

Preprints are preliminary reports that have not undergone peer review. They should not be considered conclusive, used to inform clinical practice, or referenced by the media as validated information.

An Efficient Chaotic Salp Swarm Optimization Approach Based on Ensemble Algorithm for Solving Class Imbalance Problem

Rekha G

KL University

Krishna Reddy V

KL University

chandrashekar jatoth (chandrashekar.jatoth@gmail.com)

NIT Hamirpur: National Institute of Technology Hamirpur https://orcid.org/0000-0002-8536-0210

Ugo Fiore

University of Naples Federico II: Universita degli Studi di Napoli Federico II

Research Article

Keywords: Imbalanced data, Feature selection, Ensemble algorithms, Classification, Salp swarm algorithm

Posted Date: July 23rd, 2021

DOI: https://doi.org/10.21203/rs.3.rs-611446/v1

License: (a) This work is licensed under a Creative Commons Attribution 4.0 International License. Read Full License

Version of Record: A version of this preprint was published at Soft Computing on August 20th, 2021. See the published version at https://doi.org/10.1007/s00500-021-06080-x.

An Efficient Chaotic Salp Swarm Optimization Approach Based on Ensemble Algorithm for Class Imbalance Problems

Rekha Gillala · Krishna Reddy Vuyyuru · Chandrashekar Jatoth ⊠ · Ugo Fiore

Received: date / Accepted: date

Abstract Class imbalance problems have attracted the research community but a few works have focused on feature selection with imbalanced datasets. To handle class imbalance problems, we developed a novel fitness function for feature selection using the chaotic salp swarm optimization algorithm, an efficient meta-heuristic optimization algorithm that has been successfully used in a wide range of optimization problems. This paper proposes an Adaboost algorithm with chaotic salp swarm optimization and Adaboost classifiers are thereafter trained on the features selected. Experiments show the ability of the proposed technique to find the optimal features with performance maximization of Adaboost.

Keywords Imbalanced data \cdot Feature selection \cdot Ensemble algorithms \cdot Classification \cdot Salp swarm algorithm

 Rekha Gillala

 Department of Computer Science and Engineering,

 Koneru Lakshmaiah Education Foundation, India

 E-mail: gillala.rekha@klh.edu.in

 Krishna Reddy Vuyyuru

 Department of Computer Science and Engineering,

 Koneru Lakshmaiah Education Foundation, India

 E-mail: avkrishnareddy@kluniversity.in

 Chandrashekar Jatoth (聲)

 Department of Computer Science and Engineering,National Institute of Technology Hamirpur, India

 E-mail: chandrashekar.jatoth (聲)

 Department of Computer Science and Engineering,National Institute of Technology Hamirpur, India

 E-mail: chandrashekar.jatoth@gmail.com

 Ugo Fiore

 Department of Management and Quantitative Studies, Parthenope University of Naples, Italy

 E-mail: ugo.fiore@uniparthenope.it

1 Introduction

Classification plays an important role in machine learning and data mining. Most traditional classification algorithms work well in a condition of balanced distribution of samples among the classes. However, when class distribution is imbalanced, classifiers often encounter performance degradation (Galar et al 2013). A dataset is said to be imbalanced when there are more instances in at least one of the classes compared to the others. The class with the greatest number of instances is called majority class, while classes with lower number of instances are defined minority classes (Searle and Searle 1987). In imbalanced classification problems, the minority class is often the class of interest, and the cost of misclassification is higher with the minority class than it is with the others (Maldonado et al 2014; Di Martino et al 2013; López et al 2013; Rekha and Reddy 2018; Zhai et al 2018). For example, in cancer diagnosis many more people will not have cancer in comparison to those who have it. If a patient who has cancer is mistakenly diagnosed as not having it, disaster may arise as treatment is not given (Menardi and Torelli 2014). Classification on imbalanced datasets is a challenging problem, basically because the lack of a sufficient number of samples of the minority class complicates the task of identifying combinations of features with discriminative power.

Many approaches to deal with imbalanced data have been proposed and they can be summarized into three categories: data level strategies, algorithm level strategies and cost sensitive algorithms. Data level strategies modify a dataset to overcome imbalance, trying to balance all classes. In the past fifteen years, numerous oversampling and under-sampling algorithms have been proposed. One of the most popular among them is Synthetic Minority Oversampling TEchnique (SMOTE) proposed by Chawla (Chawla et al 2002). SMOTE uses interpolation to generate synthetic samples for minority class. SMOTE-based improved sampling algorithms were very popular in data-level techniques (Ramentol et al 2012; Gao et al 2011; Verikas et al 2010). Variants of SMOTE also try to leverage on the distribution of majority class samples. Recently, Fiore (2019) introduced attraction-repulsion Weber problem, to generate samples that are as close as possible to minority samples and as far as possible to majority samples as well. Other approaches included the use of Generative Adversarial Networks to produce synthetic samples that are indistinguishable from real ones (Fiore et al 2019). Under-sampling and oversampling can improve the performance of classifiers by balancing the skewed distribution. However, they are criticized because important information may be lost with undersampling, while redundant or inappropriate samples may be generated with oversampling.

Strategies at the algorithmic level focus on changing existing classification algorithms to improve the ability of classifiers to learn from minority classes (Galar et al 2011). Cost-sensitive and one-class learning are the common techniques proposed in the literature (López et al 2012; Cao et al 2013). In order to increase the classifier learning potential, cost-sensitive algorithms adopts a cost matrix by assigning higher misclassification costs to minority samples while training the algorithm. Researchers have suggested many approaches in this respect, including the combination of data-level approaches with cost-sensitive learning, the cost of collecting features

before classification, and the cost of using class distribution prior to classification (Thanathamathee and Lursinsap 2013; Weiss et al 2013; Thai-Nghe et al 2010).

Ensemble level techniques are also widely used currently for class imbalance problems. Bagging (Sun et al 2018; Chung and Kim 2015) and Boosting (Viola and Jones 2002; Li et al 2015; Chawla et al 2003; Joshi et al 2001) are two popular ensemble classifiers. Many researchers proposed various research articles to improve the performance of AdaBoost classifier (Li et al 2019; Wang et al 2019). Haixiang et al (2016) proposed an algorithm to handle multi-class imbalance problems called BSPO-AdaBoost-KNN algorithm. The algorithm worked significantly well in identifying the important features. Yijing et al (2016) proposed an adaptive classifier to balance the skewed data distribution very efficiently. It employed feature selection and preprocessing to train multiple classifiers. Xinwu et al (2016) used graphic and math model to analyzes the multi-class Adaboost algorithm and proposed a new kind of multi-class classification method to decrease the accuracy of the weak classifier effectively. The Extreme Learning Machine (ELM) is an effective multi-classification learning algorithm but does not work for imbalanced datasets. By enclosing weighted ELM algorithms in the AdaBoost framework, Li et al (2018) suggested AdaBoost weighted composite kernel ELM. For each sample, this architecture integrates spatial and spectral information into the composite kernel. The combination of composite kernel methods and the Adaboost architecture with weighted ELM provided excellent performance in improving the accuracy of the classification. An improved AdaBoost algorithm with a weight vector was proposed by Dou and Chen (2017). In order to represent the recognition power of the base classifiers, the weight vector assigns weight to each individual class. This algorithm greatly increases the precision of classification by preventing overfitting. For imbalanced data classification, an ensemble evolutionary algorithm was suggested by Li et al (2017). For better classification, they applied genetic algorithms to the AdaBoost classifier. In (Qiaojin et al 2008) the author proposed four algorithms based on exceeded threshold value of majority class samples. The algorithms are defined based on threshold growth such as A-AdaBoost, B-AdaBoost, C-AdaBoost and D-AdaBoost. The algorithms provide effective results when handling imbalanced data. In the study, Particle Swarm Optimization (PSO) was used to resolve the problems of class imbalance, but the main disadvantage of the PSO algorithm is that in high-dimensional space it may fall into the local optimum and has a low convergence rate in the iterative process.

Many machine learning algorithms do not handle large feature sets very well (Yin et al 2013; Nikhath and Subrahmanyam 2019), thus several techniques are targeted at applying feature selection in an efficient manner. Selection of features can be categorized into two methods: (i) filter methods, and (ii) wrapper methods. In filter-based methods, features are selected based on the properties of the data without using learning algorithms, whereas wrapper methods use a learning algorithm to select the best feature subset according to some optimality criterion. In comparison to filter methods, wrapper methods are more accurate but more expensive computationally (Haixiang et al 2016; Li et al 2018). Imbalanced datasets exacerbate the problem of efficient feature selection, as there are limited samples to base choices on. Performance is also becoming increasingly important in conjunction with the surge in data volume and variety, owing to the availability of sensors. This trend is also going to acquire mo-

mentum with the adoption of paradigms such as transformative computing, especially when techniques such as cognitive analysis will be applied (Ogiela and Ogiela 2020).

In this paper, we propose a novel approach based on feature selection using Chaotic Salp Swarm Optimization (CSSA) and classification with Adaboost. Performance evaluation is done using the AUC (Area Under [the ROC] Curve). Further, CSSA is adopted to prevent being trapped in local optima and also to avoid slow convergence while selecting the features. Strengths of SSA include good convergence properties, good solution quality, the ability to work quite well on a wide range of optimization problems, adaptability, robustness, scalability, ease of implementation, and reasonable execution times. The Adaboost algorithm works well with different classification algorithms, which can be applied as weak classifiers. AdaBoost show a high degree of precision on a variety of problems.

The remainder of this paper is organized as follows. Section 2 provides an introduction to Adaboost and SSA. The illustration of feature selection using CSSA is presented in Section 3. Performance and Evaluation is presented in Section 4. Finally, the conclusions are reported in Section 5.

2 Preliminaries

This section presents the concepts of AdaBoost and bio-inspired optimization algorithm Salp Swarm Optimization (SSA).

2.1 Adaptive Boosting Algorithm (AdaBoost)

Adaboost is a boosting technique widely used to address classification problems and regression problems as well (Zhang and Chen 2017). Boosting is based on the repeated application of weak classifier, i.e. a classifier that is required to be only marginally more accurate than random guessing. Here, a decision tree is used as a base classifier. Weights are assigned to samples at each iteration and, based upon the results of previous iterations, weights are modified and the classifier is applied again. At the beginning, every sample has the same weight. If a sample is misclassified on an iteration, then its weight will be increased on the next iteration; conversely, if it is classified correctly, then its weight will be reduced. Direct application of the Adaboost algorithm to imbalanced classification problems may cause performance degradation, as the main concentration will be on misclassified samples rather than on minority samples.

2.2 Salp Swarm Optimization (SSA)

Evolutionary and swarm-based algorithms have recently been commonly used for the problem of selecting features. Swarm-based algorithms are collective optimization algorithms because improvement is applied on a set of solution, rather than on a single one. By applying agents to reach an optimal solution, these algorithms adaptively check the feature space. The bio-inspired optimization algorithms have been widely

used in feature selection such as Ant Colony Optimization (ACO) (Al-Ani 2005), Particle Swarm Optimization (PSO) (Xue et al 2012; Bewoor et al 2017; Amarendra and Reddy 2019; Namassivaya et al 2019) Artificial Bee Colony (ABC) (Schiezaro and Pedrini 2013; Sultanpure and Reddy 2018), Ant Lion Optimizier (ALO) (Emary et al 2016), cuckoo search algorithm (Thirugnanasambandam et al 2019), Salp Swarm Algorithm (SSA) (Ahmed et al 2018). The above algorithms have been widely used for feature selection but our study focus more on feature selection in imbalanced dataset.

SSA is a meta-heuristic algorithm based on nature that was proposed by Mirjalili et al (2017). The intuition of SSA emerged from the swarming behaviour of salps in deep oceans. These creatures form swarms, known as salp chains, believed to facilitate their movement in the waters through coordination. In SSA, the population of salps is divided into two roles: the leader and the followers. The leader determines the direction of movement, and the remaining salps follow suit.

The location of the *i*-th salp is represented by a *n*-dimensional vector, where *n* is the dimension of the search space (in our case the number of features).

The mathematical model for updating the position of the leader is as follows:

$$\mathbf{x}^{1} = \begin{cases} \mathbf{F} + r_{1} \left(\mathbf{l} + (\mathbf{u} - \mathbf{l}) r_{2} \right) & \text{if } r_{3} \ge 0 \\ \mathbf{F} - r_{1} \left(\mathbf{l} + (\mathbf{u} - \mathbf{l}) r_{2} \right) & \text{if } r_{3} < 0 \end{cases},$$
(1)

where \mathbf{x}^1 represents the position of the leader, \mathbf{u} and \mathbf{l} denote, respectively, the upper and lower bounds, the food location is \mathbf{F} , and r_1 , r_2 , r_3 are parameters that govern the behavior of the swarm. Note that in Eq. (1), \mathbf{x}^1 is meant to be at the current iteration, i.e., $\mathbf{x}^1(k)$, whereas all the variables in the right-hand side are meant to be at the previous iteration. For example, \mathbf{F} in Eq. (1) stands for $\mathbf{F}(k-1)$. To avoid cluttering the notation, explicit indexing has been suppressed.

The value of r_1 depends on the iteration and is determined as follows:

$$r_1 = 2\exp\left(-\left(\frac{4k}{K}\right)^2\right) \tag{2}$$

where *K* is the predetermined maximum number of iterations and the current iteration is defined by *k*. The other two parameters r_2 and r_3 are random numbers drawn from a uniform distribution in the interval [0,1]. The followers' position is modified using Eq. (3):

$$\mathbf{x}^{j}(k) = \frac{1}{2} \left(\mathbf{x}^{j}(k-1) + \mathbf{x}^{j-1}(k-1) \right)$$
(3)

where \mathbf{x}^{j} indicates the location of the *j*-th salp (which is a follower, since $j \ge 2$).

Since the location of the food source (the global optimum sought) is unknown, it is substituted by the position of the best solution achieved thus far. At each iteration, the fitness function is evaluated at the position of each salp, the best position is found and compared with \mathbf{F} . If the newly found position is advantageous with respect to the previous one, \mathbf{F} is updated accordingly.

3 The Proposed Approach for Feature Selection in Imbalanced Dataset

The SSA is a recently proposed optimization algorithm. It has unique features for solving optimization and feature selection problems. The characteristics of SSA is simple, efficient and easy to implement with only one parameter to balance exploration and exploitation. However, it suffered trapping at local optima and with low convergence rate. In feature selection problems, all solutions are limited to the binary values of 1 and 0, where 1 represents the presence of feature and 0 vice versa. For feature selection using SSA a binary representation has been used by applying equation Eq. (4), where r_4 is a random number uniformly sampled from the interval [0, 1]. Further, conventional SSA was modeled to solve continuous optimization problem and to convert into a discrete version, we applied sigmoid transfer function (Kennedy and Eberhart 1997).

$$x_i^j = \begin{cases} 1 & \text{if } s(x_i^j) \ge r_4 \\ 0 & \text{otherwise} \end{cases}$$
(4)

where $s(x_i^j) = \frac{1}{1 + e^{-x_i^j(s)}}$

3.1 Chaotic Mapping

Methods based on chaotic maps have recently been used to improve the efficiency of meta-heuristic algorithms (Sayed et al 2018). The optimal feature selection using SSA for class imbalance problems adopted chaotic mapping to avoid stagnation at local optima as mentioned in (Sayed et al 2018) and the optimal chaotic mapping to improve the original SSA's efficiency was through the logistic map. In SSA the three main parameters r_1, r_2, r_3 effects the performance. As r_2 shows a significant impact on balancing the exploitation and exploration, in our proposed approach, a logistic map is employed to adjust the r_2 parameter to avoid falling into local optima while classifying the features.

The logistic map is a function of the interval [0, 1] into itself, defined in Eq. (5)

$$y_{n+1} = \mu (1 - y_n), \quad 0 < \mu \le 4$$
 (5)

In Eq. (5), y_n is the value at the *n*-th iteration of the map, and μ is a constant whose value determines the behavior of the map. Chaos ensues for $\mu = 4$. Chaos is characterized by nonperiodicity and sensitive dependence on initial conditions. The value of the initial position y_0 is set to 0.7 as in (Sayed et al 2019).

3.2 Fitness Function

To determine the quality of each solution (salp position), a fitness function is used. The best solution for the class imbalance issue is one that increases the classifier's output with a minimal number of features selected. So, one of the effective performance evaluation for classification is to use AUC (Area Under The Curve) which

indicate a scalar value, in range [0,1] (Fawcett 2006). At each iteration, the imbalanced dataset is trained using the Adaboost classifier with the decision tree as the weak classifier. To improve the algorithm's classification performance, the error is measured using AUC as a fitness function. The AUC value is determined by using the area under the ROC curve (Galar et al 2011).

The fitness function is defined based on Eq. (6).

$$AUC_f = \sum_{i=1}^{N-1} \frac{1}{2} (FP_{i+1} - FP_i) (TP_{i+1} - TP_i)$$
(6)

In Eq. (6), the number of class thresholds to be used in the ROC curve is defined by N. The higher the value, the better the AUC approximation. The accuracy of the two classes is represented as TP_i , FP_i at the *i* class threshold (i.e. True Positive and False Positive rates, respectively).

3.3 Parameters Initialization

The salp swarm algorithm starts with random initialization of salp positions. The lower boundary and upper boundary values are initialized to 0 and 1 respectively. The population size is set to 50 for global optimization and number of iterations is set to 20. The terminating condition is when the maximum number of iterations met.

3.4 The CSSA-AdaBoost Framework

The AdaBoost algorithm combines multiple weak classifier into one single strong classifier by adjusting its weights at each iteration. The ensemble algorithms focus more on difficult examples without class differentiation. But in class imbalance problems, the majority class samples contribute more to the accuracy. Therefore, it is simpler to strengthen the true negatives instead of enhancing the true positives, which is not desirable. However, to overcoming these limitations, ensemble algorithms need be changed or combined with another technique to deal with imbalance problem.

In our approach, SSA algorithm is used to optimize the weights of the weak classifier while training using AdaBoost algorithm. SSA is one of the most popular and efficient meta-heuristic optimization algorithm with a smaller number of parameters. However, it suffers with slow convergence velocity and trapping in local optima (Sayed et al 2018). Therefore, this paper proposes an improved CSSA with AdaBoost to boost the performance and also to avoid falling into local optima. The logistic chaotic map provides high stability compared with tradition SSA (Sayed et al 2018). The logistic chaotic map with Adaboost provides an optimal result in handle imbalanced dataset. The framework of our proposed approach is presented in Figure 1.





Fig. 1: Framework for the proposed approach

4 Performance Evaluation

Our experiments were performed on a personal computer with an Intel (R) Core i5 processor and 8 GB of memory. With various classification algorithms using the Matlab tool, we have implemented our proposed algorithm. To test the performance of our proposed model, we have conducted three key experiments. The first experiment aims to examine whether the performance of our model can be enhanced by combining feature selection with boosting. Our model compares single ensemble models such as C4.5, CSSA-C4.5 and AdaBoost-C4.5 (Ahmed et al 2018; Zhang and Chen 2017) with our model. The second experiment was conducted to compare the performance of SSA and logistic chaotic SSA algorithm on 15 datasets. Finally, the proposed model has been compared with three state-of-the-art ensemble methods used to deal with imbalance class distribution problems.

4.1 Data Set

The proposed algorithm is evaluated using 15 datasets with different imbalance ratios (IR), obtained from the Keel repository¹. The imbalance ratio is defined as the proportion of the number of instances in the majority class to the number of instances in the minority one.

Table 1 shows the details of the imbalanced datasets with number of features and imbalance ratio.

Dataset	No. of features	No. of Samples	IR
Breast Cancer	9	286	2.36
Elico	7	220	1.86
Glass	9	214	1.82
Harberman	3	306	2.78
Iris	4	150	2.00
Pageblock	10	5472	8.79
Parkinsons	22	195	3.00
Splice	60	3190	1.20
Segment	19	2308	6.02
Thyroid	5	215	5.14
Vehicle	18	846	2.88
Wine	13	178	1.20
Yeast	8	1484	2.46
Balance	4	625	1.60
Zoo	16	101	1.10

Table 1: Details of the imbalanced datasets used in the experiments

¹ https://sci2s.ugr.es/keel/imbalanced.php

4.2 Measures

Table 2 illustrates the confusion matrix for binary class problems. Its entries are TP (True Positive), TN (True Negative), FP (False Positive), FN (False Negative).

In our experiment, positive instance refer to minority class and negative instance refer to majority class. The confusion matrix provides information about the actual and predicted values after classification. However, the classifier performance is evaluated based on the confusion matrix. The confusion matrix comprises four entries represented as

1. True Positive (TP): It refers to the number of positive samples that a classifier correctly predicts as positive.

2. True Negative (TN) It denotes the number of negative samples that a classifier correctly classifies as negative.

3.False Positive (FP) often referred to as false alarm; defined as the number of negative samples wrongly identified by a classifier as positive.

4.False Negative (FN) Sometimes referred to as miss; is calculated as the amount of positive samples that a classifier wrongly assigns as negative.

Table 2: Confusion Matrix

		Predicted			
		Positive	Negative		
Actual	Positive	TP	FN		
Actual	Negative	FP	TN		

4.3 Results and analysis

Three key experiments were performed to test the proposed model, as described in the previous section. The first experiment aims to examine whether the performance of our model can be enhanced by combining feature selection with boosting. Our model is compared with single ensemble models such as C4.5, CSSA-C4.5 and AdaBoost-C4.5. The second experiment was performed on 15 datasets to compare the SSA performance with logistic chaotic SSA algorithm. Finally , the proposed model was compared with three other state-of-the-art ensemble algorithms used to deal with problems of class imbalance.

For the classification of the training data set, the standard AdaBoost algorithm is used. As the number of weak classifiers increases, AUC output continues to improve, but the improve in AUC slows down dramatically when the number of weak classifiers exceeds 10. The AUC can hardly strengthen by further increasing the number of poor classifiers. Thus, in this experiment, the number of weak classifiers was set at 10. Figure 2 shows the graph growth trend of AUC (represented as y-axis) with respect to increase in the weak classifier (represented as x-axis).



Fig. 2: Number of Classifiers

Table 3: Comparison of different metrics obtained with respect to C4.5, CSSA usingC4.5, AdaBoost and CSSA-Adaboost

Datacate	Accuracy				G-Mean			
Datasets	C4.5	Adabaast	CSSA	CSSA-	C4.5	Adabaast	CSSA	CSSA-
	04.5	Adaboost	Using C4.5	Adaboost	04.5	Adaboost	Using C4.5	Adaboost
Breast Cancer	0.9247	0.9444	0.9367	0.9632	0.9435	0.9923	0.9577	0.9710
Elico	0.8322	0.8543	0.8334	0.8675	0.8532	0.9232	0.8723	0.8506
Glass	0.6577	0.9773	0.6891	0.9986	0.6007	0.9513	0.6268	0.9917
Harberman	0.5254	0.5747	0.5548	0.5812	0.5103	0.5166	0.5132	0.5811
Iris	0.9871	0.9921	0.9893	0.9934	0.9771	0.9613	0.9576	0.9712
Pageblock	0.9317	0.9989	0.9476	0.9954	0.9298	0.9998	0.9368	0.9878
Parkinsons	0.8522	0.8732	0.8673	0.9001	0.8375	0.8691	0.8723	0.8932
splice	0.5906	0.7388	0.6588	0.8111	0.3910	0.6198	0.4959	0.7348
Segment	0.9192	0.9199	0.9032	0.9942	0.9012	0.9199	0.9032	0.9942
Thyroid	0.9176	0.9786	0.9445	0.9876	0.5943	0.6007	0.8310	0.9262
Vehicle	0.7399	0.7812	0.7424	0.7912	0.7399	0.7911	0.7424	0.7912
wine	0.7636	0.9223	0.7812	0.9536	0.8135	0.8140	0.8135	0.8141
Yeast	0.5567	0.7212	0.559	0.7285	0.5610	0.6539	0.5610	0.6614
balance	0.5877	0.7422	0.7689	0.9566	0.347	0.849	0.7488	0.9891
Z00	0.7706	0.9312	0.8314	1.0000	0.7567	0.9998	0.8479	1.0000

In the first experiment, we investigated whether our model's performance can be enhanced by combining feature selection with boosting. Our model is compared with single ensemble models such as the C4.5 classifier, CSSA using the C4.5 decision tree classifier, and AdaBoost. Secondly, on 15 datasets, we aim to measure the performance of the SSA and logistic chaotic SSA algorithm. Finally, our proposed approach was compared with three state-of-the-art ensemble algorithms used to address the class imbalance problem.

In the first experiment, we investigated whether our model's performance can be enhanced by combining feature selection with boosting. Our model is compared with single ensemble models such as the C4.5 classifier, CSSA using the C4.5 classifier, and AdaBoost. We performed experiments to compare the CSSA-Adaboost with a

Datacate	F-Measure				AUC			
Datasets	C4.5	Adaboost	CSSA	CSSA-	C4.5	Adaboost	CSSA	CSSA-
	04.5	Auaboost	Using C4.5	Adaboost	04.5	Auaboost	Using C4.5	Adaboost
Breast Cancer	0.9613	0.9425	0.9923	0.9577	0.9632	0.9435	0.9923	0.9577
Elico	0.9042	0.9898	0.9180	0.9943	0.7398	0.8858	0.8137	0.8996
Glass	0.9042	0.9898	0.9180	0.9943	0.7896	0.8345	0.6268	0.9917
Harberman	0.5812	0.5103	0.5166	0.5132	0.5812	0.5103	0.5166	0.5132
Iris	0.9934	0.9771	0.9613	0.9576	0.9934	0.9771	0.9613	0.9576
Pageblock	0.9954	0.9298	0.9998	0.9368	0.9954	0.9298	0.9998	0.9368
Parkinsons	0.9001	0.8375	0.8691	0.8723	0.9001	0.8375	0.8691	0.8723
splice	0.6482	0.7478	0.7867	0.7856	0.4867	0.6534	0.5565	0.6944
Segment	0.9942	0.9012	0.9199	0.9032	0.9942	0.9012	0.9199	0.9032
Thyroid	0.6889	0.7786	0.9356	0.9989	0.5443	0.5668	0.8154	0.9289
Vehicle	0.8989	0.8152	0.8999	0.8978	0.8342	0.8945	0.8976	0.9067
wine	0.8991	0.8325	0.8967	0.8979	0.8354	0.8954	0.8987	0.9084
Yeast	0.7127	0.8856	0.7398	0.9388	0.4623	0.6122	0.5499	0.6654
balance	0.7294	0.8889	0.8704	0.9924	0.5226	0.5833	0.6332	0.7898
ZOO	0.9767	0.9999	0.9659	1.0000	0.8361	0.9822	0.9432	1.0000

Table 4: Comparison of different metrics obtained with respect to C4.5, CSSA using C4.5, AdaBoost and CSSA-Adaboost

simple C4.5 classifier, CSSA with a C4.5 classifier and AdaBoost to verify the effectiveness of the feature selection and Adaboost used in our model.

Different metrics such as precision, G-mean, F-score and AUC have been evaluated. The table 3 and 4 shows the precision, G-mean, F-Measure and AUC of our proposed CSSA-Adaboost approach and three comparative methods separately. Based on table 3 and 4, we observe that, CSSA-Adaboost obtained superior performance results when compared with three algorithms namely C4.5 classifier, CSSA with a C4.5 classifier and AdaBoost. Moreover, in terms of precision, the algorithm with feature selection performed better, which means that CSSA-Adaboost is better than Adaboost and C4.5. From our observation, these findings confirm that by eliminating redundant and unrelated features, CSSA optimization improves accuracy. We also note that in assessing the output of imbalanced data, both CSSA and Adaboost are successful in terms of accuracy and AUC. Figure 3 and 4 show the comparison of accuracy, G-mean, F-Measure and AUC of the four algorithm (Note: x-axis represents the measure and y-axis represents the various datasets).

Secondly, we have evaluated the performance of the SSA and logistic chaotic SSA algorithm on 15 datasets. The main objective of this experiment is to examine the AUC, F-Measure and G-Mean metric performance of SSA and logistic CSSA algorithms. Table 5 provides a comparison of the various metrics obtained with respect to SSA and Logistic Chaotic SSA. CSSA outperformed several datasets over SSA, as seen in Table 5.

Figure 5 shows the comparison of SSA and logistic CSSA algorithm in terms of AUC, G-mean and F-Measure. Finally, we compared our proposed model with other three state-of-the-art ensemble algorithms used to deal with class imbalance problems.

Finally, our proposed method was compared with three state-of-the-art ensemble algorithms used to deal with problems of class imbalance. In this comparison, we select three ensemble algorithms, i.e., SMOTEBoost (Chawla et al 2003), EasyEnsemble (Liu 2009), and RUSBoost (Dwiyanti et al 2016). These ensemble algorithms



Fig. 3: Comparison of Accuracy and F-Measure metrics obtained with respect to C4.5, CSSA using C4.5, AdaBoost and CSSA-Adaboost

are commonly used for comparison in the literature (Galar et al 2013; López et al 2013; Dwiyanti et al 2016). All these methods are based on sampling and boosting techniques, in which SMOTEBoost uses SMOTE for minority class over-sampling, while EasyEnsemble and RUSBoost use random majority class under-sampling techniques. All of these algorithms choose various base classifiers (we prefer C4.5 as the base classifier), such as SVM, Ripper, CART, etc. In terms of AUC, Table 6 presents the performance of CSSA-Adaboost along with the three algorithms. Figure 6 shows the comparison of CSSA-Adaboost and three state-of-the-art algorithms in terms of AUC.(Note: x-axis represents the measure and y-axis represents the various datasets).





Fig. 4: Comparison of G-Mean and AUC metrics obtained with respect to C4.5, CSSA using C4.5, AdaBoost and CSSA-Adaboost

Results indicate that all four algorithms are equivalent, as all of them have obtained the best results relative to the others on different datasets. Overall, EasyEnsemble and CSSA-AdaBoost are marginally better than the other algorithms, whereas imbalanced datasets with different IRs seem better handled with CSSA-Adaboost.

Conclusions

Swarm algorithms are commonly used to solve complex problems with optimization. In this paper, to deal with class imbalance issues, a novel chaotic salp swarm algo-







Fig. 5: Performance Comparison Using F-Measure, G-Mean and AUC Using SSA and Logistic Chaotic SSA

	AUC		F-	Measure	G-Mean		
Dataset	SSA	Logistic Chaotic SSA	SSA	Logistic Chaotic SSA	SSA	Logistic Chaotic SSA	
Breast Cancer	0.9550	0.9577	0.9550	0.9577	0.9550	0.9710	
Elico	0.8137	0.8996	0.8416	0.9943	0.8416	0.8506	
Glass	0.6268	0.9917	0.7917	0.9943	0.7917	0.9917	
Harberman	0.5166	0.5132	0.5897	0.5132	0.5897	0.5811	
Iris	0.9613	0.9576	0.9712	0.9576	0.9712	0.9712	
Pageblock	0.9998	0.9368	0.9978	0.9368	0.9978	0.9878	
Parkinsons	0.8429	0.8723	0.8732	0.8723	0.8732	0.8932	
splice	0.5565	0.6944	0.7348	0.7856	0.7348	0.7348	
Segment	0.9199	0.9032	0.9942	0.9032	0.9942	0.9942	
Thyroid	0.8154	0.9289	0.9262	0.9989	0.9262	0.9262	
Vehicle	0.7352	0.9067	0.7352	0.8978	0.7352	0.7912	
wine	0.9050	0.9084	0.955	0.8979	0.955	0.8141	
Yeast	0.5499	0.6654	0.6514	0.9388	0.6514	0.6614	
balance	0.6332	0.7898	0.9791	0.9924	0.9791	0.9891	
Z00	0.9432	1.0000	1.0000	1.0000	1.0000	1.0000	

Table 5: Comparison of different metrics obtained with respect to SSA and Logistic Chaotic SSA

Table 6: AUC Performance Comparison among State-of-the-art Algorithms with our Proposed Method

	AUC					
Dataset	SMOTEBoost	EasyEnsemble	RUSSMOTE	CSSA-Adaboost		
Breast Cancer	0.9577	0.9277	0.9177	0.9577		
Elico	0.785	0.864	0.8124	0.8996		
Glass	0.759	0.986	0.9816	0.9917		
Harberman	0.9134	0.9267	0.9367	0.5132		
Iris	0.5142	0.5212	0.5256	0.9576		
Pageblock	0.828	0.878	0.8128	0.9368		
Parkinsons	0.9672	0.9634	0.9734	0.8723		
splice	0.616	0.623	0.623	0.6944		
Segment	0.9023	0.612	0.612	0.9032		
Thyroid	0.813	1.000	1.000	0.9289		
Vehicle	1.000	1.000	1.000	0.9067		
wine	0.916	0.853	0.853	0.9084		
Yeast	0.674	0.654	0.654	0.6654		
balance	0.679	0.946	0.946	0.7898		
Z00	1.000	0.771	0.771	1.0000		

rithm with Adaboost technique (CSSA-Adaboost) is suggested. In this study, logistic chaos mapping along with the adaboost algorithm was used to improve classifier performance. To select the most discriminating features using SSA, the suggested CSSA-Adaboost algorithm was used. As a fitness function, we used AUC, making the CSSA-Adaboost classifier focus more on the accuracy with the minority class. Our proposed approach has been compared with state-of-the-art algorithms and the results indicate superior performance.

An Efficient CSSA based Ensemble Approach



Fig. 6: Performance Comparison of CSSA-Adaboost with three state-of-the-art algorithms in terms of AUC

In the future, multiple strategies such as pre-processing and cost sensitive method will be explored with CSSA-Adaboost for learning binary and multi-class imbalance classification. In addition, the implications of our framework in the perspective of explainable machine learning will be studied, and a modification that will favor the interpretation of classification decisions will be explored.

Compliance with Ethical Standards

Ethical approval: This article does not contain any studies with human participants or animals performed by any of the authors.

Conflict of Interest: Rekha G declares that she has no conflict of interest. Krishna Reddy V declares that he has no conflict of interest. Chandrashekar Jatoth declares that he has no conflict of interest. Ugo Fiore declares that he has no conflict of interest.

References

- Ahmed S, Mafarja M, Faris H, Aljarah I (2018) Feature selection using salp swarm algorithm with chaos. In: Proceedings of the 2nd International Conference on Intelligent Systems, Metaheuristics & Swarm Intelligence, ACM, pp 65–69
- Al-Ani A (2005) Feature subset selection using ant colony optimization. International journal of computational intelligence
- Amarendra C, Reddy KH (2019) Pso algorithm support switching pulse sequence isvm for six-phase matrix converter-fed drives. In: Smart Intelligent Computing and Applications, Springer, pp 559–569
- Bewoor LA, Chandra Prakash V, Sapkal SU (2017) Evolutionary hybrid particle swarm optimization algorithm for solving np-hard no-wait flow shop scheduling problems. Algorithms 10(4):121

- Cao P, Li B, Zhao D, Zaiane O (2013) A novel cost sensitive neural network ensemble for multiclass imbalance data learning. In: The 2013 International Joint Conference on Neural Networks (IJCNN), IEEE, pp 1–8
- Chawla NV, Bowyer KW, Hall LO, Kegelmeyer WP (2002) Smote: synthetic minority over-sampling technique. Journal of artificial intelligence research 16:321–357
- Chawla NV, Lazarevic A, Hall LO, Bowyer KW (2003) Smoteboost: Improving prediction of the minority class in boosting. In: European conference on principles of data mining and knowledge discovery, Springer, pp 107–119
- Chung D, Kim H (2015) Accurate ensemble pruning with pl-bagging. Computational Statistics & Data Analysis 83:1–13
- Di Martino M, Fernández A, Iturralde P, Lecumberry F (2013) Novel classifier scheme for imbalanced problems. Pattern Recognition Letters 34(10):1146–1151
- Dou P, Chen Y (2017) Remote sensing imagery classification using adaboost with a weight vector (wv adaboost). Remote Sensing Letters 8(8):733–742
- Dwiyanti E, Ardiyanti A, et al (2016) Handling imbalanced data in churn prediction using rusboost and feature selection (case study: Pt. telekomunikasi indonesia regional 7). In: International Conference on Soft Computing and Data Mining, Springer, pp 376–385
- Emary E, Zawbaa HM, Hassanien AE (2016) Binary ant lion approaches for feature selection. Neurocomputing 213:54-65
- Fawcett T (2006) An introduction to roc analysis. Pattern recognition letters 27(8):861-874
- Fiore U (2019) Minority oversampling based on the attraction-repulsion weber problem. Concurrency and Computation: Practice and Experience p e5601
- Fiore U, De Santis A, Perla F, Zanetti P, Palmieri F (2019) Using generative adversarial networks for improving classification effectiveness in credit card fraud detection. Information Sciences 479:448– 455
- Galar M, Fernandez A, Barrenechea E, Bustince H, Herrera F (2011) A review on ensembles for the class imbalance problem: bagging-, boosting-, and hybrid-based approaches. IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews) 42(4):463–484
- Galar M, Fernández A, Barrenechea E, Herrera F (2013) Eusboost: Enhancing ensembles for highly imbalanced data-sets by evolutionary undersampling. Pattern Recognition 46(12):3460–3471
- Gao M, Hong X, Chen S, Harris CJ (2011) A combined smote and pso based rbf classifier for two-class imbalanced problems. Neurocomputing 74(17):3456–3466
- Haixiang G, Yijing L, Yanan L, Xiao L, Jinling L (2016) Bpso-adaboost-knn ensemble learning algorithm for multi-class imbalanced data classification. Engineering Applications of Artificial Intelligence 49:176–193
- Joshi MV, Kumar V, Agarwal RC (2001) Evaluating boosting algorithms to classify rare classes: Comparison and improvements. In: Proceedings 2001 IEEE International Conference on Data Mining, IEEE, pp 257–264
- Kennedy J, Eberhart RC (1997) A discrete binary version of the particle swarm algorithm. In: Proceedings of the IEEE International conference on systems, man, and cybernetics. Computational cybernetics and simulation, IEEE, vol 5, pp 4104–4108
- Li K, Xie P, Liu W, Zha J (2017) An ensemble evolve algorithm for imbalanced data. Journal of Computational and Theoretical Nanoscience 14(9):4624–4629
- Li K, Zhou G, Zhai J, Li F, Shao M (2019) Improved pso_adaboost ensemble algorithm for imbalanced data. Sensors 19(6):1476
- Li L, Wang C, Li W, Chen J (2018) Hyperspectral image classification by adaboost weighted composite kernel extreme learning machines. Neurocomputing 275:1725–1733
- Li Y, Wang S, Tian Q, Ding X (2015) A boosting approach to exploit instance correlations for multiinstance classification. IEEE transactions on neural networks and learning systems 27(12):2740–2747
- Liu TY (2009) Easyensemble and feature selection for imbalance data sets. In: 2009 International Joint Conference on Bioinformatics, Systems Biology and Intelligent Computing, IEEE, pp 517–520
- López V, Fernández A, Del Jesus MJ, Herrera F (2012) Cost sensitive and preprocessing for classification with imbalanced data-sets: Similar behaviour and potential hybridizations. In: ICPRAM (2), pp 98–107
- López V, Fernández A, García S, Palade V, Herrera F (2013) An insight into classification with imbalanced data: Empirical results and current trends on using data intrinsic characteristics. Information sciences 250:113–141

- Maldonado S, Weber R, Famili F (2014) Feature selection for high-dimensional class-imbalanced data sets using support vector machines. Information Sciences 286:228–246
- Menardi G, Torelli N (2014) Training and assessing classification rules with imbalanced data. Data Mining and Knowledge Discovery 28(1):92–122
- Mirjalili S, Gandomi AH, Mirjalili SZ, Saremi S, Faris H, Mirjalili SM (2017) Salp swarm algorithm: A bio-inspired optimizer for engineering design problems. Advances in Engineering Software 114:163– 191
- Namassivaya N, Pal S, Ratnam DV (2019) Modelling of fpga-particle swarm optimized gnss receiver for satellite applications. Wireless Personal Communications 106(2):879–895
- Nikhath AK, Subrahmanyam K (2019) Feature selection, optimization and clustering strategies of text documents. International Journal of Electrical & Computer Engineering (2088-8708) 9(2)
- Ogiela L, Ogiela MR (2020) Cognitive security paradigm for cloud computing applications. Concurrency and Computation: Practice and Experience 32(8):e5316
- Qiaojin G, Libin L, Ning L (2008) Novel modified adaboost algorithm for imbalanced data classification. Computer Engineering and Applications 44(21):217–221
- Ramentol E, Caballero Y, Bello R, Herrera F (2012) Smote-rsb*: a hybrid preprocessing approach based on oversampling and undersampling for high imbalanced data-sets using smote and rough sets theory. Knowledge and information systems 33(2):245–265
- Rekha G, Reddy VK (2018) A novel approach for handling outliers in imbalance data. International Journal of Engineering & Technology 7(3.1):1–5
- Sayed GI, Khoriba G, Haggag MH (2018) A novel chaotic salp swarm algorithm for global optimization and feature selection. Applied Intelligence 48(10):3462–3481
- Sayed GI, Tharwat A, Hassanien AE (2019) Chaotic dragonfly algorithm: an improved metaheuristic algorithm for feature selection. Applied Intelligence 49(1):188–205
- Schiezaro M, Pedrini H (2013) Data feature selection based on artificial bee colony algorithm. EURASIP Journal on Image and Video Processing 2013(1):47
- Searle SR, Searle S (1987) Linear models for unbalanced data, vol 1987. Wiley New York
- Sultanpure KA, Reddy LSS (2018) Job scheduling for energy efficiency using artificial bee colony through virtualization. International Journal of Intelligent Engineering and Systems 11(3):138–148
- Sun B, Chen H, Wang J, Xie H (2018) Evolutionary under-sampling based bagging ensemble method for imbalanced data classification. Frontiers of Computer Science 12(2):331–350
- Thai-Nghe N, Gantner Z, Schmidt-Thieme L (2010) Cost-sensitive learning methods for imbalanced data. In: The 2010 International joint conference on neural networks (IJCNN), IEEE, pp 1–8
- Thanathamathee P, Lursinsap C (2013) Handling imbalanced data sets with synthetic boundary data generation using bootstrap re-sampling and adaboost techniques. Pattern Recognition Letters 34(12):1339– 1347
- Thirugnanasambandam K, Prakash S, Subramanian V, Pothula S, Thirumal V (2019) Reinforced cuckoo search algorithm-based multimodal optimization. Applied Intelligence 49(6):2059–2083
- Verikas A, Kalsyte Z, Bacauskiene M, Gelzinis A (2010) Hybrid and ensemble-based soft computing techniques in bankruptcy prediction: a survey. Soft Computing 14(9):995–1010
- Viola P, Jones M (2002) Fast and robust classification using asymmetric adaboost and a detector cascade. In: Advances in neural information processing systems, pp 1311–1318
- Wang K, Wang Y, Zhao Q, Meng D, Liao X, Xu Z (2019) Splboost: An improved robust boosting algorithm based on self-paced learning. IEEE Transactions on Cybernetics
- Weiss Y, Elovici Y, Rokach L (2013) The cash algorithm-cost-sensitive attribute selection using histograms. Information Sciences 222:247–268
- Xinwu Y, Zhuang M, Shun Y (2016) Multi-class adaboost algorithm based on the adjusted weak classifier. Journal of Electronics & Information Technology 38(2):373–380
- Xue B, Zhang M, Browne WN (2012) Particle swarm optimization for feature selection in classification: A multi-objective approach. IEEE transactions on cybernetics 43(6):1656–1671
- Yijing L, Haixiang G, Xiao L, Yanan L, Jinling L (2016) Adapted ensemble classification algorithm based on multiple classifier system and feature selection for classifying multi-class imbalanced data. Knowledge-Based Systems 94:88–104
- Yin L, Ge Y, Xiao K, Wang X, Quan X (2013) Feature selection for high-dimensional imbalanced data. Neurocomputing 105:3–11
- Zhai J, Zhang S, Zhang M, Liu X (2018) Fuzzy integral-based elm ensemble for imbalanced big data classification. Soft Computing 22(11):3519–3531

Zhang C, Chen Y (2017) Improved piecewise nonlinear combinatorial adaboost algorithm based on noise self-detection. Comput Eng 43:163–168