DOI: 10.24996/ijs.2023.64.10.37

ISSN: 0067-2904

Arabic Cyberbullying Detection Using Support Vector Machine with Cuckoo Search

Marwa Q. Saadi* , Ban N. Dhannoon

Department of Computer Science, College of Science, Al-Nahrain University, Baghdad, Iraq

Received: 3/4/2022 Accepted: 11/11/2022 Published: 30/10/2023

Abstract

Cyberbullying is one of the biggest electronic problems that takes multiple forms of harassment using various social media. Currently, this phenomenon has become very common and is increasing, especially for young people and adolescents. Negative comments have a significant and dangerous impact on society in general and on adolescents in particular. Therefore, one of the most successful prevention methods is to detect and block harmful messages and comments. In this research, negative Arabic comments that refer to cyberbullying will be detected using a support vector machine algorithm. The term frequency-inverse document frequency vectorizer and the count vectorizer methods were used for feature extraction, and the results were improved using the cuckoo search algorithm. The resulting accuracy before and after optimizing the support vector machine's hyperparameters is 85.8% and 87.1%, respectively.

Keywords: Cyberbullying, Arabic text classification, Machine learning, Support vector machine, Optimization, Cuckoo search.

كشف التنمر الإلكتروني في اللغة العربية باستعمال ألة المتجهات الداعمة مع بحث الوقواق

مروة قاسم* ، بان نديم ذنون

قسم الحاسبات، كلية العلوم، جامعة النهرين، بغداد، العراق

الخلاصة

يعد التنمر الإلكتروني من أكبر المشكلات الإلكترونية التي تتخذ أشكالًا متعددة من المضايقات باستعمال وسائل التواصل الاجتماعي المختلفة ، وقد أصبحت هذه الظاهرة حاليًا شائعة جدًا ومتزايدة، خاصة بالنسبة للأعمار الصغيرة والمراهقين. التعليقات السلبية لها تأثير كبير وخطير على المجتمع بشكل عام والمراهقين بشكل خاص. لذلك ، تتمثل إحدى طرق الوقاية الناجحة في اكتشاف الرسائل والتعليقات الضارة وحظرها. في هذا البحث ، سيتم الكشف عن التعليقات العربية السلبية التي تشير إلى التنمر عبر الإنترنت باستعمال خوارزمية آلة المتجه الداعمة. تم استعمال متجه تردد اللفظة عكسِ تردد الوثيقة وطرق ناقل العد لاستخراج الميزات، وتم تحسين النتائج باستعمال خوارزمية بحث الوقواق. الدقة الناتجة قبل وبعد تحسين المعلمة الفائقة لجهاز ناقل الدعم هي 85.8٪ و 87.1٪ على التوالي.

*Email: marwasaadi059@gmail.com

1. Introduction

Every day, many people can access a massive amount of online information and share it. All of this is because the Internet provides scientists and researchers with important information matching their needs [1]. Web content can be unstructured content that includes forums, social networking sites, and others. This content has become very popular and attractive to many users who can express their opinions freely and write their inquiries and discuss them through these sites [2]. Expressing opinions and sharing inquiries takes many forms, which may be texts, videos, or gestures, and may be used positively or negatively, such as using hateful and offensive language and harassing some users. Negative opinions may also affect the interests of companies, or abuse certain goods, or insult members of society. All of this leads to a flat appearance of cyberbullying.

Cyberbullying can be defined as an act committed by a person or group of people aimed at harming and offending a particular victim by sending a message, posting a video, or commenting on one of various social media platforms [3]. In most cases, electronic problems may be difficult to track because they are unknown and have significant negative effects on society [4].

Given the spread of this phenomenon and the importance of the subject to society, we aim through this work to build a linear support vector machine model to detect cyberbullying in the comments of a social networking site through machine language and improve the results of this model through optimizing its hyperparameters using a cuckoo search algorithm. In this work, Support Vector Machine (SVM) will be used as a machine learning algorithm, and its results will be improved by cuckoo search optimization algorithms. Related works are explained in Section 2. The proposed model of this work is illustrated in Section 3. In Section 4, the experimental results are presented. Finally, Section 5 states the conclusions of this work.

2. Related works

Several research studies have been published in the field of cyberbullying. The following selected studies are the most interesting and the most recent.

The researchers in [3] hope to create an online bullying dictionary using Chi-square, Pointwise Mutual Information (PMI), and Entropy techniques. Their datasets were collected from the Twitter API, Microsoft-Flow, and YouTube comments. Then, they were compiled into a single file containing about 100,327 tweets and comments. The results show that the PMI approach gives the best performance in detecting cyberbullying compared to the Chi-square and Entropy approaches. The PMI outperformed Chi-square and Entropy by 81 percent, compared to 62.11 percent and 39.14 percent for Chi-square and Entropy, respectively. In [5], a machine learning technique is proposed to detect bad written work. A dataset of about 19,650 tweets and posts written in Arabic was used. The random forest method produced the greatest f1-score values, according to the data, where the percentage of random forest on the data set was (94%), while Naïve B was (91%) and SVM was (93%). In [6], different machine learning techniques such as Naive Bayes (NB), Complement Naïve Bayes (CNB), and linear regression (LR) were used to classify a dataset of Arabic YouTube comments, as well as two feature extraction methods: the Count Vectorizer and the term frequency-inverse document frequency (TF-IDF) Vectorizer. The CNB classifier performed better in the feature extraction of the TF-IDF vector. When it came to extracting the count vector attribute, linear regression performed the best. In general, the models perform somewhat better in terms of feature extraction when the TF-IDF is employed, since the average F1 score is 77.9%, while it is on average 77.5. The researchers in [7] identify and characterize cyberbullying on Twitter in the Arab world, namely in Saudi Arabia. PMI and SVM techniques are used to construct a vocabulary to assist in discovering and categorizing tweets. The F1 score after applying the PMI is 50%, but it is 82% after applying the SVM to the resampled data (either downsampling or oversampling). In [8], several issues were looked into around how to safeguard an Arabic text from cyberbullying/harassment via information shared on Twitter. The Word2Vec technique is used for feature extraction. The long-short-term memory (LSTM) deep learning model outperforms other classical cyberbullying classifiers, with an accuracy of 72%. The researchers in [9] used tools such as AraBully Keywords to collect data from Twitter for their work. The total number of tweets was 17748, and the researcher used SVM in WEKA and the Python compilation tool to work in the compilation phase after preprocessing. They found that when using Light Stemmer, WEKA accurately ranked 85.49%, and when using ArabicStemmerKhoja, WEKA ranked 85.3843%, but Python correctly ranked 84.03%.

3. The Proposed Model

Cyberbullying detection includes a binary classification task on distinguishing tweets between cyberbullying and non-cyberbullying, where cyberbullying is represented as a (C) class and non-cyberbullying is represented as an (N) class. The proposed methodology and its experimental setup to detect cyberbullying are shown in Figure1.

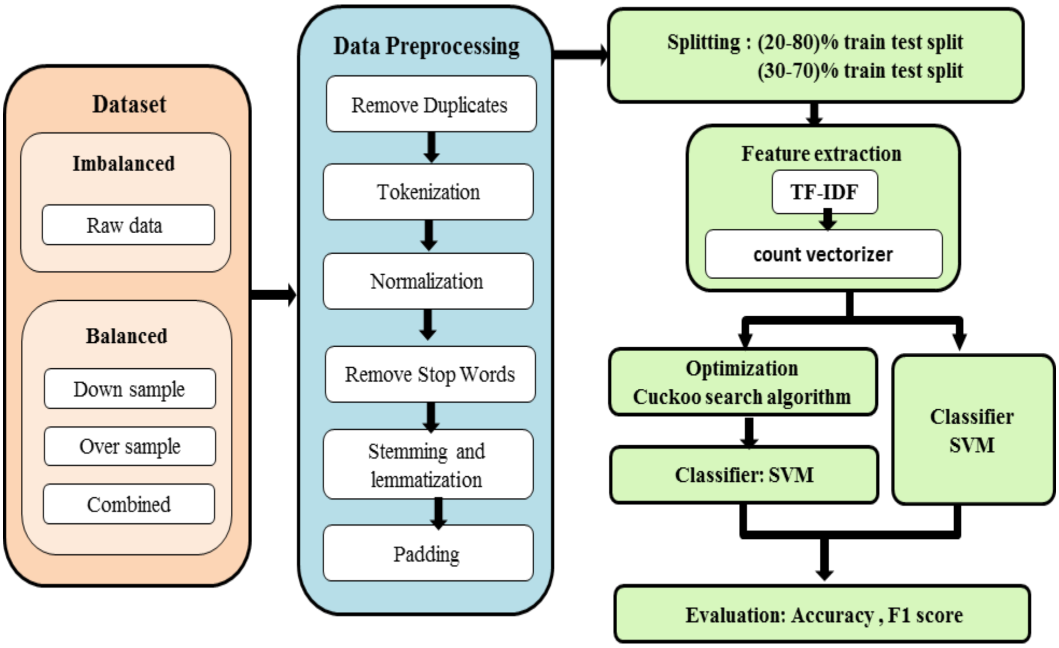

Figure 1: The Proposed Model

3.1 Dataset

The dataset was obtained from [9], where the tweets written in the Arabic language were stored in “Comma Separated Value” (CSV). The words that are mostly used to do Arabic cyberbullying include words like "ugliness," "racial discrimination," "tententiousness," "intolerance of opinion," and "dynasty." There were a total of 17748 Arabic tweets collected, where CyberBullying tweets were 14178 while the Non-CyberBullying tweets were 3570, so it could be considered an imbalanced dataset.

3.2 Imbalanced dataset

An imbalanced dataset refers to the distribution of classes that are not equal. In general, the ML algorithm works fine when all the classes have the same number of instances [10]. There are many techniques to solve imbalanced datasets (raw data (R)). These are downsampling the majority class (D), upsampling the minority class (U), and up-down sampling (UD). Minimizing the majority class through the downsampling technique means selecting randomized examples from the majority class and making them nearly equal to the number of instances of the minority class, but it is not recommended because it leads to data loss. The minority class upsampling technique injects data points (corresponding to the minority class) into the data set, so the enumeration of both labels is approximately the same and prevents the model from being skewed toward the majority class. The Synthetic Minority Oversampling Technique (SMOTE) technique is used as a synthetic data generation upsampling technique that utilizes the k-nearest neighbor algorithm to create synthetic data. For the up-down sampling technique, SMOTE with random under-sampling techniques was used simultaneously.

3.3 Data preprocessing

Preprocessing is an important step for building any classification model. Table 1 shows the main preprocessing steps and their effects on the used dataset.

Table 1: preprocessing steps

Remove duplicated	**This process deletes any duplicated tweets, thus the number of tweets decreased from 17748 to 17726 tweets.**
Tokenization	It is the process of splitting up Tweets into words called tokens, separated by commas [11]. Ex: (كذلك عاالم 2020 جدا" كئيب) → (كذلك، عاالم، 2020، جداً، كئب)
Normalization	Is the process of transforming a text into a canonical form by removing noises, such as dates, whitespaces, abbreviations, acronyms, and diacritics [12, 13]. Ex : (كذلك، عاالم، 2020، جداً، كئب) → (كذلك عام جدا كئب)
Stop Word Removal	This process is used to filter out unnecessary data by converting it into accepted forms. The removal of this unnecessary data does not affect the general meaning of the text [14]. This work obtained a list of Arabic stop words from [9] stored in an excel sheet (.xlsx). Ex: (كذلك عام جدا كئب) → (عام جدا كئب)
Stemming	It is a normalization technique in which a list of distinct words is converted into shortened root words to eliminate redundancy, this is done by removing their affixes and suffixes. Root stemming and light stemming are two types of stemming for the Arabic language [15]. Ex: Word: فاهمون , light stemming :فاهم , root stemming : فهم
Lemmatization	It is used to reduce words to their base representation by returning them to their meaning in proper form considering their morphological analysis. Ex: Word: فاهمون , Lemma: فَهْم
Padding	Since some sentences are long and others are short, post zeros padding with a maximum length equal to the longest sentence in the dataset is used to equalize the length of sentences [16].

3.4 Feature Extraction

Feature extraction refers to the procedures for selecting variables or combining them into features to reduce the amount of information that must be processed while still maintaining a correct and comprehensive characterization of the original data set.

TF-IDF is deployed for vectorization of text, which can be further used in feature mining. TF-IDF entails two factors: TF (term frequency) and IDF (inverse document frequency). TF (w) signifies the frequency of the word w in the document, count (w) and count (wn) denote the number of samples including the word w in the dataset, and n denotes the number of samples containing the word w in the corpus, as shown in Eq. (1). IDF (w) denotes the inverse file frequency of the word w in the equation and is computed using Eq. (2), where N is the number of documents in the dataset. TF-IDE can be computed as TF ∗ IDF.

$$TF(w) = \frac{count(w)}{\sum count(w_n)} \qquad \text{………… (1)}$$

$$IDF(w) = log\left(\frac{N}{df}\right) \qquad \text{………… (2)}$$

Count Vectorizer returns an encoded vector that has the same length as the entire comment and an integer count for the number of times each word appears in a comment [6].

3.5 Machine Learning

ML is a mathematical model with some parameters that must be learned from the data. However, there are some parameters, which are known as hyperparameters, that cannot be learned directly. A support vector machine is a supervised learning approach used for regression and classification [18]. It finds the isolating hyperplane that parts vector space into a sub-set of vectors; each isolated sub-set is called a data set and is assigned by one class [19]. A decent division is accomplished by the hyperplane that has a great distance to the nearest training data points of any class (so-called functional margin). In general, the bigger the margin, the lower the generalization error [20].

3.6 Hyperparameters Optimization

It is the process of selecting a set of optimal hyperparameters for a learning algorithm to reduce a predefined loss function in a given data set. Hyperparameters are very important in building robust and accurate models. They are commonly chosen by humans based on some intuition or multiplication and trial before actual training begins. These parameters help us find the balance between bias and variance and, therefore, prevent the model from overfitting or underfitting. The linear SVM model has a single hyper-parameter called the c value, and finding its best value can be treated as a search problem.

3.7 Cuckoo search (CS) algorithm

CS is a nature-inspired metaheuristic algorithm, developed by Xin-She Yang and Squash Deb in 2009 [21]. This algorithm was enhanced by the so-called Levy flights [22].To be straightforward in describing the CS algorithm, three rules are to be followed [23]:

- Each cuckoo lays one egg at a time, by choosing a nest randomly;
- High-quality eggs in the best nests will be transferred to the next iterations;
- The total number of all host nests is fixed, and the egg put by a cuckoo is found out by the host bird with a probability of P ∈ [0,1]. In this case, the host bird can either get the rub of the egg or simply let down the nest and construct a completely new nest.

The applied CS-SVM algorithm steps are described as follows:

- **Step 1**: Initializing cuckoo, number of iterations, and the range of SVM hyperparameter C value [upper bound and lower bound].
- **Step 2**: Apply cuckoo search to find the better initial hyperparameter C.
- **Step 3**: Evaluate the accuracy and f-score of the SVM classifier.
- **Step 4**: Comparing the accuracy and f-score and determining their best values.
- **Step 5**: Update the hyperparameter c until the distance of the decision boundary to classes is increased and the number of points that are correctly classified in the training set is maximized.
- **Step 6**: After converging, the best c hyperparameter is determined.
- **Step 7**: Training and testing downing with SVM classifier.

4. Experimental results

In this work, we conducted many experiments in different settings using a linear SVM classifier, tested using four corpora explained in Table (2), with some instances (cyber and non-cyber samples) in each corpus.

4.1 Splitting Dataset and Feature extraction

Each corpus was divided into (80/20) and (70/30) percent train-test datasets for classification and evaluation purposes prior to any classification process. This division gives the ability to model for training and testing on different dataset sizes. The up-sampling (U), downsampling (D), and combined (UD) sampling shown in Table (2) were performed only on the training set, with different splitting ratios of 80/20% and 70/30%. The TF-IDF method was applied to train and evaluate samples separately.

Table 2: Datasets and number of instance

Experiment setting	Corpus	Cyber	Non-cyber
Raw dataset (Imbalanced dataset)	R	14156	3570
Down (balanced dataset)	D	5000	3570
Upsampling (balanced dataset) 80/20	U	14156	12084
Upsampling (balanced dataset) 70/30	U	14156	11016
Down and upsampling (balanced dataset) 80/20	UD	14156	8679

4.2 Classification Stage

These experiments use 5-fold cross-validation on the whole corpus (outer loop) and 5-fold cross-validation on the training part (inner loop). This means that we perform validation five times for the main dataset and five times for the training dataset. The performance of the classification process was evaluated using accuracy and F1-score metrics on the four corpora, with different stemmers (lemma, light, and root). Table 3 shows the average evaluation accuracy and f1-score on an average of 5-fold cross-validation after applying TF-IDF vectorization on the training set in both cases of splitting. We noted that the classifier has an average of a very low training error (less than 1%) and an estimated 17% validation error, i.e., the estimated bias is about 0.4% and the variance about 16.5%. Thus, the model fails to generalize the tested data and it is said to be overfitted. To overcome the variance problem, the count vectorization feature extractor technique is applied, reducing the number of preprocessed features as shown in table (4). Table (5) shows the average evaluation accuracy and f1-score on an average of 5-fold cross-validation after reducing the number of features.

Table 3: SVM classifier on average of 5-fold cross-validation

Stem	Corpus	(80- 20)% train- test data	(70- 30)% train- test data

		Train	Val.	test	F1	Train	Val.	test	F1
Light	R	0.967	0.828	0.834	0.890	0.967	0.825	0.827	0.883
	U	0.989	0.847	0.854	0.904	0.990	0.844	0.851	0.903
	D	0.991	0.794	0.806	0.837	0.993	0.788	0.805	0.830
	UD	**0.967**	**0.826**	**0.860**	**0.911**	0.987	0.846	0.858	0.910
Root	R	0.934	0.843	0.840	0.897	0.934	0.840	0.841	0.897
	U	0.963	0.848	0.844	0.899	0.965	0.845	0.847	0.900
	D	0.951	0.796	0.797	0.827	0.955	0.785	0.793	0.825
	UD	0.960	0.849	0.848	0.900	0.967	0,847	0.850	0.903
Lemma	R	0.950	0.833	0.839	0.897	0.951	0.831	0.831	0.902
	U	0.979	0.832	0.832	0.889	0.974	0.845	0.848	0.899
	D	0.978	0.840	0.795	0.824	0.975	0.835	0.796	0.826
	UD	0.978	0.843	0.841	0.899	0.979	0.837	0.834	0892

Table 4: Number of features

Corpus	**R**			
	80 % Train set		**70 % Train set**	
Stem	**Before**	**After**	**Before**	**After**
Light	33614	3015	30954	3080
Root	10284	2582	9713	2598
Lemma	20579	2924	19147	2960

Table 5: SVM classifier of 5-fold cross-validation after count-vectorization

Stem	Corpus	(80- 20)% train- test data				(80- 20)% train- test data			
		Train	**Val.**	**test**	**F1**	**Train**	**Val.**	**test**	**F1**
Light	R	0.920	0.839	0.850	0.901	0.918	0.831	0.850	0.901
	U	0.942	0.845	0.849	0.900	0.956	0.840	0.845	0.896
	D	0.952	0.805	0.819	0.828	0.951	0.799	0.810	0.830
	UD	0.918	0.840	0.853	0.903	0.953	0.847	0.850	0.901
Root	R	0.914	0.848	0.855	0.904	0.916	0.845	0.856	0.905
	U	0.945	0.851	0.854	0.904	0.950	0.846	0.853	0.902
	D	0.926	0.789	0.810	0.832	0.927	0.786	0.800	0.828
	UD	0.913	0.851	0.855	0.904	0.948	0.847	0.854	0.904
Lemma	R	0.917	0.846	0.852	0.901	0.919	0.841	0.850	0.900
	U	0.945	0.852	0.856	0.904	0.950	0.848	0.855	0.903
	D	0.931	0.788	0.787	0.827	0.933	0.791	0.792	0.826
	UD	**0.946**	**0.854**	**0.858**	**0.906**	0.951	0.850	0.856	0.903

4.3 Optimization stage

After the classification step, the optimization step is applied to optimize the c-value hyperparameter in the linear SVM. A cuckoo search optimization algorithm is used for optimizing. The experiments were done on two corpora (R and UD).

Table (6) shows the accuracy and f1 measure with an optimized C hyperparameter value ranging from (0.1 to 1) and with up to 50 iterations. A higher accuracy (0.871) is obtained in an 80% training set with a 0.518 C-value applied to UD sampling and light stemming, while

(0.864) accuracy is obtained in a 70% training set with a 0.701 C-value applied to the raw dataset and light stemming. Table (7) shows a comparison between previous work and this work.

Table 6: Classification after c value optimization using cuckoo search

Corpus	Stem	(80- 20)% train- test data				(70- 30)% train- test data			
		Iteration	**C**	**Acc.**	**F1**	**Iteration**	**C**	**Acc.**	**F1**
UD	Light	**25**	**0.518**	**0.871**	**0.913**	15	0.923	0.859	0.909
	Root	16	0.246	0.857	0.909	11	0.243	0.858	0.910
	Lemma	4	0.416	0.855	0.908	33	0.294	0.854	0.907
R	Light	21	0.476	0.851	0.909	**38**	**0.701**	**0.864**	**0.910**
	Root	2	0.966	0.851	0.908	9	0.921	0.853	0.910
	Lemma	4	0.426	0.855	0.912	9	0.550	0.846	0.904

Table 7: Comparison with previous work

SVM	methods	Corpus	Acc.
Previous work [11]		R	84.04 %
This work	lemma + TF-IDF + Count Vectorized	UD	85.8%
	light stemmer + TF-IDF + Count Vectorized + CS	R	86.4 %
		UD	87.1%

5. Conclusions

The trained dataset is imbalanced. Three techniques were used to balance it and produce three corpora; downsampling (D), upsampling (U), and up-down sampling (UD). These corpora were applied and tested with the original raw data (R). Both stemming (light and root) and lemmatization were applied to reduce inflectional forms and sometimes derivationally related forms of a word to a common base form. Since the dataset is huge, especially after rebalancing and after training the model, we estimate the bias as less than 0.5% (it has a very low training error), and the variance as 16% (= 16.5%-0.5%). Thus, it has high variance, i.e., it is failing to generalize to the test data (i.e., overfitting). The count vectorization feature extractor technique was applied to overcome overfitting. After that, a cuckoo search optimization algorithm was used as an optimization method. The best accuracy and f1-measure, with 87.1% and 91.3%, respectively, were obtained.

References

[1] U. Can, and B. Alatas, "A new direction in social network analysis: Online social network analysis problems and applications," Physica A: Statistical Mechanics and its Applications, Vol. 535, pp.122372, 2019.

[2] Y. U. Sheng, and K. Subhash, "A survey of prediction using social media," arXiv preprint arXiv:1203.1647, Mar 7, 2012.

[3] B. Y. AlHarbi, M. S. AlHarbi, N. J. AlZahrani, M. M. Alsheail, J. F. Alshobaili, and D. M. Ibrahim, "Automatic cyber bullying detection in Arabic social media," Int J Engineering Research and Technology, Vol.12, No. 12, pp.2330-2335, 2019.

[4] S. Paul, S. Saha, and M. Hasanuzzaman, "Identification of cyberbullying: A deep learning based multimodal approach," Multimedia Tools and Applications, pp.1-20, 2020.

[5] T. Kanan, A. Aldaaja, and B. Hawashin, "Cyber-bullying and cyber-harassment detection using supervised machine learning techniques in Arabic social media contents," Journal of Internet Technology, Vol. 21, No. 5, pp.1409-1421, 2020.

[6] T. Alsubait, and D. Alfageh, "Comparison of Machine Learning Techniques for Cyberbullying Detection on YouTube Arabic Comments," International Journal of Computer Science & Network Security, Vol. 21, No. 1, pp.1-5, 2021.

[7] A. R. Almutairi, and M. A. Al-Hagery, "Cyberbullying Detection by Sentiment Analysis of Tweets' Contents Written in Arabic in Saudi Arabia Society," International Journal of Computer Science & Network Security, Vol. 21, No. 3, pp.112-119, 2021.

[8] E. Bashir, and M. Bouguessa, "Data Mining for Cyberbullying and Harassment Detection in Arabic Texts," International Journal of Information Technology and Computer Science, Vol. 13, No. 5, pp.41-50, 2021.

[9] S. Almutiry, and M. Abdel Fattah, "Arabic CyberBullying Detection Using Arabic Sentiment Analysis," The Egyptian Journal of Language Engineering, Vol. 8, No. 1, pp.39-50, 2021.

[10] B. A.Talpur, and D.O'Sullivan, "Cyberbullying severity detection: A machine learning approach," PloS one, Vol. 15, No. 10, pp.e0240924, 2020.

[11] Y.A.Alhaj, M.A. Al-qaness, A.Dahou, M.Abd Elaziz, D. Zhao, and J. Xiang, "Effects of light stemming on feature extraction and selection for Arabic documents classification," In Recent Advances in NLP: The Case of Arabic Language, pp. 59-79, Springer, Cham., 2020.

[12] L. S. Larkey, L. Ballesteros, and M. E. Connell, "Improving stemming for Arabic information retrieval: light stemming and co-occurrence analysis," In Proceedings of the 25th annual international ACM SIGIR conference on Research and development in information retrieval, pp. 275-282, Aug. 2002.

[13] D. H. Abd, A. T. Sadiq, and A. R. Abbas, "Political articles categorization based on different naïve bayes models," In International Conference on Applied Computing to Support Industry: Innovation and Technology, pp. 286-301, Springer, Cham, Sep. 2019.

[14] N. Shah, and S. Mahajan, "Document clustering: a detailed review," International Journal of Applied Information Systems, Vol. 4, No. 5, pp.30-38, 2012.

[15] M. Aljlayl, and O. Frieder, "On Arabic search: improving the retrieval effectiveness via a light stemming approach," In Proceedings of the eleventh international conference on Information and knowledge management, pp. 340-347, Nov. 2002.

[16] N. A. Hamzah, and B. N. Dhannoon, "The Detection of Sexual Harassment and Chat Predators Using Artificial Neural Network," Karbala International Journal of Modern Science, Vol. 7, No. 4, pp.6., 2021.

[17] V. Sindhwani, and P. Melville, "Document-word co-regularization for semi-supervised sentiment analysis," In 2008 Eighth ieee international conference on data mining, pp. 1025-1030, IEEE, Dec. 2008.

[18] J. T. Y. Kwok, "Support vector mixture for classification and regression problems," In Proceedings Fourteenth International Conference on Pattern Recognition (Cat. No. 98EX170), Vol. 1, pp. 255-258, IEEE, Aug. 1998.

[19] V. Jakkula, "Tutorial on support vector machine (svm)," School of EECS, Washington State University, Vol. 37, No. 2.5, pp.3, 2006.

[20] X. Liu, and H. Fu, "PSO-based support vector machine with cuckoo search technique for clinical disease diagnoses," The Scientific World Journal, 2014.

[21] Yang, X. S. and Deb, S., Cuckoo search via Lévy flights. In 2009 World congress on nature & biologically inspired computing (NaBIC), pp. 210-214, IEEE, Dec. 2009.

[22] I. Pavlyukevich, Lévy flights, "Non-local search and simulated annealing," Journal of computational physics, Vol. 226, No. 2, pp.1830-1844, 2007.

[23] X. S. Yang, and S. Deb, "Cuckoo search: recent advances and applications," Neural Computing and applications, Vol. 24, No. 1, pp.169-174, 2014.