An Extended Oversampling Method for Imbalanced Quranic Text Classification based on A Genetic Algorithm

Bassam Arkok[1], Akram M. Zeki [2], Roslina Othman[3], Abdulaziz Aborujilah[4]

[1, 2, 3] Kulliyyah of Information and Communication Technology, International Islamic University Malaysia, Malaysia.
[4]Malaysian Institute of Information Technology (MIIT), University Kuala Lumpur, Kuala Lumpur, Malaysia.

[1]bassam_arkok@gmail.com, [2]akramzeki@iium.edu.my, [3]roslina@iium.edu.my, [4]abdulazizsaleh@unikl.edu.my

ABSTRACT

The Quran is considered the holy book for the Muslim community, and its text contains vast amounts of knowledge and guidance that Muslims strive to extract and understand. To achieve this, Quranic text classification plays a crucial role in categorizing and organizing the vast amount of information contained within the Quranic text. However, the process of Quranic text classification is not without its challenges. One of the significant challenges in Quranic text classification is obtaining a homogenous and balanced dataset to train the classification models accurately. Due to the nature of the Quranic text, which contains various topics and themes, the distribution of Quranic text classes is often abnormal. This abnormal distribution makes it difficult to obtain a consistent and balanced dataset, which can weaken the overall classification performance. To address this issue, this paper proposes a new oversampling method that employs Genetic algorithm to generate an optimal and balanced dataset simultaneously. The proposed method is specifically tested for Quranic topics that contain several imbalanced binary classes. The results of the study demonstrate the effectiveness of the Genetic algorithm in generating a balanced dataset, which leads to better classification performance results.Overall, this paper highlights the importance of Quranic text classification and the challenges associated with it. The proposed oversampling method provides a novel solution to address the issue of imbalanced Quranic datasets and can significantly improve the accuracy of Quranic text classification.

Keywords: Imbalance classification, Quranic topics, Genetic algorithm, oversampling.

I. Introduction:

Imbalanced text classification refers to the task of assigning one or more classes to a document based on its content, when the distribution of samples across classes is highly uneven. This occurs when one class has significantly more samples than the others, and can be observed in various real-world scenarios, such as software defects, cancer gene expressions, natural disasters, telecommunications fraud, and fraudulent credit card transactions. The rates of imbalanced datasets range from 0.01% to 29.1%, as reported by Khalilia, Chakraborty et al. (2011). Classifiers tend to perform poorly on imbalanced data due to variations caused by the number of samples in each class, resulting in high accuracy for the majority class and low accuracy for the minority class, as noted by Nayal, Jomaa et al. (2017). This poses one of the most difficult challenges in the field of machine learning and has gained significant attention in recent years, as highlighted by Alibeigi, Hashemi et al. (2012) and Haixiang, Yijing et al. (2017). The common practice of optimizing the overall accuracy of classifiers using techniques such as SMOTE, RUS, and ROS can result in low sensitivity for minority classes, as pointed out by Elhassan and Aljurf (2016). Therefore, the goal should be to simultaneously maximize the sensitivity of both majority and minority classes. To achieve this, various methods have been proposed, including sampling techniques, ensemble learning, cost-sensitive learning, feature selection, and algorithmic modification, as reported by Choi (2010).

Resampling techniques aim to tackle the issue of imbalanced class distribution in the learning process. These methods are independent of the chosen classifier and are therefore more flexible. There are three types of resampling methods based on distribution: oversampling, undersampling, and hybrid. Oversampling generates synthetic samples or duplicates minority class samples to mitigate the impact of imbalanced distribution. On the other hand, undersampling balances the distribution by removing samples from the majority class. Hybrid methods use a combination of both oversampling and undersampling techniques, as noted by Haixiang, Yijing et al. (2017). The Genetic Algorithm is an iterative search and optimization method proposed by J. Holland in 1975, inspired by human genetics. It uses the principles of the evolution process, specifically survival of the fittest, to find the best solutions to a problem. Solutions are represented as chromosomes or genomes within a population, and genetic operators are applied to individuals in the population to generate new generations. The genetic algorithm involves four steps: initial population generation, selection operator, crossover operator, and mutation operator. The initial population is created using an appropriate encoding method. The selection operator chooses the optimal value based on a fitness function that evaluates how close a solution is to the optimum solution. The selected individuals are used to generate the best offspring for future generations, with the crossover operator combining genetic information from two parents to generate new offspring. The mutation operator helps to maintain genetic diversity within the population from one generation to the next, as highlighted by Mehta (2012).

The Quran is considered to be the primary source of religious text for Muslims, consisting of both instructive and narrative content. The Quranic text has a cohesive connection throughout, making it a related concept. However, finding the implied connections between chapters and verses requires deeper study. To aid in understanding the Quran, Muslims classify its topics. However, some topics have an imbalanced distribution of verses, causing inconsistencies in classification performance between classes. This is due to the lack of harmony and balance among the classes. To address this issue, a new framework using genetic algorithm is proposed to classify imbalanced Quranic topics. The proposed oversampling method, HOGA (Harmonized Oversampling method based on Genetic Algorithm), rebalances both majority and minority classes simultaneously. This binary classification approach classifies two Quranic imbalanced topics separately.

The rest of this document is structured as follows: in the following section, previous research related to the suggested approaches for enhancing the classification of imbalanced datasets is presented. Afterwards, the proposed model, including the suggested technique, is described. Following that, the experimental outcomes are presented, and ultimately, the paper concludes with concluding remarks in the last section.

II. Related Studies

n this section, various resampling methods that have been used to address the issue of imbalanced datasets in previous studies are discussed. Resampling methods aim to modify the distribution of the majority and minority classes in the training data to achieve a balanced class distribution. One of the most commonly used oversampling methods is the Synthetic Minority Oversampling Technique (SMOTE), which was introduced by Chawla, Bowyer et al. in 2002 (Popel, Hasib et al. 2018). SMOTE generates new synthetic minority samples for the minority class. Another version of SMOTE is the borderline-SMOTE method, which was proposed by Han, Wang et al. in 2005. The adaptive synthetic sampling approach, ADASYN algorithm, developed by He, Bai et al. in 2008 builds on the SMOTE method by focusing on the minority classes that are difficult to recover. The Cluster Based Over Sampling (CBO) method clusters the training data of each class individually using the k-means method, and then applies random oversampling on each cluster. Random Oversampling (ROS) attempts to balance the class distribution by replicating minority class examples randomly and generates exact duplicates of the existing samples (Elhassan and Aljurf 2016). Random Under sampling (RUS) is an undersampling method that randomly eliminates samples of the majority class to balance the class distribution..

Several studies have proposed combining over-sampling and under-sampling methods to balance imbalanced datasets. Examples of these studies include Barandela, Valdovinos et al. (2004), Estabrooks, Jo et al. (2004), Cateni, Colla et al. (2014), Dubey, Zhou et al. (2014), and Díez-Pastor, Rodríguez et al. (2015). In addition, various studies have utilized genetic algorithms (GA) to generate artificial samples for the minority class. For instance, Benchaji, Douzi et al. (2018) proposed a GA-based oversampling method that used K-means clustering to generate new samples for the minority class. Beckmann, de Lima et al. (2011) applied GA directly to obtain new instances for the minority class. Cervantes, Li et al. (2013) used support vector machines (SVM) to create support vectors and a draft hyperplane, and then employed GA to generate new data points within the classification margin or sensitive area. Cervantes, Huang et al. (2013) proposed an approach for SVM classification that obtained new artificial instances from SVs and included only the instances that enhanced SVM performance. Jiang, Lu et al. (2016) proposed a new GA-based oversampling algorithm, GASMOTE, which used different sampling rates for various instances of the minority class to determine the optimal sampling rate combination. They applied GA to find optimized sampling rates, generated a new dataset through oversampling using the optimized rates, and implemented OGA (oversampling genetic algorithm), which was similar to the approach proposed by Beckmann, de Lima et al. (2011).

These studies applied the genetic algorithm processes to resample the minority class only. So, they got optimal samples for the minority class only. In this paper, the process of oversampling will be done for the two sides; the majority and minority class to get optimal samples for both classes. To conduct that, samples of the majority and minority will be reduced by selection step of GA firstly. Then, new instances for selected samples will be generated by process of cross over until the dataset become balancing. After that, harmonized or optimal samples for both classes and balanced have been gotten simultaneously, with these steps, harmonize can be made among the whole training dataset with each other and also to rebalance it.

III. Proposed Model

The proposed model contains 7 phases are: the collecting of datasets, Preprocessing, Feature selection, the dividing of prepared data into training and testing datasets, resampling process by HOGA and other methods, the building of classifiers models, and the evaluation of resampling methods as shown in figure 1.

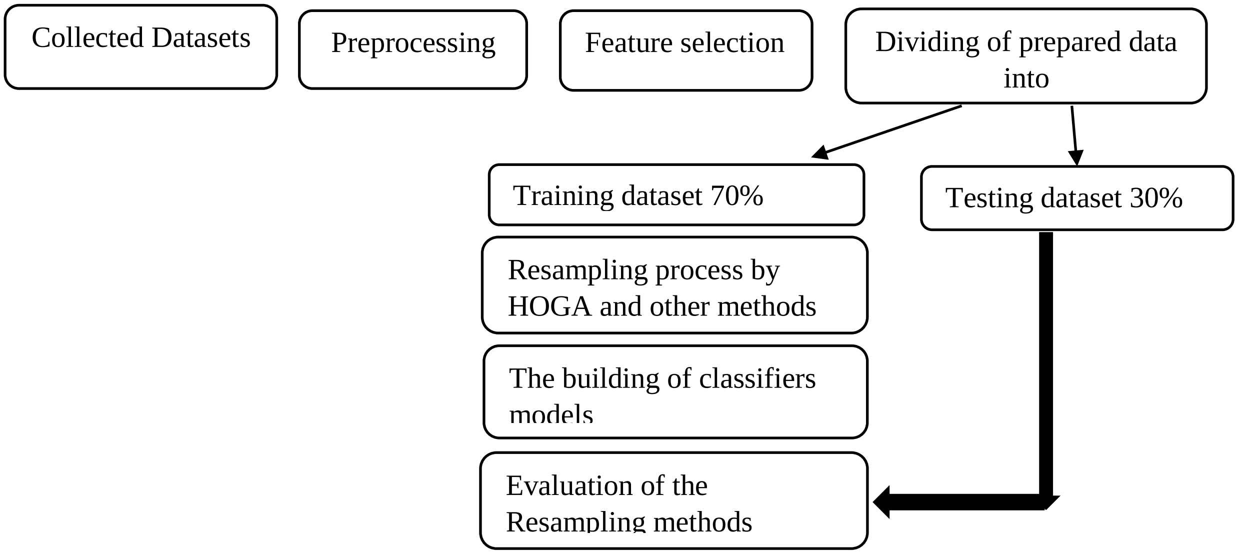
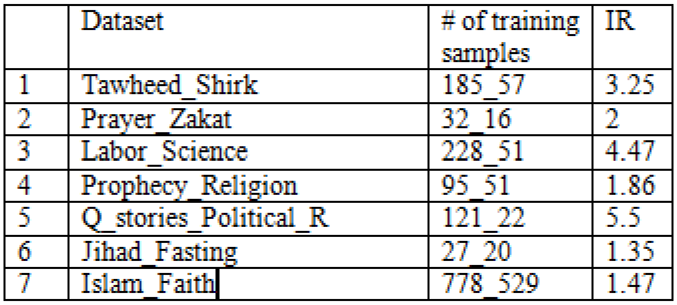

Collected Datasets
Preprocessing
Feature selection
Dividing of prepared data into
Training dataset 70%
Testing dataset 30%
Resampling process by HOGA and other methods
The building of classifiers models
Evaluation of the Resampling methods

Figure1: Workflow of the proposed framework.

a. Collected Datasets

The Quranic topics used in this paper declared by the lexicon for the Quranic topics which is used by (Al-Kabi, Alsmadi et al. 2013). The topics chosen and number of verses for each topic are illustrated in table 1. In this table, the imbalanced ratios of classes (IR) can be shown also.



	Dataset	# of training samples	IR
1	Tawheed_Shirk	185_57	3.25
2	Prayer_Zakat	32_16	2
3	Labor_Science	228_51	4.47
4	Prophecy_Religion	95_51	1.86
5	Q_stories_Political_R	121_22	5.5
6	Jihad_Fasting	27_20	1.35
7	Islam_Faith	778_529	1.47

Table 1: Quranic datasets collected for testing

b. Preprocessing

The paper under discussion employs the Quranic roots extracted by Khedher in 2017, which removes the need for any preprocessing steps to obtain the word roots. The collected data is then represented using term frequency (TF), where the roots of all the words in the Quran are formatted in the vector space. Next, the word frequency for each verse is calculated, and its corresponding feature is recorded in the vector space. These vectors are then combined to create a matrix for the classification phase. The labeled vectors in the matrix are based on the lexicon used. Moreover, any common verses that appear in both Tawheed and Shirk categories are excluded from the matrix to ensure accurate data and avoid confusion during the classification phase. For instance, if there are 24 verses that appear in both Tawheed and Shirk categories, these verses are removed from both Tawheed and Shirk classes.

Feature selection

In this step, the Chi-square algorithm is utilized to select important features for classification purposes, leading to an improvement in performance. This algorithm is commonly used for Arabic text classification, as observed in previous studies (Khorsheed and Al-Thubaity 2013, Alabbas, Al-Khateeb et al. 2016). Once the important features are identified, duplicate samples are removed again to allow for a more precise evaluation of the resampling methods. This is because a sample that exists in both the training and testing datasets will have a perfect accuracy score of 100%, which could skew the results. Thus, this issue is taken into consideration in this study to ensure more accurate results and a rigorous evaluation of the various methods.

Dividing of pre-pared data into training and testing

In this step, the represented data is divided into two parts; training and testing datasets. The training dataset is used to build the classifier models while the testing is utilized to evaluate the classifier later. The Percentage of data division is 70% and 30% for training and testing datasets, respectively.

c. Resampling process by HOGA and other different methods

Previous resampling methods are implemented in this paper to compare them with the proposed oversampling method. These methods are SMOTE, RUS, ROS, and OGA. OGA generates new samples for the minority class only based on HOGA' steps to rebalance the dataset. So, the samples of minority class have been optimized by GA that is heterogeneous with the samples of majority class. OGA is implemented in this paper to show the performance of imbalanced classification when GA is applied for both majority and minority classes. So, results of OGA are evaluated and compared beside other resampling methods with HOGA algorithm.

d. Building models of the classifiers

The tool utilized in this study for uploading the datasets and constructing the classifier models is WEKA 3.9.2, also known as Waikato Environment for Knowledge Analysis. Several well-known classifiers such as KNN, LibSVM, Random Forest, Decision Tree (J48), and Naïve Bayes are employed in this research.

Evaluation of the Resampling methods

This is the concluding phase of the suggested framework where the outcomes of various resampling methods are measured using several metrics, which are commonly used to evaluate imbalanced classification performance. These metrics include Sensitivity, Specificity, Overall Accuracy, Precision, Balanced Accuracy, F-Measure, G-mean, and Matthews Correlation Coefficient (MCC). In this study, the samples belonging to the majority class are deemed positive while those belonging to the minority class are considered negative.

TP: Positive samples that are correctly classified.

FP: Negative samples that are incorrectly classified.

TN: Negative samples that are correctly classified.

FN: Positive samples that are incorrectly classified.

The sensitivity, also known as the True Positive Rate, accuracy of positive examples, or recall, is a metric that quantifies the percentage of positive examples that are correctly identified.

$$\text{Sensitivity} = \frac{\text{TP}}{\text{TP+FN}} \dots \dots \dots \dots \dots \dots \dots \text{Equation. } 1$$

The specificity, also referred to as the True Negative Rate and accuracy of negative examples, is a measure that assesses the percentage of negative examples that are correctly identified

$$\text{Specificity} = \frac{\text{TN}}{\text{TN} + \text{FP}} \dots \dots \dots \dots \dots \dots \dots \text{Equation. } 2$$

1. Overall accuracy is a crucial parameter used to evaluate the effectiveness of a model, which is typically calculated as:

$$\text{Overall Accuracy} = \frac{\text{TP} + \text{TN}}{\text{TP} + \text{TN} + \text{FP} + \text{FN}} \dots \dots \text{Equation. } 3$$

In normal cases, accuracy alone provides adequate information to test While precision is important for evaluating the system's performance, it may not be sufficient when dealing with imbalanced datasets, as the classifier may become overly focused on the majority class, leading to biased predictions (as noted by Tang, Zhang et al. 2008, Al-Azani and El-Alfy 2017, Patel and Thakur 2017).

Positive Predictive Value, also known as Precision, measures the fraction of all the predicted positive results that are truly positive examples..

$$\text{Precision} = \frac{\text{TP}}{\text{TP} + \text{FP}} \dots \dots \dots \dots \dots \dots \dots \text{Equation. } 4$$

Balanced Accuracy is a measure that takes into account both sensitivity and specificity, providing an accurate estimate of overall model performance even when the dataset is imbalanced. It is calculated as the average of sensitivity and specificity. Unlike traditional accuracy, which can be inflated by a high number of true negatives in imbalanced datasets, Balanced Accuracy is a more reliable measure of classification performance.

$$\text{Balanced Accuracy} = \frac{\text{Sensitivity}}{\text{Specificity}} \ldots\ldots\ldots\ldots\ldots\ldots\ldots \text{Equation. } 5$$

2. The F-measure, also known as the F1 score or F-score, is a metric that combines the recall and precision measures to provide a weighted average that considers the trade-off between them.

$$\text{F} - \text{Measure} = 2 \times \left(\frac{\text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}}\right) \ldots\ldots\ldots \text{Equation. } 6$$

The G-mean is a metric that calculates the geometric mean of the specificity and recall measures. In binary classification problems, the G-mean is used to maximize the accuracy of each class when their accuracies are balanced

$$\text{G} - \text{mean} = \sqrt{\text{Sensitivity} \times \text{Specificity}} \ldots\ldots\ldots \text{Equation. } 7$$

3. The Matthews correlation coefficient (MCC) is a measure that takes into account both true and false positives and negatives in its calculation. It quantifies the correlation between the actual and predicted binary classifications, with values ranging from -1 to +1. An MCC score of +1 indicates a perfect prediction, while a score of 0 indicates a random prediction and a score of -1 denotes complete disagreement between the observed and predicted outcomes.

$$\text{MCC} = \frac{\text{TP} \times \text{TN} - \text{FP} \times \text{FN}}{\sqrt{(\text{TP} + \text{FP})(\text{TP} + \text{FN})(\text{TN} + \text{FP})(\text{TN} + \text{FN})}} \ldots\ldots \text{Equation. } 8$$

IV. HOGA as oversampling method

In this paper, a new oversampling method is proposed which is implemented for the majority and minority classes simultaneously. The proposed method is called HOGA which applies the operators of genetic algorithm to create optimized samples for the classes. This generation makes samples of majority and minority classes that have been optimized and balanced at the same time. HOGA algorithm uses the samples as they are represented in the matrix so that they are not encoded to binary codes. HOGA's steps are clarified by the following flowchart and the pseudo code of HOGA is as follow:

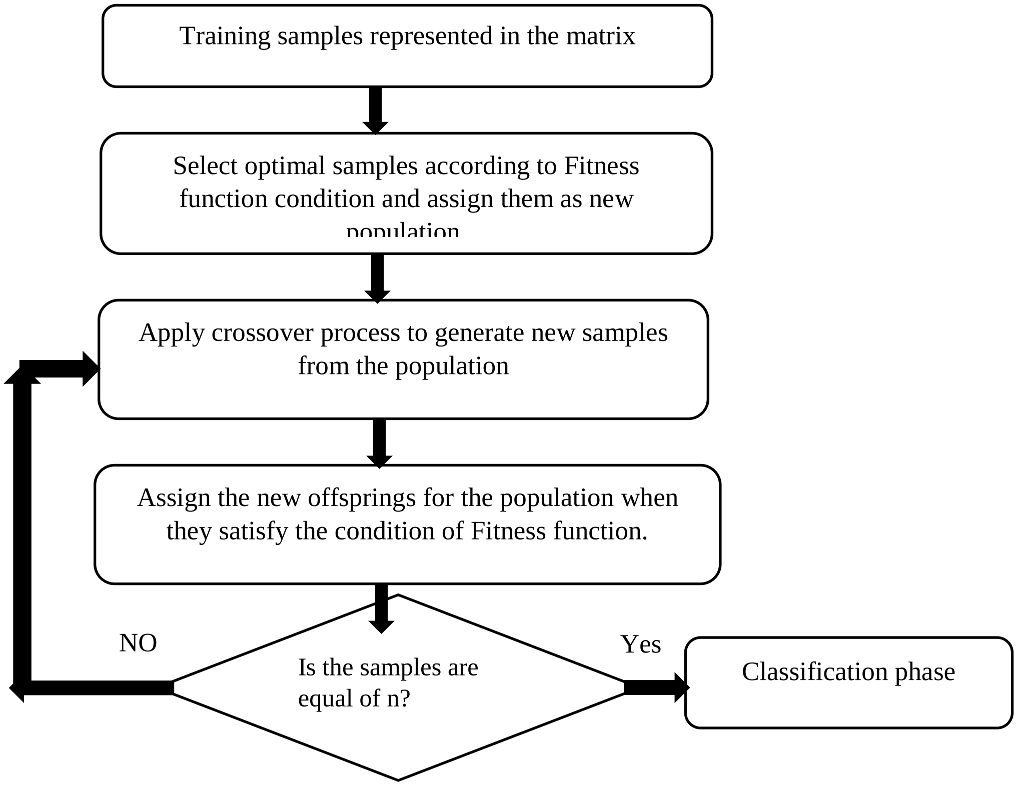

Training samples represented in the matrix
Select optimal samples according to Fitness function condition and assign them as new population
Apply crossover process to generate new samples from the population
Assign the new offsprings for the population when they satisfy the condition of Fitness function.
Is the samples are equal of n?
NO
Yes
Classification phase

Figure2: Flowchart of proposed method

- Assign value of fitness f based on data rate, and also determine the number of the majority class in the training dataset n.

- Compute the fitness value for all samples of the training dataset by fitness function.

- Select the samples only that their fitness value is less than or equal to f and assign them as new population.

a- Apply process of cross over for the population to generate new samples.

- Compute the fitness value for new samples generated and assign them for the population if their fitness values are less than or equal to f.

- Stop if the number of population's samples are equal to n; otherwise go to a line.

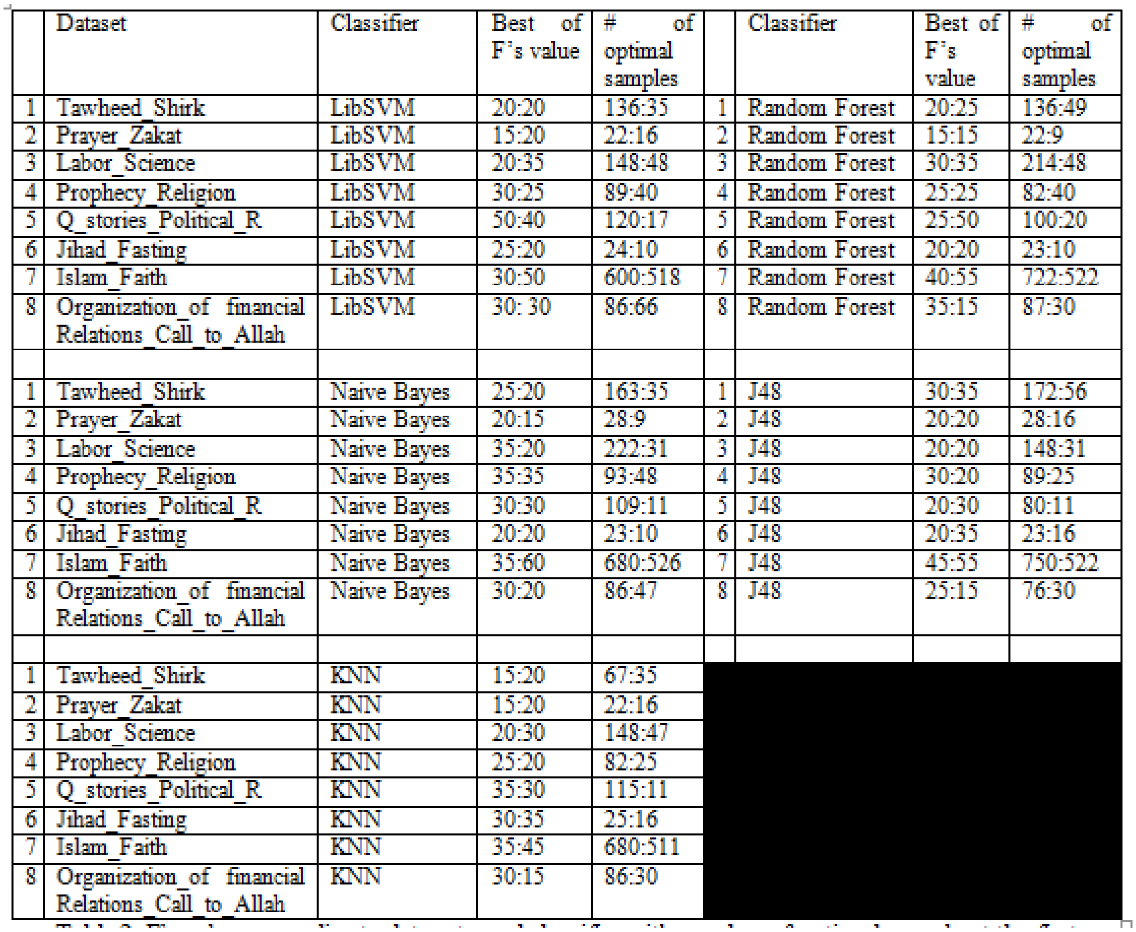
The proposed resampling method are performed by three main processes; Fitness function, Selection operator, and Crossover operator.

1- Fitness function

In order to establish the fitness function for HOGA, the mean or center sample of the training dataset is initially determined. Afterwards, the fitness values for each sample are evaluated by calculating the distance between the mean and the sample. To determine the distances, the Euclidean distance metric is utilized and the resulting fitness values for samples are multiplied by ten to convert the decimal values to integer values.

Selection process

This stage involves selecting samples from the current population based on their fitness values, with the specific selection criterion being that their fitness values must be less than or equal to a pre-determined threshold value denoted by "f." The actual value of "f" is determined by factors such as the data rate and the type of classifier being used, and in this particular study, its value ranges from 15 to 55 for all datasets as presented in Table 2. The number of optimal samples selected for the first time according to the classifiers is also included in the table.



	Dataset	Classifier	Best of F's value	# of optimal samples		Classifier	Best of F's value	# of optimal samples
1	Tawheed_Shirk	LibSVM	20:20	136:35	1	Random Forest	20:25	136:49
2	Prayer_Zakat	LibSVM	15:20	22:16	2	Random Forest	15:15	22:9
3	Labor_Science	LibSVM	20:35	148:48	3	Random Forest	30:35	214:48
4	Prophecy_Religion	LibSVM	30:25	89:40	4	Random Forest	25:25	82:40
5	Q_stories_Political_R	LibSVM	50:40	120:17	5	Random Forest	25:50	100:20
6	Jihad_Fasting	LibSVM	25:20	24:10	6	Random Forest	20:20	23:10
7	Islam_Faith	LibSVM	30:50	600:518	7	Random Forest	40:55	722:522
8	Organization_of financial Relations_Call_to_Allah	LibSVM	30: 30	86:66	8	Random Forest	35:15	87:30
1	Tawheed_Shirk	Naive Bayes	25:20	163:35	1	J48	30:35	172:56
2	Prayer_Zakat	Naive Bayes	20:15	28:9	2	J48	20:20	28:16
3	Labor_Science	Naive Bayes	35:20	222:31	3	J48	20:20	148:31
4	Prophecy_Religion	Naive Bayes	35:35	93:48	4	J48	30:20	89:25
5	Q_stories_Political_R	Naive Bayes	30:30	109:11	5	J48	20:30	80:11
6	Jihad_Fasting	Naive Bayes	20:20	23:10	6	J48	20:35	23:16
7	Islam_Faith	Naive Bayes	35:60	680:526	7	J48	45:55	750:522
8	Organization_of financial Relations_Call_to_Allah	Naive Bayes	30:20	86:47	8	J48	25:15	76:30
1	Tawheed_Shirk	KNN	15:20	67:35				
2	Prayer_Zakat	KNN	15:20	22:16				
3	Labor_Science	KNN	20:30	148:47				
4	Prophecy_Religion	KNN	25:20	82:25				
5	Q_stories_Political_R	KNN	35:30	115:11				
6	Jihad_Fasting	KNN	30:35	25:16				
7	Islam_Faith	KNN	35:45	680:511				
8	Organization_of financial Relations_Call_to_Allah	KNN	30:15	86:30				

Table 2: F's values according to data rate and classifier with number of optimal sample at the first time.

2- Crossover process

To produce new samples, HOGA algorithm utilizes a single point crossover operator. Selected samples undergo the crossover operator to generate offspring samples, which are added to the current population if their fitness values are less than or equal to f's value. This process is repeated until the number of populations equals the number of majority samples before the application of HOGA. The single point linear crossover process for mating samples is illustrated in Figure 3.

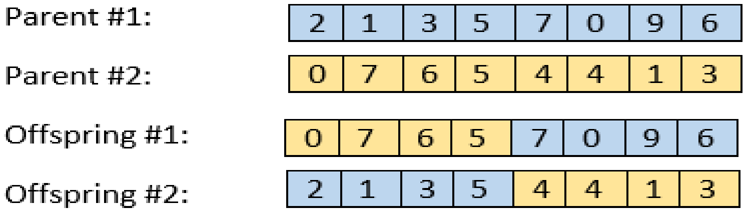

Parent #1: 2 1 3 5 7 0 9 6
Parent #2: 0 7 6 5 4 4 1 3
Offspring #1: 0 7 6 5 7 0 9 6
Offspring #2: 2 1 3 5 4 4 1 3

Figure 3: An example of single point crossover operator.

The HOGA's steps are applied for the samples of majority and minority classes simultaneously to get optimal samples for both, and also to rebalance them. So, evolutional samples can be presented for the entire training dataset that they can improve the performance of classifiers. During the experimental results, HOGA's affecting and superiority than the other methods can be noticed in the next section.

V. Experimental results

This section evaluates the performance of HOGA and compares it with the performance of other resampling methods, namely, SMOTE, RUS, ROS, and OGA. OGA was evaluated to highlight GA's effect when it is applied to both classes. For this evaluation, many metrics are used to test the performance; Sensitivity, Specificity, Precision, balanced accuracy, F-measure, G-mean, MCC, and overall accuracy. For these tests, 8 Quranic datasets with different imbalanced ratios, ranging between (1.34–5.5), were chosen as shown in Table 1.

	Sensitivity	Specificity	Precision	B_ACC	F-Measure	G-Mean	MCC	ACC
LibSVM_SMOTE	0.77	0.78	0.89	0.78	0.82	0.76	0.55	0.78
LibSVM_RUS	0.77	0.75	0.88	0.77	0.81	0.74	0.51	0.77
LibSVM_ROS	0.79	0.75	0.88	0.78	0.82	0.76	0.54	0.78
LibSVM_HOGA	**0.85**	0.83	**0.93**	**0.85**	**0.88**	**0.84**	**0.68**	**0.85**
LibSVM_OGR	0.80	**0.86**	**0.93**	0.83	0.85	0.82	0.64	0.83

Naïve Bayes_SMOTE	**0.88**	0.68	0.86	0.78	0.86	0.77	0.58	0.83
Naïve Bayes_RUS	0.78	0.80	0.88	0.79	0.83	0.79	0.54	0.78
Naïve Bayes_ROS	0.81	0.79	0.89	0.80	0.85	0.80	0.58	0.81
Naïve Bayes_HOGA	0.84	**0.87**	**0.93**	**0.86**	**0.88**	**0.85**	**0.69**	**0.85**
Naïve Bayes_OGR	0.82	0.84	0.91	0.83	0.85	0.82	0.63	0.82
KNN_SMOTE	0.86	0.62	0.82	0.74	0.84	0.73	0.51	0.79
KNN_RUS	0.81	0.61	0.83	0.71	0.81	0.68	0.45	0.76
KNN_ROS	**0.88**	0.55	0.82	0.72	0.84	0.69	0.48	0.79
KNN_HOGA	**0.88**	**0.68**	**0.86**	**0.78**	**0.87**	**0.77**	**0.58**	**0.82**
KNN_OGR	0.87	0.62	0.83	0.74	0.84	0.73	0.51	0.79
J48_SMOTE	**0.82**	0.74	0.88	0.78	0.84	0.77	0.57	0.81
J48_RUS	0.73	0.77	0.88	0.75	0.78	0.73	0.47	0.74
J48_ROS	0.79	0.73	0.87	0.76	0.82	0.74	0.51	0.78
J48_HOGA	0.81	**0.83**	**0.91**	**0.82**	**0.85**	**0.81**	**0.61**	**0.82**
J48_OGR	0.81	0.77	0.89	0.79	0.83	0.78	0.58	0.80
Random Forest_SMOTE	**0.89**	0.66	0.86	0.78	0.87	0.76	0.59	0.83
Random Forest_RUS	0.81	**0.80**	**0.90**	0.81	0.85	0.80	0.59	0.82
Random Forest_ROS	0.85	0.70	0.87	0.78	0.86	0.77	0.58	0.82
Random Forest_HOGA	0.88	**0.80**	**0.90**	**0.84**	**0.89**	**0.83**	**0.69**	**0.86**
Random Forest_OGR	0.86	0.70	0.87	0.78	0.85	0.77	0.59	0.81

Table 3 reveals the experimental results of this study which shows the superiority of the proposed method. It comprises of the averaged values over all the collected datasets for every measure according to the used classifiers.

Table 3: Average results of all datasets in the classifiers.

From the above table, the proposed method has better performance than the other methods for many measures and classifiers. For instance, in LibSVM, Naive Bayes, J48 and Random Forest classifiers, the proposed method outperformed in 7 out of 8 measures, while in the KNN classifier, HOGA outperformed than all other methods in all measures. This improvement has been done because the entire training dataset became optimized and balanced simultaneously. Therefore, the proposed resampling method improved the performance of all applied classifiers.

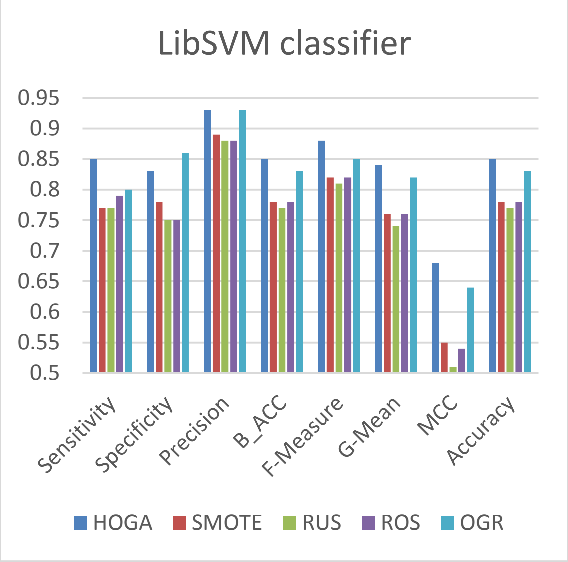


Figure 4: Averaged results of used resampling methods for LibSVM classifier.

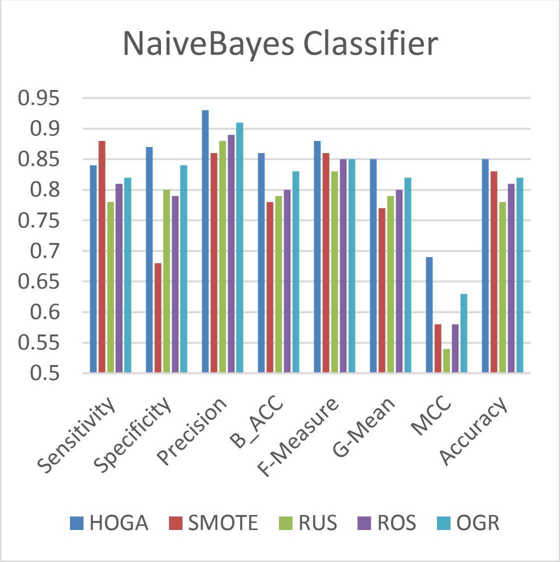


Figure 5: Average results of used resampling methods for Naïve Bayes classifier.

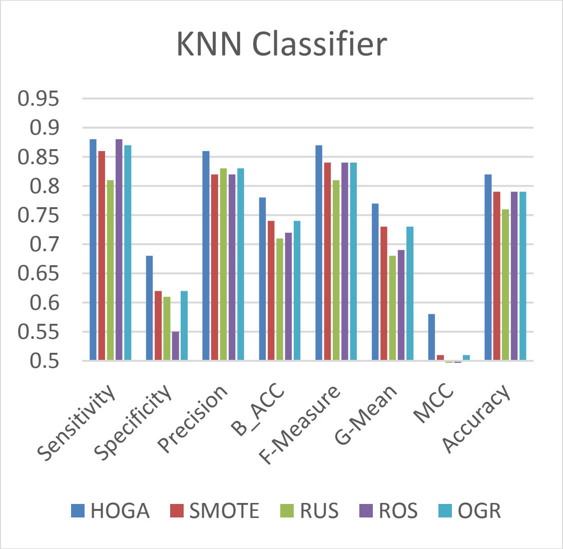


Figure 6: Averaged results of used resampling methods for KNN classifier.

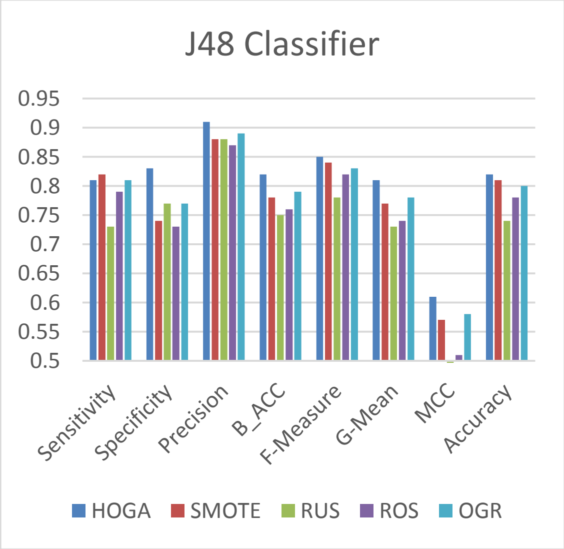


Figure 7: Average results of used resampling methods for J48 classifier.

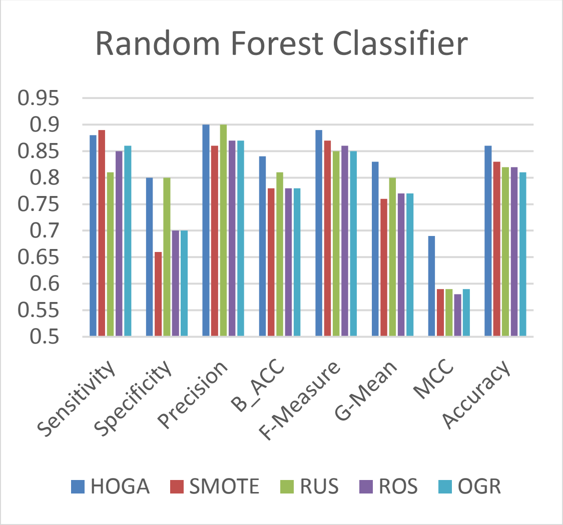


Figure 8: Averaged results of used resampling methods for Random Forest classifier.

VI. Results Discussion

This section provides a detailed explanation of the experimental results, including charts for each classifier based on the used metrics. The performance of HOGA in improving the imbalanced classification is presented in Table 3 and Figures [4-8]. HOGA outperformed all other resampling methods used to improve the performance of imbalanced classification. The following paragraphs discuss the results of different resampling methods based on evaluation measures and implemented classifiers to explore the performance of HOGA and other methods' the case of the LibSVM classifier, HOGA significantly improved sensitivity and MCC measures, showing differences of 4% to 5% compared to OGA, which was the second-best method. However, in the case of specificity, balanced accuracy, F-measure, G-mean, and overall accuracy measures, HOGA showed only a slight improvement, ranging from 2% to 3% compared to OGA. There was no improvement in the precision measure, where HOGA was equal to OGA.For the Naïve Bayes classifier, HOGA significantly improved the MCC measure, showing a difference of 7% compared to OGA. In the case of specificity, precision, balanced accuracy, G-mean, and overall accuracy measures, the difference ranged from 2% to 3% compared to OGA, which was the closest method. HOGA outperformed SMOTE in the F-measure, showing a difference of 2%, but SMOTE was the best method in the sensitivity measure, where the difference compared to HOGA was 4%

.

VIII. Regenerate response

In KNN classifier, HOGA outperformed over other techniques in all measures. The performance improved significantly in MCC measure in which the difference was 7% on the second ones, SMOTE, and OGA. While in the Specificity, Balanced Accuracy, G_mean measures, the difference was 4% on the same second ones, SMOTE, and OGA. While for the F_Measure, and overall accuracy, and Precision measures the difference was 3% between HOGA and the closest methods. Finally, HOGA, ROS, and OGA took the highest results in the sensitivity measure, where their performance was very close for this measure.

For J48 classifier, HOGA also had the better results and outperformed than the others in all measures. The biggest improvement was in the Specificity, and Balanced Accuracy measures, the difference were 6%, and 5% respectively on the second ones, RUS, and OGA. While in the Precision, G_Mean, and MCC measures, the difference was 3% on the second method, OGA. For the precision measure, the difference between HOGA and the closest one, OGA, was 2%. Finally, HOGA and SMOTE methods, the first ones, were very equivalent in the F_Measure, overall accuracy, and Sensitivity measures where the difference was 1% in these measures.

Finally, for Random Forest classifier, HOGA has better performance than the other methods. The performance improved significantly in the MCC measure in which the difference was 10% on the closest methods, SMOTE, RUS, and OGA. While in Balanced accuracy, F_Measure, G_mean, overall accuracy measures, the differences were 3% or 2% on the second ones, RUS, and SMOTE. In the Specificity, and Precision measures, HOGA and RUS had the same results, 0.80 for the Specificity, and 0.90 for the Precision, which were the best methods in these measures. They outperformed, 10%, and 3% for the Specificity, and Precision respectively, on the closest methods, ROS, and OGA. In the last measure, the SMOTE, and HOGA methods had equivalent performance in the Sensitivity measure, was 0.89 for SMOTE, and 0.88 for HOGA which were the best methods in this measure, and the difference between HOGA and the closets one, OGA, was 2%

Based on the information mentioned earlier, the resampling methods can be sorted from the strongest to the weakest as follows: the strongest method was HOGA, then OGA, and SMOTE methods, after that ROS, while the weakest method was RUS.

Moreover, the performance of applied classifiers according to the evaluation measures are summarized in the next words. Firstly, the Sensitivity measure got the highest values for all resampling method totally when the datasets were classified by KNN, and Random Forest, while it had the lowest values when the datasets were classified by J48 classifier. For the Specificity measure, the best values for it when the LibSVM, and Naïve Bayes classifier were used, in contrast, the weakest classifier for the Specificity measure was KNN. With regard to the Precision, and Balanced accuracy measures, all classifiers have superior performance for this measure except KNN classifier, had accepted performance but it is not like the others. While in F_Measure, overall accuracy, and G_Mean measures, all classifiers have excellent performance for these measures in particular in the Naïve Bayes, and Random Forest classifiers. With regard to LibSVM, the LibSVM classifier improved G_Mean measure significantly when it was combined with HOGA, and OGA methods. Finally, in MCC measure, MCC measure improved significantly by HOGA in all classifiers and outperformed over other methods. While OGA was the second method which increased in MCC's value when LibSVM, and Naïve Bayes classifiers were applied. Furthermore, the results of methods; SMOTE, RUS, OGA, and ROS are much closed when they were implemented with Random Forest classifier, but HOGA had the highest performance. Moreover, all resampling methods had very poor results in MCC measure, except HOGA, when the classification was implemented by KNN classifier. Finally, MCC's results for ROS, and RUS methods were very weak when J48 was used for the classification task.

In general, the experimental results can be summarized in these statements when the experimental results are analyzed altogether. Firstly, SMOTE with Random Forest had the best performance in the Sensitivity measure in all experiments. While HOGA with Naive Bayes had the best performance in 5 measures, namely, Specificity, Precision, Balanced accuracy, G-Mean, and MCC measures. Moreover, HOGA, and OGA methods with LibSVM classifier had the best performance in the Precision. Furthermore, HOGA with Random Forest classifier scored the highest results in 3 measures; F-Measure, MCC, and overall accuracy measures.

Therefore, HOGA with Naïve Bayes classifier achieved the best performance which improved imbalanced Quranic text classification. This finding was explored also in the work (Al-Kabi, Ata et al. 2013) when they classified Arabic Quranic topics by J48, KNN, SVM, and Naïve Bayes classifiers. In the end, they concluded that the Naïve Bayes had the best results.

X. Conclusion

In this paper, HOGA oversampling method was proposed which is designed to solve two issues of imbalanced classification problems; lack of homogeneity and balance in the datasets. To achieve these objectives, Genetic algorithm's operators are applied as oversampling method for both samples of majority and minority classes simultaneously. As shown in the results discussion section, HOGA scored the highest results over other resampling methods in many measures for all applied classifies. Also, the testing results on our datasets showed that the best performance had been achieved when HOGA was implemented with Naïve Bayes classifier. So, the proposed approach outperformed over other techniques in the classification of imbalanced Quranic text.

ACKNOWLEDGEMENT

This paper was supported by International Islamic University of Malaysia under (FRGS19-083-0691) research project.

References

Al-Azani, S. and E.-S. M. El-Alfy (2017). "Using word embedding and ensemble learning for highly imbalanced data sentiment analysis in short arabic text." Procedia Computer Science **109**: 359-366.

Al-Kabi, M. N., et al. (2013). A topical classification of Quranic Arabic text. Proceedings of the 2013 Taibah University International Conference on Advances in Information Technology for the Holy Quran and Its Sciences.

Alabbas, W., et al. (2016). Arabic text classification methods: Systematic literature review of primary studies. 2016 4th IEEE International Colloquium on Information Science and Technology (CiSt), IEEE.

Barandela, R., et al. (2004). The imbalanced training sample problem: Under or over sampling? Joint IAPR international workshops on statistical techniques in pattern recognition (SPR) and structural and syntactic pattern recognition (SSPR), Springer.

Beckmann, M., et al. (2011). Genetic algorithms as a pre processing strategy for imbalanced datasets. GECCO (Companion).

Benchaji, I., et al. (2018). Using Genetic Algorithm to Improve Classification of Imbalanced Datasets for Credit Card Fraud Detection. International Conference on Advanced Information Technology, Services and Systems, Springer.

Cateni, S., et al. (2014). "A method for resampling imbalanced datasets in binary classification tasks for real-world problems." Neurocomputing **135**: 32-41.

Cervantes, J., et al. (2013). A new approach to detect splice-sites based on support vector machines and a genetic algorithm. Iberoamerican Congress on Pattern Recognition, Springer.

Cervantes, J., et al. (2013). Using genetic algorithm to improve classification accuracy on imbalanced data. 2013 IEEE International Conference on Systems, Man, and Cybernetics, IEEE.

Chawla, N. V., et al. (2002). "SMOTE: synthetic minority over-sampling technique." Journal of artificial intelligence research **16**: 321-357.

Choi, J. M. (2010). "A selective sampling method for imbalanced data learning on support vector machines."

Díez-Pastor, J. F., et al. (2015). "Random balance: ensembles of variable priors classifiers for imbalanced data." Knowledge-Based Systems **85**: 96-111.

Dubey, R., et al. (2014). "Analysis of sampling techniques for imbalanced data: An n= 648 ADNI study." NeuroImage **87**: 220-241.

Elhassan, T. and M. Aljurf (2016). "Classification of Imbalance Data using Tomek Link (T-Link) Combined with Random Under-sampling (RUS) as a Data Reduction Method."."

Estabrooks, A., et al. (2004). "A multiple resampling method for learning from imbalanced data sets." Computational intelligence **20**(1): 18-36.

Haixiang, G., et al. (2017). "Learning from class-imbalanced data: Review of methods and applications." Expert Systems with Applications **73**: 220-239.

Han, H., et al. (2005). Borderline-SMOTE: a new over-sampling method in imbalanced data sets learning. International conference on intelligent computing, Springer.

He, H., et al. (2008). ADASYN: Adaptive synthetic sampling approach for imbalanced learning. 2008 IEEE International Joint Conference on Neural Networks (IEEE World Congress on Computational Intelligence), IEEE.

Jiang, K., et al. (2016). "A novel algorithm for imbalance data classification based on genetic algorithm improved SMOTE." Arabian journal for science and engineering **41**(8): 3255-3266.

Khalilia, M., et al. (2011). "Predicting disease risks from highly imbalanced data using random forest." BMC medical informatics and decision making **11**(1): 51.

Khedher, M. Z. (2017). "Multiword corpus of the Holy Quran." International Journal on Islamic Applications in Computer Science And Technology **5**(1).

Khorsheed, M. S. and A. O. Al-Thubaity (2013). "Comparative evaluation of text classification techniques using a large diverse Arabic dataset." Language resources and evaluation **47**(2): 513-538.

López, V., et al. (2013). "An insight into classification with imbalanced data: Empirical results and current trends on using data intrinsic characteristics." Information sciences **250**: 113-141.

Mehta, M. (2012). "Hybrid Genetic Algorithm with PSO Effect for Combinatorial Optimisation Problems." International Journal of Advanced Computer Research **2**(4): 300.

Najjarzadeh, M. and A. Ayatollahi (2008). A comparison between genetic algorithm and PSO for linear phase FIR digital filter design. 2008 9th International Conference on Signal Processing, IEEE.

Patel, H. and G. S. Thakur (2017). "Classification of imbalanced data using a modified fuzzy-neighbor weighted approach." International Journal of Intelligent Engineering and Systems **10**(1): 56-64.

Popel, M. H., et al. (2018). A Hybrid Under-Sampling Method (HUSBoost) to Classify Imbalanced Data. 2018 21st International Conference of Computer and Information Technology (ICCIT), IEEE.

Siddiqui, M. A., et al. (2013). Discovering the thematic structure of the Quran using probabilistic topic model. 2013 Taibah University International Conference on Advances in Information Technology for the Holy Quran and Its Sciences, IEEE.

Tang, Y., et al. (2008). "SVMs modeling for highly imbalanced classification." IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics) **39**(1): 281-288.