Title Frame

Progress Seminar

Multi objective feature selection using Non dominated Sorting based evolutionary algorithm

Ved Prakash
Roll No. 1903016
Department of Computer Science & Engineering
Indian Institute of Information Technology Guwahati

August 28, 2024

Agenda

Introduction: Feature Subset Selection

Feature Subset Selection Problem

To get optimal subset of relevant features which represents original feature set with increased classification accuracy

- Feature Selection is one of the important and critical data pre-processing step for building the statistical or machine learning models
- Feature Selection Algorithms reduces the dimensionality of the data by removing the redundant and irrelevant features
- It also results into lesser learning time, improved model interpretability, reduced overfitting and improved classification performance

Steps of Basic Evolutionary Algorithm

Initial Population:

	Credit History	Purpose of Credit	Age	Average Balance in Saving Account	present resident since - years	Nature of job
Individual 1	1	1	1	0	0	0
Individual 2	0	0	0	0	0	1
Individual 3	1	1	0	0	0	0
Individual 4	0	1	0	1	1	0
Individual 5	0	0	0	1	1	1
Individual 6	0	0	1	1	1	0

Fitness Calculation, Selection and Cross-Over

	Credit History	Purpose of Credit	Age	Average Balance in Saving Account	present resident since - years	Nature of job
Individual 1	1	1	1	0	0	0
Individual 4	0	1	0	1	1	0

	Credit History	Purpose of Credit	Age	Average Balance in Saving Account	present resident since - years	Nature of job
Offspring 1	1	1	1	1	1	0
Offspring 2	0	1	0	0	0	0

Mutation

	Credit History	Purpose of Credit	Age	Average Balance in Saving Account	present resident since - years	Nature of job
Offspring 1	1	1	0	1	1	0

Problem Statement

Binary Representation of Solution

In this work, binary strings have been used to represent a solution. A solution p can be represented as,

$$p = (p_1, p_2, \ldots, p_m) ; p_j \in \{0, 1\} \quad (1)$$

In Equation (1), $s_j = 0$ represents that j^{th} feature is not included in the solution s and $s_j = 1$ indicates that it has been included.

Problem Statement

Multi Objective Feature Selection Problem Statement

A multiobjective feature selection problem can be formulated as,

$$\underset{(err(s),|s|)}{\text{minimize}}\ p = (p_1, p_2, \ldots, p_m)\,; p_j \in \{0,1\}; j = 1, 2, \ldots, m \quad (2)$$

$|s|$ is number of features in solution s.

MOFS-BDE Algorithm

- The MOFS-BDE algorithm was prposed by Zhang *et. al.* [5] in their work "Binary differential evolution with self-learning for multi-objective feature selection"
- The algorithm uses the mutation strategy such that, it selects the best one among the three random vectors as the base vector, and employs the difference between the remaining two vectors as a mutation probability to be used on the base vector to generate a mutation vector for the next crossover operator. It is illustrated in Equations (3), (4).

$$C_i = \begin{cases} \sigma & X_{best}(t) \prec X_i(t) \\ min(1, F.(X_{r1}(t) \oplus X_{r2}(t)) + \sigma) & \text{Otherwise} \end{cases} \quad (3)$$

$$v_{i,j}t = \begin{cases} X_{best}(t) & C_{i,j} < rand \\ 1 - X_{best}(t) & \text{Otherwise} \end{cases} \quad (4)$$

Analyzing the mutation strategy of MOFS-BDE algorithm on some of the broadly possible scenarios

- If both bits are same then $X_{r1}(t) \oplus X_{r2}(t)$ will become zero. then (3) will become

$$C_i = \begin{cases} \sigma & \text{if } X_{best}(t) \prec X_i(t) \\ min(1, F.(0) + \sigma) & \text{Otherwise} \end{cases}$$

i.e.,

$$C_i = \begin{cases} \sigma & \text{if } X_{best}(t) \prec X_i(t) \\ min(1, \sigma) & \text{Otherwise} \end{cases}$$

Analyzing the mutation strategy of MOFS-BDE algorithm on some of the broadly possible scenarios

Since, σ is a very small turbulance coefficinet and much much less than 1 so $min(1, \sigma) = \sigma$

$$C_i = \begin{cases} \sigma & \text{if } X_{best}(t) \prec X_i(t) \\ \sigma & \text{Otherwise} \end{cases}$$

So, if both bits are same then c_i will be σ. In this case, the condition $C_{i,j} < rand$ will most of the time be true. as random number generator theoretically generates values between 0 and 1 with equal distribution. and since σ is closer to zero then it is more chance that, rand will be greater than σ and hence more chance that $v_{i,j}t$ in Equation (4) will have same bit value as $X_{best}(t)$.

Approaches for Mutation

- **If both bits are different** If both bits are different then $X_{r1}(t) \oplus X_{r2}(t)$ will become 1. then (3) will become

$$C_i = \begin{cases} \sigma & \text{if } X_{best}(t) \prec X_i(t) \\ min(1, F.(1) + \sigma) & \text{Otherwise} \end{cases}$$

i.e.,

$$C_i = \begin{cases} \sigma & \text{if } X_{best}(t) \prec X_i(t) \\ min(1, F + \sigma) & \text{Otherwise} \end{cases}$$

So, if both bits are different then C_i depends on scaling factor F and turbulance co-efficient σ. Higher the value of F and σ (as σ is very small turbulance co-efficient so it mainly depends on F), higher will be the value of C_i. and hence more chance that $v_{i,j}t$ in Equation (4) will have bit value opposite to $X_{best}(t)$.

The issue with mutation approach of MOFS-BDE algorithm

- So, the issue with approach in Equation (3) are following
  - if $X_{best}(t) \prec X_i(t)$ hen it is very very high chance that the mutant vector will borrow bit from the $X_{best}t$.
  - otherwise
    - if X_{r1} and X_{r2} have same bits then it is very very high chance that the mutant vector will borrow bit from the $X_{best}t$.
    - if X_{r1} and X_{r2} have different bits and learning rate F is small, then also it is very high chance that, the bits of mutant vector will be borrowed from $X_{best}t$.
    - Hence the true spirit of differential evolution, i..e, to utilize the difference in population is not being utilized in these scenarios and hence it might impact the global exploration capability of the algorithm.

Modified Mutation Strategy

- we have proposed a weighted mutation strategy specially suitable for the **binary** differential evolution algorithm. The proposed mutation strategy is mentioned in Equation

$$\hat{X}_i = \underbrace{\mu(floor((X_{r1} + X_{r2} + X_{r3})/1.5)}_{initial} + \underbrace{(1-\mu)(X_{best\_among\_3}.int((X_{r2} \odot X_{r3})) + int(\neg X_{best\_among\_3}).int((X_{r2} \otimes X_{r3}))}_{later} \quad (5)$$

$$\hat{X}_i(t) = \begin{cases} 0 & \text{if } X_i(t) < 0.5 \\ 1 & \text{Otherwise} \end{cases} \quad (6)$$

- $\mu = \exp(-\sqrt{g})$ and μ is a monotonically decreasing function with increase in g, *i.e.*, generation.

Modified Mutation Strategy

- **During Initial Generation:** During initial generation, when μ will be closer to 1 then $(1 - \mu)$ will be closer to 0 and hence second part of the equation will have negligible impact on the outcome and hence it can be said that, during initial generation, outcome will be guided by first part of the equation, *i.e.*, $\mu(floor((X_{r1} + X_{r2} + X_{r3})/1.5)$.
  - If none of the feature subset X_{r1}, X_{r2}, X_{r3} consist this feature then chance is less that mutation vector consist this feature.
  - But, if any two of X_{r1}, X_{r2}, X_{r3} consists this feature, then chances will be more that, during initial generation, mutation vector will also have this feature.
  - If all of the feature subset X_{r1}, X_{r2}, X_{r3} consists this feature then chances are even stronger that mutation vector consist this feature.

Modified Mutation Strategy

- **During Later Generations:** During later generations, when μ will be closer to 0 then $(1-\mu)$ will be closer to 1 and hence first part of the equation will have negligible impact on the outcome and hence it can be said that, during later generations, outcome will be guided by second part of the equation, $(1-\mu)(X_{best\_among\_3}.int((X_{r2} \odot X_{r3})) + int(\neg X_{best\_among\_3}).int((X_{r2} \otimes X_{r3}))$

$$\hat{X}_i = \begin{cases} (1-\mu)(X_{best\_among\_3}.int((X_{r2} \odot X_{r3}))) & \text{if } X_{\text{best\_among\_3}} = 1 \\ (1-\mu)\,(int(\neg X_{best\_among\_3})).int(X_{r2} \otimes X_{r3})) & \text{Otherwise} \end{cases}$$

Modified Mutation Strategy

$$\hat{X}_i = \begin{cases} (1-\mu)(int((X_{r2} \odot X_{r3}))) & \text{if } X_{\text{best\_among\_3}} = 1 \\ (1-\mu)\,(int(X_{r2} \otimes X_{r3})) & \text{Otherwise} \end{cases}$$

Since, μ will closing to zero during later generation, so $1-\mu$ will be closer to 1, hence,

$$\hat{X}_i = \begin{cases} X_{r2} \odot X_{r3} & \text{if } X_{\text{best\_among\_3}} = 1 \\ X_{r2} \otimes X_{r3} & \text{Otherwise} \end{cases}$$

Modified Mutation Strategy

- So, during later generations,
  - If $X_{\text{best\_among\_3}}$ consists this feature and any of the other two vectors selected for generating mutation vector consists this feature then chances are more that mutation vector will have this feature.
  - Also, if $X_{\text{best\_among\_3}}$ doesn't consists this feature but other two vectors selected for generating mutation vector consists this feature then there is chance that mutation vector will have this feature.
- Hence, it can be said that, the mutation operator purposed in Equation (6) is designed in such a way that, it tries to make balance between global exploration and local exploitation.

Proposed Algorithm

- We have proposed a 'Modified Mutation Strategy' to the algorithm

Table: Comparison of Existing and Modified Mutation Strategies

Aspect	Existing Mutation Strategy (MOFS-BDE)	Modified Mutation Strategy
Mutation Vector Generation	Uses the difference between remaining two vectors as a mutation probability to generate a mutation vector	Initial generation: Weighted sum of random vectors and hence guided by proportion of majority Later generation: Combination of best and remaining vectors. Hence, guided by best vector but also take care of global exploration.
Handling of Identical Bits	If bits are the same, mutation probability $C_i = min(1, \sigma)$, leading to $v_{i,j}t$ being equal to $X_{best}(t)$	During initial generation, if all vectors have the same feature, the mutation vector will likely have this feature During later generation, it is mostly guided by the best vector, but if best vector doesn't has that feature and other two vector has, then it explores that feature.
Handling of Different Bits	If bits are different, mutation probability $C_i = min(1, F + \sigma)$, leading to higher chances of $v_{i,j}t$ being opposite to $X_{best}(t)$	During the initial generation, it is mostly guided by majority and hence it will not decide just based upon x_best During later generations, if $X_{best\_among\_3}$ contains the feature, the mutation vector will likely have this feature; otherwise, it depends on the other two vectors
Exploitation vs. Exploration	Potentially limited global exploration, as the difference in population is not fully utilized	Balanced between global exploration and local exploitation, with an adaptive mutation strategy

Application on Banking Dataset for credit risk prediction

- Banks assess credit risk using KPIs like probability of default and loss given default.
- Credit risk models rely heavily on feature subset selection during data pre-processing.
- Most research in banking credit risk uses evolutionary computation focused on single-objective optimization, primarily model accuracy.
- This work formulates feature subset selection as a multi-objective optimization problem, minimizing both error rate and feature subset size.

Dataset description

- We have used four different credit-related datasets to compare existing and proposed mutation strategies.
- These datasets are German credit data [1], Australian Credit dataset [2], Taiwan Credit Dataset [4] and Credit Card Fraud Detection Dataset [3].
- The German credit dataset includes 1,000 observations and 20 attributes, each observation is labeled as either "good" or "bad" credit risk.
- Australian Credit dataset has 690 records and 15 columns.
- Taiwan Credit Dataset contains information related to credit card applications in Taiwan.
- Credit Card Fraud Detection Dataset contains transactions made by credit cards in September 2013 by European cardholders. This dataset presents transactions that occurred in two days, where it has 492 frauds out of $284,807$ transactions.

Experimental Setup

- We split the dataset into two parts, trained the model on one, and tested the performance on the other.
- We have used the multi-objective feature selection binary differential evolution (MOFS-BDE) algorithm for feature subset selection to train the model.
- The fitness function that we have used for the MOFS-BDE algorithm is, the XGBoost algorithm
- The two objectives of the MOFS-BDE algorithms are to minimize the error rate and to minimize the cardinality of the feature subset.
- We have compared the performance of the MOFS-BDE algorithm with the existing mutation strategy against our proposed mutation strategy.

Evaluation Matrics

- The metrics that we have used to compare the performance are minimum error by generation, minimum feature by generation, and hypervolume by generation.
- Hypervolume indicator is a performance metric that measures the quality of a non-dominated approximation set in multi-objective problems. Hypervolume is calculated by finding the volume of the space between a reference point and the Pareto set produced by a specific algorithm.

Evaluation Matrics

- Figure 1 represents the calculation of the hypervolume metric for a minimization problem with two objectives f_1 and f_2. X_1, X_2, and X_3 are solutions from the Pareto set of this problem. f^{ref} is reference point. The shaded area represents the hypervolume for this Pareto set. A reference point is required to calculate the hypervolume indicator value.

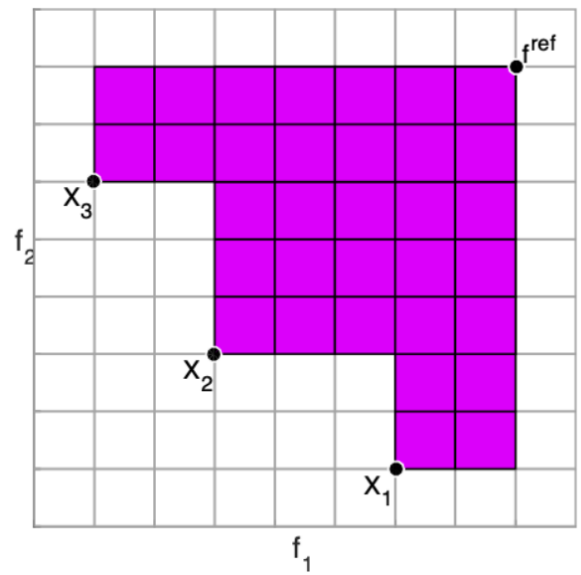
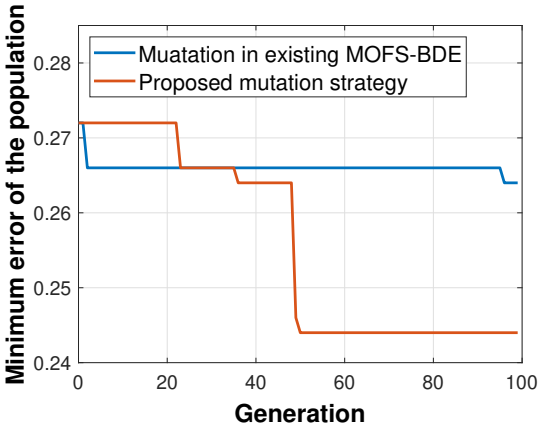
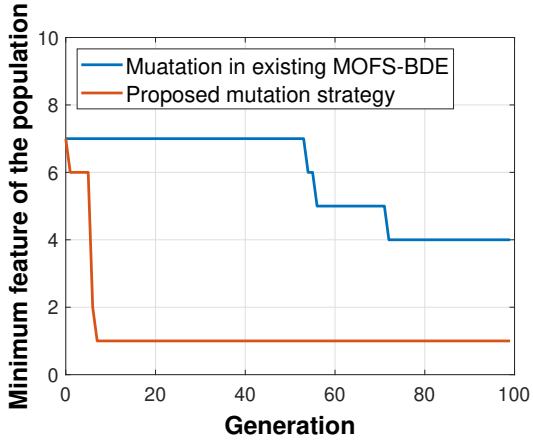


Figure: Hypervolume representation

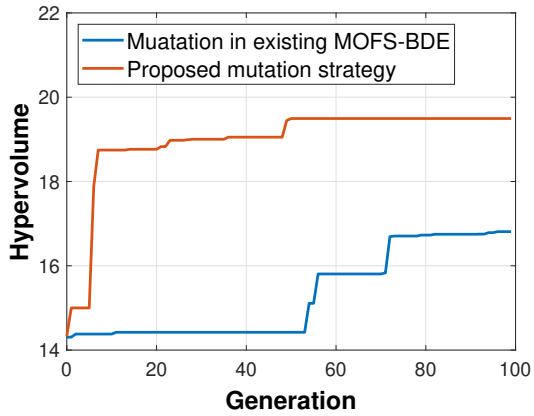
Experimental result on German Credit Dataset



(a) Minimum error by generation

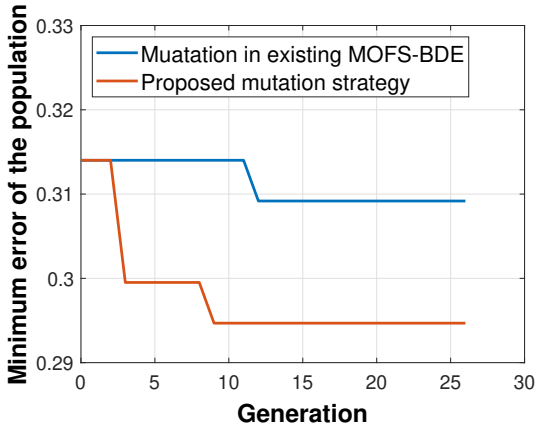


(b) Minimum feature by generation

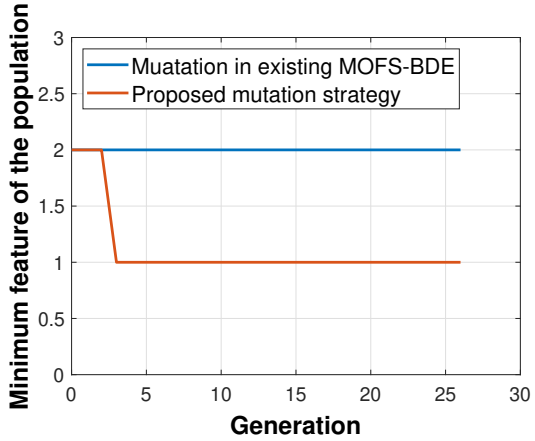


(c) Hypervolume by generation

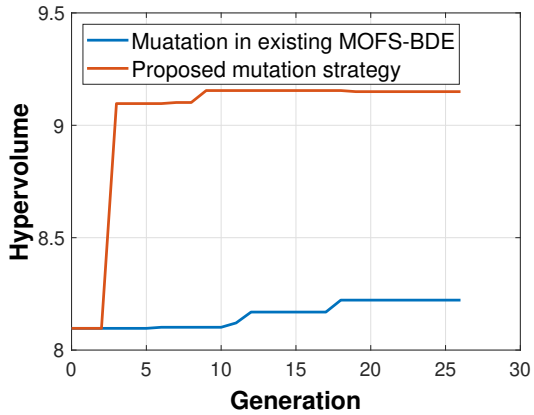
Experimental result on Australian Credit Dataset



(a) Minimum error by generation

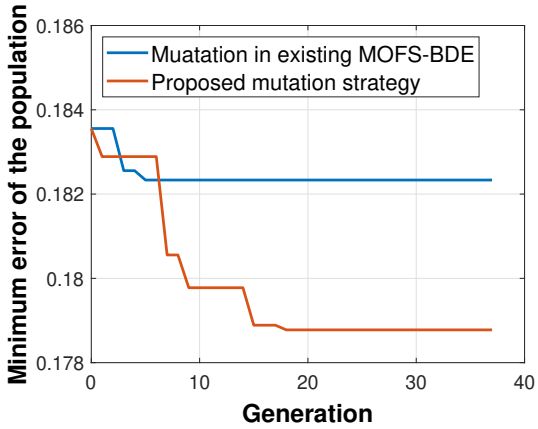


(b) Minimum feature by generation

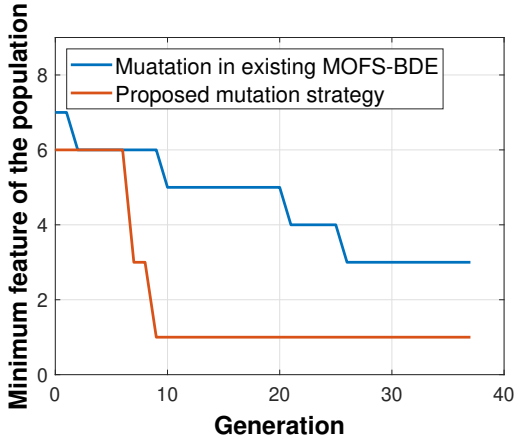


(c) Hypervolume by generation

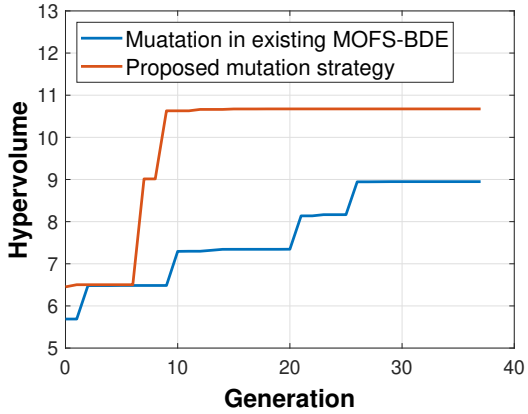
Experimental result on Australian Taiwan Dataset



(a) Minimum error by generation

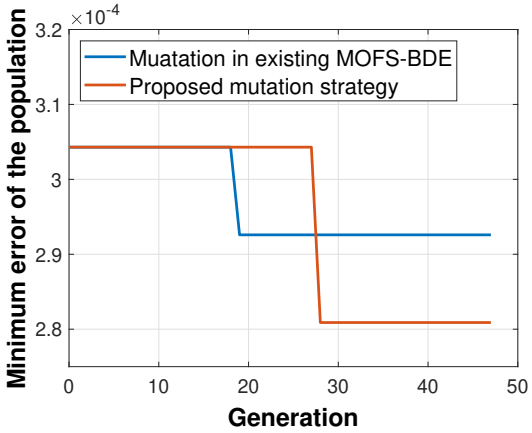


(b) Minimum feature by generation

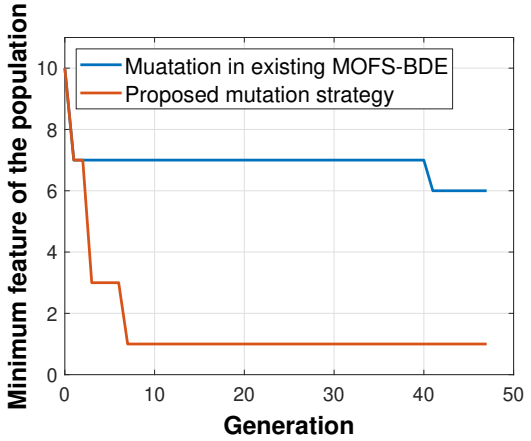


(c) Hypervolume by generation

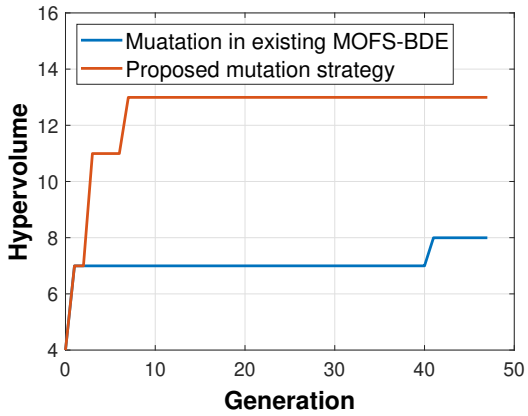
Experimental result on credit card fraud detection Dataset



(a) Minimum error by generation



(b) Minimum feature by generation



(c) Hypervolume by generation

Conclusion

- The results shows that, modified MOFS-BDE algorithm with our proposed mutation strategy performs better than the existing MOFS-BDE algorithm.
- The modified algorithm is not only minizing the error but it is also fastly reducing the feature alnogwith geneation and hence this algorithm will be efficient in datasets with more number of features.

Limitation

- The limitation of any evolutionary based algorithm including MOFS-BDE algorithm is that, they take more time to converge and are computationally slow.

Thank You !

Questions ?

References I

[1] Hans Hofmann.
Statlog (German Credit Data).
UCI Machine Learning Repository, 1994.
DOI: 10.24432/C5NC77.

[2] Ross Quinlan.
Statlog (Australian Credit Approval).
UCI Machine Learning Repository.
DOI: https://doi.org/10.24432/C59012.

[3] ULB and Andrea.
Credit Card Fraud Detection Dataset.
Kaggle Dataset, 2018.
https://www.kaggle.com/datasets/rtatman/188-million-us-wildfires.

[4] I-Cheng Yeh.
Default of Credit Card Clients.
UCI Machine Learning Repository, 2016.
DOI: https://doi.org/10.24432/C55S3H.

[5] Yong Zhang, Dun wei Gong, Xiao zhi Gao, Tian Tian, and Xiao yan Sun.
Binary Differential Evolution with Self-learning for Multi-objective Feature Selection.
Information Sciences, 507:67–85, 2020.