A Novel Approach for Feature Selection: Evolutionary Algorithm

Weiheng Hong S3643760
Supervisor: Yuan Sun

ABSTRACT An even more intelligent approach to search strategy is desirable in feature selection. To enhance its performance relies on the search strategy and evaluation criteria. We demand a more robustness approach to achieve higher accuracy in feature selection. A study suggests us to use Evolutionary Computing as the search strategy and wrap together with several classification techniques. Our experiment will design three feature selection model based on Genetic Algorithm, Particle Swarm Optimization, and Genetic Algorithm Greedy respectively to discuss their performance in terms of robustness, efficiency, and accuracy.

INDEX TERMS Evolutionary Computing, Genetic algorithm, Particle Swarm Optimization, Feature selection.

I. INTRODUCTION

In the machine learning tasks, the number of features in a given dataset can be extremely large. For example, in the high-throughput gene expression data, the number of features on a single array can be 1E+4 to 1E+6. And most of these features are typically irrelevant and redundant, which significantly slow down the training process and degrade the training accuracy. A logical way to deal with this issue is by dimensionality reduction that we can select a subset with the most relevant features for training and testing, which can highly decrease the training time and improve the accuracy. In this way, such a feature selection problem can be easily modeled as an unconstraint optimization problem with binary variables, which is suitable for evolutionary algorithms (EAs) to solve.

Several attribute selection filters have been used in data mining before, which can base on many rules. They select a subset of feature for training and testing. This highly reduces the time when doing the training, but the accuracy is not satisfactory enough since it ignores the performance of selected features on a classification algorithm. In this circumstance, using wrappers evaluation approach for feature selection base on evolutionary algorithm which can achieve the robustness is highly desirable.

The goals of this project are:

1) Modeling feature selection problems as optimization problems;
2) Selection an EC and adapting it for the feature selection tasks;
3) Evaluating the proposed feature selection model using multiple benchmark datasets and classifiers;
4) Comparing the proposed method against state-of-the-art models.

II. Literature Review

Feature selection has been the focus of the machine learning and data mining area for the past few years. By removing the redundant attributes, feature selection can reduce the dimensionality of the data, speed up the machine learning process, and increase the learning performance [1].

However, conventional exhaustive search is not suitable for such a great number of features. In order to improve the efficiency of feature selection, a good search strategy is necessary to be utilized in the feature selection. The experiment indicates the current feature selection models are using the search strategy like random search, greedy search, and heuristic search [1]-[5], but most of these existing feature selection methods still have global optima problem and high consuming problem [6]. Therefore, an efficient global search application for better solving feature selection problems is desirable.

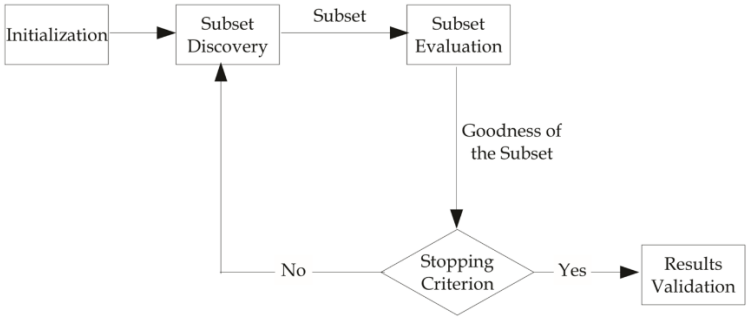


Figure 1. General feature selection process [1].

A. General Feature Selection

Figure 1 shows a general feature selection process includes five key steps:

1) Initialization: prepare the processed data as input and set the parameters with desirable value.

2) Subset Discovery: select the minimally sized feature subset that is necessary and sufficient to the target concept. The subset discovery is to generate the next candidate subset.

3) Subset Evaluation: evaluate the subset under examination by using an evaluation function.

4) Stopping Criterion: decide when to stop. Without a suitable stopping criterion, the feature selection process may run forever through the search space of subsets. Stopping criteria is based on the generation procedure defined by the input parameter in the Initialization step.

5) Results Validation: check whether the subset is valid. The feature selection process halts by outputting a selected subset of feature to the validation procedure.

Based on the evaluation criteria, feature selection algorithms can be classified into two categories: 1) filter approaches and 2) wrapper approaches [1], [8]. The main difference is that wrapper approach using a classification algorithm to evaluate the goodness of the selected feature subset, while a filter feature selection process is independent of any classification algorithm, which is computationally less expensive than wrapper algorithms. However, filter feature selection is inferior to the wrapper approach since it ignores the performance of the selected feature on a classification algorithm, whereas wrappers evaluate the feature subsets based on classification performance [1]-[3].

Feature selection is difficult as the number of features creates a large search space and the feature interaction can happen in two-way, three-way or complex multiway interactions. A feature, which is weakly relevant to the target feature, could influence the classification accuracy significantly if it is used together with some complementary features. So, in this case, the removal of such features may miss the optimal feature subset [7].

B. Existing Feature Selection

Many traditional feature selection methods evaluate features individually cannot work well since a feature subset is also necessary to be evaluated as well. The two key points in feature selection are the *search strategy*, which explores the search space to find the optimal feature subset(s), and *evaluation criteria*, which measure the quality of feature subsets to guide the search.

1) For the existing search strategy, very few feature selection methods that use an exhaustive search [1], [3], [4], while most of feature selection models apply heuristic search strategy and algorithms, such as greedy search strategy[9], [12], two-layer cutting plane algorithm [10], and backtracking algorithm [11]. The experiment shows that the heuristic search strategy performs similar to the backtracking algorithm while it used a much shorter time.

Recently, EC techniques are utilized in feature selection problems. EC techniques provide two advantages compared with traditional search methods. Firstly, it is not necessary for EC techniques to have domain knowledge and make an assumption about the

search space, such as whether the search space is linearly or nonlinearly separable, and differentiable. Secondly, the population-based mechanism can produce multiple solutions in a single run, which allow to select a set of a nondominated solution with the trade-off between the number of features and classification performance. However, EC search strategy usually causes a high computational cost as it involves a large number of evaluations. In addition, since the algorithm often select different features from different runs, which may require a further selection process for the users. In this research, we will try to address these issues to decrease the computational cost and increase the stability of the algorithm.

2) For wrapper feature selection approaches, the accuracy of classifiers of the selected features is used as the evaluation criteria. Many popular classifiers, such as Decision Tree, K-Nearest Neighbor, neural network, Logistic Regression, and Linear Regression, have been applied in the feature selection [2], [3]. Experiments indicate various classifier performs differently in a dataset and there is no one classifier that can always perform the best for different datasets. For the filter approach, various disciplines have been applied, include correlation measures, distance measures, and consistency measures.

Xue et al. have presented a comprehensive survey of several Evolutionary computing technologies for feature selection [7] and achieve some success, but they also point the potential ability of evolutionary computing has not been explored fully. The most important issue to the feature selection models is the scalability when employing in the real-world task, as both the number of the instances and the number of features are increasing. This research will try to investigate the ability of EAs and design the EC based feature selection model to solve this problem.

The literature review is organized as follows. As Evolutionary algorithm has been received much attention from the data mining and machine learning field recently, so in this research paper, three feature selection models based on the different evolutionary algorithms and fitness-calculate methods are presented in Section III, where each subsection introduces a particular EC technique for feature selection. Section IV describes the experiment methodology used to test three feature selection models. After that, the experiment results and comparison are presented in Section V, then analysis using statistical test in Section VI. The application of EC for feature selection are described in Section VI.

III. Model Design

Genetic Algorithm (GA) and Particle Swarm Optimization (PSO), as the most popular EC techniques, have been used widely in many fields. Recently, more and more feature selection models based on GAs and PSO have been witnessed with significant achievement. In this section, we will introduce three feature selection models designed to my best knowledge of GA and PSO.

To design a feature selection model based on GA and PSO, firstly, we need to think about what technology we can use to represent a feature subset as an individual. Consider about the same characteristic of binary string and only two option of each feature to select or unselect, we decide to use binary string as an individual to represent a feature string.

A) Feature Selection Based on Genetic algorithm (GA)

For the genetic algorithm, it starts with using an initialized population. By perform crossover of each two individual, each of them mutates according to the mutation rate. After that, select a group of best individuals based on its fitness from the offspring to form a new population for the next generation. After executing with a fixed number of generations as a stop criterion, it gives the best result.

According to the basic GA, we design the feature selection model based on GA includes six steps:

1) select initial genes by randomly generating a fixed number of individuals as the initial population.

2) use this population to generate the same number of crossover group by using every 2 individuals for

randomly crossover.

3) from every crossover group, randomly select 1 bit to mutate, and get the same number of mutation group, which is the offspring.

4) calculate the fitness of each 2 parents and their 2 offspring, then select 2 best individuals with the highest fitness among 4 individuals as a part of the new population.

5) save the gene with the best, the fitness of the maximum, minimum and average of each generation and print a report, then do the next generation.

6) end with all generations, print the final result includes Running time, Best accuracy/error, individual, Number of features and Feature subset(s).

The mechanism of the GA feature selection model preserves the different feature subsets with the same of the best fitness, which allows to further observation and select the most desirable feature subset.

B) Feature Selection Based on Particle Swarm Optimization (PSO)

For the particle swarm optimization, regards each individual of feature subset as a particle, it starts with generating a population of particle. While setting the particle with the best fitness among the population as the global best (p_{gd}), and setting the local best (p_{id}) from the visiting positions in the track of each particle, each particle's movement is influenced by the global best and its local best according to PSO functions:

$$v_{id+1} = w * v_{id} + c_1 * rand(0,1) * (p_{id} - x_{id}) + c_2 * rand(0,1) * (p_{gd} - x_{id}) \quad (1)$$

$$x_{id+1} = x_{id} + v_{id} \quad (2)$$

where the x_{id} represents the position of a particle and v_{id} represents its current velocity, while w is the inertia weight, c_1 and c_2 are acceleration constants. According to the basic PSO, we design the feature selection model based on PSO includes five steps:

1) randomly generate a population of particles with initial velocity to zero.

2) keep the particle with the best fitness as global best.

3) according to PSO functions, each particle keeps pointing toward a direction, and save the local best in its track.

4) save the global best fitness and its position (individual) in every iteration, then print a report.

5) end with all iterations, return the best individual and print a final report includes Running time, Best accuracy/error, individual, Number of features and Feature subset(s).

Observation and experience to the velocity, we set the inertia weight to 0.01 and both acceleration constants to 0.02 in our experiments.

C) Feature Selection Based on Generic Algorithm Greedy (GA Greedy)

While GA feature selection model calculates the fitness of both parent and offspring in every generation, which takes twice longer than PSO feature selection model. We design a new feature selection based on GA Greedy include seven steps:

1) select two initial genes (parents), which is all selected and all unselected.

E.g. [1,1,1,1,1] & [0,0,0,0,0]

2) generate a population by using these 2 parents for various crossover and get a number of individuals to form a crossover group.

3) randomly select 2 individuals from crossover group for mutation, each time mutate 1 bit from the individual and save in a mutation group.

4) select parts of individuals from crossover group and mutation group according to a ratio.

5) calculate the fitness of the population, save the gene with the best and second-best, the fitness of the maximum, minimum and average of each generation and print a report.

6) best individual and second-best individual as 2 parents for the next generation.

7) end with all generations, print the final result includes Running time, Best accuracy/error, individual, Number of features and Feature subset(s).

Method of Fitness Calculation:

To evaluation criteria, we adopt wrapper the feature selection approach. During the Python3 working

environment, we apply the classification and loss-function calculate method from Scikit-learn library.

For numeric value prediction, which the target is numeric data, we employ Linear Regression to predict the value. Then use three kinds of loss-function calculate method to calculate the error between the predicted value and test value include Mean absolute error, Root mean squared error and Median absolute error. Each calculate methodology uses together with 10-fold-cross-validation to get the fitness, which divides the data into 10 folds, train 9 folds and test the rest one for each of the fold for 10 times, then get the average of 10 results.

For nominal (categorical) value prediction, which the target is nominal, we mainly employ three classifiers for prediction, include Logistic Regression, K-nearest-neighbors Classifier (k=5), and Decision Tree Classifier. Each of them also uses together with 10-fold-cross-validation to get the fitness.

IV. Methodology

To test these three feature selection models, we mainly utilize Automobile Dataset, Adult Dataset, and Bank Marketing Dataset as our benchmark. These datasets are available online at UCI [13]. In the next three subsections, we firstly describe some detail about how we prepare these datasets, next demonstrate the experiment methodology we used in terms of numeric data prediction and nominal data prediction.

A) Data Preparation

1) for the Automobile Dataset, which includes 238 instances and 26 features which the last one is numeric target feature, a lot of issues need to process before doing the task, such as typing error, extra whitespaces, impossible values, and missing data. Firstly, there are 10 features include typing error, where they all include capital letters in the data. To deal with it, we transform them to the lowercase. Another kind of typing error that includes the repeated letter or unrelated number, such as 'turrrrbo', 'vol00112ov'. We found this kind of error in three features and refer to the data description document and replace them with their right form. Secondly, for the extra whitespaces, it can be quickly cleaned through 'str.strip()' method. Next, there are 3 impossible values include in these features:

• *symboling*, which includes the value out of the range, 4, is replaced with maximum value, 3.

• *normalized-losses*, which include the value out of the range, 25. I guess maybe it is typing error so replace it with 65.

• *price*, which includes 0.0 and it is not possible as price. Observation from this feature data, I found 0.0 is a little far away from the minimum value which is 5118. And the distribution of this data is range from 5118 to 45400 evenly, although most of them are bias to 5118. So, I replace this value with the median of this data.

Finally, for the missing value, which takes place in 7 features. Since each of these attributes is numeric and distributed evenly, I fill these missing values with their median. The reason why I use the median rather than mean is median can always better than mean, since in some case when the distribution of the data bias to one side that median is better than mean. Although we believe by using K-Nearest-Neighbor (KNN) Imputation to handle the missing value can improve the accuracy of data prediction, using this method may bias the predicted result to one classifier, which is unfair to compare different classifiers and choose which can perform best.

Since only 238 instances included in this dataset, it is not necessary to extract a part of data as a sample for our experiment and we apply with the whole dataset.

2) for the Adult Dataset, which includes 48842 instances and 15 features which the last one is nominal target feature. Consider about testing with all the dataset for our feature selection model to implement once may take more than 24 hours, and the missing value(s) happens in some instances, we separate the dataset into Missing dataset, which all instances include missing value(s), and Completed dataset, which no missing value happens in all

instances. Then, we randomly extract 1000 instances from the Completed dataset as our experiment sample. 3) for the Bank Marketing Dataset, which includes 45211 instances and 17 features which the last one is nominal target feature. Since no data issue happens in this dataset and the huge number of instances can significantly affect the processing speed based on our experiment laptop (Dell XPS 9560), we extract 1000 instances for the dataset as our experiment sample.

All the data is processed for the experiment, then we present the experiment approach according to the characteristic of the target feature in the next two subsections.

B) Numeric Data Prediction

For the Automobile Dataset, which the target feature is numeric, we apply Linear Regression Classifier running with three loss functions respectively to calculate the fitness. Each loss function, we execute it with three EC feature selection models separately, and each feature selection model implements 20 times. To the parameter setting, we uniformly set the population to 50 and generation/iteration to 100. Next, we get the average of statistic result in terms of Error of all feature, Convergence generation, Output error, and Running time. After that, we select the best loss function group according to the Convergence generation and measure the independence of each EC feature selection method using the statistical test in the Analysis section. We also run a test on the best loss function group for each EC feature selection model with the population maintain 50 but the generation change to 200.

C) Nominal Data Prediction

For the Adult Dataset and Bank Marketing Dataset, which both target features are nominal, we utilize three classifiers to calculate the fitness. Each classifier, we execute it with three EC feature selection models separately, and each feature selection model implements 20 times. To the parameter setting of Adult Dataset, we uniformly set the population to 30 and generation/iteration to 100. To the parameter setting of Bank Marketing Dataset, we uniformly set the population to 40 and generation/iteration to 100. Next, we get the average of statistic result in terms of Error of all feature, Convergence generation, Output accuracy, and Running time. After that, we select the best classifier according to the Output accuracy and measure the independence of each EC feature selection method using the statistical test in the Analysis section. We also run a test on the best classifier group for each EC feature selection model with the original population value but change the value of generation to 200. Since different datasets may have a different most-suitable classifier, we need to keep testing dataset under different classifiers.

V. Experiment and Comparison

Running 3 EC feature selection models with different classifiers and loss functions, we choose the best working environment to present our experiment results, which Automobile dataset converges fastest with Linear Regression classifier under root mean square error (RMSE) function, Adult dataset can achieve the best accuracy with Decision Tree classifier, and Bank Marketing dataset get the best accuracy when running with K-Nearest-Neighbors classifier.

A) Experiment Result 1

While using RMSE with Linear Regression classifier in Automobile dataset spends fewer generations to converge than the other loss functions, Table 1 shows the average of the results from 20 times based on 50 population and 100 generations. Compare with the Error of all features in 4316.62, GA model gives the final RMSE in 3657.10, which is the lowest RMSE among 3 EC feature selection models, but it takes twice as long as PSO model and GA Greedy model. It is worth mentioning that the GA Greedy converges at 16.2^{th} generation in average which is the fastest and it also has better output RMSE than PSO model, although it cannot always get the optimal feature subset. After that, we change the input generation to 200, preserve the setting of the other parameters, and

run a test for each of the models.

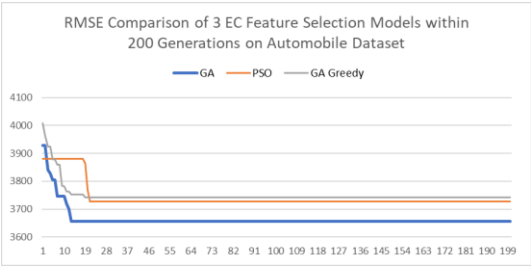
Automobile LinearRegression RMSE	Population and Generation / Iteration	Error of All Features	Convergence generation	Output Error	Running time (second)
Average from 20 tests					
GA	50, 100	4316.61	27.95	3657.10	67.369
PSO	50, 100	4316.61	42	3758.20	34.86
GA Greedy	50, 100	4316.61	16.2	3739.15	31.63
Run a test					
GA	50, 200	4316.61	13	3656.74	130.45
PSO	50, 200	4316.61	21	3726.63	63.62
GA Greedy	50, 200	4316.61	19	3742.04	63.04

Table 1. Experiment Result of Automobile Dataset Testing with Three EC Feature Selection Models Using Linear Regression Classifier and Root Mean Square Error Loss Function

As we can see from Table 1, where three EC feature selection models run a test with 200 generations, GA model still outputs the lowest RMSE as result, while PSO model and GA Greedy model have similar final RMSE. Then, we plot a line graph to show the RMSE comparison of 3 EC feature selection models in Graph 1. As we can see, GA model with the lowest RMSE from generation 2 to generation 200 and reach convergence at 3656.74 in generation 13 with the selected feature subset:

['make', 'aspiration', 'body-style', 'drive-wheels', 'engine-location', 'length', 'height', 'engine-type', 'engine-size', 'stroke', 'compression-ratio', 'peak-rpm']

PSO model converges at 21th generation but cannot find the optimal value since the value setting of the inertia weight and acceleration constants might be too large to this dataset in this running environment that it neglects the optimal feature subset, while GA Greedy model is too greedy to find the optimal solution that it converges early in a region but overlooks the other regions.



Graph 1. Root Mean Square Error Comparison of 3 Evolutionary Computing Feature Selection Models with Linear Regression Classifier and Root Mean Square Error Loss Function within 200 Generations on Automobile Dataset

B) Experiment Result 2

While using Decision Tree classifier in Adult dataset give the best accuracy compare with the other classifiers, Table 2 presents the average of the results from 20 times based on 30 population and 100 generations. GA model achieves the highest accuracy at 0.8523 among 3 EC feature selection models but takes a lot of running time than the others. GA Greedy model uses least running time, converges at the earliest, gives the lowest output accuracy since it is too greedy. The results of PSO model are between GA model and GA Greedy. Then we change the input generation to 200, keep the value of the other parameters, and run each of the models once.

Adult Decision Tree Classifier	Population and Generation / Iteration	Accuracy of All Features	Convergence generation	Output Accuracy	Running time (second)
Average from 20 tests					
GA	30, 100	0.7658	45.8	0.8523	143.67
PSO	30, 100	0.7688	37.95	0.8434	95.99
GA Greedy	30, 100	0.7608	35.6	0.8375	72.95
Run a test					
GA	30, 200	0.7688	52	0.8530	245.04
PSO	30, 200	0.7728	51	0.8400	163.47
GA Greedy	30, 200	0.7738	16	0.8380	118.43

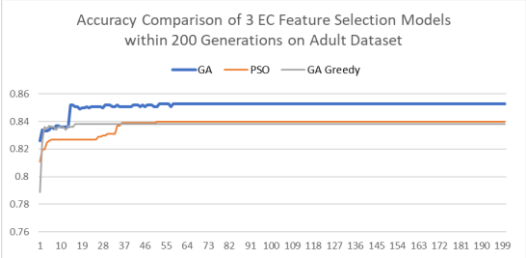
Table 2. Experiment Result of Adult Dataset Testing with Three EC Feature Selection Models Using Decision Tree Classifier

As Table 2 shows, where three EC feature selection models run a test with 200 generations, GA model keeps the best convergence accuracy at 0.8530 and GA Greedy model spends least generation to converge. A line graph is plotted to show the tracks of accuracy in 3 EC feature selection models in Graph 2. Since the Decision Tree classifier form Scikit-learn library gives different accuracy on the same feature subset each time, the accuracy does not always keep increasing form linear graph, but we still can see GA model performs the best among three feature selection models that it achieves the best accuracy at 15th generation and keep in the dominant stage in the further generations. It finally outputs with the optimal feature subset:

['workclass', 'marital-status', 'occupation', 'capital-gain', 'capital-loss']

The other models still fail to find the optimal feature string since the large value of the inertia weight and

acceleration constants in PSO model and the greed of GA Greedy model that both of them may have overlooked the best result.



Graph 2. Accuracy Comparison of 3 Evolutionary Computing Feature Selection Models with Decision Tree Classifier within 200 Generations on Adult Dataset

C) Experiment Result 3

As K-Nearest-Neighbor classifier can give the best accuracy in Bank Marketing dataset, we present the average result from 20 times based on 40 population and 100 generations in Table 3. GA model gets the highest accuracy at 0.9071 among 3 EC feature selection models, however, it spends twice as long as GA Greedy model to run a test. In addition, GA Greedy model converges at 5.05th generation in average, compare with GA model converges at 24.55th generation in average, although there is not a very big difference on output accuracy between these two models. But it still no evidence to prove GA Greedy model is superior to GA model since GA model still dominates the optimal feature subset according to its highest accuracy. So, in the next section, we will further discuss this question using statistic test.

Bank Marketing K-Nearest-Neighbors Classifier	Population and Generation / Iteration	Accuracy of All Features	Convergence generation	Output Accuracy	Running time (second)
Average from 20 tests					
GA	40, 100	0.879	24.55	0.9071	2843.61
PSO	40, 100	0.879	38.55	0.9036	1387.54
GA Greedy	40, 100	0.879	5.05	0.9067	1439.52
Run a test					
GA	40, 200	0.879	26	0.9071	4647.34
PSO	40, 200	0.879	170	0.9061	2546.09
GA Greedy	40, 200	0.879	3	0.9071	2464.70

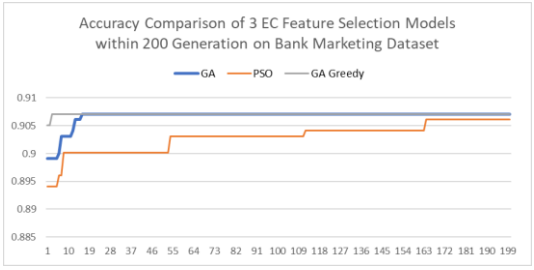
Table 3. Experiment Result of Bank Marketing Dataset Testing with Three EC Feature Selection Models Using K-Nearest-Neighbors Classifier

Then, by changing the input generation to 200, keeping the values of the other parameters, we run a test on this model. As Table 3 shows, where three feature selection run a test with 200 generations, GA model and GA Greedy model output the highest accuracy at the same time, but GA Greedy converge a 3rd generation, while GA model converges much slower. The line graph of the accuracy track of 3 models within 200 generations is plotted in Graph 3. GA Greedy model dominates the optimal stage in all generations and output with 2 best feature subsets:

(1) ['campaign', 'poutcome']

(2) ['default', 'campaign', 'poutcome']

which both of them achieve the same best accuracy. It is worth mentioning that PSO model cannot find the optimal solution since it still keeps exploring the better in all generation and does not truly arrive its convergence generation among all generations. This phenomenon happens since the inertia weight and acceleration constants of this model are too small for this dataset to converge in a reasonable range of generation.



Graph 3. Accuracy Comparison of 3 Evolutionary Computing Feature Selection Models with Decision Tree Classifier within 200 Generations on Bank Marketing Dataset

VI. Analysis

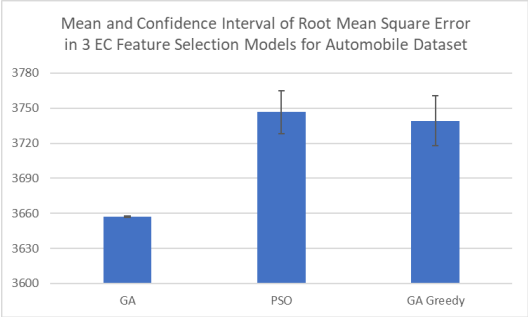
Analysis in this section applied in the 3 benchmark datasets of 3 different EC feature selection models with their best working environment. A statistical test is used based on their 20 tests result. The confidence level is set at 95% to calculate the confidence interval during the test.

A) Analysis of Experiment 1

As Graph4 shows, by comparing the confidence interval of RMSE in GA model, PSO model, and GA Greedy model, GA model do not overlap with the other models, which means there is indeed a statistically significant difference between the mean of

GA model and the mean of PSO model, the mean of GA model and the mean of GA Greedy model. For the PSO model and the GA greedy, the confidence interval of RMSE in these two models almost overlap with each other with P value in 0.0676 in T-test, which indicates no statistical significance between the means of two models.

The result indicates the comparison in Experiment 1 of the last section is reasonable that GA feature selection model achieves the best performance in the Automobile dataset among three feature selection models.

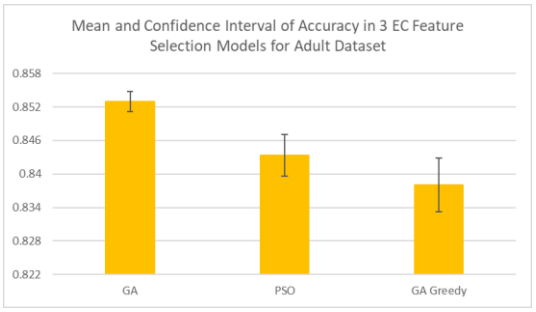


Graph 4. Mean and Confidence Interval of Root Mean Square Error in Three Evolutionary Computing Feature Selection Models for Automobile Dataset

B) Analysis of Experiment 2

From Graph 5, no overlap of the confidence interval of accuracy happens between GA model and PSO model, GA model, and GA Greedy model, while comparison result from Experiment 2 from the last section indicates GA has the best output result in average. Therefore, it can be proved with a statistically significant difference between GA model and each of the other models that GA feature selection model achieves the best performance in the Adult dataset among three feature selection models.

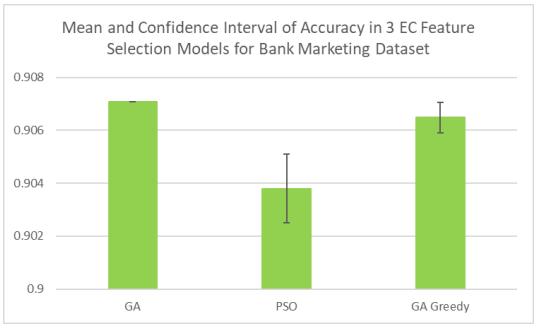
By using T-test between PSO model and GA Greedy, the P value is obtained with 0.0120, which means there still have a slight difference between the means of these two models. Therefore, according to the result from Experiment 2, GA Greedy model performs the worst among three EC feature selection model since it is too greedy to find the best feature subset.



Graph 5. Mean and Confidence Interval of Accuracy in Three Evolutionary Computing Feature Selection Models for Adult Dataset

C) Analysis of Experiment 3

Mean and confidence interval of accuracy in three EC feature selection models for Bank Marketing Dataset is presented in Graph 6. While there is an overlap happened between the GA model and GA Greedy model, we use T-test to further check whether there is a statistical significance between the two models. As the P value obtained with 0.1036 which is larger than 0.05, there is no statistically significant difference between the mean of two models, which means GA Greedy feature selection model can achieve the statistically same result as GA feature selection model in terms of output accuracy.

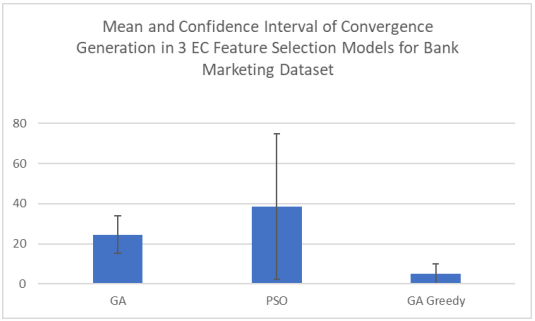


Graph 6. Mean and Confidence Interval of Accuracy in Three Evolutionary Computing Feature Selection Models for Bank Marketing Dataset

On the other hand, according to the mean and confidence interval of convergence generation in three EC feature selection models for Bank Marketing Dataset which is presented in Graph 7. Compare with the confidence intervals of GA model and GA Greedy model, no overlap occurs in these two statistical results, which means there is indeed a statistically significant difference between the running time of two models.

Therefore, enough statistical evidence can be used to prove that GA Greedy model uses a much less convergence generation than GA model and achieve the undifferentiated result with GA model, which means GA Greedy feature selection model is superior than the other in Bank Marketing dataset.

For the PSO model, it converges latest and achieves the worst result that is inferior to the other models.



Graph 7. Mean and Confidence Interval of Convergence Generation in Three Evolutionary Computing Feature Selection Models for Bank Marketing Dataset

VII. CONCLUSIONS

The results broaden the understanding of how EC techniques can be used to feature selection and which king of EC algorithm with which classifier can achieve the best performance in terms of the different datasets. After a series of experiments, comparison, and analysis, our study therefore concludes that, for the Automobile dataset, GA feature selection model with Linear Regression classifier and Root Mean Square Error can obtain the best performance in three test models; for the Adult dataset, GA feature selection with Decision Tree classifier can perform best among three EC feature section models; for the Bank Marketing dataset, GA Greedy feature selection model achieves the best performance in terms of accuracy and convergence generation among three feature selection models. However, since the time limitation, we do not have enough time to experiment the PSO with the different value of the inertia weight and acceleration constants to develop its best performance, and not time to develop with the other parameter configuration for three feature selection. For further research, we will try to explore the potential ability of evolutionary computing and compare with the state-of-the-art model in the market.

REFERENCES

[1] M. Dash and H. Liu, "Feature selection for classification," *Intell. Data Anal*., vol. 1, nos. 1–4, pp. 131–156, 1997.

[2] Y. Liu, F. Tang, and Z. Zeng, "Feature selection based on dependency margin," *IEEE Trans. Cybern*., vol. 45, no. 6, pp. 1209–1221, Jun. 2015.

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

[4] H. Liu, H. Motoda, R. Setiono, and Z. Zhao, "Feature selection: An ever evolving frontier in data mining," in *Proc. JMLR Feature Sel. Data Min*., vol. 10. Hyderabad, India, 2010, pp. 4–13.

[5] H. Liu and L. Yu, "Toward integrating feature selection algorithms for classification and clustering," *IEEE Trans. Knowl. Data Eng*., vol. 17, no. 4, pp. 491–502, Apr. 2005.

[6] Y. Liu et al., "An improved particle swarm optimization for feature selection," *J. Bionic Eng.*, vol. 8, no. 2, pp. 191–200, 2011.

[7] Xue B, Zhang M, Browne WN, Yao X. A survey on evolutionary computation approaches to feature selection. *IEEE Transactions on Evolutionary Computation*. 2016 Aug;20(4):606-26.

[8] I. Guyon and A. Elisseeff, "An introduction to variable and feature selection," *J. Mach. Learn. Res.,* vol. 3, pp. 1157–1182, Mar. 2003.

[9] A. W. Whitney, "A direct method of nonparametric measurement selection," *IEEE Trans. Comput.,* vol. C-20, no. 9, pp. 1100–1103, Sep. 1971.

[10] Q. Mao and I. W.-H. Tsang, "A feature selection method for multivariate performance measures," *IEEE Trans. Pattern Anal. Mach. Intell.,* vol. 35, no. 9, pp. 2051–2063, Sep. 2013.

[11] F. Min, Q. Hu, and W. Zhu, "Feature selection with test cost constraint," *Int. J. Approx. Reason.,* vol. 55, no. 1, pp. 167–179, 2014.

[12] T. Marill and D. M. Green, "On the effectiveness of receptors in recognition systems," *IEEE Trans. Inf. Theory,* vol. 9, no. 1, pp. 11–17, Jan. 1963.

[13] Lichman, M., 2013. UCI machine learning repository.