A Classification Model For Class Imbalance Dataset Using Genetic Programming

Article *in* IEEE Access · May 2019

DOI: 10.1109/ACCESS.2019.2915611

4 authors, including:

Amaad Mirza
COMSATS University Islamabad
4 PUBLICATIONS **168** CITATIONS

SEE PROFILE

Sohail Asghar
COMSATS University Islamabad
141 PUBLICATIONS **1,615** CITATIONS

SEE PROFILE

Muhammad Noor
COMSATS University Islamabad
20 PUBLICATIONS **426** CITATIONS

SEE PROFILE

IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received April 12, 2019, accepted May 2, 2019, date of publication May 8, 2019, date of current version June 11, 2019.

Digital Object Identifier 10.1109/ACCESS.2019.2915611

A Classification Model For Class Imbalance Dataset Using Genetic Programming

MIRZA AMAAD UL HAQ TAHIR, SOHAIL ASGHAR, AWAIS MANZOOR, AND MUHAMMAD ASIM NOOR

Department of Computer Science, COMSATS University Islamabad, Islamabad 45550, Pakistan

Corresponding author: Mirza Amaad Ul Haq Tahir (amaadmirza@gmail.com)

ABSTRACT Since the last few decades, a class imbalance has been one of the most challenging problems in various fields, such as data mining and machine learning. The particular state of an imbalanced dataset, where each class associated with a given dataset is distributed unevenly. This happens when the positive class is much smaller than the negative class. In this case, most standard classification algorithms do not identify examples related to the positive class. A positive class usually refers to the key interest of the classification task. In order to solve this problem, several solutions were proposed such as sampling-based over-sampling and under-sampling, changes at the classifier level or the combination of two or more classifiers. However the main problem is that most solutions are biased towards negative class, computationally expensive, have storage issues or taking long training time. An alternative approach to this problem is the genetic algorithm (GA), which has shown the promising results. The GA is an evolutionary learning algorithm that uses the principles of Darwinian evolution, it is a powerful global search algorithm. Moreover, the fitness function is a key parameter in GA. It determines how well a solution can solve the given problem. In this paper, we propose a solution which uses entropy and information gain as a fitness function in GA with an objective to improve the impurity and gives a more balanced result without changing the original dataset. The experiments conducted on different datasets demonstrate the effectiveness of the proposed solution in comparison with the several other state-of-the-art algorithms in term of Accuracy (Acc), geometric mean (GM), F-measure (FM), kappa, and Matthews correlation coefficient (MCC).

INDEX TERMS Imbalanced dataset, Information gain, entropy, fitness function, genetic algorithm.

I. INTRODUCTION

Over the past few decades, imbalanced problems have become one of the challenging problems in the area of data mining, machine learning [44]–[46]. As far as binary classification problems are concerned, a dataset is referred to as imbalanced if the sample quantity is not equal in both classes [43], otherwise, this problem is not very important if there is a very few difference between positive and negative class samples. Moreover, the positive class as compared to negative class is much smaller in term of sample size. Moreover, positive class generally considers the main class of interest in imbalanced problems. However, most traditional classification algorithms cannot overcome this problem as they are designed to achieve overall high accuracy, although they are more likely to mis-classify samples of positive class as a negative class. To find a good solution for both the positive and negative classes with good accuracy has become an important area of research.

Addressing this learning bias issue, the approaches developed over the last few years can be categorized as follows. The first approach is data-based approach for sampling methods, the objective for this approach is to manage or produce the well balanced dataset from the imbalanced training dataset. The sampling method includes under-sampling and oversampling. Under-sampling approaches reduce samples of the negative class, thereby helping to facilitate the learning process. While, over-sampling increases the samples of the positive class in order to achieve the desired balance level. The main drawbacks associated with these approaches are; (i) in the case of under-sampling, the useful objects are excluded from the negative class, and (ii) the over-sampling contains artificial duplicate samples that could create over-fitting. Both disadvantages affect the efficiency and performance of the classifier at the classification stage [50]–[53].

The associate editor coordinating the review of this manuscript and approving it for publication was Fatih Emre Boran.

The second approach is algorithm-based approach. The objective of this approach is to modify the classifier in order to pay more attention to the positive class. Some examples including re-sampling methods or classifier's ensemble in combination with boosting-algorithms. Furthermore, this technique isn't flexible as compared to the data level techniques, since it depends largely on a particular classifier, the modifications are therefore designed to solve the problem of class imbalance for a particular classifier [47]–[49]. Another important alternative approach for handling the class imbalance problem is cost-sensitive based approach, the idea is to assign mis-classification costs via cost matrix. The main drawback of this approach is that the cost of errors for different classes is not even in most imbalance problems and is usually unknown in most databases [41], [42]. Ensemble based approaches systematically combine data level and algorithmic level methods. However, the main drawback is adding more classifiers increases computational complexity.

In order to overcome this problem, an alternative approach is GA. GA is a powerful global search algorithm and a promising data mining and search technique for classifiers that successfully solve the range of classification problems [35]. In addition, the cost adjustment in GA can be applied to the fitness function. In the fitness function common techniques are used either by improving performance criteria or by using fixed classification costs for negative and positive class examples [40]. In previous studies such as Syafiq et al. [55] proposed a model to solve the class imbalance problem by using Genetic programming (GP) and SVM. They modeled the class imbalance problem as an optimization problem. In order to evaluate their model, they use wilt disease dataset and compare the accuracy of their model with different classifiers. Their proposed model has good accuracy as compare to existing classifiers. But the main drawback is long training time due to computational overhead. SVM has the problem of space complexity when the data size is large which also increases training time. In [58], the authors proposed a novel Diversified Error Correcting Output Code (DECOC) algorithm for solving multi-class imbalance problem. They combined the improved Error Correcting Output Codes (ECOC) to find the best classification algorithm to tackle class imbalance and diversified ensemble learning framework. The proposed technique is applied on 19 different publicly available datasets and compared the performance of their algorithm with 17 other state-of-the-art imbalance algorithms. The proposed algorithm effectively achieves the best overall performance but the weak point of DECOC is that it is computationally very expensive.

To measure the class certainty, entropy is an effective approach to information theory. Shannon [39] stated that entropy is used as a negative-logarithmic-function that finds the possibility of the occurrence of an event. Whereas information gain is used to the possible reduction in entropy. For classification task, several researchers used Shannon entropy for classification tasks [36]–[38].

For handling the class imbalance dataset problem, the scope of this research is to deeply investigate the imbalanced classification problem that we are trying to solve from a perspective of effectiveness (the ability to accurately classify an unknown dataset) and efficiency (the speed of classifying data). At present, the state-of-the-art techniques performs at data level (sampling-based under-sampling and over-sampling) [12], [27]–[29], [31], ensemble-based techniques [18], [24], [26], [30], [32], and cost-sensitive based techniques [15], [16]. However, the effectiveness and efficiency of imbalanced data learning was not fully considered. In this research, the base assumption is that classification accuracies of both classes are equally important. The theme of this research is to surpass the existing techniques.

In this paper, we proposed a new solution that uses entropy and information gain as a fitness function in GP with an objective to solve the class imbalance problem without modifying the original dataset. For doing this, we consider that the imbalance is basically impurity and the impurity is basically measured by entropy which in turn is calculated by information gain. We then calculate the fitness function to evaluate the adopted solution for classifying the imbalanced data set. This novel technique will reduce the imbalance classification problem that equally deal with positive and negative classes.

Some key objectives of the proposed research are as follows:

1) The core objective of this research is to find the way in general to deal with class imbalance dataset classification problem. In this regard, we have proposed a new approach with an objective to improve the impurity in the dataset by using entropy and information gain as a fitness function in GP.
2) Proposed fitness function will help to measure the impurity and to pure the dataset for further classification which will helps to determine the actual cause, this may be applicable for many practical applications such as anomaly-detection [61], risk-management [60], software-engineering [62], social-media-mining [63], and medical diagnosis etc.
3) All learning data are considered useful and should not be excluded (techniques of external data balancing can eliminate useful learning examples in training).
4) Analyze the effect of using new approach on original datasets and to improve the classification training time. Also to check the effectiveness of our approach as compared to other state-of-the-art algorithms.
5) Analyze the impact of using new approach and evaluate the classifier's performance by using Acc, GM, FM, kappa, and MCC.

The remainder of the paper is organized as follows: Section 2 discusses the problem statement. Section 3 provides research contributions. Section 4 provides research questions. Section 5 provides a detailed literature on imbalanced problems and discusses the advantages and disadvantages of each solution. The proposed method is provided in Section 6. Section 7 provides the experimental setup.

Several experiments and discussions are presented in Section 8 and Section 9 contains the final comments of our work.

II. PROBLEM DESCRIPTION

The classification problem of highly imbalanced class distributions in datasets can pose a significant challenge for data mining and machine learning domain. The imbalance dataset degrades the performance, overall accuracy and classification decisions are often biased towards the negative class that leads to the misclassification of the positive class samples. It also treats them as noise. In many applications, lower instances of a class are relatively more important and interesting ones [59].

Class imbalance causes many difficulties that hinder the performance of data mining and machine learning techniques [59]:

Firstly: the lack of data, few positive class samples in the training set tends the classifiers to falsely detect them and the decision boundary is far from the true one.

Secondly: the class distribution, the standard classifiers assume that the training samples are equally distributed between classes. However, the positive class ratio is very low in many real-world applications (e.g. 1:100, 1:1000 or more than 1 to 10000).

Lastly: the cost of errors is uneven and is usually unknown for different classes.

In previous studies such as Syafiq et al. in [55], proposed a model to solve the class imbalance problem by using GP and SVM. They modeled the class imbalance problem as an optimization problem. In order to evaluate their model, they use wilt disease dataset and compare the accuracy of their model with different classifiers such as SVM, GSVM and SMOTE-SVM. Four main tasks are included in the proposed algorithm i.e., attributes generator, attributes selector, train the decision function of SVM and evaluate the fitness function of SVM. Their proposed model has higher accuracy as compare to existing classifiers. However, the main drawback is long training time due to computational overhead, SVM has problem of space complexity when data is large, which also increase training time. Moreover, biggest things to understand about GAs is that the power come from the crossover, not from mutation. Most researchers seem to think that having a large population is more important than the mutation. In [58], proposed a novel DECOC algorithm for solving multi-class imbalance problem. They combined the improved ECOC to find the best classification algorithm to tackle class imbalance and diversified ensemble learning framework. The proposed technique is applied on 19 different publicly available datasets and compared the performance of their technique with 17 state-of-the-art multi-class imbalance algorithms. The proposed algorithm efficiently achieves the best overall performance with less computational complexity but the weak point of DECOC is that it is computationally very expensive.

In this paper, we present a new solution that uses entropy and information gain as fitness function in GP by considering that the imbalance is basically impurity and the impurity is basically measured by entropy which in turn is calculated by information gain. Furthermore, we are trying to solve this problem from a perspective of effectiveness (the ability to accurately classify an unknown dataset) and efficiency (the speed of classifying data). Moreover, we used original datasets without any preprocessing techniques to solve the imbalance datasets problem to achieve our objective. For handling the class imbalance problem, we use different datasets in our work as mentioned in Table 4.

III. RESEARCH CONTRIBUTIONS

In this paper, we develop a new approach that uses entropy-gain-based fitness function in GP for imbalance data learning. Our proposed approach Entropy and Information Gain based fitness function in Genetic Algorithm (EIG-GA) mostly performed well as compared to the existing approaches from experimentation. We achieve average FM score of 89.8% compared to a baseline of 78% and 83.3% and average GM we achieve 91% compared to a baseline of 83.5% and 87%, and average Kappa we achieve 86.4% compared to a baseline of 75.7% and 81.3%, and average MCC we achieve 85.3% compared to a baseline methods of 73.6% and 78.9%.

Our contribution in this research are as follows:

1) The paper shows how classification problems can be addressed with imbalanced data by using GP, with a focus on improving the imbalanced classification problem rather than the traditional methods of data balancing. This paper also shows that configuring the fitness function in GP is more important for improving the positive and negative class performance.
2) This paper uses fitness function based on entropy and information gain in GP to perform cost adjustment between the positive and negative class accuracies, which allows the imbalanced dataset to be applied directly in the learning process without first re-balancing the data (through sampling). By using this objective in the learning process, we can improve classification in both classes in term of Acc, GM, FM, kappa, and MCC. On these measures, we then compared our algorithm with state-of-the-art algorithms.
3) The novelty of this approach is that an entropy-gain-based fitness function for imbalanced data is used directly in the learning process and achieve sufficient level of accuracy.
4) In order to get more generalized the concept, we utilize the datasets from KEEL [77] and UCI [78] repositories. We then comprehensively compared with state-of-the-art algorithms in order to analyze the impact of using new approach and evaluate the classifier performance by using these measures Acc, GM, FM, kappa, and MCC.

5) This research tries to attempt implementation with 30 available datasets that show the impact of taking the class imbalance problem into account and analyze the impact of using new approach compared with state-of-the-art techniques in term of effectiveness and efficiency.
6) By using EIG-GA, the proposed method supersedes the existing state-of-the-art approaches in achieving high accuracy of almost 90% on all datasets that we used in this paper.
7) The proposed process helps specifically in remote sensing and medical field to analyze and prediction upon decision making.

IV. RESEARCH QUESTIONS

The proposed study also focuses on the following research questions:

1) From algorithm level approach, which classifier is best fit for imbalance problem?
2) From data level approach, which approach is best fit for imbalance problem?
3) From ensemble-based approach, which classifiers combined to perform well on imbalance dataset problem?
4) Is it good to make an artificial data for balancing the dataset by using sampling methods?
5) What is the best individual classifier that fit for imbalance dataset problem?

V. RELATED WORK

In literature, several optimized solutions and algorithms are given to solve the problem of the imbalanced class. In order to deal with the imbalance problems, there are four possibilities in the learning stages: first, the technique of re-sampling (mainly changes in the distribution of classes); Second, the choice of features that are at the data level; The third is the classifier's level or algorithms (handled internally in the algorithm) and the fourth is ensemble learning (a combination of two or more classifiers). In addition, mainly the focus is to address the imbalance problem either by modification in the data level, which may be obtained by sampling methods or at the algorithm level. The following are the approaches to deal with class imbalance problem.

A. DATA LEVEL

1) SAMPLING BASED TECHNIQUES

Sampling is one of the most commonly used approaches to validate data mining and machine learning models, also known as data preprocessing, which handles the imbalance problem by balancing the training data-set [9], [10]. Two approaches are used to make a balanced data-set which are under-sampling and over-sampling.

Under-Sampling Techniques:

One of the most common and simplest strategies to balance the data-set by randomly eliminating samples of the majority class till the minority and majority classes were balanced. However, this technique can cause losing useful information.

Yen and Lee [13] propose a cluster-based under-sampling approach that improves the accuracy of classification for the positive-class. They divided the training data into groups and then delegated data is selected from each group for the negative class in relation to the proportion of the negative-class samples for positive-class samples. Their effects showed that cluster-based testing improved predictability accuracy in addition to being more stable than another sampling method. Their results show that cluster-based sampling increases the accuracy of prediction and is more stable than other sampling methods.

Yua *et al.* [27] proposed ant colony optimization (ACO) based an under-sampling technique. This particular technique provided the best and most optimal balance established, although this technique requires a lot of time, instead of a simple sampling technique. Barandela *et al.* [11] proposed a method i.e., wilson's editing, an under-sampling method. It uses k-nearest neighbors from each majority class sample based on 3-NN. If the sample is poorly classified, it is excluded from the final set of data that represents the negative-class.

In [5], Rahman and Davis considered the method proposed in [4], used to separate samples in the k-cluster from the majority class and select a subset for each cluster. All subsets are then combined separately for the positive-class in order to get different training data sets. However, under-sampling is generally used, which can result in the loss of useful information, eliminating important patterns. In [34], Diao *et al.* proposes an under-sampling technique that compresses the training dataset with minimum loss of information. The main purpose of their work is to make a trade-off between training dataset size and loss of information.

Over-Sampling Techniques:

Over-sampling is used when the amount of data is insufficient for classification. It tries to balance the data by simply creating copies of typically existing samples or adding more samples to the minority class. However, this approach may cause over-fitting and computationally expensive. To deal with this problem, SMOTE based method [6] that generates synthetic samples instead of replication of existing positive-class samples. The proposed solution increases the classifier performance and learning biased to the minority-class. However, they applied this solution only for binary class samples, this can also lead to overlap between classes, Moreover, it does not take into consideration the neighbors of positive-class samples. By using SMOTE, [6] generated new samples from the original ones for the minority class for further generation. While in [7], they modified the technique in [6] and new over-sampling technique is proposed also called incremental SMOTE examining the generated samples of a synthetic minority for further generation. In [29] author's analyzes the effect of sampling approaches on highly imbalance dataset. Their experiments showed that performance depends on the

number of samples in the training-set. However, over sampling is time consuming.

In [3], Verbiest *et al.* proposed a new technique using a combination of SMOTE with fuzzy rough set theory. This technique enhances the efficiency of SMOTE with the help of removing those samples that have a low value to the fuzzy region. Recently, a new SMOTE combined with distance based under-sampling is proposed in [12].

Although the advantages of using SMOTE is to efficiently balance the data. However, it may cause noise in the data-set and other problems. In order to overcome this problem, an active learning SMOTE [28] approach is proposed, that chooses the particular best and useful samples for learning. In [31], Burnaev *et al.* discuss various ways to improve the imbalance problem by re-sampling the dataset, i.e., add or remove element from the dataset. They also investigate the impact of re-sampling on classification accuracy. Their results show that re-sample may or may not improve the classification accuracy depending on the specific task. Other ways not to re-sample the data side but to build new method in order to improve quality. Juan and Li-li [8], proposed a new over-sampling technique that uses clustering and GA. Reference [1], introduced a new wrapper based over-sampling technique. For this purpose, they used GA as a optimization search engine to find the best regions with regard to over-sampling.

B. COST-SENSITIVE BASED LEARNING METHODS

Cost-sensitive based learning approaches are designed based on cost that is imposed on a classifier when a mis-classification take place. Several studies are related to cost-sensitive learning for imbalance class distribution, like Aouada [15], improves Random Forest (RF), and integrates cost-sensitive learning for weighted RF and sampling methods for balanced RF. Kothandan [16], studied the usage of cost-sensitive learning and SMOTE sampling with two SVM steps. To deal with class imbalance, author proposes two empirical approaches [17]. The first approach is a combination of cost-sensitive with sampling whereas, the second approach optimizes the cost ratio by using cost-sensitive learning approach. They concluded the final outcomes and analyzed that the cost matrix decreases with the use of the first approach, but the second approach has better performance.

C. ONE-CLASS LEARNING BASED METHODS

In this method, the classifier learned only from the target class (minority class samples). This method increases the classifier's performance on unseen data that belongs to minority class. One-class learning can perform better when dealing with imbalanced data. Recently, Kim and Ahn [19] proposed a method that combined under-sampling with one-class SVM (OSVM). They used the nearest k-inverse neighbors to remove the emissions, to solve the problem of parameter selection in unbalanced data. However, Decision Tree (DT), Naive Bayes (NB) and many other classifier's cannot be built based on recognition-based method.

D. ENSEMBLE-BASED METHODS

Ensemble learning is one of the most frequently used approaches to solving the problem of class imbalance [24]–[26]. The main goal of ensemble is to enhance the overall performance of a single classifier. The best-known methods of the ensemble are bagging and boosting. To promote the model variance, each model is trained by bagging technique [20].

In [21], Khoshgoftaar *et al.* theoretically studied the utilization of different data sampling along with Boosting including SMOTE, Borderline SMOTE, under-sampling, over-sampling and Wilson's editing. From their analysis, they came to the particular conclusion that Boosting increases overall performance over-sampling strategies while the most effective performance is usually obtained by under-sampling.

In [32], Patel *et al.* proposed a new approach to the hybrid fuzzy weighed nearest neighbor for search, a better overall classification efficiency for both the minority and the majority classes of unbalanced data. Fuzzy classification helps to classify objects more correctly because it determines how much the object belongs to the class. Their empirical results show the improvements in the classification of unbalanced data with a different ratio of imbalance, compared to other approaches.

In [30], there is not a single solution that sufficiently solves the imbalance problem. They claimed that a number of methods exist for solving the problem of imbalanced data for specific areas of application for specific tasks. Their particular current work is basically an enhancement of entropy-based classifiers with weights multiplication in order to fix the imbalance class problem at the classifier's level.

Ricardo *et al.* [18] discussed that the solution to class imbalance problem through pairwise rankers improves efficiency of training data. They also show that by combining other approaches with these pairwise models improve performance and balance the dataset. But the problem is training time is usually higher on the very big dataset.

Van Loi *et al.* [2], addresses the problem of imbalance in the credit card data-set and solve this problem by using GA. They proposed two new fitness function based on previous studies. Their experiments show that GA overcomes the imbalance problem and without GA accuracy drastically decreases.

Khoshgoftaar *et al.* [22] and Govindaraj and Lavanya [23] combined random under-sampling with AdaBoost (RUSBoost). By using RUSBoost, they randomly eliminate samples from the majority-class to obtain the desired distribution. In [14], performance have been enhanced by applying SMOTE together with AdaBoost, and the target function uses optimization approach, such as GA.

E. OTHER RELATED WORKS

In [79], Bartosz *et al.* concentrate on ensemble method for addressing multi-class imbalance problem by applying an adaptive training algorithm. Their algorithm focuses on

creating local ensembles of the classifier and split input features space into number of clusters. Weighted-sum method is used for combination to control the degree of distribution. They applied their proposed technique on Multi-class imbalanced dataset, available on KEEL [77] repository. The main feature of their research is to create a local competence and delegation classifier and applied weighted-sum combination independently in each cluster.

A novel algorithm DECOC [58] for solving multi-class imbalance problem combines the improved ECOC to find the best classification algorithm to tackle class imbalance and diversified ensemble learning framework. The proposed technique is applied on 19 different publicly available datasets and compared the performance with 17 state-of-the-art multi-class imbalance algorithms. The proposed algorithm effectively achieves the best overall performance with best accuracy, but it is computationally expensive.

Data gravitational classification is proposed in [73] for solving the classification in which matrix of weights is associated with importance of each attributes. The proposed algorithm improves the performance in making decision boundaries by considering both local and global information. The proposed algorithm is evaluated on 35 standard and 44 imbalanced data sets. From the experimentation they achieved high accuracy, area under the curve (AUC) and Cohen's kappa rate as compared to other state of the art imbalance classification methods.

In [74], GP is applied for predicting the academic failure from high dimensional and imbalanced data. Authors collected data of the students enrolled at Academic Unit Preparation at autonomous University of Zacatecas, where student are in the age of 15 to 18 years, however the collected data contains only information of students enrolled at their first years of study with age 15–66 years. Then they preprocessed their data by applying data cleaning, data partitioning and variable transformation. Their results reveal that approaches for selecting the best features, data balancing and cost-sensitive classification have a high impact for improving the accuracy.

Entropy-based matrix learning machine (EmatMHKS) proposed by [64] for imbalanced dataset, with a fuzzy membership evaluation approach. The proposed technique has been evaluated on 10 real-world datasets including Pima-Indians, Page-blocks, Abanole19, Yeast, miRNA, Satimage, Ecoli, Transfusion and Haberman. Authors used accuracy and computational complexity as evaluation metrics. They also evaluated the performance of their proposed algorithm on statistical comparisons and applied Friedman–Nemenyi test.

In [65], Changming and Wang have proposed a novel SVM-based approach for handling two major problems namely, class imbalance and high dimensionality. The approach has an advantage to improve predictive performance of SVM formulations which deals with highly imbalanced data by eliminating irrelevant features. The proposed techniques have been evaluated on 12 microarray datasets for binary classification which include GORDON, GLIOMA, SRBCT, BHAT3, BHAT1, BHAT2, CAR2, BULL, CAR1, BHAT4, CAR3 and CAR4. Results shows that feature selection improves classification performance, AUC was improved on five out of six datasets, however the results were slightly worse on GLIOMA dataset as compared with that obtained without feature selection. This clears that gain in performance is not the only reason for performing feature selection. Another limitation of this work is that the proposed algorithm is unable to handle multi-class classification.

A class certainty based fuzzy membership evaluation has been proposed in [68] for handling class imbalance problem. Entropy-based fuzzy SVM is proposed to handle the samples with higher class certainty which signifies the importance of high-class certainty. Authors used imbalanced dataset from KEEL imbalanced datasets benchmark repository and evaluated the performance on 64 benchmark datasets. They classified datasets into three groups: low imbalanced, medium imbalanced and high imbalanced datasets. Results are compared with FSVM, SVM-SMOTE, SVM-OSS, SVM-RUS, SVM, Easy-Ensemble, AdaBoost and 1-NN.

Class imbalance problem addressed with biased SVM and weighted SMOTE proposed in [70]. Biased SVM combined with SMOTEBoost for handling the class imbalance problem, as biased SVM gives better control over sensitivity as compared to SVM. To address the problem of considering all minority data samples, authors used weighted SMOTE-Boost and evaluated on two datasets, showing better results in terms of accuracy and sensitivity. Table 1 shows the advantages and disadvantages of proposed methods based on four approaches (under-sampling, oversampling, cost-sensitive, ensemble) with regard to be able to handling the class imbalance problem and Table 2 shows learning approaches of previous works on class imbalance classification.

VI. PROPOSED METHOD

In this section, we describe the proposed procedure for addressing the problems associated with imbalanced datasets. Most classification decisions are often biased towards the negative class because of the imbalance distribution of data, which lead to the misclassification of positive class samples. Our main objective of this paper is to improve the classification accuracy of the positive class by avoiding the drawbacks of the existing methods described in the previous section.

The proposed solution enhances the GA based on the class imbalance problem. Therefore, let the problem that we are attempting to solve first be formulated. Let Y denotes the initial data set, with

$$Y^1 = \{y_1^1, y_2^1, \ldots y_n^1\} \subset Z$$

is a subset of n_1 positive class records denoting 1. While

$$Y^0 = \{y_1^0, y_2^0, \ldots y_n^0\} \subset Z$$

is a subset of a negative class of n_0 records that denotes 0. In case of an imbalanced class dataset, we have $n_1 < n_0$ which, if left unhandled can negatively affect the efficiency of a classifier. We start with Shannon entropy in our work,

TABLE 1. Advantages and disadvantages of proposed methods on class imbalance classification.

Techniques		Advantages	Disadvantages
Sampling Techniques	**Under-Sampling**	➢ Simple approach and widely used in many area applications. ➢ Can easily be implemented. ➢ It can help to improves the problem associated with run time and storage, simply by eliminating samples from the training data.	Random under-sampling may be a biased when choosing samples from the population. It may loss of useful information, which could be important for building classifier's rule. Therefore, inaccurate results get on actual test data-set.
	Over-Sampling	Rather than under-sampling the advantage of this technique leads to no information loss.	➢ Risk of over-fitting, because it replicates the minority-class samples. ➢ additional computational cost. ➢ Time consuming.
Cost-Sensitive Learning Techniques		Simple and effective methods	Ineffective if the actual cost of errors is not known.
one-class learning Technique		Simple and fast processing method	➢ Decision tree, NB and many other classifiers cannot be built by one-class learning. ➢ It is not really effective when applied together with classification algorithms that should be learned from the prevalent class.
Ensemble Technique		Overall much better classification efficiency rather than single classifier.	➢ Add more classifiers grows more complexity ➢ Time consuming (Learning Time) ➢ Over-fitting.

because entropy is a widely used in information theory. Initially entropy is used to characterize the impurity of an arbitrary set of examples or we can say entropy is a measure of uncertainty, impurity or disorder contained in each amount of information that received. Moreover, lower entropy provides more certain information. Whereas, information gain Eq.(1) is a measure of change in entropy or expected reduction in entropy.

$$E(D) = -\sum_{i=1}^{k} p_i \log_2 p_i \tag{1}$$

From Equation(1), where k is the number of classes defined in the data-set and P_i is the proportion of samples related to each class C_i from initial data-set D. If P_i is equal to 0 means if no sample related to that class in the data-set, then from Equation(1) it does not return undefined because it is multiplied by another zero P_i.

$$Gain(D, S) = E(D) - \sum_{v \in Values(A)} \frac{\mid D_v \mid}{\mid D \mid} E(D_v) \tag{2}$$

From Equation(2) where $Gain(D, S)$ represents the expected reduction in entropy due to sorting on S. Here D_v is the sum of the entropies of each subset that are weighted by the fraction of the examples $\frac{D_v}{D}$ belonging to D_v.

As far as our problem statement is concerned, we consider that an imbalance is basically impurity and the impurity is basically measured by entropy and the entropy is calculated by information gain. Therefore, if we obtain the maximum information gain on each sample in the population, this means that the impurity is removed from the data-set and proceeds to clean data. Moreover, entropy tackles much better over-fitting and achieves better results than other techniques, but normally increases time complexity. The problem here, is learning time and computational costs, so we decided to use an entropy and gain-based fitness function in GA. Since, GA is an evolutionary algorithm based on the principles or natural selection of Darwin. GA takes minimal time to find a good solution because it has a high convergence rate. Another important thing about GA is that, GA carries out parallel searches in the given solution space and less chance of getting stuck in local optima. Moreover, GA works with a solution-set rather than with one solution compared to other optimization methods. Furthermore, GA handles the complex problems with less computational efforts. In our work, we proposed a new solution based on the fitness function, using entropy and information gain. This EIG-GA is shown in Figure 1. Next, we explain the step-by-step working of our proposed work and it is shown in Algorithm 2.

A. THE EIG-GA

1) POPULATION

The EIG-GA procedure begins with a population that contains random individuals. The size of the initial population can be determined on the basis of the problem. From Table 3, we define a population of 100 individuals and the total number of generations is 50. In addition, each population is generated by the well-known method ramped-half and-half [55].

Algorithm 1 Ramped-half-and-half

Input: *max_depth*, *Pop_Size*

```
for i = 1 to i <= max_depth do
    for j = 1 to j <= Pop_Size/(2 * max_depth) do
        Pop + = Full_initialization(i)
        Pop + = Grow_initialization(i)
    end for
end for
return Pop
```

TABLE 2. Learning approaches of previous works on class imbalance classification.

Technique	Objective	Features	Remarks
SMOTE [6]	increases the classifier performance	proposed solution that biased the learning towards the minority class	the main cause is they applied this solution only for binary class samples.
SVGPM [33]	To improve classification accuracy on wilt disease dataset	Proposed a solution by using GA and SVM.	The main drawback is long training times due to computational overhead. SVM has problem when the data is large due to space complexity, which also increase training time.
Cluster based Under-sampling [5]	To balance the data-set.	The modified cluster-based under-sampling technique is used.	useful where the labels are not fixed.
Active Learning SMOTE [28]	Improved performance on learning models	introduced SVM into a SMOTE learning frame.	Performance is average also risk of over-fitting.
Over-sampling [29]	analysis the effect of sampling techniques	Performance of proposed technique compared to other classifiers	time consuming
GA [2]	Credit card fraud detection	Proposed two new fitness functions	Inscaleable
SMOTE, under-sampling, over-sampling, Boosting [21]	improving classification performance	improves the performance of software quality prediction models. For this purpose, 5 datasets are taken into consideration	Boosting improves the performance while data sampling techniques with boosting not much improve the overall performance. Risk of over-fitting.
k-nearest-neighbor [32]	classification performance for both minority and majority classes.	hybrid fuzzy weighted nearest neighbor is used	improve results depend on imbalance ratio
Entropy [30]	handling problems related with class imbalance.	Entropy based modified C4.5 algorithm is used.	Risk of over-fitting
RUSBoost [22]- [23]	improving classification performance	randomly eliminating samples from the majority-class to obtain the desired distribution.	May loss of useful information.
Adaptive ensemble selection scheme [79]	Maximizing the performance on multi-class imbalanced data	Creating local competence areas and delegating classifiers, and combination through weighted sum	Outperform existing static and dynamic ensemble selection schemes based on clustering
DECOC [58]	Multi-class imbalance classification algorithm with high accuracy	Carried out experiments on 19 datasets to empirically study the performance of DECOC in comparison with 17 state-of-the-art method	High computational complexity
Weighted Data Gravitation Classification [73]	Improve classification accuracy on imbalance dataset, AUC and F-measure	Weight learning for distance weighting to improve classification results	Ignoring noisy attributes and enhancing relevant attributes
genetic programming [74]	Predicting student's failure at school level	selecting the best attributes, cost-sensitive classification, and data balancing	Achieved high accuracy by balancing data and selecting best attributes
EmatMHKS [64]	To enhances the importance of patterns and guaranteeing their importance in getting a more flexible decision surface	fuzzy memberships to evaluate each pattern	Best average performance, Accuracy, F-measure and has less computational complexity
SVM-based approach [65]	Resolve class imbalance and high dimensionality problem	Improve predictive performance of SVM formulations which deals with highly imbalanced data by eliminating irrelevant features	Results were slightly worse for some datasets after selecting best features
EFSVM [68]	Handling class imbalance problem	Signifies the importance of high-class certainty	Robust to noise and outliers
with biased SVM with weighted SMOTE [70]	Weighted-SMOTE for handling class imbalance problem	Eliminate noise using Sets on the Minority Class	Improved accuracy and sensitivity
Geometric (mean) Support Vector Machine (GSVM) [76]	Balanced accuracy between classes for bi - classification problems	Change the bias value of the SVM decision function	Improves performance through cost-sensitive schemes for SVM without adding complexity or computational cost.

2) FITNESS EVALUATION

The main key of the EIG-GA is the fitness function. From Equation, $F_{fitness}(f(x))$ is calculated based on the entropy and information gain of each individual in order to balance the accuracy and manage the generated attributes that lead to new $f(x)$.

In addition, the fitness function is also estimated based upon how many generations are generated to achieve our good fitness to prevent overfitted classification model.

$$F_{fitness} = \frac{(max(Ent), min(IG))}{nGn} \tag{3}$$

The function $f(max(Ent), min(IG))$ is the function that selects the highest value of entropy and the lowest value of information gain, and nGn is the number of generations in $f(x)$.

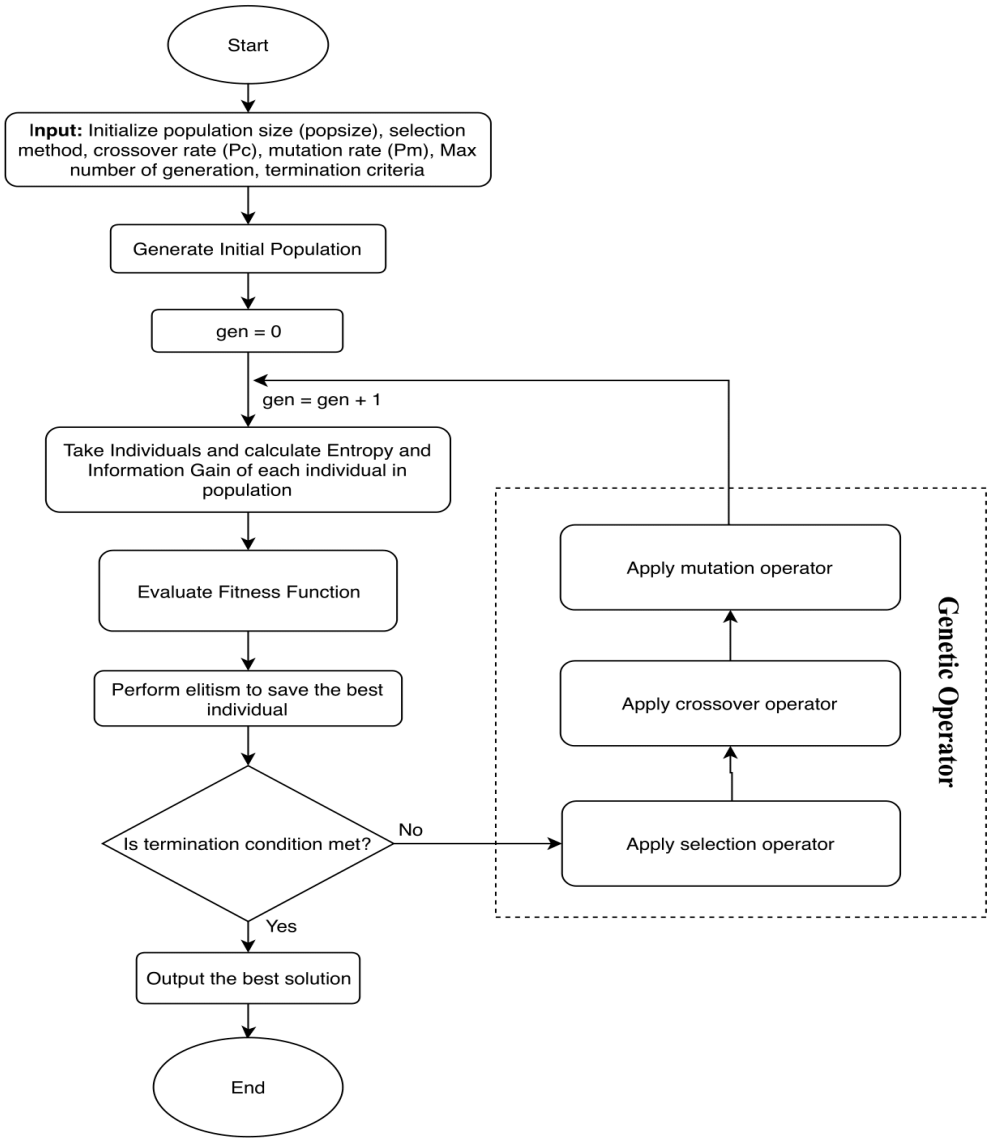

FIGURE 1. The overall flow of the proposed EIG-GA algorithm.

3) ELITISM

Elite selection is usually carried out to preserve the fittest chromosomes. In addition, it is mainly used to ensure that good-quality chromosome do not miss upon next generation when population is updated. We use the elitism strategy to preserve our best fitness scores. Once the elitism step is completed, we move on to the next phase to create a new population again.

4) CREATING A NEW POPULATION

The next step is to generate a new population after each individual fitness is evaluated. The process involves three operations to create a new population which are selection, crossover and mutation. Below is the briefly explanation of these operations for selecting parents and offspring generation.

1) **Selection:** For offspring generation, "Tournament selection method" is used for the selection of parents. First, we created a group of 10 individuals selected randomly from the current population. Furthermore, the best individual based on a fitness assessment was chosen as the first parent to breed a new individual, the same process selects a second parent. The role of parental selection is to distinguish between individuals and give preference to the best individual as next generation parents [56].
2) **Crossover:** The objective of the crossover operator is to create a new offspring from a parent pair [56]. In this work, we have used a uniform crossover (P_{cross}) 0.9 as crossover rate.
3) **Mutation:** In order to maintain the genetic diversity in the population, we have employed random replacement

Algorithm 2 EIG-GA

Input: Population size(Pop_Size), Selection method, Crossover_Rate (P_{cross}), Mutation_Rate (P_{mutate}), Max number of generation (Max_Gen), termination criteria, Fitness Function f(x)

Initialization: Generate Initial Random Population

New_Pop, Pop = 0;

for $k = 1$ to *Pop_Size* **do**
  Pop._add(_get_Random_Individual())
end for
// Evaluating the Fitness for all individual from Pop
Total_Fitness = 0;
for $i = 1$ to *Pop_Size* **do**
  $F_{fitness}$ = pop[i].Evaluate() ——————————————— (3)
  Total_Fitness + = Ffitness
end for
// Perform elitism
for $i = 1$ to *_Elitism* **do**
  New_Pop[count] = pop._getFittest()
  count++ ;
end for
while *termination condition is not met* **do**
  count = 0;
  while *count* < *Pop_Size* **do**
    //Apply Tournament Selection Method
    Individual_1 = Pop.TournamentSelection()
    Individual_2 = Pop.TournamentSelection()
    if $P_{cross} \leq 0.9$ **then**
      Indiv_1, Indiv_2 = Crossover(Individual_1, Individual2)
    end if
    if $P_{mutate} \leq 0.1$ **then**
      Indiv_1.Mutate()
      Indiv_2.Mutate()
    end if
    // Add to New_Population
    New_Pop.add(Indiv_1)
    New_Pop.add(Indiv_2)
  end while
  Pop._Set_Population(New_Pop)
  // Evaluating the Fitness for all individual from Pop
  Total_Fitness = 0;
  for $i = 1$ to *Pop_Size* **do**
    $F_{fitness}$ = pop[i].Evaluate() ——————————————— (3)
    Total_Fitness + = Ffitness
  end for
  // Perform elitism
  for $i = 1$ to *_Elitism* **do**
    New_Pop[count] = pop._getFittest()
    count++;
  end for
end while
return *Pop*

of bit with 0.1 as the mutation rate(P_{mutate}) in this work. In general, the crossover rate remains higher, while the mutation rate remains lower. The lower mutation rate maintains randomness in the population in order to prevent chromosome repetition while the higher rate of crossover prevents premature convergence in the optimum local solution. In addition, the most important thing to understand about GA is that the power comes

TABLE 3. GA Parameter.

Parameter	Value
Pop_Size	100
No_of_iterations	50
Pcross	0.9
Pmutate	0.1
_Elitism	5
Function_set	Add Y + Z Sub Y - Z Mul Y * Z Div Y / Z Logarithm base $\log_{10} Y$ Power Y^Z

from the crossover not from the mutation since most researchers believe that so having a large crossover rate is more important than the mutation. The population fitness is reassessed after crossover and mutation and compared to the previous population fitness. The whole process goes on until the criteria for termination are met. Table 3 shows the parameters for GA.

5) TERMINATING CONDITIONS

For our designed algorithm, we used the following termination conditions if one of the following conditions is met algorithm terminates the search:

1) The fitness value has reached an optimal global value.
2) The number of total generations have reached their limits.

The GA returns the best individual with the best fitness score on testing data, after the whole procedure is completed, we then compared the performance with other modern classification algorithms by using the same training and testing data-set.

VII. EXPERIMENTAL SETUP

In this section, an experimental environment is established for the propose approach to demonstrate the performance on imbalanced datasets, keeping in mind that we used the original datasets in our experiments. To evaluate the performance, 30 datasets are used from the UCI [78] and KEEL [77] repositories and compared the results with state-of-the-art algorithms as shown in Table 6. The running time of our approach with state-of-the-art algorithms on different datasets, along with average results and algorithm ranks are taken as the final outcome to evaluate, which technique has good performance in term of evaluation measures such as Acc, GM, FM, kappa, and MCC to analyze the effectiveness and efficiency. We also compare the execution time (in seconds) in order to demonstrate the efficiency of the proposed EIG-GA method and the results are shown in Table 12. Moreover, some non-parametric statistical tests are used to validate the experimental results [82], [84]. To assess whether there are significant differences in algorithm results, we performed different non-parametric statistical tests such as Iman and Davenport [85], Friedman test, Wilcoxon Signed rank test [80], [81] and Bonferroni–Dunn post-hoc test [85]. These useful non-parametric tests recommended by Demsar [83]. Our proposed approach is implemented in C++ and Matlab, we used Shark machine learning library and Open Beagle for implementation of our approach. The following subsections provides a description of the datasets, about evaluation measurements of the proposed method.

TABLE 4. IR and total instances of imbalanced datasets.

So.#	Dataset name	Source	no of instances	IR
1	wilt_disease	UCI	4889	17.54
2	abalone9-18	KEEL	731	16.4
3	glass1	KEEL	214	1.82
4	ecoli1	KEEL	336	3.36
5	vehicle0	KEEL	846	3.25
6	abalone19	KEEL	4174	129.44
7	phoneme	KEEL	5404	2.41
8	Pageblocks	KEEL	548	164
9	Wine	KEEL	178	1.5
10	monk-2	KEEL	432	1.12
11	balance	KEEL	625	5.88
12	wisconsin	KEEL	683	1.86
13	bupa	KEEL	345	0.73
14	flare-f	KEEL	1066	23.79
15	banana	KEEL	5300	1.23
16	haberman	KEEL	306	2.78
17	hepatitis	KEEL	80	5.15
18	pima	KEEL	768	1.87
19	segment0	KEEL	2308	6.02
20	thyroid	KEEL	720	36.94
21	tic-tac-toe	KEEL	958	1.89
22	vowel0	KEEL	988	9.98
23	yeast1	KEEL	1484	2.46
24	zoo-3	KEEL	101	19.2
25	lymphography	KEEL	148	40.5
26	titanic	KEEL	2201	2.1
27	new-thyroid	KEEL	215	4.84
28	hayes-roth	KEEL	132	1.7
29	dermatology-6	KEEL	358	16.9
30	saheart	KEEL	462	0.53

A. DATASET DESCRIPTION

In this subsection, we used the total of 30 datasets from KEEL [77], and UCI dataset repository [78] for the comparison of the techniques to address the class imbalance problem. Table 4 shows the dataset source and characteristics based on imbalance ratio (IR) and the total instances of each class.

B. EVALUATION CRITERIA

By convention, the data of the minority class is the positive class label and the data of the majority class is the negative class label. Accuracy is used as the main criteria to assess the performance of a classification. But this seems inadequate when dealing with the imbalanced datasets. In this work, we used other evaluation metrics that are described in

TABLE 5. Confusion matrix.

		Predicted class	
		Positive	Negative
Actual class	Positive	TP	FP
	Negative	FN	TN

this section. Furthermore, the performance of our proposed method is assessed and compared with other approaches to these evaluation metrics. Table 5 shows a confusion matrix used in this paper to construct the relevant evaluation measurements.

1) ACCURACY (Acc)

This is the simplest measure to evaluate classification models. It calculates the proportion of correctly classified instances. Formal accuracy has the following definition:

$$\text{Acc} = \frac{Number\_of\_correct\_predictions}{Total\_number\_of\_correct\_predictions} \tag{4}$$

Accuracy can also be calculated as follows in terms of positive and negative:

$$\text{Acc} = \frac{TP + TN}{TP + FP + TN + FN} \tag{5}$$

2) SENSITIVITY (SN)

Sensitivity (SN) also called Recall or True Positive Rate (TPR). SN is the proportion of actual positive examples that are correctly identified by classifier as positive, high SN shows that the class is recognized correctly.

$$\text{SN} = \frac{TP}{TP + FN} \tag{6}$$

3) SPECIFICITY (SP)

SP also called True Negative Rate (TNR). The specificity concerns the ability of the classifier to identify negative outcomes. Consider the medical test example for a certain disease to identify. The specificity of the test is the proportion of patients who don't have the disease and test it negative successfully. In other words;

$$\text{SP} = \frac{TN}{TN + FP} \tag{7}$$

4) PRECISION (PR)

In order to obtain the value of PR, the total number of positive examples correctly classified is divided by the total number of positive examples predicted. A high-precision example labeled as positive.

$$\text{PR} = \frac{TP}{TP + FP} \tag{8}$$

5) G-MEAN (GM)

GM maximizes the classification accuracy of the entire population on the basis of the balance accuracy between the positive the negative. In other words, GM is only high if both classification accuracy is high.

$$\text{GM} = \sqrt{SP * SN} \tag{9}$$

6) F-MEASURE (FM)

FM is used when the performance on both positive and negative classes needed to be high.

$$\text{FM} = 2 \times \frac{PR * SN}{PR + SN} \tag{10}$$

7) KAPPA

Kappa is an important measure of classification performance, especially on imbalanced datasets. It measures how much better the classifier compares to the assumption of the target distribution. The interpretation of the Kappa result from -1 to 1, less than 0.20 shows poor agreement or perfect disagreement; 0.21 to 0.40 shows fair agreement; 0.41 to 0.60 shows moderate agreement; 0.61 to 0.80 shows good agreement; 0.81 to 1.00 shows perfect agreement; Kappa definition is given in Equation (11):

$$\text{Kappa} = \frac{(Po - Pe)}{(1 - Pe)} \tag{11}$$

where

$$\text{Po} = \frac{TP+TN}{TP+FP+TN+FN}$$

$$\text{Pe} = \frac{(TP+FP)}{(TP+FP+TN+FN)} * \frac{(TP+FN)}{(TP+FP+TN+FN)} * \frac{(FN+TN)}{(TP+FP+TN+FN)} * \frac{(FP+TN)}{(TP+FP+TN+FN)}$$

8) MATTHEWS CORRELATION COEFFICIENT (MCC)

MCC can be defined as a correlation coefficient between target and prediction. It usually varies between -1 and $+1$. -1 shows that there is a perfect disagreement between the actual and the prediction, 1 when there is a perfect agreement. The MCC calculates the strength of the classifier by considering all four results of the confusion matrix, it can often provide a more balanced model accuracy assessment, even for imbalanced data sets [56].

$$\text{MCC} = \frac{(TP * TN) - (FP * FN)}{(TP * FP) * (TP * FN) * (TN * FP) * (TP * FN)} \tag{12}$$

VIII. RESULTS AND ANALYSIS

The experimental results of various research studies are presented and discussed in this section. Firstly, we compare the performance of state-of-the-art algorithms on various imbalanced datasets and validate the results using non-parametric statistical tests. Secondly, the computational complexity of different methods in terms of execution time (in seconds) and evaluate the efficiency of EIG-GA with state-of-the-art algorithms. Finally, a discussion is conducted to analyze the results obtained from evaluation metrics.

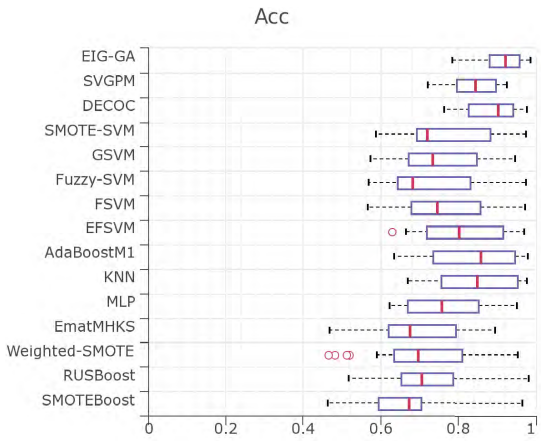

FIGURE 2. The box-plot of 15 algorithms on Acc.

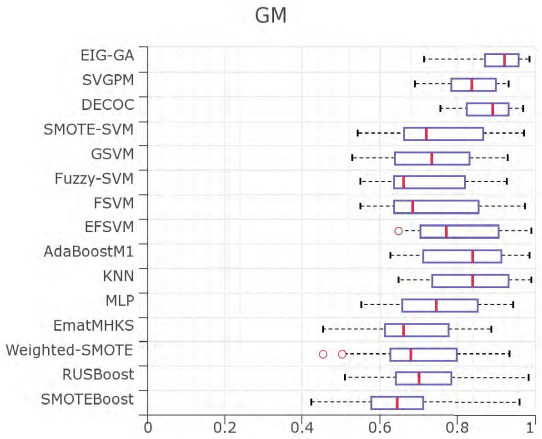

FIGURE 3. The box-plot of 15 algorithms on GM.

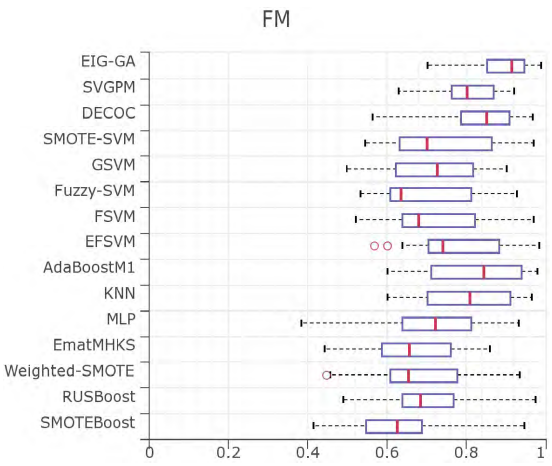

FIGURE 4. The box-plot of 15 algorithms on FM.

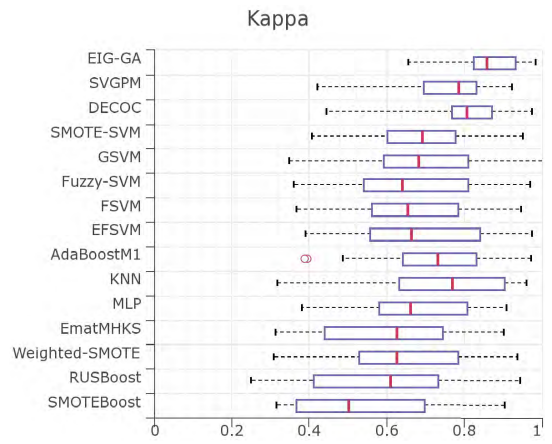

FIGURE 5. The box-plot of 15 algorithms on Kappa.

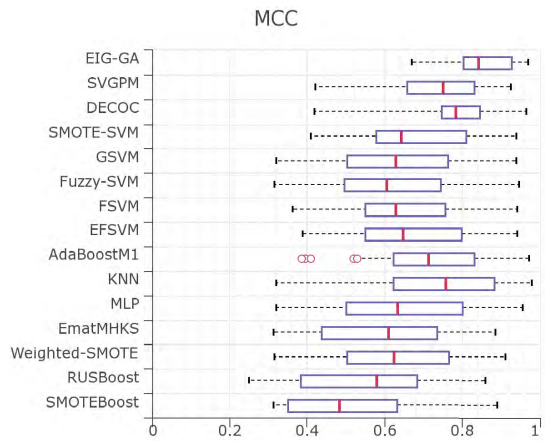

FIGURE 6. The box-plot of 15 algorithms on MCC.

A. PERFORMANCE COMPARISON OF CLASS IMBALANCE CLASSIFICATION ALGORITHMS

In this subsection, the performance of these 15 algorithms is examined, using five different evaluation metrics. In Figure 2 to Figure 6, the ACC, GM, FM, Kappa and MCC performance of each method is shown on all 30 datasets using boxplot graph. The symbol "o" in the boxplot graph means a performance that is abnormal on a dataset, that symbol "o" representing a single dataset.

From Figure 2 to Figure 6, we have the following observations and we also answer the research questions with comments:

1) EIG-GA performs best on all the five measures.
2) CUSBoost is best fit for imbalance problem on data level approach when comparing the accuracy on "flare-F" dataset.
3) Multiple classification algorithms are usually more efficient to combine than to use a single classification approach, as seen in the box-plot graph of Acc in Figure 2.
4) Under and over-sampling usually manage to balance the dataset, however, both techniques have pros and cons but here we can say no it's not good to make an artificial data unless if we have a huge difference between positive and negative classes.

5) From class imbalance perspective SVM can better fit to this type of problems but as we describe earlier in the previous section about cons of SVM.

1) ACCURACY (Acc)

The effectiveness of the EIG-GA on imbalance datasets is examined and compared with SVGPM, DECOC, FSVM, EFSVM, SMOTE-SVM, KNN, MLP, Weighted-SMOTE, RUSBoost, CUSBoost, SMOTEBoost, and AdaBoostM1. For assessment purposes, the results from different datasets (rows) and methods (columns) are shown in Table 6 by using 10-fold cross-validation. The proposed EIG-GA method is superior to other methods in 23 of 30 datasets and achieves competitive Acc results. The best results are highlighted in BOLD for the algorithms on each dataset. From Table 6, the last two rows, the average values of Acc and average ranks for all datasets are listed respectively. It is evident that:

1) In general, we have observed that EIG-GA performs better than the SVM-based methods (i.e., SVGPM, GSVM, FSVM, ESVM, and SMOTE-SVM) on almost all imbalance datasets, which validates EIG-GA method has better classification performance on imbalance datasets. This good classification behavior is based on the fact that the EIG-GA fitness function ensures that the improved Acc is not significantly different from each other, for each GA solution created in EIG-GA. Moreover, interestingly our method has equal accuracy results with DECOC on 2 datasets i.e., "phoneme" and "tic-tac-toe" which shows both methods performs best on these datasets whereas on some datasets we have worst than other methods, but overall on average Acc our method outperforms. From Table 6 we also notice that some state-of-the-art methods have worst performance on some datasets that are Weighted-SMOTE, CUSBoost, RUSBoost and SMOTEBoost.
2) The average Acc of our method is higher than the baseline methods (i.e., SVGPM and DECOC) over the adopted imbalance datasets, this shows that EIG–GA is very useful in handling imbalanced datasets compared to other algorithms.
3) EIG-GA is the leading algorithm based on average ranking results in comparison with 14 other algorithms as in Table 6, which shows its effectiveness on imbalance datasets.

1.1) STATISTICAL COMPARISON

In order to compare statistically the effectiveness of proposed algorithms with the state-of-the-art algorithms, a non-parametric statistical test is conducted i.e., Friedman test. For this test, the average ranks of the compared methods are taken from last row of Table 6. Let k denotes the no# of compared methods that is $k = 15$ and n denotes the no# of imbalance datasets that is $n = 30$ used in the experiments. Let r_i^j be the rank of jth methods on the ith datasets. The average rank of the jth algorithm is calculated as $\mathrm{R}_j = \frac{1}{n}\sum_{i=1}^{n} r_i^j$. According to the null hypothesis that all methods are equivalent, and their rankings should be equivalent to R_j, the Friedman statistic [68], [83]

$$\chi_F^2 = \frac{12n}{k(k+1)}\left[\sum_j R_j^2 - \frac{k(k+1)^2}{4}\right] \tag{13}$$

is distributed according to χ_F^2 with $(k-1)$ degrees of freedom (df), where n and k are reasonably large. Iman and Davenport [85] show the pessimistic behavior of Friedman χ_F^2 [83]. Thus, the statistic

$$F_F = \frac{(n-1)\chi_F^2}{n(k-1)-\chi_F^2} \tag{14}$$

are distributed with $(k-1)$ and $(k-1)(n-1)$ df according to the F-distribution. From Table 6, the proposed EIG-GA method with an average score of 1.63 is noted rank first, the DECOC method ranks the second with the score of 3.60 and so on. To validate that the measured average-ranks are differ significantly from the mean-rank shown by the null-hypothesis $\mathrm{R}_j = 8.0$. From the Friedman test, we have

$$\begin{aligned}\chi_F^2 &= \frac{12\times 30}{15\times 16}\Big[(1.633^2 + 5.867^2 + 3.600^2 + 8.433^2 + 9.717^2 \\ &\quad +10.73^2 + 8.3^2 + 6.483^2 + 6.05^2 + 5.217^2 + 9.367^2 \\ &\quad +11.767^2 + 10.60^2 + 10.233^2 + 12.000^2) - \frac{15\times 16^2}{4}\Big] \\ &= 199.1961\end{aligned}$$

and

$$F_F = \frac{29\times 199.1961}{30\times 14 - 199.1961} = 26.16$$

According to equation (14), for 15 methods and 30 datasets, F_F is distributed according to the F-distribution with $(15-1) = 14$ and $(15-1)(30-1) = 406$ df. The $p-value$ computed by $F(14, 406)$ is less than 0.0001[1] also critical value of $F(14, 406)$ for $\alpha = 0.05$ is 1.716. Since $\mathrm{F}_F > F(14, 406)(26.16 > 1.716)$, the test rejects the null hypothesis and therefore, it can be said that there are statistically significant differences between the Acc result of the methods. Next, we proceed with a Dunn [85] test to find which method provide best results. The CD is formulate as;

$$\mathrm{CD} = q\alpha\sqrt{\frac{k(k+1)}{6N}} \tag{15}$$

The Figure 7 demonstrates the Bonferroni-Dunn test on Acc rate with $\alpha = 0.05$, the CD of which is 3.603. This diagram shows a bar graph, the values of which are proportional to the mean rank obtained from each method [73]. The CD are displayed as a thicker horizontal line [73] and the values

[1]The p-value is computed from the source https://www.graphpad.com/quickcalcs/pValue1

TABLE 6. The Acc result of EIG-GA with 14 other algorithms on 30 imbalance datasets.

Dataset #	EIG-GA	SVGPM	DECOC	SMOTE - SVM	GSVM	FSVM	EFSVM	Ada-Boost M1	KNN	MLP	Emat-MHKS	Weighted - SMOTE	CUS - Boost	RUS - Boost	SMOTE - Boost
1	**0.9040**	0.8940	0.8840	0.8920	0.8540	0.7020	0.7882	0.8820	0.6440	0.7840	0.7884	0.7225	0.8210	0.6544	0.6852
2	0.8568	0.7288	0.8345	0.7036	0.6307	0.6813	0.6840	0.7070	**0.8633**	0.8502	0.8010	0.6458	0.7434	0.8050	0.7029
3	**0.8280**	0.7955	0.7616	0.6993	0.6921	0.6958	0.7140	0.8224	0.8271	0.7009	0.7050	0.6671	0.6272	0.7715	0.6055
4	**0.9238**	0.8968	0.9112	0.8868	0.8545	0.8739	0.9092	0.8422	0.8482	0.8958	0.7700	0.6632	0.6656	0.6770	0.6753
5	**0.9820**	0.8210	0.9202	0.8464	0.8125	0.8330	0.8648	0.8983	0.9385	0.9728	0.7442	0.8002	0.7203	0.7507	0.6228
6	0.9468	0.9003	0.9188	0.6573	0.8223	0.5658	0.5648	0.5796	0.9839	**0.9923**	0.8602	0.5266	0.6176	0.6498	0.5876
7	**0.9204**	0.8740	0.9204	0.6426	0.6723	0.6427	0.6684	0.7768	0.9019	0.8097	0.7702	0.8317	0.7050	0.6671	0.6818
8	**0.9715**	0.8934	0.9687	0.6425	0.5887	0.5803	0.6548	0.7800	0.9471	0.9471	0.9503	0.8832	0.9007	0.9370	0.8592
9	0.9702	0.9059	0.9608	0.8661	0.8224	0.6600	0.6287	0.9045	0.9551	**0.9775**	0.9116	0.7954	0.8115	0.7414	0.6810
10	**0.9509**	0.8458	0.9446	0.9450	0.8897	0.9034	0.9400	0.9606	0.7569	0.7545	0.6608	0.7166	0.6887	0.6914	0.5828
11	**0.9260**	0.7769	0.8800	0.6938	0.7152	0.6436	0.7514	0.7120	0.8544	0.9120	0.6900	0.7758	0.6226	0.5891	0.6800
12	**0.9775**	0.9226	0.9500	0.9739	0.9448	0.9727	0.9702	0.9531	0.9561	0.9634	0.8226	0.6112	0.7700	0.7100	0.5517
13	0.8500	0.7315	0.7700	0.7259	0.6697	0.6815	0.7459	0.6696	0.6318	0.6898	0.6316	0.4822	**0.8558**	0.8168	0.7800
14	0.9590	0.9164	0.9538	0.9450	0.8897	0.9038	0.8715	**0.9596**	0.9474	0.9512	0.7894	0.7844	0.8200	0.7500	0.7007
15	**0.8940**	0.8004	0.8238	0.7586	0.5726	0.6419	0.7476	0.6818	0.8724	0.7209	0.5200	0.6618	0.6216	0.5449	0.4712
16	**0.8360**	0.7217	0.7789	0.6161	0.6640	0.6440	0.6771	0.7320	0.6732	0.7352	0.6605	0.5182	0.8008	0.7884	0.6600
17	**0.8845**	0.8700	0.8256	0.6938	0.7152	0.6436	0.7514	0.8375	0.8000	0.8125	0.7213	0.6626	0.6661	0.5658	0.5841
18	**0.8552**	0.8090	0.8510	0.7148	0.7764	0.7064	0.7090	0.6653	0.7018	0.7513	0.7331	0.5872	0.6679	0.6796	0.6670
19	0.9524	0.9212	0.9447	0.8435	0.8858	0.8184	0.8240	**0.9980**	0.9969	0.9965	0.9000	0.8921	0.9523	0.9805	0.9642
20	**0.9658**	0.8859	0.9219	0.9635	0.8200	0.9298	0.9663	0.9319	0.9000	0.9361	0.8717	0.8214	0.8800	0.8318	0.7800
21	**0.7819**	0.7200	**0.7819**	0.7084	0.6109	0.5787	0.7466	0.7296	0.6885	0.6718	0.7155	0.7715	0.6224	0.7655	0.7019
22	**0.9338**	0.9097	0.9255	0.9104	0.8655	0.8659	0.8949	0.9178	0.9004	0.9579	0.9061	0.8492	0.6800	0.7424	0.6228
23	**0.8268**	0.7944	0.7646	0.6909	0.7008	0.6830	0.6921	0.7839	0.7156	0.7702	0.6472	0.5124	0.7895	0.801	0.7653
24	**0.9842**	0.9120	0.9747	0.7437	0.7528	0.6409	0.7497	0.9405	0.9702	0.9306	0.6558	0.6832	0.7200	0.7017	0.7755
25	**0.9205**	0.8292	0.9018	0.7036	0.6307	0.6387	0.6813	0.7567	0.8040	0.8378	0.7117	0.6114	0.6304	0.6112	0.6702
26	**0.8806**	0.7848	0.8279	0.6496	0.6759	0.6032	0.6676	0.7760	0.7905	0.7787	0.5900	0.7005	0.6606	0.6004	0.4604
27	**0.9802**	0.8456	0.9744	0.9552	0.8254	0.8762	0.9607	0.9348	0.9720	0.9674	0.9307	0.8242	0.8202	0.8861	0.8301
28	**0.8772**	0.7848	0.8066	0.6426	0.6723	0.6427	0.6684	0.4940	0.7272	0.6667	0.4208	0.4656	0.4025	0.5154	0.5500
29	0.8942	0.8228	0.9008	0.7698	0.9000	0.7754	0.6829	0.9000	0.8761	0.8863	**0.9031**	0.6400	0.6410	0.6935	0.6670
30	**0.9483**	0.8745	0.9048	0.5858	0.6445	0.7021	0.6682	0.7121	0.6320	0.6709	0.7905	0.6609	0.5846	0.5554	0.5501
Avg. Values	**0.9127**	0.8396	0.8829	0.7690	0.7524	0.7244	0.7615	0.8080	0.8359	0.8186	0.7524	0.6923	0.7170	0.7158	0.6705
Avg. Ranks	**1.633**	5.867	3.600	8.433	9.717	10.733	8.300	6.483	6.050	5.217	9.367	11.767	10.600	10.233	12.000

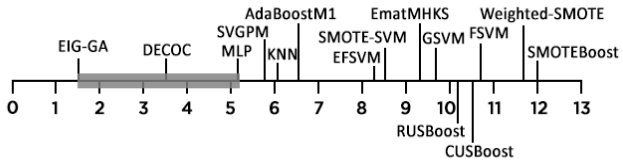

FIGURE 7. Bonferroni–Dunn test for Acc.

above this line are methods which have significantly different results from the EIG-GA control method. EIG-GA performs significantly better than SMOTEBoost ($12-1.63=10.37>3.603$) whereas Weighted-SMOTE ($12-11.77=0.23<3.603$) does not, while SMOTE-SVM is just below the critical difference, but close to it ($12-8.430=3.57\approx 3.603$). Observing from Figure 7, all other methods but DECOC and MLP performs significantly worst than EIG-GA. EIG-GA successfully overcomes the SVGPM method, which obtains the 4th average rank and 3rd average Acc.

2) GEOMETRIC MEAN (GM)

Table 7 shows the experimental results for GM. For assessment purposes, the results from different datasets (rows) and methods (columns) are shown in Table 7 by using 10-fold cross-validation. The proposed EIG-GA method is better than other methods in 19 out of 30 datasets and achieves competitive GM results, but it loses on 11 datasets. The best results are highlighted in BOLD for the comparative algorithms on each dataset. Again, EIG-GA approach performed better than the other approaches in terms of the average GM value and average ranking results. Whereas, we lose on one dataset "hepatitis" result from baseline paper SVGPM and also worst GM result from another baseline paper DECOC on "tic-tac-toe" dataset. Moreover, the proposed EIG-GA approach achieves the average GM of 0.910, which improves from DECOC 4% (0.910 − 0.870), and SVGPM 7.5% (0.910 − 0.835) according to average GM respectively. These results indicate that the generalization characteristics of EIG-GA in both classes are consistent with its GM value, which in the final experiment never falls below 0.90 or 90%. Because the EIG-GA fitness function ensures that the decision function of the overall generation is as low as possible. From Table 7, the last two rows, the average values of GM and average ranking results for all datasets are listed respectively.

TABLE 7. The GM result of EIG-GA with 14 other algorithms on 30 imbalance datasets.

Dataset #	EIG-GA	SVGPM	DECOC	SMOTE - SVM	GSVM	FSVM	EFSVM	Ada-Boost M1	KNN	MLP	Emat-MHKS	Weighted - SMOTE	CUS - Boost	RUS - Boost	SMOTE - Boost
1	**0.929**	0.864	0.912	0.868	0.847	0.771	0.845	0.838	0.683	0.825	0.846	0.773	0.854	0.688	0.720
2	0.846	0.756	0.822	0.710	0.605	0.658	0.659	0.742	**0.863**	0.858	0.818	0.614	0.752	0.828	0.645
3	**0.881**	0.839	0.792	0.689	0.684	0.634	0.668	0.751	0.842	0.720	0.704	0.658	0.632	0.784	0.601
4	**0.958**	0.923	0.94	0.905	0.885	0.874	0.902	0.826	0.822	0.920	0.683	0.664	0.628	0.678	0.672
5	**0.974**	0.778	0.822	0.844	0.810	0.820	0.857	0.872	0.898	0.973	0.741	0.800	0.718	0.747	0.628
6	0.961	0.930	0.906	0.652	0.821	0.554	0.548	0.536	0.982	**0.989**	0.854	0.502	0.596	0.637	0.567
7	**0.907**	0.873	0.902	0.635	0.626	0.617	0.624	0.727	0.901	0.787	0.762	0.817	0.693	0.653	0.681
8	**0.974**	0.884	0.755	0.628	0.567	0.563	0.638	0.700	0.931	0.931	0.943	0.852	0.887	0.937	0.832
9	0.953	0.909	0.958	0.856	0.814	0.650	0.609	0.909	0.951	**0.973**	0.902	0.755	0.805	0.731	0.610
10	0.950	0.827	0.936	0.939	0.857	0.904	0.938	**0.956**	0.709	0.745	0.658	0.704	0.667	0.688	0.522
11	**0.919**	0.759	0.862	0.665	0.702	0.629	0.749	0.672	0.834	0.910	0.652	0.752	0.606	0.589	0.672
12	**0.976**	0.929	0.958	0.971	0.928	0.927	0.972	0.931	0.944	0.934	0.776	0.611	0.752	0.708	0.548
13	0.849	0.715	0.772	0.733	0.661	0.665	0.729	0.689	0.626	0.669	0.609	0.486	**0.859**	0.810	0.767
14	0.949	0.918	0.958	0.936	0.886	0.901	0.863	**0.953**	0.914	0.912	0.752	0.778	0.826	0.742	0.653
15	**0.889**	0.800	0.828	0.765	0.526	0.639	0.716	0.669	0.864	0.679	0.500	0.620	0.636	0.508	0.422
16	**0.832**	0.727	0.779	0.586	0.64	0.640	0.601	0.729	0.662	0.712	0.655	0.522	0.768	0.778	0.635
17	0.843	**0.870**	0.836	0.667	0.712	0.636	0.6514	0.828	0.802	0.805	0.713	0.632	0.628	0.535	0.549
18	**0.921**	0.847	0.944	0.708	0.763	0.674	0.629	0.602	0.700	0.733	0.731	0.572	0.649	0.651	0.647
19	0.948	0.922	0.947	0.832	0.875	0.814	0.624	**0.989**	0.985	0.985	0.903	0.887	0.933	0.982	0.959
20	**0.963**	0.859	0.844	0.953	0.814	0.923	0.954	0.941	0.900	0.932	0.867	0.814	0.86	0.834	0.778
21	0.712	0.689	**0.78**	0.685	0.579	0.551	0.742	0.696	0.645	0.658	0.755	0.773	0.584	0.705	0.679
22	0.926	0.904	0.915	0.900	0.834	0.861	0.879	0.878	0.902	**0.949**	0.901	0.838	0.658	0.734	0.622
23	**0.822**	0.791	0.756	0.659	0.672	0.679	0.692	0.781	0.711	0.768	0.639	0.500	0.773	0.784	0.761
24	**0.985**	0.906	0.969	0.738	0.758	0.640	0.717	0.934	0.962	0.876	0.631	0.615	0.694	0.700	0.767
25	**0.917**	0.819	0.896	0.662	0.618	0.600	0.642	0.747	0.790	0.734	0.721	0.610	0.624	0.612	0.637
26	**0.868**	0.758	0.822	0.646	0.619	0.549	0.637	0.764	0.783	0.742	0.526	0.693	0.546	0.592	0.439
27	**0.983**	0.811	0.892	0.948	0.814	0.868	0.963	0.908	0.97	0.968	0.932	0.808	0.811	0.871	0.801
28	**0.875**	0.779	0.802	0.609	0.652	0.639	0.624	0.451	0.7	0.647	0.4	0.453	0.385	0.516	0.515
29	0.839	0.825	0.892	0.751	0.85	0.772	0.679	0.898	0.837	0.871	**0.903**	0.634	0.619	0.648	0.665
30	**0.946**	0.837	0.900	0.542	0.636	0.687	0.633	0.711	0.629	0.673	0.786	0.658	0.535	0.562	0.549
Avg. Values	**0.910**	0.835	0.870	0.756	0.735	0.711	0.733	0.788	0.825	0.829	0.742	0.680	0.699	0.708	0.651
Avg. Ranks	**1.900**	5.633	3.867	8.133	9.767	10.783	8.900	6.667	5.867	5.300	8.933	11.583	10.750	9.900	12.017

2.1) STATISTICAL COMPARISON

From Table 9, the proposed EIG-GA method with an average score of 1.90 is noted rank first, the DECOC method ranks the second with the score of 3.867 and SVGPM method ranks the forth with the score of 5.633 and so on. From the Friedman test, we have

$$\begin{aligned}\chi_F^2 &= \frac{12\times 30}{15\times 16}\Big[(1.900^2+5.633^2+3.867^2+8.133^2+9.767^2\\ &\quad +10.783^2+8.90^2+6.67^2+5.867^2+5.3^2+8.933^2\\ &\quad +11.583^2+10.750^2+9.900^2+12.017^2)-\frac{15\times 16^2}{4}\Big]\\ &= 189.33\end{aligned}$$

and

$$F_F = \frac{29\times 189.33}{30\times 14-189.33} = 23.80$$

According to equation (14), for 15 methods and 30 datasets, F_F is distributed according to the F-distribution with $(15-1)=14$ and $(15-1)(30-1)=406$ df. The $p-value$ computed by $F(14, 406)$ is less than 0.0001^1 which shows that the test rejects the null hypothesis and therefore, it can be said that the GM results of the compared methods are significantly different on the adopted imbalance datasets.

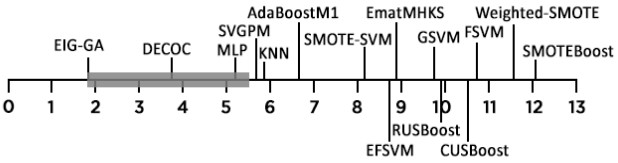

FIGURE 8. Bonferroni–Dunn test for GM.

The Figure 8 demonstrates the Bonferroni-Dunn test on GM with $\alpha = 0.05$, the CD of which is 3.603. All other methods but DECOC and MLP performs significantly worst than EIG-GA. Observing Figure 8, all other methods but DECOC and MLP performs significantly worst than EIG-GA, EIG-GA successfully overcomes the SVGPM method. Interestingly, SVGPM retains the 4th best average rank.

3) F-MEASURE (FM)

Table 8 shows the experimental results for FM. For assessment purposes, the results from different datasets (rows) and

TABLE 8. The FM result of EIG-GA with 14 other algorithms on 30 imbalance datasets.

Dataset #	EIG-GA	SVGPM	DECOC	SMOTE - SVM	GSVM	FSVM	EFSVM	Ada-Boost M1	KNN	MLP	Emat-MHKS	Weighted - SMOTE	CUS - Boost	RUS - Boost	SMOTE - Boost
1	**0.903**	0.862	0.883	0.869	0.819	0.697	0.785	0.834	0.642	0.781	0.780	0.718	0.818	0.654	0.684
2	0.861	0.739	0.842	0.692	0.626	0.642	0.648	0.702	**0.864**	0.828	0.788	0.624	0.702	0.798	0.629
3	**0.865**	0.793	0.454	0.549	0.627	0.608	0.657	0.725	0.828	0.692	0.704	0.658	0.632	0.784	0.541
4	**0.923**	0.894	0.908	0.871	0.852	0.829	0.912	0.846	0.849	0.889	0.683	0.639	0.653	0.643	0.641
5	**0.983**	0.804	0.906	0.845	0.790	0.822	0.849	0.895	0.938	0.952	0.713	0.787	0.702	0.744	0.604
6	0.946	0.901	0.919	0.622	0.783	0.532	0.519	0.528	**0.978**	0.961	0.818	0.484	0.598	0.64	0.539
7	**0.921**	0.868	0.901	0.634	0.600	0.633	0.629	0.711	0.873	0.793	0.737	0.834	0.688	0.653	0.688
8	**0.976**	0.841	0.563	0.628	0.561	0.568	0.641	0.709	0.945	0.911	0.913	0.858	0.879	0.934	0.824
9	0.939	0.868	0.941	0.848	0.800	0.622	0.578	0.883	0.924	**0.943**	0.884	0.732	0.785	0.701	0.562
10	**0.948**	0.802	0.933	0.931	0.833	0.859	0.937	0.919	0.738	0.719	0.632	0.710	0.637	0.682	0.484
11	**0.909**	0.728	0.86	0.597	0.673	0.619	0.742	0.659	0.844	0.908	0.613	0.746	0.541	0.587	0.637
12	**0.970**	0.902	0.918	0.969	0.900	0.907	0.968	0.901	0.937	0.909	0.744	0.581	0.719	0.689	0.544
13	0.847	0.711	0.770	0.713	0.642	0.606	0.720	0.688	0.615	0.600	0.568	0.445	**0.854**	0.782	0.766
14	0.912	0.910	0.898	0.932	0.834	0.889	0.831	**0.950**	0.940	0.878	0.722	0.764	0.808	0.700	0.624
15	**0.877**	0.763	0.800	0.732	0.496	0.604	0.696	0.652	0.867	0.658	0.460	0.620	0.608	0.488	0.413
16	**0.830**	0.677	0.764	0.548	0.640	0.638	0.568	0.702	0.639	0.697	0.653	0.520	0.736	0.744	0.631
17	**0.844**	0.84	0.804	0.610	0.698	0.573	0.650	0.803	0.768	0.768	0.679	0.584	0.643	0.489	0.546
18	**0.923**	0.844	0.919	0.644	0.757	0.615	0.638	0.587	0.688	0.731	0.723	0.548	0.617	0.637	0.643
19	0.943	0.92	0.929	0.830	0.868	0.774	0.602	**0.983**	0.978	0.964	0.869	0.837	0.934	0.972	0.944
20	**0.949**	0.836	0.784	0.952	0.804	0.927	0.934	0.939	0.851	0.916	0.852	0.784	0.820	0.822	0.743
21	0.700	0.627	**0.766**	0.684	0.562	0.544	0.710	0.677	0.600	0.616	0.754	0.738	0.577	0.665	0.619
22	**0.918**	0.889	0.901	0.900	0.830	0.861	0.835	0.810	0.909	0.859	0.878	0.831	0.655	0.701	0.569
23	**0.82**	0.787	0.733	0.633	0.602	0.670	0.688	0.752	0.710	0.711	0.597	0.448	0.749	0.773	0.718
24	**0.983**	0.786	0.966	0.733	0.782	0.620	0.702	0.908	0.946	0.864	0.611	0.598	0.634	0.702	0.728
25	**0.905**	0.789	0.842	0.66	0.548	0.558	0.640	0.733	0.777	0.693	0.700	0.617	0.574	0.612	0.629
26	**0.824**	0.76	0.824	0.646	0.563	0.538	0.637	0.717	0.768	0.74	0.501	0.690	0.521	0.588	0.438
27	**0.987**	0.761	0.89	0.924	0.800	0.855	0.961	0.881	0.967	0.951	0.930	0.806	0.803	0.867	0.798
28	**0.802**	0.719	0.788	0.544	0.65	0.622	0.600	0.438	0.691	0.604	0.382	0.439	0.384	0.511	0.484
29	0.803	0.741	0.732	0.734	0.846	0.760	0.674	0.878	0.809	0.867	**0.894**	0.590	0.607	0.641	0.618
30	**0.940**	0.784	0.843	0.554	0.618	0.687	0.633	0.711	0.629	0.673	0.786	0.658	0.535	0.562	0.549
Avg. Values	**0.898**	0.780	0.833	0.734	0.713	0.689	0.719	0.771	0.788	0.803	0.719	0.663	0.680	0.692	0.628
Avg. Ranks	**1.817**	6	4.6	8.133	9.9	10.533	8.517	6.783	5.25	5.95	8.833	11.283	10.6	9.75	12.05

methods (columns) are shown in Table 8 by using 10-fold cross-validation. The proposed EIG-GA method is better than to other methods in 22 out of 30 datasets and achieves competitive FM results, but it loses on 8 datasets. The best results are highlighted in BOLD for the comparative algorithms on each dataset. Again, EIG-GA approach performed better than the other approaches in terms of the average FM value and average ranking results. Whereas, we have worse FM result from baseline paper DECOC on "tic-tac-toe" dataset. Moreover, the proposed EIG-GA approach achieves the average FM of 0.898, which improves from DECOC 6.5% (0.898 − 0.833), and SVGPM 11.8% (0.898 − 0.780) according to average FM respectively. These results indicate that the generalization characteristics of EIG-GA in both classes are consistent with its FM value, which in the final experiment never falls below 0.89 or 89%. Because the EIG-GA fitness function ensures that the decision function of the overall generation is as low as possible. In the last two rows of Table 8 the average values of FM and average ranking results for all datasets are listed respectively.

3.1) STATISTICAL COMPARISON

From Table 8, the proposed EIG-GA method with an average score of 1.817 is noted rank first, the DECOC method ranks the second with the score of 4.60 and SVGPM method ranks the fifth with the score of 6.0 and so on. From the Friedman test, we have

$$\begin{aligned}\chi_F^2 &= \frac{12\times 30}{15\times 16}\Big[(1.817^2+6^2+4.6^2+8.133^2+9.9^2 \\ &\quad +10.53^2+8.517^2+6.783^2+5.25^2+5.95^2+8.83^2 \\ &\quad +11.28^2+10.6^2+9.75^2+12.05^2)-\frac{15\times 16^2}{4}\Big] \\ &= 172.54\end{aligned}$$

and

$$F_F = \frac{29\times 172.54}{30\times 14-172.54} = 20.22$$

According to equation (14), for 15 methods and 30 datasets, F_F is distributed according to the F-distribution

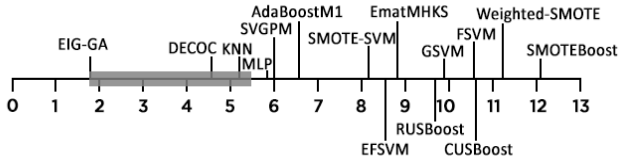

FIGURE 9. Bonferroni–Dunn test for FM.

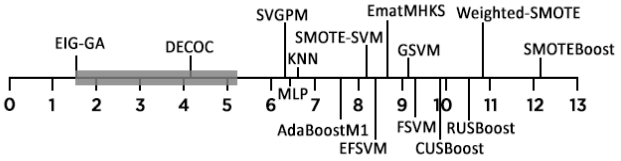

FIGURE 10. Bonferroni–Dunn test for Kappa.

with $(15-1) = 14$ and $(15-1)(30-1) = 406$ df. The p-value computed by $F(14, 406)$ is less than 0.0001[1] which shows that the test rejects the null-hypothesis and therefore, it can be said that the FM results of the compared methods are significantly different on the adopted imbalance datasets.

The Figure 9 demonstrates the Bonferroni Dunn test on FM with $\alpha = 0.05$, the CD of which is 3.603. All other methods but DECOC and KNN performs significantly worst than EIG-GA, have been observed from Figure 9. EIG-GA successfully overcomes the DECOC and SVGPM. SVGPM achieve the 5th best average results.

4) KAPPA

A further objective used to ensure the robustness and credibility of our experiments is the values of Kappa. Therefore, the Kappa is used to prove intuitively and objectively in our experiments to demonstrate consistency of the results and the reliability of the classification. The interpretation of a Kappa outcome from -1 to 1 as mentioned in previous subsection. Table 9 shows the experimental results for Kappa. For evaluation purposes, the results from different datasets (rows) and methods (columns) are shown in Table 9 by using 10-fold cross-validation. The proposed EIG-GA method is better than the other methods in 23 out of 30 datasets and achieves competitive Kappa results, but it loses on 7 datasets. However, their average values and average rankings results are significantly different. The best results are highlighted in BOLD for the comparative algorithms on each dataset. Moreover, interestingly our method have same accuracy results with DECOC on ''tic-tac-toe'' dataset and same results with MLP on ''vowel0'' dataset and also same with EmatMHKS on ''wine'' dataset, which shows the mentioned methods performs best on these datasets whereas on some datasets we have worse performance than the other methods but overall on average Kappa our method outperforms the other methods. From Table 9, our proposed work shows that the interpretation of the kappa value of EIG-GA shows that there is a good agreement with 0.864 or 86.4% as per kappa ranges. From last two rows of Table 9, the average values of Kappa and average ranking results for all datasets are listed respectively.

4.1) STATISTICAL COMPARISON

From Table 9, the proposed EIG-GA method with an average score of 1.55 is noted rank first, the DECOC method ranks the second with the score of 4.05 and SVGPM method ranks the third with the score of 6.367 and so on. From the Friedman test, we have

$$\chi_F^2 = \frac{12 \times 30}{15 \times 16}\Big[(1.55^2 + 6.367^2 + 4.05^2 + 8.233^2 + 9.033^2 + 9.333^2 + 8.33^2 + 7.6^2 + 6.767^2 + 6.45^2 + 8.7^2 + 10.983^2 + 9.867^2 + 10.55^2 + 12.183^2) - \frac{15 \times 16^2}{4}\Big] = 155.7315$$

and

$$F_F = \frac{29 \times 155.7315}{30 \times 14 - 155.7315} = 17.09$$

According to equation (14), for 15 methods and 30 datasets, F_F is distributed according to the F-distribution with $(15-1) = 14$ and $(15-1)(30-1) = 406$ df. The p-value computed by $F(14, 406)$ is less than 0.0001[1] which shows that the test rejects the null hypothesis and therefore, it can be said that the Kappa results of the compared methods are significantly different on the adopted imbalance datasets.

The Figure 10 demonstrates the Bonferroni Dunn test on Kappa with $\alpha = 0.05$, the CD of which is 3.603. All other methods but DECOC, considerably worse than EIG-GA, have been observed from Figure 10. EIG-GA successfully overcomes the DECOC and SVGPM.

5) MATTHEWS CORRELATION COEFFICIENT (MCC)

Another performance measure in this study is MCC, we analyzed the MCC scores achieved by EIG-GA and other methods to the imbalanced datasets. For assessment purposes, the results from different datasets (rows) and methods (columns) are shown in Table 10 by using 10-fold cross-validation. The proposed EIG-GA method is better than to other methods in 22 out of 30 datasets and achieves competitive MCC results, but it loses on 8 datasets. However, their average values and average rankings results are significantly different. The best results are highlighted in BOLD for the comparative algorithms on each dataset. Whereas, we have worse MCC result from baseline paper DECOC on ''tic-tac-toe'' dataset. Moreover, interestingly our method has same accuracy results with SVGPM on ''pageblocks'' which shows the mentioned methods performs best on these datasets whereas on some datasets we have worse than the other methods but overall on average MCC our method outperforms than the other methods. From Table 10, we can see that EIG-GA works best than the other methods. The 0.789 and 0.736 of MCC score achieved by the DECOC and SVGPM which is

TABLE 9. The Kappa result of EIG-GA with 14 other algorithms on 30 imbalance datasets.

Dataset #	EIG-GA	SVGPM	DECOC	SMOTE - SVM	GSVM	FSVM	EFSVM	Ada-Boost M1	KNN	MLP	Emat-MHKS	Weighted - SMOTE	CUS - Boost	RUS - Boost	SMOTE - Boost
1	**0.800**	0.644	0.759	0.712	0.693	0.412	0.574	0.633	0.249	0.555	0.572	0.435	0.626	0.260	0.330
2	0.676	0.420	0.622	0.404	0.346	0.402	0.422	0.485	**0.698**	0.663	0.415	0.462	0.698	0.454	0.358
3	0.653	0.581	0.442	0.408	0.388	0.357	0.413	0.394	0.619	0.315	**0.658**	0.342	0.722	0.247	0.319
4	**0.808**	0.719	0.764	0.655	0.623	0.801	0.799	0.617	0.623	0.776	0.639	0.653	0.348	0.326	0.335
5	**0.954**	0.737	0.851	0.814	0.778	0.821	0.832	0.699	0.831	0.925	0.726	0.736	0.702	0.728	0.488
6	0.886	0.796	0.829	0.657	0.818	0.568	0.56	0.388	0.922	**0.934**	0.852	0.521	0.565	0.642	0.580
7	**0.834**	0.768	0.862	0.743	0.640	0.667	0.642	0.626	0.769	0.799	0.8	0.748	0.811	0.700	0.646
8	**0.858**	0.830	0.808	0.747	0.608	0.410	0.364	0.588	0.667	0.717	0.802	0.786	0.808	0.753	0.732
9	**0.907**	0.868	0.900	0.887	0.86	0.812	0.542	0.534	0.902	0.848	0.907	0.900	0.709	0.811	0.741
10	**0.931**	0.828	0.913	0.911	0.877	0.900	0.912	0.906	0.632	0.748	0.632	0.678	0.594	0.588	0.487
11	**0.918**	0.770	0.872	0.588	0.720	0.642	0.708	0.710	0.808	0.908	0.648	0.724	0.62	0.574	0.443
12	**0.970**	0.922	0.951	0.878	0.942	0.968	0.868	0.953	0.952	0.949	0.82	0.502	0.77	0.621	0.505
13	0.787	0.729	0.755	0.712	0.654	0.666	0.728	0.640	0.548	0.600	0.487	0.386	**0.824**	0.778	0.778
14	0.932	0.788	0.787	0.829	0.844	0.835	0.803	**0.940**	0.934	0.901	0.713	0.698	0.802	0.587	0.622
15	**0.838**	0.686	0.774	0.701	0.505	0.538	0.619	0.528	0.822	0.566	0.403	0.412	0.388	0.323	0.314
16	**0.826**	0.522	0.708	0.445	0.397	0.601	0.432	0.556	0.500	0.645	0.405	0.312	0.787	0.733	0.521
17	**0.862**	0.844	0.806	0.699	0.658	0.608	0.728	0.817	0.707	0.803	0.668	0.449	0.586	0.384	0.365
18	**0.822**	0.625	0.796	0.562	0.594	0.548	0.564	0.508	0.331	0.445	0.588	0.434	0.513	0.542	0.489
19	0.952	0.916	0.924	0.768	0.844	0.749	0.687	**0.972**	0.967	0.958	0.808	0.824	0.935	0.942	0.902
20	**0.954**	0.828	0.838	0.933	0.774	0.864	0.94	0.921	0.745	0.883	0.867	0.820	0.878	0.823	0.714
21	**0.776**	0.624	**0.776**	0.678	0.393	0.385	0.704	0.685	0.681	0.568	0.701	0.769	0.62	0.719	0.628
22	**0.928**	0.854	0.905	0.779	0.835	0.86	0.887	0.848	0.823	**0.928**	0.868	0.812	0.568	0.702	0.610
23	**0.826**	0.801	0.758	0.648	0.700	0.664	0.582	0.778	0.712	0.751	0.588	0.468	0.645	0.81	0.760
24	**0.982**	0.910	0.973	0.688	0.752	0.640	0.738	0.940	0.9702	0.903	0.649	0.678	0.626	0.701	0.770
25	**0.832**	0.829	0.802	0.598	0.453	0.449	0.483	0.554	0.710	0.758	0.38	0.334	0.341	0.324	0.355
26	**0.858**	0.680	0.844	0.622	0.668	0.537	0.648	0.770	0.784	0.772	0.576	0.700	0.512	0.601	0.354
27	**0.978**	0.807	0.967	0.950	0.768	0.866	0.944	0.928	0.931	0.908	0.857	0.745	0.817	0.868	0.778
28	**0.848**	0.784	0.801	0.588	0.672	0.631	0.664	0.488	0.721	0.623	0.442	0.322	0.306	0.318	0.366
29	0.817	0.753	0.757	0.601	0.588	0.538	0.521	0.801	0.792	0.768	**0.828**	0.483	0.509	0.587	0.498
30	**0.914**	0.844	0.838	0.470	0.488	0.664	0.607	0.645	0.557	0.624	0.723	0.601	0.408	0.396	0.388
Avg. Values	**0.864**	0.757	0.813	0.689	0.663	0.647	0.664	0.695	0.730	0.751	0.667	0.591	0.635	0.595	0.539
Avg. Ranks	**1.55**	6.367	4.05	8.233	9.033	9.333	8.333	7.6	6.767	6.45	8.7	10.983	9.867	10.55	12.183

good but less or nearly to 0.853. This shows that EIG-GA performed much better than other methods. From Table 10, In the last two rows, the average values of MCC and average ranking results for all datasets are listed respectively.

5.1) STATISTICAL COMPARISON

From Table 10, the proposed EIG-GA method with an average score of 1.650 is noted rank first, the DECOC method ranks the second with the score of 4.333 and SVGPM method ranks the fifth with the score of 6.650 and so on. From the Friedman test, we have

$$\begin{aligned}\chi_F^2 &= \frac{12\times 30}{15\times 16}\Big[(1.65^2+6.65^2+4.333^2+7.683^2+9.45^2\\ &\quad +9.733^2+7.967^2+7.833^2+6.45^2+5.4^2+9.1^2\\ &\quad +10.533^2+9.867^2+11.217^2+12.133^2)-\frac{15\times 16^2}{4}\Big]\\ &= 162.775\end{aligned}$$

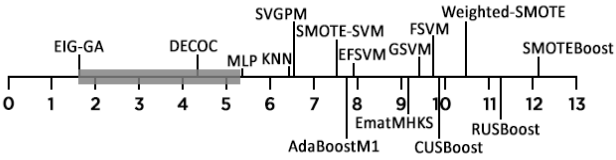

FIGURE 11. Bonferroni–Dunn test for MCC.

and

$$F_F = \frac{29\times 162.775}{30\times 14-162.775} = 18.352$$

According to equation (14), for 15 methods and 30 datasets, F_F is distributed according to the F-distribution with $(15-1)=14$ and $(15-1)(30-1)=406$ df. The $p-value$ computed by $F(14, 406)$ is less than 0.0001[1] which shows that the test rejects the null hypothesis and therefore, it can be said that the Kappa results of the compared methods are significantly different on the adopted imbalance datasets.

The Figure 11 demonstrates the Bonferroni-Dunn test on Kappa with $\alpha = 0.05$, the CD of which is 3.603. All

TABLE 10. The MCC result of EIG-GA with 14 other algorithms on 30 imbalance datasets.

Dataset #	EIG-GA	SVGPM	DECOC	SMOTE - SVM	GSVM	FSVM	EFSVM	Ada-Boost M1	KNN	MLP	Emat-MHKS	Weighted - SMOTE	CUS - Boost	RUS - Boost	SMOTE - Boost
1	**0.803**	0.684	0.763	0.652	0.694	0.433	0.590	0.636	0.250	0.561	0.570	0.443	0.630	0.260	0.333
2	0.691	0.420	0.668	0.411	0.318	0.389	0.394	0.408	0.701	**0.725**	0.425	0.445	0.654	0.464	0.346
3	**0.669**	0.589	0.417	0.408	0.326	0.313	0.413	0.395	0.619	0.319	0.635	0.342	0.639	0.247	0.328
4	**0.819**	0.722	0.771	0.638	0.623	0.729	0.799	0.661	0.603	0.733	0.639	0.643	0.341	0.328	0.325
5	**0.931**	0.674	0.735	0.810	0.766	0.820	0.763	0.704	0.831	0.925	0.722	0.731	0.700	0.678	0.485
6	0.885	0.801	0.834	0.638	0.812	0.540	0.560	0.385	0.970	**0.978**	0.846	0.478	0.537	0.640	0.554
7	0.829	0.910	0.812	**0.920**	0.635	0.564	0.622	0.623	0.728	0.885	0.774	0.740	0.786	0.587	0.634
8	**0.914**	**0.914**	0.848	0.897	0.588	0.414	0.360	0.578	0.624	0.867	0.887	0.844	0.801	0.687	0.712
9	0.933	0.924	0.787	0.904	0.847	0.754	0.540	0.528	0.898	0.944	**0.954**	0.883	0.707	0.808	0.738
10	**0.943**	0.817	0.937	0.922	0.834	0.844	0.901	0.940	0.647	0.737	0.630	0.678	0.594	0.558	0.464
11	**0.904**	0.657	0.870	0.547	0.678	0.632	0.702	0.690	0.788	0.872	0.548	0.707	0.600	0.547	0.438
12	**0.962**	0.918	0.950	0.877	0.938	0.944	0.856	0.920	0.938	0.927	0.810	0.484	0.678	0.611	0.504
13	0.754	0.724	0.751	0.708	0.648	0.628	0.720	0.633	0.545	0.587	0.482	0.355	**0.804**	0.654	0.772
14	**0.930**	0.773	0.783	0.809	0.843	0.833	0.803	0.928	0.930	0.898	0.710	0.644	0.798	0.547	0.612
15	**0.820**	0.657	0.745	0.701	0.487	0.524	0.600	0.518	0.787	0.548	0.400	0.387	0.384	0.309	0.313
16	**0.805**	0.501	0.701	0.434	0.392	0.578	0.430	0.547	0.486	0.637	0.401	0.311	0.783	0.705	0.519
17	**0.844**	0.840	0.784	0.637	0.563	0.486	0.724	0.774	0.705	0.669	0.662	0.433	0.576	0.355	0.312
18	**0.772**	0.518	0.746	0.511	0.485	0.438	0.547	0.484	0.505	0.566	0.565	0.433	0.489	0.535	0.442
19	0.918	0.868	0.920	0.693	0.844	0.711	0.571	0.914	**0.944**	0.852	0.628	0.795	0.910	0.858	0.889
20	**0.942**	0.807	0.659	0.920	0.738	0.748	0.939	0.809	0.549	0.856	0.833	0.821	0.855	0.684	0.696
21	0.758	0.518	**0.774**	0.637	0.368	0.347	0.701	0.608	0.662	0.599	0.597	0.747	0.553	0.700	0.616
22	**0.908**	0.753	0.904	0.764	0.824	0.858	0.882	0.787	0.822	0.914	0.845	0.803	0.563	0.701	0.548
23	**0.788**	0.754	0.738	0.625	0.547	0.654	0.580	0.724	0.689	0.697	0.438	0.460	0.628	0.574	0.660
24	**0.969**	0.902	0.963	0.649	0.752	0.638	0.732	0.933	0.969	0.873	0.634	0.659	0.620	0.637	0.480
25	**0.822**	0.818	0.748	0.588	0.450	0.441	0.481	0.547	0.708	0.751	0.317	0.340	0.338	0.320	0.347
26	**0.802**	0.642	0.788	0.572	0.600	0.518	0.636	0.708	0.733	0.770	0.457	0.639	0.478	0.586	0.354
27	**0.952**	0.646	0.921	0.938	0.744	0.823	0.940	0.887	0.911	0.906	0.807	0.733	0.816	0.834	0.758
28	**0.842**	0.734	0.797	0.546	0.607	0.611	0.658	0.447	0.721	0.583	0.426	0.310	0.313	0.314	0.368
29	0.786	0.749	0.728	0.601	0.547	0.542	0.514	0.800	0.788	0.765	**0.818**	0.482	0.437	0.544	0.428
30	**0.907**	0.833	0.835	0.441	0.405	0.601	0.548	0.628	0.526	0.616	0.722	0.584	0.368	0.354	0.312
Avg. Values	**0.853**	0.736	0.789	0.680	0.630	0.612	0.650	0.671	0.719	0.752	0.639	0.578	0.613	0.554	0.510
Avg. Ranks	**1.65**	6.65	4.333	7.683	9.45	9.733	7.967	7.833	6.45	5.4	9.1	10.533	9.867	11.217	12.133

other methods but DECOC performs significantly worst than EIG-GA, have been observed from Figure 11.

Next, we perform wilcoxon signed rank test [80] between EIG-GA and other state-of-the-art algorithms. This test, which Demsar [83] recommends, is a non-parametric alternative to the paired t-test that ranks the performance difference of two classifiers for each dataset, ignores signs, and compares the ranks for the positive and the negative differences [80]. To conclude all the results related with wilcoxon signed rank test for GM, FM, Kappa and MCC our general analysis is that DECOC method is the best compared to the other methods, but our EIG-GA method outperforms it, with a confidence level higher than 95%

B. TIME COMPLEXITY ANALYSIS

It is interesting to analyze the computational complexity or running time performance of all methods. From Table 12 we report the running-time efficiency of our method with 14 other methods. From Table 12, it is clear that DECOC and SVGPM is very time consuming than EIG-GA, since it needs to train a huge time to build model. Whereas we see that KNN, AdaBoostM1 [69] and MLP [69] have very efficient execution running time to build model, but these methods have worse accuracy metrics i.e., Acc, GM, FM, Kappa, and MCC in previous section. Overall, we conclude that EIG-GA has not so good or bad performance. It is concluded that the overall performance along with computational complexity, our proposed method EIG-GA performs better than other state-of-the-art algorithms.

C. DISCUSSION

In this paper, we are trying to solve class imbalance problem from a perspective of effectiveness (the ability to accurately classify an unknown dataset) and efficiency (the speed of classifying data). Moreover, we used original datasets without any preprocessing techniques. For handling the class imbalance problem, we used 14 other state-of-the-art algorithms and 30 imbalance datasets from UCI and KEEL repositories

TABLE 11. Comparisons between EIG-GA and the 9 other selected algorithms on 5 assessment metrics.

Dataset #	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	SUM
(a) Acc																															
SVGPM	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	30
DECOC	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0	1	29
SMOTE-SVM	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	30
GSVM	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0	1	29
EFSVM	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	30
AdaBoostM1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0	1	1	1	1	1	1	1	1	1	0	1	28
EmatMHKS	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0	1	29
CUSBoost	1	1	1	1	1	1	1	1	1	1	1	1	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	29
RUSBoost	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0	1	1	1	1	1	1	1	1	1	1	1	29
(b) GM																															
SVGPM	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	30
DECOC	1	1	1	1	1	1	1	1	1	0	1	1	1	0	1	1	1	1	1	1	0	1	1	1	1	1	1	1	0	1	26
SMOTE-SVM	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	30
GSVM	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0	1	30
EFSVM	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0	1	1	1	1	1	1	1	1	1	29
AdaBoostM1	1	1	1	1	1	1	1	1	1	0	1	1	1	0	1	1	1	1	0	1	1	1	1	1	1	1	1	1	0	1	27
EmatMHKS	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0	1	1	1	1	1	1	1	0	1	28
CUSBoost	1	1	1	1	1	1	1	1	1	1	1	1	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	29
RUSBoost	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0	1	1	1	1	1	1	1	1	1	1	1	29
(c) FM																															
SVGPM	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	30
DECOC	1	1	1	1	1	1	1	1	0	1	1	1	1	1	1	1	1	1	1	1	0	1	1	1	1	1	1	1	1	1	28
SMOTE-SVM	1	1	1	1	1	1	1	1	1	1	1	1	1	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	29
GSVM	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0	1	29
EFSVM	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0	1	1	1	1	1	1	1	1	1	29
AdaBoostM1	1	1	1	1	1	1	1	1	1	1	1	1	1	0	1	1	1	1	0	1	1	1	1	1	1	1	1	1	0	1	27
EmatMHKS	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0	1	1	1	1	1	1	1	0	1	28
CUSBoost	1	1	1	1	1	1	1	1	1	1	1	1	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	29
RUSBoost	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0	1	1	1	1	1	1	1	1	1	1	1	29
(d) Kappa																															
SVGPM	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	30
DECOC	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	30
SMOTE-SVM	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	30
GSVM	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	30
EFSVM	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	30
AdaBoostM1	1	1	1	1	1	1	1	1	1	1	1	1	1	0	1	1	1	1	0	1	1	1	1	1	1	1	1	1	1	1	28
EmatMHKS	1	1	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0	1	28
CUSBoost	1	0	0	1	1	1	1	1	1	1	1	1	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	27
RUSBoost	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	30
(e) MCC																															
SVGPM	1	1	1	1	1	1	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	29
DECOC	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0	1	0	1	1	1	1	1	1	1	1	1	28
SMOTE-SVM	1	1	1	1	1	1	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	30
GSVM	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	30
EFSVM	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	30
AdaBoostM1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0	1	29
EmatMHKS	1	1	1	1	1	1	1	1	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0	1	28
CUSBoost	1	1	1	1	1	1	1	1	1	1	1	1	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	29
RUSBoost	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	30

to analyze and compare the results of our method with other methods by using evaluation metrics and some statistical test to show the performance under specific conditions.

Table 11 is the comparison between EIG-GA and the 14 other state-of-the-art methods on the five assessment criteria's, where the symbol "1" refers EIG-GA is better than or equivalent in performance with the method to be compared, whereas the symbol "0" indicates that EIG-GA is worse than the compared method, e.g. the symbol "1" in the first row in Table 11 indicates that the Acc result of EIG-GA on the 1st dataset is better than of SVGPM, whereas the symbol "0" indicates the worse performance. From Table 11 we also notice that, there is no constant winner on the 30 datasets that can always beat all the other methods. However, the overall performance of EIG-GA is higher than the other methods.

TABLE 12. The running time of EIG-GA with 14 other algorithms on 30 datasets (in seconds).

Dataset #	EIG-GA	SVGPM	DECOC	SMOTE-SVM	GSVM	FSVM	EFSVM	AdaBoost-M1	KNN	MLP	EmatMHKS	Weighted-SMOTE	CUS-Boost	RUS-Boost	SMOTE-Boost
1	508.87	3428.23	1091.56	294.54	965.37	697.34	870.78	40.45	85.34	90.45	950.27	1052.34	1488.9	1866.87	1443.33
2	276.18	868.77	774.12	250.48	558.65	474.44	450.5	90.12	50.48	52.14	1098.47	978.47	687.45	634.87	676.43
3	28.34	300.41	120.52	39.54	155.48	354.58	452.89	20.18	33.48	80.5	908.8	512.39	233.56	265.43	355.89
4	56.78	404.1	352.58	300.88	88.48	458.78	787.1	44.58	34.18	66.43	434.1	510.24	455.44	378.43	445.87
5	120.89	1008.44	1096.84	838.65	458.98	322.18	785.48	98.78	28.16	89.2	1160.48	845.68	524.12	780.45	650.54
6	428.35	2258.26	2387.7	1098.47	1558.6	878.44	1097.64	98.4	150.84	78.4	1878.52	1144.6	1850.15	1502.8	1663.78
7	584.89	2765.42	2546.7	2008.4	1487.6	1340.5	989.78	94.45	255.18	86.23	1868.49	1975.48	1001.71	1008.83	1012.33
8	129.5	679.97	784.5	344.48	548.2	744.12	794.28	80.6	78.18	53.15	885.1	746.01	292.32	258.23	255.08
9	24.19	144.18	30.89	180.48	39.48	58.47	66.47	18.44	10.44	15.48	88.78	108.47	64.28	68.83	88.56
10	45.58	388.45	210.35	105.45	218.45	389.47	585.47	30.68	47.87	45.47	484.74	131.11	118.52	185.85	134.44
11	78.49	558.21	868.47	440.18	658.9	700.78	808.2	58.1	43.2	128.47	1098.78	100.45	438	1200.32	466.34
12	69.47	528.47	843.15	412.87	616.48	585.48	898.48	88.78	61.48	110.48	945.44	130.5	180.65	250.85	180.56
13	54.87	60648	787.45	129.48	543.28	474.58	554.22	30.48	10.78	148.47	348.78	100.88	421.88	390.78	329.44
14	308.48	1287.47	1524	1080.45	1434.18	1233.5	1121.47	158.48	112.16	97.48	1180.54	1628.4	1160.25	925.52	1011.67
15	722.1	3028.45	2985.48	1419.48	1916.45	1956.66	1324.26	300.24	128.48	389.48	2001.58	1488.8	1852.88	1766.58	1433.65
16	24.58	321.27	118.47	121.47	158.88	30.48	48.86	32.15	18.45	58.12	80.15	110.78	183.18	190.54	176.44
17	18.87	180.45	103.2	58.56	45.42	22.47	41.58	20.28	15.48	63.38	99.18	102.78	107.82	112.78	59.45
18	198.78	482.47	363.54	151.47	144.85	224.42	389.98	78.45	88.47	118.78	587.47	412.22	219.45	208.8	230.45
19	382.52	1966.4	1098.1	858.7	787.64	344.5	411.25	180.47	288.73	301.54	1045.51	747.14	342.58	559.44	1676.33
20	128.45	858.46	1029.45	667.4	586.47	258.4	349.9	40.44	58.88	126.54	989.25	534.11	202.44	131.47	121.89
21	230.53	1422.5	1788.44	886.4	656.7	288.3	625.33	97.48	255.44	178.58	1118.42	578.5	244.47	248.8	216.8
22	158.48	1588.47	2033.1	988.45	688.87	283.3	654.8	100.58	258.48	181.47	1007.5	528.2	302.48	255.49	424.4
23	202.47	504.4	765.44	189.8	487.6	445.5	300.47	30.55	80.7	238.69	382.1	151.4	387.48	400.42	315.8
24	30.24	300.48	131.1	98.45	78.46	37.45	31.88	38.48	50.7	66.49	50.45	80.84	45.58	98.45	104.65
25	62.48	207.48	66.78	150.41	120.28	78.45	58.2	62.17	42.17	200.14	144.8	103.55	68.42	51.48	52.12
26	288.4	2040.68	870.48	340.58	478.41	398.44	667.4	120.4	60.48	300.58	908.78	1008.48	341.45	289.47	455.31
27	22.48	300.5	125.89	87.45	34.88	55.45	28.85	110.5	5.28	89.4	108.4	77.6	120.4	150.6	164.87
28	42.57	150.45	245.58	98.68	68.78	224.4	110.8	5.9	18.87	87.48	255.7	142.2	72.45	88.6	76.66
29	45.58	203.45	89.77	187.88	167.45	145.44	382.45	8.66	65.45	55.48	108.45	48.58	67.88	50.48	48.54
30	78.58	543.87	404.12	80.88	409.45	188.45	338.87	10.89	55.45	102.45	1908.41	644.18	115.48	88.45	107.63

IX. CONCLUSIONS

Under-sampling and over-sampling methods are often used in the skewed repository to deal with the imbalanced dataset problem. However, both methods can cause either the loss of important data or adding insignificant classification data that can affect the prediction accuracy for minority examples in the imbalanced dataset. This paper proposes the new classifier EIG-GA (which uses entropy and information gain as a fitness function), as a solution to the imbalanced classification task and has obtained very good results on unseen data for many imbalance datasets. The results show that the proposed EIG-GA approach performs better than the other methods in terms of effectiveness and efficiency with regard to evaluation metrics i.e., Acc, GM, FM, kappa, and MCC. In future, we suggest that the EIG-GA should be enhanced on the basis of one of the key elements that is an improvement in the fitness function.

REFERENCES

[1] A. Ghazikhani, H. S. Yazdi, and R. Monsefi, "Class imbalance handling using wrapper-based random oversampling," in *Proc. 20th Iranian Conf. Elect. Eng.*, May 2012, pp. 611–616.

[2] V. L. Cao, N.-A. Le-Khac, M. Nicolau, M. ONeill, and J. McDermott, "Improving fitness functions in genetic programming for classification on unbalanced credit card datasets," 2017, *arXiv preprint arXiv:1704.03522*. [Online]. Available: https://arxiv.org/abs/1704.03522

[3] E. Ramentol, N. Verbiest, R. Bello, Y. Caballero, C. Cornelis, and F. Herrera, "SMOTE-FRST: A new resampling method using fuzzy rough set theory," in *Proc. WSPC*, 2012, pp. 800–805.

[4] S.-J. Yen and Y.-S. Lee, "Cluster-based under-sampling approaches for imbalanced data distributions," *Expert Syst. Appl.*, vol. 36, no. 3, pp. 5718–5727, Apr. 2009.

[5] M. M. Rahman and D. N. Davis, "Addressing the class imbalance problem in medical datasets," *Int. J. Mach. Learn. Comput.*, vol. 3, no. 2, pp. 224–228, Apr. 2011.

[6] N. V. Chawla, K. W. Bowyer, L. O. Hall, and W. P. Kegelmeyer, "SMOTE: Synthetic minority over-sampling technique," *J. Artif. Intell. Res.*, vol. 16, pp. 321–357, Jun. 2002.

[7] L. Wei, Z. Dongmei, and L. Yang, "A novel over-sampling method based on EDAs for learning from imbalanced data sets," *J. Converg. Inf. Technol.*, vol. 6, no. 11, pp. 237–247, 2011.

[8] J. DU and L.-L. Jiang, "The research of KNN and SVM classification performance on two kinds of unbalanced data set," in *Proc. 2nd Conf. Comput. Sci. Electron. Eng. (ICCSEE)*, Mar. 2013.

[9] A. Bouzerdoum and G. S. N. Phung, "Learning Patterns classification tasks with imbalanced data sets," *Pattern Recognition*. Vukovar, Croatile: In-Tech, 2009, ch. 10, pp. 193–208.

[10] V.García, J. S. Sánchez, and R. A. Mollineda, "On the effectiveness of preprocessing methods when dealing with different levels of class imbalance," *Knowl.-Based Syst.*, vol. 25, no. 1, pp. 13–21, Feb. 2012.

[11] R. Barandela, R. M. Valdovinos, J. S. Sánchez, and F. J. Ferri, "The imbalanced training sample problem: Under or over sampling?" in *Proc. Joint IAPR Int. Workshops Struct., Syntactic Stat. Pattern Recognit. (SSPR/SPR)*, 2004, pp. 806–814.

[12] M. Zorkeflee, A. M. Din, and K. R. Ku-Mahamud, "Fuzzy and SMOTE resampling technique for imbalanced data set," in *Proc. 5th Int. Conf. Comput. Inform.*, Istanbul, Turkey, Aug. 2015, pp. 638–643.

[13] S.-J. Yen and Y.-S. Lee, "Cluster-based under-sampling approaches for imbalanced data distributions," *Expert Syst. Appl.*, vol. 36, no. 1, pp. 5718–5727, Apr. 2009.

[14] B. Yuan and X. Ma, "Sampling + reweighting: Boosting the performance of AdaBoost on imbalanced datasets," in *Proc. Int. Joint Conf Neural Netw. (IJCNN)*, Jun. 2012, pp. 2680–2685.

[15] A. C. Bahnsenand, D. Aouada and B. Ottersten, "A novel cost-sensitive framework for customer churn predictive modeling," in *Proc. Decis. Anal.*, Dec. 2015 pp. 1–15.

[16] R. Kothandan, "Handling class imbalance problem in miRNA dataset associated with cancer," *Bioinformation*, vol. 11, no. 1, pp. 6–10, Jan. 2015.

[17] C. X. Ling and V. S. Sheng, "Cost-sensitive learning and the class imbalance problem," in *Encyclopedia of Machine Learning*. Biomedical Informatics Publishing Group, 2008.

[18] R. Cruz, K. Fernandes, J. S. Cardoso, and J. F. P. Costa, "Tackling class imbalance with ranking, in *Proc. Int. Joint Conf. Neural Netw. (IJCNN)*, Jul. 2016, pp. 2182–2187.

[19] T. Kim and H. Ahn, "A hybrid under-sampling approach for better bankruptcy prediction," *J. Intell. Inform. Syst.*, vol. 21, no. 2, pp. 173–190, Jun. 2015.

[20] M. Galar, A. Fernandez, E. Barrenechea, H. Bustince, and F. Herrera, "A review on ensembles for the class imbalance problem: Bagging-, boosting-, and hybrid-based approaches," *IEEE Trans. Syst., Man, Cybern. C, Appl. Rev.*, vol. 42, no. 4, pp. 463–484, Jul. 2012.

[21] C. Seiffer, T. M. Khoshgoftaar, J. Van Hulse, and A. Napolitano, "Building useful models from imbalanced data with sampling and boosting," in *Proc. 21st Int. FLAIRS Conf.*, May 2008, pp. 306–311.

[22] C. Seiffert, T. M. Khoshgoftaar, J. Van Hulse, and A. Napolitano, "RUSBoost: A hybrid approach to alleviating class imbalance," *IEEE Trans. Syst., Man, Cybern. A, Syst. Humans*, vol. 40, no. 1, pp. 185–197, Jan. 2010.

[23] M. Govindaraj and S. Lavanya, "A Combined Boosting And Sampling Approach For Imbalanced Data Classification," *Int. J. Adv. Res. Data Mining Cloud Comput.*, vol. 1, no. 1, pp. 44–50, Jul. 2013.

[24] J. Błaszczyński and J. Stefanowski, "Neighbourhood sampling in bagging for imbalanced data," *Neurocomputing*, vol. 150, pp. 529–542, Feb. 2015.

[25] M. Galar, A. Fernandez, E. Barrenechea, H. Bustince, and F. Herrera, "A review on ensembles for the class imbalance problem: Bagging-, boosting-, and hybrid-based approaches," *IEEE Trans. Syst., Man, Cybern. C, Appl. Rev.*, vol. 42, no. 4, pp. 463–484, Jul. 2014.

[26] B. Krawczyk, M. Woźniak, and G. Schaefer, "Cost-sensitive decision tree ensembles for effective imbalanced classification," *Appl. Soft Comput.*, vol. 14, pp. 554–562, Jan. 2014. doi: 10.1016/j.asoc.2013.08.014.

[27] H. Yu, J. Ni, and J. Zhao, "ACOSampling: An ant colony optimization-based undersampling method for classifying imbalanced DNA microarray data," *Neurocomputing*, vol. 101, pp. 309–318, Feb. 2013.

[28] Y. Mi, "Imbalanced classification based on active learning SMOTE," *Res. J. Appl. Sci. Eng. Technol.*, vol. 5, no. 3, pp. 944–949, Jan. 2013.

[29] L. Zhou, "Performance of corporate bankruptcy prediction models on imbalanced dataset: The effect of sampling methods," *Knowl.-Based Syst.*, vol. 41, pp. 16–25, Mar. 2013.

[30] A. Kirshners, S. Parshutin, and H. Gorskis, "Entropy-based classifier enhancement to handle imbalanced class problem," *Procedia Comput. Sci.*, vol. 104, no. 3, pp. 586–591, 2017.

[31] E. Burnaev, P. Erofeev, and A. Papanov, "Influence of resampling on accuracy of imbalanced classification," *Proc. SPIE*, vol. 9875, Dec. 2015, Art. no. 987521.

[32] H. Patel and G. S. Thakur, "Classification of imbalanced data using a modified fuzzy-neighbor weighted approach," *Int. J. Intell. Eng. Syst.*, vol. 10, no. 1, pp. 56–64, Nov. 2016.

[33] M. S. M. Pozi, M. N. Sulaiman, N. Mustapha, and T. Perumal, "A new classification model for a class imbalanced data set using genetic programming and support vector machines: Case study for wilt disease classification," *Remote Sens. Lett.*, vol. 6, no. 7, pp. 568–577, Jun. 2015.

[34] L. Diao, C. Yang, and H. Wang, "Training SVM email classifiers using very large imbalanced dataset," *J. Exp. Theor. Artif. Intell.*, vol. 24, no. 2, pp. 193–210, Sep. 2011.

[35] S. Winkler, M. Affenzeller, and S. Wagner, "Advanced genetic programming based machine learning," *J. Math. Model. Algorithms*, vol. 6, no. 3, pp. 455–480, Sep. 2007.

[36] Y. Chen, K. Wu, X. Chen, C. Tang, and Q. Zhu, "An entropy-based uncertainty measurement approach in neighborhood systems," *Inf. Sci.*, vol. 279, pp. 239–250, Sep. 2014.

[37] M. Hanmandlu,"A new entropy function and a classifier for thermal face recognition," *Eng. Appl. Artif. Intell.*, vol. 36, pp. 269–286, Nov. 2014.

[38] D. Long, V. P. Singh, "An entropy-based multispectral image classification algorithm," *IEEE Trans. Geosci. Remote Sens.*, vol. 51, no. 12, pp. 5225–5238, Dec. 2013.

[39] C. E. Shannon, "A mathematical theory of communication," *Bell Syst. Tech. J.*, vol. 27, no. 3, pp. 379–423, Jul. 1948.

[40] V. Ganganwar, "An overview of classification algorithms for imbalanced datasets," *Int. J. Emerg. Technol. Adv. Eng.*, vol. 2, no. 4, pp. 42–47, Apr. 2012.

[41] B. Krawczyk, M. Woźniak, and G. Schaefer, Cost-sensitive decision tree ensembles for effective imbalanced classification," *Appl. Soft Comput.*, vol. 14, pp. 554–562, Jan. 2014. doi: 10.1016/j.asoc.2013.08.014.

[42] A. Palacios, K. Trawińsk, O. Cordón, and L. Sánchez, "Cost-Sensitive Learning of Fuzzy Rules for Imbalanced Classification Problems Using FURIA," *Int. J. Uncertainty Fuzziness Knowl. Based Syst.*, vol. 22, no. 05, pp. 643–675, Oct. 2014. doi: 10.1142/S0218488514500330.
[43] W. Lu, Z. Li, and J. Chu, "Adaptive ensemble undersampling-boost: A novel learning framework for imbalanced data," *J. Syst. Softw.* vol. 132, pp. 272–282, Oct. 2017.
[44] N. Garcia-Pedrajas, J. Pérez-Rodríguez, and A. de Haro-García, "OligoIs: Scalable instance selection for class-imbalanced data sets," *IEEE Trans. Cybern.*, vol. 43, no. 1, pp. 332–346, Feb. 2013.
[45] S. Maldonado and J. López, "Imbalanced data classification using second-order cone programming support vector machines," *Pattern Recognit.*, vol. 47, no. 5, pp. 2070–2079, May 2014.
[46] D. Long and V. Singh, "An entropy-based multispectral image classification algorithm," *IEEE Trans. Geosci. Remote Sens.*, vol. 51, no. 12, pp. 5225–5238, Dec. 2013.
[47] W. Liu, S. Chawla, Class confidence weighted kNN algorithms for imbalanced data sets," in *Proc. 15th Pacific-Asia Conf. Adv. Knowl. Discovery Data Mining*. Berlin, Germany: Springer-Verlag, May 2011, pp. 345–356.
[48] W. Liu, S. Chawla, D.A. Cieslak, N. V. Chawla, "A robust decision tree algorithm for imbalanced data sets," in *Proc. SIAM Int. Conf. Data Mining*, Apr. 2010, pp. 766–777.
[49] Y. Li, X. Zhang, Improving K nearest neighbor with exemplar generalization for imbalanced classification," in *Proc. 15th Pacific-Asia Conf. Adv. Knowl. Discovery Data Mining (PAKDD)*, J. Z. Huang, L. Cao, J. Srivastava (Eds.). Berlin, Germany: Springer, 2011, pp. 321–332. doi: 10.1007/978-3-642-20847-8_27.
[50] I. Albisua, O. Arbelaitz, I. Gurrutxaga, A. Lasarguren, J. Muguerza, J. Pérez, "The quest for the optimal class distribution: An approach for enhancing the effectiveness of learning via resampling methods for imbalanced data sets," *Prog. Artif. Intell.*, vol. 2, no. 1, pp. 45–63, Mar. 2013. doi: 10.1007/s13748-012-0034-6.
[51] G. Menardi and N. Torelli, "Training and assessing classification rules with imbalanced data," *Data Mining Knowl. Discovery*, vol. 28, no. 1, pp. 92–122, Jan. 2014. doi: 10.1007/s10618-012-0295-5.
[52] V. López, A. Fernández, F. Herrera, "On the importance of the validation technique for classification with imbalanced datasets: Addressing covariate shift when data is skewed," *Inf. Sci.*, vol. 257, pp. 1–13, Feb. 2014.
[53] V. López, I. Triguero, C. J. Carmona, S. García, F. Herrera, "Addressing imbalanced classification with instance generation techniques: IPADE-ID," *Neurocomputing*, vol. 126, pp. 15–28, Feb. 2014.
[54] M. S. M. Pozi, M. N. Syafiq, M. N. Sulaiman, N. Mustapha, and T. Perumal, "A new classification model for a class imbalanced data set using genetic programming and support vector machines: Case study for wilt disease classification," *Remote Sens. Lett.*, vol. 6, no. 7, pp. 568–577, Jun. 2015.
[55] J. R. Koza, *Genetic Programming II: Automatic Discovery of Reusable Programs*. Cambridge, MA, USA: MIT Press, 1992.
[56] M. N. Haque, N. Noman, R. Berretta, and P. Moscato, "Heterogeneous ensemble combination search using genetic algorithm for class imbalanced data classification," *PLoS One*, vol. 11, no. 1, Jan. 2016, Art. no. e0146116.
[57] B. A. Johnson, R. Tateishi, and N. T. Hoan, "A hybrid pansharpening approach and multiscale object-based image analysis for mapping diseased pine and oak trees," *Int. J. Remote Sens.*, vol. 34, no. 20, pp. 6969–6982, Jun. 2013.
[58] J. Bi and C. Zhang, "An empirical comparison on state-of-the-art multi-class imbalance learning algorithms and a new diversified ensemble learning scheme," *Knowl.-Based Syst.*, vol. 158, pp. 81–93, Oct. 2018.
[59] S. M. A. Elrahman and A. Abraham, "A review of class imbalance problem," *J. Netw. Innov. Comput.*, vol. 1, no. 8, pp. 332–340, 2013.
[60] M. R. Sousa, J. Gama, and E. Brandão, "A new dynamic modeling framework for credit risk assessment," *Expert Syst. Appl.*, vol. 45, pp. 341–351, Mar. 2016.
[61] J. Meseguer, V. Puig, and T. Escobet, "Fault diagnosis using a timed discrete-event approach based on interval observers: Application to sewer networks," *IEEE Trans. Syst., Man, Cybern. A, Syst. Humans*, vol. 40, no. 5, pp. 900–916, Sep. 2010.
[62] S. Wang, L. L. Minku, and X. Yao, "Online class imbalance learning and its applications in fault detection," *Int. J. Comput. Intell. Appl.*, vol. 12, no. 4, Dec. 2013, Art. no. 1340001.
[63] Y. Sun, K. Tang, L. L. Minku, S. Wang, and X. Yao, "Online ensemble learning of data streams with gradually evolved classes," *IEEE Trans. Knowl. Data Eng.*, vol. 28, no. 6, pp. 1532–1545, Jun. 2016.
[64] C. Zhu and Z. Wang, "Entropy-based matrix learning machine for imbalanced data sets," *Pattern Recognit. Lett.*, vol. 88, pp. 72–80, Mar. 2017.
[65] S. Maldonado and J. López "Dealing with high-dimensional class-imbalanced datasets: Embedded feature selection for SVM classification," *Appl. Soft Comput.* vol. 67, pp. 94–105 Jun. 2018.
[66] F. J. Castellanos and J. J. Valero-Mas, J. Calvo-Zaragoza, and J. R. Rico-Juan, "Oversampling imbalanced data in the string space," *Pattern Recognit. Lett.*, vol. 103, pp. 32–38, Feb. 2018.
[67] G. Haixiang, L. Yijing, J. Shang, G. Mingyun, H. Yuanyue, and G. Bing, "Learning from class-imbalanced data: Review of methods and applications," *Expert Syst. Appl.* vol. 73, pp. 220–239, May 2017.
[68] Q. Fan, Z. Wang, D. Li, D. Gao, and H. Zha, "Entropy-based fuzzy support vector machine for imbalanced datasets," *Knowl.-Based Syst.*, vol. 115, pp. 87–99, Jan. 2017.
[69] K. Randhawa, C. K. Loo, M. Seera, C. P. Lim, and A. K. Nandi, "Credit card fraud detection using AdaBoost and majority voting," *IEEE Access*, vol. 6, pp. 14277–14284, 2018.
[70] H. Hartono, O. S. Sitompul, T. Tulus, and E. B. Nababan, "Biased support vector machine and weighted-smote in handling class imbalance problem," *Int. J. Adv. Intell. Inform.*, vol. 4, no. 1, pp. 21–27, Mar. 2018.
[71] Y. R. Alvarezr, Y. C. Mota, Y. F. Cabrera, I. G. Hilarión, Y. F. Hernández, and M. F. Dominguez, "Similar prototype methods for class imbalanced data classification," in *Uncertainty Management with Fuzzy and Rough Sets*. Cham, Switzerland: Springer, Jan. 2019, pp. 193–209.
[72] J. Bi and C. Zhang, "An empirical comparison on state-of-the-art multi-class imbalance learning algorithms and a new diversified ensemble learning scheme," *Knowl.-Based Syst.*, vol. 158, pp. 81–93, Oct. 2018.
[73] A. Cano, A. Zafra, and S. Ventura, "Weighted data gravitation classification for standard and imbalanced data," *IEEE Trans. Cybern.*, vol. 43, no. 6, pp. 1672–1687, Dec. 2013.
[74] C. Márquez-Vera, A. Cano, C. Romero, and S. Ventura, "Predicting student failure at school using genetic programming and different data mining approaches with high dimensional and imbalanced data," *Appl. Intell.*, vol. 38, no. 3, pp. 315–330, Apr. 2013.
[75] B. Krawczyk, "Learning from imbalanced data: Open challenges and future directions," *Prog. Artif. Intell.*, vol. 5, no. 5, pp. 221–232, Nov. 2016.
[76] L. Gonzalez-Abril, H. Nuñez, C. Angulo, and F. Velasco, "GSVM: An SVM for handling imbalanced accuracy between classes inbi-classification problems," *Appl. Soft Comput.*, vol. 17, pp. 23–31, Apr. 2014. doi: 10.1016/j.asoc.2013.12.013.
[77] J. Alcalá-Fdez *et al.*, "Keel data-mining software tool: Dataset repository, integration of algorithmsand experimental analysis framework," *J. Mult.-Valued Logic Soft Comput.*, vol. 17, pp. 255–287, Jun. 2011.
[78] M. Lichman. (2013). UCI Machine Learning Repository. School of Information and Computer Sciences, University of California. Irvine. [Online]. Available: http://archive.ics.uci.edu/ml
[79] B. Krawczyk, A. Cano, and M. Wozniak, "Selecting local ensembles for multi-class imbalanced data classification," in *Proc. Int. Joint Conf. Neural Netw. (IJCNN)*, Jul. 2018, pp. 1–8.
[80] F. Wilcoxon, "Individual comparisons by ranking methods," *Biometrics Bulletin*, vol. 1, no. 6, pp. 80–83, Dec. 1945.
[81] S. García and F. Herrera, "An extension on 'statistical comparisons of classifiers over multiple data sets' for all pairwise comparisons," *J. Mach. Learn. Res.*, vol. 9, no. 12, p. 2677–2694, Dec. 2008.
[82] S.r García, D. Molina, M. Molina, and F. Herrera, "A study on the use of non-parametric tests for analyzing the evolutionary algorithms' behaviour: A case study on the CEC'2005 Special Session on Real Parameter Optimization," *J. Heurist.*, vol. 15, no. 6, pp. 617–644, Dec. 2009.
[83] J. Demšar, "Statistical comparisons of classifiers over multiple data sets," *J. Mach. Learn. Res.*, vol. 7, no. 1, pp. 1–30, Jan. 2006.
[84] S. García, A. Fernández, J. Luengo, and F. Herrera, "Advanced nonparametric tests for multiple comparisons in the design of experiments in computational intelligence and data mining: Experimental analysis of power," *Inf. Sci.*, vol. 180, no. 10, pp. 2044–2064, May 2010.
[85] O. J. Dunn, "Multiple comparisons among means," *J. Amer. Stat. Assoc.*, vol. 56, no. 293, pp. 52–64, Apr. 2012.

MIRZA AMAAD UL HAQ TAHIR received the M.Sc. degree in information technology from Arid Agriculture University, Rawalpindi, Pakistan, in 2014. He is currently pursuing the M.S. degree with the Department of Computer Science, COMSATS University Islamabad. He is also a Software Professional with over a decade of experience across industry. His research interests include machine learning, data mining, and big data.

SOHAIL ASGHAR graduated (Hons.) in computer science from the University of Wales, U.K., in 1994, and received the Ph.D. degree from Faculty of Information Technology, Monash University, Melbourne, Australia, in 2006. In 2011, he joined as the Director of the University Institute of Information Technology, PMAS-Arid Agriculture University, Rawalpindi. He is currently a Professor and the Chairman of computer science with COMSATS University Islamabad. He has taught and researched in data mining (including structural learning, classification, and privacy preservation in data mining, text and web mining), big data analytics, data science and information technology areas. He has published extensively (More than 150 publications) in international journals and conference proceedings. He has also consulted widely on information Technology matters, especially in the framework of data mining and data science. In 2004, he acquired the Australian Postgraduate Award for Industry. He is a member of the Australian Computer Society (ACS), and the IEEE and also Higher Education Commission Approved Supervisor. He is in the Editorial Team of well reputed Scientific Journals. He has also served as a Program Committee member of numerous International Conferences and regularly speaks at international conferences, seminars, and workshops.

AWAIS MANZOOR received the master's degree in computer science from COMSATS University, Islamabad, Pakistan, in 2017. He is currently a Lecturer with the Department of Computer Science, The University of Lahore, Pakistan, where he teaches data mining, image processing, and algorithms analysis at undergraduate and graduate level. He is currently works toward autonomous intelligence of industrial robots. His research interests include image-based modelling, computational intelligence, pattern recognition, and machine intelligence.

MUHAMMAD ASIM NOOR received the Ph.D. degree in computer science from Johannes Kepler University Linz, Austria, in 2008. He is currently an Assistant Professor with the Department of Computer Science, COMSATS University Islamabad, Islamabad, Pakistan, where he teaches software engineering, requirement engineering, and software project management at undergraduate and graduate level.