A Cost-Sensitive Centroid-based Differential Evolution Classification Algorithm applied to Cancer Data Sets

Jamil Al-Sawwa
Department of Computer Science
North Dakota State University
Fargo, ND, USA
Email: jamil.alsawwa@ndsu.edu

Simone A. Ludwig
Department of Computer Science
North Dakota State University
Fargo, ND, USA
Email: simone.ludwig@ndsu.edu

***Abstract*—Nowadays, the collected or generated data for some real-life applications such as in the Medical domain and Intrusion Detection, are typically imbalanced. Imbalanced data sets consist of date where one class-label (minority) includes significantly fewer instances compared to other class labels. The misclassification of the minority class-label could be costly in some circumstances. Therefore, the extraction of valuable information from this kind of data poses a challenge to the scientific community. During the last decades, the researchers proposed a centroid-based classification algorithm using differential evolution (CDE) to solve data classification. However, CDE shows an inefficient performance especially when applied to imbalanced binary data sets. In this paper, we propose a cost-sensitive version of CDE based on a new objective function in order to overcome this drawback. We are using four cancer data sets that are imbalanced namely Breast, Lung, Uterus, and Stomach. Furthermore, we analyzed and investigated the performance of our proposed version of CDE for predicting the survivability of cancer patients compared to the performance of the current variants of CDE. Moreover, we compared the performance of our proposed version of CDE with the performance of five cost-sensitive machine learning algorithms. The experimental results demonstrate that our proposed version of CDE improves the performance of CDE when applied to imbalanced binary data sets. Furthermore, the performance of our proposed CDE algorithm outperformed the performance of the current variants of CDE on all data sets in terms of Area Under Curve and G-mean.**

***Keywords*—differential evolution, classification, survivability prediction, SEER data set, breast cancer, lung cancer, uterus cancer, stomach cancer**

I. Introduction

Data classification is a supervised learning task in the data mining field, which is a process of analyzing historical data using an algorithm to discover hidden patterns. The outcome is a model that is used to analyze the inputs in a testing data set in order to classify each example. In the last decades, various classification algorithms have been successfully used to tackle data classification problems in many fields such as medicine and engineering.

The skewed distribution of class labels is a significant problem that typically occurs in real-life applications such as medical diagnosis and intrusion detection; where one class-label (minority) includes significantly fewer instances compared to other class labels. Moreover, in some situations, the cost of misclassifying a minority class label could be much higher than the other class labels. For example, in medical diagnosis, an undiscovered cancer diagnosis has a higher penalty than discovering a normal situation as cancer disease. Therefore, the skewed distribution of class labels should be addressed by preprocessing data using sampling techniques before applying a classification algorithm or modifying the classification algorithm [1].

Differential Evolution (DE) is a simple and robust stochastic search method which belongs to the Evolutionary Algorithm (EA) family of optimization algorithms [2]. Storn and Price proposed EA for solving global optimization problems in continuous search spaces [3]. The idea of DE is mainly inspired by the Genetic algorithm (GA) with the difference in the way new offsprings are generated. Since its beginnings, DE has been applied to various real-world optimization problems such as single-objective and multi-objective problems and showed itself as an efficient and powerful technique for solving these types of optimization problems [4]. Furthermore, the researchers and contributors have successfully used DE to tackle data mining tasks such as clustering and classification [5], [6], [7].

During the last decade, the researchers proposed variants of CDE based on different fitness functions [5], [6] to handle data classification. However, these variants show an inefficient performance to cope with imbalanced binary data sets. In this paper, we propose a cost-sensitive version of CDE to address the previous drawback. We introduce a new objective function to handle the imbalanced binary data set by minimizing the misclassification cost instead of the misclassification error. To the best of our knowledge, this is the first work that implements the cost-sensitive version of CDE and applies it to data sets in the medical area in particular on cancer data sets. The aim is to study and investigate the capability of the cost-sensitive-based CDE on handling real-world, imbalanced, and binary data sets.

The remaining sections in this paper are as follows: Section II provides an overview of existing research in solving classification problem using nature-inspired optimization methods. Section III describes the differential evolution algorithm and illustrates the centroid-based differential evolution classification algorithm. In Section IV, we describe our proposed approach. Section V explains the data set and preprocessing. In Section VI, we present the experiments as well as the results. Finally, Section VII presents our conclusions and future work.

II. Related Work

In this section, we will present research work that is related to solve the data classification task using nature-inspired optimization algorithms.

De Falco et al. [5] proposed a new centroid-based differential evolution classification algorithm method to classify hand-segmented image parts automatically. In this work, the optimal centroids of all target labels are found by minimizing the sum of the Euclidean distances between the data instances in a training data set and the centroid of the actual target label that the data instance belongs to. Then, the optimal centroids are used to classify the instances in a testing data set according to the Euclidean distance. Comparing the approach with ten machine learning algorithms, the DE predictor outperformed nine of these algorithms in terms of misclassification rate applied to the hand-segmented image parts data set.

Another work found in [6], Luukka and Lampinen proposed an approach based on the DE method and principal component analysis (PCA) for solving the classification task. The authors analyzed and studied the performance of the DE classifier using five clinical data sets that are related to Heart disease after applying PCA to these data sets. The experimental results revealed that the performance of the DE classifier is improved by applying PCA to the data sets first. Moreover, the computational time of the DE classifier is reduced because the dimensionality of the data is decreased using PCA. Other works related to solving data classification using DE can be found in [8], [9].

De Falco et al. [10] introduced a new classification algorithm based on the Particle Swarm Optimization method (PSO). The main goal is the same as in the previous works, which is finding the optimal centroids for all labels in a training data set. Three versions of this algorithm were proposed based on different fitness functions. Using thirteen benchmark data sets taken from the UCI Machine Learning Repository [11], the third version achieved better performance than the first and second version, and outperformed five out of the nine comparison machine learning algorithms according to the averaged classification error rate.

In [12], a new classification algorithm based on artificial bee colony (ABC) was proposed to find the optimal centroids by minimizing the sum of all Euclidean distances between the current centroid position of a class label and the data instances that it belongs to. Compared to the second version of the PSO-based classifier [10] and nine classification algorithms, the experimental results revealed that the ABC-based classifier achieved the best performance in 6 out of 13 data sets in terms of the classification error rate. Moreover, the ABC-based classifier ranked second according to the averaged classification error rate over all data sets.

Firefly Algorithm (FA) is a stochastic search method that belongs to the swarm intelligence family. FA imitates the flashing lights of fireflies, which were introduced by Yang [13]. In [14], the authors proposed the classification algorithm based on FA using the same fitness function as in [5], [10], [12]. Using thirteen data sets, the authors studied and investigated the performance of the FA-based classifier compared to PSO and ABC classifiers and nine traditional classifiers. The experimental results showed that the FA-based classifier obtained the best classification error rate in eight data sets compared to other classifiers. Moreover, the FA-based classifier achieved the best averaged classification error rate. Other research related to centroid-based classification algorithms using nature-inspired algorithms can be found in [15], [16].

III. Preliminaries

A. Differential Evolution Method

DE starts with a set of NP individuals that form the population. Each individual is represented by a d-dimensional vector $\vec{x}_{i,G} = \{y_1, y_2, y_3, \ldots, y_d\}$, $i = \{1, 2, \ldots, NP\}$, which is randomly initialized in an d-dimensional problem space. Here, G is the generation.

After initialization, the fitness of the individuals are evaluated using an objective function. Then, the current population in generation G goes through the mutation, crossover, and selection operations to generate a new population in generation $G+1$. In the mutation operation, the $DE/best/1/bin$ schema is used to create a mutant vector $\vec{m_{i,G}}$ for each target vector $\vec{x}_{i,G}$ as follows:

$$\vec{m}_{i,G} = \vec{x}_{best,G} + F.(\vec{x}_{r1,G} - \vec{x}_{r2,G}) \tag{1}$$

where $best$ is the index of the best vector in the current population at generation G. $r1$ and $r2$ are random values chosen from $\{1, 2, \ldots, NP\}$, which should be mutually different and different from the values of the best vector and the target vector. The F parameter is a scaling factor of the difference vector, which controls the evolution rate of the generation. In our work, the F value is a random value within the range [0.5, 1.0] using Eq. 2 [7].

$$F = 0.5 * (1 + rand(0, 1)) \tag{2}$$

After the mutant vectors are generated, the crossover operation is carried out to improve the diversity of the population by combining target vector $\vec{x}_{i,G}$ with their mutant vector $\vec{m}_{i,G}$ using Eq. 3 [9], which will finally lead to produce a trail vector $\vec{t}_{i,G}$. The CR value is computed using Eq. 4 [7], which begins from $CR_{max} = 1.0$ and then its value linearly decreases with increasing numbers of generations until it reaches $CR_{min} = 0.5$. At the beginning, the CR value

is close to 1.0, which means that most of the target vector elements are replaced by the elements of the mutant vector. But at later generations, the CR value will be decreased linearly, which could lead to the trail vector to inherit more elements from the target vector [7]. In addition, the condition $rand_j == j$ is added to ensure at least one element is inherted from the mutant vector [9]. Each element in the trail vector $\vec{t}_{i,G}$ should be within the range [0,1].

$$\vec{t}_{j,i,G} = \begin{cases} \vec{m}_{j,i,G} & if\ (rand(0,1.0) \leq CR_G\ or\ rand_j == j) \\ \vec{x}_{j,i,G} & otherwise \end{cases} \tag{3}$$

$$CR_G = CR_{max} - \left((CR_{max} - CR_{min}) \frac{G}{G_{max}} \right), \quad G_{max} : maximum\ number\ of\ generations \tag{4}$$

The outcome of the crossover operation is the trail vectors, which are the candidate vectors for the next generation (G+1). The target and trial vectors are evaluated using an objective function to measure their fitness level. After that, "the survival of the fittest" principle is applied to choose between the target vector $\vec{x}_{i,G}$ and its trail vector $\vec{t}_{i,G}$ using the following selection rule:

$$\vec{x}_{i,G+1} = \begin{cases} \vec{t}_{i,G} & if\ (FL(\vec{t}_{i,G})\ is\ better\ than\ FL(\vec{x}_{i,G})) \\ \vec{x}_{i,G} & otherwise \end{cases} \tag{5}$$

here, FL is the fitness. According to the selection rule 5, the target vector $\vec{x}_{i,G}$ is replaced by the trial vector $\vec{t}_{i,G}$ in the next generation (G+1), if the $\vec{t}_{i,G}$ vector has a fitness level better than the $\vec{x}_{i,G}$ vector. Otherwise, the $\vec{x}_{i,G}$ vector survives to the next generation (G+1).

After the new generation (G+1) is created, the mutation, crossover, and selection operations are repeated until the maximum number of generation is reached.

B. Differential Evolution Classifier (CDE)

In [5], [6], the authors proposed variants of a centroid-based DE classification algorithm (CDE) based on different objective functions. The main idea of CDE is to find the optimal centroid of each class label in a training data set and then assign the instances of the unseen data set to the closest centroid.

In CDE, each individual is formed as follows [5]:

$$\vec{x}_i = \{\vec{v}^{c_1}, \vec{v}^{c_2}, \ldots, \vec{v}^{c_n}\} \tag{6}$$

here, $c_1, c_2, \ldots, c_n$ are the class labels in a training data set and $\vec{v}^{c_n}$ is the centroid vector of class label c_n, which is a D-dimensional vector. The centroid vectors of the initial population are initialized randomly in the D-dimensional problem space. After that, the initial population goes through repeated processes of mutation, crossover, and selection to improve the initial population as described in Section III-A.

For the fitnees evaluation, three types of objective functions were proposed to tackle data classification. The first objective function (F_1) evaluates the fitness of an individual by computing the rate of misclassification after assigning all instances to the closest centroid according to the shortest distance[6]. Mathematically, the F_1 function is described as follows:

$$F_1(\vec{x}_i) = \frac{1}{N} \sum_{j=1}^{N} \gamma(\vec{I}_j) \tag{7}$$

$$\gamma(\vec{I}_j) = \begin{cases} 1 & if \quad \hat{y} \neq y \\ 0 & otherwise \end{cases} \tag{8}$$

where N is the total number of instances in a data set, $\hat{y}$ is the predicted class label of instance $\vec{I}_j$, and y is the actual class of $\vec{I}_j$.

The second objective function (F_2) computes the fitness of an individual by taking the average of the sum of all Euclidean distances between the current centroid of class label c_n and the instances in a training data set that belong to class label c_n according to the training data set [5]. Mathematically, the F_2 function is described as follows:

$$F_2(\vec{x}_i) = \frac{1}{N} \sum_{j=1}^{N} d\left(\vec{I}_j^{c_n}, \vec{v}^{c_n}\right) \tag{9}$$

The third objective function (F_3) taken from [10], which is the best fitness function that was reported in [10] to tackle data classification. F_3 is a linear combination of two values computed by F_1 and F_2. Mathematically, the F_3 function is described as follows:

$$F_3(\vec{x}_i) = \frac{1}{2}(F_1(\vec{x}_i) + F_2(\vec{x}_i)) \tag{10}$$

IV. Proposed Approach: Cost-Sensitive Differential Evolution Classifier

The existing variants of CDE have shown efficient to do the data classification task on the balanced binary data sets [5], [6]. However, these variants suffer from handling imbalanced binary data sets. To address this drawback, we propose a cost-sensitive version of CDE in order to cope with imbalanced binary data sets by minimizing the misclassification cost instead of the misclassification error.

In our work, we propose a new objective function F_{cost}, which is a minimization objective function. In F_{cost}, first each class label should be assigned a misclassification cost. Then, F_{cost} computes the fitness level of the individual vector $\vec{x}_{i,G}$ in two steps. In the first step, all instances in a training data set are assigned to the closest centroid according to the Euclidean distance. After that, the second step is carried out by summing over the misclassification cost of all instances that are misclassified as follows:

$$F_{cost}(\vec{x}_i) = \sum_{j=1}^{N} \psi_{cost}(\vec{I}_j) \tag{11}$$

$$\psi_{cost}(\vec{I}_j) = \begin{cases} Cost^+ & if \quad \hat{y} \neq y \quad and \quad y = + \\ Cost^- & if \quad \hat{y} \neq y \quad and \quad y = - \\ Otherwise & 0 \end{cases} \quad (12)$$

where $Cost^+$ is the misclassification cost of the positive class label, $Cost^-$ is the misclassification cost of the negative class label, N is the total number of instances in the data set, $\hat{y}$ is the predicted class label of instance $\vec{I}_j$, and y is the actual class label of $\vec{I}_j$.

V. Medical Application: Data set and Data set preparation

In our work, we used the latest version of the SEER data (released in April 2018) [17], [18] to evaluate the performance of our proposed algorithm. The SEER data has been collected from various registries in the United States by the Surveillance, Epidemiology, and End Results (SEER) program of the National Cancer Institute (NCI) since 1973. The SEER data set is a public data set, which contains information of cancer incidences in the US that cover 34.6% of the US population [19].

In the SEER data set, each record represents the information of cancer incidence for one patient stored as fixed-length text. We developed a preprocessing tool using Java SDK 1.8 to extract the sub data sets of breast, lung, uterus, and stomach cancer incidences from the SEER data. The chosen features (variables) for these data sets are the same that have been used in previous research [20], [21], [22], [23] besides using three features, which are Vital Status Recode, Cause Of Death, and Survival Months to tag an instance. It should be noted here that the cancer incidences that were extracted for our experiments have been diagnosed since 2004, because some of the SEER features such as Tumor Size, Tumor Extension, and Lymph Nodes, are just reported for cancer incidences that have been diagnosed since 2004.

The survivability of cancer patients is determined by the rule that is shown in Fig 1 [20], [21], [22]. According to that rule, the patient who lives at least five years (60 months) since cancer diagnosis; this patient is tagged as "Survived". The patient who died within five years after cancer diagnosis, and the cause of death is the same type of cancer that was diagnosed at the beginning; this patient is tagged "Not Survived". Otherwise, a record is ignored.

For the patients who have more than one record in the SEER data set, the most recent record of a patient was extracted. Concerning the records which have missing values in the chosen features, we decided to exclude these records.

Table I shows the properties of the breast, uterus, lung and stomach cancers data sets after preprocessing. In Table I, we reported the number of instances and distribution percentage of each class label, and the total number of instances in each data set. From Table I, we can easily see that all data sets are highly imbalanced. For example, in the breast cancer data set, 87.67% of instances belong to class label "Survive" while the remaining 12.33% belonging to class label "Not Survive". All data sets are normalized using Min-Max normalization before applying the CDE algorithm.

if SM ≥ 60 and VSR=" alive " **then**
  tag the patient as "Survived"
else if SM < 60 and COD=" Type of cancer " **then**
  tag the patient as "Not Survived"
else
  Ignore this record
end if

Fig. 1: Rule of the survivability of cancer patients [20], [21], [22]. (VSR: Vital Status Recode, SM: Survival Months, and COD: Cause of Death)

TABLE I: Properties of Data Sets

Data set	Survive (-)	Not Survive (+)	Total
Breast Cancer	271,972 (87.67%)	38,272 (12.33%)	310,244
Lung Cancer	23,898 (18.82%)	103,092 (81.18%)	126,990
Uterus Cancer	23,337 (76.16%)	7,303 (23.84%)	30,640
Stomach Cancer	4,169 (27.65%)	10,909 (72.35%)	15,078

VI. Experiments and Results

In our work, we conducted two experiments to evaluate the robustness and performance of the cost-sensitive CDE algorithm for predicting the survivability of cancer patients. In the first experiment, we ran the CDE algorithm using the F_{cost} objective function to assess the impact of F_{cost} on the performance of CDE compared to the existing fitness functions (F_1,F_2, and F_3). For the second experiment, we compared the cost-sensitive CDE algorithm with five cost-sensitive Machine Learning (ML) algorithms according to their performance in predicting the survivability of cancer patients.

For the classifier's performance evaluation, the Area Under Curve (AUC) and Geometric Mean (G-mean) measures are used to evaluate the performance and robustness of the classification algorithms among other evaluation measures. The reason for that is that all data sets are imbalanced; thus accuracy, misclassification rate, or recall and precision measures do not reflect the accurate performance of the classification algorithms.

AUC is the most important and popular measure in medical applications, which summarizes the performance of a classifier into a single value. AUC is the calculated area under the Receiver Operating Characteristic (ROC) curve, which is calculated as follows [24], [25], [26]:

$$AUC = \frac{1 + TP_{rate} - FP_{rate}}{2} \quad (13)$$

G-mean is another measure to assess a classifier's performance on a highly imbalanced data set, which is sensitive to the True rate of the minority class label [27], [28], [29]. This measure is calculated as follows:

$$G - mean = \sqrt{sensitivity \times specificity} \quad (14)$$

TABLE II: Misclassification cost of Survive (-) and Not Survive (+) class labels for each data set

Data set	Survive (-)	Not Survive (+)
Breast Cancer	1.0	7.0 *
Lung Cancer	4.0 *	1.0
Uterus Cancer	1.0	3.0 *
Stomach Cancer	2.0 *	1.0

* Minority class label.

In our experiments, all data sets in Table I were split into two data sets; 80% of the data set was used as the training data set, and the remaining 20% as the testing data set.

It should be noted here that the misclassification cost of the class labels are specified empirically starting with one, then the cost is incremented until finding the best misclassification cost. Table II shows the best misclassification cost of the "Not Survive (+)" and "Survive (-)" class labels that were found in each data set.

Moreover, the parameters of the CDE algorithm were taken from [7], except the maximum number of generations, which are:

- Maximum number of generations = 200
- Population size NP = 10 * (Data Set Dimensionality)
- Crossover range [CR_{min} = 0.5, CR_{max} = 1.0]
- Scaling factor F = a random value in range [0.5, 1.0]

A. First Experiment

To evaluate the impact of our proposed objective function (F_{cost}) on the performance of the CDE algorithm when applied to imbalanced binary data sets, we compared its performance with the performance of three variants of CDE that are based on F_1,F_2, and F_3. Four variants of CDE are introduced based on the objective function, which are abbreviated as follows: $CDE-F_{cost}$, $CDE-F_1$, $CDE-F_2$ and $CDE-F_3$. We performed 25 independent runs for $CDE-F_{cost}$, $CDE-F_1$, $CDE-F_2$, and $CDE-F_3$ on all data sets given in Table I and reported the G-mean and AUC results. Furthermore, the G-mean and AUC results obtained by $CDE-F_{cost}$ have been compared statistically with the G-mean and AUC results obtained by $CDE-F_1$, $CDE-F_2$, and $CDE-F_3$ using the Wilcoxon Signed-Rank test at the 5% significance level and we reported the p-value. Table III and IV show the average of the G-mean and AUC results (25 independent runs), respectively, obtained by $CDE-F_{cost}$, $CDE-F_1$, $CDE-F_2$, and $CDE-F_3$ for each data set. In addition, we reported the standard deviation within brackets in both tables.

From the results in Table III, we can see that the performance of $CDE-F_{cost}$ statistically outperforms the performance of $CDE-F_1$, $CDE-F_2$, and $CDE-F_3$ in terms of G-mean for all data sets, where the p-value is 1.2290e-05 for all. The best G-mean results on the Breast, Lung, Uterus, and Stomach cancer data sets were obtained by $CDE-F_{cost}$, which were 79.60%, 83.87%, 83.97%, and 79.49%, respectively. In terms of AUC, the results in

TABLE III: G-mean results achieved by $CDE-F_1$, $CDE-F_2$, $CDE-F_3$, and $CDE-F_{cost}$

	$CDE-F_{cost}$	$CDE-F_1$	$CDE-F_2$	$CDE-F_3$
Breast Cancer	**79.60% [±0.10%]**	61.51% [±0.46%]	77.61% [±0.04%]	61.78% [±0.43%]
Lung Cancer	**83.87% [±0.09%]**	72.62% [±0.88%]	80.76% [±0.05%]	73.03% [±0.67%]
Uterus Cancer	**83.97% [±0.17%]**	79.81% [±0.63%]	83.03% [±0.04%]	80.14% [±0.49%]
Stomach Cancer	**79.49% [±0.37%]**	74.79% [±0.49%]	74.04% [±0.07%]	74.96% [±0.48%]
Average	**81.73%**	72.18%	78.86%	72.48%

TABLE IV: AUC results achieved by $CDE-F_1$, $CDE-F_2$, $CDE-F_3$, and $CDE-F_{cost}$

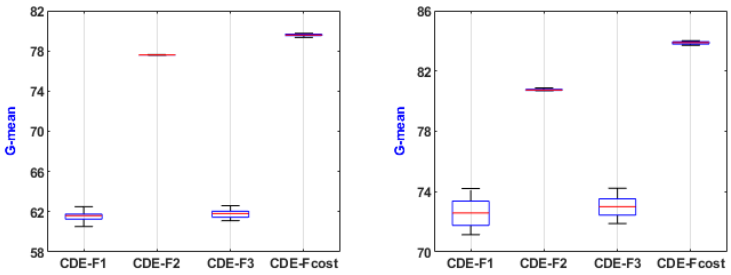
	$CDE-F_{cost}$	$CDE-F_1$	$CDE-F_2$	$CDE-F_3$
Breast Cancer	**79.61% [±0.10%]**	68.40% [±0.26%]	77.64% [±0.04%]	68.56% [±0.25%]
Lung Cancer	**83.93% [±0.09%]**	74.97% [±0.62%]	80.81% [±0.05%]	75.18% [±0.48%]
Uterus Cancer	**83.98% [±0.16%]**	80.89% [±0.47%]	83.05% [±0.04%]	81.10% [±0.40%]
Stomach Cancer	**79.60% [±0.34%]**	76.32% [±0.35%]	74.21% [±0.08%]	76.41% [±0.35%]
Average	**81.78%**	75.15%	78.93%	75.31%

Table IV revealed that $CDE-F_{cost}$ obtained the best AUC on all data sets compared to $CDE-F_1$, $CDE-F_2$, and $CDE-F_3$, where the AUC results were 79.61%, 83.93%, 83.98%, and 79.60% for Breast, Lung, Uterus, and Stomach cancer data set, respectively. Futhermore, the performance of $CDE-F_{cost}$ statistically outperforms the performance of $CDE-F_1$, $CDE-F_2$, and $CDE-F_3$ on all data sets according to the p-value of 1.2290e-05 for all. Figures 2 and 3 show the distribution of the G-mean and AUC results (from 25 independent runs) that were obtained by $CDE-F_1$, $CDE-F_2$, $CDE-F_3$, and $CDE-F_{cost}$.

To draw a conclusion over all data sets, the G-mean and AUC results are averaged over all data sets, as shown in the last row in Table III and IV. From these results, we can see that $CDE-F_{cost}$ achieved the best averaged G-mean and AUC compared to $CDE-F_1$, $CDE-F_2$, and $CDE-F_3$. Moreover, the previous results revealed another significant conclusion that is that the F_{cost} objective function improves CDE's performance when applied to imbalanced binary data set.

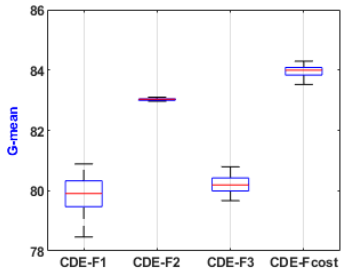
B. Second Experiment

In this experiment, we compare the performance of the cost-sensitive CDE, which is $CDE-F_{cost}$, in terms of G-mean and AUC with the performance of cost-sensitive classification algorithms. Five ML algorithms were employed that are cost-sensitive classification algorithms, which are Naive Bayes [30], Decision Tree C4.5 (J48) [31], Logistic Regression [32], RBF

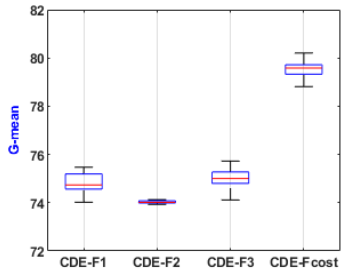


(a) Breast Cancer data set

(b) Lung Cancer data set

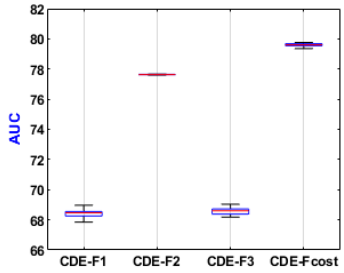


(c) Uterus Cancer data set

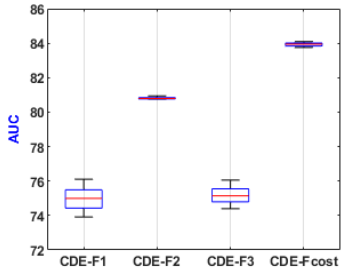


(d) Stomach Cancer data set

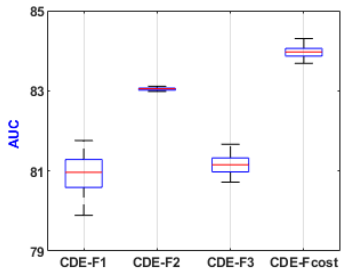
Fig. 2: Box plots of the G-mean results obtained by $CDE - F_1$, $CDE - F_2$, $CDE - F_3$, and $CDE - F_{cost}$ for Breast, Lung, Uterus, and Stomach Cancer data sets. Red bar inside the box represents the median; whiskers above and below the box represents maximum and minimum values, respectively.



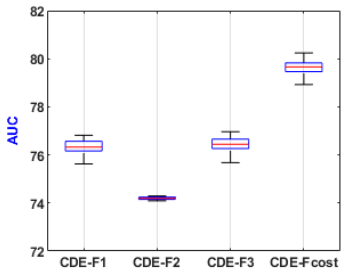
(a) Breast Cancer data set



(b) Lung Cancer data set



(c) Uterus Cancer data set



(d) Stomach Cancer data set

Fig. 3: Box plots of the AUC results obtained by $CDE - F_1$, $CDE - F_2$, $CDE - F_3$, and $CDE - F_{cost}$ for Breast, Lung, Uterus, and Stomach Cancer data sets. Red bar inside the box represents the median; whiskers above and below the box represents maximum and minimum values, respectively.

TABLE V: G-mean results for five cost-sensitive classification algorithm and $CDE - F_{cost}$

	$CDE - F_{cost}$	**Naïve Bayes**	**Decision Tree (C4.5)**	**Logistic Regression**	**RBF Network**	**IBK**
Breast Cancer	**79.60% [±0.10%]**	69.34%	75.73%	79.33%	76.15%	75.85%
Lung Cancer	**83.87% [±0.09%]**	82.67%	80.11%	83.66%	82.99%	79.88%
Uterus Cancer	**83.97% [±0.17%]**	81.99%	83.31%	83.78%	81.10%	81.43%
Stomach Cancer	79.49% [±0.37%]	74.27%	**81.12%**	79.65%	78.42%	75.50%
Average	**81.73%**	77.07%	80.07%	81.61%	79.66%	78.17%

TABLE VI: AUC results for five cost-sensitive classification algorithm and $CDE - F_{cost}$

	$CDE - F_{cost}$	**Naïve Bayes**	**Decision Tree (C4.5)**	**Logistic Regression**	**RBF Network**	**IBK**
Breast Cancer	**79.61% [±0.10%]**	71.97%	76.34%	79.39%	76.82%	75.85%
Lung Cancer	**83.93% [±0.09%]**	82.87%	80.19%	83.77%	83.01%	80.36%
Uterus Cancer	**83.98% [±0.16%]**	82.17%	83.37%	83.80%	81.31%	81.60%
Stomach Cancer	79.60% [±0.34%]	75.39%	**81.14%**	79.76%	78.42%	75.59%
Average	**81.78%**	78.10%	80.26%	81.68%	79.89%	78.35%

network [33], and IBk (5 nearest neighbors) [34]. For the cost-sensitive classification algorithms, the cost matrix should be provided by assigning the misclassification penalty for each class label. We used the same misclassification cost given in Table II for the five cost-sensitive classification algorithms. For running the cost-sensitive classification algorithm, the Waikato Environment for Knowledge Analysis (WEKA) tool version 3.6 was used [35],[36], which is one of the most popular tools for running data mining tasks. Furthermore, the default setting provided by the WEKA was used for the five ML algorithms.

Table V and VI show the G-mean and AUC obtained by the five cost-sensitive classification algorithms and $CDE - F_{cost}$, respectively, for the given data sets. It can be seen from the results in Table V and VI that $CDE - F_{cost}$'s performance outperforms the performance of all cost-sensitive classification algorithms in 3 out of 4 data sets in terms of G-mean and AUC. $CDE - F_{cost}$ achieved the best G-mean on the Breast, Lung, and Uterus cancer data sets, where the G-mean results were 79.60%, 83.87%, and 83.97%, respectively. Additionally, the best AUC results on the Breast, Lung, and Uterus cancer data sets were obtained by $CDE - F_{cost}$, where the AUC results were 79.61%, 83.93%, and 83.98%, respectively. For the Stomach cancer data set, $CDE - F_{cost}$'s performance outperforms the performance of 3 out of 5 cost-sensitive classification algorithms in terms of G-mean and AUC. The best G-mean and AUC results for the stomach cancer data set was achieved by the cost-sensitive Decision Tree (C4.5) algorithm.

Finally, we summarized the results by taking the average of the G-mean and AUC results of each algorithm over all data sets as shown in the last row in Table V and VI. From the average results, we can deduce that $CDE - F_{cost}$ achieved the best averaged G-mean and AUC results, which are 81.73% and 81.78%, respectively, outperforming all cost-sensitive classification algorithms.

VII. Conclusion and Future Work

In this work, we proposed the cost-sensitive differential evolution ($CDE - F_{cost}$) based on a new objective function F_{cost}. Two experiments were conducted using four real medical data sets extracted from the SEER data to assess the performance of $CDE - F_{cost}$ starting by evaluating the impact of the objective function F_{cost} on the performance of the centroid-based differential evolution classification algorithm (CDE) and then comparing the performance of the $CDE - F_{cost}$ with the performance of the five cost-sensitive classification algorithms with respect to the G-mean and AUC measures.

The findings revealed that CDE efficiently handles highly imbalanced binary data sets using the F_{cost} objective function. Additionally, F_{cost}'s performance outperformed the performance of three existing objective functions (F_1, F_2, and F_3)

with respect to G-mean and AUC. Furthermore, $CDE - F_{cost}$ performed best in 3 out of 4 cancer data sets compared to the five cost-sensitive classification algorithms. For the Stomach cancer data set, $CDE - F_2$'s performance outperformed the performance of three out of the five cost-sensitive classification algorithms. Overall, the experimental results showed that $CDE - F_{cost}$ can be successfully used to cope with highly imbalanced binary data sets. In addition, it is competitive compared to all cost-sensitive classification algorithms.

For future research, we will apply the cost-sensitive CDE algorithm to another real-life application such as intrusion detection and fraud detection, and study and analyze the algorithm's performance. Furthermore, we will propose a parallel version of the cost-sensitive CDE algorithm using the Hadoop Map-Reduce or Spark framework to deal with large data and investigate the scalability and performance of the algorithm.

REFERENCES

[1] B. Krawczyk, "Learning from imbalanced data: open challenges and future directions," *Progress in Artificial Intelligence*, vol. 5, no. 4, pp. 221–232, 2016.

[2] N. Madavan, "On improving efficiency of differential evolution for aerodynamic shape optimization applications," in *10th AIAA/ISSMO Multidisciplinary Analysis and Optimization Conference*, 2004, p. 4622.

[3] R. Storn and K. Price, "Differential evolution-a simple and efficient heuristic for global optimization over continuous spaces," *Journal of global optimization*, vol. 11, no. 4, pp. 341–359, 1997.

[4] U. K. Chakraborty, *Advances in differential evolution*. Springer, 2008, vol. 143.

[5] I. De Falco, A. Della Cioppa, and E. Tarantino, "Automatic classification of handsegmented image parts with differential evolution," in *Workshops on Applications of Evolutionary Computation*. Springer, 2006, pp. 403–414.

[6] P. Luukka and J. Lampinen, "A classification method based on principal component analysis and differential evolution algorithm applied for prediction diagnosis from clinical emr heart data sets," in *Computational Intelligence in Optimization*. Springer, 2010, pp. 263–283.

[7] A. Abraham, S. Das, and A. Konar, "Document clustering using differential evolution." in *IEEE Congress on Evolutionary Computation*, 2006, pp. 1784–1791.

[8] P. Luukka and J. Lampinen, "Differential evolution classifier in noisy settings and with interacting variables," *Applied Soft Computing*, vol. 11, no. 1, pp. 891–899, 2011.

[9] D. Koloseni *et al.*, "Differential evolution based classification with pool of distances and aggregation operators," *Acta Universitatis Lappeenrantaensis*, 2015.

[10] I. De Falco, A. Della Cioppa, and E. Tarantino, "Facing classification problems with particle swarm optimization," *Applied Soft Computing*, vol. 7, no. 3, pp. 652–658, 2007.

[11] D. Dua and C. Graff, "UCI machine learning repository," 2017, http://archive.ics.uci.edu/ml, University of California, Irvine, School of Information and Computer Sciences.

[12] D. Karaboga and C. Ozturk, "A novel clustering approach: Artificial bee colony (abc) algorithm," *Applied soft computing*, vol. 11, no. 1, pp. 652–657, 2011.

[13] X.-S. Yang, *Nature-inspired metaheuristic algorithms*. Luniver press, 2010.

[14] J. Senthilnath, S. Omkar, and V. Mani, "Clustering using firefly algorithm: performance study," *Swarm and Evolutionary Computation*, vol. 1, no. 3, pp. 164–171, 2011.

[15] R. Omidvar, A. Eskandari, N. Heydari, F. Hemmat, and S. Esmaeili, "Data clustering using by chaotic sspco algorithm," *Majlesi Journal of Electrical Engineering*, vol. 11, no. 2, 2017.

[16] R. Razavi-Far, V. Palade, and E. Zio, "Invasive weed classification," *Neural Computing and Applications*, vol. 26, no. 3, pp. 525–539, 2015.

[17] "Surveillance, epidemiology, and end results (seer) program (www.seer.cancer.gov) research data (1973-2015), national cancer institute, dccps, surveillance research program, released april 2018, based on the november 2017 submission."

[18] "Seer data," https://seer.cancer.gov/data/ , Accessed on Dec-10-2018.

[19] "Overview of the seer program," https://seer.cancer.gov/about/overview.html , Accessed on Jan-10-2019.

[20] A. Bellaachia and E. Guven, "Predicting breast cancer survivability using data mining techniques," *Age*, vol. 58, no. 13, pp. 10–110, 2006.

[21] E. S. H. Pour, "Stage-specific predictive models for cancer survivability," Ph.D. dissertation, The University of Wisconsin-Milwaukee, 2016.

[22] E. S. H. Pour and R. J. Kate, "Stage-specific survivability prediction models across different cancer types," in *AMIA Annual Symposium Proceedings*, vol. 2017. American Medical Informatics Association, 2017, p. 1421.

[23] D. Delen, G. Walker, and A. Kadam, "Predicting breast cancer survivability: a comparison of three data mining methods," *Artificial intelligence in medicine*, vol. 34, no. 2, pp. 113–127, 2005.

[24] A. Tharwat and T. Gabel, "Parameters optimization of support vector machines for imbalanced data using social ski driver algorithm," *Neural Computing and Applications*, pp. 1–14, 2019.

[25] E. K. Aydogan, M. Ozmen, and Y. Delice, "Cbr-pso: cost-based rough particle swarm optimization approach for high-dimensional imbalanced problems," *Neural Computing and Applications*, pp. 1–19, 2018.

[26] M. Antonelli, P. Ducange, and F. Marcelloni, "An experimental study on evolutionary fuzzy classifiers designed for managing imbalanced datasets," *Neurocomputing*, vol. 146, pp. 125–136, 2014.

[27] U. Bhowan, M. Johnston, M. Zhang, and X. Yao, "Reusing genetic programming for ensemble selection in classification of unbalanced data," *IEEE Transactions on Evolutionary Computation*, vol. 18, no. 6, pp. 893–908, 2014.

[28] A. Tharwat, "Classification assessment methods," *Applied Computing and Informatics*, 2018.

[29] C. Zhang, K. C. Tan, H. Li, and G. S. Hong, "A cost-sensitive deep belief network for imbalanced classification," *IEEE transactions on neural networks and learning systems*, no. 99, pp. 1–14, 2018.

[30] G. H. John and P. Langley, "Estimating continuous distributions in bayesian classifiers," in *Eleventh Conference on Uncertainty in Artificial Intelligence*. Morgan Kaufmann, 1995, pp. 338–345.

[31] R. Quinlan, *C4.5: Programs for Machine Learning*. San Mateo, CA: Morgan Kaufmann Publishers, 1993.

[32] N. Landwehr, M. Hall, and E. Frank, "Logistic model trees," vol. 95, no. 1-2, pp. 161–205, 2005.

[33] D. Broomhead and D. Lowe, "Multivariable functional interpolation and adaptive networks," *Complex Systems*, vol. 2, no. 3, 1988.

[34] D. Aha and D. Kibler, "Instance-based learning algorithms," *Machine Learning*, vol. 6, pp. 37–66, 1991.

[35] M. Hall, E. Frank, G. Holmes, B. Pfahringer, P. Reutemann, and I. H. Witten, "The weka data mining software: an update," *ACM SIGKDD explorations newsletter*, vol. 11, no. 1, pp. 10–18, 2009.

[36] I. H. Witten, E. Frank, M. A. Hall, and C. J. Pal, *Data Mining: Practical machine learning tools and techniques*. Morgan Kaufmann, 2016.