International Journal of Engineering Applied Sciences and Technology, 2020
Vol. 4, Issue 10, ISSN No. 2455-2143, Pages 65-74
Published Online February 2020 in IJEAST (http://www.ijeast.com)

IMPLEMENTATION OF INTELLIGENT DOCUMENT RETREIVAL MODEL USING NEURO-FUZZY TECHNOLOGY

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

James, G. G.
Department of Computer Science
Ebonyi State University,
Abakaliki-Nigeria

Onu, F. U.
Department of Computer Science
Ebonyi State University,
Abakaliki-Nigeria

Abstract - The field of computer science has gathered more weight since the advent of the internet, not only to the computer scientist and engineers but also to professionals of all categories. This is so because of the remarkable benefits it offers to the body of knowledge. Sharing information has been made easy, even among people from various parts of the world as the World Wide Web becomes a dumping zone for various categories of information. Searching for information on the internet to study state of the art for research purpose to a large extent depends on our ability to track all related topics and classify them into groups of similar topics. As information grows rapidly over the web, it becomes more difficult for researchers to find the information they are looking for. The rapid growth of information on the web and the inadequacy of the conventional search engines to retrieve relevant information based on user's request have motivated this research. This work extends earlier Fuzzy Information Retrieval models by adding more fuzzy linguistic values, fuzzy variables, using different membership functions, using rules that consider the document structure and optimizing the throughput using Nuero-fuzzy system. The document is quantified by describing it using features/linguistic variables that contribute most to its relevance to the query like the Lexical density, term weight, and document similarity vector as well as word ratio. Linguistic values are assigned to each of these variables that associate them with membership degrees. An ANFIS inference engine was built using the seugeno method that handle these variables to measure the degree of document relevance to the query. The essence of applying the neuro-fuzzy techniques is to build an adaptive intelligent information retrieval system which will track web documents into similar topic using an unsupervised machine learning technique to reduce the percentage of irrelevant documents that are retrieved and presented to users. It was found that using Adaptive Neuro-Fuzzy Inference System improved the performance slightly by 0.22641 representing 22.64% over the Fuzzy Inference System(FIS) results thereby guarantee retrieval of most relevant information that meet the user's request.

Keywords: Fuzzy Systems, Neural Network, Hybrid Systems, IR.

I. INTRODUCTION

Searching for information on the internet to study state of the art for research purpose to a large extent depends on our ability to track all related topics and classify them into groups of similar topics. As information grows rapidly over the web, it becomes difficult for researcher to find the information they are looking for. The rapid growth of information on the web and the inadequacy of the conventional search engines to retrieve relevant information based on user's request have motivated this research. Tracking, classification and retrieval of documents required tools that could recognize patterns in data as well as process imprecise information. FL lacks the capability to learn from previous data and NN equally lacks the capability to handle imprecise and incomplete data. This makes Neuro-Fuzzy systems one of the best options for document tracking, as the weakness of Fuzzy Logic and Neural Network are complimented whilst the strength of the individual components are enhanced. This work extends earlier fuzzy IR models by adding more fuzzy linguistic values, fuzzy variables, using different membership functions, using rules that consider the document structure and optimizing the throughput using Nuero-fuzzy system.

The search for information is always a major issue for researchers. Often time, people travel over a distance to track the most needed data for their research work, thereby making the task of research difficult. This has been the tradition until researchers in the field of information technology came up with the techniques of intelligent search engine, which used the web to facilitate the search for information. This paper proposes a hybrid intelligent search system based on neuro-fuzzy paradigm for document tracking and retrieval. Its characteristic feature is a capability to take into account both the imprecision and uncertainty pervading the textual information representation.

Over the years, many intelligent search systems such as fuzzy ontologies systems, Fuzzy Clustering systems etc, has been proposed and developed to solve the problem of tedious search for information via web, in other to overcome the limitations of the existing search engines. This system has added tremendous contributions to the body of knowledge, but has some limitations. Agents systems have become one of the most active and lively research areas in computer science worldwide [3].

International Journal of Engineering Applied Sciences and Technology, 2020
Vol. 4, Issue 10, ISSN No. 2455-2143, Pages 65-74
Published Online February 2020 in IJEAST (http://www.ijeast.com)

According to Walt and Rudiger (2002), an agent is a software program that can perform specific task for a user and purses a degree of intelligence that permits it to perform parts of its tasks autonomously and to interact with its environment in a useful manner. The dissertation proposed a neuro-fuzzy based model for classification of search results based on the strength of words in the query. The neuro-fuzzy clustering technique will be employed to classify the documents into groups of similar topics for specific knowledge. Fuzzy Logic (FL) is a logic that its ultimate goal is to provide foundations for approximate reasoning using imprecise propositions based on fuzzy set theory. It serves mainly as apparatus for fuzzy control, analysis of vagueness in natural language and several other application domains. FL has the capability of handling imprecise, incomplete and vague information, as well as ability to represent partial truth. FL is limited by its inability to learn from previous data.Neural-Network (NN) (or Artificial Neural Network (ANN)) is a collection of interconnected elements or nodes [1]. The nodes are termed simulated neurons as they attempt to imitate the functions of biological neurons. In artificial neural netwoks (ANNs) are considered as family of models inspired by biological neural networks (the central nervous systems of animals, in particular the brain) which are used to estimate or approximate functions that can depend on a large number of inputs and are generally unknown [4,5].

According to Udo, S. (2016), NN is considered as mathematical models of biological nervous systems, having capabilities of fault tolerance, parallelism, learning from training data, recalling memorized information and generalizing to the unseen patterns. NN is limited by the inability to handle imprecise and incomplete data. Similarly, neural network can approximate a function, but it is impossible to interpret the result in terms of natural language. Hence, the need for the fusion of neural networks and fuzzy logic in neuro-fuzzy models provide learning as well as readability.

Neuro-Fuzzy systems are derived from the combination of Neural Network (NN) and Fuzzy Logic (FL) techniques which result in hybrid intelligent system. According to Yuayuan et al (2009), Neuro-Fuzzy hybrid systems offer solutions to real life problems by synergizing the human-like reasoning capabilities of FL with the learning and connectionist structure of NN. In the hybridization, the weakness of Fuzzy Logic and Neural Network are complimented whilst the strength of the individual components are enhanced.The essence of applying the neuro-fuzzy techniques is to build an adaptive intelligent information retrieval system which will cluster internet documents into similar topic using an unsupervised machine learning techniques to reduce the percentage of irrelevant documents that are retrieved and presented to users.

II. RESEARCH DESIGN METHODOLOGY

The method employed for this work is the Object-Oriented Analysis and Design and specifically the waterfall system development method. Object–Oriented Analysis and Design (OOAD) is the procedure of identifying software engineering requirements and developing software specifications in terms of a software system's object model, which comprises of interacting objects. An object-oriented software development methodology is a software engineering process that through the generic development phases of *analysis*, *design*, *implementation* and *test*–based on analyzing existing methodologies and techniques, identifying their strengths and weaknesses, and producing a set of requirements defining the characteristics of the target methodology. The methodology can then be developed through making utmost use of existing techniques in such a way as to satisfy the requirements.
The following steps/techniques shall be used in achieving the aim and objectives of the study:

(i) Review of intelligent search engine, information retrieval systems, Document clustering, documents clustering techniques, document clustering algorithms, neural network (NN), fuzzy logic (FL), and adaptive neuro-fuzzy classification shall be carried out.
(ii) Human experts in the field of fuzzy logic and neural network will be interviewed.
(iii) Several work and journals on intelligent search systems shall be consulted
(iv) The neuro-fuzzy based search model will be design using the neuro-fuzzy platform of Mathlab.
(v) The system developed in (iv) above, shall be tested and the result on electronics library document retrieval shall be evaluated to assess its functionality and adequacy of the proposed system.

III. OPERATIONAL DATA COLLECTION, PRE-PROCESSING AND ANALYSIS

Eight Hundred and Ninety-One (891) rows of weighted retrieved data from Text Retrieval Conference (TREC) of 2011. TREC One source for standard test collections commonly used in obtaining information for IR. In TREC large test collections of documents along with their relevance records to a large set of topics or information needs are made available for competitor systems. Early TRECs consisted of fifty topics with their relevance record evaluated against a variant subset of documents that can have 100,000 different documents in each subset. There are now many test collections that are built with the same format as TREC collections. One such collection is the CLEF 2009 INFILE collection which is used in the IR-FIS experiments. 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 obtained advanced in the following major stages.

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

International Journal of Engineering Applied Sciences and Technology, 2020
Vol. 4, Issue 10, ISSN No. 2455-2143, Pages 65-74
Published Online February 2020 in IJEAST (http://www.ijeast.com)

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.

IV. SYSTEM REQUIREMENTS

a. Hardwar Requirement Justification

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

b. Software Requirements

The following are the software tolls required for the implementation of an NFDTRS using ANFIS:

(i) MySQL database 5.7.14 from WAMP server 3.0.6
(ii) Java Programming Language
(iii) MatLAB 2015A
(iv) Microsoft Windows 10 Ultimate Operating System

V. 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 proceeded after some interval to the authentication windows as shown in figure 1 and 2 below:

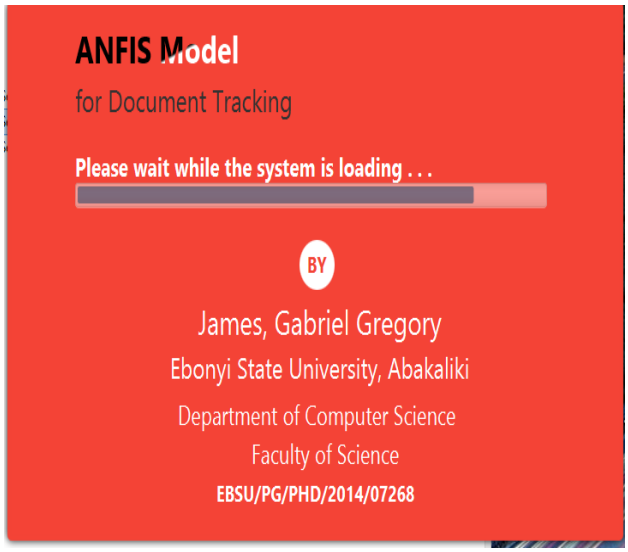

Fig. 1: Welcome Screen

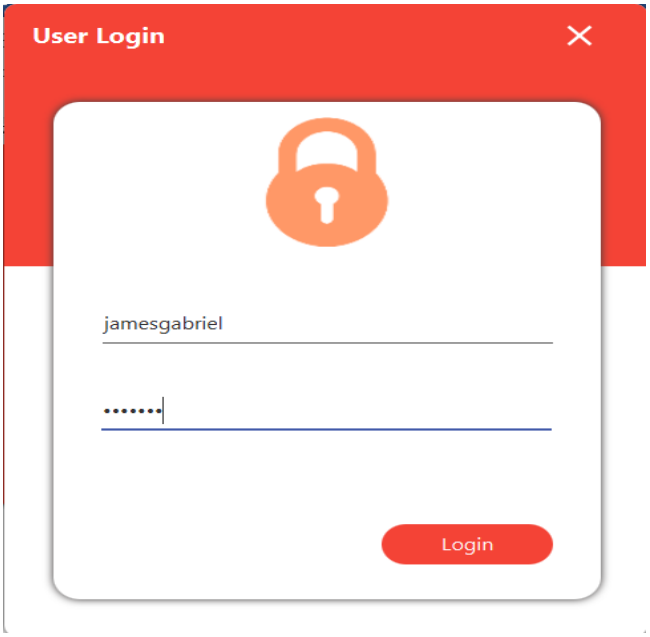

Fig. 2: Login Screen

A successful authentication is established when a correct user name and password are entered, a click on the "Login" button displays the main menu as depicted in figure 3.

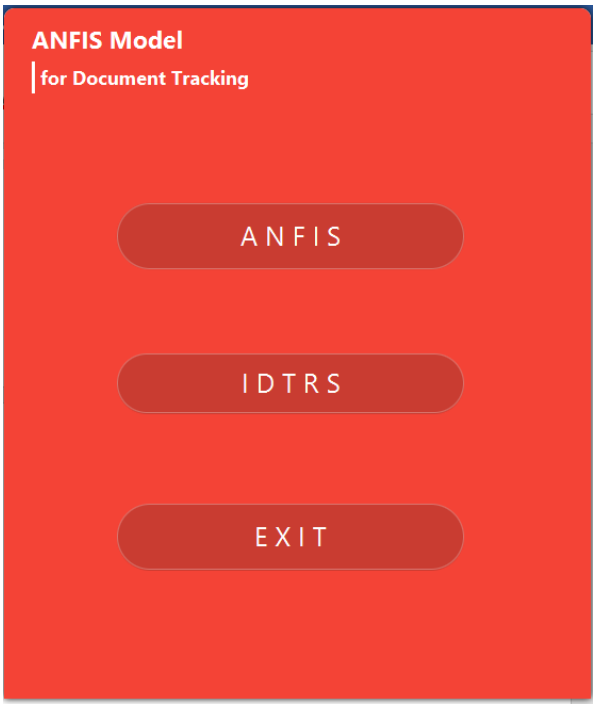

Fig. 3: Data Input Interface

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 4 below:

International Journal of Engineering Applied Sciences and Technology, 2020
Vol. 4, Issue 10, ISSN No. 2455-2143, Pages 65-74
Published Online February 2020 in IJEAST (http://www.ijeast.com)

Fig. 4: Data Input Interface

To track the data, click the plus sign "+" its will produce an object that browses the data item file as shown in figure 5, 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 will be displayed as in Appendix E:

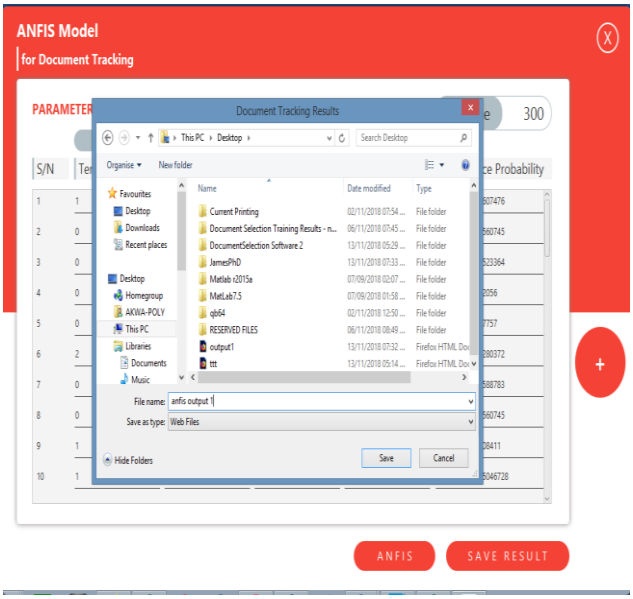

Fig. 5: Snap Shot of the ANFIS output file sever

If the user clicks on the IDTRS (Intelligent document tracking and retrieval system) menu, the ANFIS documents retrieval platform appears as depicted in figure 6 below:

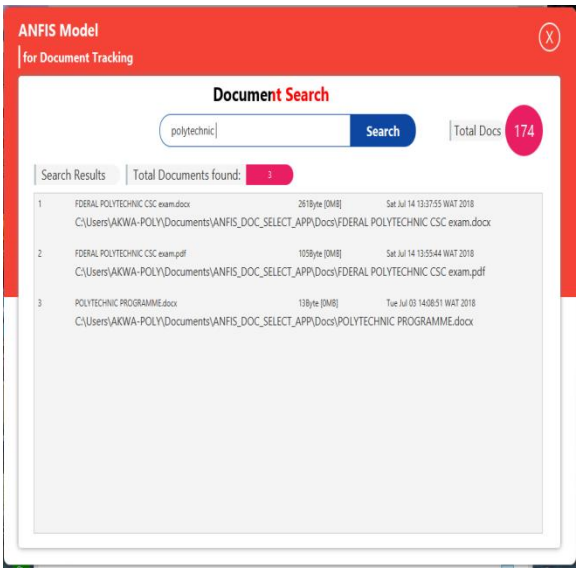

Fig. 6: Intelligent Document Tracking and Retrieval System

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 7 below:

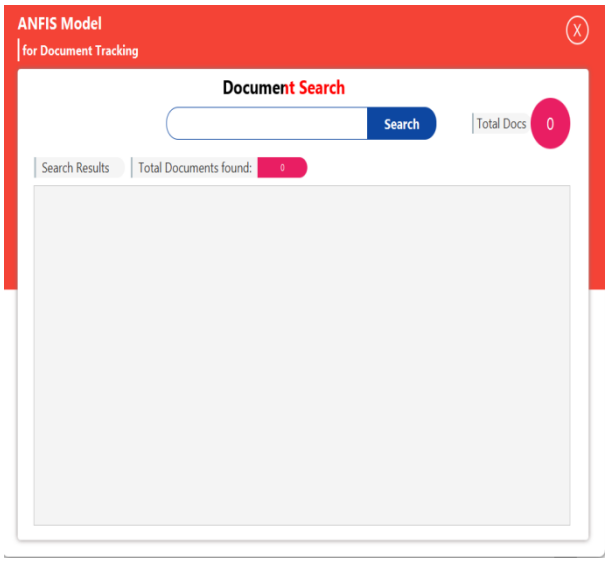

Fig. 7: IDTRS Retrieval Result

VI. TESTING AND INTEGRATION OF ANFIS NFDTRS

The system was tested in a window-based environment and the output was the output was analysed to find the coloration between the various input variables and their corresponding effect on the output variable. The Mathlab FIS platform was used to analized the model of Lea and Bee, 2012 Fuzzy based system and the new neuro-fuzzy based system the two outputs were correlate and the differences were noted as shown in Table 1 below. It was observed:

International Journal of Engineering Applied Sciences and Technology, 2020
Vol. 4, Issue 10, ISSN No. 2455-2143, Pages 65-74
Published Online February 2020 in IJEAST (http://www.ijeast.com)

Table 1: Result of the five

FIS Output	ANFIS Output	Optimized 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
-0.234308411	0.018691589	0.271691589
-0.215616822	0.037383178	0.48664486
0.018028037	0.271028037	0.010009346
-0.224962617	0.028037383	0.234308411
-0.243654206	0.009345794	0.458607477
-0.038046729	0.214953271	0.234308411
-0.056738318	0.196261682	0.178233645
-0.131504673	0.121495327	0.290383178
-0.094121495	0.158878505	0.281037383
-0.066084112	0.186915888	0.159542056
-0.159542056	0.093457944	0.206271028
-0.206271028	0.046728972	0.262345794
-0.196925234	0.056074766	0.243654206
-0.206271028	0.046728972	0.271691589
-0.187579439	0.065420561	0.234308411
-0.206271028	0.046728972	0.224962617
-0.234308411	0.018691589	0.897859813
0.410551402	0.663551402	-0.36382243
-0.206271028	0.046728972	0.710943925
0.251672897	0.504672897	-0.232981308
-0.234308411	0.018691589	0.327766355
-0.159542056	0.093457944	1.010009346
0.59746729	0.85046729	0.374495327
0.718962617	0.971962617	0.196925234
0.66288785	0.91588785	0.187579439
0.59746729	0.85046729	-0.092794393
0.251672897	0.504672897	-0.186252336
-0.187579439	0.065420561	0.253
-0.187579439	0.065420561	0.916551402
0.475971963	0.728971963	-0.335785047
-0.112813084	0.140186916	0.122158879
-0.206271028	0.046728972	0.206271028

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

VII. SYSTEM EVALUATION

A total 0f 891 data set as shown in Appendix 5.1 was retrieved from Retrieval Conference (TREC) of 2011 INFILE collections. Data such as Term Weighting, Lexical Density, represent Similarity Vector and Word Ratio respectively which represent. the dataset was divided into three parts; training, validation and testing dataset in the ratio of 3:5:1 respectively which translates to 300 records for training, 484 data points for validation and 107 records for testing of the system. Since, there was no missing row components in the data set, there was no pre-processing implementation on the dataset. The screen shot presented previously in figure 1 to 8 above, shows the java programming implementation screens of the system. When corelating the results by combining the input variables and comparing their combine effects on the output, the following observations were noted:

a. Relationship between input variables and corresponding effect on the output

Performance evaluation of the interactions between Document Tracking and Retrieval parameters to examine the interaction that coexist between these parameters and how it can optimally enhance information retrieval.

i. Relationship between Lexical Density and Document Similarity Vector

Documents similarity is the computational evaluations of the level of closeness of one document with respect to the other in term of context or content. Whilst the lexical density is the evaluation of the proportion of content words in the query to a paragraph or the entire document to be retrieved. As DSV explore how a set of documents can be represented as vectors in a common vector space, LD

International Journal of Engineering Applied Sciences and Technology, 2020
Vol. 4, Issue 10, ISSN No. 2455-2143, Pages 65-74
Published Online February 2020 in IJEAST (http://www.ijeast.com)

evaluate whether the word in question is a content word. For a document to be retrieved as a relevant document that meet the user's need, strong considerations must be placed on document similarity vector and lexical density. Table 2 shows the analysis of the input range of LD and DSV with their corresponding reaction to the acceptability probability of the retrieved document. When the LD input values lies between 0% to 60% and the DSV input values lies between 0 to 2.4, then the output (i.e. the acceptability of the document) not likely, less likely and moderately likely and such the document may not meet the user's requirement. But when the input values of LD lies between 61% to 100% and that of DSV lies between 2.41 to 4.0 then the output will be More likely and most likely and as such document will meet the user's need. Figure 8 shows the relationship between the LA, DSV and the output variable.

Table 2: Relationship between LD and TW

Lexical Density (LD) in %	Document Similarity Vector (DSV) in unit	Acceptability (Output)
0-20 –VLD	0-0.80 – VS	0-0.20 – NL
21-40 – LD	0.81-1.60 – S	0.21-0.40 – LL
41-60 – SD	1.61-2.40 – MS	0.41-0.60 – ML
61-80 – HD	2.41-3.20 – SS	0.61-0.80 – ML
81-100- VHD	2.21-4.00 – NS	0.80-1.00 - ML

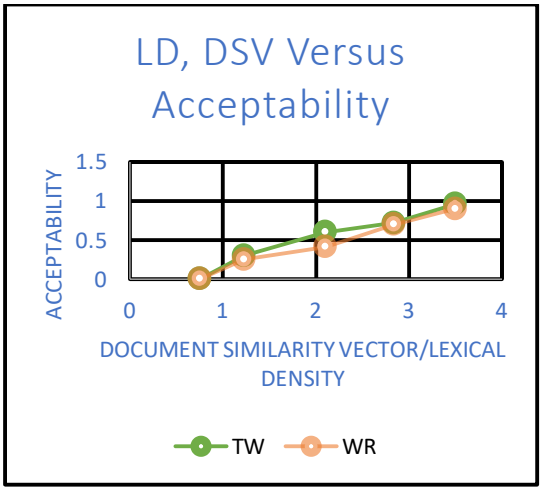

Fig. 8: Coloration Plot between LD and DSV

The lexical Density and document similarity vector are most necessary to exist in their upper value for the document to be selected.

b. Relationship between Lexical Density and Word Ratio

The Word ratio is the relational number of repetitions of each keyword in the extracted set of words. Whilst the lexical density is the evaluation of the proportion of content words in the query to a paragraph or the entire document to be retrieved. As WR evaluate the number of times one value contains or is contained within the other, LD evaluate whether the word in question is a content word. For a word in the query to be gauge in retrieving relevant document that meet the user's need it is very important that the ratio of word and the lexical density be considered. Table 3 shows the analysis of the input range of LD and WR with their corresponding reaction to the acceptability probability of the retrieved document. When the LD input values lies between 0% to 60% and the WR input values lies between 0 to 0.60, then the output (i.e. the acceptability of the document) not likely, less likely and moderately likely and such the document may not meet the user's requirement. But when the input values of LD lies between 61% to 100% and that of WR lies between 0.61 to 1.0 then the output will be More likely and most likely and as such document will meet the user's need. Figure 9 shows the relationship between the LA, WR and the output variable.

Table 3: Relationship between LD and WR

Lexical Density (LD) in %	Word Ratio (WR) in unit	Acceptability (Output)
0-20 –VLD	0-0.20 – VC	0-0.20 – NL
21-40 – LD	0.21-0.40 – C	0.21-0.40 – LL
41-60 – SD	0.41-0.60 – SC	0.41-0.60 – ML
61-80 – HD	0.61-0.80 – NC	0.61-0.80 – ML
81-100- VHD	0.80-1.00 – NR	0.80-1.00 – ML

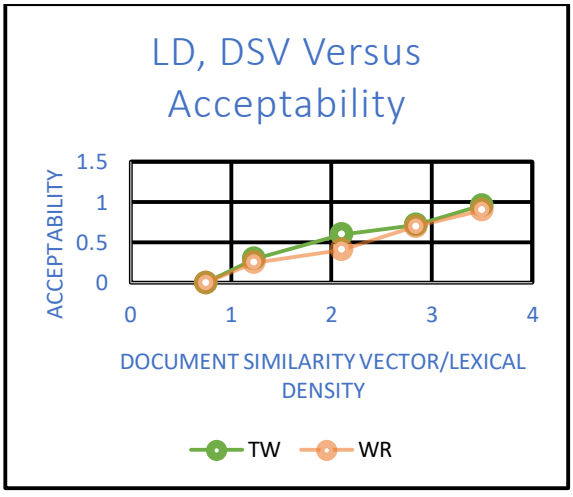

Fig. 9: Coloration Plot between LD and DSV

The lexical Density and document similarity vector are most necessary to exist in their upper value for the document to be selected.

c. Relationship between Lexical Density and Word Ratio

The Word ratio is the relational number of repetitions of each keyword in the extracted set of words. Whilst the lexical density is the evaluation of the proportion of content words in the query to a paragraph or the entire document to be retrieved. As WR evaluate the number of times one value contains or is contained within the other, LD evaluate whether the word in question is a content word. For a word in the query to be gauge in retrieving

International Journal of Engineering Applied Sciences and Technology, 2020
Vol. 4, Issue 10, ISSN No. 2455-2143, Pages 65-74
Published Online February 2020 in IJEAST (http://www.ijeast.com)

relevant document that meet the user's need it is very important that the ratio of word and the lexical density be considered. Table 3 shows the analysis of the input range of LD and WR with their corresponding reaction to the acceptability probability of the retrieved document. When the LD input values lies between 0% to 60% and the WR input values lies between 0 to 0.60, then the output (i.e. the acceptability of the document) not likely, less likely and moderately likely and such the document may not meet the user's requirement. But when the input values of LD lies between 61% to 100% and that of WR lies between 0.61 to 1.0 then the output will be More likely and most likely and as such document will meet the user's need. Figure 9 shows the relationship between the LA, WR and the output variable.

Table 4: Relationship between LD and WR

Lexical Density (LD) in %	Word Ratio (WR) in unit	Acceptability (Output)
0-20 –VLD	0-0.20 – VC	0-0.20 – NL
21-40 – LD	0.21-0.40 – C	0.21-0.40 – LL
41-60 – SD	0.41-0.60 – SC	0.41-0.60 – ML
61-80 – HD	0.61-0.80 – NC	0.61-0.80 – ML
81-100-VHD	0.80-1.00 – NR	0.80-1.00 - ML

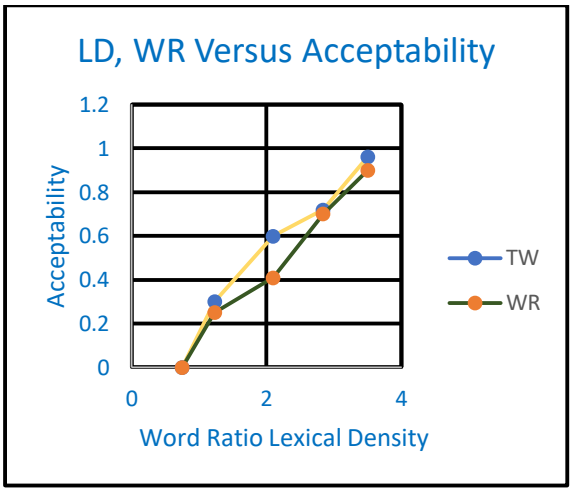

Fig. 10: Coloration Plot between LD and WR

The lexical Density and word ratio are most necessary to exist in their upper value for the document to be selected.

d. Relationship between Document Similarity Vector and Word Ratio

For two documents to be considered similar to each other, the ratio of the numbers of repeated word in the two documents must either be the same or very close. Table 4 shows the analysis of the input range of DSV and WR with their corresponding reaction to the acceptability probability of the retrieved document. When the DSV input values lies between 0 to 2.40 and the WR input values lies between 0 to 0.60, then the output (i.e. the acceptability of the document) not likely, less likely and moderately likely and such the document may not meet the user's requirement. But when the input values of DSV lies between 2.61 to 4.00 and that of WR lies between 0.61 to 1.0 then the output will be More likely and most likely and as such document will meet the user's need. Figure 10 shows the relationship between the DSV, WR and the output variable.

Table 5: Relationship between DSV and WR

Document Similarity Vector (DSV) in unit	Word Ratio (WR) in unit	Acceptability (Output)
0-0.80 – VS	0-0.20 – VC	0-0.20 – NL
0.81-1.60 – S	0.21-0.40 – C	0.21-0.40 – LL
1.61-2.40 – MS	0.41-0.60 – SC	0.41-0.60 – ML
2.41-3.20 – SS	0.61-0.80 – NC	0.61-0.80 – ML
2.21-4.00 – NS	0.80-1.00 – NR	0.80-1.00 – ML

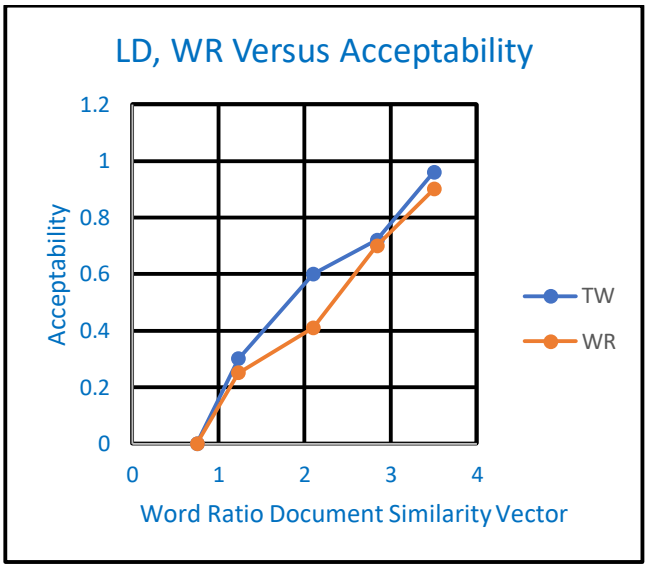

Fig. 11: Coloration Plot between LD and TW

The Document Similarity Vector must be at the threshold and Word Ratio at the upper value for the document to be selected.

e. Relationship between Lexical Density and Term Weighting

The term weight is the measure of the importance of each word in a sentence that forms the user's query to a paragraph or the entire document to be retrieved. Whilst the lexical density is the evaluation of the proportion of content words in the query to a paragraph or the entire document to be retrieved. As TW evaluate the importance of the word, LD evaluate the weather the word in question is a content word. For a word in the query to be gauge in retrieving relevant document that meet the user's need it is very important LD and TW be considered. Table 5 shows the analysis of the input range of LD and TW with their corresponding reaction to the acceptability probability of the retrieved document. When the LD input values lies between 0% to 60% and the TW input values lies between 0 to 0.60, then the output (i.e. the acceptability of the document) not likely, less likely and moderately likely and such the document may not meet the user's requirement. But when the input values of LD lies between 61% to 100% and that of TW lies between 0.61 to 1.0 then the

International Journal of Engineering Applied Sciences and Technology, 2020
Vol. 4, Issue 10, ISSN No. 2455-2143, Pages 65-74
Published Online February 2020 in IJEAST (http://www.ijeast.com)

output will be More likely and most likely and as such document will meet the user's need. Figure 11 shows the relationship between the LD, TW and the output variable.

Table 6: Relationship between LD and TW

Lexical Density (LD) in %	Term Weighting (TW) in Frequency	Acceptability (Output)
0-20 –VLD	0-0.20 – VWF	0-0.20 – NL
21-40 – LD	0.21-0.40 – WF	0.21-0.40 – LL
41-60 – SD	0.41-0.60 – LF	0.41-0.60 – ML
61-80 – HD	0.61-0.80 – SF	0.61-0.80 – ML
81-100- VHD	0.80-1.00 – VSF	0.80-1.00 - ML

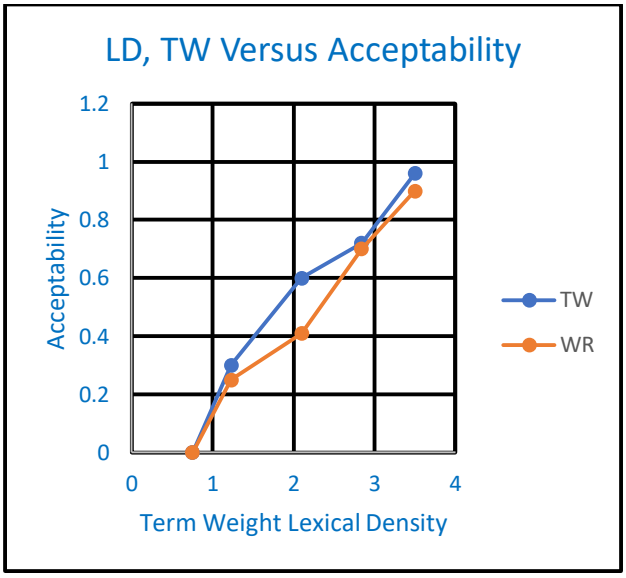

Fig. 12: Coloration Plot between LD and TW

The lexical Density and term weight are most necessary to exist in their upper value for the document to be selected.

f. Relationship between Term Weighting and Word Ratio

The term weight is the measure of the importance of each word in a sentence that forms the user's query to a paragraph or the entire document to be retrieved. Whilst Word ratio is the relational number of repetitions of each keyword in the extracted set of words. As TW evaluate the importance of the word, WR evaluate the relational number of repetitions of each keyword in the extracted set of words. For a word in the query to be gauge in retrieving relevant document that meet the user's need, the measure of term weight and word ratio is very important each paragraph or entire document to be retrieved. Table 6 shows the analysis of the input range of TW and WR with their corresponding reaction to the acceptability probability of the retrieved document. When the WR input values lies between 0 to 0.60 and the TW input values lies between 0 to 0.60, then the output (i.e. the acceptability of the document) not likely, less likely and moderately likely and such the document may not meet the user's requirement. But when the input values of WR input values lies between 0.61 to 1.00 and that of TW lies between 0.61 to 1.0 then the output will be More likely and most likely and as such document will meet the user's need. Figure 12 shows the relationship between the WR, TW and the output variable.

Table 7: Relationship between WR and TW

Word Ratio (WR) in unit	Term Weighting (TW) in frequency	Acceptability (Output)
0-0.20 – VC	0-0.20 – VWF	0-0.20 – NL
0.21-0.40 – C	0.21-0.40 – WF	0.21-0.40 – LL
0.41-0.60 – SC	0.41-0.60 – LF	0.41-0.60 – ML
0.61-0.80 – NC	0.61-0.80 – SF	0.61-0.80 – ML
0.80-1.00 - NR	0.80-1.00 – VSF	0.80-1.00 - ML

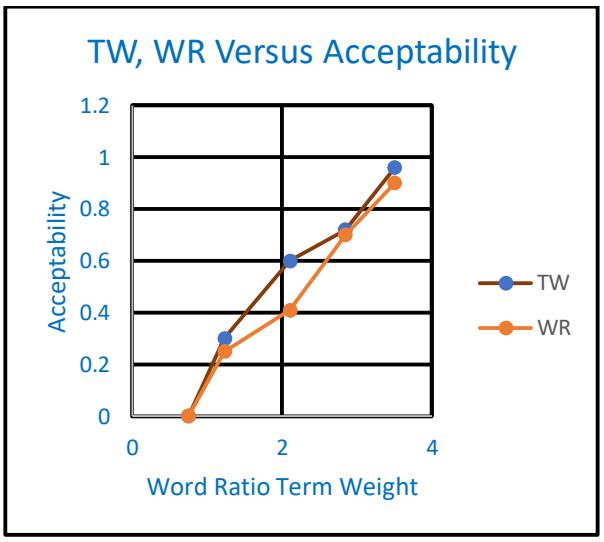

Fig. 13: Coloration Plot between WR and TW

The word ratio and term weight are most necessary to exist at the same threshold for the document to be selected.

VIII. 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 MatL AB and the graph of the sensitivity of the input to the output was displayed as shown in figure 13. The results of the sensitivity test are shown in Table 7.

International Journal of Engineering Applied Sciences and Technology, 2020
Vol. 4, Issue 10, ISSN No. 2455-2143, Pages 65-74
Published Online February 2020 in IJEAST (http://www.ijeast.com)

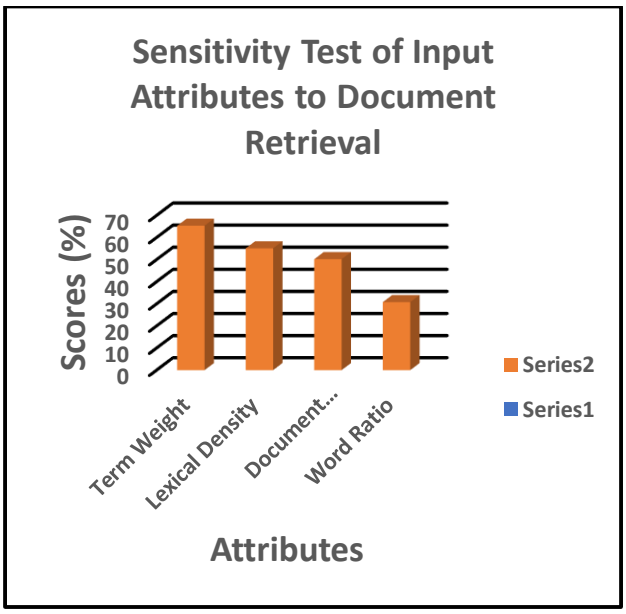

Fig. 14: Sensitivity of Pipeline Variable to Document Retrieval

Table 8: Sensitivity Test of Input Attributes to Flow rate

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

Table 8 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.

IX. CONCLUSION

From the research, 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 the error tolerance of 0.0001 specified in the train FIS. The idea behind using a checking data set for model validation is that after a certain point in the training, the model begins over fitting the training data set. In principle, the model error for the checking data set tends to decrease as the training takes place up to the points that over fitting begins, and then the model error for the checking data suddenly increases. Over fitting is accounted for by testing the FIS trained on the training data against the checking data, and chosen the membership function parameter to be those associated with the minimum checking error if these errors indicate model over fitting. ANFIS model was designed using Takagi Sugeno inference mechanism. Java programming language was used on a windows 10 platform to integrate the Matlab ANFIS output for better optimization and MySQL database 5.7.14 from WAMP server 3.0.6 was used as the back-end engine. To validate this work, data was collected by retrieving test data from Text Retrieval Conference (TREC) of 2011. We used the hybrid supervised learning approach in training our network. The training, testing and checking KMSI values of 0.026347, 0.026073, 0.025819 respectively were observed in the hybrid learning process at 150 epochs. The ANFIS processes faster and have a minimal error of 0.026344 at 300 epochs. Average error of 0.047283 was observed in the hybrid algorithm against the average error of 0.024642 with hybrid learning at 300 epochs. Therefore, ANFIS evaluation of NFDTRS with hybrid learning performed better than the FIS evaluation. IR-ANFIS system succeeded in enhancing the results achieved by previous IR systems which used fuzzy logic. As one can see, system proved to outperform IR-FIS and other industry standard search engines. Secondly, it was discovered that the standard use of Neuro-Fuzzy techniques as it is used in other fields slightly improved the performance in the information retrieval field.

X. REFERENCES

1. Frank, W. (2002). “FUZZY CLUSTERING IN DOCUMENT CLASSIFICATION”; Knowledge based Systems Group; Paderborn, Pp. 30. .2002
2. Franke, J., Nakhaeizadeh, G., and Renz, I. (2003). “Text Mining, Theoretical Aspects and Applications.” Springer, ISSN 0302-9743, ISBN 3-540- 69572-9, 2003.
3. Fuhr, N. (1991). "A probabilistic learning approach for document indexing. ACM Transactions on Information Systems", Vol. 9, Pp. 223-248., 1991.
4. Iwok, S O (2018) A Model of Intelligent Packet Switching in Wireless Communication Networks. PhD Thesis, Department of Computer Science, Ebonyi State University Abakaliki.
5. Salton, G., (1983). "Extended Boolean information retrieval ", Vol. 26, Pp. 600-609, Communications of the ACM, 1983.
6. Soundarya, V. and Manjula, D. (2016). “Neuro-Fuzzy Classification Techniques for Sentiment Analysis using Intelligent Agents on Twitter Data”; International Journal of Innovation and Scientific Research. ISSN 2351-8014 Vol. 23 No. 2 May 2016, pp. 356-360. © 2015 Innovative

Space of Scientific Research Journals http://www.ijisr.issr-journals.org/

7. Udoh, S. S. (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.
8. Yuanyam, C., Limin J., and 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.
9. Wu, B-F., Chiu, C. C., and Chen, Y. L. (2004). "Algorithms for compressing compound document images with large text/background overlap." *IEE Proceedings-Vision, Image and Signal Processing* 151.6 (2004): 453-459
10. Wang, S., Socher, R.; Perelygin, A.; Wu, J. Y.; Chuang, J and Manning, C. D. (2017). Baselines and bigrams: Simple, good sentiment and topic classification. In *ACL*, 90–94.
11. Zhong, S. (2005). Generative Model-based document clustering: A comparative study. Knowledge and Information Systems, 8(3):374-384.