Development of Hybrid Intelligent based Information Retreival Technique

Gregory Gabriel James
Department of Computer Science,
Rhema University, Aba,
Abia State, Nigeria

Abugor Ejaita Okpako
Department of CyberSecurity,
Faculty of Computing,
University of Delta,
Agbor,
Delta State, Nigeria

C. Ituma
Department of Computer Science,
Ebonyi State University,
Abakili,
Ebonyi State,Nigeria

J.E. Asuquo
Department of Chemistry,
University of Uyo, Nigeria

ABSTRACT
To find information over the internet to a certain level, depends on our capacity to track all related subjects and classify them into bunches of comparative themes. As the domain of information is enlarging over the internet , the time consumption and the difficulties experienced by researchers to find a relevant material that meets the user's specified request increases, thereby putting the researchers into a state of dilemma at the cause of searching for relevant information that meets their need. The pursuit to trim down the challenges of impasse faced by researchers as well as time exhausted to filter relevant materials in the pools of irrelevant materials have motivated this research. The work aims at developing a Neuro-fuzzy intelligent search framework for tracking and recovery of web archives. *The method used was Object-Oriented analysis and Design (OOAD).* A hybrid intelligent framework – based tracking system was utilized as the finest choice for tracking archives, since the shortcomings of Neural Network and Fuzzy Logic based tracking system were complemented while their individual qualities are upgraded. This paper expands prior Fuzzy-based information retrieval approaches through increasing the Fuzzy variables and their linguistic values by utilizing distinctive rules and functions that characterized the record. The mapping of input to output parameters was achieved by applying the triangular membership's functions. Adaptive neural fuzzy inference system model also utilized the Takagi Sugeno inference mechanism. It was observed that using ANFIS improved the hybrid intelligent framework – based tracking system performance slightly with 0.22641 representing 22.64% over the Fuzzy Inference System (FIS) results, thereby guarantee retrieval of most relevant documents that met the user's request.

Keywords
Intelligence, ANFIS model, Neuro-fuzzy, Geno Method

1. INTRODUCTION
Finding materials in the internet is constantly a complicated job among researchers. Generally, most researchers travel a very long way to gather required materials to meet their academic needs and this makes research quit expensive. Information technology changed the practice and brought about intelligent search methods. This intelligent search methods use the internet in facilitating the information search. The paper introduced an intelligent based paradigm for document tracking and retrieval. The classification of documents into similar topics and specific knowledge groups was done using neuro-fuzzy clustering method. The reason for utilizing this strategy is to construct a versatile intelligent data recovery framework. The framework will cluster web records into specific knowledge groups and similar themes by applying unsupervised machine learning procedure which decreases the rate of unimportant records recovered and displayed.

2. ANALYSIS OF THE PROPOSED SYSTEM
This new method is based on Neuro-Fuzzy approach which enhances the search for document. The merging of two advance technologies to form a single hybrid brilliant model makes the systems the leading technology for archive tracking and recovery. The reason for embracing the Neuro-Fuzzy strategy is to plan a versatile intelligent data recovery framework that utilized unsupervised learning procedures in diminishing the rate in which insignificant reports are recovered and displayed.

2.1 Architecture of the System
The Figure 1 shows the model of the proposed NFDTRS:

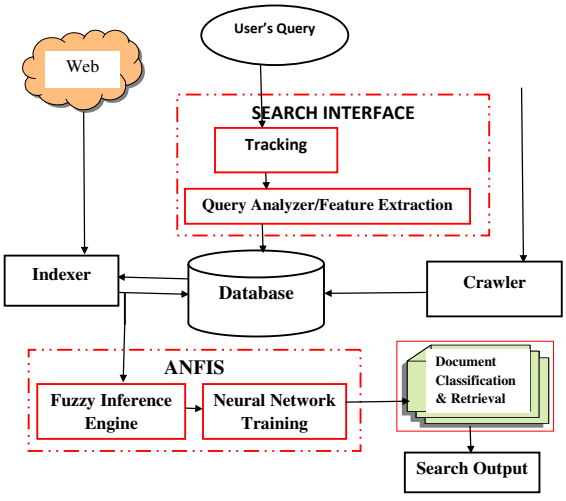

Fig. 1: The Proposed Neuro-Fuzzy Architecture

2.2 Conceptual Designs for the Software
The framework of this intelligent model in figure 1 is an improved work of [6]. The above figure is made up of different elements of the entire system with their functions clearly stated. The framework comprises the following components:

1. Search Interface
2. Database Model

3. Crawler
4. Indexer
5. ANFIS Model
6. Document Classification and Retrieval.

2.2.1 Search Interface

The main component of the Search interface is the query analyzer and features extraction. When a user enters query at the user interface, this component processes the request of the researcheras well as searching for appropriatetext. At First, the URL request is track, processed and a keyword is assigned to it [2], [4]. The admin based subsystem retrieves significant documents and forward same to the document database in preparation for classification according to this comparison. The component has the capability to use feedback, in which its grades the greatestapplicable documents from the set of recovered documents and thereforeforwardthe query as fresh request. The intelligent system generates a freshrequest from the existingtext-based documents and queststhe web repeatedly the web [6]. Thetriple subsystems platform that analyzed the query may also be given as a triple layer model in thediagram as provided in figure 2.

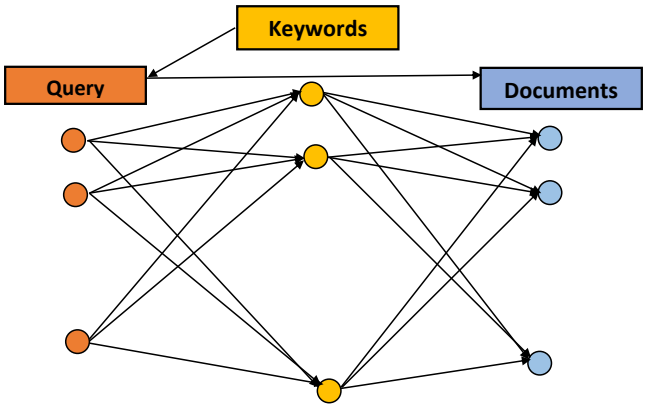

Fig. 2: Query Expansion System Framework (Source: Lee and Kim, 2012)

The front sub layer of this framework isquery sub-layer, the next is the important words sub-layer and next to the second is the document sub-layer. In this wise, users may type queries that are associated with an important word, based on the fact that the important documents are tracked from the document sub-layer [8]. Conversion of the material received by the query sub-layer addicted to keyword sub-layer and conversion of materialby the keyword sub-layer to text-based document sub-layer may then be substituted by neural networks.

Unseen and background stratum is applied by the enquiry illustration built upon the mathematical model:

$$net_{yj} = \sum_{i=1}^{N} w_{ij}\, x_i(t) + \theta_{yj}\ , j = 1 \ldots M \quad . \quad .(1)$$

The neuron y_j of these unseen stratums are summarized using the below sigmoid evolutionary function as:

$$y_j = f\left(net_{yj}\right) = \frac{1}{1+e^{-net_{yj}}}, \quad . \quad . \quad . \quad .(2)$$

To understudy neural network in this wise, a back propagation algorithmic model is implemented as:
If y_j in equation 1 is $f\left(net_{yj}\right)$ then equation 2 becomes

$$net_{yj} = \sum_{i=1}^{N} w_{ij}\, x_i(t) + \theta f\left(net_{yj}\right) \quad . \quad . \quad .(3)$$

If y_j is $\left(\frac{1}{1+e^{-net_{yj}}}\right)$ then equation (1) becomes,

$$net_{yj} = \sum_{i=1}^{N} w_{ij}\, x_i(t) + \theta\left(\frac{1}{1+e^{-net_{yj}}}\right) \quad .(4)$$

In machine learning system as well as pattern recognition and image processing systems, considered from initial set of restrained data and forms derived values (features) which is meant to be educated and non-stagnant, enabling the successive training and generalization phases. In most instances championing to enhanced human explanations [3]. Feature extraction is associated to reduction of dimensions. Feature extraction is all about the reduction in the quantity of resources that is needed to explain a big quantity of dataset [1].

2.2.2 Crawler

Mostly, ways through which web documents could be gathered are accumulating intelligent-based agents. The intelligent-based agents are mostly called crawlers, though may at times be identified by many other names as spiders, ants, automatic indexers, bots, web robots, and worms. Crawlers are provided with the beginning point on the internet known as kernel URL. This dynamic based software robots may then go ahead to traverse through the internet mining URLs that takes place in the document they visited at the moment[5], [9]. To develop this nature of software robots is a challenging job, because it involves many issues that cannot be foreseeing by the designer that may make it crashed or even make it unbalanced and impulsive behavior. In this paper, the crawler that allows an extreme linkage recovery count was developed. Sequel to this, it was possible to stop the crawler after retrieving to a given quantity of internet document from a given URL seed.

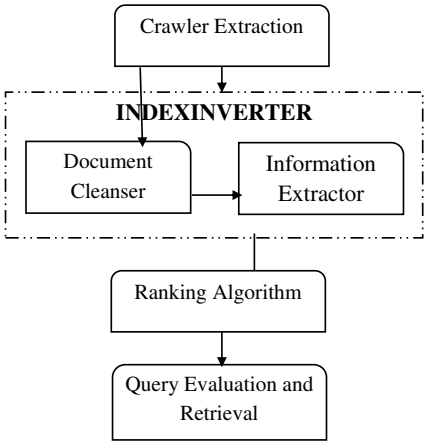

Fig. 3: Crawler Extracted System Framework (Source: Lee and Kim, 2012)

2.2.3 Indexer

Immediately the pages have been successfully extracted by the crawling system, these pages are been transformed to a page warehouse like database for future analysis by indexer. Automatic Indexing is an indexing that is propelled in automation; this is the main methodology implemented to catalog internet text-based documents [2], [5], [7]. This is made up of the arrangement of the documents gathering based on content. The indexer controls the businesses of each document in other to make retrieval of web documents by topics possible. The technology, make the use of unvarying terms employed to allow effective index and retrieval of research materials

through document coding and eradicating unwanted and unimportant materials.

2.3 FIS Generation of the Hybrid Intelligent System

Fuzzy Inference System membership function generation is targeted towards three major areas to include: Generation of range of input membership function, generation of output membership function and the projected fuzzy output that is expected to be optimized by the ANFIS system.

2.3.1 Generation of FIS Input Membership Function

The FIS input variables used for modelling the system are: Term Weighting (t_w) represented as α_1, Lexical Density (L_d) represented as α_2, Document Similarity (d_{sim}) represented as α_3, and Word Ratio (w_r) represented as α_4 respectively. In this section, each of these input variables shall be analyzed for the purpose of clarity.

2.3.2 Generation of the FIS Input Membership Function for Term Weighting

In fact, the advantages or degree of importance of individual word in the document could be calculated. Weight Wi of word i could be computedthrough the oldtf.idf scheme [6]. In this paper the pattern oftf.isf (term frequency, Inverse sentence frequency) was adopted:

$$W_i = tf_i x isf_i = tf_i x \log_{ni} N, \quad . \quad . \quad . \quad .(5)$$

Wheretfi represent the term frequency of word i in the web document, N represent the total amount of sentence and ni represents the amount of sentencewhich the word i appears. ApplyingEqu.(5), the term weight score for the sentencesis calculated thus:

$$f_5 = \frac{\sum_{i=1}^{k} W_i(S)}{\mathrm{Max}(\sum_{i=1}^{K} W_I(S))}, \quad . \quad . \quad . \quad .(6)$$

In this wise, Wi (S) represents the term weight of word i in sentence S and k is the summation of words in the sentence S. Term Weighting is given the variables Very Strong Frequency (VSF), Strong Frequency (SF), Low Frequency (LF), Weak Frequency (WF), and Very Weak Frequency (VWF) respectively. The linguistics variable has the universe of Discourse between 0 to 1:

$$\alpha_1: \begin{cases} 1 \\ 0 \end{cases}$$

2.3.3 Generation of the FIS Input Membership Function for Lexical Density

The computation of the percentage of lexemes in the documents is referred to Lexical Density; it is however, the percentage computation of the level of enlightenment a text could be[9], [10]Example, oral scripts have a tendency of having lower lexical density (around 45%) compared to printed text (above 50%) [5], [7], [8]. Lexemes' occurrence level, function words' frequencies and lexical density. Therefore, Lexical density describes the proportion of content words in the document, like, nouns, verbs, adjectives, adverbs. Documents are measured `dense` if it is made up of more content phrases compared to the total number of words, i.e. lexical and functional words. It can be computed as follows:

$$L_d = \frac{N_{lexical\ tokens}}{N_{tokens}} \times 100 \quad . \quad .(7)$$

In Language construct, lexical density is made up of the estimation of lexemes in each grammatical and contents unit of words in total. In discourse analysis, lexical density is used to proffer an expressive constraint that could vary from register and genre. Oral texts tend to have a lower lexical density than printed ones, for example.

Lexical density can be computed as:

$$L_d = \left(\frac{N_{lex}}{N}\right) x\ 100 \quad . \quad . \quad . \quad .(8)$$

As such;

L_d= *represents lexical density* analyzed text's

N_{lex} = represents number of nouns, adjectives, verbs, and adverbsin the case document.

N = represents the total number of words in the case document.
Lexical density is assigned the variables Very High Density (VHD), High Density (HD), Slightly Density (SD), Low Density (LD), Very Low Density (VLD). The linguistics variable has the universe of Discourse (This variable symbols used here is by no meanconservative; rather, areindiscriminatelytaken as nonce by way of illustrating the illustration in demand.)

The range is between 0 to 100%:

$$s\alpha_2: \begin{cases} 100 \\ 0 \end{cases}$$

2.3.4 Generation of the FIS Input Membership Function for Document Similarity

Documents are bags of word or collections of words. The similarities of documents are*computational*assessments of closeness of one document compared to the other in term of perspective or content. This paper, presents a vector space analysis. It involves an arithmetical construct that is made up of a collection of elements called vectors that can be added together and reproduced "scaled" by numbers of documents, known as scalars in this context. This means, a vector *v* that *is*stated as summation of elements as,

$$V = A_1V_{i1}+A_2V_{i2}+\ldots+A_NV_{IN} \quad . \quad . \quad . \quad (9)$$

This means$A_K$can bereferred to as scalars or weights and v_{in}provided as the components or elements. In other for easy exploration, a set of documents can be factored as vectors in a common vector space. V(d) that represented the vector derived from document $\boldsymbol{d_I}$ with a single component for every dictionary word.

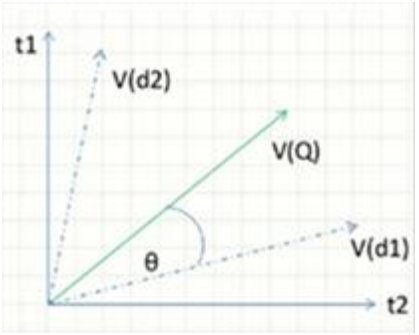

Fig. 4: Documents Similarity Vector Graph

Documents in this pool could be regarded as a collection of vectors in vector space, such that there exist a single axis for each word.

Documents similarity vector is assigned with linguistics variables Very Similarly (VS), Similar (S), Moderately Similar (MS), Slightly Similarly (SS), Not Similar (NS). The linguistics variable has the universe of Discourse between 0 to 4:

2.3.5 Generation of the FIS Input Membership Function for Word Ratio

The quantifiable kith and kin existing among two amounts presenting the number of times one value contain or are contained within the other. Word ratio in this wise is a measure of the interactive number of repetitions of each content word present in the retrieved categories of words; and is considered with the use of mathematical model below:

$$F_n(j) = \frac{F(j)}{\sum_{i=1}^{n} F(i)} \text{x } 100 \quad . \quad . \quad . \quad .(10)$$

A

such;

$F_{(j)}$represents the level of occurrence of the Jth keyword,

$F_{n(j)}$ represents normalization frequency, i.e. probability, of occurrence.

N represents the entire number of retreived keywords.

Word ratio is assigned with linguistics variables Very Close (VC), Close (C), Slightly Close (SC), Not Close (NC), Not Related (NR). The linguistics variable has the universe of Discourse between 0 to 1:

$$\alpha_4 : \begin{cases} 1 \\ 0 \end{cases}$$

2.3.6 Generation of FIS Output Membership Function

The output membership function ranges from not likely, less likely, moderately likely, more likely and most likely. The ranges are elaborated in table 1 below:

Table 1: Table Showing the Range of Output Membership Functions

Value	Membership
0	Not Likely
1	Less Likely
2	Moderately Likely
3	More Likely
4	Most Likely

3. PROJECTED FUZZY OUTPUT TO BE OPTIMIZED BY THE ANFIS

The expected output result is term the search-space. That is the probability of tracking the required file(s) that is most likely relevant to the user's need at any point in time. Search Space has to be 1 for every retrieval stage.

3.1 Fuzzification Procedure

The fuzzification of NFDTRS into linguistics terms is shown in the equation below:

$$\text{Var(x)} = \begin{cases} NotLikely \\ LessLikely \\ ModeratelyLikely \\ MoreLikely \\ MostLikely \end{cases}$$

However, the terms not likely, less likely, moderately likely, more likely, and most likely respectively are used to determine the likelihood of the file to be tracked.

3.2 Input Parameters and Fuzzy Rules

In this paper, fuzzy input and output variables are used in the rule construction and inference to direct the fuzzy inference mechanism. Takagi Sugeno inference mechanism was adopted in other to articulate the rules based on the Takagi (2012), If x is A and y is B, then

$$f = px + qy + r \quad . \quad . \quad . \quad (11)$$

At this point, x and y are considered the input variables; A and B are fuzzy sets of the input variables; f is the output variables; p, q, and r, are considered the consequent parameters.

$$\text{Weight(x)} \begin{cases} 0.2 & \text{If x is not likely} \\ 0.4 & \text{If x is less likely} \\ 0.6 & \text{If x is moderatley likely} \\ 0.8 & \text{If x is more likely} \\ 1.0 & \text{If x is most likely} \end{cases}$$

However, the output variable at this point is calculated through the search of the variable value of the center of gravity of the membership function.

4. FUZZY LOGIC MODEL

Triangular affiliation function based architecture characterizes all crisp values into degrees of memberships within the domain of discourse. Figure 5 presents the fuzzy logic model based conceptual architecture.

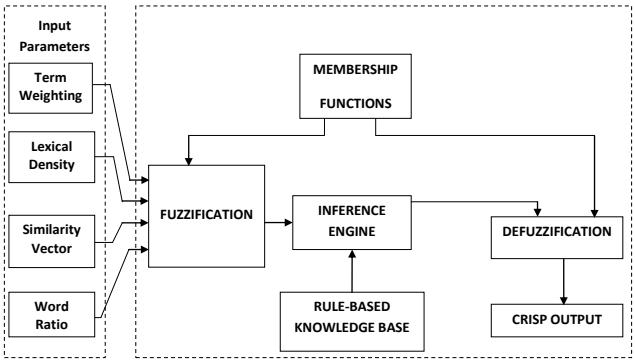

Fig. 5: Conceptual Architecture of Fuzzy Logic Model

4.1 The Fuzzy Logic Model Components

The model is made up of the following components which constitute the fuzzy logic model presented in this paper;

1. **Inference Engine:**this module create a new fuzzy set by assessing the rules of rule base alongside the fuzzy set from fuzzification [8], [10].

2. **Fuzzification Module:**Fuzzification links the crisp inputs to the type-1 fuzzy set making use of the affiliation functions [10].
3. **Defuzzification Module:**It links inference engine's fuzzy set to the crisp output utilizing defuzzification strategy of the center of gravity [10].
4. **Membership Function:** Membership function is often an equation in mathematics which maps crisp inputs to arelationship between 0 and 1, known as a fuzzy set [10].
5. **Knowledge Base:** Knowledge b**ase is** typically a database or rules produced from experts' information to be utilized by an inference engine [6], [8], [9], [10].

4.2 Fuzzy Logic Fuzzification Process

At this point, in every input and output variable selected, fouraffiliation functions (MF) was descrined, namely – Term Weighting, Lexical Density, Similarity Vector, and Word Ratio. A category is defined for each of the variable. These categories are called fuzzy term such as low frequency, weak frequency, very weak frequency etc. the triangular membership function was employed. For this reason, it required at least three points (a, b, c) to define one Membership Function (MF) of a variable. The triangular membership function is defined as;

$$f(x;a,b,c)=\begin{cases}0, & x\le a\\ \dfrac{x-a}{b-a}, & a\le x\le b\\ \dfrac{c-x}{c-b}, & b\le x\le c\\ 0, & c\le x\end{cases} \quad . \quad (12)$$

Where:
a - the left leg of the membership function
b–represents the center of function
c - the right leg of the function
x - the crisp input
f - a mapping function

i. Universe of discourse

The Universe of Discourse is the range of all potential values of the fuzzy linguistic variables. The following universe of discourse is defined for the linguistic variables;

The definition of membership functions are:

Table 2: Fuzzy Universe of Discourse

Linguistic Variable	Lower Bound	Upper Bound	Variable Type
Term Weighting	0	1	Input
Lexical Density	0	100	Input
Similarity Vector	0	5	Input
Word Ratio	0	1	Input
Acceptance Probability	0	1	Output

a. Membership function for term weighting

$$\mu_{vwf}(x;0,0,0.06)=\begin{cases}0, & x\le 0\\ \dfrac{x-0}{0-0}, & 0\le x\le 0\\ \dfrac{0.06-x}{0.06-0}, & 0\le x\le 0.06\\ 0, & 0.06\le x\end{cases}$$

$$\mu_{wf}(x;0.0263,0.08947,0.157)=\begin{cases}0, & x\le 0.0263\\ \dfrac{x-0.0263}{0.08947-0.0263}, & 0.0263\le x\le 0.08947\\ \dfrac{0.157-x}{0.157-0.08947}, & 0.08947\le x\le 0.157\\ 0, & 0.157\le x\end{cases}$$

b. Membership function for lexical density

$$\mu_{vld}(x;0,11.47,23.9)=\begin{cases}0, & x\le 0\\ \dfrac{x-0}{11.47-0}, & 0\le x\le 11.47\\ \dfrac{23.9-x}{23.9-11.47}, & 11.47\le x\le 23.9\\ 0, & 23.9\le x\end{cases}$$

c. Membership function for similarity vector

$$\mu_{ns}(x;0,0.59,1.226)=\begin{cases}0, & x\le 0\\ \dfrac{x-0}{0.59-0}, & 0\le x\le 0.59\\ \dfrac{1.226-x}{1.226-0.59}, & 0.59\le x\le 1.226\\ 0, & 1.226\le x\end{cases}$$

d. Membership function for word ratio

$\mu_{nr}(x; 0,0,0.05789)$

$$= \begin{cases} 0, & x \le 0 \\ \dfrac{x-0}{0-0}, & 0 \le x \le 0 \\ \dfrac{0.05789-x}{0.05789-0}, & 0 \le x \le 0.05789 \\ 0, & 0.05789 \le x \end{cases}$$

e. Membership function for acceptance probability

$\mu_{li}(x; 0,0,0.25)$

$$= \begin{cases} 0, & x \le 0 \\ \dfrac{x-0}{0-0}, & 0 \le x \le 0 \\ \dfrac{0.25-x}{0.25-0}, & 0 \le x \le 0.25 \\ 0, & 0.25 \le x \end{cases}$$

ii. Membership function plot

Fuzzy logic toolbox in Matlab 7.5.0 is used to plot the membership functions used in this work;

iii. Membership function plot

Fuzzy logic toolbox in Matlab 7.5.0 is used to plot the membership functions used in this work;

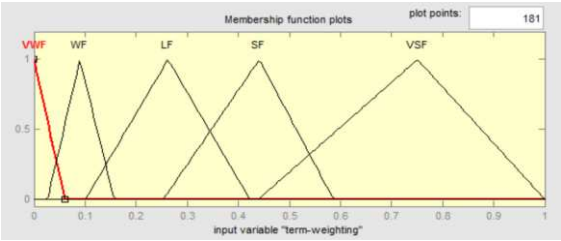

Fig.6: Membership Function for Term Weighting

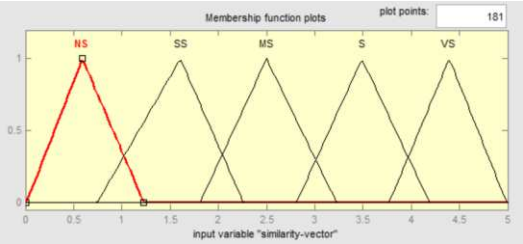

Fig. 7: Membership Function for Lexical Density

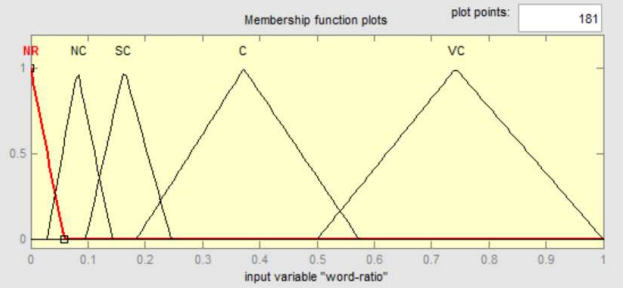

Fig.9: Membership Function for Word Ratio

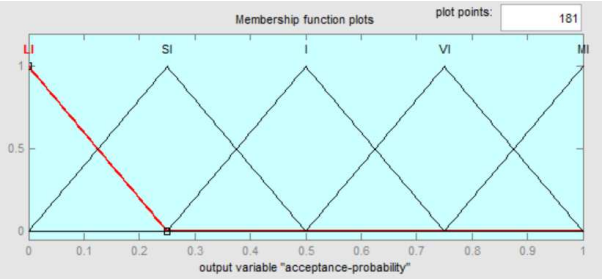

Fig. 10: Membership Function for Acceptance Probability

4.3 Rule Base of the Fuzzy Set

The following conditional statement defines the fuzzy rule

$$R^l: IF\ x_1\ is\ \tilde{F}_1^l\ and \dots\ x_p\ is\ \tilde{F}_p^l\ THEN\ y\ is\ \tilde{G}_1^l$$

While:

L = 1, … M, is the rule base number

R = present rule

p = number of variables

x_p = the variable of p

$\tilde{F}_p^l$= rule l of p

$\tilde{G}_1^l$=rule l output

Membership matrix

This shows the degree of membership at various levels of the crisp inputs. The membership matrix is computed by substituting the different crisp input into the triangular membership function. The membership matrix for this work is generated from membership function evaluator software presented in Figure 11;

Fig. 11: Membership Function Evaluator

1. Membership Matrix for Term Weighting

Table 3: Membership Matrix for Term Weighting

FUZZY SET	CRISP INPUT							
	0.1	**0.2**	**0.4**	**0.5**	**0.6**	**0.7**	**0.89**	**0.9**
WVF	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.0
WF	0.872	0.00	0.00	0.00	0.00	0.00	0.00	0.0
LF	0.00	0.618	0.140	0.00	0.00	0.00	0.00	0.0
SF	0.00	0.00	0.882	0.518	0.00	0.00	0.00	0.0
VSF	0.00	0.00	0.00	0.217	0.574	0.93	0.39	0.3

2. Membership Matrix for Lexical Density

Table 4: Membership Matrix for Lexical Density

FUZZY SET	CRISP INPUT							
	10	**20**	**30**	**40**	**60**	**70**	**80**	**90**
VLD	0.837	0.326	0.00	0.00	0.00	0.00	0.00	0.00
LD	0.00	0.361	0.994	0.348	0.00	0.00	0.00	0.00
MD	0.00	0.00	0.00	0.262	0.359	0.00	0.00	0.00
HD	0.00	0.00	0.00	0.00	0.230	0.996	0.238	0.00
VHD	0.00	0.00	0.00	0.00	0.00	0.00	0.340	0.830

3. Membership Matrix for Document Similarity Vector

Table 5: Membership Matrix for Document Similarity Vector

FUZZY SET	CRISP INPUT							
	1	**2**	**2.5**	**3**	**3.5**	**4**	**4.5**	**5**
NS	0.369	0.00	0.00	0.00	0.00	0.00	0.00	0.00
SS	0.340	0.343	0.00	0.00	0.00	0.00	0.00	0.00
MS	0.00	0.268	0.975	0.319	0.00	0.00	0.00	0.00
S	0.00	0.00	0.00	0.275	1.00	0.275	0.00	0.00
VS	0.00	0.00	0.00	0.00	0.00	0.374	0.813	0.00

4. Membership Matrix for Word Ratio

Table 6: Membership matrix for word ratio

FUZZY SET	CRISP INPUT							
	0.12	**0.21**	**0.32**	**0.42**	**0.5**	**0.61**	**0.78**	**0.98**
NR	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
NC	0.38	0.00	0.00	0.00	0.00	0.00	0.00	0.00
SC	0.32	0.472	0.00	0.00	0.00	0.00	0.00	0.00
C	0.00	0.132	0.69	0.790	0.380	0.00	0.00	0.00
VC	0.00	0.00	0.00	0.00	0.00	0.44	0.880	0.080

4.4 Fuzzy Inference Mechanism

The inference engine evaluates the rules against a fuzzy set. In this work, a Mamdani type inference mechanism is used. Given the crisp input vector v = [0.4, 40, 2.5, 0.32], their degree of membership calculated from respective triangular membership functions are given as;

4.4.1 Fuzzy Value

Table 7: Fuzzy Inference Mechanism

LINGUISTIC VARIABLE			
Term Weighting (0.4)	**Lexical Density (40)**	**Similarity Vector (2.5)**	**Word Ratio (0.32)**
$\mu_{VWF} = 0.0$	$\mu_{VLD} = 0.0$	$\mu_{NS} = 0.0$	$\mu_{NR} = 0.0$
$\mu_{WF} = 0.0$	$\mu_{LD} = 0.348$	$\mu_{SS} = 0.0$	$\mu_{NC} = 0.0$
$\mu_{LF} = 0.0$	$\mu_{MD} = 0.262$	$\mu_{MS} = 0.975$	$\mu_{SC} = 0.0$
$\mu_{SF} = 0.518$	$\mu_{HD} = 0.0$	$\mu_{S} = 0.0$	$\mu_{C} = 0.697$
$\mu_{VSF} = 0.217$	$\mu_{VHD} = 0.0$	$\mu_{VS} = 0.0$	$\mu_{VC} = 0.0$

4.4.2 Defuzzification

At this point, the fuzzy set was defuzzified by using the center of gravity defuzzification method presented in the equation below;

$$COG = \frac{\sum_x^b \mu_A(x)x}{\sum_x^b \mu_A(x)}, \quad . \quad . \quad . \quad (13)$$

At this point $\mu_A(x)$ represents the gradation of membership of x in a set A.

4.5 Theoretical Framework of Hypothesis of the Fuzzy Level

The Frame is considered based on Term Weighting, Lexical Density, Word Ratio as well as similarity Vector as input components and Acceptance Probability as the output component. The framework is as shown in figure 11.

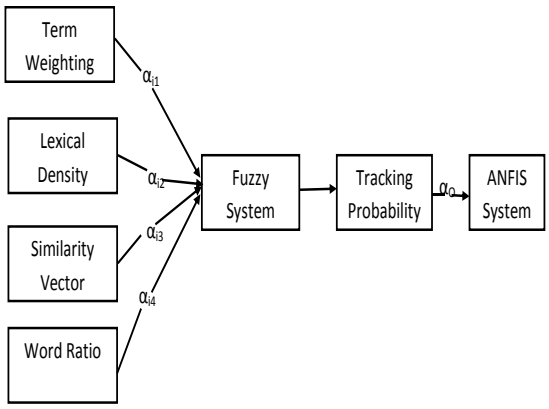

Fig. 11: Theoretical Framework of Fuzzy Model Feeding the Output (αO) into ANFIS for Optimization.

5. IMPLEMENTATION OF ANFIS MODEL

5.1 ANFIS System

The ANFIS model is a hybrid of Neural Network and Fuzzy logic. The ANFIS model for sugeno-type fuzzy inference system makes use ofcross learning algorithm to classify the factors of sugeno-type fuzzy inference mechanism. This means that there exist someamalgamation of least-square method and the back-propagation gradient descent algorithm for training the fuzzy Inference System's membership function parameters to match a trained data set. This ANFIS model as implemented in this paper is based on a standard five (5) layer structure which comprises of the Membership Function Layer, Product Layer, Normalization Layer, and the Consequent Layer as well as summation layer. The structure of the ANFIS model applied in this technology is presented in Figure 12:

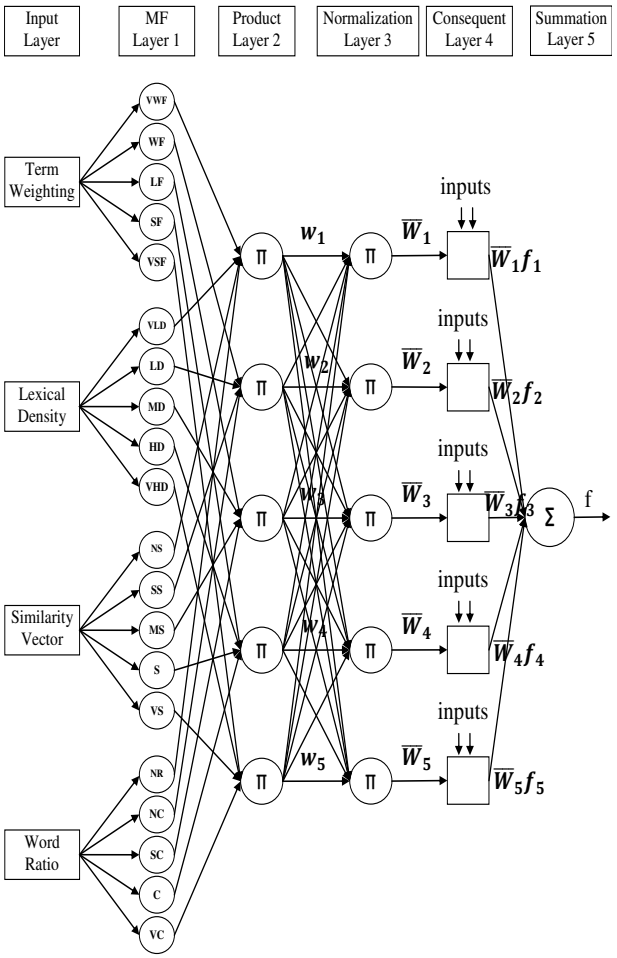

Fig. 12: The ANFIS Model

5.2 Decryption of the ANFIS Model

The ANFIS model used in this work is made of five (5) layers. These layers are described below;

1. **Input layer:** Every nodes of this layer holds the input to the system. In the context of this system, the inputs are – Term Weighting, Lexical Density, Similarity Vector, and Word Ratio.
2. **Layer 1 (MF layer):** This layer is called the membership function layer. Each node I in this layer, is a parameterized triangular membership function (i.e. VWF, WF, LF, SF, VSF, VLD, LD etc.). The output of this layer is computed as;

$$O_{1,i} = \mu_{TermWeig\ hting\ _{i-5}}, (a) for\ i = 1, \dots, 5$$
$$O_{1,i} = \mu_{LexicalDensity\ _{i-10}}, (a) for\ i = 6, \dots, 10$$
$$O_{1,i} = \mu_{SimilarityVector\ _{i-15}}, (a) for\ i = 11, \dots, 15$$
$$O_{1,i} = \mu_{WordRatio\ _{i-20}}, (a) for\ i = 15, \dots, 20$$

   a. Where:
   b. $O_{1,i}$ - is the output of layer 1 of term i
   c. $\mu_{TermWeig\ hting_i}(a)$ - is the grade of affiliation of a in the membership function $TermWeighting_i$
   d. $TermWeighting_i$ - signifies the directory of affiliation function I belonging to a variable's $TermWeighting$.
   e. $a, b, c, and\ d$ - are the inputs.
   f. For instance - $TermWeighting_i$for $i = 1$, points to VWF.
3. **Layer 2 (product layer):** The nodes of the layer computes the firing strength of a rule using equation below;
   a. $O_{2,i} = w_i = \mu_{TermWeig\ hting_i}(a) * \mu_{LexicalDensity\ _i}(b) * \mu_{SimilarityVector\ _i}(c) * \mu_{WordRatio\ _i}(d) for i = 1, \dots, 4 \quad . \quad . \quad . \quad . (14)$
   b. The auto-generated rules are m^n,as m represents the total number of membership functions in every inputs and n represents the number of inputs. Here, the total number of rule generated is $5^4 = 625$ rules.
4. **Layer 3 (normalization Layer):**Every node of thecurrents layer, normalized the firing strength of the rule by computing the ratio of the ith rule's firing strength in relation to sum of all rules firing strength using equation as given;
   a. $O_{3,i} = \overline{w}_i = \frac{w_i}{w_1 + w_2 + w_3 + w_4}, \quad i = 1, \dots, 4 \quad . \quad . \quad . (15)$
   b. Where:
   c. w_1 −Fired strength of first rule
   d. w_2 −Fired strength of second rule
   e. w_3 −Fired strength of third rule
   f. w_4 −Fired strength of fourth rule
   g. w_i −ith rule's fired strength
   h. $\overline{w}_i$ −represents the normalized fired strength of ith rule
5. **Layer 4 (consequent Layer):** The nodes of this layer represents the consequent part of a fuzzy rule with node function;
   a. $f_i = p_i a + q_i b + r_i c + s_i d + t_i \quad (16)$
   b. $O_{4,i} = \overline{w}_i f_i = \overline{w}_i (p_i a + q_i b + r_i c + s_i d + t_i), \ i = 1,,4$
   c. Hence:
   d. $\overline{w}_i$ −represents the regularized fired strength of ith rule?
   e. $\{p_i a + q_i b + r_i c + s_i d + t_i\}$ −representsfirst order polynomial of ith rule's consequent part. This parameters $\{p_i + q_i + r_i + s_i + t_i\}$ are identified during the training process of ANFIS.
6. **Layer 5 (summation layer):**Layer 5 ensures the computation of the sum of outputs of all rules from former layer.

$$O_{5,i} = \sum_{i=1}^{4} \overline{w}_i f_i = \frac{\sum_{i=1}^{4} \overline{w}_i f_i}{w_1 + w_2} \quad .(17)$$

5.3 ANFISLearning Parameter Configuration

ANFIS learns by identifying adaptable parameters of the membership function (a, b, c) for left leg, enter and right leg of the triangular membership function as well as $\{p_i + q_i + r_i + s_i + t_i\}$so as to ensure the minimization of the error between definite and predictable output. This paper ascertains thatregular two pass learning procedure of ANFIS

wasused. This learning process is based on a hybrid of gradient descent (GD) and least square estimators (LSE).

6. OPERATIONAL DATA COLLECTION,E- PROCESSING AND ANALYSIS

Eight Hundred and Ninety-One (891) rows of weighted extracted data

collected fromText Retrieval Conference (TREC) of 2011. TREC is one of the reliable and valid source of data collection for standard test data often applied in obtaining information for IR. In TREC bulky test pools of documents together with their importancechronicles to very hugecollection of subjects or materialrequired are available for contender system. First TRECs consists of 50 topics with evaluated of significant records against alternative sub-set of documents that could have 100,000 diversedocuments in individual subset. Sequel to this, There exist several test pools that are assembled with the same arrangement as TREC collections. CLEF 2009 INFILE collection is an example of such collectionswhich are used in the IR-FIS experimentations. The data contains four (4) variables arranged in the following order: Var1 represent Term Weighting; Var2 represent Lexical Density; Var3 represent Similarity Vector; and Var4 represent Word Ratio respectively. The data can be collected in advanced through the steps below:

1. Data collection
2. Fuzzy Inference System
3. ANFIS Implementation

The datasets were divided into three (3) parts. The first part was used to form the training data, the remaining part joined together were used as checking or testing data for the system.The large percentage of the accuracy of any method performed on any data lies within the data itself. If data overlaps, or data has other undesirable qualities, these degrades the performance of algorithms that utilizes the data. Row(s) as training data: Row 1 to 300 was used to form training data. Row(s) as checking Data: Row 301 to 500 and row 501 to 600 were combined to form checking data. Row(s) as Testing Data: Row 148 to 295 and row 500 to 709 were combined to form testing data.

6.1 ANFIS Generation and Initialization

Figure 13 shows the linear output of FIS parameters. These parameters can be initialized depending on one's inclination. ANFIS can also automatically initialize these parameters. The parameters consist of the following: 5 MFs, 4 inputs, and 1 linear output.

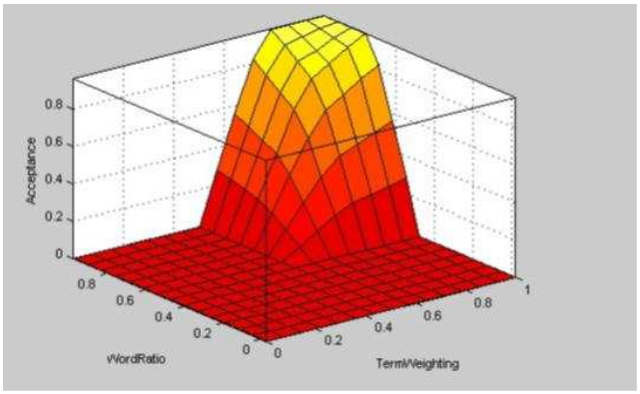

Figure 14: Term Weighting to Word Ratio (Surface Viewer)

Figure 14 shows the ANFIS surface viewer comparing term weighting and word ratio to determine the documents tracking and retrieval acceptability rate. The acceptability rate is at 0.75 to 1.0 if term weighting is 0.75 to 1.0 and word ratio is 0.75 to 1.0; at this level, the document is considered fits for the user's specified requirement and hence retrieved and presented to the user.

Figure 15 shows the ANFIS surface viewer comparing lexical density and word ratio to determine the documents tracking and retrieval acceptability rate. The acceptability rate is at 0.75 to 1.0 if lexical density is 75% to 100% and word ratio is 0.75 to 1.0. At this level, the document is considered fits for the user's specified requirement and hence retrieved and presented to the user.

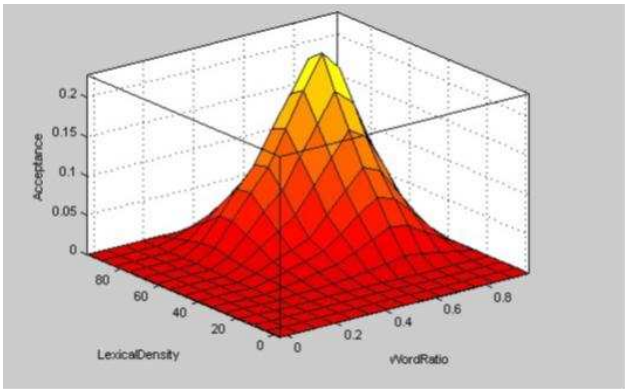

Figure 15: Lexica Density to Word Ratio (Surface Viewer)

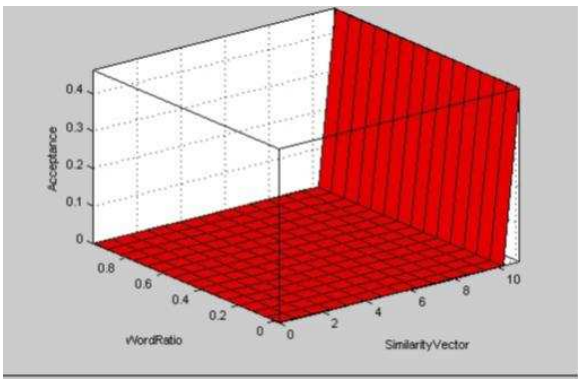

Figure 16: Surface Viewer (WordRatio to Similarity Vector)

Figure 16 shows the ANFIS surface viewer comparing document similarity vector and word ratio to determine the documents tracking and retrieval acceptability rate. The acceptability rate is at 0.75 to 1.0 if document similarity vector is 4.1 to 5.0 and word ratio is 0.75 to 1.0. at this level, the document is considered fits for the user's specified requirement and hence retrieved and presented to the user.

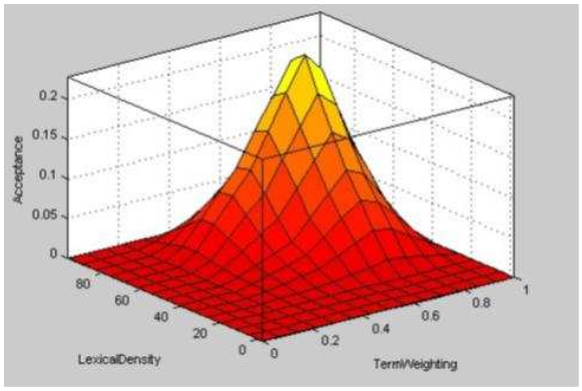

Figure 1: Surface Viewer (Lexical Density to Term Weighting)

Figure 17 shows the ANFIS surface viewer comparing term weighting and document similarity vector to determine the documents tracking and retrieval acceptability rate. The acceptability rate is at 0.75 to 1.0 if term weighting is 0.75 to 1.0 and document similarity vector is 4.1 to 5.0. at this level, the document is considered fits for the user's specified requirement and hence retrieved and presented to the user.

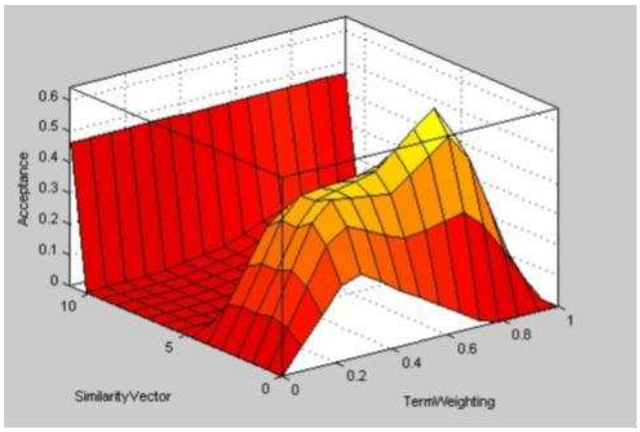

Figure 2: Term Weighting to Similarity Vector (Surface Viewer)

Figure 18 shows the ANFIS surface viewer comparing document similarity vector and lexical density to determine the documents tracking and retrieval acceptability rate. The acceptability rate is at 0.75 to 1.0 if document similarity vector is 4.01 to 5.0 and lexical density is 75% to 100%. at this level, the document is considered fits for the user's specified requirement and hence retrieved and presented to the user.

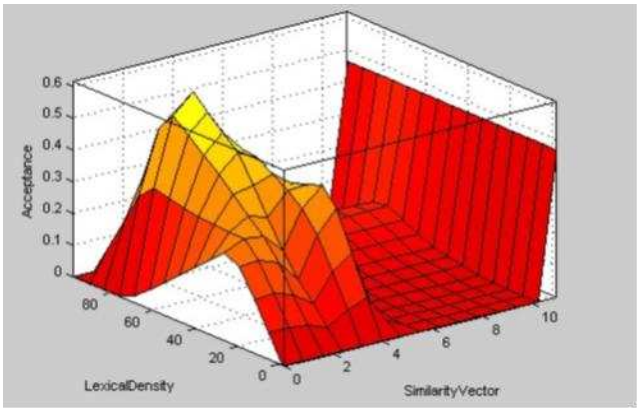

Figure 3: Surface Viewer (Similarity Vector to Lexical Density)

Figure 19 shows the ANFIS surface viewer comparing term weighting and lexical density to determine the documents tracking and retrieval acceptability rate. The acceptability rate is at 0.75 to 1.0 if term weighting is 0.75 to 1.0 and lexical density is 75% to 100%. At this level, the document is considered fits for the user's specified requirement and hence retrieved and presented to the user.

6.2 ANFIS Model Validation

From observations, at epoch 300, the testing error value of 0.19379 is observed between the computed data and the desired output. The observed error value is far greater than the error tolerance of 0.0001 specified in the train FIS. The conceptaheadof this is to check the model validationdata set and thereafter after at agiven point in the training, the model started behaving attributes of over fitting on the training data set. By implication, the checking data set model error inclines to decline as the training is on courseuntil over fitting started, then the checking data set model error swiftlyrises. This over fitting can be accounted for by testing the FIS trained on the training data against the checking data, and chosen the membership function parameter to represents the dataconnectedtoleast checking error suppose the errors shows attributes of over fitting model.

7. SYSTEM REQUIREMENTS

7.1 Hardware Requirements and Justifications

For an initial deployment of the system, only a single machine is required. The recommended minimum system should have 2Ghz Processor at Intel core i3; 2GB RAM, and disk space of at least 300GB required.

7.2 Software Requirements and Justifications

The software tools required for the implementation of the intelligent system are:

1. MySQL database 5.7.14 from WAMP server 3.0.6
2. Java Programming Language
3. MATLAB 2015A
4. Microsoft Windows 10 Ultimate Operating System

Java programming language and NetBeans integrated development environment serve as frontend engine and provide extensive library for the development of NFDTRS-ANFIS application with object-oriented concepts of modularity and reusable codes based on data abstraction, polymorphism, encapsulation and inheritance. Java is platform-independent; is

potable and can run on different systems [1], [4]. It is used to communicate with MATLAB. MySQL relational database language is used in this work as the backend engine because of the rich flexibility in the storage, modification, retrieval and support for very large data set. It features an impressive library of performance enhancement procedure such as query cashing and multi-threading. MySQL has security and control capabilities particularly through the GRANT and REVOKE command repertoire. These security capabilities enable the protection of data and information from unauthorized users. MySQL is poised with inherent capabilities and structure that supports collaboration with PHP (MySQL) environment. It is scalable, easy to used and is well suited for remote application development [8], [9], [10]. Windows 10 OS provide better scheduling of task and processes. It offers much more control over the task and processes without hassles as compared to Windows 7. It also offers amazing workspace since work space helps to minimize the clutter,[1], [2].

8. ANFIS IMPLEMENTATION PROCEDURE FOR NFDTRS

A Neuro-Fuzzy Based Document Tracking and Retrieval System is activated by clicking on its icon on the desktop. This is followed by display of the introductory screen (i.e. the splash screen) and then preceded after some interval to the authentication windows as shown in figure 20 and 21:

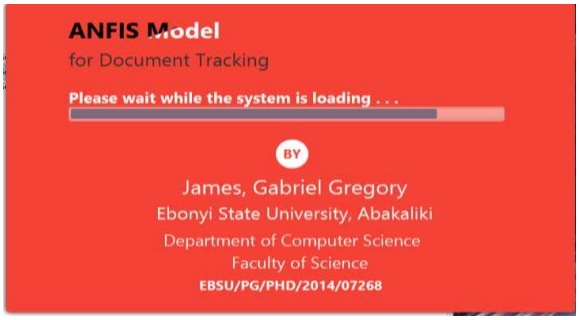

Fig. 20: Welcome Screen

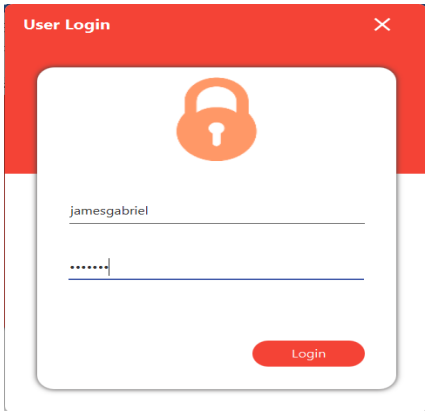

Fig. 21: Login Screen

Successful authentication can be proven when correct user's identities are entered, such that when the "Login" button is clicked, it displays the main menu as depicted in figure 22.

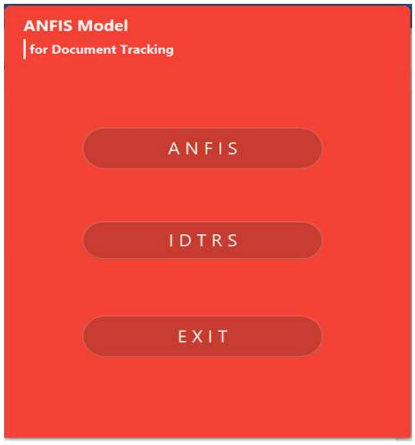

Fig. 22: Main Menu

The main menu consisted of three menus, that is the ANFIS Model, IDTRS and Exit. When the Exit menu is clicked, the application is terminated. The user click on the ANFIS button, the ANFIS Model interface is displayed as depicted in diagram 23:

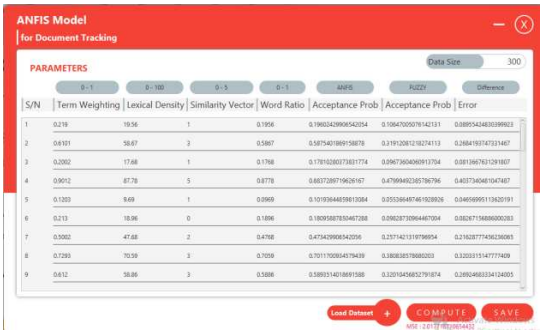

Fig. 23: Data Input Interface

To track the data, click the plus sign "+" it will produce an object that browses the data item file as shown in figure 24, when the file is carefully selected, and click Ok the system will upload the input data into the ANFIS editor then click the ANFIS button to train the data and the ANFIS trained output, the FIS trained output and the error differences will be displayed as in Appendix E:

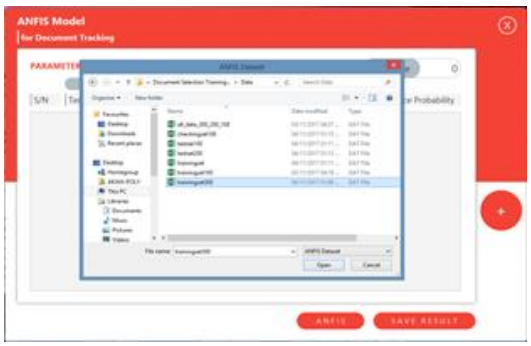

Fig. 24: Snap Shot of the Data Upload into the File Sever

When the ANFIS process is completed and the optimized output is produced then click the "SAVE RESULT" button to save the output and the object as depicted in figure 25 will be display for the file name to be provided and the output is save with pdf file with .pdf file extension.

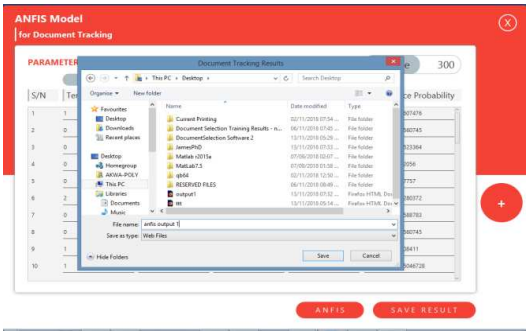

Fig. 25: Snap Shot of the ANFIS output File Sever

If the user clicks on the IDTRS (Intelligent document tracking and retrieval system) menu, the ANFIS documents retrieval platform appears as depicted in figure 26 below:

Fig. 26: Intelligent Document Tracking and Retrieval System

When the user query is typed into the URL, the system will search for the relevant document and retrieved with specifications of the documents parameters as depicted in figure 26 below:

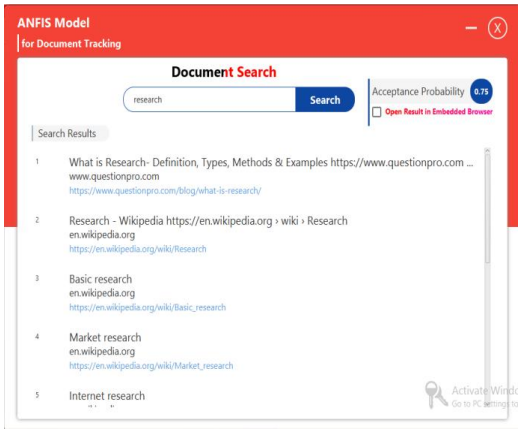

Fig. 26: IDTRS Retrieval Result

If the ***Open Result in Embedded Browser*** option is selected, the system will display the resulting document retrieved in the embedded browser. Otherwise, the document will be displayed in the active browser of your computer.

9. TESTING AND INTEGRATION OF ANFIS-NFDTRS

The system was tested in a window-based environment and the output was analyzed to find the coloration between the various input variables and their corresponding effect on the output variable. The Mathlab FIS platform was used to analyzed the model of [6] Fuzzy based system and the new neuro-fuzzy based system the two outputs were correlate and the differences were noted as shown in Table 8 below:

Table 8: Testing optimized result of the Prototype system

FIS Output	ANFIS Output	Optimize Result
0.176906542	0.429906542	0.066084112
-0.010009346	0.242990654	0.10346729
-0.159542056	0.093457944	0.981971963
0.569429907	0.822429907	-0.466626168
-0.150196262	0.102803738	0.271691589
-0.131504673	0.121495327	0.48664486
0.102140187	0.355140187	0.140850467
-0.010009346	0.242990654	0.346457944
0.083448598	0.336448598	-0.064757009

In the result above, it was observed that the ANFIS output is better than the FIS output by 0.253. This makes neuro-fuzzy method a better technology to implement in Information Retrieval better than other methods.

9.1 Sensitivity Analysis

Sensitivity analysis was performed to determine the level of contributions or degree of significance of inputs to output. In the sensitivity analysis, selection of trials name, the dataset to use in the sensitivity analysis and the best connection weights was carried out. In Matlab and the graph of the sensitivity of the input to the output was displayed as shown in figure 27. The results of the sensitivity test are shown in Table 9.

Table 9: Sensitivity test of the attributes to document Retreival Rate

S/N	Attributes	Code Sensitivity	Score (%)	Group
1	Term Weight	TW0.653155	65.32	A
2	Lexical Density	LD 0.551048	55.10	B
3	Document Similarity Vector	DSV 0.501936	50.19	B
4	Word Ratio	WR 0.307525	30.75	C

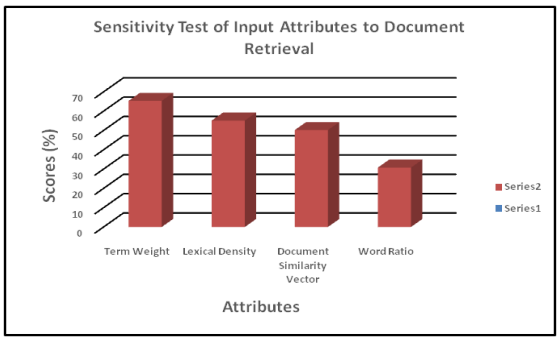

Fig. 27: Sensitivity of Input Variables to Document Retrieval

Table 9 shows the percentage input contributions to output. The contributions were segmented into four (4) groups. Group A were input which scored above 60%, Group B scored above 50% but less than or equals to 60%, Group C scored above 40% and below but less than 50% while inputs with less than 1% score were grouped in F. TW has the highest score of 63.32%, in the determination of retrieval possibility of relevant document from the web. LD and DSV have a score of 55.10% and 50.19% respectively in the sensitivity analysis. The variable WR scored 30.75% which contributed 30.75% to the determination of output. No variable had below 1%. Four (4) scorers from Group A, B and C were TW, LD, DSV and WR respectively.

10. SUMMARY

This paradigm was developed in three (3) phases which is made up offuzzifications, inference systems, and defuzzifications. The triangular membership function (Tmf) was used to map the input parameters to the output parameters. Takagi Sugeno inference engine was used in planning for ANFIS model. Java programming language was utilized on windows 10 to coordinate the output of ANFIS using Matlab, MySQL, WAMP server 3.0.6, and database 5.7.14. Information from Text Recovery Conference of 2011 was used to validate this work.

When using this approach in training, it was observed that the training, testing, and checking of the KMSI within the hybrid learning process at 150 epochs gave their results as 0.025819, 0.026073, and 0.026347 respectively. The proposed system is faster when compared with the present state of the art and it has a very negligible error of 0.025344 at 300.0 epochs.The average error of 0.0467283 was achieved within the proposed method against the average error of 0.024542 with hybrid learning algorithm at 300.0 epochs. In this manner, the ANFIS assessment of the hybrid learning algorithm performed superior to FIS approach.

11. CONCLUSION

This work was done through the introduction of ANFIS variables and values, implementation of more fuzzy rules and using the Triangular-shape membership function for the term weight, lexical density, documents similarity vector and the word ratio.In conclusion:

1. The existing search engine was adequately analyzed and their weaknesses established.
2. A Neuro-Fuzzy based paradigm was developed that enhances tracking and retrieval of web document.
3. The simulation the model was carried out using MatLab 2013 with emphases on the four indices for the file retrieval as the basic input parameters.

12. REFERENCES

[1] Iwok, S O (2018). A Model of Intelligent Packet Switching in Wireless Communication Networks. PhD Thesis, Department of Computer Science, Ebonyi State University Abakaliki.

[2] Udoh, Samuel Sunday (2016) Adaptive Neuro-Fuzzy Discrete event System Specification for Monitoring Petrol Product Pipeline. PhD Dessertation of the Department of Computer Science, Federal University of Akure.

[3] Yuanyam, C., Limin J., Zundong Z., (2009) Mamdani Model Based Adaptive Neural Fuzzy Inference System and its Application.International Journal of Information and Mathematical Sciences 5(1), 2229-2235.

[4] Chu, H., (2003). Information representation and retrieval in the digital age, American Society for Information Science and Technology, ISBN 1-57387-172-9, Vol. 9, Pp. 111-112, 2003.

[5] Chen, H., Shank, G., Iyer, A., & She, L., (1998). A machine learning approach to inductive query by examples: An experiment using relevance feedback, ID3, genetic algorithms, and simulated annealing", Journal of the American Society for Information Science, Vol. 49, Pp. 693-705, 1998.

[6] Lea, J. and Keem, B. K. (1992), "Fuzzy models for pattern recognition: methods that search for structures in data", IEEE Press, New York, NY, 1992.

[7] Hearst, M. A. (1999), "The use of categories and clusters for organizing retrieval results," Strzalkowski, T. (ed.), Natural Language Information Retrieval, Kluwer Academic Publishers, Netherlands, pp.333-374.

[8] Anagnostopoulos, C. Anagnostopoulos, V. Lumos, E. Kayafas, (2004).Classifying web pages employing a probabilistic neural network", IEEE Proceedings on Software 151 (3), pp. 139-150, 2004.

[9] T. Kohonen, (1995). Self-Organizing Maps(Springer Verlag, Berlin, 1995).Kohononen invented the clustering approach known as self-organizing feature maps, inspired by the retinatopic, tonotopic, and somatotopic maps found in the brain.

[10] Zadeh, L.A.(1998). Fuzzy Logic, Computer, Vol. 1, No. 4, Pp. 83-93, 1988.