Research Article

Euclidean Distance Based Similarity Measurement and Ensuing Ranking Scheme for Document Search from Outsourced Cloud Data

S.N. Manoharan[a], Soumya Ranjan Jena[b]and A. Ilavendhan[c]

[a]Associate Professor, Department of CSE, School of Computing, Vel Tech RangarajanDr.Sagunthala R&D Institute of Science & Technology, Chennai, Tamil Nadu, India (ORCID: 0000-0003-3144-490X)
[b]Assistant Professor, Department of CSE, School of Computing, Vel Tech RangarajanDr.Sagunthala R&D Institute of Science & Technology, Chennai, Tamil Nadu, India (ORCID: 0000-0003-1099-5649)
[c]Assistant Professor, Department of CSE, School of Computing, Vel Tech RangarajanDr.Sagunthala R&D Institute of Science & Technology, Chennai, Tamil Nadu, India(ORCID: 0000-0001-9241-120X)
[a]kritimanoharan@gmail.com, [b] soumyajena1989@gmail.com, [c] ilavendhans@gmail.com

Article History: Received: 10 November 2020; Revised 12 January 2021 Accepted: 27 January 2021; Published online: 5 April 2021

Abstract: In this paper, we propose the Euclidean Distance based Similarity Measurement and Ensuing Ranking (EDSMER) scheme to aid effective document search from outsourced cloud data. It is another attempt to find an alternative to binary based approaches. In this approach, the User or the Data owner needs to filter out the suitable keywords for the document and then the index is prepared. To provide security and privacy, both the data and the index are encrypted and moved to the cloud space. The application of Euclidean Distance based Similarity Measurement and Ensuing Ranking (EDSMER) scheme for document searching takes place after the authorized user requests for the documents through query terms. Initially the authorized user sends a query to Cloud Service Provider to retrieve all the documents which are mapped with the keywords provided by him. The proposed algorithm calculates the distance between the query terms and the index terms. The minimum the distance, the more it is closer towards each other and vice-versa. Our Euclidean Distance based Similarity Measurement and Ensuing Ranking (EDSMER) scheme greatly enhances the system functionality by sending the most relevant documents instead of transmitting all documents back. The experimental validations are performed on RFC and FIRE dataset. Through experimental analysis, we prove that our proposed approach is secure and efficient as well as exhibits better recall and precision rate in the IR system to deal with the document-retrieval process.

Keywords: Document retrieval. Cloud computing • Security • Euclidean Distance • Outsourced data • Information retrieval system

1. Introduction

This paper is intended to develop a document searching algorithm based on ranking through finding of similarity between two set of data by way of Euclidean distance calculation. The ultimate objective of this algorithm is to rank the relevant documents based on the received user query words.

Whenever, diverse sets of data are available, it is necessary to pick the common features between them for the purpose of analysis overlaid on the objectives of the study. Several methods and principles are available to put forth this objective. Here we have applied pure mathematical approach to measure the distance between the two terms, which eventually gives us the degree of closeness between them. Based on this degree of closeness, we can able to extract the best matching words against the reference input words.

2. System Description

Our proposed system includes the following three major stake holders:

1. Data Owner
2. Cloud Service Provider (CSP)
3. Authorized user

Their functions and characteristics are given in Table 1.

Table 1 Stake holders of Proposed System

Stake holders	Functions
Data Owner	(i) Holds legal ownership of data (ii) Extraction of proper keywords (iii) Index preparation (iv) Encryption of data and index (v) Send to CSP
Cloud Service Provider (CSP)	(i) Stores the encrypted data and index (ii) Provides access to authenticated user through the trap door function
Authorized User	(i) Initiates the search operation by sending the query keywords to the CSP (ii) Provide the public key pair for authentication

2.1Theory behind Euclidean Algorithm and Euclidean Distance

Let us take a collection of N documents $\{n_1, n_2, n_3 \dots\dots n_k\}$ being converted and stored in the Cloud space and the user query is Q. The expected retrieved information from EDSMER system is$\{r_1, r_2, r_3 \dots\dots r_k\}$. In the process of getting$r_1 tor_k$, the calculation of relevant score and subsequent distance calculation are involved. Every document is assigned with a relevance score. Regarding the judgment of relevance, the result can be either 0 or 1 if the binary relevance method is used and the result can be 0, 0.5 and 1 for the graded system for relevancy.

In general, the Euclidean distance between any two points X and Y is the length of line which connects them. By the reference of Cartesian coordinates, if two points X and Y are described as, X= (X1, X2, X3,….,Xn) and Y= (Y 1,Y 2,Y 3,….Yn). Then the distance between X and Y can be calculated using Pythagoras formula as,

d(X,Y) = $\sqrt{(y_1 - x_1)^2 + (y_2 - x_2)^2 + \cdots (y_n - x_n)^{\,2}}$

d(X,Y) = $\sqrt{\sum_{i=1}^{n} (y_i - x_i)^2}$

In this manner, the Euclidean distance between the ideal scores $I = \{I_1, I_{2,} \dots., I_k\}$ and obtained scores can be calculated by,

Distance (I,r) =$\sqrt{\sum_{i=1}^{n} (r_i - I_i)^2}$

r and I are the Euclidean vectors because any point on the Euclidean space can be regarded as vectors.

The length of this vector from its origin is called as the Euclidean length or the Euclidean norm.

$$\|I\| = \sqrt{I_1^2 + I_2^2 + \cdots I_n^2}$$

It can also be written as its dot product,

$$\|I\| = \sqrt{I.I}$$

In the particular direction, the relationship between I and r is given by,

r-I = (r_1-I_1, r_2-$I_{2\,,}$$r_3$-$I_{3,}$ …. r_n-I_n)

The displacement between the points are:

$$\|r - I\| = \sqrt{(r - I).\ (\mathrm{r - I})}$$

The general approach to calculate the Euclidean distance is summarized below:

- Pre-process the two sets of data. One data is from cloud database and another one is from user query.
- Calculate the one-dimensional distance between the first keyword from user query and the keywords from Cloud database.
- Based on the number of keywords, the dot operation is performed.
- The distance for every keyword can be calculated.

2.2Description of EDSMER based Information Retrieval

This method also consists of two directional processes. The first one is from the Data owner to CSP and the second one is from the authenticated data user to CSP. The first process involves the preliminary security and data preparation tasks, while the second process involves, finding the smallest distance keywords, which in turn will give the best match to the query input and subsequent efficient ranking also. This operation is described below in detail.

Initially the Data owner extract keywords from numerous contents, then these keywords are bundled together to form the index. After creation of index, the encryption must be done to protect privacy. Both the document and the index are encrypted and stored in the Cloud database. We used asymmetric encryption in this method. With these steps, the first stage of process is completed.

In the second stage, the user query is processed with keywords from CSP in the EDSMER algorithm to find the smallest distance between the words using Euclidean approach. Based on the distance calculations, the ranking is prepared and revealed to the authenticated user.

The complete EDSMER algorithm is explained in two different stages namely,

(a) Set-up stage

(b) Retrieval stage

(a) Set-up Stage

This stage is the primary stage of EDSMER algorithm based IR from encrypted Cloud data. This stage involves the following three processes:

(i) Key generation
(ii) Index preparation
(iii) Encryption

(i) Key Generation

In this stage a pair of keys is generated. Pair refers to combination of public key and private key. Encryption will be done with public key. But the cipher text resulted from this process can be extracted back only if the corresponding pair of private key is applied.

(ii) Index Generation

Step 1: Representation of document

Let us assume that the data owner has the document as a collection of ‘n’ number of files and mathematically, it can be represented as,

D = $\{f_1, f_2, f_3 \dots f_n\}$

Step 2: Extraction of Keywords

Every file contains certain keywords. Let us denote the keywords as $K_{w1}, K_{w2}, K_{w3} \dots K_{wn}$ for the particular document. All the keywords from the incorporated files needs to be extracted and grouped under a single entity as suggested by,

$K_{wj} = \{K_{w1}, K_{w2}, K_{w3} \dots K_{wn}\}$

Step 3: Index creation

While combining the keywords to form an Index, it is necessary to differentiate each and every keyword. One such way to differentiate is to provide a tag, which tells about the weight or some other attributes of the keyword.

(iii) Encryption

The EDSMER algorithm incorporates the asymmetric encryption. In this type of cryptographic techniques, two different keys are used for encryption and decryption respectively.

(b) Retrieval Stage

This second stage begins the process from the input fed by authorized user. This stage consists of three steps namely:

(i) Trap door creation
(ii) Ranking based on EDSMER algorithm

(iii) Decryption

(i) Trap-door creation

The input received from the authorized user is not encrypted, but the data in CSP are encrypted. Then in order to compare the two different kinds of keywords, it is required to generate the one-way mathematical function called Trap-Door function.

(ii) Ranking based on EDSMER algorithm

After the trap door is created, the inputted keywords from authorized user are received by EDSMER algorithm. Assume that there is ‘n’ number of keywords in the user query, then ‘n’ number of times, the Euclidean distance is calculated with respect to all the keywords in the index. After completing all the manipulations, the distance values are pooled up and finally the rank has been prepared.

EDSMER Algorithm

```
Step 1: Authorized user inputs the keywords through query
Step 2: The received inputs forms a Database of keywords
Step 3: Every keyword of this database is compared with the index stored in Cloud storage and the Euclidean minimum distance is calculated
Step 4: All the minimum distance values are pooled up and least ‘n’ number of results are filtered out
Step 5: Ranking based on the minimum distance is completed
Step 6: Ranked documents are displayed to the user
```

(iii) Decryption

After completion of ranking, the results need to be converted into plain text. This can be done by decryption process. At the data owner side, encryption is performed, by which the index and the document are converted into cipher text. It has to be reverted back to plain text before delivery to the user. The type of encryption used in EDSMER method is asymmetric encryption. It consists of two keys. Public key is used for encryption while the private key is used for decryption.

The algorithm for decryption is given below:

Decryption Algorithm

```
Step 1:  Compute $D_1$ and $D_2$
         $D_1 = \frac{(q+1)}{4^{n+1}} \bmod (q-1)$
         $D_2 = \frac{(r+1)}{4^{n+1}} \bmod (r-1)$
Step 2:  Compute a and b,
         $c = y_{n+1}^{D1} \, mod \, q$
         $d = y_{n+1}^{D2} \, mod \, r$
Step 3:  For all the ranked items, j to m, compute y as,
         $y = ckq + dlp$
Step 4:  Continue till completion of list
Step 5:  Perform XOR operation between q and y
```

3. Dataset Description

RFC and FIRE datasets are used for the experimentation. Details about these datasets are given below:

Table 1 Dataset Description of EDSMER Algorithm

Dataset	Queries	Documents	Total number of Unique Terms	Average document length
RFC	150-200	7,41,857	14,83,71,201	350kB
FIRE	128-176	4,56,343	6,27,56,469	275kB

4. Results and Discussions

Our proposed method is evaluated using two real time databases namely, RFC and FIRE. The metrics used for the analysis are given as follows:

1. Time required for generating the trap door function. (with respect to number of documents and number of queries)
2. Computational cost
3. Communication cost
4. Response time of server
5. Recall rate
6. Mean Average Precision (MAP)
7. F-Measure

Two standard mechanisms namely, TRSE and RRSE are chosen for comparing the performance and hence analyze the attributes of our proposed EDSMER system. This comparison restricted to the above mentioned items 1 to 4. For the rest of the items (5 to 7) few other algorithms have been taken.

Those are delivered by the following authors:

1. Toniye (2015)
2. Parapar (2015)
3. Wang (2012)
4. Singh (2016)
5. Yu (2014)

Sections 1.3.1 to 1.3.5 discusses about these performance comparisons.

4.1Time Required for Generating the Trap Door Function

The time required for generation of trap door function is analyzed below.

The trapdoor function generation time is compared against the number of queries. Our proposed method took only 120 to 150 seconds consistently throughout the sample size of 500 to 2500 numbers of documents. Also our method took only half of the time taken by the TRSE and only one-fourth of the time taken by RRSE systems.

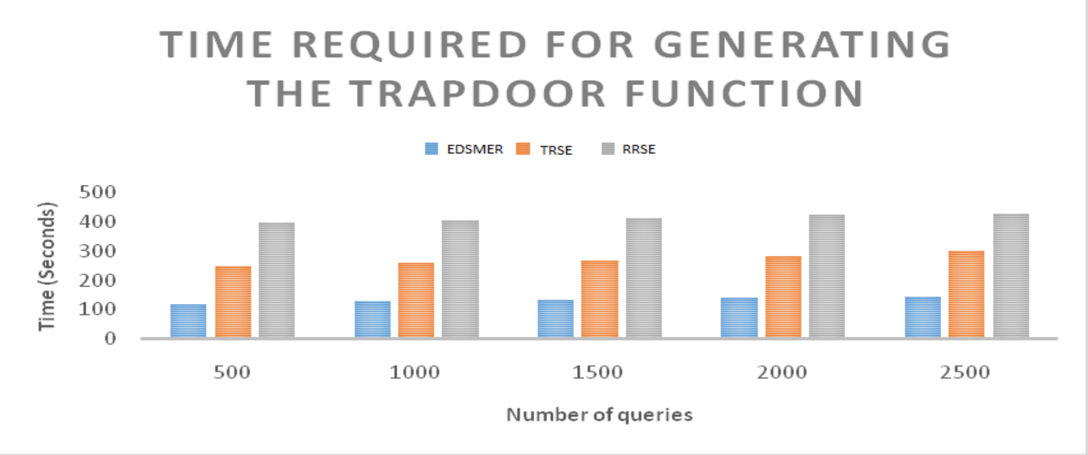

Figure 1. Time required for generating the trapdoor function against number of queries

4.2Computational Cost

The time taken by the system to complete a task, in general, is called as computational cost. In this work, the task is to prepare the ranking.

Figure 2 depicts this performance graphically. In general, the three methods under analysis has taken from 490 seconds to 2400 seconds to complete the process.

For the document size of 1 GB, EDSMER takes 490 seconds, but TRSE and RRSE takes about 500 and 510 seconds respectively. There is not much difference in the performance. For the document size of 2GB and 3GB, all the three methods performed relatively the same. But there is a wide deviation observed when the document size increases. Between 4GB to 6GB, the EDSMER algorithm outperforms the other two methods.

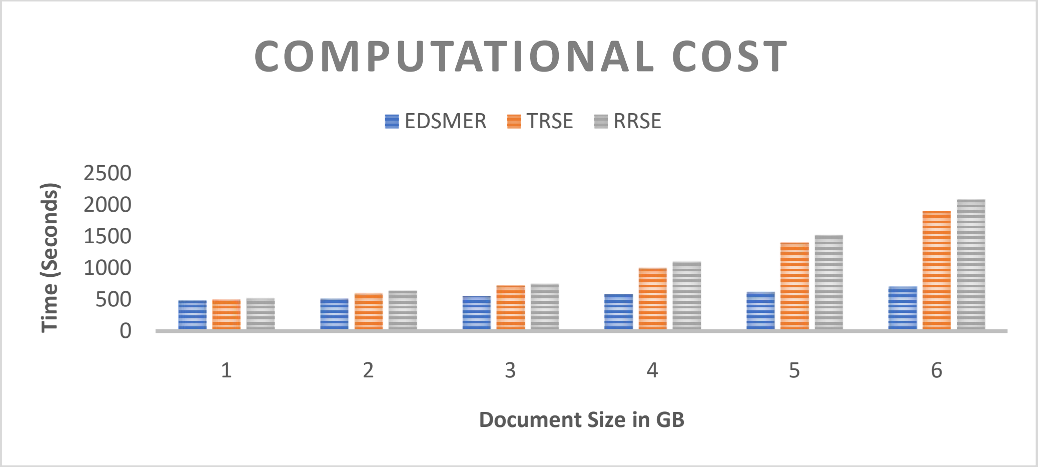

Figure 2. Computational cost comparison

4.3 Response Time of Server

The response times taken by the server along with the ‘m’ values are depicted in Figure 3 and Figure 4. The plot for the response time of server is made between the number of queries and the response time. The number of queries was taken up to 1000 and the response time varied from 1 second to 7 seconds.

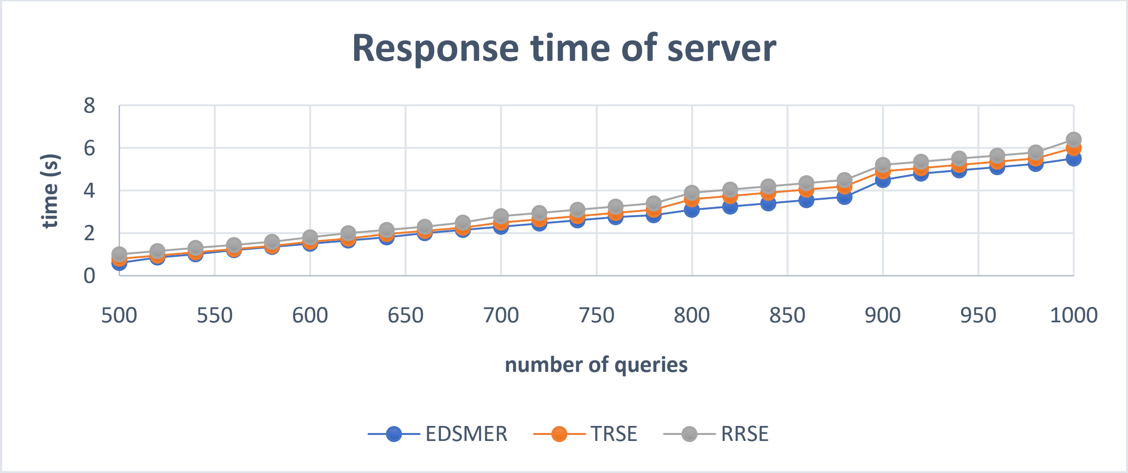

Figure 3. Comparison of the response time of server

Up to the size of 700 documents, there is not much difference observed among the three systems. But, thereafter from 700 to 1000 number of documents EDSMER algorithm outperforms the TRSE and RRSE algorithms. The same trend is replicated in ‘m’ values as well.

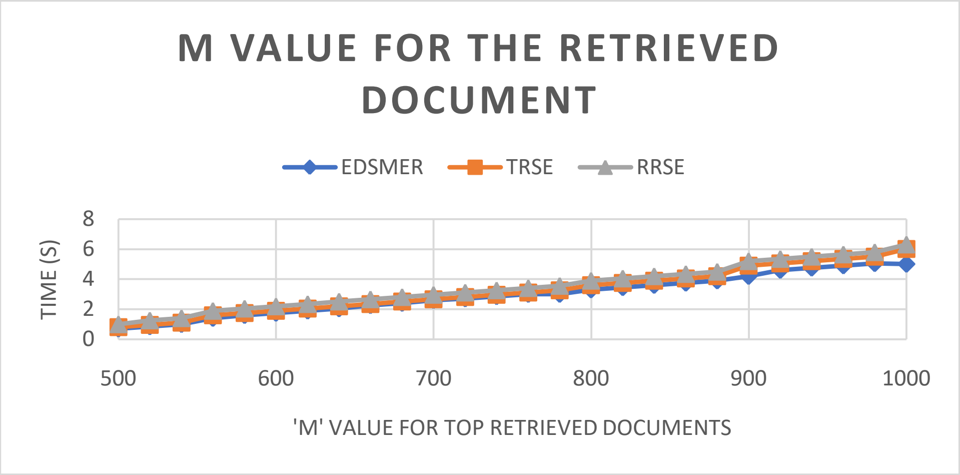

Figure 4. Comparison of ‘m’ values of retrieved document

4.4Communication Cost

The communication cost covers the entire to and from of the communication, in general. In our work, we considered the total time taken by the system to receive, process and complete the entire task. We have taken 200 to 1200 queries for consideration. If we look at the sample size of 200 to 800, our system outperforms the TRSE and RRSE mechanisms. In the remaining time period, it is consistent with the other two mechanisms.

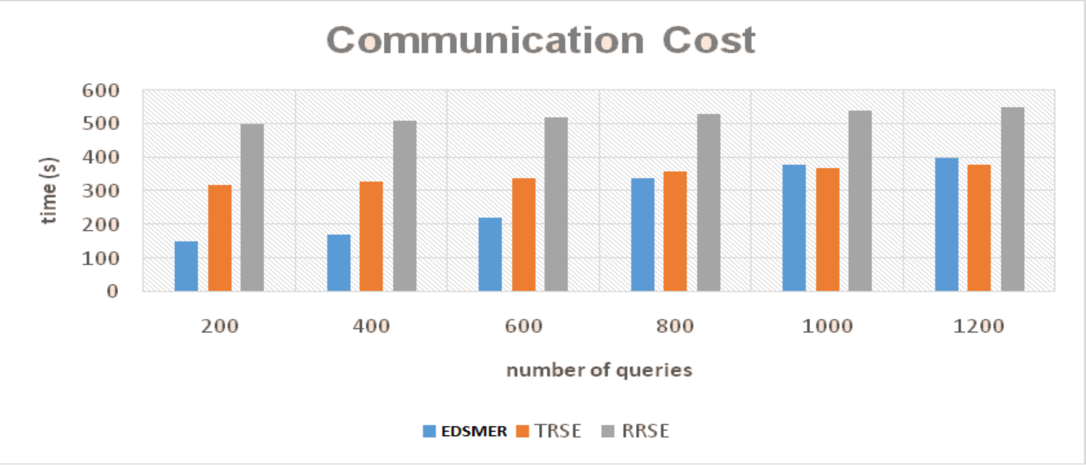

Figure 5. Comparison of Communication Cost

4.5Performance Measures Comparison

Three parameters were taken for measuring the performance of our proposed method in two different datasets.

(i) Recall

It refers to the fraction of relevant documents that are successfully retrieved from the total pool of documents.

(ii) Mean Average Precision (MAP)

It refers to Mean Average Precision. This score gives us the average value of precision of each query. It is calculated by the ratio of sum of precision to the total number of queries.

(iii) F-Measure

F-Measure or F-Score is a harmonic mean of precision and recall; hence, higher the F-Measure, higher the information retrieval.

For analysis of the above mentioned metrics, two dimensional approach was followed in this paper. First the RFC dataset was studied with EDSMER and the other five mechanisms to understand the robustness of our system. Then FIRE dataset was used for the same comparison. Figure 6 exhibits the performance of EDSMER against the other five different mechanisms in RFC dataset. Figure 7 explains the performance in FIRE dataset.

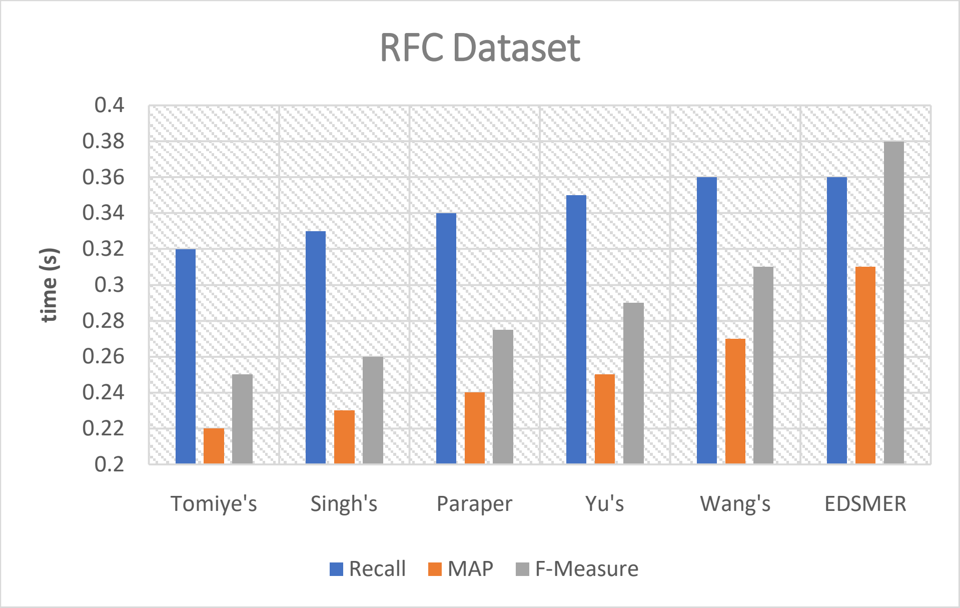

Figure 6. Comparison of Recall, MAP and F-Measure (RFC)

From Figure 6, we can understand that the EDSMER algorithm scores well among all the five different mechanisms being compared.

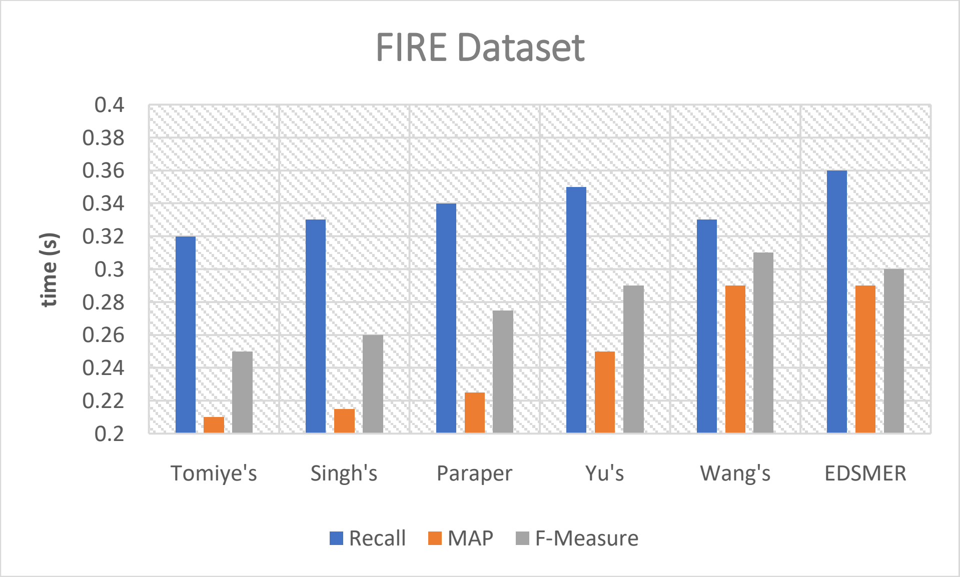

Figure 7. Comparison of Recall, MAP and F-Measure (FIRE)

The same trend is exhibited in FIRE data set as given by Figure 7. But the striking factor is that, if we compare the performance of EDSMER between the two datasets, it performed well in RFC than FIRE.

5. Conclusion& FutureResearch Enhancements

The Euclidean Distance Based Similarity Measurement and Ensuing Ranking (EDSMER) scheme for document search from outsourced Cloud data is narrated from the core principles to the results of experimentation, in this paper. It is yet another attempt to find an alternative to binary based approaches.

The approach of Euclidean distance performed well for the larger document sizes. Hence it can be viewed as an alternative to binary approach. In fact, this scheme performed well than the reference systems, TRSE and RRSE. To conclude, the EDSMER algorithm produced good performance among all the parameters taken for testing. This mechanism outperformed its counterparts in several metrics, but the recall rate is slightly lower than few other systems taken for comparison. Hence this is the critical area to be developed further. Since this approach has a good potential for Information retrieval, this area can be taken up to be developed as a future work.

References

Cecchini, RL, Lorenzetti, CM, Maguitman, AG &Brignole, NB 2008, 'Using genetic algorithms to evolve a population of topical queries', Information Processing & Management, vol. 44, no. 6,

pp. 1863-1878.

Chang, YC &Mitzenmacher, M 2005, 'Privacy preserving keyword searches on remote encrypted data'. In Proceedings of ACNS'05.

Charles Clarke, Nick Craswell& Ian Soboro 2004, 'Overview of the trec 2004 terabyte track. In Proceedings of TREC '04', pp. 500-261.

Chris Buckley & Stephen E Robertson 2008, Relevance feedback track overview: Trec. In TREC.

Chris Buckley, Gerard Salton, James Allan & Amit Singhal 1994, 'Automatic query expansion using smart: Trec3'. In Proceedings of TREC, pp. 69-80.

Cleverdon, CW 1991, 'The significance of the Cranfield tests on index languages', in Proceedings of the 14th annual international ACM SIGIR conference on Research and development in information retrieval, Chicago, Illinois, United States, pp. 3-12.

Cloud Security Alliance 2009, 'Security guidance for critical areas of focus in cloud computing', http://www. cloudsecurityalliance.org. [5] Z. Slocum, "Your google docs: Soon in search results?" http:// news.cnet.com/8301-17939 109-10357137-2.html.

Curtmola, R, Garay, JA, Kamara, S &Ostrovsky, R 2006, 'Searchable Symmetric Encryption: Improved definitions and efficient constructions'. In Proceedings of ACM CCS'06.

Diaz, F & Metzler, D 2006, 'Improving the estimation of relevance models using large external corpora'. In Proceedings of the 29th annual international ACM SIGIR conference on Research and development in information retrieval. New York: ACM, pp. 154-161.

Dilip Kumar Sharma & Sharma, AK 2010, 'A Comparative Analysis of Web Page Ranking Algorithms', in International Journal on Computer Science and Engineering.

Dokmanic, R. Parhizkar, J. Ranieri and M. Vetterli, "Euclidean Distance Matrices: Essential theory, algorithms, and applications," in IEEE Signal Processing Magazine, vol. 32, no. 6, pp. 12-30, Nov. 2015, doi: 10.1109/MSP.2015.2398954.

Fox, EA & Shaw, JA 1994, 'Combination of multiple searches', in Proceedings of the 2nd Text Retrieval Conference, pp. 243-252.

Giambattista Amati, Claudio Carpineto& Giovanni Romano 2004, 'Query difficulty, robustness, and selective application of query expansion', In Proceedings of ECIR '04, pp. 127-137.

Goh, EJ 2003, Secure indexes. Cryptology ePrint archive, report 2003/216. http://eprin t.iacr.org/.

Guo, Z, Zhang, H, Sun, C, Wen, Q & Li, W 2018, 'Secure multi-keyword ranked search over encrypted cloud data for multiple data owners'. Journal of Systems and Software, vol. 137, pp. 380-395.

Soumya Ranjan Jena, and Zulfikhar Ahmad, "Response Time Minimization of Different Load Balancing Algorithms in Cloud Computing Environment", IJCA, Vol 69, No. 17, Pages 22-27, May 2013.

Soumya Ranjan Jena, and Bhushan Dewan, "Improving Quality-of-Service Constraints of Cloud Data Centers", IEEE, 2nd International Conference on Computing for Sustainable Global Development, BVICM, New Delhi, 2015.

Soumya Ranjan Jena, Sudarshan Padhy, and Balendra Kumar Garg, "Performance Evaluation of Load Balancing Algorithms on Cloud Data Centers", IJSER, Vol 5, 3, Pages 1137-1145, 2014.

Soumya Ranjan Jena, V. Vijayaraja, and Aditya Kumar Sahoo, "Performance Evaluation of Energy Efficient Power Models for Digital Cloud", INDJST, Vol 9, 48, Pages 1-7, 2016.

Soumya Ranjan Jena, and L.Shridhara Rao," A Study on Energy Efficient Task Scheduler over Three-Tier Cloud Architecture using Green Cloud", JARDCS, Vol 9, 18, 2017.

Soumya Ranjan Jena, Sukant Kishoro Bisoy and Bhushan Dewan, "Performance Evaluation of Energy Efficient Power Models for Differnent Scheduling Algorithms in Cloud using Cloud Reports", IEEE, GUCON 2019, International Conference on Computing, Power and Communication Technologies, pages. 880-891, Galgotias University, Greater Noida, U.P, India.

Soumya Ranjan Jena, Raju Shanmugam, Rajesh Kumar Dhanaraj, Kavita Saini, "Recent Advances and Future Research Directions in Edge Cloud Framework", IJEAT, Volume 9, Issue 2, Pages. 439-444, December 2019.

Soumya Ranjan Jena, RajuShanmugam, Kavita Saini, Sanjay Kumar, "Cloud Computing Tools: Inside Views and Analysis", Proceedia of Computer Science, Volume 173, Pages. 382-391, 2020.