Hindawi
Security and Communication Networks
Volume 2022, Article ID 9784628, 1 page
https://doi.org/10.1155/2022/9784628

Retraction

Retracted: An Information Entropy Embedding Feature Selection Based on Genetic Algorithm

Security and Communication Networks

Received 14 November 2022; Accepted 14 November 2022; Published 23 November 2022

Security and Communication Networks has retracted the article titled "An Information Entropy Embedding Feature Selection Based on Genetic Algorithm" [1] due to concerns that the peer review process has been compromised.

Following an investigation conducted by the Hindawi Research Integrity team [2], significant concerns were identified with the peer reviewers assigned to this article; the investigation has concluded that the peer review process was compromised. We therefore can no longer trust the peer review process, and the article is being retracted with the agreement of the Chief Editor.

References

[1] Y. Wu, "An Information Entropy Embedding Feature Selection Based on Genetic Algorithm," *Security and Communication Networks*, vol. 2022, Article ID 7111034, 10 pages, 2022.

[2] L. Ferguson, "Advancing Research Integrity Collaboratively and with Vigour," 2022, https://www.hindawi.com/post/advancing-research-integrity-collaboratively-and-vigour/.

Hindawi
Security and Communication Networks
Volume 2022, Article ID 7111034, 10 pages
https://doi.org/10.1155/2022/7111034

Research Article

An Information Entropy Embedding Feature Selection Based on Genetic Algorithm

Yuzheng Wu

School of Mathematics, Sichuan University, Chengdu 610000, China

Correspondence should be addressed to Yuzheng Wu; 2018141471009@stu.scu.edu.cn

Received 5 March 2022; Accepted 13 April 2022; Published 25 May 2022

Academic Editor: Muhammad Arif

Feature selection is of vital importance to reduce information redundancy and deal with the invalidation of basic classification approaches for massive dataset and too many features. In order to improve the classification accuracy and decrease time complexity, an algorithm with intelligent optimization genetic algorithm and weight distribution based on information entropy is proposed, called EEGA. Information entropy of features is defined as the population labels in GA rather than the direct iteration of individual fitness. Experiments have been performed by using several standard databases with four fitness algorithms. Experimental results have proved that EEGA performs better based on the measure of accuracy. Furthermore, it can significantly reduce the required time when figuring out the better results.

1. Introduction

Owing to the pervasive use of Internet and rapid performance boost of information technology, people are exposed to an increasing amount of information in a day. Such huge amount of information is hard to handle, which may cause some trouble in information processing like the dimension disaster. Reducing the dimension of the samples can effectively solve the information redundancy but unreasonable dimensionality reduction will cause information loss that affects data analysis and accuracy of classification. Therefore, it is of vital importance to keep essential information with low-dimensionality, which is sort of technology called feature reduction [1].

Feature reduction could be divided in two categories, feature extraction and feature selection. Raw features can be transformed by feature extraction into a set of features with statistical significance or kernel, and feature selection means choosing the best feature subset from all feature sets. In recent years, researches of feature selection can be summarized into three algorithms, filter, wrapper, and embedder. Todorov [2] uses valid distance metric into Relief [3], a classic filtering feature selection algorithm. Peng et al. [4] propose a wrapped algorithm named FACO, which combines the ant colony optimization algorithm and feature selection. Rao et al. [5] apply artificial bee colony algorithm into decision tree (embedded one), achieving global optimization. Wang et al. [6] improve the AdaBoost approach based on a weighted feature selection in traditional filters, which obtains significant boost on classification accuracy. All above algorithms improve the traditional feature selection methods to increase the accuracy of classification; however, they did not take dynamic changes in weight distribution into consideration. What is more, a large number of redundant calculations increase the time complexity. Based on these issues, a weight distribution algorithm combining Information Entropy Theory and genetic algorithm is proposed for feature reduction.

Information entropy is defined as the average amount of information excluding redundancy, which is the quantified form of information. In 1948, Shannon proposed Information Entropy Theory inspired by thermodynamic entropy and mentioned that there is redundancy in any information and the size of the redundancy is related to the probability or uncertainty of the occurrence of each symbol in the information. So the weight among indicators could be measured by calculating the information entropy. Genetic algorithm (GA) is originated from computer simulation

studies of biological systems, which is a random global search and optimization method inspired by biological evolution mechanism in nature [7]. It is an efficient, parallel, global search algorithm, which can automatically acquire and accumulate knowledge about the search space during the search process and adaptively control the search process to obtain the best solution.

In this paper, EEGA (Entropy Embedding Genetic Algorithm for feature selection), a weight distribution model, is proposed, combining characteristics of filter and wrapper algorithms. Based on Information Entropy Theory, EEGA achieves calculation of weights and selection of features. GA is extra introduced for threshold optimization in EEGA, dynamically controlling the feature selection. Classification performance of EEGA is compared with traditional Random Forest and other wrapper algorithms. The comparison outcomes confirmed the effectiveness of our algorithm.

The other parts of the paper are arranged as follows. The main content of Section 2 is basic theory of Information Entropy Theory and genetic algorithm. Section 3 introduces the proposed EEGA algorithm. Section 4 is the performance comparison and the result analysis. Section 5 is the summary of the paper, and the future work is eventually left to Section 5, too.

2. Problem Analysis

2.1. Feature Selection. In classification questions about massive dataset or dataset with too many features, fast, accurate, and stable classification is the most concerned. Feature selection provides classification with a direct approach that is replacing the whole by some significant features with most information based on specific criterion. Through feature selection, feature dimensions and redundant features can be reduced, while important features can be retained, so that the learning and generalization ability can be both boosted.

The algorithms of feature selection are mainly divided into filter, wrapper, and embedder. The filter scores each feature according to correlation or other information criterion and then sets a threshold or the number of thresholds for selection, while the wrapper one is an iteration process about recursive feature elimination, whose main principle is deleting the features with poor weights, recalculating the features for the overall model in the prediction model with weighted features, and then repeating recursion continuously until expectation. Embedded selection is to integrate the feature selection process with the learner training process, so the feature selection is automatically performed during training process.

Wrapper has better classification effect and is more targeted for feature selection, while filter generally involves a noniterative computation, which can be much faster than continuous iterations. Hence, fast and accurate feature selection can be acquired by the combination of filter and wrapper algorithms. Therefore, the filter based on Information Entropy Theory and genetic algorithm, an efficient wrapper, are both introduced for feature selection.

2.2. Information Entropy. Entropy was defined by Clausius in 1854 as a measure of system chaos in thermodynamics. Later, entropy theory gradually expanded to other scientific fields. For example, information entropy is proposed by Shannon in the field of information and life entropy is proposed by Schrödinger in the life sciences. In information theory, the more chaotic something is, the greater the entropy is, and the greater the amount of information needed to determine something. Therefore, the uncertainty of things can be measured by information entropy. In other words, information entropy is an index to quantify the uncertainty of events. So the characteristics of information entropy could be utilized to achieve weight distribution.

For a feature, the greater the entropy is, the less information it contains (the greater the cost of entropy reduction), for the reason that its assigned weight is smaller. Firstly, uncertainty function g is defined as

$$\begin{aligned} g &= \log\left(\frac{1}{p}\right) \\ &= -\log p, \end{aligned} \tag{1}$$

where p is related probability. Then, the measurement of information entropy $H(G)$ can be realized by calculating the expectation of the uncertainty function:

$$\begin{aligned} H(G) &= E(G) \\ &= \sum_{g \in G} g \cdot p \\ &= -\sum p \log p, \end{aligned} \tag{2}$$

where G is the set of all possible events.

2.3. Genetic Algorithm. The genetic algorithm proposed by Holland adopts the binary coding method and realizes iterative optimization by simulating the natural selection mechanism of "survival of the fittest." The algorithm flow of GA is shown in Figure 1.

From the perspective of biology, a population in nature is composed of many individuals and each individual has its own unique gene. It is true that individuals pass their genes to their offspring through mating and reproduction, and in this process, the genes of the parents will cross over. Besides, considering time iteration, these genes may also mutate. According to the "survival of the fittest" theory, the natural environment will weed out those unfit individuals, which makes their genes disappear. Under these conditions, after a sufficiently long period of replacement, what will be left will be the individuals best suited to the environment, so that we can also obtain the genetic makeup that is the optimal choice (assuming environmental conditions do not change).

From a mathematical point of view, the gene carried by an individual can be regarded as a feasible solution to the problem. The crossover and mutation of genes can be regarded as the change of the problem solution. And the environment can naturally be understood as fitness function. Under the same fitness function, genes closer to the optimal

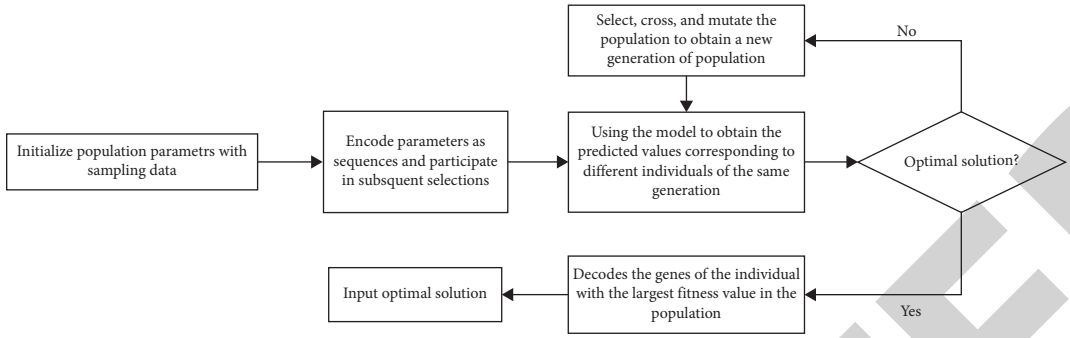

Figure 1: Genetic algorithm process.

Table 1: Definition of terms related to genetic algorithm.

Terminology	Explanation
Gene genotype phenotype evolution	Basic genetic unit in biology, binary in GA Internal representation of trait chromosomes, binary sequence in GA External manifestations of trait chromosomes, parameters after binary decoding The process in which the population gradually adapts to the living environment and the quality is continuously improved
Selection reproduction crossover mutation coding decoding individual	Select a number of individuals from the population with a certain probability DNA is transferred into newly created cells by replication Crossover of DNA at the same location on two chromosomes to form two new chromosomes Mutation during replication produces new chromosomes with new traits The genetic information in DNA is arranged in a certain pattern on a long chain (binary code) Mapping binary to real numbers Chromosomal entity

solution have higher scores. In each iteration, genes with lower scores are discarded. With continuous crossover and mutation, as long as suitable iteration periods are set, a global optimal solution could be achieved.

There are some biological terminologies in genetic algorithm [8]. In order to describe them clearly, accurately, and normatively, here explain the meanings of the nouns and basic concepts in Table 1.

First and foremost, what must be prioritized is population initialization, creating a population by randomly selecting features. Then use binary to encode a chromosome as a genotype (randomly generate a string of 0, 1 of a specific length, representing an individual's genes). Length is decided by the distribution of the weights calculated by the previous information entropy method. The population size is set, too. Next step is estimating the fitness of the individual in the current environment through fitness function (such as SVM, Random Forest, etc.).

Implementing the iteration of parent and child requires selection. According to the fitness of individuals in the population, individuals with high fitness are selected from the current population by an approach named roulette. Roulette is to determine the amount of each individual's inheritance into the next generation population through the individual fitness size and the probability proportional to it. The greater the individual fitness, the higher the probability of survival, and the greater the probability of genes being passed on to the next generation.

Also, mating probability needs to be set, resplicing the binary strings representing two individuals to obtain a new binary code, thereby generating a new individual. Furthermore, generate mutation points in gene sequences randomly, which reverses the original genes of the mutation points according to the mutation probability threshold (change 0 to 1, 1 to 0). Therefore, generate new individuals.

When the dominant gene changes are no longer significant or the number of iterations reaches the preset epochs, the algorithm is terminated and the genotypes of outstanding individuals are outputted. Finally, decoding genotype into phenotype is the optimal solution.

3. Entropy Embedding Genetic Algorithm for Feature Selection

For massive dataset with complex features and huge number of samples, the smooth progress of the classification will be influenced by the high time complexity and the noise generated by the unimportant features. Hence, dimensionality reduction technique named feature selection is necessary, where wrapper algorithms (GA, ant colony, algorithm, etc.) optimize through intelligent search. However, these wrapper algorithms are easy to be affected by irrelevant features and different search strategies, causing low accuracy and even overfitting. Therefore, it is of vital importance to select the most relevant features before optimizing, which is related to weight distribution, for the reason that

Information Entropy Theory can provide high-quality weight allocation for wrapper algorithms due to its ability to measure the amount of information in indicators. Entropy Embedding Genetic Algorithm for feature selection (EEGA) proposed in this paper, which is a mixed algorithm about filter and wrapper, combines Information Entropy Theory with genetic algorithm for optimizing to realize feature selection.

3.1. Weight Distribution Based on Information Entropy Theory. For chosen dataset, values of information entropy about every feature can be exported by Information Entropy Theory, achieving a kind of direct weight distribution. The results of weight distribution serve as labels for individual features for the composition of individuals in each population in GA. Entropy calculation especially for features are represented as follows.

At first, what must be prioritized is dimension unification by standardizing data based on positive and negative definition of features. The formulas used are shown in equations (3) and (4)

$$x_{ij}^{*} = \frac{x_{ij} - \min}{\max - \min}, \text{if positive}, \tag{3}$$

$$x_{ij}^{*} = \frac{\max - x_{ij}}{\max - \min}, \text{ if negative}, \tag{4}$$

where max and min, respectively, represent the maximum and minimum values of all sample data in each feature.

Then, calculate the proportion p_{ij} of the value of sample i under the feature j as

$$\begin{aligned} p_{ij} &= \frac{x_{ij}^{*}}{\sum_{i=1}^{n} x_{ij}^{*}}, i \\ &= 1, \cdots, n; j \\ &= 1, \ldots m. \end{aligned} \tag{5}$$

Next, figure out the entropy e_j of feature j by equation (2).

$$e_j = -\frac{1}{\log n} \sum_{i=1}^{n} p_{ij} \log\left(p_{ij}\right). \tag{6}$$

Accordingly, the weight of the feature can be indicated in

$$\begin{aligned} w_j &= \frac{1 - e_j}{m - \sum_j^m e_j}, j \\ &= 1, \ldots m. \end{aligned} \tag{7}$$

In this paper, the process of weight distribution based on information entropy is described in Algorithm 1.

3.2. Genetic Algorithm with Entropy Label. Genetic algorithm (GA) is a kind of optimization algorithm, aiming to search the global optimal point. And due to its characteristic of dynamic adjustment, its derived results are not easy to fall into local optimum. In classification, the high accuracy of classification is the prime gist, as well as the optimal direction in GA. What is more, the calculation of classification accuracy relied on fitness function, which can be any basic classification algorithm such as decision tree, KNN, etc. or just linear discriminant function like Fisher, in GA.

First job is population construction. In common genetic algorithm, each individual in population is the data with different feature clustering. However, it is too redundant with carrying a large number of low-correlated features for heavy fitness calculations in massive dataset. Therefore, we introduce features with entropy labels rather than raw data of features and use independent threshold for individual in population rather than the feature clustering. By this approach, features can be chosen by the comparison of the threshold and entropy labels, and population iterations in GA are transformed into the simple threshold iterations, which effectively simplifies the algorithm. For population initialization, each threshold corresponds to a unique binary code, called gene sequence or DNA sequence. The maximum length of gene sequence is decided by features and the distribution of their entropies, which would better be set first, called DNA size. Pop size that is the number of individuals in this generation should also be set in this step. Then, based on pop size, randomly generate binary sequences with DNA size as the primary population. And every individual in primary population has its own normalized decimal value.

The second noteworthy step is the calculation of different individual fitness. Compare the information entropy value of each feature in the data with the individual threshold, and the features that meet the standard will be selected. Finally, a new data list will be formed with calculating its fitness, that is, the classification accuracy. After the above process, each individual in this generation owns its fitness value.

Then, it is the natural selection algorithm in GA, which uses roulette approach to decide genotypes of next generation. As the name suggests, the principle of roulette is calculating the proportion of fitness of each individual to the sum of fitness, so every individual has its own area on the turntable based on the proportion. One genotype of an individual is selected by generating a random number (looks like pointer on this turntable), where the number of repetitions is equal to the size of population. Therefore, these selected genotypes will participate in crossover and mutation later. Besides, at the part of calculation of the proportion of fitness, normalization should be finished first, and a penalty function is proposed, shown in equation (8), smoothing the fitness and emphasizing the merit of dominant individuals.

$$F_{\text{new fit}} = 2 \sin\left(\frac{\pi}{2}\alpha \cdot f\right) \cdot f, \tag{8}$$

where f is fitness, $F_{\text{new fit}}$ is the processed fitness function, $y(f) = 2 \sin(\pi/2\alpha \cdot f)$ represents penalty function, and $\alpha \in (0, 1]$ is the penalty coefficient.

In roulette, cumulative probability is calculated to allow each genotype to own its distribution in $(0, 1]$. Then, generate random numbers in $(0, 1]$ to realize the consequent of

Input: NumPy independent variable dataset D
Output: Weight list w_j_list
(1) Normalize based on custom functions
(2) Export data shape, the number of rows is n, the number of columns is m
(3) Calculate the Proportion $p_{ij} = x_{ij}^* / \sum_{i=1}^{n} x_{ij}^*,\ i = 1, \ldots, n;\ j = 1, \ldots m$
(4) Figure out the entropy $e_j = -1/\log n \sum_{i=1}^{n} p_{ij} \log(p_{ij})$
(5) Calculate the weight of each indicator $w_j = 1 - e_j / m - \sum_j^m e_j$
(6) Form weight list w_j_list

Algorithm 1: Information entropy weight distribution.

Input: Population pop, fitness list fit_value
Output: New population pop
(1) Normalize pop and construct penalty map for fit_value as $F_{\text{new fit}} = 2 \sin(\pi/2\alpha \cdot \text{fit\_value}) \cdot \text{fit\_value}$
(2) Calculate the fitness sum and cumulative probability
(3) Create cumulative probability list *cum*
(4) **For** i in range(len(pop)) **do**
(5) Generate 0-1 random numbers into two lists $ms1, ms2$
End for
(6) Sort $ms1, ms2$
(7) Traverse the list without exceeding the scope of the list $ms1$ and *cum*. If the value of this position of $ms1$ is less than the value of the current position of *cum*, the genotype at the current position in *pop* is stored in the temporary gene register new pop 1. Position number in $ms1 + 1$, otherwise the current position +1, until finish traversing.
(8) Perform step 7 on $ms2$ and pop so that new pop 2 is formed.
(9) Compare the value of each position of new pop 1 and new pop 2, select the greater value and put it into pop

Algorithm 2: Selection in GA.

Input: Population pop, CROSSOVER_RATE, MUTATION_RATE
Output: New population new_pop
(1) **For** father in pop **do**
(2) child = father
(3) If generate 0-1 random numbers, less than CROSSOVER_RATE
(4) Select another individual in the population and use that individual as the mother.
(5) Randomly generate crossover points in the second half of the DNA sequence. child gets the genes of the mother behind the junction.
(6) If generate 0-1 random numbers, less than MUTATION_RATE
(7) Randomly generate mutation points and change the value of this binary point
(8) Put child into list new_pop
End for

Algorithm 3: Crossover and mutation in GA.

turntable. In connection with massive dataset, double roulette mechanism is introduced, which chooses better result in each selection of genotype, significantly improving model quality even compared with original genetic algorithm model for halving the population (because original roulette mechanism may meet the loss of dominant population with low pop size). What is more, model time complexity is also reduced, especially using complex discriminant model in fitness calculation.

The process of crossover is an imitation in nature that means the crossover of DNA at the same location on two chromosomes to form two new chromosomes. In this paper, randomly generated crossover points are restricted to the second half of the parental DNA sequence in order to better simulate the natural characteristics of the child inheriting the shape of the parent. The description of selection, crossover, and mutation in genetic algorithm are separately shown in Algorithms 2 and 3.

Repeat the above process according to the set number of iterations. Or the average fitness of the population does not change significantly; then end the loop. Overall, EEGA can be summarized as these following points and its technology roadmap is shown in Figure 2.

(i) Import the dataset and complete data cleaning.

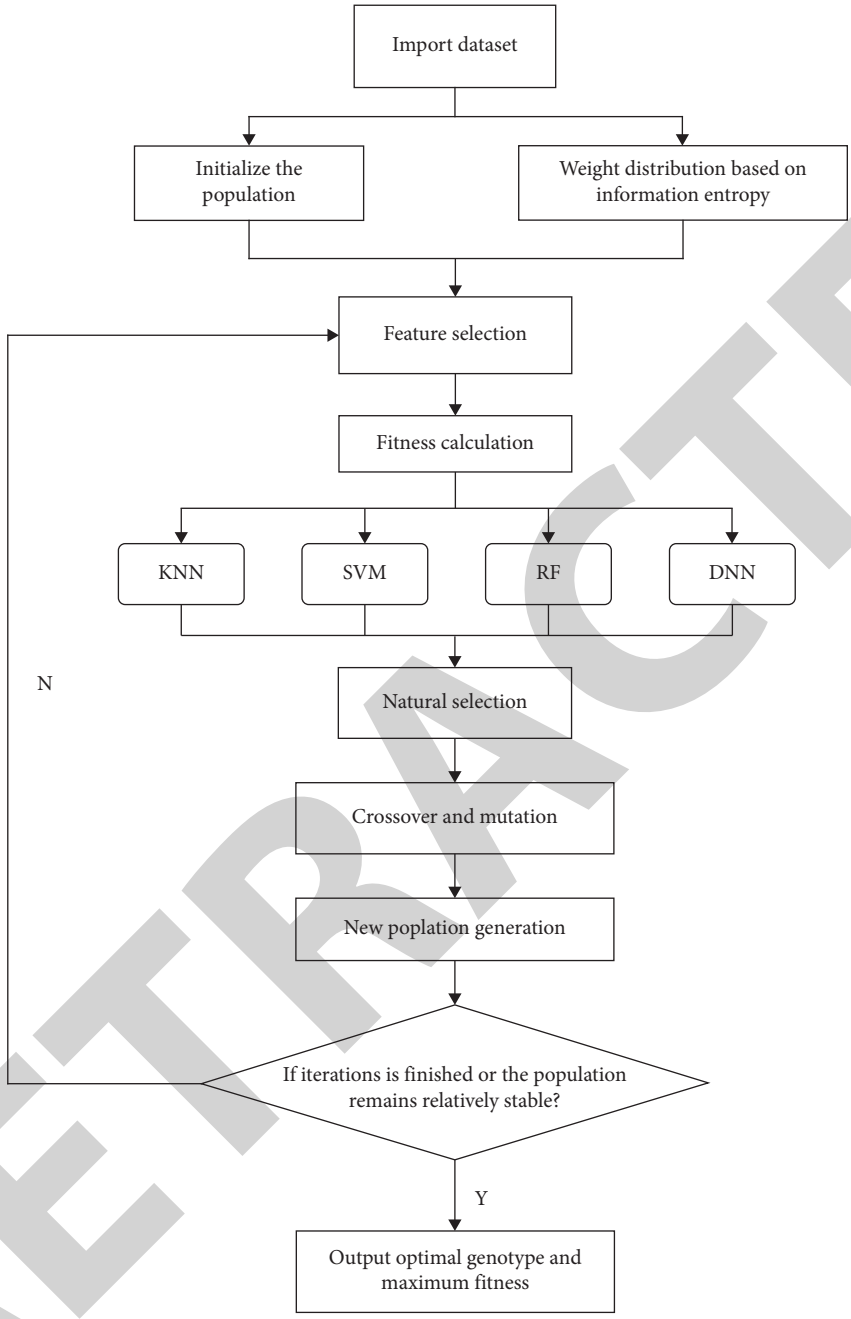

Figure 2: Algorithm framework of EEGA.

(ii) Calculate the information entropy of the feature about the dataset.

(iii) Initialize the population.

(iv) Select features according to weights calculated by information entropy.

(v) Calculate the fitness, or in other words, classification accuracy relied on fitness function, which can be any basic classification algorithm or just linear discriminant function.

(vi) Realize the iterative operation in GA, including selection, crossover, and mutation.

(vii) Stop the genetic algorithm after the number of iterations is reached or the population remains relatively stable.

(viii) Output optimal genotype and maximum fitness.

4. Experiment

4.1. Dataset. The datasets used for the experiment in this paper are all real datasets, all of which are from UCI Machine Learning Repository [9]. In this experiment, five datasets with different characteristics are chosen to test the generality of EEGA, which are from chemistry, medicine, and

Table 2: Detail of datasets.

No	Name	Feature number	Sample size	Category	Missing value	Area
1	Cardiotocography [10]	34	2126	3	N/A	Life
2	Musk [11]	168	6598	2	No	Physical
3	Parkinson speech [12]	26	1040	2	N/A	Life
4	Room occupancy estimation [13]	16	10129	4	N/A	Computer
5	Spambase [14]	57	4601	2	Yes	Computer

computer science. Some of them have missing values and outliers, so data preprocessing has been achieved first. The characteristics of the datasets are organized in Table 2. As is shown in Table 2, the number of samples ranges from 1040 to 10129, and the number of features ranges from 16 to 168.

4.2. Experimental Results and Discussions. The performance of Entropy Embedding Genetic Algorithm for feature selection (EEGA) is visualized in five datasets by two evaluation indicators: accuracy and time complexity. Because EEGA has built-in models of individual fitness of different populations and the model can be seen as an optimization algorithm of these built-in models for classification problem, some basic approaches about classification like Random Forest (RF) are selected as built-in models of EEGA to make comparison about themselves. The specific experimental results can be found below.

4.2.1. Accuracy Assessment. There is no doubt that accuracy is the most important evaluation criterion in classification problems. Because most of classification in machine learning is supervised learning, accuracy can be defined as the proportion of test samples which are correctly classified based on class labels. Hence, the fitness target of genetic algorithm should be set as accuracy in this experiment. What is more, the fitness models, or in other words, the built-in models of GA, are specified as KNN, DNN, SVM, and RF here. Besides, because the results of RF are not fixed, the accuracy of RF and RF with EEGA is tested five times and average is made.

First and foremost, what must be preferentially calculated is information entropy of dataset. And the results of it are shown as Figure 3.

Then, some parameters should be set in EEGA before experiment. The crossover rate and mutation rate are set as 0.8 and 0.02, respectively. Also, population size is 40. And generation cycle is 40, too (while the model would call out the loop in advance in most situation in actual experiment). DNA size is 8 except 9 in Musk dataset. Finally, penalty rate in penalty function equation (8) represents 0.8.

As for built-in models in EEGA, import KNN, DNN, SVM, and RF directly through the sklearn module. For the division of the data set, k-fold cross-validation method is used and the parameter k here is ten. K-fold cross-validation uses the technique without repeated sampling. Each sample point has only one chance to be included in the training set or test set during each iteration. The principle is to randomly divide the dataset into k packages, one of which is used as the test set each time, and the remaining $k-1$ packages are used as the training set for training. Finally, the average of the test results is taken as the final result, shown in

$$Acc = \sum_{i=1}^{k} acc_{\text{test}_i}, \tag{9}$$

where $\text{acc}_{\text{test}_i}$ is the accuracy result on the test set for each fold.

Table 3 shows the comparison about classification accuracy between original algorithms and EEGA. Also, Figure 4 shows the changes of the four algorithms in indicator accuracy before and after the EEGA is introduced.

It can be seen in Table 3 that all performances of four algorithms in five datasets are improved by introducing EEGA. In Cardiotocography dataset, accuracies of all improved models are more than 98%, with approximately 1% to 19% increase. However, no model is more than 90% accurate in Musk, but there are still accuracy boosts within about single digit percent. The limited improvement may be caused by huge 168 features so that genetic algorithm needs bigger DNA size to contain the feature thresholds, while because of the limitation of equipment, the maximum DNA size is set 9. Model performance can be further improved in Musk. The greatest improvement in this experiment occurs in Parkinson Speech using KNN with EEGA, from 63.942% to 99.327% (approximately 55.3% increase shows visually in Figure 4 about Parkinson Speech). Even though some algorithms without EEGA have already achieved excellent accuracy such as DNN and RF in Parkinson Speech and Room Occupancy Estimation dataset, they still obtain significant boost to directly 100%. The outstanding performance of EEGA with all four algorithms illustrates the method can still perform good feature selection in the face of samples with a large amount of data or complex features. Finally, for massive dataset Spambase, two of the fitness algorithms keep smaller accuracy improvement, while the performance of KNN rises to 87.981% (origin is 77.919%), and SVM also obtains 18% magnification.

Besides, through the iteration of genetic algorithm, the optimal genotype, in other words, optimal threshold for feature selection, is figured out. Similarly, optimal threshold in RF is chosen when the maximum accuracy is taken out in five times. The results about it are sorted in Table 4.

It can be easily seen in Table 4 that the optimal threshold is not the same in different datasets with different original algorithms; even the accuracies are equivalent in one dataset, Cardiotocography (0.976 with KNN and 0.305 with SVM), which may be on account of difference in fitness calculation or multiple optimal genotypes, except the situation of 100%

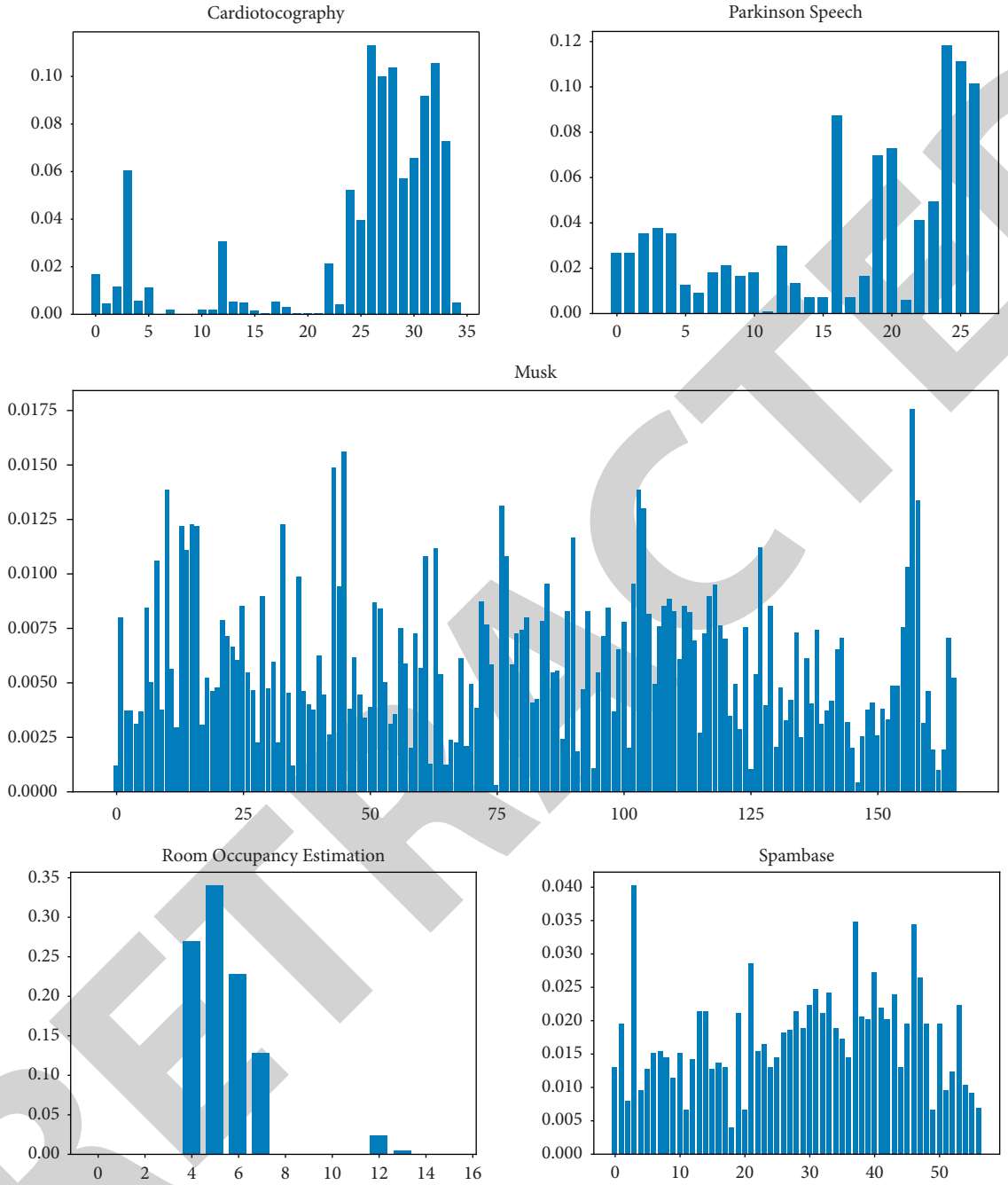

Figure 3: Weight distribution about five datasets.

Table 3: Comparison about classification accuracy.

	Cardiotocography (%)	Musk (%)	Parkinson speech (%)	Room occupancy estimation (%)	Spambase (%)
KNN	80.995	77.242	63.942	95.883	77.919
KNN with EEGA	98.401	84.953	99.327	97.166	87.981
DNN	91.624	81.029	95.481	97.956	91.936
DNN with EEGA	98.495	85.087	100	100	93.936
SVM	77.846	88.408	79.519	96.761	71.072
SVM with EEGA	98.401	89.510	99.808	97.344	89.785
RF	97.971	79.999	99.904	97.660	94.066
RF with EEGA	98.542	87.225	100	100	95.087

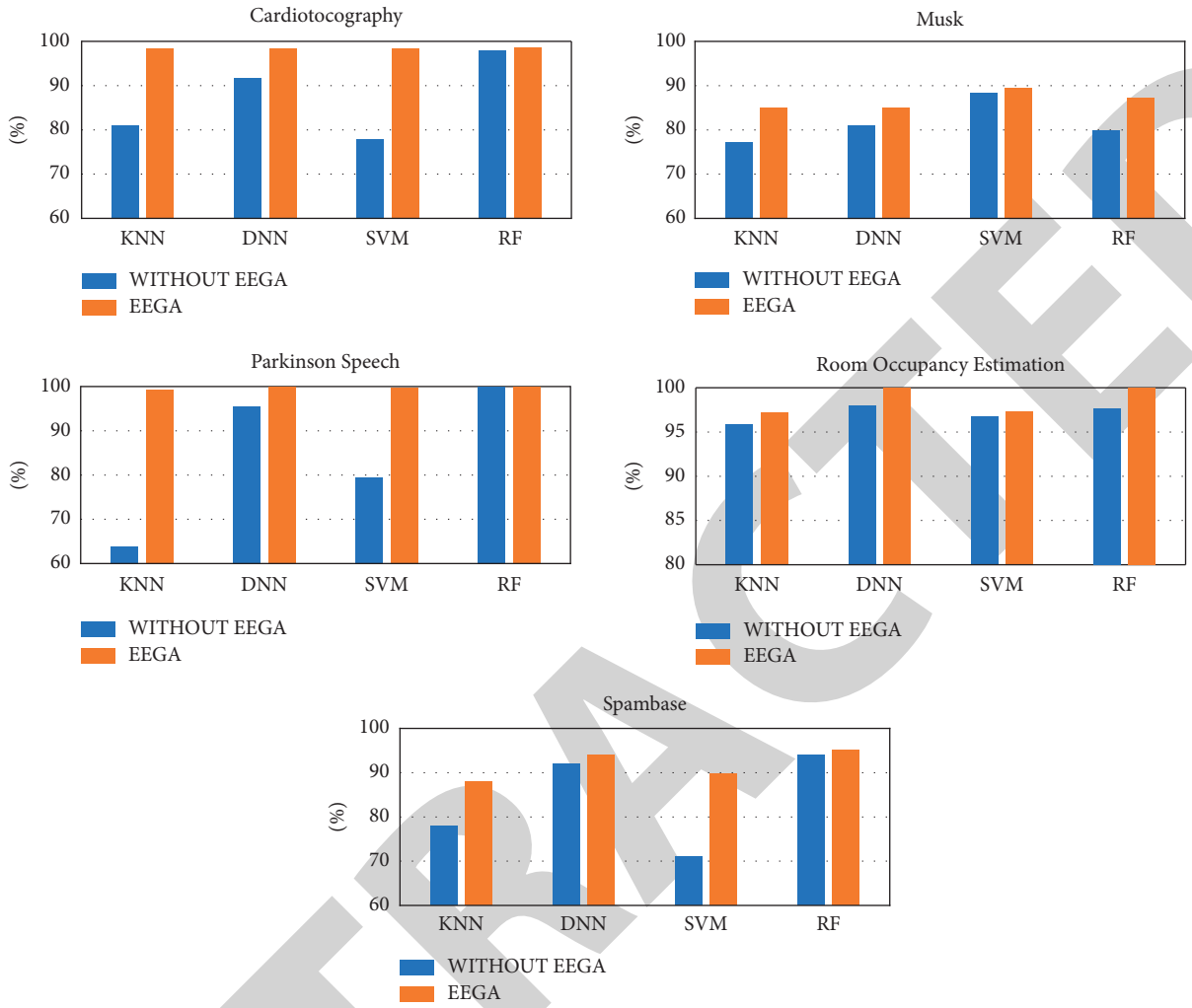

Figure 4: Accuracy about five datasets.

Table 4: The optimal threshold in EEGA for feature selection.

	Cardiotocography	Musk	Parkinson speech	Room occupancy estimation	Spambase
KNN with EEGA	0.976	0.375	0.699	0.758	0.789
DNN with EEGA	0.582	0.117	0.808	0.793	0.133
SVM with EEGA	0.305	0.555	0.797	0.840	0.223
RF with EEGA	0.281	0.457	0.808	0.793	0.863

Table 5: Comparison about time complexity (per second).

	Cardiotocography	Musk	Parkinson speech	Room occupancy estimation	Spambase
KNN	0.17310	1.25101	0.12497	0.68997	0.61050
KNN with EEGA	0.14054	1.14162	0.04682	0.48594	0.54809
DNN	4.68476	19.93732	10.3715	16.45996	11.59904
DNN with EEGA	3.89772	19.16979	7.96646	13.41001	10.45439
SVM	1.09468	6.74315	0.40744	3.39282	7.24195
SVM with EEGA	1.08049	5.74477	0.39053	3.12939	7.22855
RF	2.13242	20.03398	1.80042	3.55173	5.04623
RF with EEGA	2.02833	18.74099	1.76169	2.64556	4.93956

accuracy. However, the calculation of optimal thresholds provides a new possibility of feature selection that obtains the result of optimal genotype with shallower iteration to make new data for classification.

4.3. Time Assessment. Time complexity is also one of the important evaluation indicators to determine the quality of the classification model. Many classification algorithms would rather sacrifice accuracy to reduce model time complexity. In some theory, time complexity can be simply described as the running time of algorithm. And this paper follows this definition. The running time is measured and calculated between original algorithms and EEGA, shown in Table 5. The final running time is the average of the results of five runs.

As shown in Table 5, through feature selection of EEGA, the time complexity of the algorithms is generally reduced, where DNN obtains the greater performance boost in time assessment in range between 9.9% and 23%. The rest of the algorithms make running time decrease but not all evidently, while most of improvement is more than 5% and the most significant dropping about time complexity is KNN with EEGA for Parkinson Speech (62%), which illustrates that EEGA makes great contribution on time complexity reduction. Besides, as the complexity of the model increases and the number of features grows, the advantages of this model will be further highlighted.

5. Conclusion

In this paper, an improved feature reduction algorithm is proposed called Entropy Embedding Genetic Algorithm for feature selection (EEGA), which is the combination of filter and wrapper methods. The main principle of it is making use of information entropy to calculate feature weights and using the distributed weights as the labels of features into population iteration in genetic algorithm instead of the whole data into generation which greatly reduces model complexity. What is more, because of the decrease of features, that is, significant features get more weight, the accuracy of the model has also generally improved. The performance improvement mentioned above has been proved by the experiment about the comparison between EEGA and original algorithms without EEGA. The experiments have been performed by using several standard databases with different fitness methods. Classification accuracy has been improved by maximum 20% and each error rate is limited to 15%. The similar boost is proved about reducing model time complexity in range between approximately 5% and 23%. The improvement in accuracy and the reduction in time complexity both demonstrate the effectiveness of EEGA in feature selection [15].

Data Availability

The datasets used and/or analyzed during the current study are available from the corresponding author on reasonable request.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

References

[1] I. K. Fodor, *A Survey of Dimension Reduction techniques*, Lawrence Livermore National Lab, CA (US), 2002.

[2] A. Todorov, "An overview of the RELIEF algorithm and advancements," *Statistica L Approaches to Gene X Environment Interactions for Complex Phenotypes*, 2016.

[3] I. Kononenko, "Estimating attributes: analysis and extensions of RELIEF," *Machine Learning: ECML-94*, Springer, in *Proceedings of the European conference on machine learning*, pp. 171–182, May 1994.

[4] H. Peng, C. Ying, S. Tan, and B. Z Hu, "An improved feature selection algorithm based on ant colony optimization," *IEEE Access*, vol. 6, pp. 69203–69209, 2018.

[5] R. Haidi, S. Xianzhang, K. R. Ahoussou et al., "Feature selection based on artificial bee colony and gradient boosting decision tree," *Applied Soft Computing*, vol. 74, 2019.

[6] Y. Wang and L. Feng, "An adaptive boosting algorithm based on weighted feature selection and category classification confidence," *Applied Intelligence*, vol. 51, pp. 1–22, 2021.

[7] T. V. Mathew, "Genetic algorithm," *Report submitted at IIT Bombay*, 2012.

[8] J. H. Holland, *Adaptation in Natural and Artificial Systems*, University of Michigan Press, Ann Arbor, 1975.

[9] D. Dua and C. Graff, "UCI machine learning repository," University of California, School of Information and Computer Science, Irvine, CA, 2019, http://archive.ics.uci.edu/ml.

[10] D. Ayres-de-Campos, J Bernardes, A Garrido, and L Pereira-Leite, Sisporto 2.0: a program for automated analysis of cardiotocograms," *The Journal of Maternal-Fetal Medicine*, vol. 9, no. 5, pp. 311–318, 2000.

[11] T. G. Dietterich, A. Jain, R. Lathrop, and T. Lozano-Perez, "A comparison of dynamic reposing and tangent distance for drug activity prediction," *Advances in Neural Information Processing Systems*, vol. 6, pp. 216–223, Morgan Kaufmann, San Mateo, CA, 1994.

[12] B. E. Sakar, M. E. Isenkul, C. O. Sakar et al., "Collection and analysis of a Parkinson Speech dataset with multiple types of sound recordings," *IEEE Journal of Biomedical and Health Informatics*, vol. 17, no. 4, pp. 828–834, 2013.

[13] A. P. Singh, V. Jain, S. Chaudhari, F. A. Kraemer, S. Werner, and V. Garg, "Machine learning-based occupancy estimation using multivariate sensor nodes," in *Proceedings of the 2018 IEEE Globecom Workshops (GC Wkshps)*, Abu Dhabi, UAE, December 2018.

[14] D. Hush, S. Clint, and S. Ingo, "Stability of unstable learning algorithms," *Machine learning*, vol. 67, no. 3, pp. 197–206, 2006.

[15] C. E. Shannon, "A mathematical theory of communication," *Bell System Technical Journal*, vol. 27, no. 3, pp. 379–423, 1948.