

Sentiment analysis of imbalanced Arabic data using sampling techniques and classification algorithms

Maisa J. Al-Khazaleh¹, Marwah Alian^{1,3}, Manar A. Jaradat²

¹Department of Basic Sciences, Faculty of Science, The Hashemite University, Zarqa, Jordan

²Department of Computer Engineering, Faculty of Engineering, The Hashemite University, Zarqa, Jordan

³Faculty of Information Technology, The World Islamic Sciences and Education University, Amman, Jordan

Article Info

Article history:

Received Jan 29, 2023

Revised Jul 14, 2023

Accepted Aug 2, 2023

Keywords:

Arabic sentiment analysis

Ensemble learning

Hyper parameter tuning

Imbalanced dataset

Machine learning

Over-sampling

Under-sampling

ABSTRACT

Sentiment analysis is a popular natural language processing task that recognizes the opinions or feelings of a piece of text. Microblogging platforms such as Twitter are a valuable resource for finding such people's opinions. The majority of Arabic sentiment analysis studies indicated that the data utilized to train machine learning algorithms is balanced. In this paper, we investigated the impact of sampling techniques and classification algorithms on an imbalanced Arabic dataset about people's perceptions of COVID-19, with the majority of opinions reflecting people's fear and stress about the pandemic, and the minority reflecting the belief that the pandemic was a hoax. The experiments concentrated on analyzing the imbalanced learning of Arabic sentiments using over-sampling and under-sampling techniques on seven single machine learning algorithms and two common ensemble algorithms from the bagging and boosting families, respectively. Results show that resampling-based approaches can overcome the difficulty of an imbalanced dataset, and the use of over-sampled data leads to better performance than that of under-sampled data. The results also reveal that using oversampled data from synthetic minority over-sampling technique (SMOTE), borderline-SMOTE, or adaptive synthetic sampling with random forest classifier is the most effective in addressing this classification problem, with F1-score value of 0.99.

This is an open access article under the [CC BY-SA](https://creativecommons.org/licenses/by-sa/4.0/) license.



Corresponding Author:

Maisa J. Al-Khazaleh

Department of Basic Sciences, Faculty of Science, The Hashemite University

Zarqa, Jordan

Email: maisa@hu.edu.jo

1. INTRODUCTION

The corona epidemic invites people to use social media to share their thoughts, hold discussions, and express their feelings about the pandemic. Microblogging, such as Twitter, is a popular and widely used social media platform in which people submit short blogs about daily life occurrences. One of the most important occurrences that has spread worldwide is the coronavirus (COVID-19), where some people express their anxiety and tension about COVID-19 while others simply believe it is a rumor. With the large number of tweets regarding the epidemic, a detailed observation at how people felt during the pandemic can be made. Despite the significant increase in the number of infections and deaths, some people still believe that COVID-19 is a rumor. Twitter is a rich source of data for scholars who want to study emotions in depth [1].

Sentiment analysis (SA), also known as opinion mining, is an important field in natural language processing (NLP) that determines the direction of emotions represented in a text as positive, negative, or neutral and provides a suggestion about the feelings in a text. It becomes an indispensable tool for developing

recommendation systems, monitoring brands, or assessing survey replies or evaluations, and it aids in spotting significant concerns in real time [2]. Furthermore, SA can be applied at several levels, such as document, paragraph, sentence, or aspect [3]. In addition, SA is used to build models that can predict and classify sentiments from sentences using text analysis and machine learning (ML) approaches [4].

The classification task's success is influenced by the quality of data and ML method. Given the availability of the requisite NLP tools for English texts, several researchers concentrated on SA for the English Language [5]. SA can take one of three approaches: ML, lexicon-based, or a mixed approach [6]. The supervised learning strategy in ML employs labeled datasets to train the model to predict sentiments, while unsupervised learning focuses on unlabeled data to identify a possible structure [7]. Support vector machines (SVM), naive bayes (NB), and k-nearest neighbors (KNN) are among the most commonly used ML techniques in text classification.

However, class imbalance in datasets is a critical issue that affects the performance of these methods. The imbalanced dataset problem is experienced when various classes have substantially varied proportions in the dataset; the majority classes account for a large part of the data, while minority classes have a small amount of data [8]. When gathering data for SA, we may encounter data imbalance; if the nature of this data is ignored, ML algorithms will be biased toward the majority class, resulting in the misclassification of the minority class compared with the majority class. To address class imbalance, many solutions have been developed, which may be at the data or algorithm level.

Resampling is used in data-level approaches to balance the dataset; typical examples include over-sampling techniques, under-sampling techniques, and combinations of both. Algorithm-level strategies focus on altering the insight of learning algorithms that use cost-sensitive learning to generalize in favor of the minority class [9]. The ensemble learning approach combines cost-sensitive learning with performance-enhancing algorithms, such as bagging, boosting, and stacking. Aside from these strategies, one of the most important concerns to address when dealing with data imbalance is the metrics used to evaluate the model.

In the classification process for a highly imbalanced dataset, the use of some measures such as accuracy can be misleading because the classifier may always predict the majority class without performing any analysis and may have a high accuracy score, which is clearly erroneous. Despite being the world's sixth most spoken language [10], Arabic does not receive great interest like English because of its dialectal variety and complex structure and the limitations in the annotated Arabic datasets. Classical Arabic or quranic Arabic; modern standard Arabic, which is the formal written and spoken Arabic Language taught in the Arab world; and dialectical or colloquial Arabic, which is informal and does not follow any grammatical rules [3], are the three categories of Arabic Language.

Numerous studies on Arabic SA have been conducted on balanced datasets, while the imbalanced learning of Arabic sentiment has received little attention. To the best of our knowledge, SMOTE is the most common sampling technique used to balance Arabic datasets, and when using the ensemble approach, the random forest (RF) classifier is predominantly used in most research. In addition, most of the existing research was conducted on small imbalanced datasets.

To address this research gap, we performed our analysis on an imbalanced Arabic dataset that has 15779 samples, using various sampling and ensemble techniques to address the imbalance problem. This study aims to improve the imbalanced learning of Arabic sentiments. Experiments were conducted at several levels, and resampled data was used to train different single and ensemble classifiers to study the effects of these techniques and find an optimized classifier that can distinguish the largest number of negative Arabic tweets while maintaining the performance of the model in terms of positive tweets. The contributions of this study are summarized as follows: i) the methods used for SA with imbalanced Arabic datasets were compared and ii) the impact of using sampling techniques to train single and ensemble classifiers and find the best classifier using different evaluation metrics was analyzed.

The rest of the paper is organized as follows: section 2 discusses and summarizes the research papers from the literature related to imbalanced Arabic learning. Section 3 introduces the proposed method and the approaches adopted for the current classification task. In section 4, the conducted experiments are described and the results are discussed. Finally, conclusion is provided in section 5.

2. LITERATURE REVIEW

Most of the existing Arabic SA research was conducted on balanced datasets. Furthermore, classification algorithms can perform better on balanced datasets than on imbalanced datasets, so re-sampling techniques, including under-sampling and over-sampling, have been adopted to balance datasets [11]. Mountassir *et al.* [12] investigated the impact of four under-sampling techniques, namely, remove similar, remove farthest, remove by clustering, and random remove. They conducted experiments on two Arabic and

one English imbalanced datasets. They used NB, SVM, and KNN classifiers with a g-performance metric to evaluate the results. The random under-sampling (RU) technique yielded the best results.

Al-Azani and El-Alfy [13] conducted three studies to address the imbalanced dataset problem. In 2017, they studied the impact of the SMOTE over-sampling technique on an imbalance dataset of tweets in dialectal Arabic, and the experiments were evaluated on basic and ensemble classifiers using the accuracy, F1, precision, and recall evaluation metrics. The results show that using SMOTE with ensemble classifiers increased the performance by 15% compared with the baseline experiments. In 2018, Al-Azani and El-Alfy [13] studied the impact of the bootstrap aggregating algorithm with SMOTE on a concatenated version of imbalanced Arabic dialectal twitter datasets, namely, Syrian tweets [14], Arabic sentiment tweets dataset (ASTD) [15], ArTwitter [16], tweet corpus for subjectivity and sentiment analysis (SSA) [17], and Semeval-2017 [18]. NB, KNN, and decision tree (DT) classifiers were evaluated in terms of F1, Matthews's correlation coefficient (MCC), geometric mean (GM), and area under the receiver operator characteristic curve (AUC) values with different imbalance ratios. The experimental results show that balanced bagging classifiers produced the best results [19].

In 2020, El-Alfy and Al-Azani [20] investigated the performance of nine ML classifiers on highly imbalanced Arabic tweet datasets using neural word embedding and over-sampling techniques. The performance is discussed in terms of various measures, like AUC, GM, and F1. The results reveal that the stochastic gradient descent (SGD) classifier with over-sampling exhibits the best performance, achieving the highest GM value.

Al-Sorori *et al.* [21] analyzed the impact of using synthetic minority over-sampling technique and edited nearest neighbors (SMOTENN) to balance an Arabic dataset collected from Twitter. They used Word2Vec word embedding with various single and ensemble ML classifiers. Their experiments showed that using SMOTENN improves F1 score for both single and ensemble classifiers where the best result obtained by nuSVM produced an average F1 score value of 99.07.

Khalifa and Elnagar [22] focused on studying the performance of their Twitter dataset in its imbalanced and balanced versions using term frequency-inverse document frequency (TF-IDF) and word embeddings. They conducted a comparative evaluation of the gradient boosting, logistic regression (LR), nearest centroid, DT, multinomial NB, SVM, XGBoost (XGB), RF, and AdaBoost classifiers and investigated the performance of the MLP and condensed nearest neighbor (CNN) deep learning classifiers. The LR classifier using TF-IDF produced the best F1 result of 87.71% on imbalanced training data.

Addi and Ezzahir [23] conducted a study to address the imbalance problem in the hotel Arabic-reviews dataset [24] using various under-sampling and over-sampling techniques on SVM, NB, and RF classifiers. They evaluated their results using accuracy and F1 metrics and concluded that under-sampling techniques, namely, edited nearest neighbors (ENN), the repeated ENN rule, tokek links, and the neighborhood cleaning rule, showed the best results among the sampling techniques. Recently, Al-Hashedi *et al.* [25] used the COVID-19 Arabic tweets dataset to investigate the effect of the Word2Vec word embedding and SMOTE over-sampling techniques on several single and ensemble ML classifiers. Their experiments showed that ensemble classifiers and SMOTE outperform base classifiers without SMOTE in terms of F1 score.

In the work of Obiedat *et al.* [26], different versions of an imbalanced dataset about restaurant reviews collected from the Jeeran website were examined using different over-sampling techniques, such as SMOTE, SVM-SMOTE, adaptive synthetic sampling (ADASYN), and borderline-SMOTE (BSMOTE). The authors proposed an approach that combines particle swarm optimization (PSO) and SVM and compared the results of this hybrid approach (PSO-SVM) with different single and ensemble classifiers such as SVM, LR, RF, DT, KNN, and XGBoost. They reported that the proposed PSO-SVM approach is superior to the aforementioned classifiers. The best result was obtained from version 3 of their dataset using BSMOTE with a GM value of 0.81.

In this paper, we propose the use of several single and ensemble classifiers that identify the majority of negative Arabic tweets while maintaining the model's performance in terms of positive tweets to improve the imbalanced learning of Arabic sentiments. Ridge classifiers, LR, SGD, SVM, DT, KNN, and Gaussian NB are used as single classifiers, while RF and AdaBoost are employed as ensemble classifiers. Table 1 presents a comparison between the previously mentioned studies on the problem of imbalanced Arabic datasets.

The comparison highlights the most important points in these studies in terms of the dataset used, the imbalance ratio, and the resampling methods. The imbalance ratio (IR) is the ratio of the number of samples in the majority class to that in the minority class [27]. IR indicates the extent of imbalance in the dataset, that is, the higher the IR is the larger the load of imbalance in the dataset is. In Table 2, we present a comparison between these studies in terms of classifiers, year of publication, evaluation metrics, and their best results.

Table 1. Comparison of classifiers and results of related work

Ref.	Dataset	Negative samples	Positive samples	IR	Resampling techniques
[12]	ACOM (their work)	284	148	0.52	Under-sampling (remove similar, remove farthest, remove by clustering, and random removal)
	DS1				
	DS2	462	462	1.63	
	SINAI	145	1701	11.73	
[13]	Syrian tweets	1350	488	2.77	Over-sampling (SMOTE)
[19]	DS1	377	724	1.92	Over-sampling (SMOTE)
	DS2	242	724	2.99	
	DS3	100	724	7.24	
[20]	Syrian tweets	1350	448	3.01	Over-sampling (random over-sampling (RO), SMOTE, adaptive synthetic)
[21]	COVID-19 Twitter dataset	530	240	2.21	Combination of over-sampling and under-sampling (SMOTENN)
[22]	Twitter Arabic dialect dataset and TADE dataset				No resampling techniques (the authors equate the category samples with the samples of lower sentiment scores from TADE dataset)
[23]	Hotel Arabic-reviews dataset (HARD)	4743	25256	5.32	Under-sampling techniques (RU, NearMiss, cluster centroids CNN, repeated edited nearest neighbor rule, neighborhood cleaning rule, tomek links) over-sampling techniques (SMOTE, BSMOTE)
[25]	COVID-19 tweets	363	663	1.83	Over-sampling (SMOTENC)
[26]	Their collected dataset restaurant reviews from Jeeran website)	640	2150	3.36	Over-sampling (SMOTE, SVM-SMOTE, ADASYN and BSMOTE)
Our work	Twitter Arabic dataset about COVID-19	3603	12176	3.38	Under-sampling techniques (RU, CNN, and one-sided selection (OSS) over-sampling techniques (RO, SMOTE, BSMOTE and ADASYN

Table 2. Comparison of classifiers and results of related work

Ref	Year	Single classifiers	Ensemble classifiers	Evaluation metrics	Best results
[12]	2012	NB, SVM and KNN	None	G-performance	DS1: SVM using RU achieved (72.1) DS2: NB using RU achieved (67.9) SINAI: NB using RU achieved (87.6)
[13]	2017	SGD, LR, and Gaussian NB	Bagging, boosting, and stacking	Precision, recall, F1, and accuracy	Stacking-based ensemble using SMOTE achieved accuracy value of (85.28) and (63.95) F1 value
[19]	2018	NB, KNN and DT	None	F1, MCC, GM and AUC	Using bagging classifier with SMOTE DS1: DT achieved F1 value (0.7436) DS2: KNN achieved F1 value (0.7490) DS3: KNN achieved F1 value (0.8402)
[20]	2020	SGD, LR, linear-SVM, GNB, nearest neighbor, and DT	RF, gradient boosting, soft-voting, and stacking	Confusion matrix, ACC, AUC, APR, F1, FMI, MCC and GM	SGD classifier using over-sampling techniques achieved GM value (0.781)
[21]	2021	nuSVM, LSVM, SGD, LRCV, and BNB)	RF and voting	Accuracy, recall, F1, and precision	nuSVM using SMOTTEN achieved F1 value (99.07)
[22]	2020	LR, nearest centroid, DT, multinomial NB, SVM) deep learning (MLP, and CNN)	Gradient boosting, XGB, RF, AdaBoost	F1	LR using TF-IDF achieved F1 value of (87.71%) on imbalanced training data and MLP achieved F1 value of (86.16) on balanced training data
[23]	2020	SVM, NB	RF	F1 and accuracy	SVM RF using ENN or RENN achieved F1 value of (97%)
[25]	2022	SVM, LSVM, SGD, LRCV, and BNB	RF and Voting	Accuracy, recall, precision and F1	Nu SVM using SMOTENC achieved F1 value of (93.48)
[26]	2022	SVM, LR, DT, KNN	XGBoost and RF	accuracy, F-measure, GM AUC	Version 3 of their dataset using BSMOTE achieved GM value of (0.81)
Our work		Ridge classifier, LR, SGD, SVM, DT, KNN, and gaussian naïve bayes (G-NB)	RF and AdaBoost	Accuracy, recall, precision, F1-score, GM, AUC, and confusion matrix	oversampled data from SMOTE, BSMOTE, and ADASYN with the RF classifier with an F1 value of (0.99)

3. PROPOSED METHOD

The aim of this research is to build a robust model to for predicting Arabic sentiment for COVID-19 tweets. Some people believe that COVID-19 is a real and dangerous virus, while others believe that it is just a rumor. The steps of our research methodology are provided in Figure 1, and the details of these steps are presented in the following subsections.

As shown in Figure 1, the first step for conducting the proposed model starts by collecting Arabic COVID-19 related tweets using a Twitter API based on pre-specified search terms. Then, these tweets were saved in a CSV file. Next, the collected data was cleaned and manually labeled. After that, we used CountVectorizer to extract the features and to represent the input to the classifier. In the next step, the dataset is divided into training and testing sets, with 90% for training set and 10% for testing set, as illustrated in Table 3. The classification accuracy metric and other metrics were used, such as recall, precision, F1-score, GM, and AUC. In addition, a confusion matrix was built to provide an overview of the mislabeled data that the classifier provides and obtain an improved view of how well our model performs.

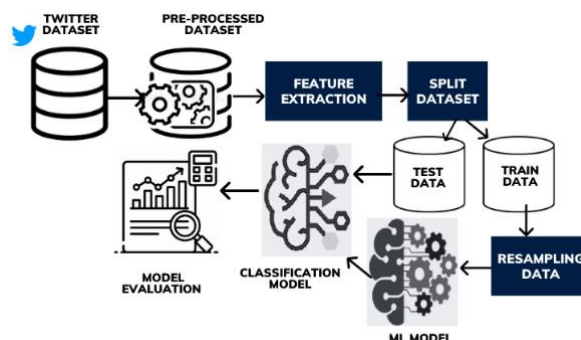


Figure 1. Research methodology workflow

Table 3. Units for magnetic properties

	Negative	Positive	
Testing	360	1,218	
Training	3,243	10,958	
Total	3,603	12,176	15,779

3.1. Data collection

Arabic tweets were collected. The collected tweets were written in standard or dialect Arabic. We finally obtained 15,779 Arabic tweets related to COVID-19 from 75,794 tweets using the Tweepy python library and a Twitter API [28]. Search queries were determined according to the most frequently used words about COVID-19 among people on social media platforms. The collected data was saved in a CSV file for the preprocessing phase. The number of tweets decreased because of the presence of duplicated tweets (i.e., retweets), which were excluded with tweets that do not represent feelings, such as news and decisions.

3.2. Data preprocessing

The gathered tweets are not clean. The tweets contained noise, such as stop words and special characters, requiring the application of preprocessing techniques to prepare the data for the classification process. First, duplicate tweets were removed. Then, all non-Arabic words, letters, URLs, names, hash tags, numbers, diacritics, and special characters were removed using a python regular expression. Some tweets contained words with repeating letters for emphasis; these words were handled by returning them to their correct format by removing duplicate letters [29]. To improve the accuracy of the predictive model, normalization was used to unify analogous letters [30]. In the preprocessing phase, we did not apply any stemming because most of the collected tweets were written using dialect language, which complicates the stemming process.

3.3. Data annotation

Given the complexity of the morphology and the diversity of the Arabic dialect, we manually annotated the dataset. Three sentiment labels were given to the tweets, namely, positive, negative, or neutral. A negative sentiment is given to tweets that reflect people's views of COVID-19 as a lie or rumor; a positive sentiment is given to tweets that reflect people's beliefs about the existence of COVID-19 and the necessary procedures to protect themselves; and a neutral label is given to tweets that do not carry any kind of emotion, such as news, facts, and decisions. The dataset contains 12,176 positive tweets out of the 15,779 tweets, and the remaining 3,603 tweets were labeled as negative. Figure 2 shows the distribution of the dataset in terms of positive and negative, providing evidence of an imbalanced dataset.

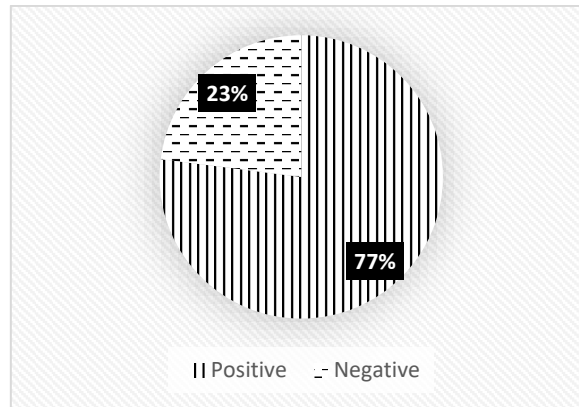


Figure 2. Dataset distribution

3.4. Data representation

ML algorithms cannot directly deal with texts, so the text must be tokenized and then encoded into numerical representation that can be processed by ML algorithms. In this work, we used the CountVectorizer technique, which converts text into word count vectors, where each unique word has a unique dimension. The resulting sparse encoded vectors are transformed back into arrays that contain the occurrences of each word [31].

3.5. Model implementation

The model implementation process started by reading the dataset from the CSV file into a pandas data frame and extracting features using CountVectorizer to obtain 37,501 input features. To run the experiments, we utilized seven single classifiers, namely, the ridge classifier, LR, SGD, SVM, DT, KNN, and Gaussian NB (G-NB). We also used two ensemble classifiers, namely, RF and AdaBoost. RF, which is a collection of DTs, is one of the most popular bagging techniques and it provides better predictive performance than a single DT classifier because it attempts to reduce the variance and the chance of classifier overfitting classifier [32], while AdaBoost, which belongs to boosting algorithms, has received great attention in classification problems.

In this work, we used AdaBoost (AdaBST) with a DT classifier as a weak learner for training. However, AdaBST relies on weighted training samples and iteratively increasing weights for incorrectly classified samples and reducing weights for correctly classified samples to reduce the total error and ensure the accurate prediction of incorrectly classified samples [33]. In addition, we conducted three experiments. The first experiment studied the performance of the basic default behavior of ML classifiers with GridSearchCV to automatically select the optimal parameters for the classifier with a 10-fold cross validation, without applying any resampling techniques. The second experiment used several over-sampling techniques, including random over-sampling (RO), SMOTE, borderline-SMOTE (BSMOTE), and ADASYN. The last experiment is similar to the second one, but it was conducted with under-sampling techniques, including condensed nearest neighbors (CNN), one-sided selection (OSS), and random under-sampling (RU). However, in RO technique, more random samples are added to the minority class in the training dataset to match the number in the majority class. This technique is performed by duplicating the minority class samples multiple times to complete the training dataset [34]. It is simple and fast but does not use any heuristics. Furthermore, no information is lost but the possibility of overfitting may increase.

While SMOTE technique increases the number of samples in the minority class of the imbalanced dataset by finding k nearest neighbors of random samples to add more synthetic instances on the basis of similarities in the feature space. SMOTE should only be applied on the training data to avoid creating new samples that might appear in the testing data, which could provide misleading results.

BSMOTE is an extended version of SMOTE. In this technique, borderline minority class points that are near the decision surface are used to add new samples instead of using normal minority points that are far from the borderline [35]. ADASYN uses data points of a minority class that have many neighbors from the majority class. These points are called “hard to learn” data points, which are used to generate new synthetic samples using a probability density function [11]. On the other hand, CNN is one of the condensation methods [36] that condense the original dataset by looking for a minimal consistent subset that does not result in performance degradation [37] while OSS is a modified version of CNN introduced by Kubat and Matwin [38]; which combines the CNN rule and totem links. This technique creates a new balanced dataset that includes

all minority class samples, removing noise, borderline, and redundant samples from the majority class and retaining the normal majority class samples. RU randomly eliminates samples from the majority class to balance the training dataset distribution. This technique is similar to the RO technique in simplicity and speed and also does not use heuristics. However, RU may lead to the loss of valuable information to fit the model.

3.6. Evaluation metrics

Evaluating the performance of ML models on imbalanced datasets using accuracy is insufficient to judge the quality of the model because of the accuracy paradox. In this work, the quality of the classifier output was evaluated by several metrics, namely, precision (prec.), recall (rec.), F1-score, GM, and AUC, to consolidate a reliable unbiased evaluation. Recall is the measure of actual positives that are predicted correctly by the ML model out of all the positives. Precision is the measure of actual positives out of all the positives predicted correctly by the model. GM and F1-score combine precision and recall metrics. GM measures the accuracy on the positive and negative class samples [38], while F1 is the harmonic mean of precision and recall. The receiver operator characteristic (ROC) curve is an easy-to-interpret graph that visualizes the performance of a binary classifier; ROC summarizes the performance in one value called AUC which is the area under the receiver operator characteristic curve.

The AUC determines the classifier's ability to distinguish between positive and negative classes. We can use the AUC to compare the classifiers models, where a good model has an AUC value near 1, which indicates that the model can predict negative and positive labels correctly, while bad models have an AUC value near to 0 [39]. To visualize the performance of the classifier, we also plotted a confusion matrix that shows the truly classified and miss-classified tweets for both positive and negative classes [40].

4. RESULTS AND DISCUSSION

The main goal of the experiments in this work is to balance the dataset using various sampling approaches and then apply and compare several classifiers to determine the best-functioning classifier that can differentiate as many negative Arabic tweets as possible while maintaining the model's performance with respect to positive tweets. The positive tweets in the dataset are referred to the majority class and the negative tweets as the minority class. Initially, in the baseline experiment, the entire dataset was passed to the classifier after splitting it into the training and testing sets. Grid search was applied to perform hyper parameter optimization for the learning algorithms. A dictionary of parameters for the grid search was defined to find the best combinations that were optimized by 10-fold cross-validation over a parameter grid. Two functions were applied; the "fit" function was applied to train the classifier with optimal parameters, and the "predict" function was applied to test the classifier.

In Table 4, we highlight the best results in this experiment; the results indicate that ensemble classifiers outperform single classifiers. LR, SVM, SGD, and DT exhibited a comparable performance, but the best among them is DT. However, DT classifier achieved the highest F1 score of 0.84, while the GNB and ridge classifiers obtained the lowest values in all evaluation metrics.

Table 4. Results of experiment 1: baseline performance of classifiers on the original imbalanced dataset with test size 0.1

	Model	Prec.	Rec.	F1	Acc.	GM	AUC	Best parameters
Single	Ridge	0.81	0.65	0.72	0.61	0.56	0.59	{'alpha': 0.5, 'max_iter': 11, 'normalize': true}
	LR	0.82	0.84	0.83	0.74	0.58	0.68	{'C': 100, 'dual': false, 'max_iter': 100, 'penalty': 'l2', 'solver': 'lbfgs'}
	SVM	0.82	0.81	0.81	0.71	0.58	0.67	{'C': 20, 'gamma': 0.001, 'kernel': 'linear'}
	SGD	0.82	0.81	0.82	0.73	0.60	0.65	{'alpha': 0.01, 'max_iter': 5, 'penalty': 'l2'}
	DT	0.83	0.85	0.84	0.75	0.60	0.66	{'criterion': 'gini', 'max_depth': 50, 'max_features': 'auto', 'min_samples_leaf': 1, 'min_samples_split': 100}
Ensemble	KNN	0.78	0.81	0.80	0.69	0.48	0.59	{'leaf_size': 10, 'n_neighbors': 3, 'weights': 'distance'}
	G-NB	0.80	0.66	0.72	0.62	0.56	0.57	{'var_smoothing': 1e-06}
	RF	0.81	0.93	0.87	0.78	0.54	0.75	bootstrap: true, 'criterion': 'gini', 'max_depth': none, 'max_features': 'auto', 'min_samples_leaf': 1, 'min_samples_split': 2, 'n_estimators': 100
	AdaBST	0.79	0.97	0.87	0.78	0.40	0.70	'learning_rate': 0.8, 'n_estimators': 54

The confusion matrices for the best single (DT) and ensemble classifiers (RF) in Figures 3(a) and (b) reflect the class imbalance according to the poor scores of F1, recall, and precision for negative tweets (minority class) compared with positive tweets (majority class). Figures 3(a) and (b) show that these models were confused when they made predictions.

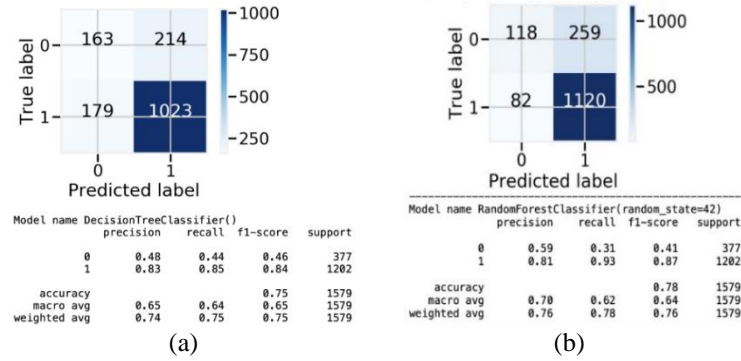


Figure 3. The performance results of baseline experiment: (a) decision tree classification report, confusion matrix and (b) random forest classification report, confusion matrix

Figure 4 shows the AUC values for all the classifiers. RF was the best classification model in this experiment; the highest value was achieved by RF. The results show that resampling techniques are needed to represent the dataset for evaluating classifiers well. The next experiment was conducted using over-sampling methods to balance the dataset and GridSearchCV to fine tune the parameters. In this experiment, we applied RO, SMOTE, BSMOTE, and ADASYN for each classifier.

In Table 5, the performance results show an improvement after over-sampling the training set and RF exhibits superior performance for all metrics compared with other classification models. It is shown from Table 5 that the ridge classifier using SMOTE achieves a high F1 score value of 0.98 compared with other single classifiers, while KNN and AdaBST were the worst classifiers. The results also reveal that RO has a lower performance than SMOTE, BSMOTE, and ADASYN in terms of F1 score. In the last experiment, we used the same strategy as the second one but with under-sampling techniques to study the impact of applying under sampling on the dataset.

Table 5. Results of experiment 2: using over-sampling techniques

Model	RS						SMOTE					
	Prec.	Rec.	F1	Acc.	GM	AUC	Prec.	Rec.	F1	Acc.	GM	AUC
Ridge	0.97	0.85	0.90	0.86	0.88	0.94	1.00	0.97	0.98	0.98	0.98	1.00
LR	0.98	0.89	0.93	0.90	0.91	0.94	0.97	0.92	0.95	0.92	0.92	1.00
SVM	0.96	0.86	0.91	0.87	0.88	0.94	0.91	0.97	0.94	0.90	0.82	0.94
SGD	0.97	0.88	0.92	0.89	0.90	0.94	0.91	0.98	0.94	0.91	0.81	0.91
DT	0.85	0.94	0.89	0.83	0.68	0.52	0.98	0.93	0.96	0.93	0.94	0.94
KNN	0.95	0.73	0.83	0.77	0.80	0.94	0.85	0.95	0.89	0.83	0.65	0.81
G-NB	0.99	0.94	0.96	0.95	0.96	0.94	0.96	0.98	0.97	0.96	0.93	1.00
RF	0.94	0.99	0.96	0.94	0.88	0.99	0.99	1.00	0.99	1.00	1.00	1.00
AdaBST	0.87	0.51	0.65	0.57	0.62	0.68	0.87	0.51	0.65	0.57	0.62	0.68
	BSMOTE						ADAYSN					
Ridge	0.89	0.96	0.92	0.88	0.77	0.91	0.89	0.96	0.92	0.88	0.76	0.91
LR	0.88	0.99	0.93	0.89	0.75	0.91	0.88	0.99	0.93	0.89	0.76	0.91
SVM	0.91	0.97	0.94	0.90	0.81	0.94	0.90	0.97	0.94	0.90	0.81	0.94
SGD	0.90	0.97	0.94	0.90	0.80	0.91	0.90	0.98	0.94	0.90	0.80	0.91
DT	0.98	0.94	0.96	0.94	0.94	0.93	0.98	0.92	0.95	0.93	0.94	0.91
KNN	0.84	0.95	0.89	0.83	0.65	0.93	0.84	0.94	0.89	0.82	0.64	0.91
G-NB	0.96	0.98	0.97	0.95	0.93	0.92	0.97	0.97	0.97	0.95	0.93	0.91
RF	1.00	0.99	0.99	0.99	0.99	1.00	1.00	0.99	0.99	0.99	0.99	1.00
AdaBST	0.86	0.60	0.71	0.62	0.64	0.69	0.86	0.59	0.70	0.62	0.64	0.69

The results of applying under-sampling are illustrated in Table 6. The results show that using random under-sampling and condensed nearest neighbors (CNN) produces high precision values and low recall values. Therefore, we have a picky classifier that did not predict many tweets as positive (i.e., people believe in COVID-19) and miss-predicted many actual positive tweets. In addition, the results show that the behavior of classifiers using OSS is better than that using RO and CNN in terms of F1, accuracy, GM, and AUC.

We studied AdaBST performance using grid search for parameter optimization and made some manual visualizations to measure the effect of the number of base estimators (n estimators) and learning rate hyper parameters, as shown in Figure 7. Figure 8 shows a comparison between all the classifiers with all the sampling techniques in terms of F1 score. The results show that over-sampling outperforms random and CNN under-sampling techniques.

Table 6. Results of experiment 3: using under-sampling techniques

Model	RU					
	Prec.	Rec.	F1	Acc.	GM	AUC
Ridge	0.98	0.71	0.83	0.77	0.83	0.91
LR	0.97	0.73	0.84	0.78	0.83	0.90
SVM	0.98	0.72	0.83	0.78	0.83	0.91
SGD	0.98	0.71	0.83	0.77	0.83	0.91
DT	0.94	0.50	0.66	0.60	0.67	0.90
KNN	0.91	0.65	0.76	0.68	0.71	0.84
G-NB	0.99	0.72	0.83	0.78	0.83	0.84
RF	1.00	0.59	0.74	0.69	0.77	0.95
AdaBoost	0.94	0.32	0.48	0.47	0.55	0.70
Model	CNN					
	Prec.	Rec.	F1	Acc.	GM	AUC
Ridge	0.98	0.77	0.86	0.81	0.85	0.93
LR	0.98	0.82	0.89	0.85	0.88	0.93
SVM	0.97	0.77	0.86	0.81	0.85	0.92
SGD	0.97	0.79	0.87	0.82	0.86	0.93
DT	0.93	0.35	0.51	0.49	0.57	0.63
KNN	0.87	0.82	0.85	0.77	0.71	0.80
G-NB	0.99	0.82	0.90	0.86	0.90	0.89
RF	0.89	0.96	0.92	0.88	0.78	0.95
AdaBoost	0.90	0.41	0.56	0.52	0.59	0.70
Model	OSS					
	Prec.	Rec.	F1	Acc.	GM	AUC
Ridge	0.97	0.83	0.90	0.85	0.88	0.95
LR	0.98	0.88	0.93	0.89	0.90	0.95
SVM	0.97	0.83	0.89	0.85	0.87	0.94
SGD	0.97	0.87	0.92	0.88	0.89	0.94
DT	0.99	0.99	0.99	0.99	0.99	0.99
KNN	0.83	0.96	0.89	0.82	0.61	0.85
G-NB	0.99	0.93	0.96	0.94	0.96	0.95
RF	0.87	1.00	0.93	0.88	0.72	0.98
AdaBoost	0.80	0.91	0.85	0.76	0.52	0.71

Ridge, LR, SVM, SGD, RF, and AdaBST have a good performance when applied with any under-sampling approaches, as shown in Figure 5. The best classifier performance was exhibited by the DT using OSS, with a value of 0.99 for both AUC and GM, while the CNN decreased the performance of the DT. Interestingly, AdaBST performed well without using any sampling technique, as shown in Figure 6.

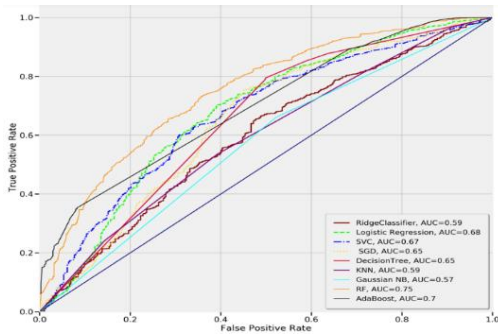


Figure 4. AUC values in baseline experiment

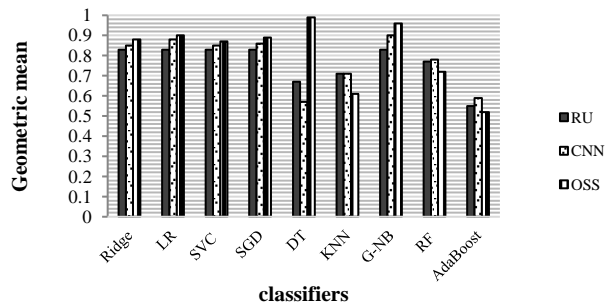


Figure 5. Geometric mean of classifiers using under-sampled data

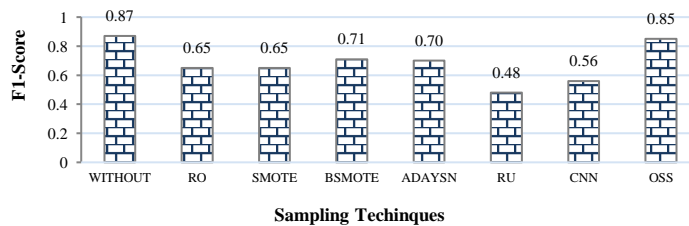


Figure 6. Adaboost performance in all experiments in terms of F1-score

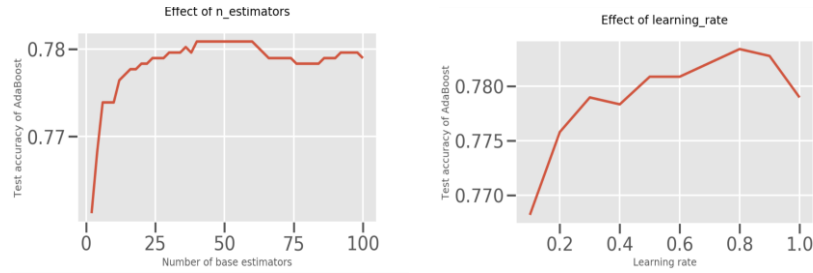


Figure 7. Effect of Adaboost hyper parameters

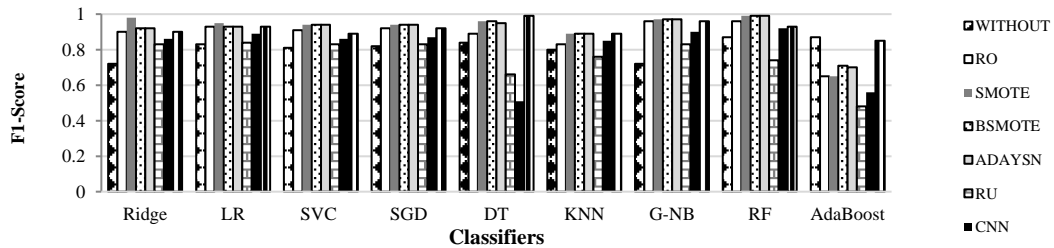


Figure 8. Comparison of different sampling techniques performance with classifiers

The results of our experiments show the benefits of balancing the training dataset before applying the classifier. However, over-sampling techniques outperform under-sampling techniques because the latter depends on eliminating samples, which may lead to the exclusion of important features that would negatively affect the performance of the classifier models. Considering all the criteria, we found that the best model is RF with SMOTE, BSMOTE, and ADASYN over-sampling, which was able to fully distinguish positive and negative labels and achieve the goal.

5. CONCLUSION

Although there are numerous studies on Arabic SA using ML algorithms have been conducted, most of these studies them deal with balanced datasets. In the context of imbalanced classification, most studies used small datasets. This paper gives provides an overview on the impact of using a data-level sampling approach within a classification task before training single and ensemble classifiers. These methods turn transform the an imbalanced dataset into a balanced dataset. The results indicate that the models performed poorly on the imbalanced dataset, while the and a balanced dataset tends to increase the classification accuracy.

Our experiments were conducted on single, bagging-based, and boosting-based ensemble classifiers. In addition, we focused on how resampling techniques specifically affect the performance of both single and ensemble classifiers. The experiments revealed that over-sampling and under-sampling provide good results for various classifiers when evaluated using different metrics, such as F1, accuracy, and AUC, compared with the poor performance using an imbalanced dataset.

The over-sampling approaches (SMOTE, BSMOTE, ADASYN) produced superior results, while the OSS under-sampling approach is the best among the under-sampling approaches. However, over-sampling approaches outperform under-sampling approaches because no data is lost and a considerable feature set is provided in over-sampled data compared with the under-sampled data, resulting in the enhanced performance of the classifiers. The RF ensemble classifier using SMOTE, BSMOTE, or ADASYN over-sampled data exhibits good efficiency with F1 value of 0.99.

Surprisingly, the performance of the AdaBST classifier was ambiguous. AdaBST produced better results on the original dataset with hyper parameter tuning than those using sampling approaches. We argue that the high dimensionality of the data space, the existence of noise, and base learners influenced AdaBST's performance. In future work, the insufficient information on the performance of AdaBST can be addressed through further investigation.




REFERENCES

- [1] M. Al-Khazaleh, M. Alian, M. Biltawi, and B. Al-Hazaimeh, "Sentiment Analysis for People's Opinions about COVID-19 Using LSTM and CNN Models," *International journal of online and biomedical engineering*, vol. 19, no. 1, pp. 135–154, Jan. 2023, doi: 10.3991/ijoe.v19i01.35645.
- [2] D. Zimbra, A. Abbasi, D. Zeng, and H. Chen, "The state-of-the-art in twitter sentiment analysis: A review and benchmark evaluation," *ACM Transactions on Management Information Systems*, vol. 9, no. 2, pp. 1–29, Jun. 2018, doi: 10.1145/3185045.
- [3] R. Duwairi and M. El-Orfali, "A study of the effects of preprocessing strategies on sentiment analysis for Arabic text," *Journal of Information Science*, vol. 40, no. 4, pp. 501–513, Aug. 2014, doi: 10.1177/0165551514534143.
- [4] B. Yu, J. Zhou, Y. Zhang, and Y. Cao, "Identifying restaurant features via sentiment analysis on yelp reviews," *arXiv preprint arXiv:170908698*, 2017.
- [5] A. Ghallab, A. Mohsen, and Y. Ali, "Arabic Sentiment Analysis: A Systematic Literature Review," *Applied Computational Intelligence and Soft Computing*, vol. 2020, pp. 1–21, Jan. 2020, doi: 10.1155/2020/7403128.
- [6] M. Birjali, M. Kasri, and A. Beni-Hssane, "A comprehensive survey on sentiment analysis: Approaches, challenges and trends," *Knowledge-Based Systems*, vol. 226, p. 107134, Aug. 2021, doi: 10.1016/j.knosys.2021.107134.
- [7] A. D'Andrea, F. Ferri, P. Grifoni, and T. Guzzo, "Approaches, Tools and Applications for Sentiment Analysis Implementation," *International Journal of Computer Applications*, vol. 125, no. 3, 2015.
- [8] K. Ghosh, A. Banerjee, S. Chatterjee, and S. Sen, "Imbalanced Twitter Sentiment Analysis using Minority Oversampling," in *2019 IEEE 10th International Conference on Awareness Science and Technology, iCAST 2019 - Proceedings*, IEEE, Oct. 2019, pp. 1–5, doi: 10.1109/ICAwST.2019.8923218.
- [9] Q. Wang, Z. H. Luo, J. C. Huang, Y. H. Feng, and Z. Liu, "A novel ensemble method for imbalanced data learning: Bagging of extrapolation-SMOTE SVM," *Computational Intelligence and Neuroscience*, vol. 2017, pp. 1–11, 2017, doi: 10.1155/2017/1827016.
- [10] Eberhard, David M., G. F. Simons, and C. D. Fennig, *Ethnologue: Languages of Africa and Europe*, Twenty-Fif. SIL International, 2022.
- [11] H. He and E. A. Garcia, "Learning from imbalanced data," *IEEE Transactions on Knowledge and Data Engineering*, vol. 21, no. 9, pp. 1263–1284, Sep. 2009, doi: 10.1109/TKDE.2008.239.
- [12] A. Mountassir, H. Benbrahim, and I. Berrada, "An empirical study to address the problem of unbalanced data sets in sentiment classification," in *Conference Proceedings - IEEE International Conference on Systems, Man and Cybernetics*, IEEE, Oct. 2012, pp. 3298–3303, doi: 10.1109/ICSMC.2012.6378300.
- [13] S. Al-Azani and E. S. M. El-Alfy, "Using Word Embedding and Ensemble Learning for Highly Imbalanced Data Sentiment Analysis in Short Arabic Text," *Procedia Computer Science*, vol. 109, pp. 359–366, 2017, doi: 10.1016/j.procs.2017.05.365.
- [14] S. M. Mohammad, M. Salameh, and S. Kiritchenko, "How translation alters sentiment," *Journal of Artificial Intelligence Research*, vol. 55, pp. 95–130, Jan. 2016, doi: 10.1613/jair.4787.
- [15] M. Nabil, M. Aly, and A. F. Atiya, "ASTD: Arabic sentiment tweets dataset," in *Conference Proceedings - EMNLP 2015: Conference on Empirical Methods in Natural Language Processing*, Stroudsburg, PA, USA: Association for Computational Linguistics, 2015, pp. 2515–2519, doi: 10.18653/v1/d15-1299.
- [16] N. A. Abdulla, N. A. Ahmed, M. A. Shehab, and M. Al-Ayyoub, "Arabic sentiment analysis: Lexicon-based and corpus-based," in *2013 IEEE Jordan Conference on Applied Electrical Engineering and Computing Technologies, AEECT 2013*, IEEE, Dec. 2013, pp. 1–6, doi: 10.1109/AEECT.2013.6716448.
- [17] A. Mourad and D. Kareem, "Subjectivity and Sentiment Analysis of Modern Standard Arabic and Arabic Microblogs," in *Proceedings of the 4th Workshop on Computational Approaches to Subjectivity, Sentiment and Social Media Analysis*, 2013, pp. 55–64.
- [18] S. Rosenthal, N. Farra, and P. Nakov, "SemEval-2017 Task 4: Sentiment Analysis in Twitter," in *Proceedings of the Annual Meeting of the Association for Computational Linguistics*, Stroudsburg, PA, USA: Association for Computational Linguistics, 2017, pp. 502–518, doi: 10.18653/v1/s17-2088.
- [19] S. Al-Azani and E. S. M. El-Alfy, "Imbalanced Sentiment Polarity Detection Using Emoji-Based Features and Bagging Ensemble," in *1st International Conference on Computer Applications and Information Security, ICCAIS 2018*, IEEE, Apr. 2018, pp. 1–5, doi: 10.1109/CAIS.2018.8441956.
- [20] E. S. M. El-Alfy and S. Al-Azani, "Empirical study on imbalanced learning of Arabic sentiment polarity with neural word embedding," *Journal of Intelligent and Fuzzy Systems*, vol. 38, no. 5, pp. 6211–6222, May 2020, doi: 10.3233/JIFS-179703.
- [21] W. Al-Sorori et al., "Arabic Sentiment Analysis towards Feelings among Covid-19 Outbreak Using Single and Ensemble Classifiers," in *International Conference on Intelligent Technology, System and Service for Internet of Everything, ITSS-IOE 2021*, IEEE, Nov. 2021, pp. 1–6, doi: 10.1109/ITSS-IOE53029.2021.9615256.
- [22] Y. Khalifa and A. Elnagar, "Colloquial Arabic Tweets: Collection, Automatic Annotation, and Classification," in *2020 International Conference on Asian Language Processing, IALP 2020*, IEEE, Dec. 2020, pp. 163–168, doi: 10.1109/IALP51396.2020.9310507.
- [23] H. A. Addi and R. Ezzahir, "Sampling techniques for Arabic Sentiment Classification: A Comparative Study," in *ACM International Conference Proceeding Series*, New York, NY, USA: ACM, Mar. 2020, pp. 1–6, doi: 10.1145/3386723.3387899.
- [24] A. Elnagar, Y. S. Khalifa, and A. Einea, "Hotel arabic-reviews dataset construction for sentiment analysis applications," in *Studies in Computational Intelligence*, 2018, pp. 35–52, doi: 10.1007/978-3-319-67056-0_3.
- [25] A. Al-Hashedi et al., "Ensemble Classifiers for Arabic Sentiment Analysis of Social Network (Twitter Data) towards COVID-19-Related Conspiracy Theories," *Applied Computational Intelligence and Soft Computing*, vol. 2022, pp. 1–10, Jan. 2022, doi: 10.1155/2022/6614730.
- [26] R. Obiedat et al., "Sentiment Analysis of Customers' Reviews Using a Hybrid Evolutionary SVM-Based Approach in an Imbalanced Data Distribution," *IEEE Access*, vol. 10, pp. 22260–22273, 2022, doi: 10.1109/ACCESS.2022.3149482.
- [27] R. Zhu, Y. Guo, and J. H. Xue, "Adjusting the imbalance ratio by the dimensionality of imbalanced data," *Pattern Recognition Letters*, vol. 133, pp. 217–223, May 2020, doi: 10.1016/j.patrec.2020.03.004.
- [28] W. Ahmed, P. A. Bath, and G. Demartini, "Chapter 4: Using Twitter as a Data Source: An Overview of Ethical, Legal, and Methodological Challenges," 2017, pp. 79–107, doi: 10.1108/s2398-601820180000002004.
- [29] D. Gamal, M. Alfonse, E. S. M. El-Horbaty, and A. B. M. Salem, "Twitter Benchmark Dataset for Arabic Sentiment Analysis," *International Journal of Modern Education and Computer Science*, vol. 11, no. 1, pp. 33–38, Jan. 2019, doi: 10.5815/ijmecs.2019.01.04.
- [30] R. M., H. M., and M. Hussein, "Improving Arabic Text Categorization using Normalization and Stemming Techniques," *International Journal of Computer Applications*, vol. 135, no. 2, pp. 38–43, Feb. 2016, doi: 10.5120/ijca2016908328.
- [31] A. Kulkarni and A. Shivananda, "Converting Text to Features," in *Natural Language Processing Recipes*, Berkeley, CA: Apress, 2019, pp. 67–96, doi: 10.1007/978-1-4842-4267-4_3.




- [32] L. Breiman, "Bagging predictors," *Machine Learning*, vol. 24, no. 2, pp. 123–140, Aug. 1996, doi: 10.1007/bf00058655.
- [33] X. Feng, "Research of sentiment analysis based on adaboost algorithm," in *Proceedings - 2019 International Conference on Machine Learning, Big Data and Business Intelligence, MLBDBI 2019*, IEEE, Nov. 2019, pp. 279–282, doi: 10.1109/MLBDBI48998.2019.00062.
- [34] D. P. Chatterjee, S. Mukhopadhyay, S. Goswami, and P. K. Panigrahi, "Efficacy of Oversampling Over Machine Learning Algorithms in Case of Sentiment Analysis," in *Advances in Intelligent Systems and Computing*, IEEE, 2021, pp. 247–260, doi: 10.1007/978-981-15-5619-7_17.
- [35] H. Han, W. Y. Wang, and B. H. Mao, "Borderline-SMOTE: A new over-sampling method in imbalanced data sets learning," in *Lecture Notes in Computer Science*, 2005, pp. 878–887, doi: 10.1007/11538059_91.
- [36] A. Onan and S. Korukoğlu, "Exploring performance of instance selection methods in text sentiment classification," in *Advances in Intelligent Systems and Computing*, 2016, pp. 167–179, doi: 10.1007/978-3-319-33625-1_16.
- [37] P. E. Hart, "The Condensed Nearest Neighbor Rule," *IEEE Transactions on Information Theory*, vol. 14, no. 3, pp. 515–516, May 1968, doi: 10.1109/TIT.1968.1054155.
- [38] M. Kubat and S. Matwin, "Addressing the curse of imbalanced training sets: one-sided selection," *ICML*, vol. 97, no. 1, 1997.
- [39] J. Huang and C. X. Ling, "Using AUC and accuracy in evaluating learning algorithms," *IEEE Transactions on Knowledge and Data Engineering*, vol. 17, no. 3, pp. 299–310, Mar. 2005, doi: 10.1109/TKDE.2005.50.
- [40] A. Luque, A. Carrasco, A. Martín, and A. de las Heras, "The impact of class imbalance in classification performance metrics based on the binary confusion matrix," *Pattern Recognition*, vol. 91, pp. 216–231, Jul. 2019, doi: 10.1016/j.patcog.2019.02.023.

BIOGRAPHIES OF AUTHORS






Maisa J. Al-Khazaleh    is a faculty member at Basic Sciences Department, Faculty of Science, Hashemite University, Zarqa, Jordan. She received her B.Sc. degree in Computer Science from Jordan University of Science and Technology in 2007, and then she received the M.Sc. degree in Computer Science from Yarmouk University in 2009. Her research interests are primarily in the area of machine learning and natural language processing. She can be contacted at email: maisa@hu.edu.jo.



Marwah Alian    is a faculty member at Basic Sciences Department, Faculty of Science, Hashemite University, Zarqa, Jordan. She received her B.Sc. in Computer Science from Hashemite University in 1999. After graduation, she worked as a programmer then as a teacher in many high schools in Jordan then she received the M.Sc. degree in Computer Science from The University of Jordan in 2007. In 2021, she received her PhD from the Department of Computer Science in Princess Sumaya University for Technology. She has a number of publications in elearning systems, data mining, mobile applications, adaptive learning, mobile learning, and natural language processing. She can be contacted at email: Marwahn@hu.edu.jo.



Manar A. Jaradat    is a faculty member at Computer Engineering Department, Faculty of Engineering, Hashemite University, Zarqa, Jordan. She received her B.Sc. degree in computer engineering from Yarmouk University, Jordan, in 2009, and then she received the M.Sc. degree in Computer Engineering from Jordan University of Science and Technology in 2014. Her research interests are primarily in the area of machine learning, natural language processing, and supply chain network modeling. She can be contacted at email: manara@hu.edu.jo.