SERBIAN JOURNAL OF ELECTRICAL ENGINEERING
Vol. 19, No. 3, October 2022, 351-368

UDC: 621.396.2:519.688 **DOI: https://doi.org/10.2298/SJEE2203351A**

On the Application of Wavelet Transform and Huffman Algorithm to Yorùba´Language Syntax Text Files Compression

Kamoli Akinwale Amusa[1], Adeoluwawale Adewusi[2], Tolulope Christiana Erinosho[1], Sule Ajiboye Salawu[3], David Olugbenga Odufejo[1]

Abstract: Most algorithms of data compression were developed with English language as target text syntax**.** However, this paper approaches the problem of Yorùba´text files compression via the use of Discrete Wavelet Transform (DWT) and Huffman algorithm. Text files in Yorùba´language syntax are first converted into signal format that are then decomposed using DWT. The decomposed ASCII code representation of the text files are subsequently encoded using Huffman algorithm. Twenty different variants of DWTs taken from four families of wavelet filters (Haar, Daubechies, Symlets and bi-orthogonal) are considered to select the optimal DWT for Yorùba´ text files compression. Furthermore, experiments are carried out in the proposed compression scheme with six different Yorùba´text files extracted from the open sources as input data sets. It is found that out of the twenty variants of DWT investigated, sym6 gives the best output for effective Yorùba´text files compression, due to its relatively high compression ratio, high compression factor and lowest compression error. Thus, sym6 as a wavelet transform is suitable for lossy text compression algorithm meant for Yorùba´ language syntax text files.

Keywords: Text file, Compression, Wavelet transform, Huffman coding, Yorùba´ language syntax.

1 Introduction

Growth in telecommunication services which results into the integration of several technologies have placed more demand for efficient storage and transmission of data. Data could either be digital or non-digital; and could be in form of speech, audio, text, video, and computer information. Non-digital data

[1]Electrical and Electronic Engineering Department, Federal University of Agriculture, Abeokuta, Nigeria;
E-mails: amusaka@funaab.edu.ng; erinoshotc@funaab.edu.ng; davidodufejo@gmail.com
[2]Electrical Engineering and Computer Science Department, Technische Universität, Berlin, Germany;
E-mail: a.adewusi@campus.tu-berlin.de;
[3]Computer Education Department, Aminu Saleh College of Education, Azare, Nigeria;
E-mail: salawu_sul@yahoo.com

are easily rendered into a discrete format for easy transmission, while digitised text files data transfer and storage occupy substantial volume of data exchange on the Internet among digital libraries and archival organisations [1]. This brings to the fore, the need for compression of text files to facilitate efficient utilization of transmission resources, in terms of bandwidth and throughput, as well as the memory requirement for storage.

The first task in text file compression is the rendering or extraction of representation of the text data such that it can be treated as a discrete signal. The ASCII code representation and its extension, Unicode representation; provide avenue for coding of text file characters emanating from different languages of the World [2]. Once a text file is rendered into the signal representation, the actual compression routine can then be invoked to reduce the file size without compromising the integrity of the text file content.

Several methods have been proposed in the literature for text file compression. Approaches of lossless data compression include Huffman coding [3], arithmetic coding [4] and dictionary lexicons-based method [5, 6], which is based on statistical distribution of characters of text files. Other lossless compression methods employ substitution technique in its construction [7]. In lossy compression approach, redundant characters are basically removed from the signal representation of the text files and the remaining characters are thereafter processed for the text file reconstruction.

From the foregoing, it is obvious that lossy compression will produce a relatively higher compression ratio than any of the lossless compression methods. However, the reconstructed file is just an approximation of the original text file data. Ample efforts have been expended in the quest for reliable lossy compression methods in the literature [8–10], with the main aim of improving the text file compression ratio. Typical examples of lossy text file compression techniques include the wavelet transform-based method [11–14], dropped vowels, letter mapping and character replacement algorithm [15]. Moreover, some other investigators have approached the problem of lossy text file compression via the combination of two methods to realize hybrid technique. Azad et al. [16] proposed a two-level method for text file compression. At the first level, text file is reduced without compression using a word look-up table, while text file is compressed using the deflate compression algorithm at the second level. This algorithm compresses the text data using both Huffman coding and LZ-77 algorithm. Through this arrangement, the authors claimed an un-quantitative amount of save in the memory requirement by purely English text-based data. Sidhu and Garg [17] presented a hybrid compression method involving dynamic bit reduction and Huffman coding for text data compression.

In this paper, a combination of wavelet transforms and Huffman coding techniques is applied to the problem of text file compression, specifically to text file that are based on Yorùbá language syntax. The background theory on wavelet transform and Huffman coding is briefly reviewed in Section 2, while the proposed methodology was presented in Section 3. The performance metrics adopted for the evaluation of the proposed text file compression method are given as well as database of text files used are reviewed in Section 4, while results and discussion follow in Section 5 and Section 6 concludes the paper.

2 Background Theory

2.1 Wavelet transform

Wavelets are orthonormal basis functions which enable transformation of signals from their original domain into another domain in order to facilitate faster and efficient processing. Mathematically, a wavelet is any function $\psi\left(t\right)\in L^2\left(R\right)$ where $L^2\left(R\right)$ is a space containing square-integrable functions on the set of real number R. The necessary and sufficient condition (admissibility condition) for the existence of $\psi(t)$ is $\psi(t)=0$, which implies that

$$\int_{-\infty}^{\infty}\psi(t)\,\mathrm{d}t=0\,. \tag{1}$$

The implication of (1) is that all functions having zero integral are wavelets, which are generated from a single basic function called the mother wavelet via wavelet parameters: dilating (scaling) and translating parameters.

Suppose $\psi(t)$ denotes the mother wavelet with *a* and *b*, representing the dilating and translating parameters, respectively, the wavelet basis $\psi_{a,b}(t)$ is expressible as

$$\psi_{a,b}(t)=\frac{1}{\sqrt{a}}\psi\left[\frac{t-b}{a}\right], \tag{2}$$

where $a,b\in R,\ a\neq 0$ and the factor $\left(\sqrt{a}\right)^{-1}$ is used as the normalization factor to ensure that the transformed signal has equal energy at every scale.

Depending on the values of the input signal and those of dilating and the translating parameters (a,b), wavelet transforms are classified either as Continuous Wavelet Transform (CWT) or Discrete Wavelet Transform (DWT). In the former, the input signal and the parameters $\left(a,b\right)$ are continuous while in the latter, (a,b) and the input signal assume discrete values.

The CWT representation of signals often contain redundant information due to the continuous nature of parameters (a,b) over R. This constitutes a major drawback in the use of CWT for data compression applications. In order to overcome this limitation, DWT (which is the sampled version of the continuous signal) is widely applied for compression tasks. The DWT is expressed as follows [12, 18]:

$$\boldsymbol{c}_{j,k} = \sum_{n=-\infty}^{\infty} \mathrm{x}(n)\,\boldsymbol{\Phi}_{j,k}(n) \tag{3}$$

and

$$\boldsymbol{d}_{j,k} = \sum_{n=-\infty}^{\infty} \mathrm{x}(n)\,\boldsymbol{\psi}_{j,k}(n)\,, \tag{4}$$

where $\boldsymbol{c}_{j,k}$ and $\boldsymbol{d}_{j,k}$ are the approximated and detailed coefficients, respectively; $j,k \in Z$, Z being a set of positive integers; $\mathrm{x}(n)$ is the input discrete signal; and $\boldsymbol{\Phi}(n)$ represents the scaling function such that

$$\boldsymbol{\Phi}_{j,k}(n) = \sqrt{2^j}\,\boldsymbol{\Phi}\left(2n-k\right) \tag{5}$$

and $\boldsymbol{\psi}(n)$ is the mother wavelet, which allows

$$\boldsymbol{\psi}_{j,k}(n) = \sqrt{2^j}\,\boldsymbol{\psi}\left(2n-k\right). \tag{6}$$

Reconstruction of the original signal $\mathrm{x}(n)$ from the decomposed or transformed wavelets is done via inverse transform described as:

$$\mathrm{x}(n) = \sum_{k=-\infty}^{\infty} \boldsymbol{c}_{l,k}\boldsymbol{\Phi}_{l,k}(n) + \sum_{j=l}^{\infty}\sum_{k=-\infty}^{\infty} \boldsymbol{d}_{j,k}\boldsymbol{\psi}_{j,k}(n)\,, \tag{7}$$

where l is the last level of the decomposition stage employed in the wavelet transformation.

In order to facilitate speedy wavelet transformation process, it has been proposed elsewhere [18] that (3) can be replaced by

$$\boldsymbol{c}_j(n) = \sum_{m=-\infty}^{\infty} \mathrm{h_a}(m-2n)\,\boldsymbol{c}_{j+1}(m) \tag{8}$$

and (4) by

$$\boldsymbol{d}_j(n) = \sum_{m=-\infty}^{\infty} \mathrm{g_a}(m-2n)\,\boldsymbol{c}_{j+1}(m)\,, \tag{9}$$

where $\mathrm{h_a}$ and $\mathrm{g_a}$ are low-pass and high-pass filters, respectively. The two filters $\mathrm{h_a}$ and $\mathrm{g_a}$ are related by

$$\mathrm{g_a}(n) = (-1)^n\,\mathrm{h_a}(n-1)\,. \tag{10}$$

In this paper, four functions from the mother wavelets of the DWT are employed, which are Haar, Daubechies, Symlets and bi-orthogonal. The qualitative descriptions of the four functions are highlighted in what follows.

Haar wavelet is the simplest orthonormal wavelet basis. Conceptually, it is simple, memory efficient, with no overlapping windows and this account for its low computational overhead. On the other hand, its major drawback lies in its discontinuity which is a potential issue in application specific tasks like noise removal and compression. The Daubechies wavelet transforms are described in a similar fashion as the Haar transform by evaluating the running averages and differences via appropriation of the scalar products with scaling signals and wavelets. Owing to its orthogonal and compact characteristics, it has enjoyed wide applications in texture analysis. This wavelet type has balanced frequency responses with non-phase responses that are non-linear.

Bi-orthogonal wavelets belong to the class of compactly supported symmetric wavelets. It is most times desirable that a filter coefficient exhibits a symmetric characteristic such that a linear phase response is obtainable. In this regard, bi-orthogonality is characterized by two scaling functions that can produce diverse multi-resolution analysis which results in two unique wavelet functions. Further details and necessary conditions on the realization of this class of wavelet can be found in [19]. Symlets is a modification to the class of Daubechies wavelets [20]. It is characterized by an increasing symmetry. While Daubechies wavelets have a maximal phase, Symlets wavelets exhibit a minimal phase structure. Furthermore, Symlets wavelets show a good performance with increasing signal to noise ratio of reconstructed signals when applied to de-noising task [21].

Having highlighted and introduced the wavelet transform, Huffman encoding is introduced to complete the two components of the proposed compression method.

2.2 Huffman encoding technique

Huffman encoding is a popular lossless data compression algorithm that yields variable-length code for different input characters. It is widely applied in signal processing tasks, especially in the area of image compression. It is specifically developed to reduce the redundancy in coding without compromising the quality of data. The length of the code is strictly a function of the frequency of characters usage in the data string. While a character that is used often is associated with the smallest code, a character that is least used is assigned the longest code. Huffman source coding technique comprises of two main parts: the creation of a Huffman tree for the data string, and assignment of codes to characters of the data string though traversing of the created Huffman tree. Steps involved in the Huffman algorithm are highlighted below [22]:

(i) A list of symbols frequencies is prepared along with the probabilities of symbols usage in descending order.

(ii) A node, which is a binary tree, is created from the obtained probabilities.

(iii) Two lowest probabilistic symbols in the cluster are retrieved and have their probability values aggregated to form a new probability. These probabilities are governed by a descending order.

(iv) A parent node is formed with the left and right branches marked as child node 1 and child node 0, respectively.

(v) The two nodes having the smallest probabilities are changed with the tree list updated in order to create a new node. If updated list has only one node, then the process ends otherwise, repeat (ii) – (v).

On the whole, the level of computational complexity for the process of encoding data string using Huffman algorithm is $\mathrm{O}(n \log n)$.

2.3 Description of the data

Yorùbá language is a dialect cluster of mostly inhabitants of south-western region of Nigeria, southern Republic of Benin and central region of Republic of Togo, all in West Africa. Traces of Yorùbá language speakers can be found in Brazil and Cuba, where the language is named Nago or Lucumi. Besides reference to combinations of dialects and people speaking the language, Yorùbá, as a word refers to the standard form of the language in the text form. Yorùbá belongs to Niger-Congo language from the Yoruboid branch of Defoid, Benue-Congo. Yoruboid comprises of Igala (spoken by people residing to the east of Yoruba land, the Edekiri group (spoken by many in Nigeria and Benin), which includes Ede Ica, Ede Cabe, Ifẹ̀, Ede Ije, Ede Nago), Itsekiri and Yorùbá cluster proper, which has over fifteen variants. The Yorùbá proper cluster is often classified into three main dialect zones [23]: Northwest Yorùbá (Abẹ́òkúta, Ìbàdàn, Ọ̀yọ́, Ọ̀sun and Lagos areas), Central Yorùbá (Ìgbóńǹà, Ifẹ̀, Ekìtì, Àkúrẹ́, Ẹfọ̀n and Ìjẹ̀sà areas), South east Yorùbá (Òkìtìpupa, Òǹdó, Ságámù and Ìjẹ̀bú).

Common Yorùbá often regarded as the standard Yorùbá is a distinct member of the dialect cluster that is used in writing text of the language that is learnt at school and utilize by media practitioners in their work. Its origin can be traced to the 1850's effort of Bishop Samuel Àjàyí Crowther who published a Yorùbá grammar in his quest to translate the Bible into Yorùbá language. Common Yorùbá has its peculiar features such as the simplified vowel harmony style among others, however, its orthography reflects (to a large extent) the Abẹ́òkúta dialect while the morpho-syntax is closely related more to the Ọ̀yọ́-Ìbàdàn dialects [23 – 24].

3 Methods

Fig. 1 illustrates the proposed compression scheme in blocks. Besides the input and output, five other stages involved, are:

(i) Conversion of the text files into signal representation;

(ii) Decomposition of the signal representation of the text file using DWT;

(iii) Encoding of the decomposed version of the signal representation of the text file using Huffman algorithm;

(iv) Decoding of the Huffman encoded data string to obtain compressed estimates of the original signal; and

(v) Reconstruction of the decoded data signal via Inverse DWT to have compressed version of the text file.

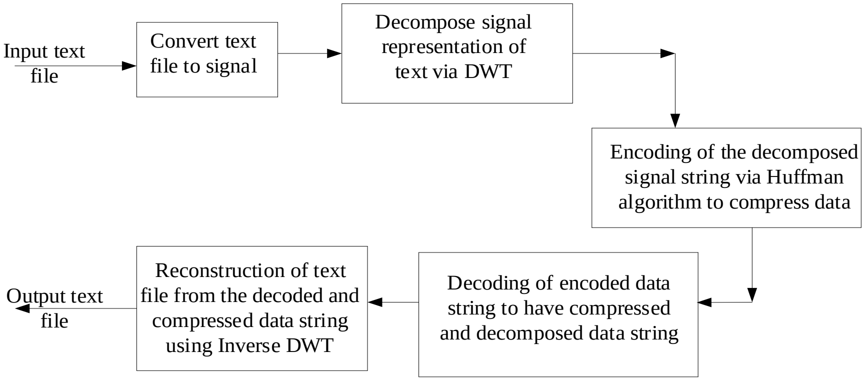
Input text file
Convert text file to signal
Decompose signal representation of text via DWT
Encoding of the decomposed signal string via Huffman algorithm to compress data
Decoding of encoded data string to have compressed and decomposed data string
Reconstruction of text file from the decoded and compressed data string using Inverse DWT
Output text file

Fig. 1 – *Block diagram of the proposed text file compression scheme.*

The first stage, which involves the conversion of the text file into appropriate signal representation, can be described as pre-processing of the text file. Here, the text file is read as input data and it is subsequently transformed into proper signal representation that is suitable for decomposition by wavelet transform. Irrespective of the format the text file is prepared or saved, the file is opened, copied and saved into .txt format. Files saved in .txt format are read and converted into ASCII codes. In other word, conversion of the text files into signal representation is actually the sequence of ASCII codes of the letters of the text files. It is in this format that DWT decomposition is implemented. The proposed compression scheme was implemented in MATLAB environment. It is worth noting that the text file representation is one dimensional (1-D), thus the default value of maximum decomposition level of five for 1-D signal is adopted.

Since the signal representation of the text file is a 1-D input signal, application of DWT into such signal yields two output sequences namely "low-pass" and "high-pass" sequences. The transform is easily realized by directly invoking finite impulse response convolution.

Consider a one-dimensional input signal $\boldsymbol{X} = \{x_n\}$, its DWT is implemented as

$$\{\boldsymbol{\Phi}_i\} = \frac{1}{\sqrt{2}} \sum_i a_k \, \mathrm{x}(2i+k), \tag{11}$$

$$\{\boldsymbol{\psi}_i\} = \frac{1}{\sqrt{2}} \sum_i (-1)^k \, a_{2N-1-k} \, \mathrm{x}(2i-k), \tag{12}$$

where N is a positive integer, $k = 1, 2, \ldots, 2N-1$, and a_k is a real number.

It ought to be stated here that a total of twenty variants of DWTs, are selected from families of the four mother wavelets: Haar, Daubechies, Symlets and bi-orthogonal, are utilized in this work for the decomposition of the signal representations of text files. The intuition behind this is to enable a systematic analysis of the performance of each of the DWT variants for comparison purposes and consequent selection of top performing discrete filter for actual compression task of the Yorùbá language text file. The specific DWT filters selected are Haar (haar), Daubechies (db1 – db10), Symlets (sym2 – sym8), and bi-orthogonal (bior1.1 – bior1.3).

Encoding of the decomposed data string from the signal representation of the text file as output from DWT is realized using Huffman algorithm based on the frequency of characters present. The output of this stage yields compressed data string representation of the text file. Furthermore, compressed data string is decoded and inverse DWT is performed on the decoded compressed data string in order to reconstruct approximate representation of the text file. It worth noting that the compression scheme proposed in this work is an outcome of integration of lossy (wavelet transform) and lossless (Huffman algorithm) approaches. The fixed form soft thresholding scheme is used in the estimation of error involves in the MATLAB implementation of the wavelet transformation of the signal representation of text files.

The data employed for this work is Yorùbá language text files. Common Yorùbá comes with seven oral {i, u, e, o, ɛ (ẹ), ɔ (ọ), a} and five nasal {(ĩ (in), ũ (un), ɛ̃ (ẹ̀n), ɔ̃ (ọn), ã (an/ ọn)} vowels. No diphthongs exist in Yorùbá language and the sequences of vowels are pronounced separately as syllables. Yorùbá, being a tonal language, has every syllable in its construct bearing one of the three tones identified as: high (denoted with an acute accent (´), low (indicated with the grave accent (`), and mid (which is unmarked). The mid tone is the default tone and a long vowel in Yorùbá language can bear two tones [25]. This tonal structure changes the entire meaning of a word form that has exactly the same number of consonants and vowels.

Basically, the constituent order of the sentence structure in Yorùbá language is subject, verb and object. When a bare verb is used, it indicates a completed action; tense and aspect in Yorùbá language are identified by pre-verbal particles. In addition, serial verb constructs and associative construction, which consists of juxtaposing the noun phrase in the order modified-modifier, are common in Yorùbá language [26]. All these are examples of syntactic differences that exist between the text documents that are written in the Yorùbá language and those prepared in English language.

4 Performance Evaluation

In order to evaluate the performance of the proposed text file compression scheme, besides computations of the compressed file size arising from the use of different variants of DWT, three standard performance metrics employed are, Compression Ratio (CR), Compression Factor (CF) and Compression Error (CE). Let the original un-compressed text file size be represented as UCFS while the compressed file size output from the compression scheme is CFS, and that $\boldsymbol{X} = \{x_1, x_2, x_3, \cdots x_N\}$ is the signal string representation of the uncompressed text file whilst $\boldsymbol{X}_c = \{x_{c1}, x_{c2}, x_{c3}, \cdots x_{cN}\}$ is that of the compressed version of the text file. Then,

$$\text{CR}=\frac{\text{UCFS}}{\text{CFS}} \tag{13}$$

and

$$\text{CF}=\frac{\text{UCFS-CFS}}{\text{UCFS}}\,, \tag{14}$$

while

$$\text{CE} = \frac{\sqrt{\sum_i \left(\left|x_i - x_{ci}\right|\right)^2}}{\sum_i \left|x_i\right|}\,. \tag{15}$$

Since the work involves compression of text files that are prepared in Yorùbá language syntax, the experiment is carried out on text files that are prepared in Yorùbá language syntax. Three text files formed the composition of the Yoruba text files database employed in the experiment. Text file shown in Fig. 2, a short poem composed in Yorùbá language, is the first text file. The second one is extracted from [27]. The third file is an extract of the work reported in [28] where all non-Yorùbá words and references are expunged to give a sizable pure Yorùbá syntax text. The fourth test data is produced by reproducing the text file two, three times (increasing the text volume of test file two three times) while the fifth test data contains a mixture of test file two and test file 3. The last test data file contains text file three that is reproduced three times. Altogether, six text files of different sizes were experimented. **Table 1** contains information about six different Yorùbá text files employed in the experiment.

Table 1
Filenames and sizes of text files employed for the experiment.

Experiment No.	Composition of text files employed	File sizes [kB]
1	Text-file 1	0.67
2	Text-file 2	4.52
3	Text-file 3	21.40
4	Three copies of Text-file 2 merged together	13.50
5	Text-files 2 and 3 copied and merged together	32.90
6	Two copies of Text-file 3 merged together	56.80

Tọ́jú ìwà rẹ ọ̀rẹ́ mi
Tọ́jú ìwà rẹ ọ̀rẹ́ mi
Ọlá a máa ṣí lọ nílé ẹni
Ẹwà a sì máa ṣí lára ènìyàn
Ṣùgbọ́n ìwà ní í bá ‘ni dé sàárè
Èéfín nìwà, rírú ní í rú
Ènìyàn gb’ókèèrè níyì
Ṣùgbọ́n súnmọ́ ni, l’a fi ń mọ̀’ṣe ẹni
Ìwà kò ní í foníwà sílẹ̀
Ìwà ọmọ l’ó ń sọmọ lórúkọ
Ọmọ dára ó ku ìwà
Ara dára ó ku aṣọ
Ẹsẹ̀ dára ó ku bàtà
B’énìyàn dára tí kò níwà
Ó padanù ohun ribiribi
Ìwà rere l’ẹ̀ṣọ́ ènìyàn
Ṣùúrù baba ìwà, ìwà baba àwúre

Fig. 2 – *Text-file* 1 *employed in experiment* 1.

Having described metrics for performance evaluation of the proposed compression scheme as well as text files for experimentations, presented in what follows are obtained results are discussion.

5 Results and Discussion

The results of decomposition before compression of the six Yoruˋbaˊlanguage syntax text files using DWT variants; dealing with them as signals (non-stationary signal files) is our major undertaken in this section. In order to isolate and select a DWT variant that yields the best results in terms of performance metrics adopted for this work, each of the twenty DWT variants employed is put test using the six text files described in **Table 1**. **Table 2** gives numerical values of computed compression ratios, while **Table 3** presents specific numerical values of compression factors associated with each of the adopted variants of DWT in the proposed compression scheme from the six experiments.

Table 2

Compression ratio resulting from the experimentation of the proposed compression scheme and the corresponding DWT variants employed.

S/N	DWT variants	Values of CR associated with DWTs in experiments					
		1	2	3	4	5	6
1	Haar	1.67	1.52	1.57	1.52	1.61	1.72
2	db1	1.67	1.52	1.57	1.52	1.61	1.72
3	db2	1.66	1.51	1.53	1.51	1.59	1.69
4	db3	1.60	1.48	1.49	1.48	1.52	1.61
5	db4	1.62	1.50	1.51	1.50	1.56	1.66
6	db5	1.63	1.50	1.53	1.50	1.58	1.68
7	db6	1.49	1.44	1.46	1.44	1.47	1.55
8	db7	1.64	1.50	1.53	1.51	1.59	1.70
9	db8	1.61	1.49	1.51	1.49	1.56	1.66
10	db9	1.64	1.50	1.52	1.50	1.58	1.69
11	db10	1.65	1.27	1.53	1.51	1.59	1.69
12	sym2	1.66	1.51	1.53	1.51	1.59	1.69
13	sym3	1.60	1.48	1.49	1.48	1.52	1.61
14	sym4	1.65	1.50	1.53	1.50	1.59	1.40
15	sym5	1.65	1.50	1.53	1.50	1.59	1.69
16	sym6	1.65	1.50	1.53	1.50	1.59	1.70
17	sym7	1.61	1.48	1.51	1.48	1.57	1.67
18	sym8	1.49	1.43	1.45	1.43	1.46	1.54
19	bior1.1	1.67	1.52	1.57	1.52	1.61	1.72
20	bior1.3	1.66	1.52	1.56	1.52	1.61	1.72

Results presented in **Table 2** reveal that the DWTs behave differently with varying compression ratio in the six experiments. The lowest compression ratio recorded is 1.49 corresponding to db6 and sym8 (for experiment 1); 1.27 when db10 is used (in experiment 2); 1.45 associated with sym8 (in experiment 3); 1.43 corresponding to sym8 (in experiment 4); 1.46 also for sym8 (in experiment 5); and 1.54 for sym8 (in experiment 6). The highest values of recorded CR are 1.67 (Haar, db1, bior1.1); 1.52 (Haar, db1, bior1.1, bior1.3); 1.57 (Haar, db1, bior1.1); 1.52 (Haar, db1, bior1.1, bior1.3); 1.61 (Haar, db1, bior1.1, bior1.3); and 1.72 (Haar, db1, bior1.1, bior1.3) for experiments 1 – 6, respectively. The average values of CR, for experiments 1 – 6, are 1.626, 1.4835, 1.5225, 1.4960, 1.5695 and 1.6555, respectively.

Table 3
Compression factor resulting from the experimentation of the proposed compression scheme and corresponding DWT variants employed.

S/N	DWT variants	Values of CF (%) associated with DWTs in experiments					
		1	2	3	4	5	6
1	Haar	40.12	34.21	36.31	34.21	37.89	41.86
2	db1	40.12	34.21	36.31	34.21	37.89	41.86
3	db2	39.76	33.77	34.64	33.77	37.11	40.83
4	db3	37.50	32.43	32.89	32.43	34.21	37.89
5	db4	38.27	33.33	33.77	33.33	35.90	39.76
6	db5	38.65	33.33	34.64	33.33	36.71	40.48
7	db6	32.89	30.56	31.51	30.56	31.97	35.48
8	db7	39.02	33.33	34.64	33.77	37.11	41.18
9	db8	37.89	32.89	33.77	32.89	35.90	39.76
10	db9	39.02	33.33	34.21	33.33	36.71	40.83
11	db10	39.39	21.26	34.64	33.77	37.11	40.83
12	sym2	39.76	33.77	34.64	33.77	37.11	40.83
13	sym3	37.50	32.43	32.89	32.43	34.21	37.89
14	sym4	39.39	33.33	34.64	33.33	37.11	28.57
15	sym5	39.39	33.33	34.64	33.33	37.11	40.83
16	sym6	39.39	33.33	34.64	33.33	37.11	41.18
17	sym7	37.89	32.43	33.77	32.43	36.31	40.12
18	sym8	32.89	30.07	31.03	30.07	31.51	35.06
19	bior1.1	40.12	34.21	36.31	34.21	37.89	41.86
20	bior1.3	39.76	34.21	35.90	34.21	37.89	41.86

Similar to CR results, each of the adopted DWT variants behaves differently in terms of the compression factor. The lowest values of CF are 32.89% (associated with db6 and sym8 in experiment 1), 21.26% (corresponding to db10 in experiment 2), 31.03% (recorded by sym8 in experiment 3), 30.56% (linked with db6 in experiment 4), 31.51% (returned by sym8 in experiment 5) and 35.06% (had by sym8 in experiment 6). On the high side, CF figures for experiments 1 – 6, are 40.12% (Haar, db1, bior1.1), 34.21% (Haar, db1, bior1.1, bior1.3), 36.31% (Haar, db1, bior1.1), 34.21% (Haar, db1, bior1.1, bior1.3), 37.89% (Haar, db1, bior1.1, bior1.3), and 41.86% (Haar, db1, bior1.1, bior1.3), respectively. The average values of the CF figures are 38.436%, 32.488%, 33.1355%, 36.238% and 39.448% for experiments 1 – 6, respectively.

Owing to the low performance of some DWT variants, in terms of CR and CF (as can be inferred from **Tables 2** and **3**, respectively), it is essential to prune the number of DWT variants. Thus, to successfully prune the compression models, a selection criterion for satisfactory DWT compression model is proposed such that, a DWT variant must have its value of CR (or CF) greater than the average value obtainable in at least four out of the six experiments.

The following DWTs met the above stated selection criterion: Haar, db1, db2, db5, db7, db9, db10, sym2, sym4, sym5, sym6, bior1.1 and bior1.3. Others are dropped and considered not suitable for further analysis. These thirteen DWT variants are further analysed to arrive at the most suitable DWT for the proposed compression scheme for Yorùbá language syntax text files.

Presented in **Table 4** are computed compression errors, associated with the use of the thirteen DWT variants that satisfied the CR (or CF) selection criterion, for the six experiments.

Table 4
Compression error associated with the usage of selected DWTs in the proposed compression scheme.

S/N	DWT variants	Values CE associated with experiments $(\times 10^{-10})$					
		1	2	3	4	5	6
1	Haar	0.54	2.34	20.27	7.09	27.96	51.46
2	db1	0.54	2.34	20.27	7.09	27.96	51.46
3	db2	15.05	5.85	8.63	10.06	36.08	46.13
4	db5	40.49	12.42	155.39	5.18	111.96	271.92
5	db7	47.44	7.61	74.48	4.08	22.52	106.01
6	db9	562.59	87.86	507.41	110.74	32.95	571.32
7	db10	127.87	26.18	203.52	6.76	70.09	293.08
8	sym2	15.05	5.85	8.63	10.06	36.08	46.13
9	sym4	7.44	8.30	50.51	22.13	82.95	144.65
10	sym5	0.90	7.76	61.01	22.78	90.22	163.99
11	sym6	7.46	1.42	5.91	1.32	1.54	11.63
12	bior1.1	0.54	2.34	20.27	7.09	27.96	51.46
13	bior1.3	0.55	2.34	19.45	7.12	27.98	51.05

A cursory look at **Table 4** shows that in experiment 1, db9 has the highest CE figure of 562.59×10^{-10} while the lowest value recorded is 0.54×10^{-10} (corresponding to Haar, db1 and bior1.1). For experiment 2, sym6 and db9 have the lowest and highest CE values of 1.42×10^{-10} and 87.86×10^{-10}, respectively while in experiment 3, the lowest and highest values of CE are 5.91×10^{-10} and 507.41×10^{-10}, corresponding to sym6 and db9, respectively. In the case of experiments 4, 5 and 6, sym6 has the lowest CE figures with values of $(1.32, 1.54, 11.63)\times 10^{-10}$, respectively, in that order, whereas the highest values of CE are 110.74×10^{-10} (associated with db9); 111.96×10^{-10} (returned by db5); and 571.32×10^{-10} (corresponding to db9) for experiments 4, 5 and 6, respectively, in that order.

From the foregoing, it is obvious from that the followings DWT variants are associated with high values of CE in at least four out of the six experiments: db2, db5, db7, db9, db10 and sym2 while Haar, db1, bior1.1 and bior1.3 equally have relatively high values of CE except in the first, second and fourth experiments where CE figures are relatively small **(Table 4)**. It is only sym6 that consistently exhibited low values of CE in all of the six experiments. This suggests that sym6 output, in comparison with those of other twelve DWT variants, is the best. The thirteen DWT models, the compressed file size is thereafter determined for each of the resultant compression algorithms with respect to the six set of experimental simulations (**Table 5)**.

Table 5
Compressed file size associated with selected DWTs.

S/N	DWT variants	Compressed file size [kB]					
		1	2	3	4	5	6
1	Haar	0.401	3.054	14.003	8.880	20.445	33.791
2	db1	0.401	3.054	14.003	8.880	20.445	33.791
3	db2	0.404	3.074	14.307	8.950	20.687	34.320
4	db5	0.410	3.095	14.370	9.017	20.820	34.593
5	db7	0.408	3.081	14.286	8.969	20.657	34.279
6	db9	0.408	3.098	14.386	9.015	20.778	34.479
7	db10	0.406	3.651	14.322	8.944	20.743	34.410
8	sym2	0.404	3.074	14.307	8.950	20.687	34.320
9	sym4	0.406	3.098	14.361	9.021	20.678	41.508
10	sym5	0.406	3.086	14.359	8.993	20.702	34.342
11	sym6	0.406	3.095	14.354	9.014	20.681	34.245
12	bior1.1	0.401	3.054	14.003	8.880	20.445	33.791
13	bior1.3	0.404	3.048	14.023	8.883	20.488	33.908

Based on the results of associated compression error with DWT variants, it is easily observable that each of these DWT variants: sym4, sym5, Haar, db1, bior1.1 and bior1.3 perform better than sym6 (**Table 4**) in the first experiment going by their CE figures. However, in terms of the actual compressed file size, the margin is very small as Haar, db1 and bior1.1 have comparable reduced compressed file size advantage of 0.005 kB (0.75% of un-compressed file size) while bior1.3 has 0.002 kB (0.3% of un-compressed file size) over the compressed file size yielded by sym6. In the remaining five experiments (2 – 6), sym6 has the lowest CE values, which are $(1.42, 5.91, 1.32, 1.54, 11.63)\times 10^{-10}$, respectively (**Table 5**). The closest CE figures to those of sym6 in experiments 2 - 6, are 2.34×10^{-10} (for Haar, db1 and bior1.1), 8.63×10^{-10} (for db2 and sym2), 4.08×10^{-10} (for db7), 22.52×10^{-10} (for db7) and 46.13×10^{-10} (for db2 and sym2), respectively, in experiments 2–5, and 6 respectively. It is noteworthy that marginal gains of reduced compressed file size recorded by DWT variant(s) with closest CE figures are 0.041 kB (0.91% of un-compressed file size), 0.047kB (0.22% of un-compressed file size), 0.045kB (0.33% of un-compressed file size), 0.014 kB (0.04% of un-compressed file size), respectively, for experiments 2 – 5 whereas sym6 perform better than the closest DWT variant in the sixth experiment by having reduced file size of 0.075 kB (0.13% of un-compressed file size).

The trend of marginal gains of reduced compressed file size recorded over sym6 by other DWT variants that have close values of CE figures from experiments 2-5 shows that the performance of sym6 improves with increase in the size of the input text file. The original sizes (in kB) of the input text files 2, 4, 3, 5 and 6, are 4.52, 13.50, 21.40, 32.9 and 56.80, respectively. In terms of reduced compressed file size, the marginal gains of the compression scheme using sym6 over other DWT variants with closer CE figures, are -0.91%, -0.33%, -0.22%, -0.04% and +0.13%, respectively, for experiments 2, 4, 3, 5 and 6. Thus, if the emphasis is placed on DWT with the lowest value of CE, it is obvious from these results that sym6 performance surpasses those of others nineteen DWT variants employed in this paper, for the development of compression scheme for Yorùbá language syntax text files. The only exception recorded in the first experiment may be due to the relatively small size of the text file involved, as it can be observed that the performance of sym6 improves with increasing size of the text file input into the compression scheme.

In order to prove compression improvement, the performance of the hybrid of sym6 DWT and Huffman scheme adjudged here (as the most suitable for Yorùbá text files compression) is compared with those of existing methods in the literature. Consequently, three existing algorithms adopted as baselines are binary Huffman method, arithmetic coding and LZ coding. The test files in Table 1 and all the test procedures are followed. The final results are displayed in Table 6.

Results show that hybrid combination of sym6 DWT and Huffman algorithm in the proposed compression scheme for Yorùbá text files compression, perform better than each of those existing three methods as the approach returns the highest compression ratio in all text files employed experimentation (**Table 6**). This translates to the lowest compressed file size in each case.

Table 6

Comparison of compression ratios associated with different coding schemes

Text file size [kB]	Binary Huffman coding	Arithmetic coding	LZ coding	Hybrid Sym6-Huffman coding
0.67	1.52	1.52	1.50	1.65
4.52	1.33	1.35	1.18	1.50
13.50	1.47	1.46	1.35	1.53
21.40	1.34	1.35	1.42	1.50
32.90	1.45	1.44	1.33	1.59
56.80	1.50	1.49	1.31	1.70

6 Conclusion

The outcomes of this study show that quite a number of DWT variants can be employed in the realization of Yorùbá text files compression. However, it is established in this work that out of the array of twenty variants of DWT, which comprises of the Haar (haar), Daubechies (db1- db10), Symlets (sym2- sym8), and bi-orthogonal (bior1.1- bior1.3), sym6 is the most suitable candidate for the compression scheme meant for Yorùbá text files. It exhibits the best characteristics of relatively high compression ratio, high compression factor and lowest compression error. It is further shown that performance of proposed compression scheme utilizing sym6 improves with increase in the size of input file to be compressed. The compression ratio resulting from the use the sym6 DWT and Huffman algorithm in hybrid form shows an improved performance over baseline compression models (binary Huffman coding, arithmetic coding and LZ coding). Conclusively, sym6 is the most suitable DWT for the development of lossy text compression algorithm for Yorùbá language syntax text files. This invariably conserves the memory space required for the storage of such files and also enhances the transmission capability of the files.

7 References

[1] A. Jain, R. Patel: An Efficient Compression Algorithm (ECA) for Text Data, Proceedings of the International Conference on Signal Processing Systems (ICSPS), Singapore, Singapore, May 2009, pp. 762 – 765.

[2] D. Ewell: UTN #14: A Survey of Unicode Compression, 2004, Retrieved 14th August, 2018.

[3] D. A. Huffman: A Method for the Construction of Minimum-Redundancy Codes, Proceedings of the IRE, Vol. 40, No. 9, September 1952, pp. 1098 – 1101.

[4] J. Rissanen, G. G. Langdon: Arithmetic Coding, IBM Journal of Research and Development, Vol. 23, No. 2, March 1979, pp. 149 – 162.

[5] J. Ziv, A. Lempel: Compression of Individual Sequences via Variable-Rate Coding, IEEE Transactions on Information Theory, Vol. 24, No. 5, September 1978, pp. 530 – 536.

[6] R. Gupta, A. Gupta, S. Agarwal: A Novel Data Compression Algorithm for Dynamic Data, Proceedings of the IEEE Region 8 International Conference on Computational Technologies in Electrical and Electronics Engineering (SIBIRCON), Novosibirsk, Russia, July 2008, pp. 266 – 271.

[7] M. Burrows, D. J. Wheeler: A Block-Sorting Lossless Data Compression Algorithm, Systems Research Center, Palo Alto, 1994.

[8] I. H. Witten, T. C. Bell, A. Moffat, C. G. Nevill-Manning, T. C. Smith, H. Thimbleby: Semantic and Generative Models for Lossy Text Compression, The Computer Journal, Vol. 37, No. 2, January 1994, pp. 83 – 87.

[9] K. Kaufman, S. T. Klein: Semi-Lossless Text Compression, International Journal of Foundations of Computer Science, Vol. 16, No. 6, December 2005, pp. 1167 – 1178.

[10] M. Cannataro, G. Carelli, A. Pugliese, D. Sacca: Semantic Lossy Compression of XML Data, Proceedings of the 8th International Workshop on Knowledge Representation Meets Databases (KRDB), Roma, Italy, September 2001, pp. 1 – 10.

[11] C. S. Burus, R. A. Gopinath, H. Guo: Introduction to Wavelets and Wavelet Transforms: A Primer, 1st Edition, Prentice-Hall Inc., London, Sydney, Toronto, 1998.

[12] A. Graps: An Introduction to Wavelets, IEEE Computational Science and Engineering, Vol. 2, No. 2, 1995, pp. 50 – 61.

[13] L. Cheng, X. Ji, F. Zhang, H. Huang, S. Gao: Wavelet-Based Data Compression for Wide-Area Measurement Data of Oscillations, Journal of Modern Power Systems and Clean Energy, Vol. 6, No. 6, November 2018, pp. 1128 – 1140.

[14] T.- C. Wu, K.- C Hung, J.- H Liu, T.- K. Liu: Wavelet-Based ECG Data Compression Optimisation with Genetic Algorithm, Journal of Biomedical Science and Engineering, Vol. 6, No. 7, July 2013, pp. 746 – 753.

[15] V. Palaniappan, S. Latifi: Lossy Text Compression Techniques, Proceedings of the 15th International Workshops on Conceptual Structures (ICCS), Sheffield, UK, July 2007, pp. 205 – 210.

[16] M. A. K. Azad, R. Sharmeen, S. Ahmad, S. M. Kamruzzaman: An Efficient Technique for Text Compression, Proceedings of the 1st International Conference on Information Management and Business (IMB), Taipei, Taiwan, March 2005, pp. 467 – 473.

[17] A. S. Sidhu, E. M. Garg: Research Paper on Text Data Compression Algorithm Using Hybrid Approach, International Journal of Computer Science and Mobile Computing, Vol. 3, No. 12, December 2014, pp. 1 – 10.

[18] S. G. Mallat: A Theory for Multiresolution Signal Decomposition: The Wavelet Representation, IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol. 11, No. 7, July 1989, pp. 674 – 693.

[19] A. Cohen, I. Daubechies, J.- C. Feauveau: Biorthogonal Bases of Compactly Supported Wavelets, Communications on Pure and Applied Mathematics, Vol. 45, No. 5, June 1992, pp. 485 – 560.

[20] A. Glowacz: Diagnostics of Direct Current Machine based on Analysis of Acoustic Signals with the Use of Symlet Wavelet Transform and Modified Classifier based on Words, Eksploatacja i Niezawodnosc – Maintenance and Reliability, Vol. 16, No. 4, 2014, pp. 554 – 558.

[21] M. S. Chavan, N. Mastorakis, M. N. Chavan, M. S. Gaikawad: Implementation of SYMLET Wavelets to Removal of Gaussian Additive Noise from Speech Signal, Recent Researches in Communications, Automation, Signal Processing, Nanotechnology, Astronomy and Nuclear Physics, 2011, pp. 37 – 41.

[22] S. Haykin: Communication Systems, 4th Edition, John Wiley & Sons, Inc, New York, Chichester, Brisbane, 2001.

[23] A. Adetugbo: Towards a Yorùba´ Dialectology, Yoruba Language and Literature, Edited by A. Afọlayan, pp. 207 – 224, University Press, Ibadan, 1982.

[24] J. G. Fagborun: The Yoruba Koine – Its History and Linguistic Innovations, Lincom Europa, München, 1994.

[25] P. O. Ogunbowale: The Essentials of the Yoruba Language, University of London Press, London, 1970.

[26] O. Awobuluyi: Essentials of Yoruba Grammar, Oxford University Press Nigeria, Ibadan, 1978.

[27] M. Adekunle: Ladugbo Ọgbon, Oyo: High Rank Educational Services, 2013, pp. 1 – 4; 11 – 12. (In Yoruba)

[28] M. K. Akinleye: Aroko Pipa Lori Awọn Itakun Iroyin Toro-Fonkale, Yoruba: Journal of the Yoruba Studies Association of Nigeria, Vol. 9, No. 4, 2018, pp. 164 – 186.