

A Master Slave Parallel Genetic Algorithm for Feature Selection in High Dimensional Datasets

Shaveta Tatwani, Ela Kumar



Abstract: Feature Selection in High Dimensional Datasets is a combinatorial problem as it selects the optimal subsets from N dimensional data having 2^N possible subsets. Genetic Algorithms are generally a good choice for feature selection in large datasets, though for some high dimensional problems it may take varied amount of time - few seconds, few hours or even few days. Therefore, it is important to use Genetic Algorithms that can give quality results in reasonably acceptable time limit. For this purpose, it is becoming necessary to implement Genetic Algorithms in an efficient manner. In this paper, a Master Slave Parallel Genetic Algorithm is implemented as a Feature Selection procedure to diminish the time intricacies of sequential genetic algorithm. This paper describes the speed gains in parallel Master-Slave Genetic Algorithm and also discusses the theoretical analysis of optimal number of slaves required for an efficient master slave implementation. The experiments are performed on three high-dimensional gene expression data. As Genetic Algorithm is a wrapper technique and takes more time to find the importance of any feature, Information Gain technique is used first as pre-processing task to remove the irrelevant features.

Keywords: Feature Selection, Genetic Algorithm, High Dimensional Data, Machine Learning, Master-Slave.

I. INTRODUCTION

Since the last decade, an overwhelming amount of data is available in many fields, such as bio-medical, World Wide Web, social networking, e-commerce and sensor network etc. This as we know, day-by-day is ever increasing in volume and dimension and this diversification in volume and dimensionality of data necessitates that the vital information be extricated efficiently from the prevalent data and the acquired knowledge be generalized to unrevealed new data of the future. Hence, Feature Selection (FS) techniques are developed to decrease the dimensionality of data by removing the noisy features and thus, increasing the performance of machine learning techniques [1][2]. Feature selection has

following important goals:

- To maximize the accuracy of classifier or any machine learning technique.
- To minimize the time and space complexity of machine learning technique.
- The extraction of useful features which help to interpret the results.

Hence, FS is an important technique which helps develop efficient machine learning models. It is a search technique that maximizes the machine learning performance by choosing an optimal subset of features from high-dimension feature set. As it is a combinatorial problem, researchers have successfully applied evolutionary algorithms, such as Genetic Algorithm [3][4], Swarm Particle Optimization [5][6], Ant Colony Optimization [7][8] as a FS technique. These methods have been effective, but the challenge is that the amount of computation needed is huge. In order to overcome this, a Parallel Genetic Algorithm is used as feature selection technique in this paper.

Genetic Algorithms (GAs) have made their way from university labs into commercial settings. They are already in use to find the solution for difficult and harder problems which cannot be solved otherwise. In case of NP hard and NP complete problems, either the solution is not feasible, or the algorithm takes unacceptably prolonged duration to solve the problem. A genetic algorithm commences its computation with arbitrary solutions to the problems. Each solution is then attributed a fitness which is calculated with the objective function of the optimization problem. Various genetic operators, such as reproduction, crossover and mutation are used to foster evolution of the initial population to new population. This evolution of successive populations by a genetic algorithm continues unless certain benchmark is met, or a suitable solution is detected.

Sometimes the population has explosive combinatorial growth and even simple GA take unacceptably long time. Therefore, two things are important for GAs:

- It should find a good quality solution
- It should find the solution efficiently within some time constraints

To achieve both the above objectives, it becomes necessary to implement the GAs in parallel [9][10]. GAs are inherently parallel and hence they are very suitable candidates for parallelization. One of the easiest and straight-forward ways to parallelize GAs is to use Master-Slave implementation.

Manuscript published on 30 September 2019

* Correspondence Author

Shaveta Tatwani*, Department of Computer Science and Engineering, Indira Gandhi Delhi Technical University for Women, New Delhi-110006, India. E-mail: shaveta.tatwani@gmail.com

Ela Kumar, Department of Computer Science and Engineering, Indira Gandhi Delhi Technical University for Women, New Delhi-110006, India. E-mail: ela_kumar@rediffmail.com

© The Authors. Published by Blue Eyes Intelligence Engineering and Sciences Publication (BEIESP). This is an [open access](https://creativecommons.org/licenses/by-nc-nd/4.0/) article under the CC-BY-NC-ND license <http://creativecommons.org/licenses/by-nc-nd/4.0/>

Master-Slave parallel GAs distributes the initialization, evaluation of objective function or the application of genetic operators among several processors. The master is the process in which the population is cumulated, and the GA is implemented. In contrast, the slaves are the processes which initialize or assess the section of population in parallel. Similar to the serial GA, each individual in the population has competition with the rest of the individuals and also has the possibility of breeding with any of the others. In master-slave implementation, selection and recombination continue to be global and because of this Master-Slave GAs are also referred to as global GAs.

The Master-Slave algorithm is synchronous provided that it pauses and awaits the fitness evaluation for the whole population prior to advancing into the next generation. The technique of probing the space is precisely identical for a synchronous Master-Slave and the simple GA. Master-Slave GA can be implemented in an asynchronous manner provided the algorithm does not hold-up for the slow processor to conclude their allocated function evaluations and thus, there is no precise segregation between generations.

Master-slave GAs are useful and significant for various reasons:

1. The master-slave GAs examines the search space in precisely identical way as serial GA, because of which the existing design standards for simple GAs can be utilized directly.
2. These GAs are popular because of their ease of implementation.
3. In certain instances, Master-Slave GA has been able to increase the performance of simple GAs.

II. EXISTING WORK ON FEATURE SELECTION

The existing basic FS methodologies can be classified into three models: filter model, wrapper model, and Hybrid models [11]. Filtering models [12] are based on statistical analysis of features and they do not consider the classification accuracy. They rank the features depending upon statistical model used and the subsets of features with higher ranks are selected. However, in wrapper models the importance of attribute is calculated based on performance of classifier. Hence, in wrapper models, search strategy and optimization methods are used in feature selection process. As wrapper models depends on classifier performance, they are more accurate but also, more time-consuming than filter models. Hybrid models combine the features of filter model and wrapper model to take the benefit of fast computation of filter models and accuracy of Wrapper models [13].

The process of selecting attributes in wrapper models is an NP-hard problem. For feature selection, a diverse range of search methods have been applied, some of which are complete search, greedy search, heuristic search, and random search, but they suffer from local optima problem. Therefore, efficient global search techniques based on evolutionary computation are used to solve this problem.

Genetic algorithms are one of the progressive techniques widely used in FS process [14]. Each chromosome in GA represents the feature vector which is a binary string, and each bit of this binary string is correspondent to a feature. '1' and

'0' denote that the features they refer to are selected and not selected, respectively. This is aimed at finding the binary vector which would attain the finest performance with the least count of ones. Zhu and Hu used Hybrid method for feature Selection. They used mutual information in filter phase and then genetic algorithm in wrapper phase, i.e., conditional mutual information is first used to filter the features and the optimal feature subset is found by application of the genetic algorithm [15]. A novel genetic operator is proposed to enhance the performance of the genetic algorithm with the objective to minimize the search space [16]. This made the training quicker and helped achieve improved performance. Real biological datasets were used by them to test their algorithm. A multi-objective genetic algorithm based on fuzzy similarity measures is also applied for FS problem [17]. A parallel genetic algorithm is applied to feature selection problem by Chen and Lin [18]. In this paper, a coarse-grain parallel GA is used to optimize the feature subset and SVM classifier parameters. Othman Soufan et al. developed a tool DWFS for choosing attributes based on distributed genetic algorithm wrapper technique. They tested their tool on microarrays using different classifiers [19].

Ant Colony Optimization (ACO) algorithms are also used for dimensionality reduction and they differ mainly in the solution methods. It is a problem which explores the smallest distance covered based on graph. It utilizes the process of the search of actual ants for the smallest distance covered leading to their nourishment [20][21]. Feature subsets are built iteratively using the rule of pheromone that are released and the measurements of heuristic knowledge through the use of simulated ants. The primary distinction between the ACO and GA is that there are no genetic operators in ACO. Ants use the pheromone update rule to keep themselves updated and track the path using their own memory. The background of ACO is more astute and its implementation is much easier in comparison to GAs. The application of ACO to several challenges which are not typically illustrated on graphs is apparent. An exemplification of this is [22] when an ant methodology based on graph was adopted for finding solution to feature selection. In this, there can be representation of candidate solutions in directed graph and a solution to the attribute choice is a representation of each route which is covered by an ant in a sequence. Bello et al. [23] has another instance where a challenge is regarded as a complex system in which nodes are representative of attributes and bidirectional links connect all nodes. Pheromones are related to nodes. Ants conduct a progressive form of selecting in which every one of these beings enlarges its subset steadily with each step by addition of advanced attributes. A multi objective Ant colony optimization is also deployed in microarray data [24]. An unusual hybrid ACO-based classifier model is introduced which integrates ACO and Support Vector Machines (SVMs) to enhance the precision in classifying with a minimal and relevant subset of attributes. [25]. ACO has been utilized for gene selection on microarray data [26][27].

These research works show enough early evidence for ACO being a suitable algorithm for gene subset selection. The Partial Swarm Optimization (PSO) algorithm is a program which works on the principle of worldwide perusal of the population and a replacement resolution. The fundamental theory of PSO was essentially motivated by research of the actions of animals such as the way fish swim together in schools [28] [29].

Essentially, PSO is an elementary, yet powerful procedure which requires a small number of defining factors but usually achieves most appropriate results of the challenge at hand immaculately. Even though the PSO was developed as an alternative to typical progressive methods, such as genetic algorithm (GA), it is not based on crossover and mutation operator. PSO has found successful application in various problems, such as search and optimization method [30]. Wahono and Suryana [31] used Particle Swarm Optimization (PSO) as a method for attribute selection. They employed bagging technique as a classifier which is an ensemble technique where more than one classifier is constructed and voting of the group of classifiers is the basis for the final classification decision. A binary PSO is also applied for feature selection for Arabic text categorization [32], whereas Sivakumar and Chandrasekar [33] made use of an altered continual PSO to select attributes in wrapper-based method with k-nearest neighbor classifier which performed as a fitness function. Lung CT scan images were used to test this algorithm.

III. MASTER SLAVE GENETIC ALGORITHM FOR FEATURE SELECTION

A. Working Principle

In this study, the hybrid model is implemented, in which master-slave parallel genetic algorithm is used in a wrapper phase. As experiments are conducted on three high-dimensional gene expression data, hence information gain feature selection is deployed in pre-processing stage to eliminate inapplicable features. The working principal of our proposed method is illustrated in figure 1.

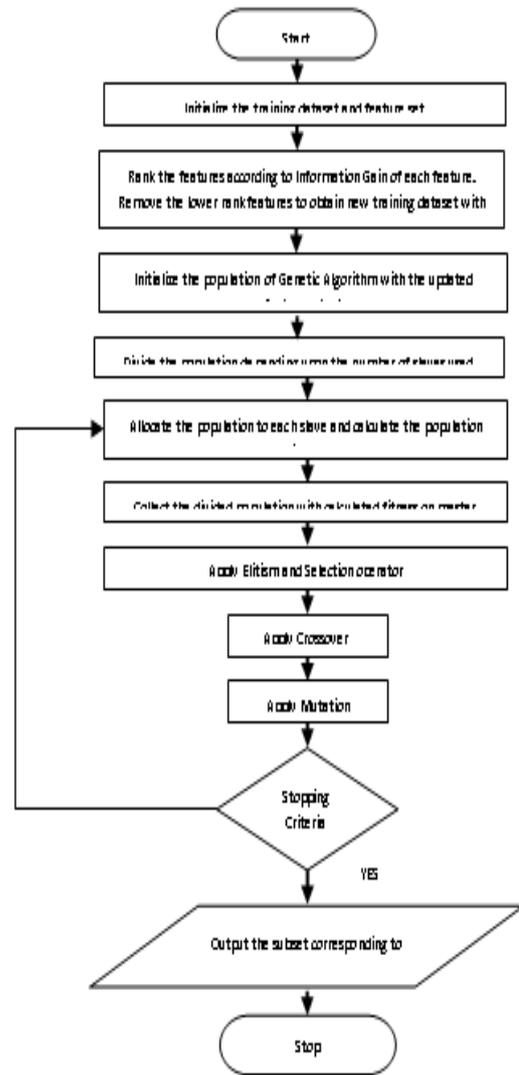


Fig 1: Flowchart of the proposed Master Slave GA FS

B. Parameter Setting

- **Chromosome Representation:** The feature vector of the dataset is representative of each chromosome. It a binary string with 0's and 1's. The bit value 1 denotes that the feature is present, whereas 0's represents its absence.
- **Objective Function:** The objective function is given by: $F(X) = \text{Accuracy of Random Forest Classifier}$
- **The Population Size:** The population of first generation is initialized randomly with 20 chromosomes. Genetic operators are the means by which one generation is converted to the next generation. The population size remains constant over the generations.

- **Genetic Operators:** Selection operator chooses only those individuals (chromosomes) which are highly fit. Fitness is calculated based on objective function. Less fit individuals are discarded and do not take part in the procreation. Mutation and crossover operators are applied on parent chromosome (selected for reproduction to form new chromosomes (offspring's)). Chromosomes are selected by the use of Roulette Wheel Selection. One point crossover with 0.8 crossover rate and simple bit flip mutation is used with mutation rate 0.08 is used for experimentation.

- **Termination Criteria:** In this study, genetic algorithm is executed till 100th generation, and then process will stop.

IV. COMPUTATION TIME OF MASTER SLAVE GA

The analysis investigates the time taken for one generation of the algorithm to be executed. The principal trade-off between computation and communication is identified, and it is used to look for the configuration that lowers the execution time to the minimum possible. In this analysis, the time for selection, crossover and mutation is disregarded as being negligible as compared to the evaluation time and communication time. There is also an assumption in this analysis that equal number of genomes is allocated to all the processors making the communication time constant.

Taking into consideration all the contribution from computation and communication, the elapsed time for one generation of the parallel GA is estimated as [34]:

$$T_p = pT_c + T_f/p$$

Where T_p is the execution time for one generation, T_c is the communication cost, n is the population size, T_f is the time to compute one individual's fitness score and p is the number of processors. p is equal to $S+1$, where S is number of slaves used. It is observed from above equation that with usage of more slaves, there is a decline in the computation time as desired whereas there is an increase in the communication time.

This trade-off presupposes that an optimal number of processors exist which lead to minimization of the execution time. Optimal number of slaves can be found by computing $\partial T_p / \partial p = 0$ and solve for p to obtain

$$p^* = \sqrt{nT_f/T_c}$$

This can be expressed as $= \sqrt{n\gamma}$. Where $\gamma = T_f/T_c$. Hence the optimal number of slaves is $S^* = p^* - 1$.

V. EXPERIMENTS AND RESULTS

A. Datasets

In this study, the experiments are performed on three gene-expression data sets, which is available in R package 'datamicroarray' [35].

Table- I: Datasets

S. No.	Dataset	Sample Size	No. of Classes	No. of Features
1	Colon Cancer	62	2	2000
2	Leukaemia	72	2	7129
3	Prostate Cancer	102	2	12,533

- The Colon Cancer data includes the gene expression data of about 2000 genes and 62 samples taken from colon-cancer patients. Out of these 62 samples, 40 are tumorous marked as 'negative' and 22 are non- tumorous marked as 'positive'
- The Leukaemia data set consist of 72 samples of different patient, out of which 47 patients are having acute lymphoblastic leukaemia (ALL) and 25 patients are having acute myeloid leukaemia (AML). The data matrix contains 7129 gene expressions.
- Prostate Cancer Data Set contains probes for approximately 12,533 genes and 102 samples. Out of these 102 samples: 52 are tumour samples and 50 are non-tumours.

B. Experimental setup

All the involved algorithms are implemented in R 3.5.1. The results are taken on Intel Core i5-8300H Processor which is maximum of 4Ghz clock speed. The machine is equipped with 4GB RAM and 1 TB Hard drive. It is 4 core processor with 8MB cache. To parallelize the genetic algorithm the Open MP compiler is used with R interface.

C. Results

To study the speed gain of our proposed method, it is implemented, and its execution time is compared with varying number of cores. The results are shown below as table II-IV and figure 2-4:

Table-II: Execution time of Master Slave GAFS on Colon Dataset with varying number of slaves

No. of Slaves	Execution Time (in sec)
0	120
1	98
2	96
3	102
4	104

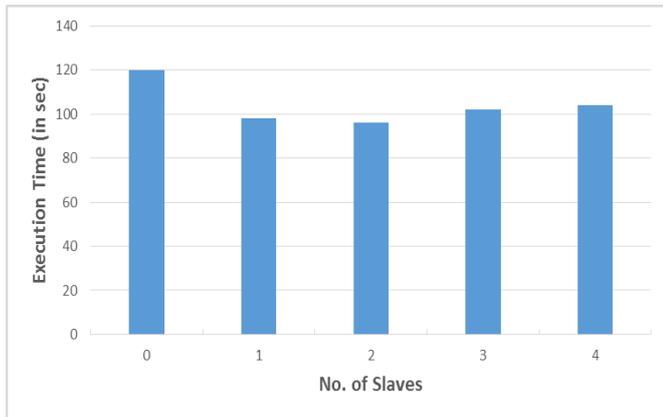


Fig. 2: Execution time of Master Slave GAFS on Colon Dataset with varying number of slaves

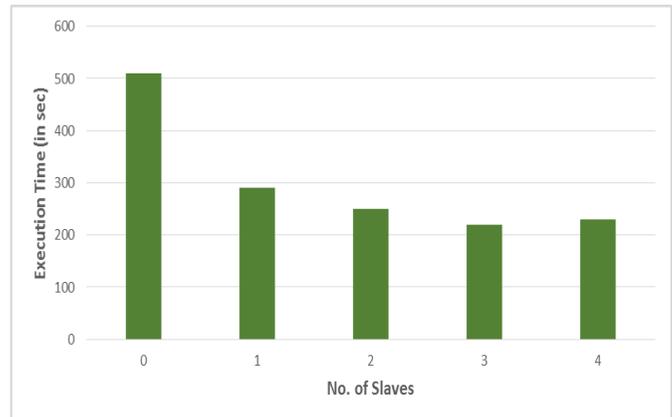


Fig. 4: Execution time of Master Slave GAFS on Prostate Cancer Dataset with varying number of slaves

Table- III: Execution time of Master Slave GAFS on Leukaemia Dataset with varying number of slaves

No. of Slaves	Execution Time (in sec)
0	345
1	210
2	190
3	180
4	200

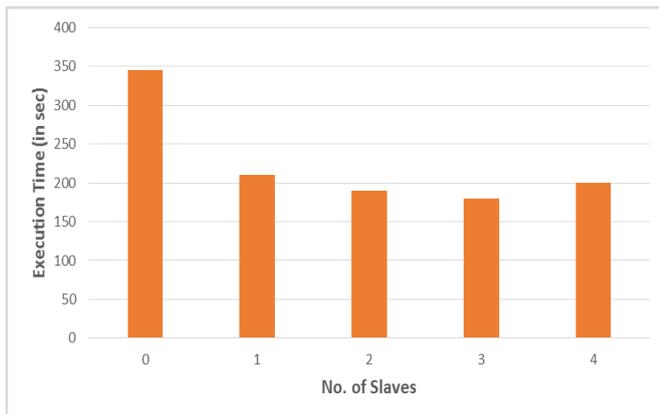


Fig. 3: Execution time of Master Slave GAFS on Leukaemia Dataset with varying number of slaves

Table- IV: Execution time of Master Slave GAFS on Prostate Cancer Dataset with varying number of slaves

No. of Slaves	Execution Time (in sec)
0	510
1	290
2	250
3	220
4	230

VI. CONCLUSION

In this study, we present Master-Slave Genetic Algorithm as a wrapper Feature Selection Technique. It is implemented and tested with ease on three Gene Expression datasets. As these datasets have thousands of genes, hence information gain is utilized which led to the removal of inapplicable attributes. During the experiment the number of slaves was incremented in steps to identify the gain in computing time. Results show that it decreases the execution time significantly with increase in the number of slaves. Therefore, the proposed method reduces the dimensionality and execution time significantly without sacrificing classification performance. This implementation has given enhanced performance of GA in terms of speed gain. However, the speed gain starts deteriorating when the communication overhead starts increasing as more number of slaves are involved.

REFERENCES

1. G David L. Donoho, "High-Dimensional Data Analysis: The Curses and Blessings of Dimensionality," AMS Math Challenges Lecture, 2000.
2. Wei Wang, Jiong Yang, "Mining High-Dimensional Data", The Data Mining and Knowledge Discovery Handbook, Springer Heidelberg, pp. 793-799, 2005.
3. S. Jungjit and A.A. Freitas, "A New Genetic Algorithm for Multi-Label Correlation-Based Feature Selection", Proceedings of the European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning (ESANN2015), 22-24 April 2015.
4. S. Jungjit and A.A. Freitas, "A Lexicographic Multi-Objective Genetic Algorithm for Multi-Label Correlation-Based Feature Selection", Proceedings of the Companion Publication of the 2015 Annual Conference on Genetic and Evolutionary Computation, ACM, pp no 989-996, 2015
5. Bing Xue , Mengjie Zhang, Will 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, Dec. 2013.
6. Lucija Brežočnik, "Feature selection for classification using particle swarm optimization, IEEE EUROCON 2017 -17th International Conference on Smart Technologies, Ohrid, Macedonia, 6-8 July 2017.
7. Victor-Emil Neagoe, Elena-Catalina Neghina, "Feature selection with Ant Colony Optimization and its applications for pattern recognition in space imagery", 2016 International Conference on Communications (COMM), Bucharest, Romania, 9-10 June 2016

8. Kanan H.R., Faez K., Taheri S.M, "Feature Selection Using Ant Colony Optimization (ACO): A New Method and Comparative Study in the Application of Face Recognition System", In: Perner P. (eds) *Advances in Data Mining Theoretical Aspects and Applications*, Lecture Notes in Computer Science, vol 4597, 2007.
9. Saroj, Shaveta, "Research Review and Unresolved Problems in Parallel Genetic Algorithms", 6th International Conference on Cognitive Systems (ICCS-2004), NIIT, New Delhi, December 14-15, 2004.
10. Safa Ibrahim Adi and Mohammed Aldasht, "Parallel Evolutionary Algorithms for Feature Selection in High Dimensional Datasets", *American Journal of Computer Science and Engineering Survey*, AJCSES, Vol.6, No. 01, pp. 013-021, 2018.
11. Z.M. Hira, D.F. Gillies, "A review of feature selection and feature extraction methods applied on microarray data, *Advances in Bioinformatics*", ID 198363, 1-13, 2015.
12. Artur J. Ferreira, Mario T. Figueiredo, "Efficient feature selection filters for high-dimensional data", *Pattern Recognition Letters*, Vol. 33(13), pp. 1794-1804, 2012.
13. V. Bolon-Canedo, N. Sanchez-Marono, A. Alonso-Betanzos, J.M. Benitez, F.Herrera, "A review of microarray datasets and applied feature selection methods", *Information Science* 282, pp. 111–135, 2014.
14. F. O. Catak, T. Bilgem, "Genetic Algorithm based Feature Selection in High Dimensional Text Dataset Classification", *WSEAS TRANSACTIONS on INFORMATION SCIENCE and APPLICATIONS*, Volume 12, pp. 290-296, 2015
15. S. Zhu, B. Hu, "Hybrid Feature Selection Based on Improved Genetic Algorithm", *TELKOMNIKA*, vol. 11, no. 4, pp. 1725-1730, 2013.
16. Sancho Salcedo-Sanz, Fernando Pérez-Cruz, et. Al., "Enhancing Genetic Feature Selection through Restricted Search and Walsh Analysis, *IEEE Transactions on Systems, Man, and Cybernetics-Part*, vol. 34, no. 4, pp. 398-406, 2004.
17. Nahook, H. and Eftekhari, M., "A Feature Selection Method Based on \cap - Fuzzy Similarity Measures Using Multi Objective Genetic Algorithm", *International Journal of Soft Computing and Engineering*, vol. 3, no. 2, pp. 37-41, (2013).
18. Zhi Chen, Tao Lin, Ningjiu Tang, and Xin Xia, "A Parallel Genetic Algorithm Based Feature Selection and Parameter Optimization for Support Vector Machine", *Scientific Programming*, Volume 2016, Article ID 2739621, 2016.
19. Othman Soufan, Dimitrios Klefogiannis, Panos Kalnis, Vladimir B. Bajic, "DWFS: A Wrapper Feature Selection Tool Based on a Parallel Genetic Algorithm", *PLOS ONE*, <https://doi.org/10.1371/journal.pone.0117988>, February 26, 2015.
20. Mehdi Hosseinzadeh Aghdam, Nasser Ghaseem-Aghae, Mohammad Ehsan Basiri, "Text feature selection using ant colony optimization", *Expert Systems with Applications*, Volume 36, Issue 3, Part 2, pp. 6843-6853, April 2009.
21. J. Dhaliya Sweetlin, H. Khanna Nehemiah, A.Kannan, "Feature selection using ant colony optimization with tandem-run recruitment to diagnose bronchitis from CT scan images", *Computer Methods and Programs in Biomedicine*, Vol. 145, pp. 115-125, July 2017.
22. K. Lee, J. Joo, J. Yang, and V. Honavar, "Experimental Comparison of Feature Subset Selection Using GA and ACO Algorithm, *International Conference on Advanced Data Mining and Applications*, LNCS vol. 4093, pp. 465–472, Springer, 2006.
23. R. Bello, A. Nowe, Y. Caballero, Y. Gomez et. al., "A Model Based on Ant Colony System and Rough Set Theory to Feature Selection, *GECCO'05*, Washington, DC, USA, 2005.
24. R. Jensen, Q. Shen, "Finding Rough Set Reducts with Ant Colony Optimization, the 2003 Workshop on Computational Intelligence, UK, pp. 15-22, 2003.
25. Z. Yan and C. Yuan, "Ant colony optimization for feature selection in face recognition," in *Biometric Authentication*, vol. 3072 of *Lecture Notes in Computer Science*, pp. 221–226, Heidelberg, 2004.
26. Sina Tabakhi, Ali Najafi, Reza Ranjbar, Parham Moradi, "Gene selection for microarray data classification using a novel ant colony optimization" *Neurocomputing*, Vol. 168, pp. 1024-1036, November 2015.
27. Tresna Maulana Fahrudin ; Iwan Syarif ; Ali Ridho Barakbah , "Ant colony algorithm for feature selection on microarray datasets", 2016 *International Electronics Symposium (IES)*, Denpasar, Indonesia, 29-30 Sept. 2016.
28. J. Kennedy, R. Eberhart, "Particle Swarm Optimization, proceedings of IEEE International Conference on Neural Networks", pp. 1942–1948, 1995.
29. Y. Shi, R. Eberhart, "A modified particle swarm optimizer, *IEEE International Conference on Evolutionary Computation*, vol. 189, no. 2, pp. 69-73, 1998.
30. N. K. Jain, Uma Nangia, Jyoti Jain, "A Review of Particle Swarm Optimization", *Journal of Institution of Engineers*, Volume 99, Issue 4, pp 407–411, August 2018.
31. R. Wahono, N. Suryana, "Combining PSO based Feature Selection and Bagging Technique for Software Defect Prediction, *International Journal of Software Engineering and Its Applications*, vol.7, no.5, pp. 153-166, 2013.
32. B. Zahran, G. Kanaan, "Text Feature Selection using Particle Swarm Optimization Algorithm, *World Applied Sciences Journal*, vol. 7, pp. 69-74, 2009.
33. S. Sivakumar, C. Chandrasekar, "Modified PSO Based Feature Selection for Classification of Lung CT Images, *International Journal of Computer Science and Information Technologies*, vol. 5, no. 2, pp. 2095-2098, 2014.
34. Erick Cantú-Paz, David E.Goldberg, "Efficient parallel genetic algorithms: theory and practice", *Computer Methods in Applied Mechanics and Engineering*, Vol 186, Issues 2–4, pp. 221-238, June 2000.
35. <https://github.com/ramhiser/datamicroarray/wiki/>

AUTHORS PROFILE



Shaveta Tatwani received her M.Tech in Computer Science and Engineering in 2004 from Guru Jambheshwar University of Science & Technology, Hisar, Haryana, India. From 2004 till 2018, she worked as Assistant Professor at Amity School of Engineering and Technology, New Delhi. She has 14+ years of rich

experience in teaching various computer science subjects to B.Tech graduates and guiding 20+ major projects in the field of Big Data, Grid Computing, and Cloud Computing etc. Also, she worked as research scholar at Technical University of Munich, Germany for one year. There, working in the area of High Performance Computing, she got hands-on experience in Parallel Programming Tools. Currently, Shaveta is pursuing PhD in the field of Machine Learning Techniques from Indira Gandhi Delhi Technical University of Woman, Delhi. Her current research interests are Machine Learning Techniques, Big Data Analysis and High Performance Computing.



Ela Kumar is a HOD (Department of Computer Science and Engineering), Dean (Student affairs) and Chief Proctor at Indira Gandhi Delhi Technical University for Woman, New Delhi, India. She has completed her PhD from University of Delhi in 2003 and done M.Tech (Computer Science and Technology) from IIT Roorkee in 1990. In her 29 years of experience, she has guided about 7 PhD students, 43 M.Tech students and more than 50 B.Tech projects in the field of Web Analytics, Information Retrieval and Artificial Intelligence. She is member of various international and national academic and professional bodies. She is member of Doctoral Research Committee of Delhi Technological University, Birla Institute of Technological and Management, and Amity University. She has authored 5 books in field of Artificial Intelligence, Natural Language Processing, Knowledge Engineering and related areas. She has published more than 100 research papers in national/international journals. She is also member of Editorial Board of few technical journals.