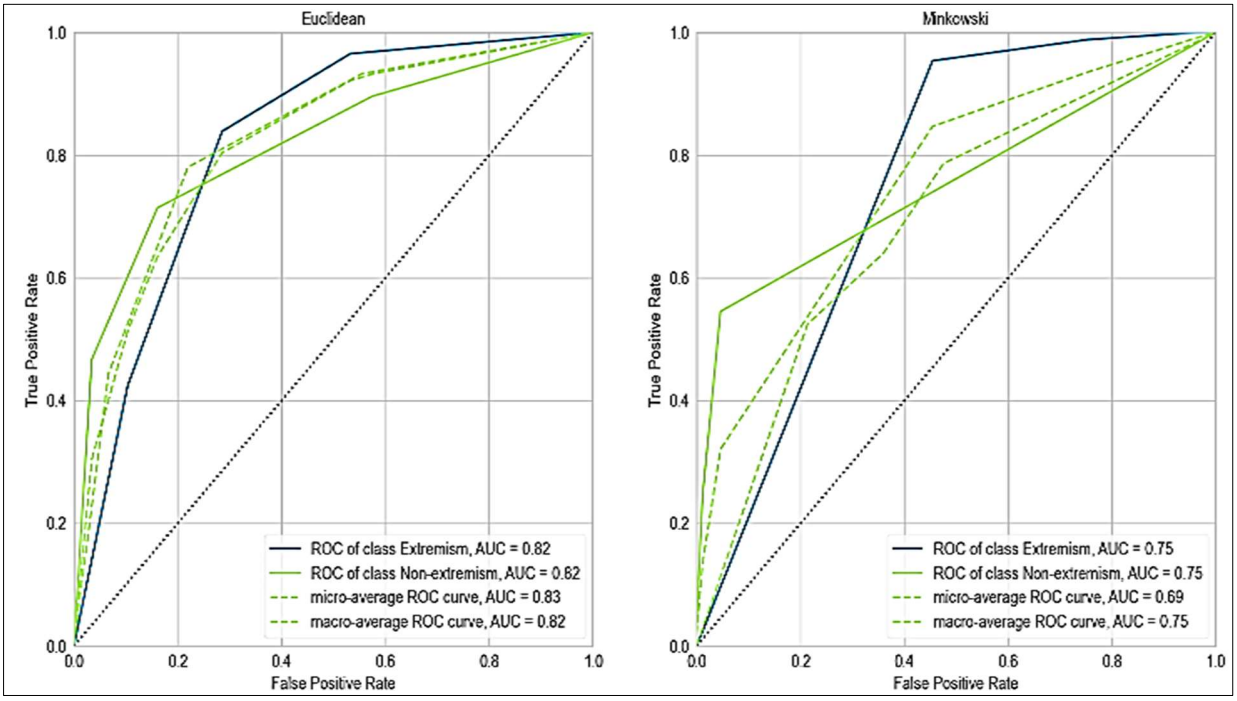

Figure 11. ROC AUC curve of Euclidean and Minkowski distance measures with TF-IDF feature weighting for all corpora

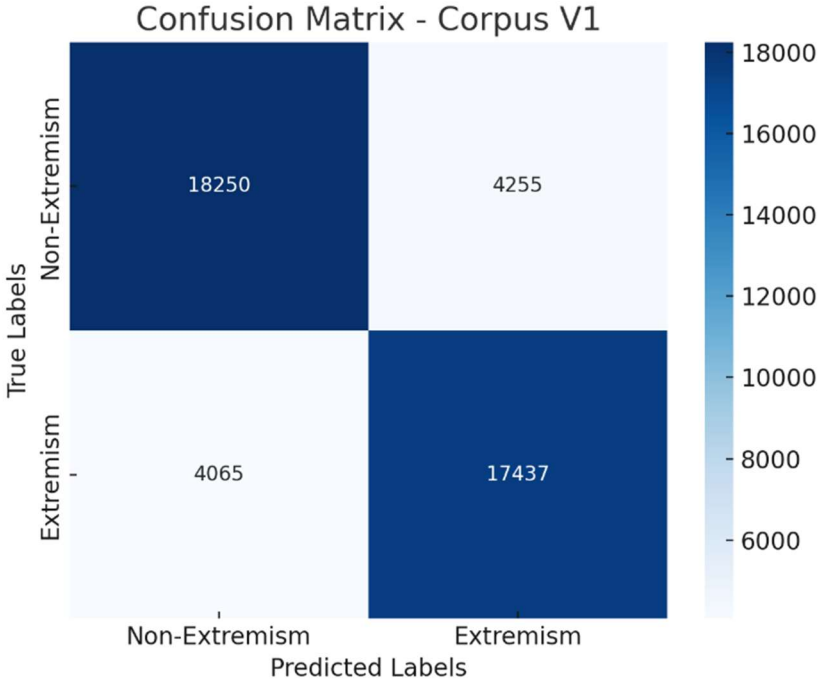

Figure 12. Computed Confusion matrix of Corpus V1

Table 13. Comparison table of our study with a baseline method

Method	Methodology	Accuracy
Our proposed method	TF-IDF with varied N-gram sequences	81.097% (Corpus V1), 81.707% (Corpus V2)
Kanan & Fox (2016) [34]	P-Stemmer with SVM classifiers	78.00%

6.Conclusions

The study's findings showed that the SVM model excelled as the most effective classifier in accurately determining and classifying sentiment polarity from tweet texts. TF. The IDF was one of the feature extraction methods that performed very well, particularly when coupled with SVM to work on corpus V3. The experiments tested with three orders of n-gram, although Uni-ngrams performed better for the machine learning techniques used, due to small size of the datasets. In full evaluation, over the comparison of different configurations for extremist sentiment detection, this configuration precedes all and especially provides the highest accuracy rates 81.097% and 81.707% in corpus V1 and V2 correspondingly.

These results are promising when compared with the existing (Kanan & Fox) approach for extremist Arabic text classification had it due to the following better performance rates : 74.00 % in accuracy even though perfected use of SVM and P-Stemmer we obtained 78.00 %. The analysis demonstrated several limitations. This included two types of feature extraction (by frequencies and by TF-IDF) and three sizes of n-grams. Machine learning techniques performed well for Uni-ngrams but degraded with the higher value of n, tri0 hard occurred near in all algorithms and a possible general situation turned out to be the push from Bi-ngrams (that what plays fine) to Tri-ngrams since we have no correlation between them which is a very steep move towards pushing this transition (what described as zero-relation constraint). This is the so-called lexical gap between smaller uni-ngrams and, although significant in English, does not capture distant semantic relations between words in text.

Additionally, the higher gram size caused a zipf effect and lead to a larger vector size - creating a "time-consuming constraint" by hiking up the dimension of vectors. The poor precision of the zero-relation constraint, as well as its extensive restriction, results in such bias. To solve these problems and enhance the accuracy of sentiment analysis in Arabic on extremism, further research can be carried out using a hybrid approach. Such a novel idea could take the best aspects of each of different feature extrac- tion techniques and machine learning algorithms to tackle what this work had identified as its problems. Especially for modeling capable of leveraging a large n-gram but without the loss in the performance or accuracy is very important. In addition, more advanced techniques such as deep learning, which will presumably be able to model the meaning relationship within longer n-grams more effectively, could lead to an even better and faster sentiment analysis.

Lastly, expanding the dataset size could also mitigate some of the constraints related to the small corpus size and provide more robust training for the algorithms. Then, we will address the potential for comparing our method with recent deep learning-based approaches like LSTM and BERT over the expanding dataset in future work.

Acknowledgment

The authors express their gratitude to the Centre for Research and Innovation Management (CRIM), Universiti Teknikal Malaysia Melaka (UTeM) for their valuable support in this research.

Declaration of competing interest

The authors declare that they have no known financial or non-financial competing interests in any material discussed in this paper.

Funding information

The author declares that they have received no funding from any financial organization to conduct this research.

References

[1] E. Ash, G. Gauthier, and P. Widmer, "Relation: Text semantics capture political and economic narratives," *Polit. Anal.*, vol. 32, no. 1, pp. 115–132, 2024. https://doi.org/ 10.1017/pan.2023.11

[2] S. Aldera *et al.*, "Exploratory data analysis and classification of a new Arabic online extremism dataset," *IEEE Access*, vol. 9, pp. 161613–161626, 2021. https://doi.org/ 10.1109/ACCESS.2021.3132007

[3] J. Atwan, M. Wedyan, Q. Bsoul, A. Hamadeen, R. Alturki, and M. Ikram, "The effect of using light stemming for Arabic text classification," *Int. J. Adv. Comput. Sci. Appl.*, vol. 12, no. 5, pp. 768–773, 2021. https://doi.org/ 10.14569/IJACSA.2021.0120588

[4] F. Şahin, E. Doğan, M. R. Okur, and Y. L. Şahin, "Emotional outcomes of e-learning adoption during compulsory online education," *Educ. Inf. Technol.*, vol. 27, pp. 7827–7849, 2022. https://doi.org/ 10.1007/s10639-022-10949-0

[5] H. K. Duan, M. A. Vasarhelyi, M. Codesso, and Z. Alzamil, "Enhancing the government accounting information systems using social media information: An application of text mining and machine learning," *Int. J. Account. Inf. Syst.*, vol. 48, p. 100600, 2023. https://doi.org/ 10.1016/j.accinf.2022.100600

[6] M. A. H. Wadud, M. Mridha, and M. M. Rahman, "Word embedding methods for word representation in deep learning for natural language processing," *Iraqi J. Sci.*, vol. 63, no. 3, pp. 1349–1361, 2022. https://doi.org/ 10.24996/ijs.2022.63.3.34

[7] T. Shaik, X. Tao, C. Dann, H. Xie, Y. Li, and L. Galligan, "Sentiment analysis and opinion mining on educational data: A survey," *Nat. Lang. Process. J.*, vol. 2, p. 100003, 2023. https://doi.org/ 10.1016/j.nlp.2023.100003

[8] W. H. Abdulsalam, S. Mashhadani, S. S. Hussein, and A. A. Hashim, "Artificial Intelligence Techniques to Identify Individuals through Palm Image Recognition," *Int. J. Math. Comput. Sci.*, vol. 20, no. 1, pp. 165–171, 2025. https://doi.org/ 10.69793/ijmcs/01.2025/abdulsalam

[9] A. A. Ahmed, "Intelligent Arabic Text Categorization: Initial Study and Proposed Methodology on Classifying Arabic Text Using Enhanced Naïve Bayes Classification Approach," *J. Adv. Res. Dyn. Control Syst.*, vol. 10, Special Issue, pp. 2512–2521, 2018

[10] R. H. Ali and W. H. Abdulsalam, "Attention-Deficit Hyperactivity Disorder Prediction by Artificial Intelligence Techniques," *Iraqi J. Sci.*, vol. 65, no. 9, pp. 5281–5294, 2024. https://doi.org/ 10.24996/ijs.2024.65.9.39

[11] S. M. Al-Ghuribi, S. A. M. Noah, and S. Tiun, "Unsupervised semantic approach of aspect-based sentiment analysis for large-scale user reviews," *IEEE Access*, vol. 8, pp. 218592–218613, 2020. https://doi.org/ 10.1109/ACCESS.2020.3042050

[12] H. M. Alshahrani, "Classification of Arabic extremist web content through Arabic textual analysis," Ph.D. dissertation, Univ. Strathclyde, Glasgow, UK, 2020

[13] A. Omar, T. M. Mahmoud, T. Abd-El-Hafeez, and A. Mahfouz, "Multi-label Arabic text classification in online social networks," *Inf. Syst.*, vol. 100, p. 101785, 2021. https://doi.org/ 10.1016/j.is.2021.101785

[14] M. Fernandez and H. Alani, "Artificial intelligence and online extremism: Challenges and opportunities," in *Predictive Policing and Artificial Intelligence*, Routledge, 2021, pp. 132–162.

[15] S. M. Al-Ghuribi, S. A. Noah, and S. Tiun, "Various Pre-Processing Strategies for Domain-Based Sentiment Analysis of Unbalanced Large-Scale Reviews," in *Proc. Int. Conf. Adv. Intell. Syst. Inf.*, 2021, pp. 204–214. https://doi.org/ 10.1007/978-3-030-58669-0_19

[16] S. M. Al-Ghuribi, S. A. Mohd Noah, M. A. Mohammed, N. Tiwary, and N. I. Y. Saat, "A Comparative Study of Sentiment-Aware Collaborative Filtering Algorithms for Arabic Recommendation Systems," *IEEE Access*, vol. 12, pp. 174441–174454, 2024. https://doi.org/ 10.1109/ACCESS.2024.3491205

[17] R. A. Alraddadi and M. I. E.-K. Ghembaza, "Anti-Islamic Arabic text categorization using text mining and sentiment analysis techniques," *Int. J. Adv. Comput. Sci. Appl.*, vol. 12, no. 8, pp. 776–785, 2021. https://doi.org/ 10.14569/IJACSA.2021.0120888

[18] S. M. Al-Ghuribi and S. Alshomrani, "A simple study of webpage text classification algorithms for Arabic and English languages," in *2013 Int. Conf. IT Convergence Secur. (ICITCS)*, Macao, China, 2013, pp. 1–5. https://doi.org/ 10.1109/ICITCS.2013.6717800

[19] A. Rekik, S. Jamoussi, and A. B. Hamadou, "A recursive methodology for radical communities' detection on social networks," *Procedia Comput. Sci.*, vol. 176, pp. 2010–2019, 2020. https://doi.org/ 10.1016/j.procs.2020.09.237

[20] N. A.-H. and A. B. Kamsin, "An effective text classifier using machine learning for identifying tweets' polarity concerning terrorist connotation," *I. J. Inf. Technol. Comput. Sci.*, vol. 5, pp. 19–29, 2021. https://doi.org/ 10.5815/ijitcs.2021.05.02

[21] M. Asif, A. Ishtiaq, H. Ahmad, H. Aljuaid, and J. Shah, "Sentiment analysis of extremism in social media from textual information," *Telemat. Inform.*, vol. 48, p. 101345, 2020. https://doi.org/ 10.1016/j.tele.2020.101345

[22] M. Elgamal, "Sentiment Analysis Methodology of Twitter Data with an application on Hajj season," *Int. J. Eng. Res. Sci.*, vol. 2, no. 1, pp. 82–87, 2016.

[23] B. A. H. Murshed, J. Abawajy, S. Mallappa, M. A. N. Saif, S. M. Al-Ghuribi, and F. A. Ghanem, "Enhancing big social media data quality for use in short-text topic modeling," *IEEE Access*, vol. 10, pp. 105328–105351, 2022. https://doi.org/ 10.1109/ACCESS.2022.3210338

[24] I. A. Alzuabidi, L. S. Jamil, A. A. Ahmed, S. A. M. Noah, and M. K. Hasan, "Hybrid Technique for Detecting Extremism in Arabic Social Media Texts," *Elektron. Elektrotechnika*, vol. 29, no. 5, pp. 70–78, 2023. https://doi.org/ 10.5755/j01.eie.29.5.34446

[25] B. A. H. Murshed, J. Abawajy, S. Mallappa, M. A. N. Saif, and H. D. E. Al-ariki, "DEA-RNN: A Hybrid Deep Learning Approach for Cyberbullying Detection in Twitter Social Media Platform," *IEEE Access*, vol. 10, pp. 25857–25871, 2022. https://doi.org/ 10.1109/ACCESS.2022.3153675

[26] B. A. H. Murshed *et al.*, "Short text topic modelling approaches in the context of big data: taxonomy, survey, and analysis," *Artif. Intell. Rev.*, vol. 56, pp. 1–62, 2022. https://doi.org/ 10.1007/s10462-022-10254-w

[27] A. Noaman and S. Al-Ghuribi, "A new approach for Arabic text classification using light stemmer and probabilities," *Int. J. Acad. Res.*, vol. 4, no. 3, pp. 114–121, 2012.

[28] A. A. Ahmed *et al.*, "Arabic Text Detection Using Rough Set Theory: Designing a Novel Approach," *IEEE Access*, vol. 11, pp. 68428–68438, 2023. https://doi.org/ 10.1109/ACCESS.2023.3292150

[29] S. Aldera, A. Emam, M. Al-Qurishi, M. Alrubaian, and A. Alothaim, "Exploratory data analysis and classification of a new Arabic online extremism dataset," *IEEE Access*, vol. 9, pp. 161613–161626, 2021. https://doi.org/ 10.1109/ACCESS.2021.3132007

[30] E. Miranda, M. Aryuni, Y. Fernando, and T. M. Kibtiah, "A study of radicalism contents detection in Twitter: Insights from support vector machine technique," in *2020 Int. Conf. Inf. Manage. Technol. (ICIMTech)*, Bandung, Indonesia, 2020, pp. 549–554. https://doi.org/ 10.1109/ICIMTech50096.2020.9211246

[31] I. Aljarah *et al.*, "Intelligent detection of hate speech in Arabic social network: A machine learning approach," *J. Inf. Sci.*, vol. 47, no. 4, pp. 483–501, 2021. https://doi.org/ 10.1177/0165551520917651

[32] P. Verma, "Extremism detection on social media using svm text classifier," *J. Pharm. Negat. Results*, vol. 13, p. 3748, 2022. https://doi.org/ 10.47750/pnr.2022.13.S09.439

[33] M. Fernandez and H. Alani, "Artificial intelligence and online extremism: Challenges and opportunities," in *Predictive Policing and Artificial Intelligence*, Routledge, 2021, pp. 132–162. *(Note: Duplicate of [14])*

[34] T. Kanan and E. A. Fox, "Automated Arabic text classification with P-Stemmer, machine learning, and a tailored news article taxonomy," *J. Assoc. Inf. Sci. Technol.*, vol. 67, no. 11, pp. 2667–2683, 2016. https://doi.org/ 10.1002/asi.23609