Hindawi
Scientific Programming
Volume 2018, Article ID 2691759, 10 pages
https://doi.org/10.1155/2018/2691759

Research Article

Feature Reduction Based on Hybrid Efficient Weighted Gene Genetic Algorithms with Artificial Neural Network for Machine Learning Problems in the Big Data

Tareq Abed Mohammed,[1,2] **Shaymaa Alhayali**,[1] **Oguz Bayat**,[1] **and Osman N. Uçan**[1]

[1]*Altinbas University, College of Engineering, Istanbul, Turkey*
[2]*Kirkuk University, College of Science, Kirkuk, Iraq*

Correspondence should be addressed to Tareq Abed Mohammed; tareq.mohammed@altinbas.edu.tr

Received 27 May 2018; Revised 8 August 2018; Accepted 11 October 2018; Published 30 October 2018

Academic Editor: Marco Aldinucci

A large amount of data being generated from different sources and the analyzing and extracting of useful information from these data becomes a very complex task. The difficulty of dealing with big data arises from many factors such as the high number of features, existence of lost data, and variety of data. One of the most effective solutions that used to overcome the huge amount of big data is the feature reduction process. In this paper, a set of hybrid and efficient algorithms are proposed to classify the datasets that have large feature size by merging the genetic algorithms with the artificial neural networks. The genetic algorithms are used as a prestep to significantly reduce the feature size of the analyzed data before handling that data using machine learning techniques. Reducing the number of features simplifies the task of classifying the analyzed data and enhances the performance of the machine learning algorithms that are used to extract valuable information from big data. The proposed algorithms use a new gene-weight mechanism that can significantly enhance the performance and decrease the required search time. The proposed algorithms are applied on different datasets to pick the most relative and important features before applying the artificial neural networks algorithm, and the results show that our proposed algorithms can effectively enhance the classifying performance over the tested datasets.

1. Introduction

In recent years, the major increase in the amount of generated data makes it very important to develop new robust and scalable tools that are able to extract the hidden knowledge and information from the big datasets [1]. When the dataset that we are dealing with has a massive volume of data and includes both structured and unstructured data, it is called a big data [2, 3]. The big data becomes a specific and separated field in computer engineering society since it is difficult to be processed using the traditional database and software techniques. Big data has other different specific properties, such as the velocity which refers to the speed at which data are being generated, the variety which means the existence of structured and unstructured data, and the variability which means the inconsistencies of the data. The main objective of big data is to help people and companies to improve their operations and make faster and more intelligent decisions. Recently, the big data technologies have received increasing attention from researchers and companies, and many conferences and journal special sessions are established to discuss their issues and characteristics [4, 5].

One of the most critical problems of big data is the degrading of the performance of the machine learning and data mining algorithms when dealing with such large amount of data [6]. This can happen because of many factors such as the existence of a large number of features, the existence of lost data, and the high computations of traditional machine learning and data mining algorithms which makes them unsuitable to efficiently deal with large datasets. Several new classification techniques are proposed to overcome the challenges of big data which can classify the

big data according to the data format during the processing and the type of classification required. Most of the proposed classification methods are based on the specific selected applications and may not give good results if it is applied to other big data applications [7]. In order to classify the data efficiently, usually a convenient algorithm is needed to extract the relevant information from a large amount of data as a prestep and then the classification algorithm can be applied [8, 9]. Two main approaches are used to reduce the available data before applying the classification algorithms which are filtering the data and feature reduction. In this paper, we will concentrate on the feature reduction methods which identify the most important features (rather than using all of them) only and use them in next classification steps.

The classification process can be defined as a method for identifying the category or the class (it should be two or more classes) of a certain piece of data based on a system that was trained using data whose class is known. In the real world, there are classification problems everywhere, and we can find hundreds or thousands of real-world classification problems [6, 10]. In the classification of big data problems, the feature selection is very important since it addresses the problem of large dimension by choosing only the most relevant features that can lead to correct classification. The process of eliminating the irrelevant and redundant features is called feature reduction or feature selection, and it has many advantages such as reduced training time, decreased complexity of the learned classifiers, and enhanced performance of the classification results. Although the feature selection algorithms are used before the classification run, it is very important and can significantly affect the results of the classification; this is because the existence of redundant and irrelevant features may cause the build of the incorrect classification system during the training process.

In this paper, three efficient genetic algorithms are proposed to pick the relative and important features before applying the artificial neural networks algorithm. The proposed algorithms use the new mechanism which is the weight-based correction for each feature, which can guide the searching process quickly to optimal solutions. The results show that our proposed algorithms can effectively enhance the classifying performance over the tested datasets.

This paper is organized as follows. Section 2 reviews the related work regarding the methods used to enhance the data mining algorithms when applied on big datasets. In Sections 3, 4, and 5, we present and explain our algorithms to enhance the artificial neural networks to be able to deal with large feature datasets and discussion of datasets. The experiments and the discussions are given in Section 6. Section 7 concludes the paper.

2. Related Work

By reviewing the literature, it can be noted that the feature selection algorithms gain increasing interest, especially in the big data fields. In this section, we will summarize the research work on the feature selection problem and try to list the most important algorithms that are proposed to address this problem. Feature selection can be used with many machine learning algorithms such as regression, clustering, and classification, whereas in this paper, we will concentrate only on the feature selection with classification.

In literature, the feature selection algorithms can be classified into two categories: filter methods and wrapper methods [11, 12]. The wrapper methods usually use the classification algorithm to measure the performance of the tested feature selection method. On the contrary, the filter feature selection algorithms are independent of any classification algorithm and use other scientific methods to measure the goodness of each feature. The filter-based feature selection methods are often less computationally expensive than the wrapper methods, since it does not need the run of the classification algorithm to test the considered method. However, the wrapper methods usually obtain better results and performance than the filter methods [13, 14].

One of the earliest works on feature selection is the usage of the greedy search methods such as sequential forward selection and sequential backward selection. In [15], the authors proposed a method of measurement selection to identify the best subset of features based on a forwarded selection technique. The used evaluation method uses a nonparametric estimation of the error probability given a finite sample set. The main advantage of this method is the direct and nonparametric evaluation of measurement subsets. On the contrary, the sequential backward selection proposed in [16] tried to develop a formal method to measure the effectiveness of a set of features or tests. The authors mainly consider the following question: "what constitutes an effective set of tests, and how is this effectiveness dependent on the correlations among, and the properties of, the individual tests in the set?" [16]. Unfortunately, both of forwarding selection and sequential backward selection methods suffer from a problem called the nesting effect, which happens as a result of removing or selecting a feature once only. This means that if a feature is removed in an early step, it cannot be used in next steps. To overcome this problem, another approach is proposed in [17] to merge the two methods together by applying the forward selection method one time and then follow it with multiple runs of the sequential backward selection method. Much other research works are proposed to enhance the performance of the forwarding selection and sequential backward selection methods by using the floating search methods as in sequential backward floating selection and sequential forward floating selection [18, 19].

In another work, Fan Mina, Qinghua Hub, and William Zhu proposed a feature selection algorithm that includes a test cost constraint problem [20]. The new algorithm uses the backtracking algorithm which is a well-known algorithm used to solve many specific optimization problems. The authors argued that the backtracking algorithm is convenient and efficient to be used to solve the feature selection problem on medium-sized data. In addition, another heuristic algorithm is developed to be used in parallel with the backtracking algorithm to make it more scalable and able to work on large datasets. The experimental results of this algorithm demonstrate that the developed heuristic algorithm can identify the optimal solution of the problem in many cases.

After the development of evolutionary computation algorithms (EC), many researchers tried to use these algorithms to solve the problem of feature selection. For example, in [21], the authors presented a genetic algorithm that is modified to consider the bounds of the generalization error in the support vector machines (SVMs). The proposed algorithm was compared to the other traditional algorithms and approved its validity when solving such feature selection problems. Oh et al. [22] proposed a new genetic algorithm by modifying the existing one to be more suitable for feature selection. The main objective of the new proposed algorithm is the hybridization of the local search operation and the genetic algorithm to make tuning for the search process. According to the authors, the hybridization process can produce a significant improvement in the final performance of the genetic algorithm.

Recently, some hybrid bioinspired heuristic approaches were proposed to reduce the feature size of the input data such as the work of Zawbaa et al. [23], whereas a hybrid algorithm is proposed to handle the large-dimensionality small-instance set feature selection problems. In [24], another algorithm is proposed to handle the feature selection problem using Levy Antlion optimization. The flower pollination algorithm [25] is used also in another research to make an attribute reduction after modifying it using new adaptive techniques to handle such problems.

The multiobjective evolutionary algorithms are also used to reduce the number of selected features. In [26], the authors presented the first research on multiobjective optimization particle swarm optimization to solve the feature selection problem using the particle swarm optimization (PSO). The algorithm works by generating a set of nondominated solutions to be considered as the candidates feature subsets. The authors investigated two multiobjective algorithms based on PSO. The first algorithm uses the nondominated sorting algorithm and PSO to where the second algorithm uses crowding distance, dominance relation, and mutation to search for the best solutions. The results of comparing the two proposed multiobjective algorithms with other feature selection algorithms show that the PSO multiobjective algorithms can significantly outperform the other algorithms and get better results. Recently, there were many other new algorithms that proposed to solve the feature selection problem using multiobjective evolutionary algorithms using different techniques [27–29].

3. Proposed Techniques

Feature selection problems become one of the most important problems in big data society. The main issue of such problems is the existence of large search space, which can be considered as NP-hard problems that cannot be solved until testing all the search space. Another issue in feature selection is the feature interaction problem which leads to the translation of some features from relevant to redundant or weakly features. On the contrary, some features become very important when combined with other features. The evolutionary computation (EC) algorithms have a very useful property that makes them the best choice to solve feature selection problems, which is it does not require any domain knowledge or assumption about the search space to solve the feature selection problem. Another advantage of the EC algorithms is the process of evolving a set of solutions (called as the population in EC) which speeds up the process of converging to the optimal solutions. Therefore, our proposed algorithm to solve the feature selection problem involves the hybridization of both machine learning algorithms and evolutionary algorithms, as described in the rest of this section.

3.1. Main Steps of the Proposed Algorithm. Our proposed algorithm mainly merges the well-known artificial neural network (ANN) algorithm as a classification algorithm with a new and efficient evolutionary algorithm called the weighted gene genetic algorithm (WGGA). Figure 1 shows the main steps of the proposed algorithm. Firstly, the dataset is read and entered to be used in the process of generating a random set of features. After that, the generated feature sets are used to classify the dataset using the ANN algorithm. According to the results of classification, our proposed WGGA algorithm generates new sets of features that are a candidate to have a better performance. The evolving and optimization process is repeated many times until reaching the stop criteria. If the stop condition is not satisfied, the process of evaluating new feature sets is continuing to search for better solutions. The stop condition can be reaching a maximum number of iterations, reaching a predetermined performance value, or maybe a hybridization of both cases to avoid very long running time.

3.2. The Weighted Gene Genetic Algorithm (WGGA). In literature, there are many evolutionary algorithms proposed to solve several optimization problems. In this paper, a new efficient genetic algorithm is presented which was especially developed to handle the feature selection problem. We called it weighted gene genetic algorithm (WGGA), since it stores weight for each gene in order to enhance the convergence ability of the algorithm. Figure 2 summarizes the steps of the proposed algorithm.

The proposed weighted gene genetic algorithm (WGGA) uses the binary representation to encode the solutions. Therefore, each solution is represented by an array that has a size equal to the number of features in the dataset. Each feature is represented by one variable in the array, and value 1 indicates that this feature will be used in the classification process of the ANN, whereas value 0 indicates that it will not be included. Figure 3 shows an example of the encoding of two solutions when the number of features is equal to 10. In the first row of Figure 3, there are 10 elements in the array where 6 of them have a value of 1 and 4 have a value of 0, which indicates that the first, the third, the fourth, and the eighth features will not be used by this solution. In the second solution, the second and seventh features will only be excluded from the classification process.

In the first step, the WGGA algorithm initializes the population randomly and then the fitness function of each solution in the population is computed using the ANN algorithm. The fitness value of each solution is the

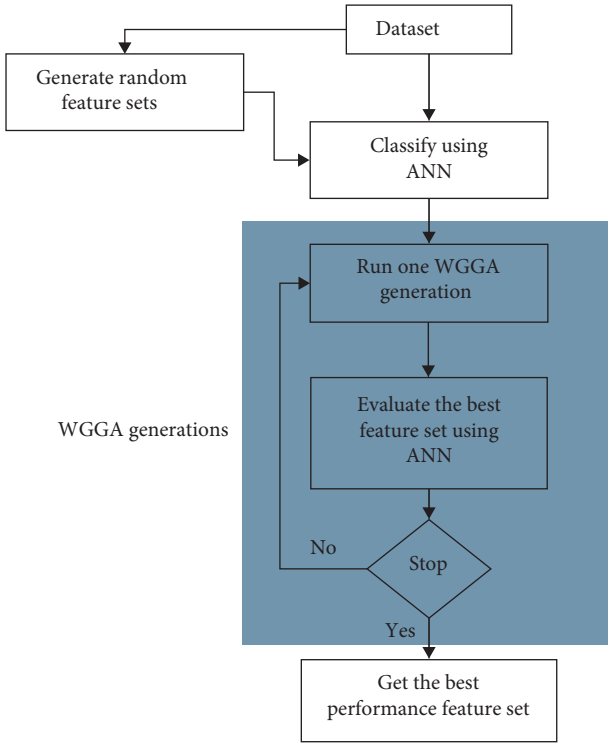

Figure 1: The main steps of the proposed algorithms to solve the feature selection problem.

classification accuracy of that feature set using the input dataset. It can be calculated using the following equation:

$$\text{Accuracy of ANN} = \frac{TP + TN}{TP + TN + FP + FN}, \quad (1)$$

where TP, TN, FP, and FN indicate the true positive, true negative, false positive, and false negative of the tested sample. The weight of each gene is also initialized by zero to be used in next steps.

Based on the evaluation of each solution, the best two solutions are selected to be used for the crossover operation. In this work, a one-point binary crossover is selected to be applied on the best two solutions. In this crossover, a point is randomly selected and then the tails of the two parents are swapped to generate the new off springs. Figure 4 shows an example of the one-point crossover.

In the mutation step, simply according to the mutation percent, a fixed number of genes are chosen and flagged for all population solutions. To ensure the validity of solutions, it is not allowed to be a solution with all genes equal to zero in the population. Therefore, in our algorithm after the crossover and mutation processes, the solutions are checked again and if a solution is found that has all genes are zeros then it is translated to become a gene with all genes equal to one.

Finally, another important step is carried out which is the correction of genes according to their weights. In each generation, the selected features for the best solution are used to increase the weights of that features. This process is accomplished by simply adding one for each selected feature of the best solution in each generation. After that, these stored data are used to correct the false change that may occur in the

```
Begin
Ds = ReadDataSet ()
Pop = InitializePop()    // initialize randomly the population
WghSolutions = Initialize(0)   // initialize the weights by zeros
While  termination condition not satisfied do
.
Res = EvaluateANN(Pop)          // evaluate all solutions
Perf = ComputePerformance(Res)
Sol1 = FindFirstBestSol(Pop, Perf)
Sol2 = FindSecondBestSol(Pop, Perf)
WghSolutions = UpdateSolutionWeights(WghSolutions,Sol1)
// Crossover
   OnePointCrossover(Sol1,Sol2)
// Mutation
BinaryFlagMutation(Pop)
TestValidity(Pop)
If iteration > CC, then
            CorrectGenes(Pop, WghSolutions )
end
end //    while end
end //    Algorithm   end
```

Figure 2: The main steps of the proposed weighted gene genetic algorithm (WGGA).

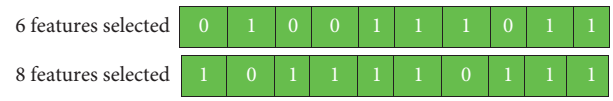

Figure 3: Two examples for binary encoding used in the WGGA algorithm.

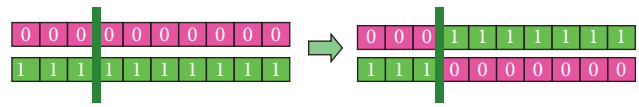

Figure 4: One-point crossover example that is used in the WGGA algorithm.

mutation and crossover processes. This process is carried out after the running of suitable number of generations to ensure the reliability and correctness of the collected information regarding the features. Therefore, it is carried out after CC number of generations as described in Figure 2. When this mechanism is applied, at the end of each generation, the witness weight of each feature is computed by dividing its value in the weighted array by the number of iteration that finished at that time, and the resulted value is compared with another two parameters called high parameter (HP) and low parameter (LP). For each gene (represents a feature), if its

Table 1: The properties of the selected datasets and their features and samples numbers.

Name of date set	Number of features	Number of samples	Description
Heart	13	270	Heart medical analysis
Lung cancer	56	32	Lung cancer analysis
WBCD	9	699	Breast cancer database
Phishing	30	11055	Phishing websites features
Messidor	19	1151	Messidor image set prediction
Sport articles	59	1000	Sports articles for objectivity analysis data set

weight value is greater than HP then it is directly assigned to a value of 1. On the contrary, if its weight value is less than LP, then it is directly assigned to a value of 0. Using this mechanism, the genetic algorithm can concentrate more on the weak or semiweak features, where the strong features are always selected. This process can significantly improve the performance of the feature selection process and decrease the search time as described in next section.

Regarding the computational cost of our proposed algorithm, it can be seen that it does not add any significant cost to the genetic algorithm. This is because our mechanism mainly depends on gathering information about the current population and storing it. Therefore, only an additional small memory is needed for the storing process, and the other used mechanisms have a very small cost which can be ignored. As all embedded feature selection algorithms, the proposed algorithm needs more time comparing with filter-based feature selection algorithms that are presented in literature.

4. Proposed Algorithms

In this section, the setting and results of the empirical experiments are presented to ensure the performance of the proposed algorithms. As mentioned in the previous section, three genetic algorithms are proposed and incorporated in our experiments as follows.

4.1. Low Weighted Gene Genetic Algorithm (LWGGA). This algorithm uses our proposed weight-based mechanism to exclude the weak features from the selected feature set if they have very low weights. In this case, the flagged features (continuously changing from one to zero or from zero to one) will mostly be out of the selected features after the run of the certain number of generations.

4.2. High Weighted Gene Genetic Algorithm (HWGGA). This algorithm uses our proposed weight-based mechanism to include the strong features always in the selected feature set if they have very high weights. In this case, the strong features (continuously being selected in the best solutions) will mostly be selected after the run of a certain number of generations.

4.3. Weighted Gene Genetic Algorithm (WGGA). This algorithm uses both of the low and high weighted mechanisms described in previous LWGGA and HWGGA algorithms. The merger of these two mechanisms makes the genetic algorithm to concentrate on the important features which enhance the convergence ability of the algorithm and decreases the search time.

Table 2: The parameters used in the genetic algorithm of our proposed algorithms.

The parameter name	Used value
Size of population	10
Type of crossover	One point (0.7)
Type of mutation	Flagged (0.2)
Parents selection	Best two
Stop criteria	Number of generations

Another two algorithms are included in the experiments which are the artificial neural network (ANN) and the normal feature selection genetic algorithm merged with the ANN algorithm denoted as (GA + ANN).

5. Diseases Datasets

To investigate the performance of the proposed algorithms, six datasets from different sources and with different features are selected. Table 1 summarizes the features of the tested datasets. As the table shows, the datasets are selected from different fields and the number of features (attributes) is selected to be in different ranges to test our proposed algorithms using different levels [30].

(1) Heart medical analysis
(2) Lung cancer analysis
(3) Breast cancer database
(4) Phishing websites features
(5) Messidor image set prediction
(6) Sport articles

6. Results and Discussions of Experiments

We used the ANN algorithm with two hidden layers. In addition, validation and training techniques are used to ensure more efficient results. For all datasets, the percent of training is selected to be 40%, the validation percent is 30%, and the testing ratio is also 30%. The parameters that we used in the genetic algorithm in all algorithms are presented in Table 2. In the first experiment, we compared the performance of the five explained algorithms using the six datasets. In this experiment, a population size of 10 and iteration number of 40 are used for all algorithms. The results of this experiment are computed using Equation (1) and presented in Table 3.

Table 3: Classification performance comparison between the different proposed algorithms.

$P = 10$, it = 40	Heart	Lung cancer	WBCD	Phishing	Messidor	Sport articles
Num of features	13	56	9	30	19	59
ANN	0.837	0.250	0.969	0.919	0.686	0.823
GA + ANN	0.859	0.935	0.974	0.927	0.735	0.849
LWGGA + ANN	0.859	0.935	0.974	0.927	0.749	0.854
HWGGA + ANN	0.859	0.968	0.975	0.927	0.748	0.854
WGGA + ANN	0.8667	0.969	0.976	0.927	0.752	0.854

Table 4: The performance of the WGGA + ANN algorithm using different number of generations.

# of generations	Heart	Lung cancer	WBCD	Phishing	Messidor	Sport articles
1	0.837	0.250	0.969	0.919	0.686	0.635
10	0.866	0.912	0.972	0.927	0.735	0.826
20	0.867	0.937	0.972	0.927	0.738	0.849
40	0.870	0.969	0.976	0.927	0.752	0.849
60	0.870	0.969	0.976	0.927	0.752	0.851
80	0.870	0.969	0.976	0.927	0.752	0.854

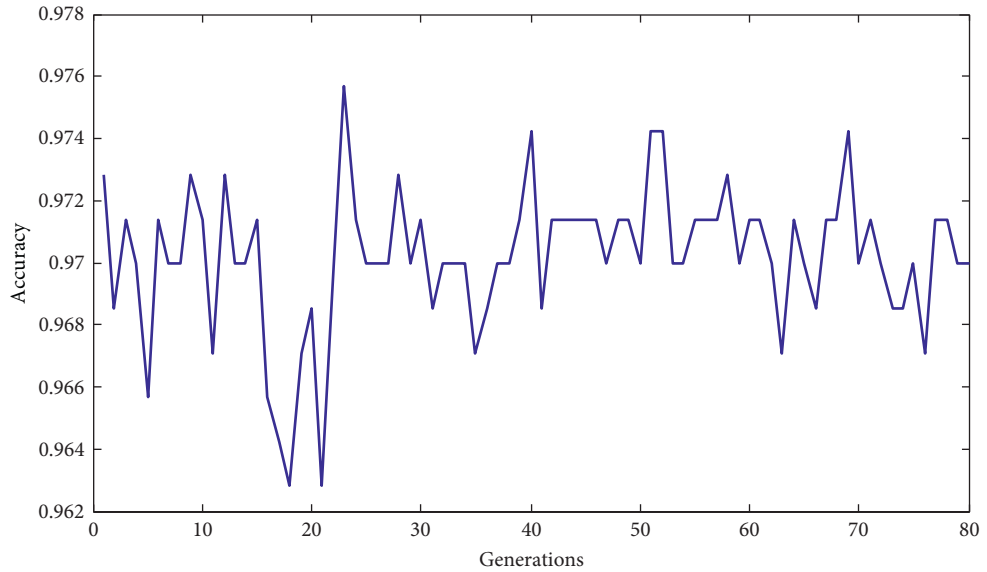

Figure 5: The accuracy of the WGGA algorithm for the WBCD dataset by running 80.

As the table shows, the proposed three algorithms significantly enhance the performance of the ANN algorithm. The best algorithm in all results is the WGGA, since it outperforms all other algorithms in four datasets and gets the same performance with other algorithms in one dataset. The second-best algorithm is the HWGGA algorithm which gets acceptable results in more than two datasets. The results of this table ensure the performance of our proposed weight-based genes mechanism which makes the genetic algorithm concentrates on the flagged features during the search and depends slightly on changing the strong features (which is included in all best solutions).

On the contrary, the other comparing algorithms try to find a better set of features by only randomly selecting different set of features, which get good results but needs long time. The proposed algorithms take some experiences from the first generations of the genetic algorithm, and then it uses these experiences to distinguish between the features that should be always included in the best feature set, the features that should always be excluded from the best feature set, and the feature that are not checked yet. According to this, the proposed algorithm can quickly converge to the best feature set by saving the efforts of searching on the already checked good and bad features and check the other features that are not known yet.

Moreover, the results show that the WGGA algorithm benefits from merging the two mechanisms of LWGGA and HWGGA algorithms which makes it the best algorithm. It is also important to note that the enhancement ratio is varying from one dataset to other; for example, in the Lung cancer dataset, the performance is enhanced very well (from 0.25 to

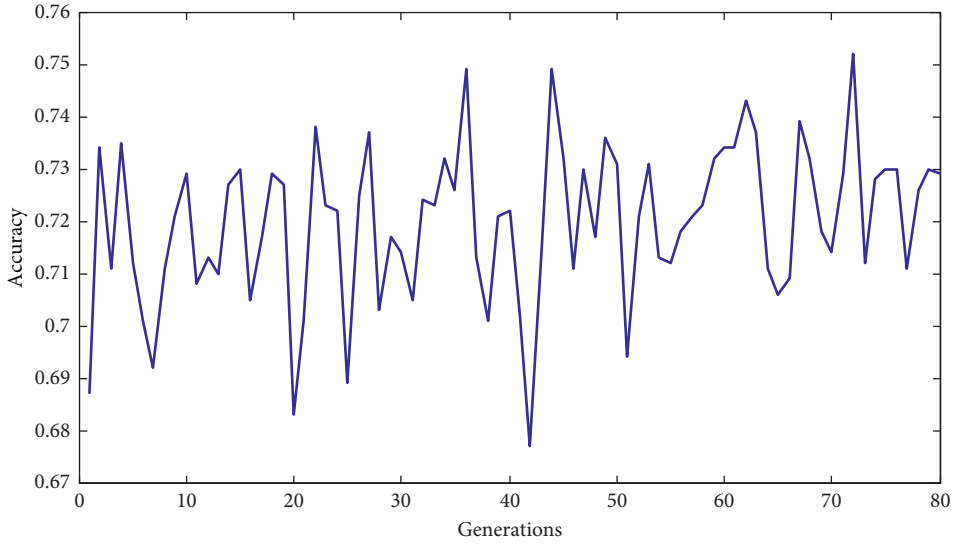

Figure 6: The accuracy of the WGGA algorithm for the Messidor dataset by running 80 generations.

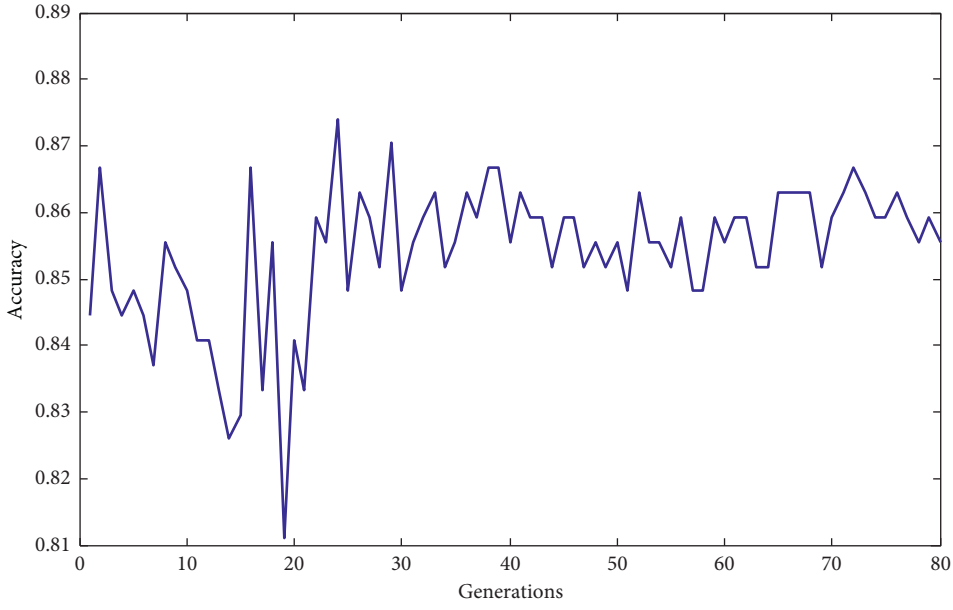

Figure 7: The accuracy of the WGGA algorithm for the WBCD dataset by running 80 generations.

0.969), whereas in the Phishing dataset, it only enhances by a small value (from 0.919 to 0.927).

In the second experiment, we will investigate the effect of increasing the number of generations on the performance of the WGGA algorithm. Five different generation numbers are used from 10 to 80, and the six datasets are tested again; the accuracy is computed using Equation (1). As the results of Table 4 show, the performance of the proposed algorithm can convergence fast, and the accuracy does not enhance significantly after 40 generations in most of the tested datasets. These results again ensure that the incorporated new mechanisms make the optimization process of the genetic algorithm very effective and reach the optimal solution quickly.

Figures 5–10 show the accuracy of the proposed algorithm WGGA for the six datasets when using 80 generations. The figures show the best accuracy in each generation as computed from Equation (1). The figures again ensure the quick convergence of our proposed algorithm which is clear since the best value occurs usually before the 40th generation. We can also note that the algorithms can fluctuate between good and bad values during the evolving process. This means that the best solution in later steps may become worse than the best solution of earlier steps which first seems not good, where in fact it is a good aspect, since it gives the algorithm the ability to search in worst solution to get better solutions. Therefore, we can see from figures that the accuracy becomes bad and then it can get a solution better than all previous solutions as in Figure 1 in the 21 through 25 generations. In addition, we can see that the average of accuracies in last generations is much better than the values

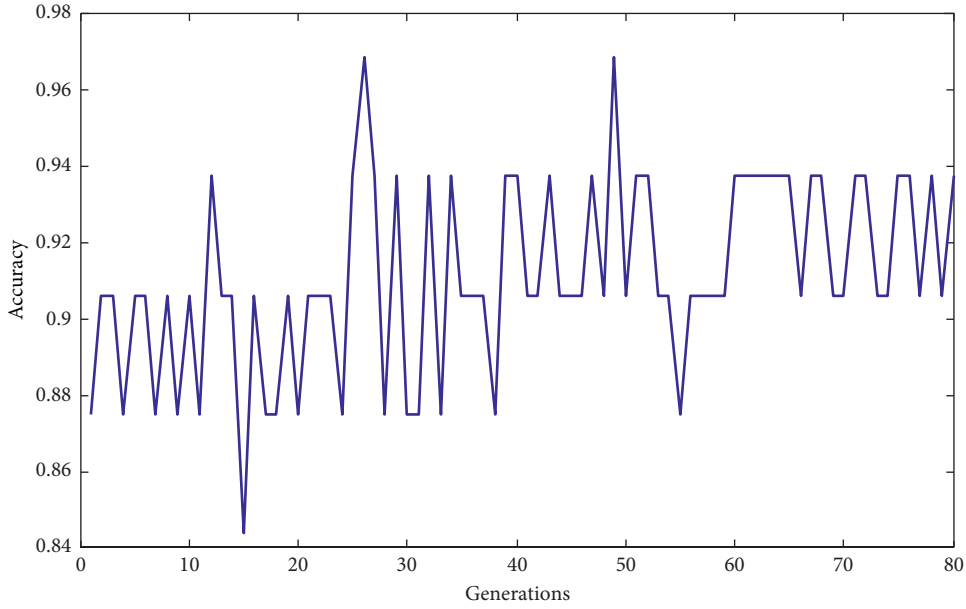

Figure 8: The accuracy of the WGGA algorithm for the Lung cancer dataset by running 80 generations.

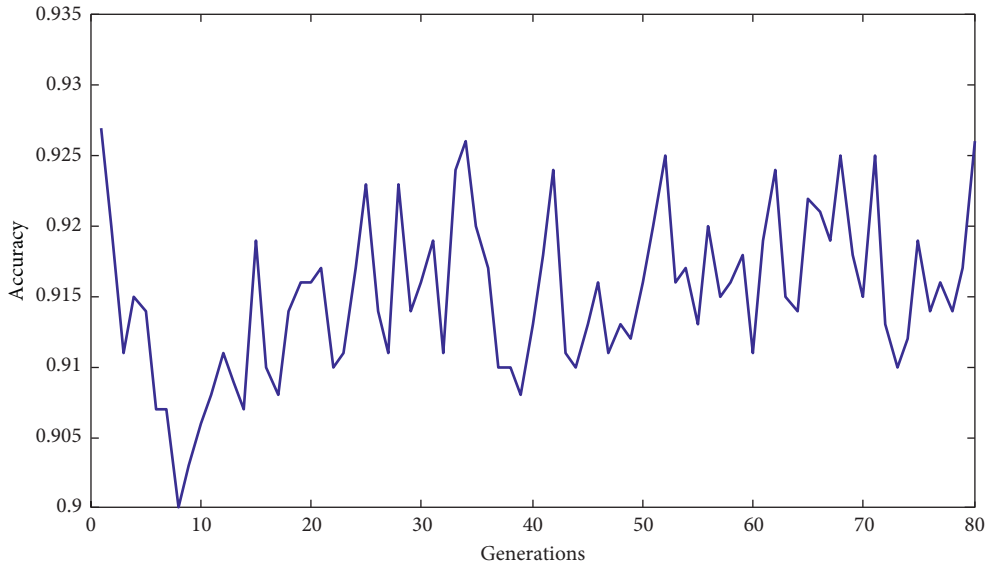

Figure 9: The accuracy of the WGGA algorithm for the Phishing dataset by running 80 generations.

of earlier generations which more emphasize the effectiveness of our proposed algorithm.

As in any algorithm, there are some few drawbacks of the proposed algorithm. Firstly, since our algorithm needs to gather some information regarding the features, it starts activating its weighted gene mechanism after number of generations (after collecting the needed data). This process takes some time but it can be neglected when the number of generations is big. Secondly, in dynamic environment, the proposed algorithm may not work efficiently especially in very fast changing environments. This is because the proposed algorithm depends on previous static experience, whereas in dynamic environments, this experience becomes unimportant because of frequent changing.

7. Conclusion and Future Work

To overcome the big data complexity problem, the feature reduction becomes one of the most effective solutions that are used nowadays. In this paper, a set of hybrid and efficient algorithms are proposed to classify the datasets that have large feature size by merging genetic algorithms with the artificial neural networks. The genetic algorithms are used as a prestep to significantly remove the irrelevant features from the datasets before handling that data using machine learning techniques. Three new genetic algorithms are proposed and incorporated in the ANN algorithm which is low weighted gene genetic algorithm (LWGGA), high weighted gene genetic algorithm (HWGGA), and weighted gene genetic

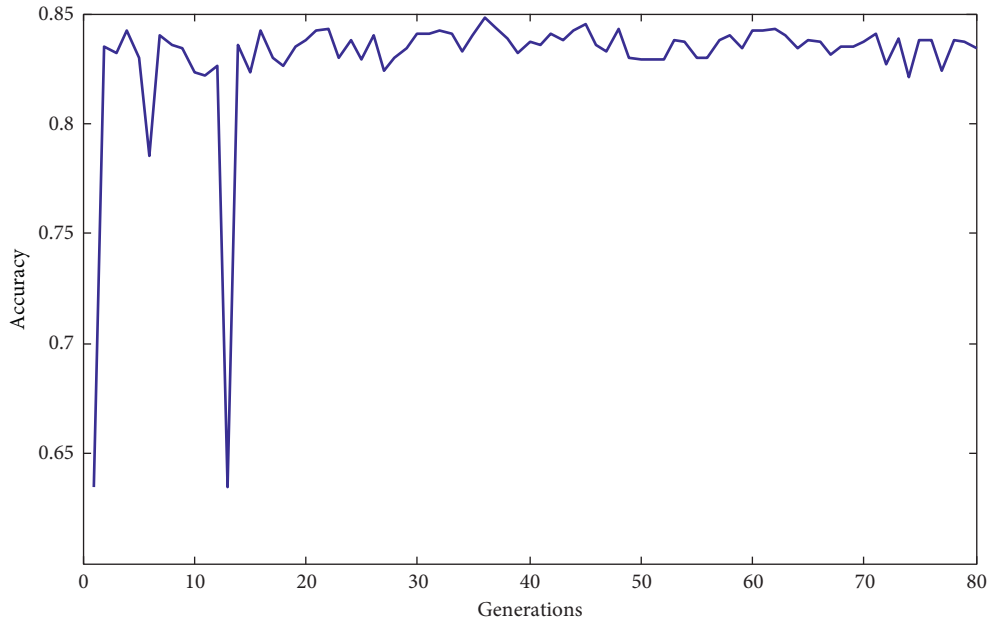

Figure 10: The accuracy of the WGGA algorithm for the Sport dataset by running 80 generations.

algorithm (WGGA). The proposed algorithms use a new gene-weight mechanism that can significantly enhance the performance and decrease the required search time. The proposed algorithms are applied on six datasets to pick the most relative and important features before applying the artificial neural networks algorithm, and the results show that our proposed algorithms can effectively enhance the classifying performance over the tested datasets.

In future work, we are planning to compare our proposed algorithms against more evolutionary algorithms such as PSO and ACO. At the same time, the new proposed weight-gene mechanism can be merged with other algorithms. We expect that this mechanism may get better results if it is checked using other evolutionary algorithms.

Data Availability

Datasets are available in UCI Machine Learning Repository at http://archive.ics.uci.edu/ml.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

References

[1] S. J. Walker, "Big data: a revolution that will transform how we live, work, and think," *International Journal of Advertising*, vol. 33, no. 1, pp. 181–183, 2014.

[2] J. Manyika, M. Chui, B. Brown et al., *Big Data: The Next Frontier for Innovation, Competition, and Productivity*, McKinsey, New York, NY, USA, 2011.

[3] P. Zikopoulos and C. Eaton, *Understanding Big Data: Analytics for Enterprise Class Hadoop and Streaming Data*, McGraw-Hill Osborne Media, New York, NY, USA, 2011.

[4] A. Katal, M. Wazid, and R. H. Goudar, "Big data: issues, challenges, tools and good practices," in *Proceedings of 2013 Sixth International Conference on Contemporary Computing (IC3)*, pp. 404–409, IEEE, Noida, India, August 2013.

[5] C. P. Chen and C. Y. Zhang, "Data-intensive applications, challenges, techniques and technologies: a survey on Big Data," *Information Sciences*, vol. 275, pp. 314–347, 2014.

[6] X. Wu, X. Zhu, G. Q. Wu, and W. Ding, "Data mining with big data," *IEEE Transactions on Knowledge and Data Engineering*, vol. 26, no. 1, pp. 97–107, 2014.

[7] Z. Obermeyer and E. J. Emanuel, "Predicting the future—big data, machine learning, and clinical medicine," *New England Journal of Medicine*, vol. 375, no. 13, pp. 1216–1219, 2016.

[8] R. Ho, *Big Data Machine Learning*, 2012.

[9] I. H. Witten, E. Frank, M. A. Hall, and C. J. Pal, *Data Mining: Practical Machine Learning Tools and Techniques*, Morgan Kaufmann, Burlington, MA, USA, 2016.

[10] G. George, M. R. Haas, and A. Pentland, "Big data and management," *Academy of Management Journal*, vol. 57, no. 2, pp. 321–326, 2014.

[11] M. Dash and H. Liu, "Feature selection for classification," *Intelligent Data Analysis*, vol. 1, no. 1–4, pp. 131–156, 1997.

[12] I. Guyon and A. Elisseeff, "An introduction to variable and feature selection," *Journal of Machine Learning Research*, vol. 3, pp. 1157–1182, 2003.

[13] H. Liu and Z. Zhao, "Manipulating data and dimension reduction methods: feature selection," in *Encyclopedia of Complexity and Systems Science*, pp. 5348–5359, Springer, Berlin, Germany, 2009.

[14] H. Liu, H. Motoda, R. Setiono, and Z. Zhao, "Feature selection: an ever evolving frontier in data mining," in *Proceedings of Fourth International Workshop on JMLR Feature Selection in Data Mining*, vol. 10, pp. 4–13, Hyderabad, India, June 2010.

[15] A. W. Whitney, "A direct method of nonparametric measurement selection," *IEEE Transactions on Computers*, vol. C-20, no. 9, pp. 1100–1103, 1971.

[16] T. Marill and D. M. Green, "On the effectiveness of receptors in recognition systems," *IEEE Transactions on Information Theory*, vol. 9, no. 1, pp. 11–17, 1963.
[17] S. D. Stearns, "On selecting features for pattern classifier," in *Proceedings of 3rd International Conference on Pattern Recognition*, pp. 71–75, Coronado, CA, USA, November 1976.
[18] P. Pudil, J. Novovičová, and J. V. Kittler, "Floating search methods in feature selection," *Pattern Recognition Letters*, vol. 15, no. 11, pp. 1119–1125, 1994.
[19] Q. Mao and I. W.-H. Tsang, "A feature selection method for multivariate performance measures," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 35, no. 9, pp. 2051–2063, 2013.
[20] F. Min, Q. Hu, and W. Zhu, "Feature selection with test cost constraint," *International Journal of Approximate Reasoning*, vol. 55, no. 1, pp. 167–179, 2014.
[21] H. Frohlich, O. Chapelle, and B. Scholkopf, "Feature selection for support vector machines by means of genetic algorithm," in *Proceedings of 15th IEEE International Conference on Tools with Artificial Intelligence, 2003*, pp. 142–148, Sacramento, CA, USA, November 2003.
[22] I. S. Oh, J. S. Lee, and B. R. Moon, "Hybrid genetic algorithms for feature selection," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 26, no. 11, pp. 1424–1437, 2004.
[23] H. M. Zawbaa, E. Emary, C. Grosan, and V. Snasel, "Large-dimensionality small-instance set feature selection: a hybrid bioinspired heuristic approach," *Swarm and Evolutionary Computation*, vol. 42, pp. 29–42, 2018.
[24] E. Emary and H. M. Zawbaa, "Feature selection via Levy Antlion optimization," *Pattern Analysis and Applications*, pp. 1–20, 2018.
[25] W. Yamany, H. M. Zawbaa, E. Emary, and A. E. Hassanien, "Attribute reduction approach based on modified flower pollination algorithm," in *Proceedings of International Conference on Fuzzy Systems (FUZZ-IEEE)*, pp. 1–7, Istanbul, Turkey, August 2015.
[26] B. Xue, M. Zhang, and W. N. Browne, "Particle swarm optimization for feature selection in classification: a multi-objective approach," *IEEE Transactions on Cybernetics*, vol. 43, no. 6, pp. 1656–1671, 2013.
[27] T. M. Hamdani, J. M. Won, A. M. Alimi, and F. Karray, "Multi-objective feature selection with NSGA II," in *Proceedings of International Conference on Adaptive and Natural Computing Algorithms*, pp. 240–247, Springer, Berlin, Heidelberg, April 2007.
[28] L. S. Oliveira, R. Sabourin, F. Bortolozzi, and C. Y. Suen, "Feature selection using multi-objective genetic algorithms for handwritten digit recognition," in *Proceedings of 16th International Conference on Pattern Recognition, 2002*, vol. 1, pp. 568–571, IEEE, Quebec City, Canada, August 2002.
[29] M. Morita, R. Sabourin, F. Bortolozzi, and C. Y. Suen, "Unsupervised feature selection using multi-objective genetic algorithms for handwritten word recognition," in *Proceedings of Seventh International Conference on Document Analysis and Recognition, 2003*, pp. 666–670, IEEE, Edinburgh, UK, August 2003.
[30] D. Dua and E. K. Taniskidou, *UCI Machine Learning Repository*, University of California, School of Information and Computer Science, Irvine, CA, USA, 2017, http://archive.ics.uci.edu/ml.