International Journal of Engineering Applied Sciences and Technology, 2018
Vol. 3, Issue 5, ISSN No. 2455-2143, Pages 46-58
Published Online September 2018 in IJEAST (http://www.ijeast.com)

DATA COMPRESSION ALGORITHMS: ENHANCED J-BIT ENCODING APPROACH

Nirmal Sharma[1], Mahtab Alam[2]
[1]Computer Science, KKU University Abha, Saudi Arabia
[2]Computer Science & Engineering, Department of Noida International University,
Greater Noida, U.P., INDIA,

Abstract- Data compression is very useful approach for compressing the data in every field. The size of databases is increases time to time and it needs to compress for storage and retrieval. The objective of this paper is to shrink the size of statistics through compression techniques using different algorithms. We are proposing a new compression algorithm based on J-bit encoding techniques. This algorithm for designing, optimizing and splitting the input data by dividing into two parts 1 for "true" and 0 for "false". Data compression algorithm compresses the data and reduces the size of database. Data warehouse has large volume of storage capacity that can be used by J-bit encoding for largest data bits. It is hard task to reduce the size of database and minimize the retrieval time. This proposed algorithm of data compression is named as Enhanced J-Bit Encoding (EJBE). The procedure will controls individually bit of statistics of a folder and to optimize the size deprived of trailing slightly statistics subsequently using decoding which is confidential to lossless data compression. It is conducive for enumerated data that provides the output of any format of data.

Keywords- Algorithm**,** data compression, database, Encoding.

I. INTRODUCTION

The database plays a vital role of data mining for organization to provide the better result. Huge volume or composite statistics continuously produces difficult. Consequently, data compression is greatly mandatory through all organization for horizontal, consistent and timely availability of the information. Nearby is a quantity of data compression procedure established through scholars to wrapping the data and image files. Slight effort has finished in this study to diminish the scope of figures and diminish the access period of enticing the registers after figures warehouse [1]. The data are stored in the buffer area of the computer memory. The buffer is a consecutive section of system reminiscence that catch additional than unique examples of the similar information caring. The buffer excess disorder happens once a sequencer efforts to deliver or inscribe external the limits of a block of assigned reminiscence or once a sequencer efforts to place statistics in a reminiscence extent historical a buffer [2][3]. To avoid the buffer overrun problem industries desperately require data compression to store the data within stipulated amount of buffer area.

This paper organized as section 1 covers the general introduction of data compression; section 2 covers the algorithm which are associated with our proposed algorithm related to data compression by earlier researchers, in section 3 we are proposing the multiple algorithm based on J-Bit, Burrow wheel, Zero Run-Length Coding and Huffman algorithm and introduce a new technique by combining all these algorithms, followed by the combination of all earlier proposed algorithm in Section 4. In section 5 result and conclusion are discussed. Finally, in section 6 conclusion part of this research work is described.

Compression method is very popular techniques. This technique is used on very large data contents in logical and physical database. In physical, database the data are stored in bits forms as input stream whereas on logical database, the particular data are stored in the forms of data contents in the output stream and they swap mutual data contents with a little bit of programming code. Logical method is compressing the data in database.

II. ASSOCIATED ALGORITHMS

2.1 J-Bit Encoding: J-bit encoding compress the file by reducing the bit size without losing the data. At the time of decompression the original data is obtain. The main objective of J-bit encoding is to combine the other compression algorithm and mitigate the compression fraction. It enhances J-bit

compression algorithm; the proposed algorithm is discussed in section III [4].

2.2 Burrow-Wheel Forward Transform: This algorithm is comprises in several stages and the input and output of the data are taken arbitrarily, but the size of the data is mentioned in the input stage. The same size of data is obtained after decompression process; it means it works on lossless data compression techniques. It consist three phases during the entire compression process namely, BWT (Burrow Wheel Technique), MTF (Move-To-Front) and EC (Entropy Encoding). In this proposed algorithm we reduced the complexities of the compression algorithm by removing the Entropy Encoding phase.

2.3 Zero Run-Length Coding: It proposed the algorithm to compress the data by arranging the most attainable and least attainable data. There are so many techniques to decrease dimension in data warehouse by eliminating redundancy methods as Zero Run length Coding [5][6].The entire data is divided into two parts in such a way that their numbers of step are used. It is implicit orders of zeros to smaller threads. Today we clarify the practice of the Zero Run-Length Coding phase by phase 233300000/9 (production of the instance since the Move-To-Front Transform) as instance contribution.

2.4 Burrows-Wheeler Backward Transforms: This technique is just reverse of Burrow Wheel Forward Transformation technique. In this technique, the decompression of the complex data is performed into simpler form. The algorithm reverses the process in different order of data compression. The input of the data contents are used in L-column i.e. left hand side and it covers the creative data input. The entire algorithm is discussed in section IV [7].

2.5 Huffman Coding: This is used for bits information and reduced the size of database by bits format. It has used fixed-length encoding like ASCII is suitable since the limits among typescripts are simply resolute and the design allow aimed at individually character is totally immovable (i.e. 'A' is continuously precisely 65) [8].

2.6 Shanon Fano Encoding: Shannon–Fano algorithm was developed by Claude Elwood Shannon and Robert Fano. It is a method to find the probabilities of prefix code which are based on set of symbols. According to this coding, the most probable symbols are getting privilege over slightest feasible ciphers, and formerly decided into two groups whose entire prospects are as adjacent as conceivable to existence identical. Altogether ciphers before have the initial figures of their encryptions allocated; ciphers in the initial usual accept "0" and ciphers in the next usual accept "1". The Algorithm is proposed as under:

III. PROPOSED ALGORITHM

The proposed algorithm works on bits of data that should be reduced in size and enhance the input mechanism handle by J-bit encoding (JBE) based on other algorithm. Proposed algorithm divides the contribution statistics obsessed by two parts wherever the initial part resolve refuge unique nonzero byte and the subsequent part resolve refuge bit assessment light position of nonzero and zero bytes. Together parts formerly can be wrapping clearly through supplementary statistics compression procedure to achieve extreme compression fraction. The compression mechanism can be defined as given step:

Phase1- Input per byte that can be any kind of database.

Phase2- It control input byte by way of nonzero or zero byte.

Phase 3- Output of nonzero byte obsessed by figures I and inscribe bit '1' into provisional byte figures, or individual inscribe bit '0' obsessed by provisional byte figures used for zero contribution byte.

Phase 4- Replication phase 1-3 while waiting for provisional byte statistics occupied through 16 bits of statistics.

Phase 5- If provisional byte statistics occupied through 16 bits formerly inscribe the byte assessment of provisional byte statistics obsessed by figure byte II.

Phase 6- Define provisional byte statistics.

Phase 7- Replication phase 1-6 while waiting for conclusion of folder is touched.

Phase 8- Inscribe joint production statistics

a) Inscribe original contribution measurement.

b) Print figure I.

c) Print figure II.

Phase 9- Figure I and figure II can be wrapping definitely previously joint checked through supplementary compression algorithm,

International Journal of Engineering Applied Sciences and Technology, 2018
Vol. 3, Issue 5, ISSN No. 2455-2143, Pages 46-58
Published Online September 2018 in IJEAST (http://www.ijeast.com)

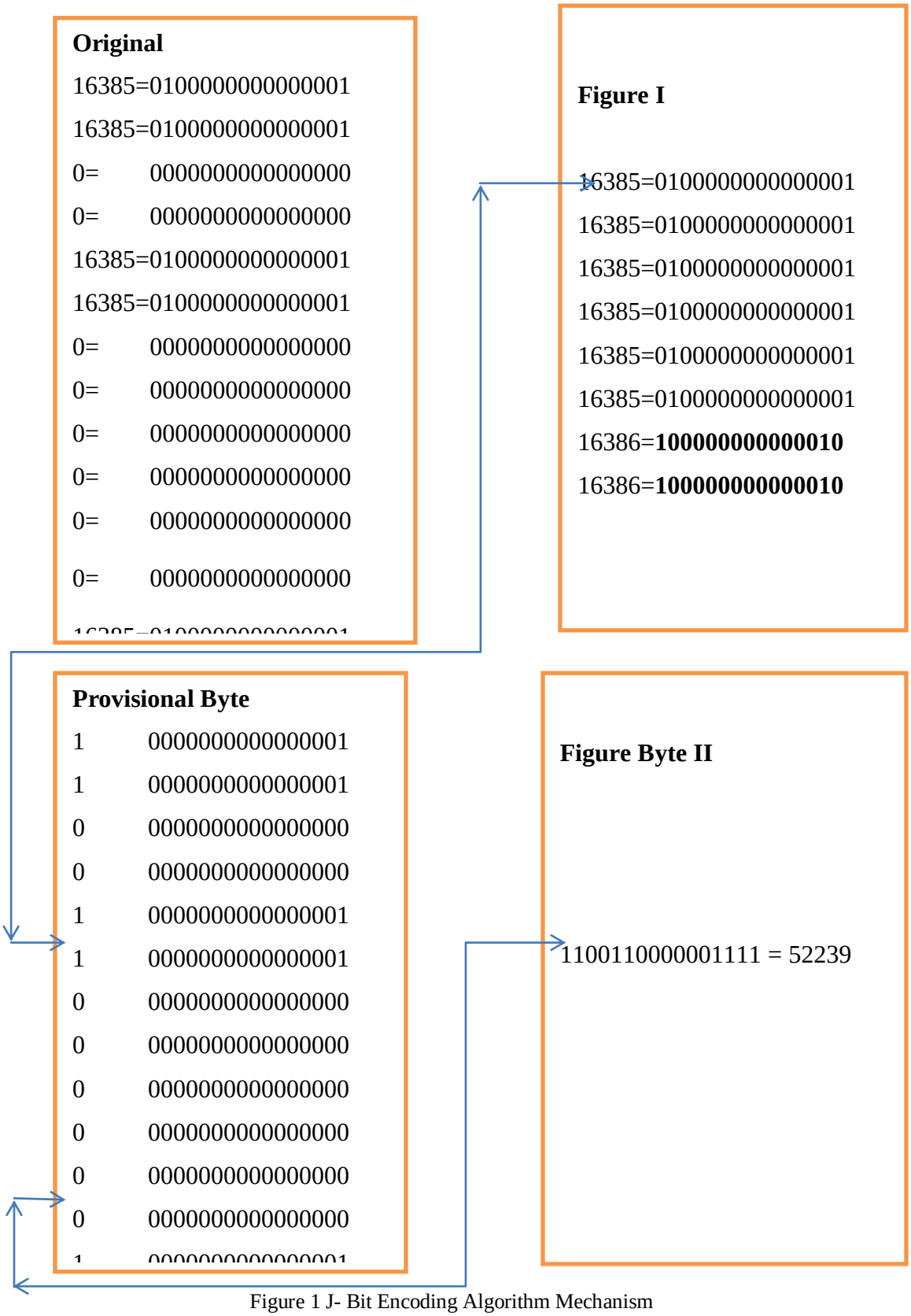

Figure 1 J- Bit Encoding Algorithm Mechanism

International Journal of Engineering Applied Sciences and Technology, 2018
Vol. 3, Issue 5, ISSN No. 2455-2143, Pages 46-58
Published Online September 2018 in IJEAST (http://www.ijeast.com)

Figure 1 displays one-by-one compression mechanism pictorially on behalf of Enhanced J-bit Encoding (EJBE). The unique input data appears obsessed by the first of the outcome resolve be applied by method of statistics on behalf of figure I and figure II size. The data compression mechanism can be defined as following steps:
Step 1- Accept the original input database size.
Step 2- It compacted distinctly, decompress figure I and figure II.

Step 3- Accept figure II per bits.

Step 4- It controls whether input bit is '0' or '1'.

Step 5- It displays the result, uncertainty input bit has'1' before input then display figure I displays the result, uncertainty input bit has '0' before display zero byte to outcome.

Step 6- Replication steps 2-5 until original input data is reached.

IV. ALGORITHM COMBINATION COMPARISON

The data compression algorithm has used five combinations and search out. Following combinations are best for compression relation:

1. ZRL+BWFT
2. BWFT+MTF+EC
3. BWFT+MTF+EC+SFE
4. ZRL+BWBT+EC+SFE+JBE
5. ZRL+BWT+MTF+JBE+**EJBE**

4.1. Burrows-Wheeler Transform Algorithm

It is also data compression techniques. It can be refunded to their unique data with the decompression. It kinds the typescripts of the contribution through the outcome that equal typescripts are nearby composed.

Input- BWT+MTF+EC-output

Process steps:

Step 1- Order the measurement of the contribution n times between them, thus replace respectively row unique ciphers to the correct compared to the earlier row.

Step 2-To arrange the rows lexicographical. Show the outcome of this phase is the L-column (Last column and First column) then index value arranged the grid that contains the unique contribution.

It describes the technique below with MISSISSIPI as sample input.

Step-1

INPUT STREAM

0	M	I	S	S	I
	S	S	I	P	I
1	I	M	I	S	S
	I	S	S	I	P
2	P	I	M	I	S
	S	I	S	S	I
3	I	P	I	M	I
	S	S	I	S	S
4	S	I	P	I	M
	I	S	S	I	S
5	S	S	I	P	I
	M	I	S	S	I
6	I	S	S	I	P
	I	M	I	S	S
7	S	I	S	S	I
	P	I	M	I	S
8	S	S	I	S	S
	I	P	I	M	I
9	I	S	S	I	S
	S	I	P	I	M

Step-2

LEXICOGRAPHY SORTING:-

	F-COL			L-COL	
0	I	M	I	S	I
	S	P	S	I	M
1	I	P	I	S	I
	S	I	S	I	S

International Journal of Engineering Applied Sciences and Technology, 2018
Vol. 3, Issue 5, ISSN No. 2455-2143, Pages 46-58
Published Online September 2018 in IJEAST (http://www.ijeast.com)

2	I	S	I	S	I
	S	I	S	I	S
3	I	S	I	S	I
	S	I	S	I	S
4	M	S	S	M	P
	M	I	M	P	S
5	P	S	S	P	M
	I	M	P	S	P
6	S	I	S	I	S
	I	S	I	S	I
7	S	I	S	I	S
	I	S	I	S	I
8	S	I	M	I	S
	I	S	I	S	I
9	S	I	P	I	S
	P	S	I	M	I

OUTPUT: - MSSSSPIIII/9

The above given strings are arranged in such a way that all the identical characters are intact together, and this string is consider as the input for the next stages [8].

4.2. Move-To-Front Transform Algorithm

It shows the global index value. Additionally we add input data in global list Y.
Process steps:

Step 1- The leading this character follows input data separately since the index value to global list Y
Step 2- Then If we allocation the secure cipher of preceding phase in the universal slope Y happening key location 0 then allocation altogether ciphers unique location toward the correct which situated in the universal slope Y formerly ancient location of the secure cipher.

Step 3- Replication above two steps consecutively on behalf of the additional typescripts of contribution then apply on behalf of altogether replications adapted universal slope since the earlier replication. Outcome of this phase contains of altogether protected directory locations then the directory assessment of the arranged grid since the Burrows-Wheeler Transform which covers the unique contribution. The directory assessment won't procedure in the Move-To-Front Transform [11].

It describes the procedure one by one through MSSSSPIIII/9 (outcome of the instance since the Burrows-Wheeler Transform) as instance contribution. It applies Y = [I, M, P, S] as universal slope then don't a universal slope of altogether typescripts of ASCII-Code, since the instance remains improved to current and informal to comprehend by the slighter universal slope.

STEP 1:

CONTRIBUTION: MSSSSPIIII/9
Y= I M P S

STEP 2: SAVE INDEX POSTION 1
Y= I M **P** S
Y= M I P S

STEP 3: SAVE INDEX POSTION 3
Y= M I P S
Y= S M I P

STEP 4: SAVE INDEX POSTION 3
Y= S M I P
Y= P S M I

STEP 5: SAVE INDEX POSTION 3
Y= P S M I
Y= I P S M

STEP 6: SAVE INDEX POSTION 0
Y= I P S M
Y= I P S M

STEP7: SAVE INDEX POSTION 0
Y= I P S M
Y= I P S M

STEP 8: SAVE INDEX POSTION 0
Y= I P S M
Y= I P S M

STEP 9: SAVE INDEX POSTION 0
Y= I P S M
Y= I P S M

STEP 10: SAVE INDEX POSTION 0
Y= I P S M
Y= I P S M

STEP 11: SAVE INDEX POSTION 0

International Journal of Engineering Applied Sciences and Technology, 2018
Vol. 3, Issue 5, ISSN No. 2455-2143, Pages 46-58
Published Online September 2018 in IJEAST (http://www.ijeast.com)

Y= I P S M
Y= I P S M

Output: 1333000000/9

4.3. Zero Run-Length Coding

It is arrangement of zeros to smaller thread. It describes the method one by one 233300000/9 (outcome of the instance after the Move-To-Front Transform) as instance contribution.
1333000000/9
REPLACING 1 BY 2 AND 3 BY 4 the output string become 2444000000
OUTPUT = 244411 (Six zero will be coded as '11').

Process Steps:

Step 1- All ciphers increases of the contribution which are larger as 0 by 1
Contribution: 1333000000 = 2444000000

Step 2- The arrangements coded of zeros through string arrangements of 0 and 1
It applies on this argument an instance coding counter.
Quantity of zeros coding string

1	0
2	1
3	00
4	01
5	10
6	11

2444000000 = Outcome: 244411
STEP-1

Save order	Index	F- column	L-column
-	0	I	M
-	1	I	S
-	2	I	S
-	3	I	S
-	4	M	S
-	5	P	P
-	6	S	I
-	7	S	I
-	8	S	I
-	9	S	I

STEP-2

Save order	Index	F- column	L-column
-	0	I	M

4.4. Burrows-Wheeler Backward Transforms

Suffix arrangement is similarly a significant difficult in data compression, particularly used for compression arrangements that are constructed arranged the Burrows-Wheeler Transform [12]
This transform process is reverse back algorithm. It identify input data left hand side value to be replace in L-column then the directory assessment is arranged after F-column matrix which covers the creative input data. Following practice is:
Process steps:
Step 1- We are sorted data from the L-column input alphabetical to F-column input data.
Step 2- After that it stores the cipher, and then located now F-column arranged the contribution directory assessment.
Step 3- If we search the character in matrix index value i, whereby, the index value i is used the character in L-column which is similar to the protected cipher of the preceding phase then this cipher takes the similar quantity of equal cipher through lesser directory standards now L-column equally the protected character of preceding phase now the F-column.
Step 4- We can protect the cipher which is situated now right hand side F-column arranged the index value 'i'.
Step 5- Above 3 and 4 steps are replicated and pause, once directory assessment 'i' identical to the contribution data directory. The outcome is contained all protected characters.

International Journal of Engineering Applied Sciences and Technology, 2018
Vol. 3, Issue 5, ISSN No. 2455-2143, Pages 46-58
Published Online September 2018 in IJEAST (http://www.ijeast.com)

-	1	I	S
-	2	I	S
-	3	I	S
-	4	M	S
-	5	P	P
-	6	S	I
-	7	S	I
-	8	S	I
-	9	S	I

STEP-3

Save order	**Index**	**F- column**	**L-column**
-	0	I	M
-	1	I	S
-	2	I	S
-	3	I	S
1	4	M	S
-	5	P	P
-	6	S	I
-	7	S	I
-	8	S	I
-	9	S	I

STEP-4

Save order	**Index**	**F- column**	**L-column**
-	0	I	M
-	1	I	S
-	2	I	S
2	3	I	S
1	4	M	S
-	5	P	P
-	6	S	I
-	7	S	I
-	8	S	I
-	9	S	I

STEP-5

Save order	**Index**	**F- column**	**L-column**
10	0	I	M
8	1	I	S
5	2	I	S
2	3	I	S
1	4	M	S
9	5	P	P
7	6	S	I
6	7	S	I
4	8	S	I
3	9	S	
		I	

International Journal of Engineering Applied Sciences and Technology, 2018
Vol. 3, Issue 5, ISSN No. 2455-2143, Pages 46-58
Published Online September 2018 in IJEAST (http://www.ijeast.com)

Finally, it replicates the process steps, to search the additional cipher of the unique contribution. It recognizes that the rows are in lexicographical direction; consequently the direction of matching cipher after F-column is resolute through the succeeding cipher of the rows. The direction of the matching ciphers after L-column is resolute through the ciphers of F-column and the succeeding ciphers of the rows. Formerly, it can accept that the direction of the matching ciphers now together columns are identical, since the identical ciphers now together columns absolute around the direction of matching ciphers [10].

V. HUFFMAN CODING ALGORITHM

The ASCII set has 256 characters with identical occurrence. In this table has used 90 and unique characters. This algorithm precedes assistance of the difference between occurrences and practices little bit space on behalf of the commonly happening ciphers on the amount of consuming to apply additional universe on behalf of every one of additional infrequent ciphers. For instance as *variable-length* encoding—it is involved 2 or 3 bits then additional ciphers are involved 7, 10, or 12 bits. Reserves character after not consuming to practice a occupied 8 bits used for the greatest corporate ciphers creates for consuming to practice additional 8 bits used for infrequent ciphers then generally result is that particular folder virtually continuously desires little bit storage.

5.1 Encoding ASCII Statement

In this instance, it is successful to procedure through input data is encrypting the certain string

I

"MISSISSIPPI STATE" By the usual encoding ASCII, this 17 ciphers string needs 13 * 8 = 104 total bits. Following ASCII table shows under:

Character	ASCII Value	Binary Value
M		77
	01001101	
I		73
	01001001	
S		83
	01010011	
P		80
	01010000	
T		84
	01010100	
A		65
	01000001	
E		69
	01000101	
Space		32
	00100000	

Instance **"MISSISSIPPI STATE"** string would be encoded in ASCII standard through **77 73 83 83 73 83 83 73 80 80 73 32 83 84 65 84 69**. It is not basically strong by individuals; it would be reproduced as the subsequent stream of bits (each byte is divided into boxes):

010 011 01	010 010 01	010 100 11	010 100 11	010 010 01	010 100 11	010 100 11	010 010 01
010 100 00	010 100 00	010 010 01	001 000 00	010 100 11	010 101 00	010 000 01	010 101 00
010 001 01							

5.2 A data compressed encoding

The primary we can communicate roughly ASCII standards encrypting is that with 8 bits per cipher can be exceptionally extensive data. Even if it permits used for the option of indicating 256 dissimilar ciphers, these individual seven dissimilar ciphers now expression. It is annoying to encrypt, and therefore might distinguish between designs through little bits. It might be design unusual coding counter objective on behalf of expression with 3 bits on behalf of every cipher. Producing as an encrypting is unimportant: it produces a slope of exclusive ciphers, and formerly drives complete then allocate respectively dissimilar encrypted quantity since 0 to N-1. On behalf of instance, as possible 3-bit encoding (i.e.7 possible arrangements);

Character	Digit	Bit Design
M	0	000
I	1	001
S	2	010
P	3	011
T	4	100
A	5	101
E	6	110
Space	7	111

Consuming this table, **"MISSISSIPPI STATE"** is encoded as **0 1 2 2 1 2 2 1 3 3 1 7 2 4 5 4 6**
Denoted as binary numbers

International Journal of Engineering Applied Sciences and Technology, 2018
Vol. 3, Issue 5, ISSN No. 2455-2143, Pages 46-58
Published Online September 2018 in IJEAST (http://www.ijeast.com)

000 001 010 010 001 010 010 001 011 011 001 111 010 100 101 100 110

Three bits per ciphers are applying for encrypting string involves 51 bits in its place of unique 136 bits; squeezing 62.5% string its unique scope.

Yet, the binary coded illustration to decode first would requirement to recognize the distinct mapping used, meanwhile consuming **000** for '**M'** is not regular preparation then now statistic, now it pattern, apiece compacted string practices its individual goal of representing, is not certainly corresponding several additional bits. Different types of caption and sorted supplementary folder would need to stay devoted before involved by encrypted illustration delivers the mapping data. Caption would revenue up particular supplementary storage would change into our compression reserves. On behalf of huge sufficient folder, yet, reserves after frilling depressed per character cost would prospective offset and outflow of supplementary table space.

5.3 Variable-length encrypting

Does it fall the condition that entirely ciphers revenue up the similar quantity of bits? Through consuming little bits to encrypt ciphers alike **'p'**, **'h'**, and storage to ensue regularly and supplementary to encrypt ciphers alike **'y'** and **'o'** to ensue fewer regularly, it could remain competent to wrapping data smooth advance. It would future display to produce counter under, then at the present impartial revenue our expression used for this situation that is denotes execute Huffman encrypting used for the string **"MISSISSIPPI STATE"**:

Character	Weightage	Bit Pattern
M	1	000
I	4	10
S	5	01
P	2	100
T	2	001
A	1	111
E	1	110
Space	1	101

Every cipher consumes exclusive bit design encrypting, then not altogether ciphers practice identical quantity of bits. String **"MISSISSIPPI STATE"** encrypted intense the directly above variable-length encryption counter is:

000	10	01	01	10	01	01	10	100	100
10	10	0	00	11	00	11			

	1	1	1	1	1	0

Total of 42 bits are required in encoded expression, we are cutting an insufficient additional bits from the fixed-length description. What is delicate around a variable-length code is that we not at all may certainly decide the limitations between ciphers in the encrypted stream of bits once decrypting. We enter the character in boxes and show the bit pattern. Then lacking this utility, we can surprise how we will recognize original cipher is encrypted through binary bits **01** or the other three bits **010** before possibly impartial the original bit **0**? It aspect on encrypting now counter overhead, we resolve realize that individual unique these possibilities are probable. It is not ciphers which is encrypts to particular bit **0** then not a bit ciphers, i.e. encrypts to arrangement **010** or **0100** or **01000** designed on behalf of substance. Yet, a cipher encrypts to **01** then i.e. **'S'**. Unique vital structures of counter created through Huffman coding is the prefix stuff: Characters are not encrypting of prefix or new one (**'S'** is encrypted by **01** formerly not at all ciphers encrypting resolve twitch by **01** and ciphers is not encrypted to impartial **0**). By assurance, not any uncertainty in defining wherever the cipher limitations are. It twitches interpretation since the commencement, collecting bits now an order up to it catches a competition. It directs culmination of the cipher and it transfers on decrypting and succeeding cipher.

VI. Shanon Fano Encoding

Shannon and Fano invented coding practice to produce a binary encryption decimal hierarchy [11]. Let symbols X_i are coming from somewhere, such that $1 \leq i \leq 6$, in the BCD (Binary Coding Decimal) format with probabilities P(Xi), is given in Table 1, at a rate Rs = 9.6 kbaud (baud=symbol/second).
State (i) the information rate and (ii) the data rate of the source.

Table 1 Binary Coding Decimal probabilities.

Xi	P(Xi)	BCD word
A	.12	000
E	.10	001
I	.28	010
M	.02	011
P	.14	100
S	.34	101

International Journal of Engineering Applied Sciences and Technology, 2018
Vol. 3, Issue 5, ISSN No. 2455-2143, Pages 46-58
Published Online September 2018 in IJEAST (http://www.ijeast.com)

We have to calculate the information rate and data rate before and after applying the Shanon Fano algorithm [11].

Before Shanon Fano Algorithm (With the help of Entropy)

6.1 Entropy of source:

$H = -\sum_1^6 P(xi).Log_2P(x_i)$

= -.10 * Log_2 0.10 + -.12 * Log_2 0.12 + -.28 * Log_2 0.28 + -.02 * Log_2 0.02 + -.14 * Log_2 0.14 + -.34 * Log_2 0.34

= -.10 * -3.3219 + -.12 * -3.0588 + -.28 * -1.8365 + -.02 * -5.6438 + -.14 * -2.8365 + -.34 * -1.5563

= .33219 + .31265 + .51422 + .11287 + .39711 + .52914

H= 2.19818 bits/symbol

Information Rate R = H.Rs = 2.19818 * 9600 = 21102.53 bits/sec = 21103 bits/sec

Data Rate = 3 * 9600 = 28800 (Before Shanon Fano)

6.2 Shanon Fano Coding

Table 2 Binary Coding Decimal Codes

X	P(X)	Steps	Code
S	.34	0	0
I	.28	1 0	10
P	.14	1 1 0	110
A	.12	1 1 1 0	1110
E	.10	1 1 1 1 0	11110
M	.02	1 1 1 1 1 1	111111

As we have taken only 6 characters (Bits) of possible code length, Therefore the maximum code word length = n – 1 = 5, where n = number of characters.

Average code word length with 5 buffer size is:
d = .34 * 1 + .28 * 2 + .14 * 3 + .12 * 4 + .10 * 5 = .34 + .56 + .42 + .48 + .50 = 2.30 [bits/symbol]
Data Rate: ***d****RS = 2.30 * 9.6 kbaud(Given) Kbaud = Kilobits/second

2.30 * 9600 = 22080

(Bits/Second) (After Shanon Fano)

Compression factor: 3 [bits]/d [bits] = $\frac{3\,[bits]}{2.30[bits]}$ = 1.304

Coding efficiency before Shannon-Fano:

CE = $\frac{Information\ rate}{Data\ rate} = \frac{21103}{28800}$ = 73.27%

Coding efficiency after Shannon-Fano:

CE = = $\frac{Information\ rate}{Data\ rate} = \frac{21103}{22080}$ = 95.57% = 96%

Hence Shanon Fano Algorithm brought the coding efficiency closer to 100%.

Table 3 Access Time Performance of Database

Database (n)	Database Size	Evolution Time(Milliseconds) (X)
NIU_CSE	170050	2500
NIU_ECE	195000	2580
NIU_CE	205560	2600
NIU_ME	218310	2759
NIU_BIO	220830	2801
NIU_EE	230940	2950
NIU_EEE	240050	2995
NIU_SBM	245000	3001
NIU_SLA	250500	3035
NIU_SOS	260500	3050
NIU_LAW	280500	3080
NIU_NURSING	290500	3100
NIU_ARCH	291000	3150
NIU_EDU	292000	3160
NIU_FINEART	293000	3190
NIU_MEDIA	310500	3200
NIU_BDS	320000	3250
NIU_MBBS	330000	3290
NIU_BSW	340500	3320
NIU_MSW	345000	3380

International Journal of Engineering Applied Sciences and Technology, 2018
Vol. 3, Issue 5, ISSN No. 2455-2143, Pages 46-58
Published Online September 2018 in IJEAST (http://www.ijeast.com)

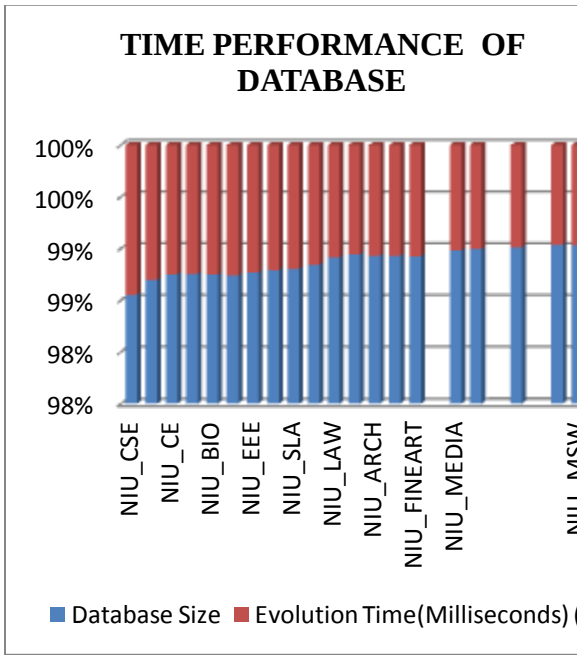

Figure 5 Access Time Performance of Database

A database supplies huge volumes of departmental statistics and all particulars of a division. The above statistics can be applied to resolve a number of queries, and it can be applied to increase the functionality of an association in a short time. Not only users are able to access the database contents, but this data is also providing integrity. But also users are able to manage this information with current data. All the above types of problem can be resolved by proposed algorithm. The figure-5 depicts that as the size of database increases the performance related to access time is reducing consequently.

VII. RESULT OF COMBINATION

Table 4 Combination of 24 Bit Ratio algorithms

S.no.	Types of combination	Result	Record
1	ZRL+BWFT	85.65	100
2	BWFT+MTF+EC	85.73	100
3	BWFT+MTF+EC+SFE	66.76	100
4	ZRL+BWBT+EC+SFE+JBE	56.73	100
5	ZRL+BWT+MTF+JBE+**EJBE**	55.42	100

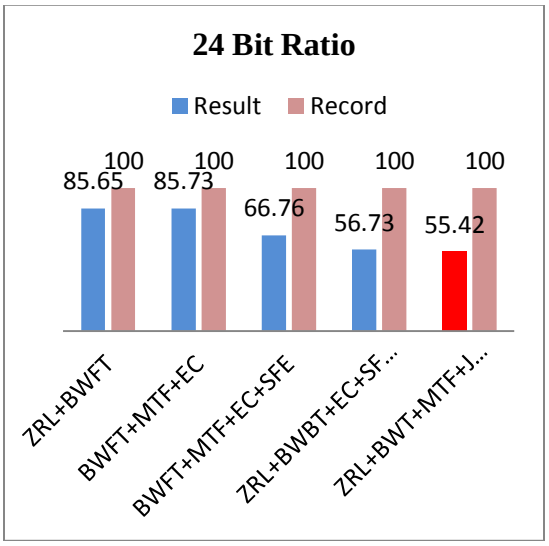

Figure 6 Combination of 24 Bit Ratio algorithms

This Figure indicates that 24-bits bitmap pictures compacted by decent compression relation through procedures that collective through Enhanced J-bit Encoding.

Table 5 Combination of 64 Bit Ratio algorithms

S.no.	Types of combination	Result	Record
1	ZRL+BWFT	89.4	100
2	BWFT+MTF+EC	89.52	100
3	BWFT+MTF+EC+SFE	80.51	100
4	ZRL+BWBT+EC+SFE+JBE	78.61	100
5	ZRL+BWT+MTF+JBE+**EJBE**	62.52	100

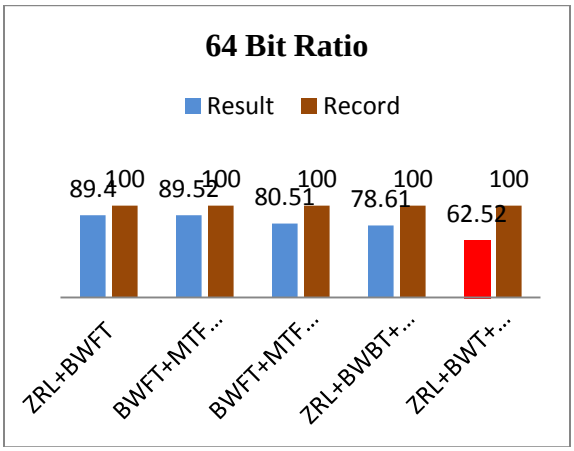

Figure 7 Combination of 64 Bit Ratio algorithms

This Figure indicates that 64-bits bitmap pictures compacted by improved compression relation through procedures that collective through Enhanced J-bit Encoding. A 64 bits bitmap pictures are most composite figures than 24 bits it is enhanced additional pigment. A 64 bit bitmap image are achieved greatest compression relation, smooth supposed that resolve reduction excellence of innovative picture.

Table 6 Comparison of 64 Bit Ratio algorithms

S.no.	Types of combination	Result	Record
1	ZRL+BWFT	64.81	100
2	BWFT+MTF+EC	65.82	100
3	BWFT+MTF+EC+SFE	54.51	100
4	ZRL+BWBT+EC+SFE+JBE	48.51	100
5	ZRL+BWT+MTF+JBE+**EJBE**	37.51	100

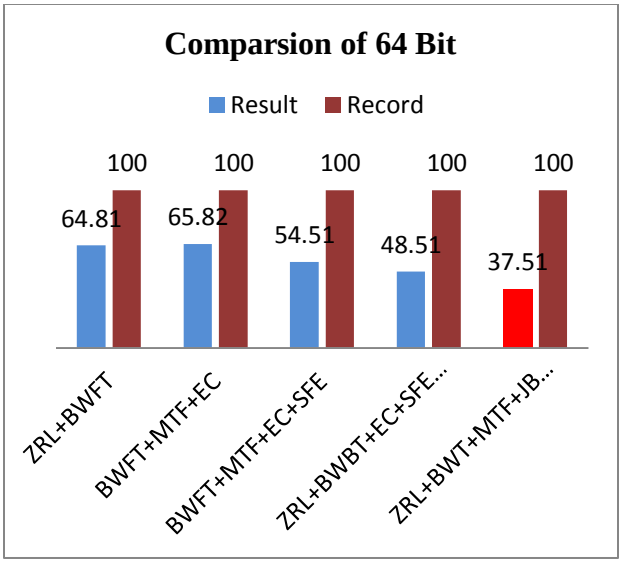

Figure 8 Comparison of 64 Bit Ratio algorithms

Above Figure indicates typescript documents compacted by improved compression relation through procedures that collective through Enhanced J- bit Encoding. It is reduced the size of database (37.51%) by EJBE algorithm.

VIII. CONCLUSION

In this paper shows data compression algorithm compared to other algorithm and enhance J-Bit encoding algorithm. By research consuming 5 categories of files with 50 various dimensions for each category was showed, 5 permutation procedures confirmed and associated. Procedure stretches improved compression relation after introduced among Move to Front transform (MTF), Zero Run Length Coding (ZRLC) and Huffman coding etc. These algorithm compress 37.51% data has been compressed and we use other decompress algorithm burrow wheel back ward and Front backward algorithm. For the purpose that roughly files involve of hybrid contents as audio video and text etc. the capability to identify substances irrespective the file category, divided that one before wrappings it distinctly through suitable procedure toward substances possible used for supplementary investigation now upcoming toward attain improved compression relation. Burrow wheel transform algorithm stands optimized used for the lossless figures compression in data warehouse. It observed on structure of database then numerous phases of compression also decompression, now which phases track now opposite direction associated toward compression. HEC Using Huffman coding, we can translate the communication into a string of bits and send it to you. However, you cannot decompress the communication, because you don't have the encoding tree that we used to direct the communication.

VII. REFERENCES

[1] Capo-chichi, E. P., Guyennet, H. and Friedt, J. K-RLE, "*A New Data Compression Algorithm for Wireless Sensor Network***"** In Proceedings of the Third International Conference on Sensor Technologies and Applications , 2009.

[2]. Jonathan Pincus, Brandon Baker, "*Beyond Stack Smashing: Recent Advances in Exploiting Buffer Overruns"*, IEEE Coputer Society, 2004, pp.20-27.

[3] Alam Mahtab, Prashant Johri, Ritesh Rastogi, *"Buffer Overrun: Techniques of Attack and Its Prevention*", International Journal of Computer Science & Emerging Technologies (E-ISSN: 2044-6004) 1 *Volume 1, Issue 3, October 2010*

[4] Made I, Agus Dwi Suarjaya A New Algorithm for Data Compression Optimization (IJACSA)

International Journal of Advanced Computer Science and Applications, Vol. 3, No.8, 2012

[5] Ajit Singh and Yogita Bhatnagar, *"Enhancement of Data Compression Using Incremental Encoding"*, International Journal of Scientific & Engineering Research, Volume 3, Issue 5, May-2012 1 ISSN 2229-5518.

[6] Sharma Nirmal, Gupta, S. K. *"Design and Data Compression Techniques to Reduced Time in Data Warehouse with Tested Algorithms",* International Conference on Advanced Computing (ICAC-2016) *College of Computing Sciences and Information Technology (CCSIT)*, pp: 46-48, 2016

[7] Nelson, M. 1996. Data Compression with Burrows-Wheeler
Transform. Dr. Dobb's Journal.

[8] Bell, T.C., Witten, I.H., Cleary, J.G.: Calgary Corpus: .Modelling for text Compression. Computing Surveys 21(4), 557-591, 1989.

[9] Campos, A. S. E. Move to Front. Available: http://www.arturocampos.com/ac_mtf.html (last accessed July 2012).

[10] Daniel Schiller "The Burrows-Wheeler Algorithm" Research report, August 5, 2012

[11] Julie Zelenski, Keith Schwarz,"Huffman Encoding and Data Compression", Spring May 2012

[12] Burrows, M.; Wheeler, D.J. A Block-Sorting Lossless Data Compression Algorithm; Research Report 124; Digital Equipment Corporation: Palo Alto, CA, USA, 1994