International Journal of Engineering and Advanced Technology (IJEAT)
ISSN: 2249-8958 (Online), Volume-8 Issue-6, August 2019

A Novel Clustering Algorithm to Process Big Data Using Hadoop Framework

D. Jayalatchumy, P. Thambidurai, D. Kadhirvelu

Abstract: ***The real challenge for data miners lies in extracting useful information from huge datasets. Moreover, choosing an efficient algorithm to analyze and process these unstructured data is itself a challenge. Cluster analysis is an unsupervised practice to attain data insight in the era of Big Data. Hyperflated PIC is a Big Data processing solution designed to take advantage over clustering. It is a scalable efficient algorithm to address the shortcomings of existing clustering algorithm and it can process huge datasets quickly. HPIC algorithms have been validated by experimenting them with synthetic and real datasets using different evaluation measure. The quality of clustering results has also been analyzed and proved to be highly efficient and suitable for Big Data processing.***

Keywords: ***Inflation, Hyperflation, Deflation, Power method, Hadoop, MapReduce***

I. INTRODUCTION

Several terabytes of data are being generated every second. Handling those data, processing and extracting useful information from them is itself a big challenge. Existing sequential algorithms have to be refined and tools have to be parallelized to manage this growth. Tools like clustering some distributed frameworks like MapReduce are needed to analyze them as it needs large computing power and distributed storage. Hence, a new algorithm is designed to address these issues and they are also evaluated using different datasets to prove its efficiency and applicability for handling Big Data. Power method is a simple iterative method to find the dominant eigen value and eigen vector for a matrix A where λ is the largest of eigen value and v^0 is the largest of eigen vector. The power method is found using [1] $v^{t+1} = cWv^t$, where v^t is the vector at iteration t, W is the Square matrix, c is the normalizing constant to keep v^t from getting too large or too small. The power method does not compute matrix decomposition [1]. Hence, it is used for larger sparse matrix. The power method converges slowly since it finds only one eigen vector and its difficult when it happens to be more than one dimension[6].

Revised Manuscript Received on August 30, 2019.
* Correspondence Author
Mrs. D. Jayalatchumy*, Computer Science and Engineering from Pondicherry University. Pondicherry. India.
Prof. Dr. P. Thambidurai, Computer science from the Alagappa University, Karaikudi, India.
Mr. D. Kadhirvelu, Associate Professor in Krishnaswamy College of Engineering and Technology, Cuddalore India.

PIC is a family of spectral clustering [6] that uses the power method to compute the largest eigen vector. Instead of finding n eigen vectors PIC finds only one pseudo eigen vector that is a combination of eigen vectors. A pseudo eigen vector is actually not the eigen vector but it is created linearly from it. The computational complexity of PIC is $O(n^2)$ since, it requires matrix vector multiplication.

It is computationally fast and effective. It can be used for solving larger datasets. But, it becomes infeasible for matrices where its dimension is very large. Hence, the convergence can be said as power iteration clustering algorithm converges fast locally and it is slow globally [1]. Moreover PIC finds only one eigen vector which leads to inter collision problem. Hence, new algorithms like IPIC and HPIC are proposed and are implemented on a distributed framework like Hadoop.

II. METHODOLOGY

A. Inflation Technique

It is a technique that uses the Lagrangian constraint [9] to obtain an optimal solution. It incorporates both the objective function and constraint equation. The constraints inflate the vectors to make it converge quickly over a global minimum [3]. This constraint induces an inflation to obtain an optimal solution. Lagrangian multiplier is used to regulate the inflation where the lowest eigen vector grows exponentially fast relative to their neighbors. The nearby eigen values can also be found. The Lagrangian approach is given by [9]

$$L = \sum_i x_i^2 - \sum_{ij} x_i A_{i,j} x_j + \lambda\left(\sum_i x_i^2 - 1\right)$$

where, λ is the Lagrangian multiplier that enforces normalization. The main advantage of this technique is that it converges quickly by reducing the number of iterations and its execution time. IPIC uses both the objective function and the Lagrangian constraints to find its convergence.

B. Deflation Technique

Deflation is a process of modifying a matrix by removing the outcome of the eigen vector such that the acquired matrix is equivalent as that of the original [2]. In the deflation technique the small eigen values of a matrix are replaced with zero to improvise the convergence of iterative linear solvers. It enables an accurate representation of the solution to be computed. The advantages of deflation is that by freezing it at the beginning of the Krylov decomposition the remaining space of decomposition remains orthogonal to it [5].

This gives the opportunity to compute more than one independent eigen vector corresponding to multiple eigen value. The operations are saved in contraction phase. Schurs deflation ensures mutual orthogonality among the pseudo eigen vectors obtained. Moreover it is easy and unique. It has better utilisation of and can be adjusted at every stage of iteration. Schurs deflation for a matrix A_{t-1} with eigen vector v_t which creates a new matrix A_t [5].

$$A_t = A_{t-1} - \frac{A_{t-1}x_t x_t^T A_{t-1}}{x_t^T A_{t-1} x_t}$$

The eigen value of x_t on the matrixA_t is 0, since $A_t x_t = 0$. This indicates the elimination of effect of $v_t on\ A_{t-1}$. Schurs complement deflation preserves position semi definiteness. Let $A_{t-1} \in S_t^p,\ then \forall\ x,$

$$x^T A_t x = x^T A_{t-1} x - \frac{x^T A_{t-1} x_t x_t^T A_{t-1} x}{x_t^T A_{t-1} x_t} \geq 0$$

$$x^T A_{t-1} x x_t^T A_{t-1} x_t - (x^T A_{t-1} x_1)^2 \geq 0$$

by the Cauchy LipSchitz inequality and $x_t^T A_{t-1} x_t \geq 0$ as $A_{t-1} \in S_t^p$. Also, Schurs complement deflation render x_1 , left and right orthogonal to A_t, since A_t is symmetric. Hyperflation or hyperinflation is a condition where the performance of the results becomes so good that it eliminates the drawbacks of the existing system, but retains its advantage. This is obtained by combing the effects of inflation and deflation using a Lagrangian constraint and Schurs deflation technique.

III. HYPERFLATED POWER ITERATION CLUSTERING ALGORITHM

The HPIC algorithm initialises by obtaining the input datasets and converts it into an affinity matrix. Similarity function is found using cosine similarity and the distance between the vectors using Euclidean distance. The obtained similarity matrix thus obtained is normalised using its row sum. Next, the initial vector v_0 is generated using infinity norm. Matrix vector multiplication is performed using the power method new vectors and velocity is generated. It is normalised to avoid early convergence or divergence. Now, Schurs deflation is applied to remove the effect of the generated vector. The largest eigen vector is now subtracted from the given matrix. The matrix obtained is now similar to the original matrix. The power method is reapplied to the matrix obtained to find the largest eigen vector. Deflation proceeds and the process continue until all the desired eigen vectors are found. This method avoids inter collision and also ensures that there are no redundant vectors. Finally, the acceleration is generated from the difference between the vectors.

To determine the convergence of the algorithm, the Lagrangian constraint [9] along with the objective function is considered. Here, the condition to be checked is that the difference between the acceleration should be minimum i.e. the difference between the vectors should be approximately zero. The additional constraints induced here is that the difference between the maximum vector and minimum vector is halved. Now the generated vectors are clustered using K - means algorithm. The Euclidean distance is examined. Based on the Euclidean distance each vector is assigned to the cluster based on its minimum distance. This procedure is repeated for all its vectors. The clustered centroids are updated and the vectors are assigned. This process continues and the final assignment is done. The vectors are the clusters from which the results are extracted. The procedure [10] for HPIC is shown in Algorithm1. The advantages of HPIC are that it is computationally fast when compared to PIC and K-means. It has faster convergence leading to reduced execution time. The number of iterations obtained using HPIC is less. It is accurate than existing algorithms

Algorithm 1: Hyperflated Power Iteration Clustering

Step 1. Input the dataset and the cluster value k.

Step2.Construct a similarity function using fully connected graph based on method like Gaussian similarity function on a given set $s(x_i, x_j) = \exp\left(-\frac{||x_i - x_j||_2^2}{2\sigma^2}\right)$ where σ is the scaling parameter that controls the kernel width

Step 3. Build the affinity matrix A with $a_{ij} = s(x_i, x_j)$ if $i \neq j$ and $a_{ij} = 0$ if $i=j$

Step 4. Obtain the diagonal matrix $D_{ii} = \sum_j A_{ij}$

Step.5. Normalize the obtained matrix by dividing by its row sum. Find the eigen vector with an arbitrary vectorv^0, $v^0 = \frac{R}{||R||}$ where R is the row sum of W.

// Perform matrix vector multiplication

Step.6.Calculate new vector and new velocity $v^t = \gamma W v^{t-1} \& \delta^{t+1} = |v^{t+1} - v^t|$

// Perform Deflation

Step 7. Find the k[th] vector and now remove the effect of $v_k$on W using Schurs deflation $W_l = W_{l-1} - \frac{W_{l-1} v_l v_j^T W_{l-1}}{v_l^T W_{l-1} v_l}$

Step 8. Repeat steps 6 on incrementing the value of t

// Perform inflation and check for convergence using constraints

Step.9. Increment the value of t and check for it convergence, using the constraint $(|\delta^t - \delta^{t-1}| \simeq 0\ \&\&\ \frac{v_{max} - v_0}{2}\ != 0)$

Step 10. Cluster the points $v_t$using K- means algorithm.

IV. ALGORITHMIC DESIGN HPIC IN MAPREDUCE

The algorithm is implemented in the distributed framework Hadoop using single node and multinode setup. The algorithmic design of HPIC is as follows. The input is given to the system which is divided into blocks and given to m mappers depending on the input size. The corresponding <key, value> list is the input to the map.

Now, the similarity matrix is found using the cosine similarity function. This matrix is normalized and the intermediate <Key,Value> pairs from the mapper are generated. The reducer takes this <Key,Value> pairs and calculates the row sum. Now, the mapper gets the row sum and the power method is applied to find the largest of the eigen vector. The schurs complement deflation is applied at this stage to remove the effect of the vector obtained from the original matrix[4].

Once the deflation process is complete the power method is reapplied to get the Eigen vectors and the process continues. Here, the computational complexity increases because of the deflation process. The time taken for finding the first Eigen vector is same as the time for computing the other Eigen vectors. In order to reduce the computational complexity the Lagrangian constraints are induced along with the objective function. The convergence is decided based on the condition where the variation between the vectors approximately be inclined to zero and also the difference between the first and the last vector does not become zero. New vectors and velocities are obtained and finally the iteration stops when the conditions are satisfied. The reducer now collects the obtained vectors from the mapper and emits the vectors. Now the K- means algorithm is applied to the final vectors and the cluster results are written to the output disk. The pseudocode [10] for HPIC in MapReduce is shown in Algorithm 2.

Algorithm 2: Pseudocode for HPIC in MapReduce

Map1

Step 1. m Mappers reads data to split into<key, value> pairs

Step2. Corresponding <key,value> list is input to the map.

Step 3. Calculate the similarity matrix from the similarity function $s(x_i, x_j), s(x_i, x_j) = \exp(-\frac{||x_i - x_j||_2^2}{2\sigma^2})$

Step 4. Find the affinity matrix

Step 5. Normalize the similarity matrix $W=D^{-1}A$

Reduce 1

Step 6. Get the intermediate <Key, Value> pairs from the mapper

Step 7. Calculate the row sum R[i] using its norm.

Step 8. Obtain the initial vector $v^0 = \frac{R}{||R||}$

Map2

Step 9. Calculate new vector and velocity.

repeat

$v^t = \gamma W v^{t-1}$

$\delta^{t+1} = |v^{t+1} - v^t|$

// Remove the effect of v^t from W_l

$W_l = W_{l-1} - \frac{W_{l-1} v_l v_j^T W_{l-1}}{v_l^T W_{l-1} v_l}$ // Schurs deflation

t = *t+1*;

// Inflate the vectors by adding constraints and make the algorithm converge.

$W_t = \frac{f(x^{k+1}) - f(x^k)}{t}$; $\delta t \sim \sqrt{\frac{e_1 - e_0}{e_{max} - e_0}}$

Until

$(|\delta^t - \delta^{t-1}| \simeq 0 \;\&\&\; \frac{v_{max} - v_0}{2} != 0)$

Reduce 2

Step 10. Cluster vectors..

Step 11. Emit pair(matrix, vector)

Step 12. Apply K-means and cluster on the final vector v^t

Step 13. *WRITE* cluster file to output *DIRECTORY*

Output clusters

V. EXPERIMENTAL RESULTS

Experiments were carried out using an Intel core i3 processor, with 2.5 GHZ of processor speed, 4 GB RAM and 320 GB HDD. To evaluate the quality of clustering, various parameters like Execution time, Entropy, F-measure, Accuracy and Normalized Mutual Information (NMI) have been used [7] [11]. This parameter calculates the goodness or the quality of the clusters formed and the similarity between the algorithms including its performance. Moreover, two sets of datasets are considered to evaluate the performance of HPIC. One set of dataset consists of the benchmark datasets from the UCI machine learning repository namely the MNIST, BIRCH, IRIS, HOUSE, and the BREAST dataset. The others datasets is a set of generated datasets of varying sizes 2 MB, 10 MB,50 MB,100 MB,250 MB,500 MB and 800 MB to check its scalability, reliability speedup and efficiency[7][10][11]. The execution time comparison of the HPIC algorithm using MapReduce is shown in Table I.

Table 1. Comparison of Execution Time of HPIC with PIC and K-means

Algorithms / Dataset(MB)	K-means	PIC	HPIC
	Execution Time (sec)		
2	7	5	3
10	35	25	15
50	175	125	75
100	350	250	150
250	875	625	375
500	1750	1250	750
800	2900	2000	1200

From the Table 1 it is inferred HPIC outperforms the other algorithms. Its average computational time is less than the compared algorithms particularly when the dataset size is large. The overall complexity of the matrix vector multiplication is given as $O(n^3)$ with the computational complexity $O(n^2)$. The Accuracy, F-measure , NMI and Entropy[4] were analyzed for IRIS dataset and the results are tabulated in Table II.

Table II. Comparison of Evaluation Metrics of HPIC with PIC and K- means

Dataset (IRIS) (150*4)	Algorithms		
	K- means	PIC	HPIC
Accuracy	0.68	0.71	0.73
F-Measure	0.8	0.85	0.89
NMI	0.74	0.79	0.82
Entropy	0.41	0.39	0.32

The algorithm is made to run in single machine environment and in MapReduce environment with eight data nodes and it is tested for its reliability. In a single machine environment when the machine fails then the entire task fails. There is no reliability here, whereas in the MapReduce environment since there are more than one DataNodes and even if one DataNode shuts down the algorithm can still run using the DataNodes and can produce the same result but with an increase in time. Fig. 1. demonstrates the reliably of HPIC.

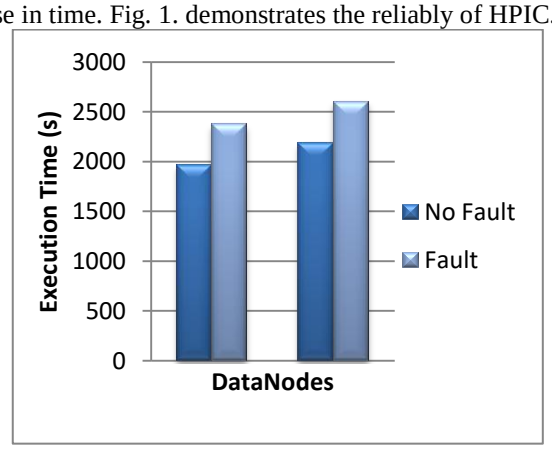

Fig. 1. Reliability of HPIC

The reason is Hadoop is fault tolerant.[9] When a fault occurs other nodes perform the task assigned by the JobTracker. The task will be computed and it can be said that it has good reliability in MapReduce [8] environment. Scalability means the capability of the parallel systems to enhance the speedup in proportionate to the number of processors. Increasing the number of processors decreases its efficiency. The increase in problem size effectively increases the efficiency. A scalable parallel system should increase its efficiency by increasing the number of processors and the problem size simultaneously. The efficiency of parallel algorithm represents the utilization of the cluster during execution. Fig.2. shows the chart representing its scalability.

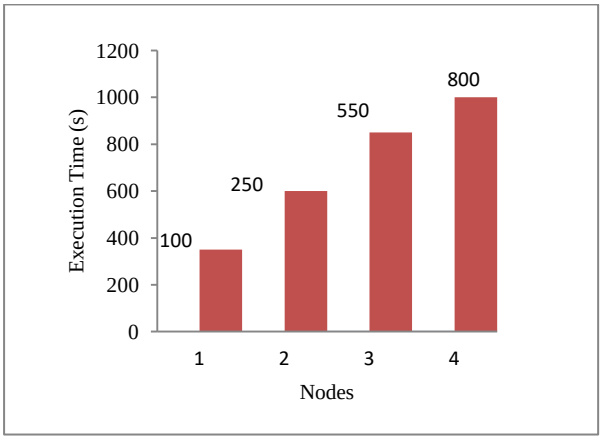

Fig. 2. Scalability of HPIC

From the figure it is understood that as the number of nodes and the size of processing the datasets increases, the running time changes accordingly. To prove the efficiencies of the algorithm for processing larger datasets the algorithm is implemented using Amazon EC2. Here datasets are generated of increasing size of upto 8GB and the execution time was noted based on the execution time the speedup and efficiency has been analyzed and shown in Fig. 3. and Fig. 4.

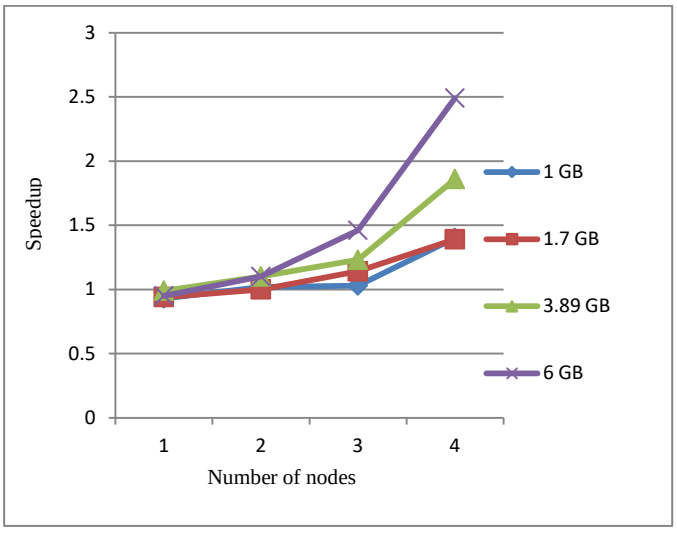

Fig. 3. Speedup of HPIC for larger dataset in Amazon EC2

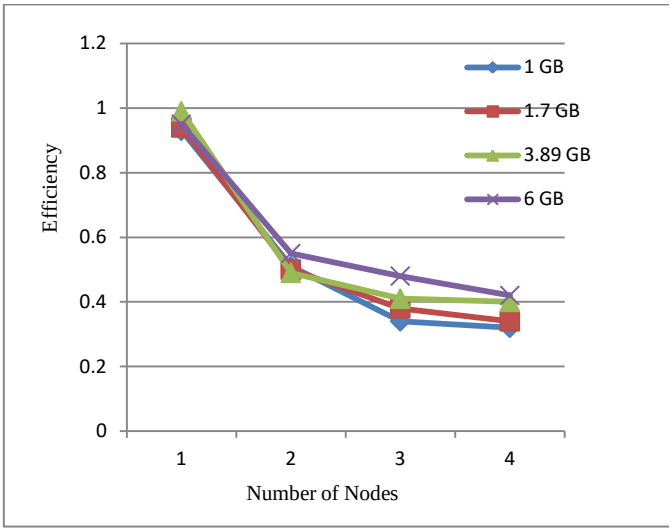

Fig. 4. Efficiency of HPIC for larger dataset in Amazon EC2

From the results it has been noticed that the performance of the proposed algorithms is high. They can work for larger datasets and are highly scalable. As the number of nodes and the data size increases they are able to withstand and run effectively with increase in execution time.

Since they support increasing data size it can handle Big Data. The implementation in MapReduce environment makes it not only scalable but also fault tolerant. It avoids the risk of node failure. The communication cost can be reduced. They are highly accurate and reliable.

VI. CONCLUSION

Extracting and processing useful information is a challenging task. To overcome it a few algorithms like IPIC and HPIC have been developed. They were implemented for various datasets using single machine environment and Hadoop environment. The proposed algorithms have been validated and found to be scalable and fast. From the results obtained it is inferred that there is a reduction in execution time of 40% using IPIC and by 43% using HPIC when compared to PIC thereby increasing its efficiency by 57%. The accuracy of HPIC has been improved by 9% than K-Means and 4% than PIC thereby attaining an accuracy of 73%. Similarly, the F-measure value of HPIC was found to be .89 which tends to 1 implying the performance of the system is high. The amount of information shared between the clusters and the partition were estimated and it was found to be 0.82. This depicts the quality of the clusters being high. The ambiguity found between the clusters was 0.32 which was less compared to the other algorithms.. It was implemented on the cloud server Amazon EC2 with large data size to test its scalability. The algorithm is scalable can be used to cluster Big Data.

REFERENCES

1. Frank Lin frank and William. W,. Power Iteration Clustering. Proceedings of the 27th International Conference on Machine Learning Haifa, (2010).
2. Anh Pham The, Nguyen Duc Thang, La The Vinh, Young-Koo Lee, and Sungyoung Lee,. Deflation based Power Iteration Clustering, Applied Intelligence, Springer. vol.39, pp. 367-385, (2013)
3. Eric J.Heller, Lev Kaplan, Frank Polimann,. Inflationary dynamics for matrix eigenvalue problems. PNAS, vol.105, no.22, pp. 7631-7635(2008)
4. Tom Davis, . Iterated Functions. http:// www. geometer.org/ mathcircles (2012)
5. Lester Mackey, "Deflation Methods for Sparse PCA," In Advances in Neural Information Processing Systems, vol. 21, pp. 1017–1024. (2009)
6. Weizhong Yan, Umang Brahmakshatriya et al., p-PIC: Parallel Power Iteration Clustering for Big Data. Journal of Parallel and Distributed Computing, Models and Algorithms for High Performance Distributed Data Mining, vol. 73, no. 3, pp. 352–359, (2013).
7. Jun Zhang,"Performance and Scalability,"Parallel Computing.Chapter 7. www.cs. Uky .edu/~jzhang /CS621/chapter7.pdf
8. Hardik Pandya, "Setting Up Hadoop Multi-Node Cluster On Amazon Ec2," http://letsdobigdata.wordpress.com/2014/01/13/setting-up-hadoop-multi-node-cluster-on-amazon-ec2-part-1/
9. https://en.wikipedia.org/wiki/Lagrange_multiplier
10. D.Jayalatchumy and P. Thambidurai,. Inflated Power Iteration Clustering Algorithm to Optimize Convergence using Lagrangian Constraint,. Perspectives and Application in Intelligent Systems, vol. 465, pp. 227-237, (2016).
11. Ethan Zhang,Yi Zhang, "F-Measure." Encyclopedia of Database Systems, Springer. pp 1147, 2009. doi: 10.1007/978-0-387-39940-9_483 ISBN. 978-0-387-35544-3.

AUTHORS PROFILE

Mrs. D. Jayalatchumy received her B.Tech degree in 2004, M.Tech degree in 2008 and her Ph.D degree in 2018 in Computer Science and Engineering from Pondicherry University. Pondicherry. India. She joined Perunthalaivar Kamarajar Institute of Engineering and Technology, Karaikal, India in 2009 and currently working as an Assistant Professor in Computer Science and Engineering. Her area of interest includes Data Mining, Big Data and Data structures. She has published many research papers in International conferences and Journals.

Prof. Dr. P. Thambidurai is a Member of IEEE Computer Society. He received his PhD degree in Computer science from the Alagappa University, Karaikudi, India in 1995. From 1999, he served as Professor and Head of the Department of Computer Science & Engineering and Information Technology, Pondicherry Engineering College, Puducherry, India, till August 2006. Now he is the Principal for Perunthalaivar Kamarajar Institute of Engineering and Technology (PKIET) an Government institute at Karaikal, India. His areas of interest include Natural Language Processing, Data Compression and Real-time systems. He has published over 50 research papers in International Journals and Conferences. He is a Fellow of Institution of Engineers (India). He is a Life member of Indian Society for Technical Education and Computer Society of India. He served as Chairman of Computer Society of India, Pondicherry Chapter for two years..

Mr. D. Kadhirvelu has completed his MCA degree from Pondicherry University in 2004 and his M.Tech Degeree from Manonmaniam University in 2012. He served as an Associate Professor in Krishnaswamy College of Engineering and Technology, Cuddalore India from 2005 till 2015. He is currently working at a private concern. She has published many research papers in International conferences and Journals. His areas of interest include Wireless Networks, Data Mining and Algorithms.