AN EFFECTIVE COLLABORATIVE FILTERING FOR UNPREFERED ITEMS BY USING L- INJECTION

[1]V.Swathi Kiran,[2]Dr.M.Sujatha, [3]Dr. R. Jegadeesan 4K.Srikanth Reddy,[5]K.Vishalini Reddy,[6]A.Bhavani

[1,4,5,6]SFinal year Students Of Information Technology,**[2,3]**Associate Professor-CSE

[1,2,3,4,5,6]Jyothishmathi Institute of Technology & Science**,** Karimnagar**,**India.

ABSTRACT: In recent years there has been a dramatic increase in the amount of online content. Recommender systems form a specific type of Information Filtering (IF) technique. To date a number of recommendation algorithms have been proposed, where collaborative filtering is one of the most famous and adopted recommendation technique. Collaborative filtering recommender systems recommend items by identifying other users with similar taste and use their opinions for recommendation. In the last decade, the amount of customers and online information has grown rapidly, yielding the big data analysis problem for recommender systems. Consequently, traditional recommender systems often suffer from scalability and inefficiency problems when processing or analysing such large-scale data. Due to this, the implementation of these algorithms on single node machine is time consuming and fail to meet the computing requirement of large data sets. Distributed processing of big data across multiple clusters of nodes can help to improve the performance in such cases. In this paper, the former collaborative filtering recommendation algorithm is designed to parallel on MapReduce framework and uses Pearson correlation as similarity metric. Apache Hadoop is parallel distributed framework. Hadoop distributed file system(HDFS) allows distributed processing of big data across multiple clusters of nodes.

Key Words: *Recommendation, Collaborative filtering, Pearson correlation, Apache Mahout, Hadoop*

I. INTRODUCTION

In recent years, the volume of data present online has grown exponentially. A major portion of this data is related to internet-based different platforms. The evaluation of such data and/or the extraction of information is difficult due to its huge volume. It is cumbersome for an individual or an organization to obtain the desired results in a timely manner. Hiring the right talent is a challenge faced by all companies. This challenge is amplified by the high volume of applicants if the business is labor intensive, growing and faces high attrition rates. One example of such a business is IT services run out of growth markets. In a typical services organization, professionals with varied technical skills and business domain expertise are hired and assigned to projects to solve customer problems. In the past few years, IT services including consulting, software development, technical support and IT outsourcing has witnessed explosive growth, especially in growth markets like India and China. For in-stance, according to a NASSCOM (National Association of Software and Services Companies of India) study, the total number of IT and IT enabled services professionals in India has grown from 284000 in 1999-2000 to over 1 million in 2004-2005 [1]. More recent estimates suggest that this industry employs more than 2 million professionals in India alone. For organizations in the IT Services business, growth in business is synonymous with growth in the number of employees and recruitment is a key function. Hiring large number of IT professionals in growth markets poses unique challenges. Most countries in growth markets have large populations of qualified technical people who all aspire to be part of the explosive growth in the IT Services industries. Thus, a job posting for a Java programmer can easily attract many tens of thousands of applications in a few weeks. Most IT Services companies are inundated with hundreds of thousands of applicants. For example, Infosys, one of the largest IT Outsourcing companies in India, received more than 1.3 million job applications in 2009. However, only 1% of them were hired. To give the context for work, consider a typical recruitment process. This is illustrated in Figure. The process starts when a business unit decides to hire employees to meet its business objectives. The business unit creates a job pro le that specifies the role, job category, essential skills, location of the opening and a brief job description detailing the nature of work. It might also specify the total work experience that the prospective employee should possess, along with the desired experience level for each skill. The job openings are advertised through multiple channels like on {line job portals, newspaper advertisements, etc. Candidates who are interested to apply for the job opening upload their profile through a designated web-site. The website typically provides an on{line form where the candidate enters details about her application like personal information,[2] education and experience details, skills, etc. We call this Candidate Meta {data. The candidates can also upload their resumes through the website. The objective of allowing the candidate to enter meta {data in an on{ line form is to capture the information in a more structured format to facilitate automated analysis. However, real life experience suggests that most candidates do not specify a lot of information in the on [3]{line forms and hence Candidate Meta{data is often incomplete} Once the applications of prospective candidates are received, they are subjected to careful scrutiny by a set of dedicated screeners. It is shown in the below figure 1.

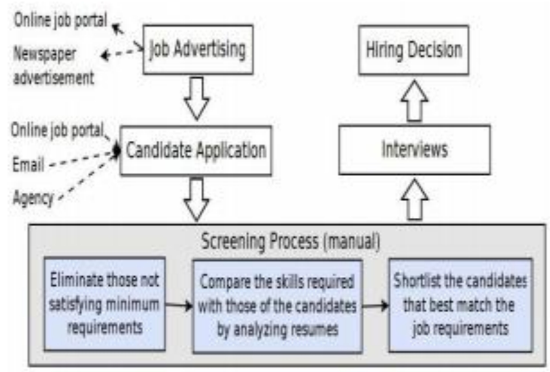

Figure 1: Recruitment process with manual screening

The screeners typically proceed as below [4]: 1. Understand the requirement for the job opening, in terms of the skills that are mandatory and those that are optional but preferable, the experience criteria if any, preference for the location of the candidate etc. Also, note the kind of work that will be performed as part of the job role. 2. Look through each of the applications, and reject those who do not have the minimum years of experience or the skills required for the job. 3. Out of the remaining candidates, and the best match for the job. This requires the recruiter to read the resume in detail and compare it with the job pro le. Since the number of candidates who can be interviewed is limited, the recruiter has to make a relative judgment on the candidates. The top few candidates, who are shortlisted during the screening, undergo further evaluation in the form of interviews, written tests, group discussions etc. The feedback from these evaluation processes is used to make the final hiring decision. Recommender Systems (RS) present an automated and efficient solution to this problem. Recommender systems analyze the user profile/behavior [5] and suggest products/services relative to the user's interests. The recommender system technology plays an important role in various e-commerce applications by helping individuals to find right items in a large option space, which match their interests. The problem of recommending jobs to users is fundamentally different from traditional recommendation system problems such as recommending books, products, or movies to users. While the entire above have a common objective to maximize the engagement rate of the users, one key difference is that a job posting is typically meant to hire one or a few employees only, whereas the same book, product, or movie could be potentially recommended to hundreds of thousands of users for consumption. [6] Ideal job recommendation system would need to recommend the most relevant jobs to users. A job recommender system is expected to provide recommendations in 2 ways: firstly recommending most eligible candidates for the specified job, to the recruiters and secondly, recommending jobs to the aspiring candidates according to their matching profiles. The focus of this paper is the second part only i.e. to recommend jobs to the candidates according to their matching profiles.

II. RELATED WORK

Collaborative filtering (CF) has become one of the most popular recommendation algorithms during recent years [7] due to its high accuracy and efficiency. Collaborative filtering based recommender systems have been applied in many domains, e.g., product recommendation [8], news recommendation [9], [10], video recommendation [11], music recommen1564 Proceedings of the 50th Hawaii International Conference on System Sciences | 2017 URI: http://hdl.handle.net/10125/41342 ISBN: 978-0-9981331-0-2 CC-BY-NC-ND dation [12] etc. Generally, collaborative filtering methods can be classified into two main categories [13]: memory-based collaborative filtering and model-based collaborative filtering. Memory-based CF methods can build some correlations of clients/products based on clients' historical records and then adopt such correlations to predict their future interests [14]. On the contrary, model-based CF methods first train a model based on clients' historical records, and then predict the future interests of clients based on the model [15]. Most of existing model-based collaborative filtering methods rely on both positive ratings and negative ratings of clients on products to train the model, which are not directly applicable to our scenario because there is no negative ratings in B2B markets. Therefore, the base models in this paper are selected from memory-based collaborative filtering methods, which do not suffer from the "missing negative rating" issue. Ensemble methods have been proved to be more accurate than single models, and two of the most popular ensemble methods are boosting and bagging. Boosting methods can integrate the power of a set of "weak" learners to achieve a learner with better performance. Gradient boosting is one of popular boosting method which can construct additive learning models by sequentially fitting a simple parameterized function (base learner) to minimize residuals by some loss function at each iteration. Another popular method is AdaBoosting, which can set higher weights to wrongly predicted examples during iterations to help choose base learners to minimize the overall training error. The above boosting methods can achieve good performance if the base learners can be trained or easily obtained. However, this cannot be easily guaranteed in top-N recommendation application, in which no negative examples are available for model learning. Different from boosting, bagging method can build base learners by changing the set of training examples and then use the average outputs of all the base learners, so that ensemble results can achieve better generalization performance. Bagging can achieve good performance if the base learners are not very stable, e.g., decision-tree or neural network. However, memory-based collaborative filtering methods are similar to kNN methods, which have been proved to be stable. Therefore, bagging cannot substantially improve recommendation accuracy as in other applications. In addition, some of the existing ensemble methods have special requirements about the base learners. For instance, gradient boosting decision tree can only deal with the case that all the base learners are decision trees. On the contrary, the proposed GreedyBoost method can be adopted to integrate any kinds of base learners, e.g., user-based collaborative filtering, item-based collaborative filtering and Naıve Bayes recommendation in this work. This flexibility is important because many real-world problem cannot be easily solved by one kind of methods but a variety of different kinds of methods[16].

III. DISTRIBUTED COLLABORATIVE FILTERING

3.1 Methodology

Assume that there are n users and m elements that are categorized by them. Make P is the user preference data matrix, where Pij is the classification provided by user i to element j, and i ∈ {1,. . . , N} j ∈ {1,. . . , M}. We set Pij = 0 if I did not classify element j, and request that the actual labels be nonzero. P is a scattered matrix with many missing classifications (the density is 0.03 for each Movie data set). We will talk about "customers" and "longer" although the two functions may occur on the same devices. It assumes that each user has a client, Longer counts a total of client data (turned from customer ratings). For simplicity, we will assume that there is a longer n, the least organized state (peer to peer). We assume that the fraction α> 0.5 of the clients are the longest trustworthy, meaning follow the protocols correctly. However, no one is trusted enough to see unencrypted user data. Collaborative filtering methods generally use weighted groups of the nearest votes to extrapolate user preferences. These methods call "Live 3 Ways". Neighborhood methods ignore public relations between user preferences. In fact, there are global linear relationships between user ratings and used in the eigentaste algorithm by Goldberg et al. Allah. [17]. Eigentaste is still an adjacent method, but uses actual user ratings expectations in a low-dimensional area. This space is calculated with a single value analysis of the class matrix. Goldberg showed that this projection before matching the neighbor improves performance, and describes the linear basis vectors as "eigentastes". This indicates that the prediction of the assessment may be done using a global linear approximation only for the class set. In practice, we found that this works well In tests done using the "Every movie" database, labels from the linear form are as good as the best of the current algorithms. In a later section, we compare it to neighborhood ways using surveys from Herlocher and Breese et. Allah. [18]. We build the linear area of k dimensions that approximates the user preference matrix P better in the lower squares. Suppose that A represents a row matrix A ∈ R k × m. Now k m m where m is the number of data elements and the orthonormality state means that AAT = I. P drop on A is P AT A. The remaining modeling error is E = P - P AT A and we want to reduce the sum of the squares of the components of this error matrix, which is e = tr (EET). This is simplified to e = tr (P P T) - tr (P AT AP T) and the minimum error is obtained when maximizing tr (P AT AP T). The optimization problem is then found like this

$$A = \sup_{U:\ UU^T=I} \mathrm{tr}(PU^TUP^T)$$

This optimization uses a convergent gradient scheme, which is discussed in detail in Annex 1. In fact, we show that in addition to A, we can obtain a partial analysis of the singular value (SVD) of P using an encrypted account. Our algorithm is a direct application of the associated gradient method, although there is a non-trivial change of basis at each step [19]. There are more effective ways to calculate SVD, but our goal is to calculate it in a reasonable time using the similarity of the encoded format. The concurrent gradient system allows us to reduce the calculation to a series of vector extensions for user data. In practice, it converges rapidly, with 40-60 repetitions on typical data [20].

IV. EXPERIMENT RESULTS

4.1. Experimental Setup

4.1.1. Dataset Description.

In this section, we evaluate the proposed GreedyBoost method on a real-world dataset, which is collected from a commercial B2B company. The company develops, manufactures and markets computer hardware and software, and also provides IT-related services. Table 1 1568 describes the detailed information of the dataset. Table 2 describes customer features, i.e., country, industry and region, which are adopted in the Naıve Bayes recommendation algorithm. We select customer purchase records from 2011 to 2014 as training data, and use the data in 2015 as testing. During the training of the ensemble model, we randomly select 20% data from training set as validation set, which is used to determine the optimal weight of each base model as it is shown in the below table 1and table 2.

#records	#customers	#products	period
~ 1.3 million	~ 180k	627	2011 - present

TABLE 1: Dataset Description (purchase records)

#countries	#regions	#industries
165	7	24

TABLE 2: Dataset Description (customer features)

4.1.2. Evaluation Metrics.

In top-N recommendation scenario, we can regard the recommendation as binary classification problem, because the label of each client-product pair in the test set is 1 or 0. i.e., purchased or not. And the recommendation scores can be used to predict the probability of clients purchasing products. To this end, we use Receiver Operator Characteristic (ROC) curves, which shows how the number of correctly classified positive examples varies with the number of incorrectly classified negative examples. However, ROC curves can present an overly optimistic view of an algorithm's performance if there is a large skew in the class distribution. Therefore, area under ROC curves (AUC) can be used to evaluate the overall performance of each algorithm. Meanwhile, we also evaluate the proposed GreedyBoost method using precision and recall metrics, which are two commonly used in evaluating top-N recommendation algorithms. Precision-Recall curves can show the trend of recall varying with precision, and the area under PR curves (AUPR) can be used to evaluate the overall performance of each algorithm. For both the two evaluation metrics, higher value means better accuracy. Precision and Recall can be computed as follows

$$\begin{aligned} Precision &= \frac{|I_r \cap I_u|}{|I_u|} \\ Recall &= \frac{|I_r \cap I_u|}{|I_r|} \end{aligned}$$

where Ir is the set of recommended items and Iu is the set of items that are liked by user u. The ROC curve represents the relationship between sensitivity and specificity, in which sensitivity is the same as recall but specificity is slightly different from precision. As pointed out by Davis and Goadrich [13], Precision-Recall curves provide more informative pictures of algorithm performance when dealing with highly skewed datasets. Note that, GreedyBoost can optimize towards AUC and AUPR independently. Therefore, if the targeted dataset is highly skewed, we can see in the below figure 2.

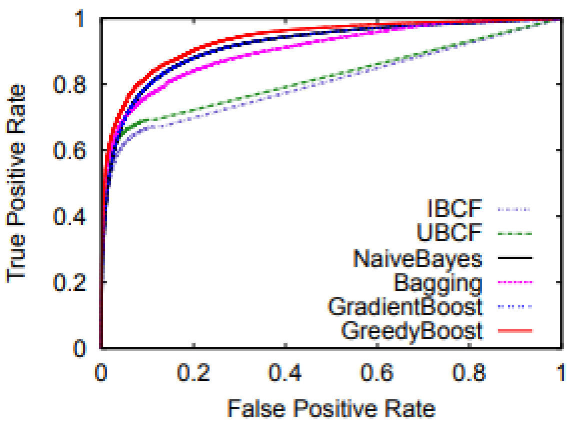

Figure.2: ROC curve comparison

GreedyBoost improves the band model towards AUPR. Otherwise, we can let GreedyBoost improve towards AUC.

4.2. Base Model Selection

The performance of the proposed GreedyBoost method is determined by the selection of base models. In this section, we adopt the three kinds of base models as described in Section 3.1. For UBCF and IBCF, we vary the number of neighbors to form diverse base models. In particular, we adopt 100, 200 and 300 as the numbers of neighbors in UBCF to form three different UBCF-based models and adopt 50, 100, and 150 as the numbers of neighbors in IBCF to form three different IBCF-based models. In total, we have seven base models in GreedyBoost: three UBCF-based models, three IBCF-based models and a NaiveBayes model.

4.3. Accuracy Comparison

In this section, we compared Greedy Boost in two classic ways, for example, Bagging [16] and Gradient Boosting, as well as all the basic models adopted in the group on the above-mentioned assessment criteria AUC and AUPR. Table 3 shows AUC scores for three basic models and three different band methods. We can see from the results that GreedyBoost achieves better performance than all other methods compared. Figure 2 shows the ROC curve for all these models, and the same trend can be seen in Table 3 below.

Model	AUC Score
User-based Collaborative Filtering	0.8172
Item-based Collaborative Filtering	0.8050
Naïve-Bayes	0.9181
Bagging	0.9044
Gradient Boosting	0.9178
GreedyBoost	**0.9361**

TABLE 3: Comparison of Area under ROC Curve (AUC)

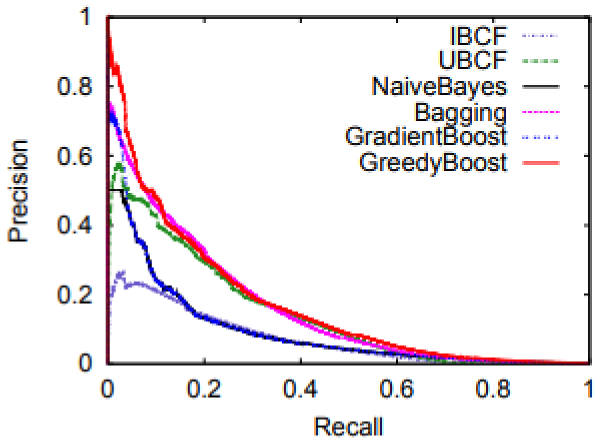

Figure.3: PR curve comparison

Table 4 shows the area below the public relations curve (AUPR) for all three group models and three core models. Note that the exact loop curves show the results averaged on all clients, so that the trends differ slightly from those in the ROC curve. However, the proposed GreedyBoost method still outperforms all other comparison methods. Figure 3 shows the PR curves of all methods compared

Model	Area under PR
User-based Collaborative Filtering	0.1415
Item-based Collaborative Filtering	0.0687
Naïve-Bayes	0.0902
Bagging	0.1557
Gradient Boosting	0.0971
GreedyBoost	**0.1703**

TABLE 4: Comparison of Area under PR Curve

In general, the proposed GreedyBoost method is superior to the other five methods by 2.0% - 16.3% relatively in terms of AUC and 9.4% - 147.9% relatively in terms of AUPR, respectively. The main reasons why the proposed GreedyBoost method achieves better resolution are as follows. First, the proposed method can be directly improved towards the specific scale of assessment. On the contrary, the fill method does not contain such a property because it attempts to improve the model's accuracy by preventing basic forms of flight from using random distribution. The gradient enhancement method can reduce the training error, but the low training error, etc., means the absolute minimum error, does not mean the height of AUC or AUPR because they do not directly correlate. Second, the proposed method has a good generalization performance, so that it does not shine as easily as the gradient increase.

4.4. Efficiency Analysis

To further evaluate the efficiency of Greedy Boost's proposed method, compare GreedyBoost calculation time with increased fill and gradient. In these experiments, the maximum number of repetitions is set to 200 as shown in Table 5 below**.**

Model	Computation Time (seconds)
Bagging	13.1
Gradient Boosting	1,206.9
GreedyBoost	902.3

TABLE 5: Efficiency Comparison

GreedyBoost and Gradient Boosting and convergence threshold is set to 0.0001. For all the three compared methods, we set the number of base models to 7. The numbers reported in Table 5 are the average results over 5 separate runs. Note that computation times for the base model generation process are not included in Table 5 because they can be pre-trained in B2B recommendations. We can see from Table 5 that the computation time for GreedyBoost and Gradient Boosting are much higher than Bagging, which is because training is required to learn model parameters for GreedyBoost and Gradient Boosting. And for Bagging, only a simple average is required to do the ensemble. However, the accuracy of Bagging converges with the number of base models, which can typically be as large as several hundred. Since the training time of base models is much longer than the training time of ensemble, the proposed GreedyBoost method can achieve a much lower overall computation overhead compared to Bagging. GreedyBoost and Gradient Boosting are of similar computation complexity. But compared with Gradient Boosting, GreedyBoost can reduce computation time by approximately 1/4 in the experiments, which is mainly due to faster convergence speed. In conclusion, Greedy Boost can achieve a good level of efficiency compared with popular ensemble methods

V. CONCLUSION

Data can not be used in the form of reviews, opinions, feedback, observations, and complaints as direct, large data for the Recommendation system. This data first needs to be filtered / converted according to requirements. Thus, through this paper an overview of the need to recommend using a distributed frame is done. The proposed system uses user-based collaborative filtering technology to recommend A classy bayes classifier and Hadoop are used as a distribution window. The Apache Mahout framework provides flexibility in using pre-existing algorithms. Because it is built on the Hadoop frame, it eliminates the scalability problem. Because of Hadoop, the system is highly scalable and error-tolerant and can handle a data set of millions of ratios on a single node. The proposed implementation is an independent platform. The experimental results show that the proposed system greatly improves the performance and scalability of the test system through the current approach.

REFERENCES

[1] K. Wei, J. Huang, and S. Fu. A survey of e-commerce recommender systems. In 2007 International Conference on Service Systems and Service Management, pages 1-5, June 2007.
[2] Chenrui Zhang , Xueqi Cheng An Ensemble Method for Job Recommender Systems. RecSys Challenge '16, September 15 2016, Boston, MA, USA 2016 ACM
[3] N. D. Almalis, G. A. Tsihrintzis and N. Karagiannis, "A content based approach for recommending personnel for job positions," IISA 2014, The 5th International Conference on Information, Intelligence, Systems and Applications, Chania, 2014, pp. 45-49.
[4] M. Balabanovic, and Y. Shoham, "Fab: Content-based, Collaborative Recommendation. Communications of the ACM," vol. 40, no. 3, pp. 66- 72, 1997.
[5] M. Ramezani, L. Bergman, R. Thompson, R Burke, and B. Mobasher, "Selecting and Applying RecommendationTechnology," In proceedings of International Workshop on Recommendation and Collaboration in Conjuction with International ACM on Intelligence User Interface, 2008.
[6] BadulSarwar, G. Karypis, J. Konstan, and J. Riedl, "ItemBased Collaborative Filtering Recommendation Algorithms," Proceedings of the 10th International Conference of World Wide Web, pp. 285-295, 2001.
[7] G. Linden, B. Smith, and J. York, "Amazon.com Recommendations: Item-to-Item Collaborative Filtering," IEEE Internet Computing, vol. 7, no. 1, pp. 76–80, 2003
[8] D. Mladenic, "Text-learning and Related Intelligent Agents: A Survey," IEEE Intelligent Systems, vol. 14, no. 4, pp. 44–54, 1999.
[9] RJ. Mooney and L. Roy, "Content-Based Book Recommending Using Learning for Text Categorization," in Proceedings of DL "00: Proceedings of the Fifth ACM Conference on Digital Libraries, New York, NY, ACM pp. 195-204, 2000.
[10] N. D. Almalis, G. A. Tsihrintzis and N. Karagiannis, "A content based approach for recommending personnel for job positions," IISA 2014, The 5th International Conference on Information, Intelligence, Systems and Applications, Chania, 2014, pp. 45-49.
[11] G. Zaltman, How customers think: essential insights into the mind of the market. Harvard Business School Press, 2003.
[12] Q. Zhao, Y. Zhang, D. Friedman, and F. Tan, "E-commerce recommendation with personalized promotion," in Proceedings of the 9th ACM Conference on Recommender Systems, ser. RecSys '15. ACM, 2015, pp. 219–226.
[13] J. Wang and Y. Zhang, "Opportunity model for e-commerce recommendation: right product; right time," in Proceedings of the 36th international ACM SIGIR conference on Research and development in information retrieval, ser. SIGIR '13. ACM, 2013, pp. 303–312.
[14]J.Mangalindan. (2012) Amazon's recommendation secret. [Online]. Available: http://fortune.com /2012 /07 /30 /amazonsrecommendation-secret/
[15] Alibaba. (2016) Financials and metrics. [Online]. Available: http://www.alibabagroup.com/en/ir/financial/
[16] X. Zhang and H. Wang, "Study on recommender systems for businessto-business electronic commerce,," Communications of the IIMA, vol. 5, no. 4, 2005.
[17] G. Adomavicius and A. Tuzhilin, "Toward the next generation of recommender systems: A survey of the state-of-the-art and possible extensions," IEEE Transactions on Knowledge and Data Engeering, vol. 17, no. 6, pp. 734–749, 2005.
[18] G. Linden, B. Smith, and J. York, "Amazon.com recommendations: itemto-item collaborative filtering," IEEE Internet Computing, vol. 7, no. 1, pp. 76–80, 2003. [19] J. A. Konstan, B. N. Miller, D. Maltz, J. L. Herlocker, L. R. Gordon, and J. Riedl, "Grouplens: Applying collaborative filtering to usenet news," Commun. ACM, vol. 40, no. 3, pp. 77–87, 1997.
[20] A. S. Das, M. Datar, A. Garg, and S. Rajaram, "Google news personalization: Scalable online collaborative filtering," in Proceedings of the 16th International Conference on World Wide Web, ser. WWW '07. ACM, 2007, pp. 271–280.
[21] .Jegadeesan,R.,Sankar Ram M.Naveen Kumar JAN 2013 "Less Cost Any Routing With Energy Cost Optimization" International Journal of Advanced Research in Computer Networking,Wireless and Mobile Communications.Volume-No.1: Page no: Issue-No.1 Impact Factor = 1.5

[22]. Jegadeesan,R.,Sankar Ram, R.Janakiraman September-October 2013 "A Recent Approach to Organise Structured Data in Mobile Environment" R.Jegadeesan et al, / (IJCSIT) International Journal of Computer Science and Information Technologies, Vol. 4 (6) ,Page No. 848-852 ISSN: 0975-9646 Impact Factor:2.93

[23] . Jegadeesan,R., Sankar Ram October -2013 "ENROUTING TECHNICS USING DYNAMIC WIRELESS NETWORKS" International Journal of Asia Pacific Journal of Research Ph.D Research Scholar 1, Supervisor2, VOL -3 Page No: Print-ISSN-2320-5504 impact factor 0.433

[24] Jegadeesan,R., Sankar Ram, M.S.Tharani (September-October, 2013) "Enhancing File Security by Integrating Steganography Technique in Linux Kernel" Global journal of Engineering,Design & Technology G.J. E.D.T., Vol. 2(5): Page No:9-14 ISSN: 2319 – 7293

[25] Ramesh,R., Vinoth Kumar,R., and Jegadeesan,R., January 2014 "NTH THIRD PARTY AUDITING FOR DATA INTEGRITY IN CLOUD" Asia Pacific Journal of Research Vol: I Issue XIII, ISSN: 2320-5504, E-ISSN-2347-4793 Vol: I Issue XIII, Page No: Impact Factor:0.433

[26] Vijayalakshmi, Balika J Chelliah and Jegadeesan,R., February-2014 "SUODY-Preserving Privacy in Sharing Data with Multi-Vendor for Dynamic Groups" Global journal of Engineering,Design & Technology. G.J. E.D.T.,Vol.3(1):43-47 (January-February, 2014) ISSN: 2319 –7293

[27] Jegadeesan,R.,SankarRam,T.Karpagam March-2014 "Defending wireless network using Randomized Routing process" International Journal of Emerging Research in management and Technology

[28] Jegadeesan,R.,T.Karpagam, Dr.N.Sankar Ram , "Defending Wireless Network using Randomized Routing Process" International journal of Emerging Research in management and Technology ISSN: 2278-9359 (Volume-3, Issue-3) . March 2014

[29]. Jegadeesan,R., Sankar Ram "Defending Wireless Sensor Network using Randomized Routing "International Journal of Advanced Research in Computer Science and Software Engineering Volume 5, Issue 9, September 2015 ISSN: 2277 128X Page | 934-938

[30] Jegadeesan,R., Sankar Ram,N. "Energy-Efficient Wireless Network Communication with Priority Packet Based QoS Scheduling", Asian Journal of Information Technology(AJIT) 15(8): 1396-1404,2016 ISSN: 1682-3915,Medwell Journal,2016 (Annexure-I updated Journal 2016)

[31] Jegadeesan,R.,Sankar Ram,N. "Energy Consumption Power Aware Data Delivery in Wireless Network", Circuits and Systems, Scientific Research Publisher,2016 (Annexure-I updated Journal 2016)

[32] Jegadeesan,R., Sankar Ram , and J.Abirmi "Implementing Online Driving License Renewal by Integration of Web Orchestration and Web Choreogrphy" International journal of Advanced Research trends in Engineering and Technology (IJARTET) ISSN:2394-3785 (Volume-5, Issue-1, January 2018