SENTIMENT CLASSIFICATION USING OPTIMIZATION TECHNIQUES

A PROJECT REPORT

Submitted by

HARSH MODI

180170116021

MAHARSHI PRAJAPATI

180170116037

In partial fulfilment for the award of the degree of

BACHELOR OF ENGINEERING

in

Information Technology

Vishwakarma Government Engineering College, Ahmedabad.

Gujarat Technological University, Ahmedabad

April, 2022

VISHWAKARMA GOVERNMENT ENGINEERING COLLEGE

Visat - Gandhinagar Road, Chandkheda, Ahmedabad-382424

CERTIFICATE

This is to certify that the project report submitted along with the project entitled **Sentiment Classification Using Optimization Techniques** has been carried out by **HARSH MODI** under my guidance in partial fulfilment for the degree of Bachelor of Engineering in Information Technology, 8th Semester of Gujarat Technological University, Ahmedabad during the academic year 2021-22.

Prof. Dipak Patel	Dr. Vibha Patel
Internal Guide	Head of the Department

VISHWAKARMA GOVERNMENT ENGINEERING COLLEGE

Visat - Gandhinagar Road, Chandkheda, Ahmedabad-382424

CERTIFICATE

This is to certify that the project report submitted along with the project entitled **Sentiment Classification Using Optimization Techniques** has been carried out by **MAHARSHI PRAJAPATI** under my guidance in partial fulfilment for the degree of Bachelor of Engineering in Information Technology, 8th Semester of Gujarat Technological University, Ahmedabad during the academic year 2021-22.

___________________ ___________________

Prof. Dipak Patel	Dr. Vibha Patel
Internal Guide	Head of the Department

VISHWAKARMA GOVERNMENT ENGINEERING COLLEGE

Visat - Gandhinagar Road, Chandkheda, Ahmedabad-382424

DECLARATION

We hereby declare that the Project report submitted along with the Project entitled **Sentiment Classification Using Optimization Techniques** submitted in partial fulfilment for the degree of Bachelor of Engineering in Information Technology to Gujarat Technological University, Ahmedabad is a bonafide record of original project work carried out by us at Vishwakarma Government Engineering College under the supervision of Professor Dipak Patel and that no part of this report has been directly copied from any students' reports or taken from any other source, without providing due reference.

Name of the Student	Sign of Student
1. Harsh Modi	________________
2. Maharshi Prajapati	________________

ACKNOWLEDGEMENT

We would like to begin by expressing our heart-felt gratitude to our advisor, Professor Dipak Patel, for his constant guidance and support without which we could not have completed this project. We consider our self to be extremely privileged to have been his student. I am very grateful to him for being very patient and for all his time that she spent in discussing the various subjects of this project, and for constantly guiding us in the right path with the help of which this project came to a successful completion.

Last but not the least we would like to mention a heartful thanks to everyone who has somehow helped us in this journey to complete this project but whose name does not find a place in this acknowledgement.

TABLE OF CONTENT

ABSTRACT

Sentiment Analysis also known as Opinion Mining refers to the use of natural language processing, text analysis to systematically identify, extract, quantify, and study affective states and subjective information. Sentiment analysis is widely applied to reviews and survey responses, online and social media, and healthcare materials for applications that range from marketing to customer service to clinical medicine. As the amount of content that is created on social media is constantly increasing, more and more opinions and sentiments are expressed by people in various subjects. In this respect, sentiment analysis and opinion mining techniques can be valuable.

Initially, using different conventional techniques of neural network, Sentiment Analysis had performed on the single domain review dataset but accuracy of that was unsatisfiable and cannot be used on use cases. In process of improving accuracy of model, distinct swarm intelligence based optimization techniques came at our glance which we have applied to improvise the model accuracy. Moreover, most popular swarm intelligence base technique known as the Grey Wolf Optimizer (GWO) developed by computer researcher Professor Seyedali Mirjalili have been taken in consideration for improvisation purpose with distinct variations of the GWO.

We have successfully improved the test accuracy of the classification model by 3% to 15% compared to the conventional methods available on Keras [12] and train accuracy has improved around 1% to 6% for the same.

LIST OF FIGURES

LIST OF TABLES

CHAPTER-1: INTRODUCTION

Sentiment is an attitude, thought, or judgment prompted by feeling. Sentiment analysis, which is also known as opinion mining, studies people's sentiments towards certain entities. From a user's perspective, people are able to post their own content through various social media, such as forums, micro-blogs, or online social networking sites. From a researcher's perspective, many social media sites release their application programming interfaces (APIs), prompting data collection and analysis by researchers and developers. However, those types of online data have several flaws that potentially hinder the process of sentiment analysis. The first flaw is that since people can freely post their own content, the quality of their opinions cannot be guaranteed. The second flaw is that ground truth of such online data is not always available. A ground truth is more like a tag of a certain opinion, indicating whether the opinion is positive, negative, or neutral. In recent years, we have witnessed that opinionated postings in social media have helped reshape businesses, and sway public sentiments and emotions, which have profoundly impacted on our social and political systems.

1.1 Project Summary

A basic task in sentiment analysis is classifying the polarity of a given text at the document, sentence, or feature/aspect level — whether the expressed opinion in a document, a sentence or an entity feature/aspect is positive, negative, or neutral. Advanced, "beyond polarity" sentiment classification looks, for instance, at emotional states such as "angry", "sad", and "happy".

In this project we analysed or rather classify the sentiments in two parts i.e., positive and negative with the help of Artificial Neural Network. Initially, using different conventional techniques of neural networks, Sentiment Analysis had performed on the single domain review dataset but accuracy of that was unsatisfiable and cannot be used on use cases. In process of improving accuracy of model, distinct swarm intelligence based optimization techniques came at our glance which we have applied to improvise the model accuracy. Moreover, most popular swarm intelligence base technique known as the Grey Wolf Optimizer (GWO) developed by computer researcher Professor Seyedali

Mirjalili have been taken in consideration for improvisation purpose with distinct variations of the GWO.

To conclude, throughout this project we successfully managed to improve the testing accuracy of the model approximately by 3% to 15% and also improved training accuracy by 1% to 6% using various variation of GWO.

1.2 Project Scope

There is a vast scope of Sentiment Analysis, we already know that being a natural language processing (NLP) technique which is used to analyse if a given statement is negative or positive and thus by knowing the sentiment behind an opinion it's easy to use it for our benefit, so some of the applications include-

- To classify if a tweet/social media post is negative or positive this can be used by big corporations to identify if the recent social media activity is for or against their company and to address all the needy negative comments.
- To predict an outcome of an election, we can compile and analyse large amount of data from social media and news to see which candidate has more positive support towards them.
- E-Commerce companies can use this to uncover customer attitudes on service, products etc so as to improve the quality of that product.
- Companies can also use it to check their newly launched product response in market by classifying the posts related to the product.

Specifically, considering this project also contains vast variety of possibilities when it comes to its application like –

- The optimizer has simple architecture.
- This project is based on one optimizer and its variations.
- Optimizer is inconvenient for moderate machine specification on very large data.
- For the data pre-processing part, it's unique for different dataset.
- This project is easy to implement and easily suitable other Neural Networks but it requires some minor upgrade.
- The GWO optimizer doesn't give optimal answers in all Benchmark Functions.

- The project is not suitable for Image and Video Dataset. It is only work with dataset that contains Text information.
- It requires performing whole process for multiple times to achieve better result.

1.3 Objectives

- Opinions are a usually subjective expression that describes people's sentiments towards something, now it's easy for a human mind to understand if an opinion directed towards them is good or bad thus sentiment analysis is easy for a human being.
- Now we will do the same Sentiment Analysis for a large chunk of data using a natural language processing technique which will determine whether data is positive, negative or neutral.
- Our Main Objective is to improve overall accuracy of the model which is higher than the conventional method.
- It should be work faster for any type of application.
- It should make project more scalable.
- Implementing GWO in conventional artificial neural network.
- And improve the optimizing capacity of the GWO.
- At the end of this project our model will be easily able to detect whether the sentiment of the opinion presented to it is positive or negative.
- Now, it is also important to make the model accurate thus our focus will be to increase accuracy of our model.

1.4 Plan of work

1. Understand the conventional artificial neural network methods deeply.
2. Implement different preprocessing techniques in the dataset.
3. Implement Artificial Neural Network in Python.
4. Apply forward and backward propagations on neural network.
5. Improve scalability of the artificial neural network model.
6. Optimizing different factors of neurons such as weight and bias.
7. Identifying demerits of traditional artificial neural network model.

8. Finding the solution those demerits.
9. Improve the capacity of that solution by adding our own features in model.

1.5 Literature Review/Background Study

1.5.1 Fundamentals

1. **Deep understanding of customs [1]**
    - It contains detailed explanations about each topic which have been used form decades for sentiment classification.
    - For example: Optimizers, activations functions, kernal initializer, etc.
2. **GWO-Grey Wolf Optimizer [2]**
    - Explains swarm intelligence-based optimization techniques with detailed studies done by computer researcher Prof. Seyedali Mirjalili.
    - It is based on the hierarchy and the hunting mechanism of the wolves, whereas in our scenario we are hunting the optimal solution and wolf population is variables we are trying the optimize for getting optimize answers.
3. **Different aspects of the artificial neural network [3]**
    - Playlist of all the topic required to understand the minor details about artificial neural network.
    - For example, underestimation, overestimation, convergence, divergence etc.

1.5.2 Research paper published in leading journals and conferences

1. **A survey on sentiment analysis challenges [4]**
    - This paper presents a survey on the sentiment analysis challenges relevant to their approaches and techniques with their conclusions and theoretical and technical classification of the problems.
2. **Comparison of Sentiment Analysis and Domain Adaptation Techniques with Research Scopes [5]**
    - Provided us more insights about the sentiment analysis and its research scope.

3. **Multi-Source Domain Adaptation in Sentiment Analysis using Optimized Neural Network and Cross-Domain Semantic Library [6]**
    - Existing domain adaptation based sentiment analysis techniques are described and their drawbacks are discussed.
    - Also provides the insights about new variant of the GWO known as the IGWO-Improved Grey Wolf Optimizer.

4. **GOW-Grey Wolf Optimizer[7]**
    - This work proposes a new meta-heuristic called Grey Wolf Optimizer (GWO) inspired by grey wolves (Canis lupus).
    - The GWO algorithm mimics the leadership hierarchy and hunting mechanism of grey wolves in nature. Four types of grey wolves such as alpha, beta, delta, and omega are employed for simulating the leadership hierarchy.
    - In addition, the three main steps of hunting, searching for prey, encircling prey, and attacking prey, are implemented.

5. **Review of the different enhancements of GWO [9]**
    - This paper shows the summary of different enhancement done by the researchers in the GWO.
    - It says that majority of the development in the GWO algorithm has done in following four aspects of the algorithm:
        1. Change in parameter
        2. Modification of Position update equation
        3. Hybridization with other search techniques
        4. Introduction of new operators into GWO

6. **A Fuzzy Hierarchical Operator in the Grey Wolf Optimizer Algorithm [10]**
    - This paper is based on the popular artificial intelligence technique known as “Fuzzy Logic” and its application in this optimization algorithm of GWO.
    - Conventional GWO is based on the binary decisions of the alpha, beta and delta wolves but this one suggests that it does not dependent on any one wolf binarily all the time.

- It enhances its decisive mechanism by applying fuzzy logic.
- For example, in GWO as the iteration goes on the credibility of the delta wolf decreases and the alpha wolf's credibility increases.

7. **A Novel Grey Wolf Optimizer Algorithm with Refraction Learning [11]**
   - This advancement of GWO has made based on very popular optical physics' concept of refraction learning.
   - As the light ray changes the path after entering in the different medium that concept has used in this advancement as pushing the variable in the completely opposite direction from their current situation.
   - This enhances its capacity to jump out of the local optima and search in wider search space.

8. **Modified Grey Wolf Optimizer for Global Engineering Optimization [12]**
   - This paper discusses the improvement of the GWO by change in the iteration decay equation.

   $$a = 2\left(1 - \frac{t}{T}\right) \longrightarrow a = 2\left(1 - \frac{t^2}{T^2}\right) \qquad (1.5.1)$$

   - As seen above it increases the degree of the decay equation which in turn allow us to search in search space more efficiently.

Table 1.1 Literature Review

Fundamentals		Remarks
1	Deep understanding of customs	Forward-Brackword propogation,Activation function etc.
2	GWO-Grey Wolf Optimizer	Detailed study of the optimizer
3	Different aspects of the artificial neural network	Concept of artificial neural network like underestimation, over-Estimation, preprocessing, python keras module etc.
Research paper published in leading journals and conferences		
1	survey on sentiment analysis challenges	Theoratical and technical challenges
2	Comparison of Sentiment Analysis and Domain Adaptation Techniques with Research Scopes	Deep in to Sentiment analysis
3	Multi-Source Domain Adaptation in Sentiment Analysis using Optimized Neural Network and Cross-Domain Semantic Library	Basic architecture of Sentiment analysis
4	GWO-Grey Wolf Optimizer	Seyedali Mirjalili's research paper on GWO
5	Review of the different enhancements of GWO	Classification of GWO enhancements
6	A Fuzzy Hierarchical Operator in the Grey Wolf Optimizer Algorithm	• Artificial intelligence technique in GWO • Non-binary credibility of the wolves in each iteration
7	A Novel Grey Wolf Optimizer Algorithm with Refraction Learning	• Optical physics' law of refraction is applied • Pushing the optimizer out of local optima when stuck • Better search space exploration
8	Modified Grey Wolf Optimizer for Global Engineering Optimization	• Improving the iteration decay equation • Exploration = 70 % • Exploitation = 30 %

1.6 Technologies Used

Table 1.2 Used Technologies Table

Components	Reason
Development Tool 1. Google Colab 2. Jupyter Notebook 3. Virtual Studio	write and execute arbitrary python code through the browser, and is especially well suited to machine learning
Database 1. Excel	To store data which can be processed while preparing model that data can be used to train and test the model
Programming Language 1. Python	The Python language comes with many libraries and frameworks that make coding easy. This also saves a significant amount of time.
Libraries 1. Tensorflow 2. Numpy 3. Pandas 4. Keras etc.	TensorFlow is a Python library for fast numerical computing created and released by Google. It is a foundation library that can be used to create artificial neural network models directly or by using wrapper libraries that simplify the process built on top of TensorFlow

- **Development Tool:**
  1. Google Colab
  2. Jupyter Notebook
  3. Virtual Studio Code
- **Database:**
  1. Excel: To store data which can be processed while preparing model that data can be used to train and test the model.
- **Programming Language:**
  1. Python: The Python language comes with many libraries and frameworks that make coding easy. This also saves a significant amount of time.
- **Libraries**
  1. Tensorflow: TensorFlow is an open-source library developed by Google primarily for artificial neural network applications.

2. Numpy: NumPy can be used to perform a wide variety of mathematical operations on arrays.
3. Pandas: Pandas is mainly used for data analysis and associated manipulation of tabular data in Dataframes.
4. Keras: Keras is a neural network Application Programming Interface (API) for Python that is tightly integrated with TensorFlow, which is used to build machine learning models.
5. Other Libraries such as – time, math, etc.

CHAPTER- 2 Grey Wolf Optimizer and its Variations

2.1 Grey Wolf Optimizer

This paper contains all the things about the great swarm intelligence based(meta-heuristic inspired by the Grey wolf) technique GWO. The GWO algorithm mimics the leadership hierarchy and hunting mechanism of grey wolves in nature. Four types of grey wolves such as alpha, beta, delta, and omega are employed for simulating the leadership hierarchy.

2.1.1 Phases of Grey Wolf hunting

1. Tracking, chasing, and approaching the prey
2. Pursuing, encircling, and harassing the prey until it stops moving
3. Attack towards the prey

Hunting behaviour of grey wolves: (A) chasing, approaching, and tracking prey (B-D) pursuiting, harassing, and encircling (E) stationary situation and attack

Fig. 2.1 Hunting Behaviour of gray wolves

2.1.2 Mathematical Model of Hierarchy, Tracking, Encircling and Attacking of Prey:

- **Social Hierarchy:** Alpha, Beta, Delta and Omega are the symbols in which the heard is divided for the mathematical understanding. Like Alpha is the leader of the group and beta and delta follows the alpha in the control over the heard over the group. Omega wolves are the wolves that don't have any kind of the control in the group they are meant to be following the instruction of the main three wolves in the top of the pyramid.

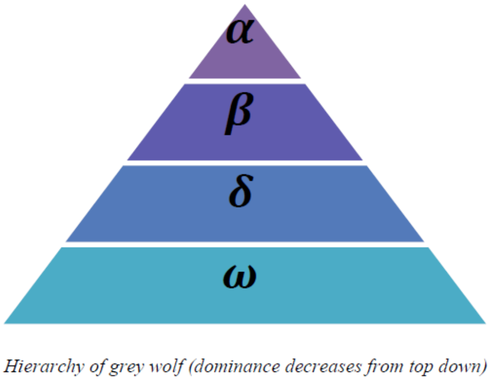


Fig 2.2 Hierarchy of grey wolf

- **Encircling the Prey:** As mentioned above, grey wolves encircle prey during the hunt. In order to mathematically model encircling behaviour the following equations are proposed:

$$\vec{D} = \left|\vec{C}.\overrightarrow{X_p}(t) - \vec{X}(t)\right| \tag{2.1.1}$$

$$\vec{X}(t+1) = \overrightarrow{X_p}(t) - \vec{A}.\vec{D} \tag{2.1.2}$$

t = Current Iteration

C, A = Coefficient Vectors

Xp = Position vector of the prey

X = Position vector of the Grey Wolf

The vectors and are calculated as follows:

$$\vec{A} = 2\vec{a}.\overrightarrow{r_1} - \vec{a} \tag{2.1.3}$$

$$\vec{C} = 2.\overrightarrow{r_2} \tag{2.1.4}$$

a = Linearly decreasing form 2 to 0 over the iterations

r1, r2 = random variables in range [0, 1]

- **Hunting:** Grey wolves have the ability to recognize the location of prey and encircle them. The hunt is usually guided by the alpha. The beta and delta might also participate in hunting occasionally. However, in an abstract search space we have no idea about the location of the optimum (prey). In order to mathematically simulate the hunting behaviour of grey wolves, we suppose that the alpha (best candidate solution) beta, and delta have better knowledge about the potential location of prey. Therefore, we save the first three best solutions obtained so far and oblige the other search agents (including the omegas) to update their positions according to the position of the best search agent.

The following formulas are proposed in this regard:

$$\overrightarrow{D_\alpha} = \left|\vec{C}.\overrightarrow{X_\alpha} - \vec{X}\right|, \overrightarrow{D_\beta} = \left|\vec{C}.\overrightarrow{X_\beta} - \vec{X}\right|, \overrightarrow{D_\delta} = \left|\overrightarrow{C_3}.\overrightarrow{X_\delta} - \vec{X}\right| \quad (2.1.5)$$

$$\overrightarrow{X_1} = \overrightarrow{X_\alpha} - \overrightarrow{A_1}.\left(\overrightarrow{D_\alpha}\right), \overrightarrow{X_2} = \overrightarrow{X_\beta} - \overrightarrow{A_2}.\left(\overrightarrow{D_\beta}\right), \overrightarrow{X_3} = \overrightarrow{X_\delta} - \overrightarrow{A_3}.\left(\overrightarrow{D_\delta}\right) \quad (2.1.6)$$

$$\vec{X}(t+1) = \frac{\overrightarrow{X_1}+\overrightarrow{X_2}+\overrightarrow{X_3}}{3} \quad (2.1.7)$$

Figure below shows how a search agent updates its position according to alpha, beta, and delta in a 2D search space. It can be observed that the final position would be in a random place within a circle which is defined by the positions of alpha, beta, and delta in the search space. In other words, alpha, beta, and delta estimate the position of the prey, and other wolves updates their positions randomly around the prey.

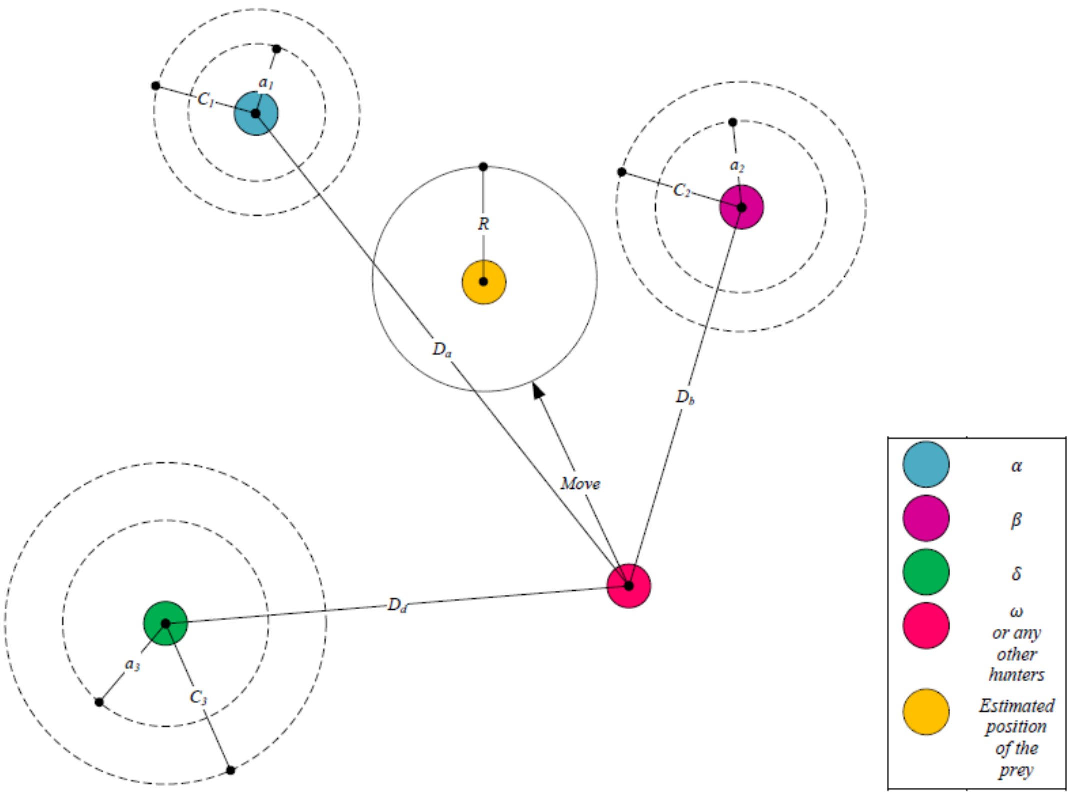


Fig 2.3 Position updating in GWO

- **Attacking Prey (Exploitation):** Wolves attack prey when it stops moving much and to stimulate this situation, we are decreasing the value of "a" over the iterations from 2 to 0 which intern allow wolves to exploit the search space and attack the wolf.

  With change in "a", "A" also changes between [-a, a] randomly which in tern becomes the decisive factor weather to attack the prey or not.

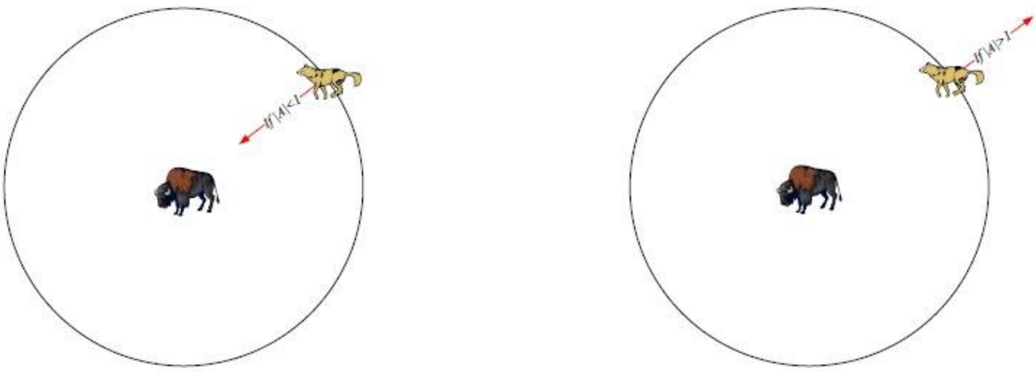


Fig. 2.4 Attacking Prey

If |A| < 1 ➔ Attack the prey

|A| > 1 ➔ Explore the search space.

- **Search for Prey(Exploration):** It is based on the potions of Alpha, Beta and Delta wolves. They tend to diverge from the prey to search in the global search space and converge in order to attack the prey which is called attack the prey as explained before. For divergence we choose values of A randomly and when the scenario like |A| > 1 comes up.

  In terms of Variable C which favours the exploration which contains random values between 0 and 2. Moreover, this stochastically emphasize(C > 1) or deemphasize(C < 1) the effect of prey in defining distance.

  This assists GWO to show a more random behaviour throughout optimization, favouring exploration and local optima avoidance.

2.1.3 GWO Algorithm:

Initialize the grey wolf population X_i *(i = 1, 2, ..., n)*
Initialize a, A, and C
Calculate the fitness of each search agent
X_α*=the best search agent*
X_β*=the second best search agent*
X_δ*=the third best search agent*
while *(t < Max number of iterations)*
    for *each search agent*
        Update the position of the current search agent by equation (2.1.7)
    end for
    Update a, A, and C
    Calculate the fitness of all search agents
    Update X_α, X_β, *and* X_δ
    t=t+1
end while
return X_α

Fig. 2.5 GWO Algorithm

2.2 Fuzzy Hierarchical Operator in the Grey Wolf Optimizer Algorithm [10]

- This paper is based on the popular artificial intelligence technique known as "Fuzzy Logic" and its application in this optimization algorithm of GWO.

- Conventional GWO is based on the binary decisions of the alpha, beta and delta wolves but this one suggests that it does not dependent on any one wolf binarily all the time.

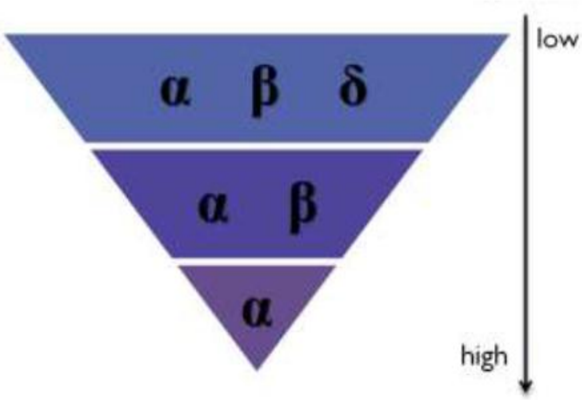


Fig 2.6 Priority of wolves

- In each iteration it depends on each wolf by some fraction of their credibility as shown in the figure above.
- It enhances its decisive mechanism by applying fuzzy logic.
- For example, in GWO as the iteration goes on the credibility of the delta wolf decreases and the alpha wolf's credibility increases.
- It can also be increased by the using the reverse hierarchy as shown in the figure below:

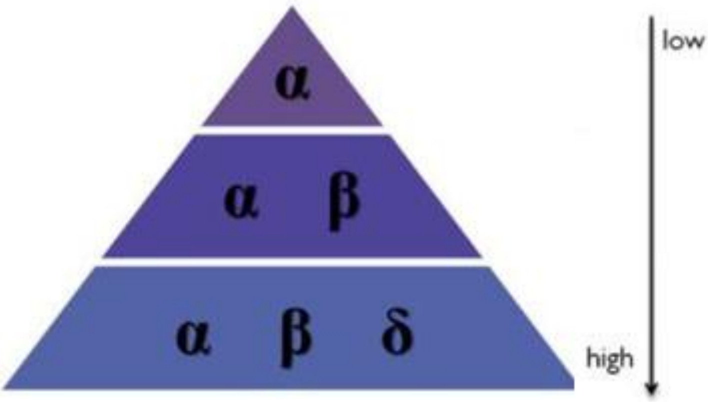


Fig 2.7 Priority of wolves

- For wolves hierarchy it can not be used as at the commencement of the hunt the credibility of all the wolves would be equal.
- As shown below the technical implementation can be done:

Rules in increase
1.- if(iterations is low) then (alpha is medium) (beta is medium) (delta is medium)
2.- if(iterations is medium) then (alpha is medium) (beta is medium) (delta is low)
3.- if(iterations is high) then (alpha is high) (beta is medium) (delta is low)

Fig 2.8 Fuzzy Logic rules

- As part of GWO enhancement we have decided to reduce the credibility of the Delta wolf to zero as the algorithm reaches to the final iteration.

2.3 A Novel Grey Wolf Optimizer Algorithm with Refraction Learning [11]

- This advancement of GWO has made based on very popular optical physics' concept of refraction learning.

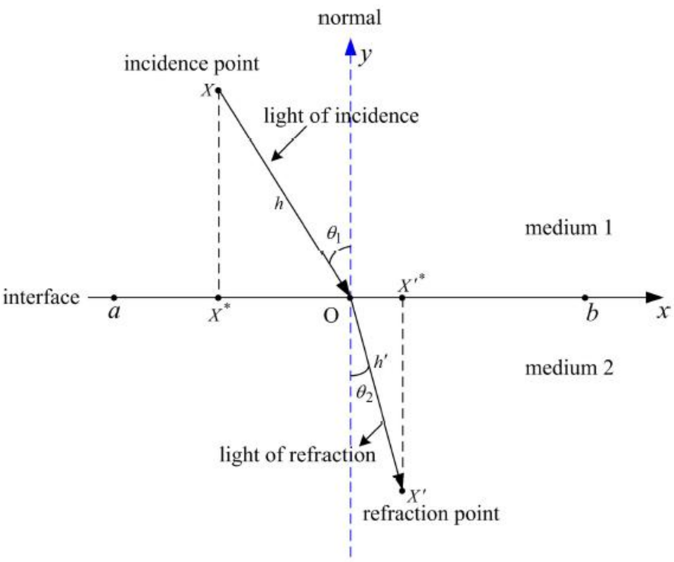


Fig 2.9 Principle of refraction of light.

- As the light ray changes the path after entering in the different medium that concept has used in this advancement as pushing the variable in the completely opposite direction from their current situation as shown in the figure above.
- This enhances its capacity to jump out of the local optima and search in wider search space.
- Moreover, it also helps to send the search agent in the exact opposite direction of current seach.
- It works on the mathematical formula given below.

$$X_j'^* = \left(\frac{a_j + b_j}{2}\right) + \left(\frac{a_j + b_j}{2kn}\right) - \left(\frac{X_j^*}{kn}\right) \quad (2.3.1)$$

Where X_j^* and $X_j'^*$ are the jtg dimention of X_j^* and $X_j'^*$, respectively, a_j and b_j are the jth dimension minimum and maximum values of decision variable

- In our case upper limit and the lower limit comes as the same so it can be seen that final equation turns out as below for ideal values of variables considered as

$$X'^* = a + b - X^* \quad (2.3.2)$$

2.4 Modified Grey Wolf Optimizer for Global Engineering Optimization [12]

- This paper discusses the improvement of the GWO by change in the iteration decay equation.

$$a = 2\left(1 - \frac{t}{T}\right) \longrightarrow a = 2\left(1 - \frac{t^2}{T^2}\right) \quad (2.4.1)$$

- As seen above it increases the degree of the decay equation which in turn allow us to search in search space more efficiently.

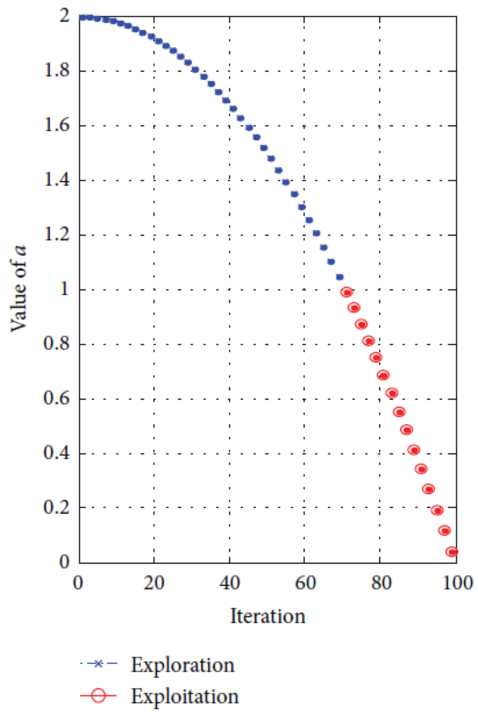


Fig. 2.10 mGWO graph

Fig 1.7 Graph above helps us to understand that by using the exponential decay function, the number of iterations used for exploration and exploitation is 70% and 30%.

- As we compare the results of this variation with the conventional GWO it looks like below:

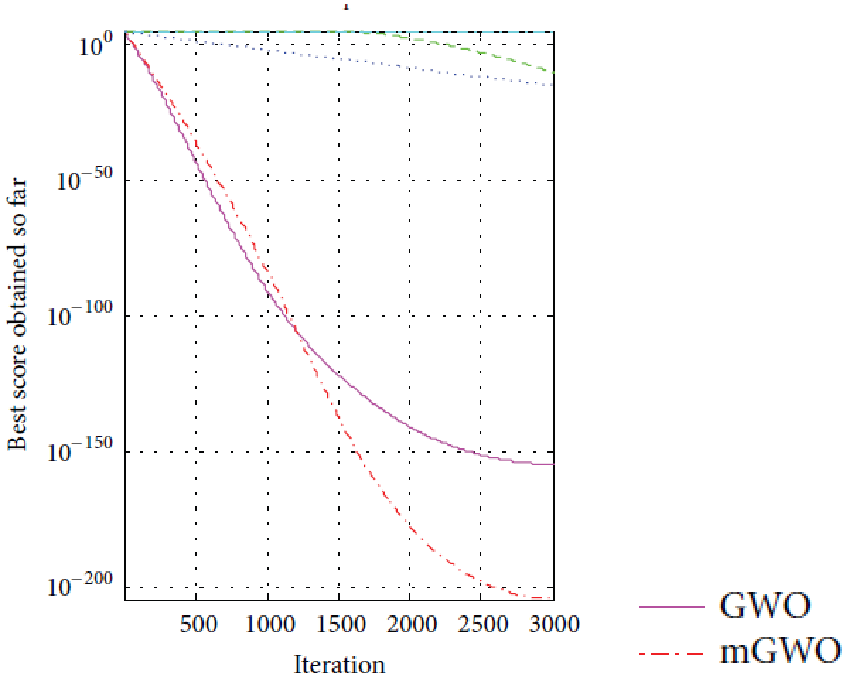


Fig 2.11 GWO vs mGWO

- As it can be seen in the figure it gives better results than the GWO in the long run.

2.5 Ours variations :

- As we can notice that all the improvement in the GWO has done in the distinct are of the possible changes but here we have tried to improvise the GWO by combining the two possible area of change in the GWO.

1. **mFLGWO** – Modified Fuzzy Logic GWO
    - Here we mixed up two already existence variables mentioned in the section "mGWO" and "FLGWO" in literature review of this report.
    - FLGWO solves the problem of credibility of the wolves over the iterations and the mGWO solves the problem of less exploration in the GWO.
2. **mRLGWO** – Modified Refraction Learning GWO
    - Here we mixed up two already existence variables mentioned in the section "mGWO" and "RLGWO" in literature review of this report.
    - RLGWO solves the problem of the local optima and the mGWO helps to improve exploration in the GWO

CHAPTER-3: IMPLEMENTATION

3.1 Architecture

The presented work intends to introduce a new framework for tagging sentiments for a target domain by labelled data from a source domain. This includes major phases like (i) pre-processing (ii) Construction of TF-IDF matrix (iii) Removal of High frequency removal words (iv) Train Classifier (v) Different Optimizer based classification (vi) Sentiment classification, illustrated in Figure below In this context, we explain the proposed sentiment classification process on considering restaurant review given by the customers.

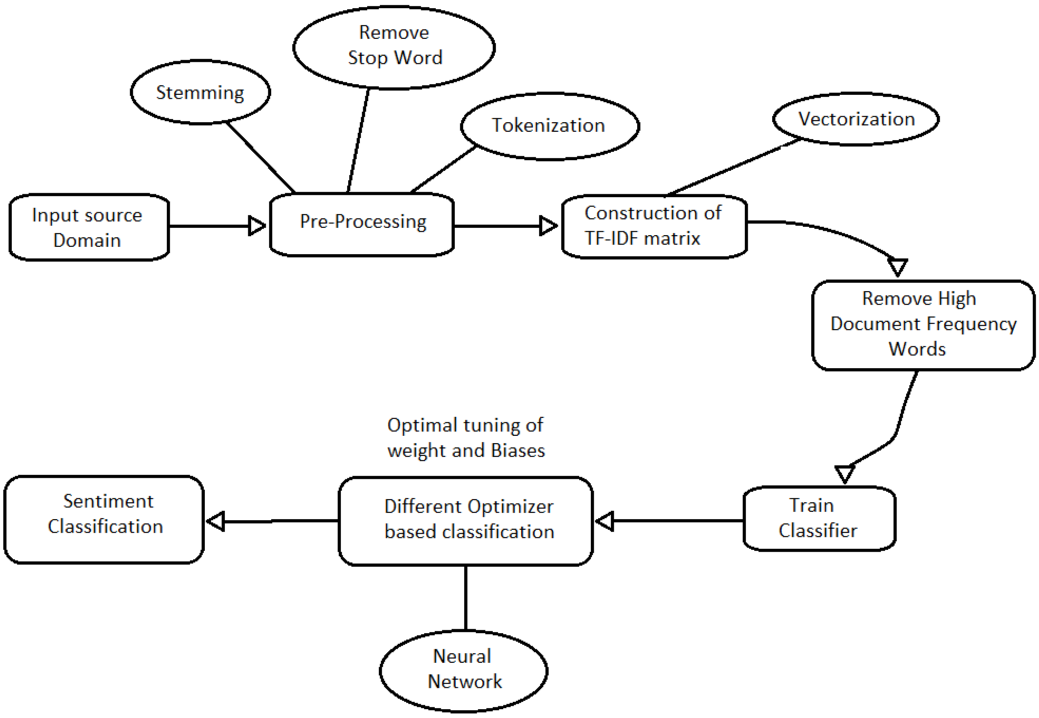


Fig. 3.1 Implementation of sentiment classification process

3.1.1 Pre-Processing

In this step we apply different pre-processing techniques in our data set

- Stemming: Stemming is the process of producing morphological variants of a root/base word. Stemming programs are commonly referred to as stemming algorithms or stemmers. A stemming algorithm reduces the words “chocolates”, "chocolatey", "choco" to the root word, "chocolate" and "retrieval", "retrieved", "retrieves" reduce to the stem "retrieve".

- Stop word removal: A stop word is a commonly used word (such as "the", "a", "an", "in") that a search engine has been programmed to ignore.
- Tokenization: Tokenization is the process of tokenizing or splitting a string, text into a list of tokens. One can think of token as parts like a word is a token in a sentence, and a sentence is a token in a paragraph.
  - For Example:-

review	Label
You Were born with Potential	0
You Were born with goodness and trust	0
You were born with ideals and dreams	0
You were born with greatness	0
You were born with wings	1
You are not meant for crawling, so do not	1
You have wings	0
learn to use them and fly	0

Table 3.1 Original Dataset

review	Label
you were born with potenti	0
you were born with good and trust	0
you were born with ideal and dream	0
you were born with great	0
you were born with wing	1
you are not meant for crawl so do not	1
you have wing	0
learn to use them and fli	0

Table 3.2 Dataset after Stemming and Removing Non-Ascii Character

review	Label
born potenti	0
born good trust	0
born ideal dream	0
born great	0
born wing	1
not meant crawl not	1
wing	0
use fli	0

Table 3.3 Dataset after Removing Stopwords

3.1.2 Construction of Tf-Idf Matrix:

- TF-IDF stands for Term Frequency Inverse Document Frequency of records. It can be defined as the calculation of how relevant a word in a series or corpus is to a text.

- Term Frequency: In document d, the frequency represents the number of instances of a given word t. Therefore, we can see that it becomes more relevant when a word appears in the text, which is rational. Since the ordering of terms is not significant, we can use a vector to describe the text in the bag of term models. For each specific term in the paper, there is an entry with the value being the term frequency.

$$Term\ Frequency(TF) = \frac{\text{Frequency of a term in the document}}{\text{Total number of terms in documents}} \quad (3.1.1)$$

- Document Frequency: This tests the meaning of the text, which is very similar to TF, in the whole corpus collection. The only difference is that in document d, TF is the frequency counter for a term t, while df is the number of occurrences in the document set N of the term t. In other words, the number of papers in which the word is present is DF.
- Inverse Document Frequency: Mainly, it tests how relevant the word is. The key aim of the search is to locate the appropriate records that fit the demand. Since tf considers all terms equally significant, it is therefore not only possible to use the term frequencies to measure the weight of the term in the paper.

$$Inverse\ Document\ Frequency(IDF) = \log\left(\frac{total\ number\ of\ documents}{number\ of\ documents\ with\ term\ t}\right) \quad (3.1.2)$$

For Example:

	0	1	2	3	4	5	6	7
born	0.489446651	0.368888	0.368888	0.489447	0.556451	0	0	0
crawl	0	0	0	0	0	0.5	0	0
don	0	0	0	0	0	0.5	0	0
dream	0	0	0.657237	0	0	0	0	0
fli	0	0	0	0	0	0	0	0.57735
good	0	0.657237	0	0	0	0	0	0
great	0	0	0	0.872033	0	0	0	0
ideal	0	0	0.657237	0	0	0	0	0
learn	0	0	0	0	0	0	0	0.57735
meant	0	0	0	0	0	0.5	0	0
not	0	0	0	0	0	0.5	0	0
potenti	0.872033242	0	0	0	0	0	0	0
trust	0	0.657237	0	0	0	0	0	0
use	0	0	0	0	0	0	0	0.57735
wing	0	0	0	0	0.830881	0	1	0

Table 3.4 Tokenization and Tf-Idf Vectorization Process

3.1.3 Removal of High Document Frequency Words:

In this step we remove some words that are repeated many times in dataset because it considered as a stop words and it is not important.

3.1.4 Train the Classifier:

In this step we apply a artificial neural network method called Artificial Neural Network.

- **Artificial Neural Network:** Artificial Neural Networks contain artificial neurons which are called units. These units are arranged in a series of layers that together constitute the whole Artificial Neural Networks in a system. A layer can have only a dozen units or millions of units as this depend on the complexity of the system. Commonly, Artificial Neural Network has an input layer, output layer as well as hidden layers. The input layer receives data from the outside world which the neural network needs to analyze or learn about. Then this data passes through one or multiple hidden layers that transform the input into data that is valuable for the output layer. Finally, the output layer provides an output in the form of a response of the Artificial Neural Networks to input data provided. In the majority of neural networks, units are interconnected from one layer to another. Each of these connections has weights that determine the influence of one unit on another unit.

As the data transfers from one unit to another, the neural network learns more and more about the data which eventually results in an output from the output layer.

- Optimized NN is used for sentiment classification.As the work concerns on three source domains, thekeywords of those domains are trained using NNmodel. Here, the extracted keywords (words) ofdomains denoted by w are subjected to NNclassification. NN considers the words w as inputspecified by Eq. (3.1.3), where nu signifies the total countof words.

$$w = \{w_1, w_2, \ldots, w_n\} \quad (3.1.3)$$

- The model includes input, output, and hiddenlayers. The output of the hidden layer $e^{(H)}$ is defined as in Eq. (3.1.4), where F refers to the “activation function”, $\hat{\imath}$ and j refers to the neurons of hidden and input layers correspondingly, $W^{(H)}_{(B\hat{\imath})}$ denotes bias weight to $\hat{\imath}^{th}$ hidden neuron, n_i symbolizes count of input neurons and $W^{(H)}_{(J\hat{\imath})}$ denotes the weight from j^{th} input neuron to $\hat{\imath}^{th}$ hidden neuron. The output of the network $\widehat{G_0}$ is determined as in Eq. (3.1.5), where ô refers to the output neurons, n_h indicates the number of hidden neurons, $W^{(G)}_{(B\hat{o})}$ denotes output bias weight to the $\hat{o}^{th}$ output layer, and $W^{(G)}_{(\hat{\imath}\hat{o})}$ specifies the weight from $\hat{\imath}^{th}$ hidden layer to $\hat{o}^{th}$ output layer. Consequently, the error amongst the predicted and actual values is computed as per Eq. (6) that should be reduced. In Eq. (3.1.6), n_G symbolizes the output neuron count, $G_{\hat{o}}$ and $\widehat{G_{\hat{o}}}$ refers to the actual and predicted output respectively.

$$e^{(H)} = F\left(W^{(H)}_{(B\hat{\imath})} + \sum_{j=1}^{n_{\hat{\imath}}} W^{(H)}_{(J\hat{\imath})} w\right) \quad (3.1.4)$$

$$\hat{G}_{\hat{o}} = F\left(W^{(G)}_{(B\hat{o})} + \sum_{j=1}^{n_h} W^{(G)}_{(\hat{\imath}\hat{o})} e^{(H)}\right) \quad (3.1.5)$$

$$Er^* = argmin \sum_{\left\{W^{(H)}_{(B\hat{\imath})}, W^{(H)}_{(J\hat{\imath})}, W^{(G)}_{(B\hat{o})}, W^{(G)}_{(\hat{\imath}\hat{o})}\right\}=1}^{n_G} \left|G_{\hat{o}} - \hat{G}_{\hat{o}}\right| \quad (3.1.6)$$

As mentioned above, the training of NN model iscarried out using a new IGWO algorithm viaoptimizing the weights W=$W^{(H)}_{(B\hat{\imath})}$, $W^{(H)}_{(J\hat{\imath})}$, $W^{(G)}_{(B\hat{o})}$ and $W^{(G)}_{(\hat{\imath}\hat{o})}$. Thus, the sentiments of words are attained as outputs. The objective function OF of the presented work is defined in Eq. (3.1.7)

$$OF = Min(Er^*) \quad (3.1.7)$$

Above mentioned equations is our key to attach different optimizers with the ANN. Here we are meant to optimize the weight and biases contained by the error equation in order to minimize the error and improve the model accuracy. Different optimizers we have applied discussed in next section.

3.1.5 Different Optimizer based Classification:

In this stage we apply different types of Optimizers in our Ann model. Optimizers that uses in this project are follows:

- **SGD:** The word 'stochastic' means a system or a process that is linked with a random probability. Hence, in Stochastic Gradient Descent, a few samples are selected randomly instead of the whole data set for each iteration. In Gradient Descent, there is a term called "batch" which denotes the total number of samples from a dataset that is used for calculating the gradient for each iteration. In typical Gradient Descent optimization, like Batch Gradient Descent, the batch is taken to be the whole dataset. Although, using the whole dataset is really useful for getting to the minima in a less noisy and less random manner, but the problem arises when our datasets get big.
- **RMSPROP:** RMSprop stands for Root Mean Square Propagation. RMSprop optimizer doesn't let gradients accumulate for momentum instead only accumulates gradients in a particular fixed window. It can be considered as an updated version of AdaGrad with few improvements. RMSprop uses simple momentum instead of Nesterov momentum.
- **ADAM:** Adaptive Moment Estimation (Adam) is among the top-most optimization techniques used today. In this method, the adaptive learning rate for each parameter is calculated. This method combines advantages of both RMSprop and momentum .i.e. stores decaying average of previous gradients and previously squared gradients.
- **NDAM:** NAdam is a short form for Nesterov and Adam optimizer. NAdam uses Nesterov momentum to update gradient than vanilla momentum used by Adam.
- **ADAGRAD:** Adaptive Gradient Algorithm (Adagrad) is an algorithm for gradient-based optimization. The learning rate is adapted

component-wise to the parameters by incorporating knowledge of past observations.

- **ADAMAX:** AdaMax is an alteration of the Adam optimizer. It is built on the adaptive approximation of low-order moments (based off on infinity norm). Sometimes in the case of embeddings, AdaMax is considered better than Adam.
- **ADADELTA:** Adaptive Delta (Adadelta) optimizer is an extension of AdaGrad (similar to RMSprop optimizer), however, Adadelta discarded the use of learning rate by replacing it with an exponential moving mean of squared delta (difference between current and updated weights). It also tries to eliminate the decaying learning rate problem.
- **FTRL:** Follow The Regularized Leader (FTRL) is an optimization algorithm best suited for shallow models having sparse and large feature spaces. This version supports both shrinkage-type L2 regularization (summation of L2 penalty and loss function) and online L2 regularization.
- **Grey Wolf Optimizer (GWO):** The GWO algorithm mimics the leadership hierarchy and hunting mechanism of grey wolves in nature. Four types of grey wolves such as alpha, beta, delta, and omega are employed for simulating the leadership hierarchy. In addition, the three main steps of hunting, searching for prey, encircling prey, and attacking prey, are implemented.
- **mGWO:** modified GWO is an extension of GWO, which focuses on proper balance between exploration and exploitation that leads to an optimal performance of the algorithm.
- **RLGWO:** Refraction Learning GWO is an extension of GWO. In this a new operator, called refraction learning, is essentially an opposite-learning strategy that is inspired by the principle of light refraction in physics. This proposed operator is applied to the current global optima of the swarm in theGWO algorithm and is benecial to help the population for jumping out of the local optima.

- **FLGWO:** This optimizer is based on the popular artificial intelligence technique known as “Fuzzy Logic” and its application in this optimization algorithm of GWO.

CHAPTER-4 RESULTS

The Sentiment classification proposed in the architecture has implemented in the python language by using bunch of libraries shown in the tools and technology part. Here, implementation has done on the restaurant database of restaurant feedback given by the customers available at "https://www.kaggle.com/datasets/shub99/sentiment-analysis-data". As per objective we have successfully improved the test accuracy of the classification model by 3% to 15% compared to the conventional methods available on Keras [12] and train accuracy has improved around 1% to 6% for the same.

4.1 Environment

All the optimizers have tested in the same environment specified as below:

- No. of Input = 1254
- No. of Output = 1
- No. of hidden layers = 1
- No. of neurons in hidden layer = 10
- Error function = Mean Square error
- Activation function = RELU, Tanh
- Weight and Bias Initializer = He Uniform

4.2 Evaluation measures

<table>
<tr><td colspan="2"></td><td colspan="2">Actual labels</td></tr>
<tr><td colspan="2"></td><td>Positive</td><td>Negative</td></tr>
<tr><td rowspan="2">Predicated</td><td>Positive</td><td>TP</td><td>FP</td></tr>
<tr><td>Negative</td><td>FN</td><td>TN</td></tr>
</table>

Fig. 4.1 Confusion Matrix

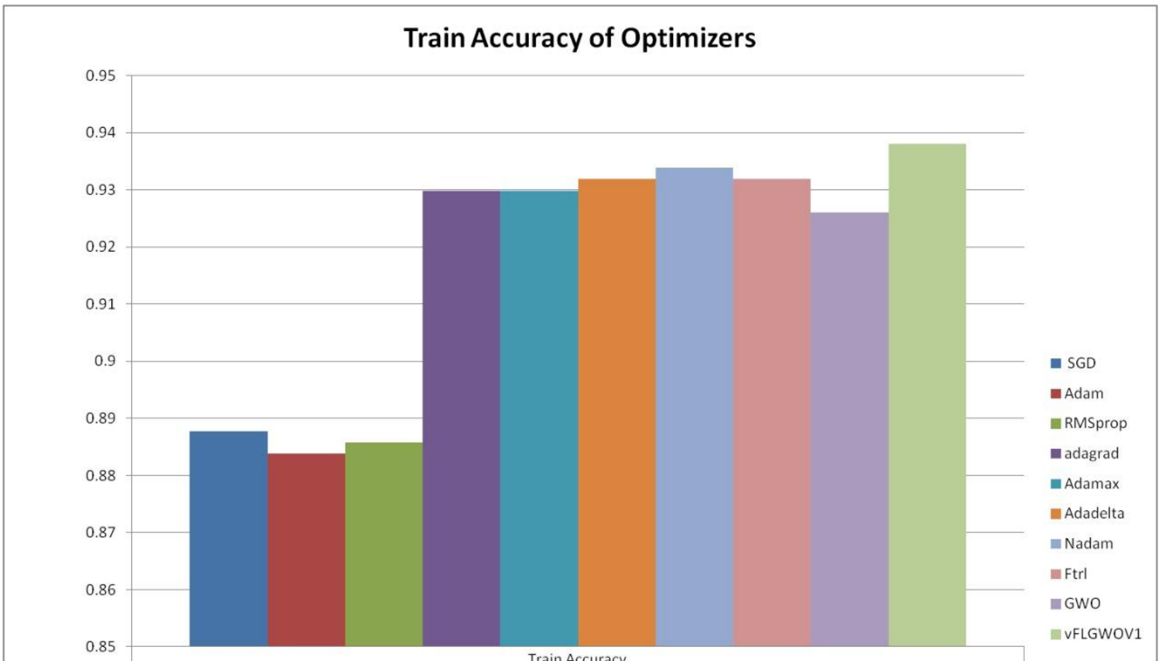
Precision is a ration of correctly identified positive labels to all predicted positive labels. So Precision= (TP/(TP+FP)). Recall is a ratio of correctly identified positive labels to actual positive labels that is Recall=(TP/(TP+FN). F-measure is a harmonic mean of precision and recall. Which is formulated as F-measure= 2 x (Precision x Recall)/(Precision + Recall).

Specificity is a ration of correctly identified negative labels to actual negative labels present in dataset that is Specificity = (TN/(TP+FP)). Accuracy is a portion of correctly predicted labels to all predicted labels which is given as Accuracy = (TP/(TP+TN+FP+FN)). Negative Predicted Value (NPV) is a probability that how negative test prediction is accurate which is given by NPV=(TN/(TN+FN)).

4.3 Outcomes

Graphs below shows the Training and Testing accuracy for sentiment analysis using different optimizers listed in section

As seen in the figures below we have successfully managed to improve the training and testing accuracy of the model by 1% to 6% and 3 % to 15% respectively.



	Train Accuracy
SGD	0.887775551
Adam	0.883767535
RMSprop	0.885771543
adagrad	0.929859719
Adamax	0.929859719
Adadelta	0.931863727
Nadam	0.933867735
Ftrl	0.931863727
GWO	0.926
vFLGWOV1	0.938

Fig. 4.2 Training Accuracy of Different Optimizers

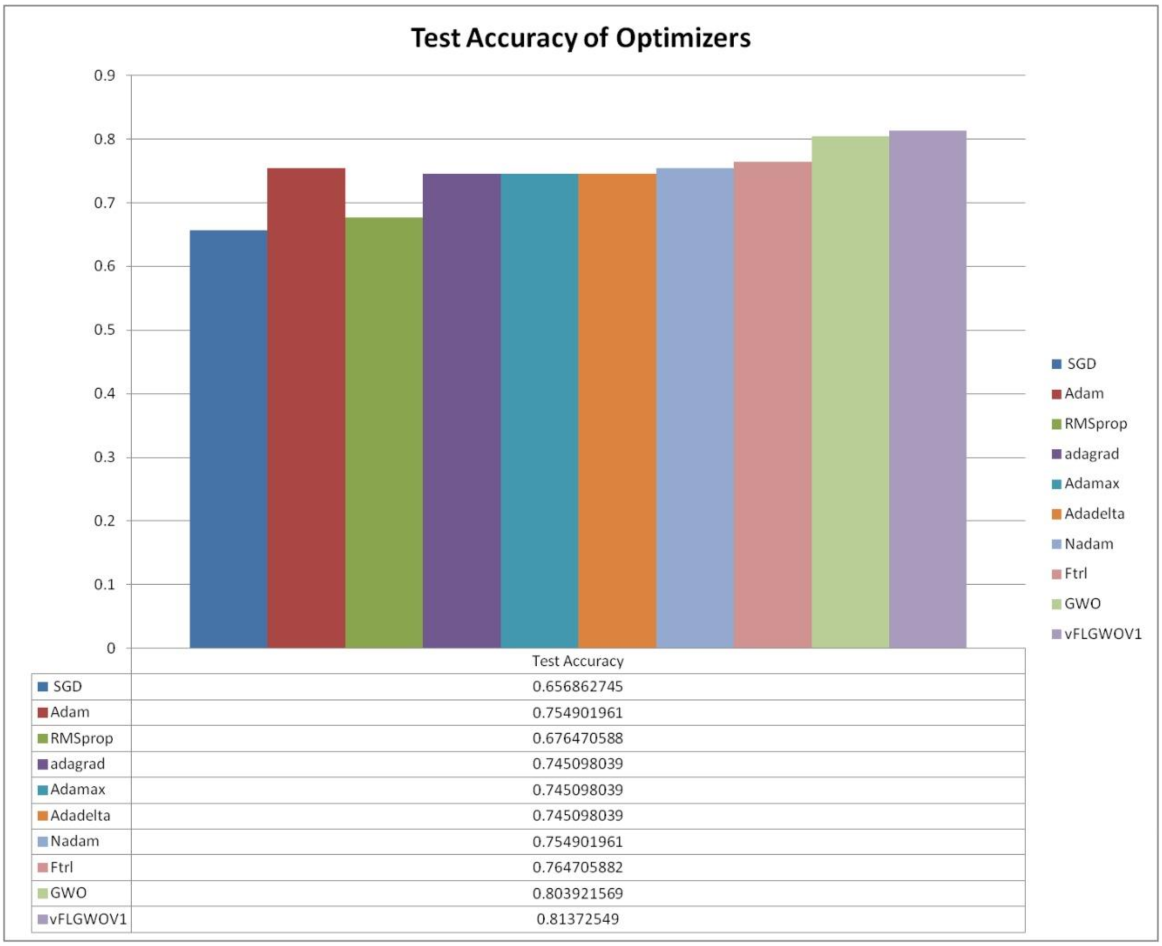


Fig. 4.3 Test Accuracy of Different Optimizers

4.4 Benchmark Function

We have applied our different variations, mentioned below, of the GWO on different benchmark function and we got the following results.

1. GWO
2. mGWO
3. RLGWO
4. mRLGWO
5. FLGWO
6. mFLGWO

4.4.1 Elliptic Benchmark Function

- As it can seen in the fig 4.4, for multimodal Elliptic benchmark function all the variations including our gives very much optimal answer for the function. All the the functions gives the same answer as it can be observed in the figure.

Elliptic	$f_{11}(x) = \sum_{i=1}^{n} (10^6)^{(i-1)/(n-1)} x_i^2$	[-100,100]

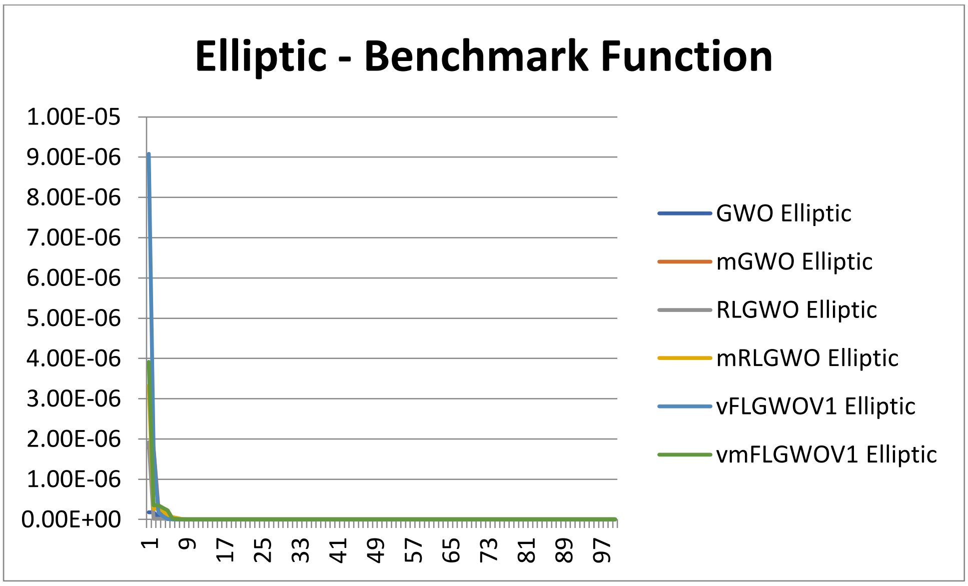


Fig. 4.4 Elliptic Benchmark Function

4.4.2 Salomon Benchmark Function

- As it can seen in the fig 4.5, for multimodal Salomon benchmark function all the variations including our gives optimal answers for following function. However, among all the variations FLGWO gives the fastest answer but not much optimal. Moreover, mFLGWO, RLGWO tend give most optimal solution among others.

Salomon $f_{21}(x) = 1 - \cos\left(2\pi\sqrt{\sum_{i=1}^{n} x_i^2}\right) + 0.1\sqrt{\sum_{i=1}^{n} x_i^2}$ [-100,100]

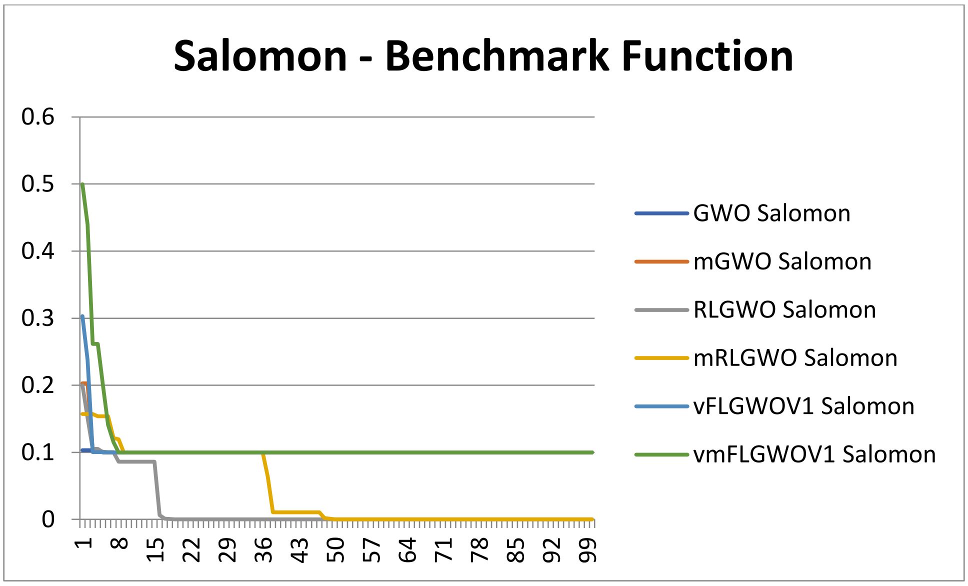


Fig. 4.5 Salomon Benchmark Function

4.4.3 Alpine Benchmark Function

- As it can seen in the fig 4.6, for multimodal Alpine benchmark function all the variations including our gives very much optimal answer for the function. All the the functions gives the same answer as it can be observed in the figure. However, GWO gives the fastest solution and mGWO gives slowest solution. Moreover, mRLGWO gives the most accurate solution.

Alpine $f_{16}(x)=\sum_{i=1}^{n}\left|x_i \cdot \sin(x_i)+0.1 \cdot x_i\right|$ [-10,10]

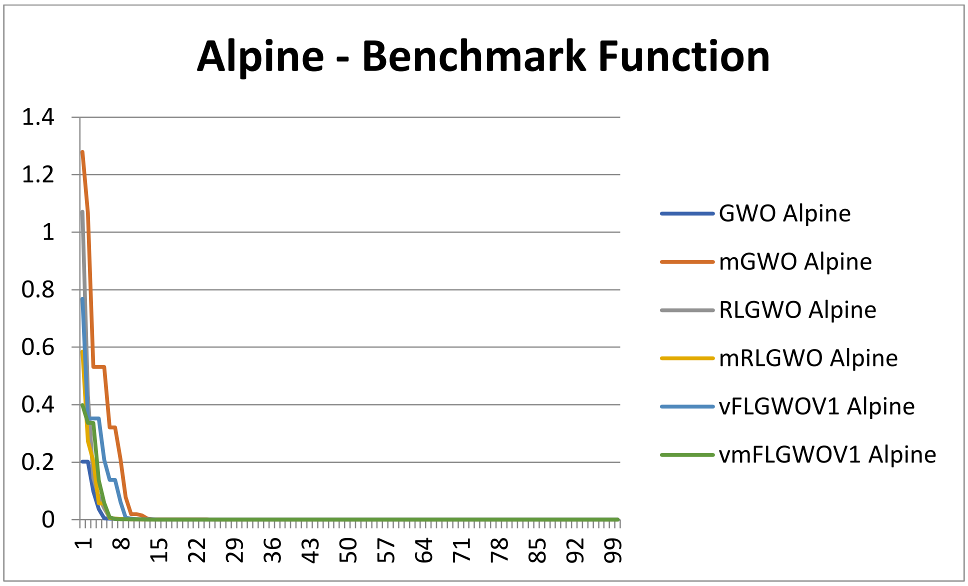
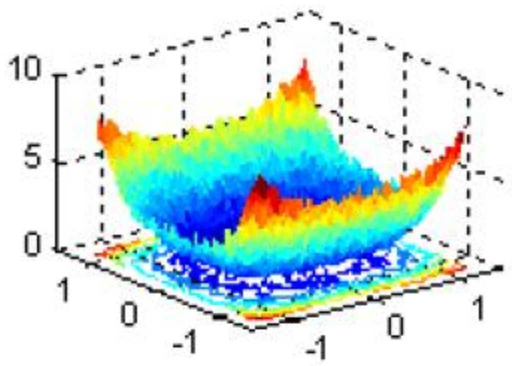


Fig. 4.6 Alpine Benchmark Function

4.4.4 Sum Square Benchmark Function

- As it can seen in the fig 4.7, for unimodel Elliptic benchmark function all the variations including our gives very much optimal answer for the function. All the the functions gives the same answer as it can be observed in the figure. Apparently, it can be seen that FLGWO gives the fastest answer and RLGWO gives slowest solution. At the long run FLGWO gives the most accurate solution.

Sum-Square $\quad f_7(x) = \sum_{i=1}^{n} ix_i^2 \quad$ [-10,10]



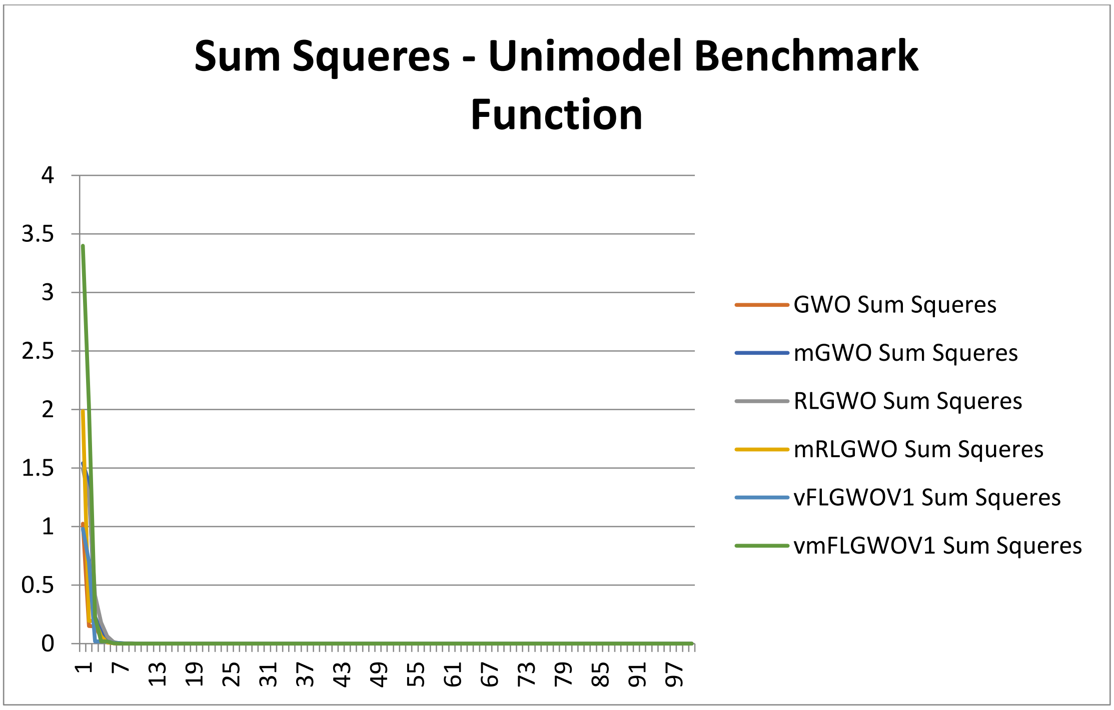
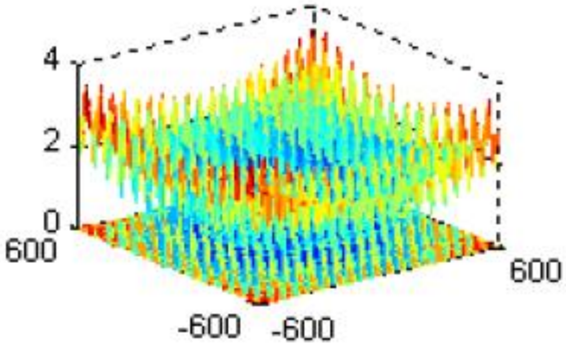


Fig. 4.7 Sum Square Benchmark Function

4.4.5 Griewank Benchmark Function

- As it can seen in the fig 4.8, for multimodal Griewank benchmark function all the variations including our gives very much optimal answer for the function. RLGWO, mRLGWO, mFLGWO gives most accurate solutions and mFLGWO gives the fastest answer and GWO gives the slowest answer.

Griewank $\quad f_{14}(x) = \frac{1}{4000}\sum_{i=1}^{n} x_i^2 - \prod_{i=1}^{n} \cos\left(\frac{x_i}{\sqrt{i}}\right) + 1 \quad$ [-600,600]



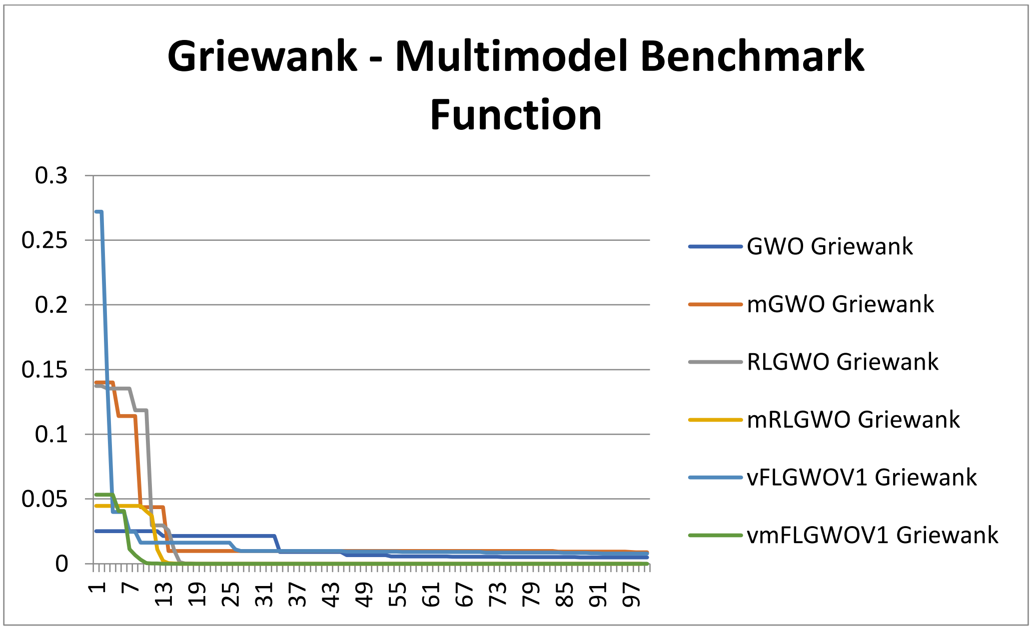
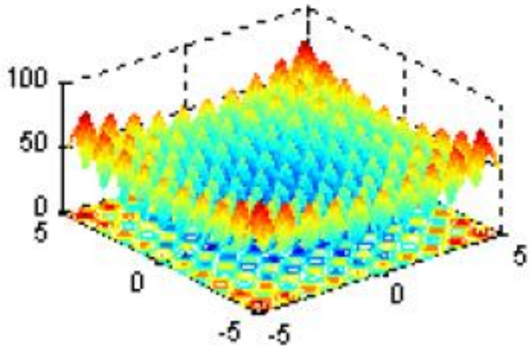


Fig. 4.8 Griewank Benchmark Function

4.4.6 Rastrigin Benchmark Function

- As it can seen in the fig 4.9, for multimodel Rastrigin benchmark function all the variations including our gives very much optimal answer for the function. All the the functions gives the same answer as it can be observed in the figure. As it can be seen that FLGWO gives fastest and mGWO gives the slowest solution.

Rastrigin $f_{12}(x) = \sum_{i=1}^{n} [x_i^2 - 10\cos(2\pi x_i) + 10]$ [-5.12,5.12]



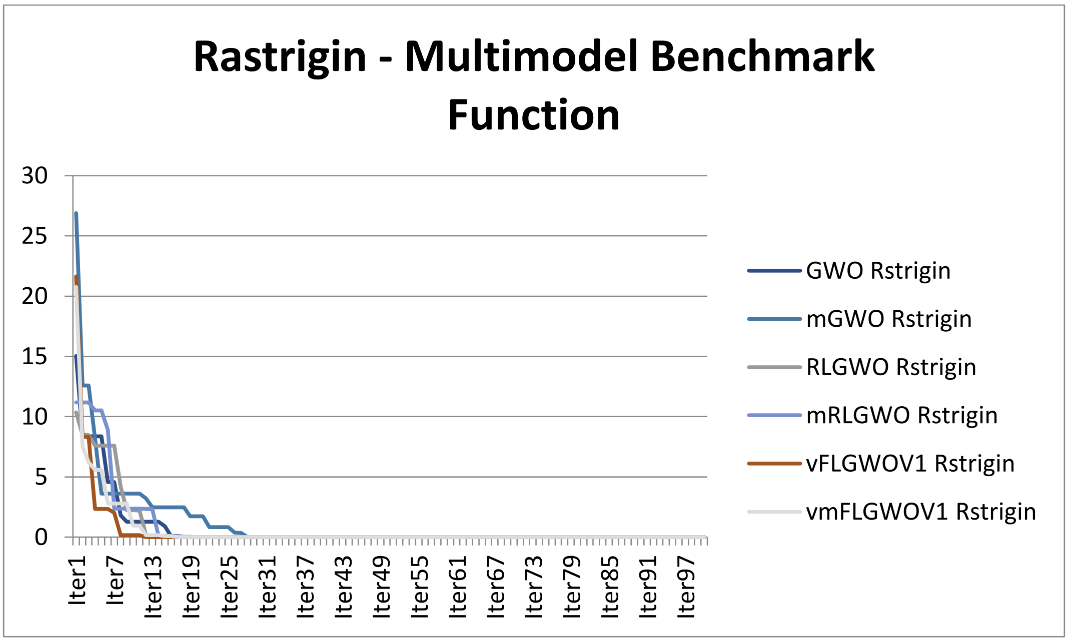


Fig. 4.9 Rastrigin Benchmark Function

4.4.7 Sphere Benchmark Function

- As it can seen in the fig 4.10, for unimodel Sphere benchmark function all the variations including our gives very much optimal answer for the function. All the the functions gives the same answer as it can be observed in the figure.

Sphere $f_1(x) = \sum_{i=1}^{n} x_i^2$ [-100,100]

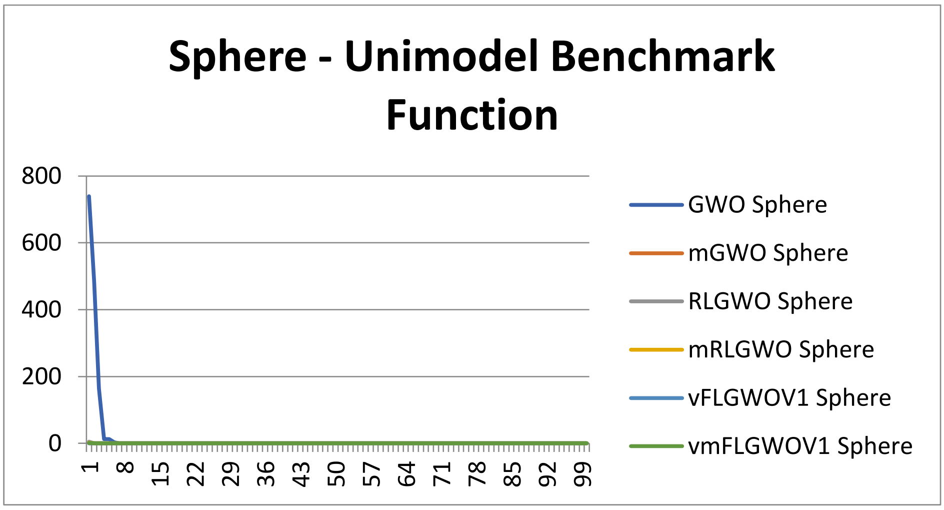


Fig. 4.10 Sphere Benchmark Function

4.4.8 Inverted Cosine Benchmark Function

- As it can seen in the fig 4.11, for multimodel Elliptic benchmark function all the variations including our gives very much optimal answer for the function. All the the functions gives the same answer as it can be observed in the figure. From the graph it can be stated that fastest and slowest solutions given by GWO and mFLGWO respectively.

Inverted Cosine $f_{15}(x) = 0.1n - \left(0.1\sum_{i=1}^{n} \cos(5\pi x_i) - \sum_{i=1}^{n} x_i^2\right)$ [-1,1]

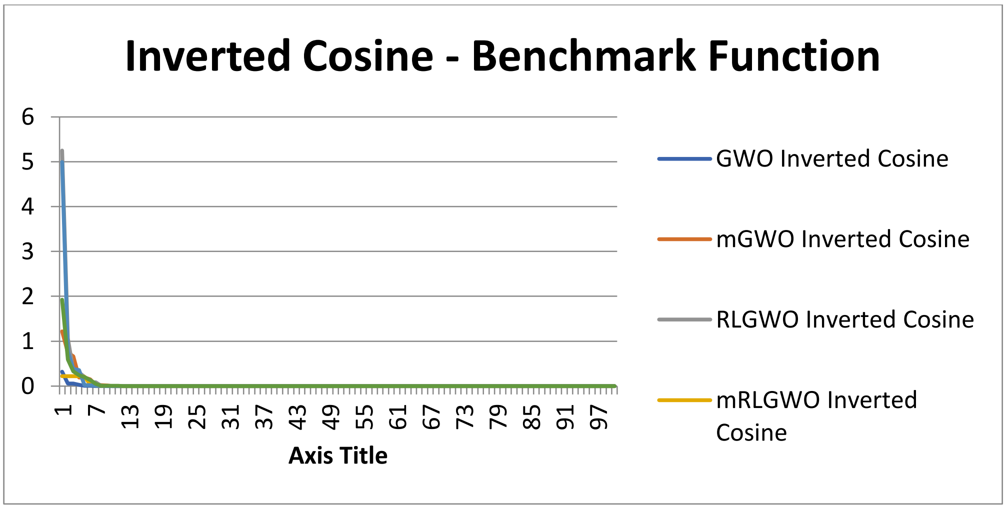


Fig. 4.11 Inverted Cosine Benchmark Function

Table 4.1 Benchmark Function Table

Index	Benchmark Function	GWO	mGWO	RLGWO	mRLGWO	vFLGWO	vmFLGWO
1	Rstrigin	0	0	0	0	0	0
2	Sphere	1.78E-39	1.36E-52	3.15E-40	1.47E-51	1.90E-51	1.21E-57
3	Inverted Cosine	0	0	0	0	0	0
4	Elliptic	3.88E-53	5.48E-57	5.74E-49	5.47E-61	3.38E-61	1.79E-61
5	Salomon	0.099873	0.099874	2.30E-21	5.30E-15	0.099873	0.099873
6	Alpine	2.24E-05	9.88E-16	1.04E-23	1.15E-29	1.66E-05	1.11E-28
7	Sum Squeres	4.13E-46	1.29E-52	6.10E-43	3.79E-50	5.97E-56	4.81E-54
8	Griewank	0.00494	0.008931	0	0	0.007862	0

As shown in the table different version of the GWO works better than the GWO on the various benchmark functions.

4.5 Different variations on the sentiment analysis:

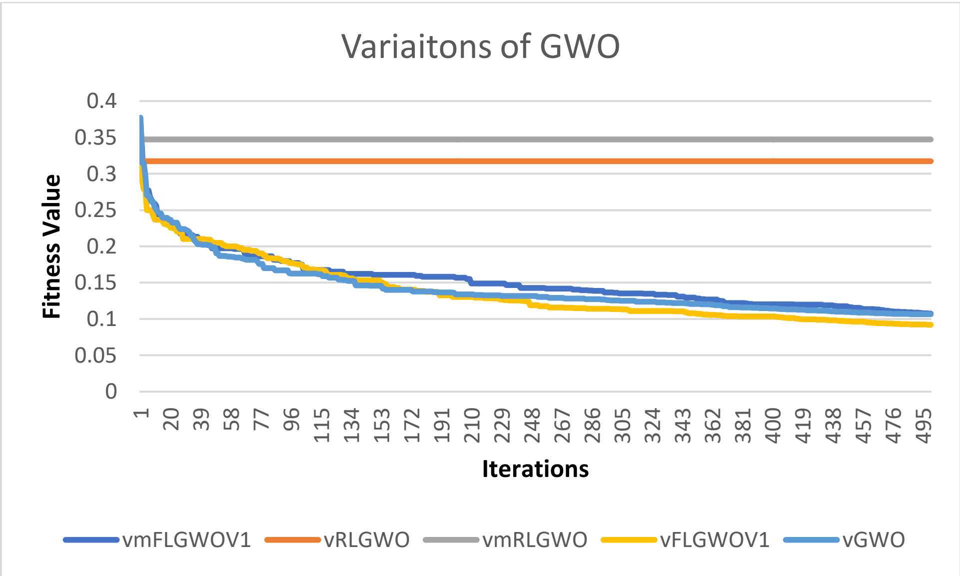


Fig. 4.12 Variation of GWO

- As it can be seen in the graph above we can see that amongst all the variations of the GWO, FLGWO gives the highest accuracy for sentiment analysis amongst all including our own variations.

CONCLUSION

We have seen that Sentiment Analysis can be used for analyzing opinions in blogs, articles, Product reviews, Social Media websites, Movie-review websites where a third person narrates his views. We also studied NLP and Machine Learning approaches for Sentiment Analysis. We have seen that sentiment analysis has many applications and it is important field to study. Sentiment analysis has Strong commercial interest because Companies want to know how their products are being perceived and also Prospective consumers want to know what existing users think.

Throughout this project we have successfully managed to improve the testing accuracy and training accuracy of the sentiment analysis model by the 3% to 15% and 1% to 6% respectively. In process of improving the sentiment analysis' model's accuracy we have tried various variations of the GWO on the ANN as well as the Benchmark functions and so far we have figured it out that existing variation of GWO known as the FL-GWO (Fuzzy Logic - GWO) enhances the model accuracy by greater margins than GWO.

After applying so many variations and so many optimizers including most popular once gives less accuracy than GWO and FLGWO. So far GWO's fuzzy logic variation is most accurate and gives highest accuracy for sentiment analysis.

REFERENCES

[1] https://youtu.be/aircAruvnKk

[2] https://seyedalimirjalili.com/gwo

[3] https://youtube.com/playlist?list=PLZoTAELRMXVPGU70ZGsckrMdr0FteeRUi

[4] https://www.sciencedirect.com/science/article/pii/S1018363916300071

[5] Comparison of Sentiment Analysis and Domain
Adaptation Techniques with Research Scopes, IEEE Xplore Part Number: CFP21F70-ART; ISBN: 978-1-7281-8501-9

[6] Multi-Source Domain Adaptation in Sentiment Analysis using Optimized Neural Network and Cross-Domain Semantic Library, https://inass.org/wp-content/uploads/2021/08/2021103147.pdf

[7] Final GWO - https://seyedalimirjalili.com/gwo

[8] Review of the different enhancements of GWO - https://ieeexplore.ieee.org/document/8928579

[9] A Fuzzy Hierarchical Operator in the Grey Wolf Optimizer Algorithm - https://www.sciencedirect.com/science/article/abs/pii/S1568494617301825

[10] A Novel Grey Wolf Optimizer Algorithm With Refraction Learning - https://ieeexplore.ieee.org/document/8689025

[11] Modified Grey Wolf Optimizer for Global Engineering Optimization - https://www.hindawi.com/journals/acisc/2016/7950348/

[12] Keras Weights and Bias optimizers - https://keras.io/api/optimizers/