CHAPTER 4

GBDT-BASED SENTIMENT CLASSIFICATION

4.1 INTRODUCTION

People communicate their perspectives, arguments, and feelings about their daily life on Social Media (SM) platforms (e.g., Twitter and Facebook). Twitter stands as a global micro-blogging service which comes with a brief message known as Tweets. Incorrect Grammar sentence, Freestyle writing, abbreviations and typographical errors are some noise that occur in twitter text. Sentiment Analysis based on a tweet published by the consumer, and Opinion Mining (OM) of their client testimonials is another famed research topic. The texts have been gathered from individual tweets by way of Opinion Mining and automatic-Sentimental Analysis based on ternary categories, namely Positive, Neutral, and Negative. It's quite challenging for the researchers to determine thoughts as a consequence of its limited dimensions, misspells, abbreviations, and slangs for Twitter data. To understand the development in Tree based sentiment Classification, a sentimental analysis and its classification task in twitter analysis is analyzed. They are used to reduce enormous and crucial work assignment. Additionally, sentimental analysis have most issues in feature selection and classification in twitter. To end this, a classification framework, gradient boosted decision tree-based sentiment is introduced for twitter data which recognizes oppressive feature subsets utilizing a gradient boosted decision normalization. In this Chapter, with the assistance of this Gradient Boosted Decision Tree classifier (GBDT), suggests that an efficient SA and Sentiment Classification (SC) of all Twitter data. At first, the

Twitter Data expands pre-processing. Then, the pre-processed information is processed with HDFS MapReduce. Now, the Features are extracted in the processed information, and then effective features are chosen using the Improved Elephant Herd Optimization (I-EHO) technique. Score values have been calculated for every one of these selected features and given into the classifier. Experiential outcomes are analyzed and contrasted with another traditional strategies to demonstrate the highest performance of the proposed method.

With the help of GBDT algorithm, Twitter data in this study suggests an efficient SA and SC. In order to get Twitter data, the processing is done ahead of time. The data is carried out using HDFS Map reduce. Here, the features processed data and then I-EHO has been selected by using techniques that can enhance efficient operation**.** The score value is calculated and categorized for each of those selected features. Finally, the GBDT classifier classification data has classified features as negative, positive, or neutral. Empirically distinguished methodology with regard to the prediction and use of another conventional techniques for better performance is introduced.

4.2 CLASSIFICATION BASED DECISION TREE

Sentiment analysis is a Place of Natural Language Processing refers to an opinion or attitude expressed by a person towards their goal. SA is process of collecting and analyzing information based upon the individual opinion, views, and thoughts. SA is done with the help of Natural Language Processing (NLP), Statistical models and different machine learning algorithms for extracting features from large data. The search for an understanding of common approach for making a good decision can improve their knowledge and

communication level at a point where the importance of sentimental term future selection and classification is enhanced through gradient features. Now-a-days, Twitter is boom globally over the Internet, growing quickly. Such a vast quantity is close to the SA. Meeting organizations are once in the area to track the views of users who are being viewed.

To optimize and make decisions classifies to categorize the tags the gradient booster with help of sentimental terms by prefers search option. First, Twitter starts implementing data. During pre-processing, Twitter data is Tokenization, Stop words Removal, Filtering, Lemmatization, Hashtag removal and Multiword grouping. Filter. The data that starts with the functionality is made using the HDFS Map Reduce. By the mode, this function is normal emoji count, pictogram points, symbols representation like question, Similes etc., positive trace count, negative trace word count, thoughts, unigram, engram, two, etc.

Efficient functions are selected using I-EHO technology. As the scores are calculated by classifying all of those selected features and input into the value classifier. In conclusion, the data, positive or negative, classified as positive is neutralized by GBDT. In addition, the proposed framework is illustrated in detail using Figure 4.1.

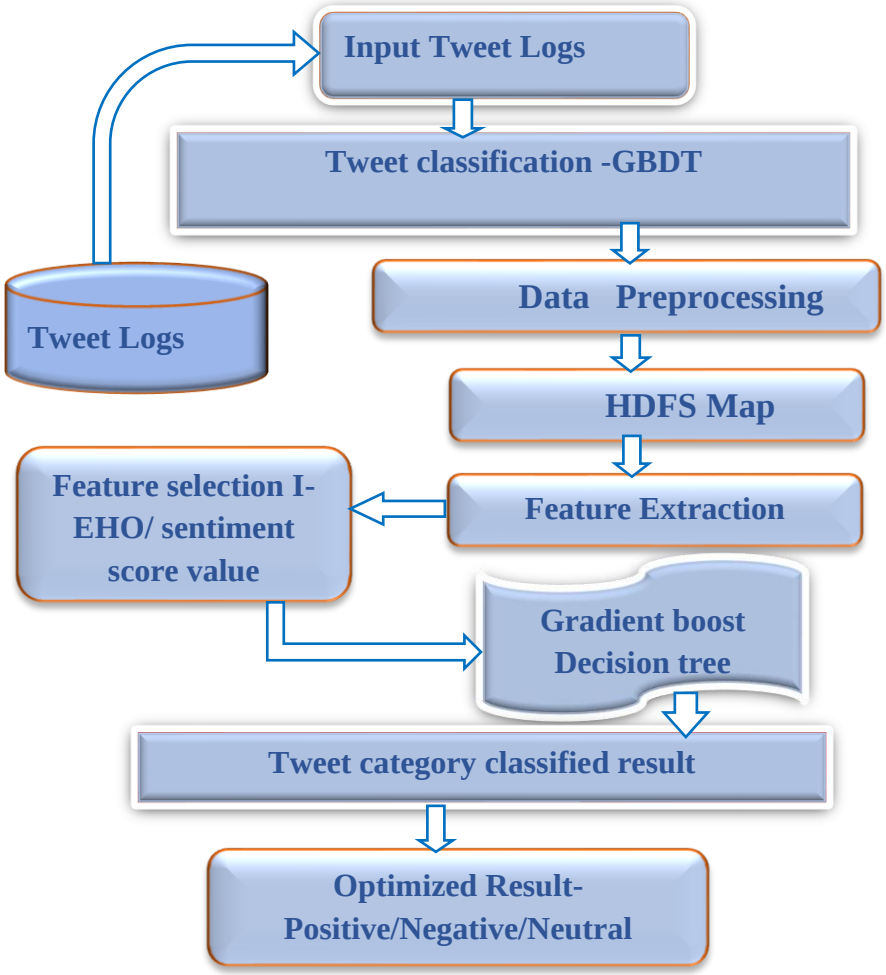

Figure 4.1 Proposed GBDT Framework

4.2.1 Pre-Processing

The primary step in the proposed system is the preprocessing of the dataset. Analysis of data is done so that no misleading results are obtained. The Twitter dataset comprises numerous different tweets. Such tweets might comprise words and symbols. The aim of the proposed system is to distinguish betwixt the symbols with and without emotions. In the preprocessing stage, the Twitter data is subjected to a) tokenization, b) stop-words removal, c) filtering, d) lemmatization, e) hashtag removal and f) multi-word grouping.

4.2.1.1 Tokenization

It's the Way of dividing the strings into words, Symbols, phrases, key words, and other special elements termed tokens. Tokens could function as phrases, individual words, as well as the whole sentences. Certain characters (i.e. punctuation marks) are abandoned during tokenization process. These are ways to split an array of tweet tags as key words which represent emoji's, keys, words, phrases.

4.2.1.2 Removal of stop words

The utmost commonly seen Stop Word that are often encountered in the database are "a", "and", "but", "how", or", "what" etc. These words evade the words from getting indexed. In the proposed Sentimental Analysis of twitter data, the Stop Words do not comprise any information related to emotions. Therefore, the stop words must be removed. The proposed system is programmed to disregard these SWs

4.2.1.3 Filtering

There's no Requirement for Tweeters to compose their status upgrades officially. Character elongations or reproduction is not uncommon in tweets. Largely such words may be a powerful sentiment indicator. An important thing would be to eliminate these repeated characters so as to acquire a normal meaningful word. A purposeful word is achieved by filtering redundant letters by one word. For example, the sentence 'Sooooo Saddddd' is transmuted to 'So Sad' during filtering.

4.2.1.4 Lemmatization

Types of a word so that they might be analyzed as one thing, recognized from the term's lemma or even a dictionary type. It's used to receive

a valid purposeful root term. Ordinarily, Lemma is the main term. Unlike coming, lemmatization will finish with a legitimate root term.

4.2.1.5 Hash tag - removal

Here, all hash-tags with a number-sign “(#)" in front of the words or un-spaced phrases are eliminated as of a given text.

4.2.1.6 Grouping of multiword

Here, a purposeful Token is achieved by grouping the words that are resultant achieved as of the prior measures. This previous preprocessing method includes grouping sequential tokens together to one token provided that it's found in a specific record. Here, the resultant grouping of the resulting words, such as in the previous step of object embedded tokens, will also be successful.

4.2.2 Data Processing

After pre-processing, these words will be arranged in structural processing by processing their data. Pre determinant process will be used to reduce the Providence HDFS graph

4.3 HADOOP DISTRIBUTED FILE SYSTEM (HDFS)

Hadoop Distributed File System (HDFS) and Apache Hadoop Map Reduce are taken from the Google’s MapReduce and Google File System (GFS). HDFS proffers a trusted method of managing pools of Big Data and bolsters the fastest data transport betwixt nodes. MapReduce suggests a programmatic layout for information processing. In the proposed work, MapReduce eliminates the redundant sensor information in regard of time.

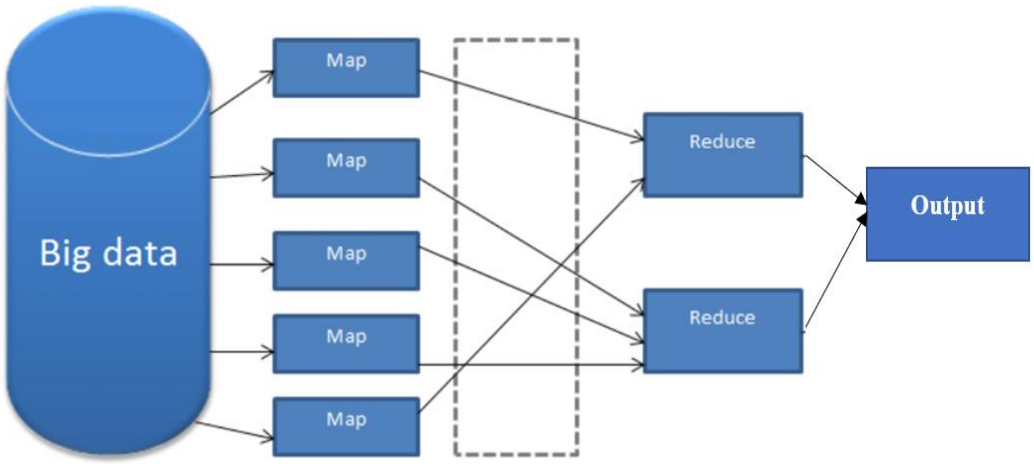

Figure 4.2 HDFS Map Reduce Framework

The proposed study reduces unnecessary sensor data that maps out over time. In the above Figure 4.2 shows the HDFS Map Reduce Framework. The HDFS diagram is drawn using a generalized model using the minimization structure.

4.3.1 Map Reduce

The Hadoop's layout of map/reduction is based on designing a structure to run immense info or data collections by tricking them into instant chunks of procedures. To function on immense information collections, the map/reduce uses broadcast strategies on a cluster of systems from the group. It stands as a programming facet at the core of Apache Hadoop for proffering immense scalability across innumerable Hadoop clusters on the commodity hardware. On a Hadoop bunch, this MapReduce procedures huge un-structured datasets utilizing the distributed algorithm. Hadoop applications execute such jobs. The word MapReduce signifies two tasks namely Map Job and Reduce Job. To create key-value pairs, map project procedures the datasets. Those key-value-pairs are inputted to reduce job, which incorporates them to achieve desired outcomes. The input along with the output signal is stored in HDFS.

MapReduce comprises 'Two' functions

i) Map () Function

ii) Reduce() Function

4.3.2 Mapping Function

The Map () function Present on the Master Node (MN) Sections the input data into minute subprocesses to disseminate them to the worker node. This worker Node functions the moment procedures and sends the acknowledgments to the Master Node. The sub-operations are processed in association for varied attempt. Mainly, to disseminate the work amongst all of the map nodes, the input data is partitioned. Every data is known and then redirected into the first node. Hence, the pairs called as tuples are formed. For the whole nodes, the mapping function is exactly the same. Second, the tuples are shipped to the reduce the nodes

4.3.3 Reduce Function

The results of the comprehensive sub-operations are constructed by the Reduce () function. The results are incorporated for producing aggregated decision-centric results delivered as an Acknowledgment of the first large needs. The Reducer nodes process the whole tuples. Thus, all pairs bearing the exact same key are all counted up. Also, It's upgraded as the worth of that Specific key.

4.3.4 Feature Extraction

A segment of the Information termed feature may be used as a character that may aid in solving forecast problems. The quality in addition to quantity of features is quite important for the outcomes generated by the chosen model. It identifies the features of the datasets which are especially practical in discovering sentiments. The principal target is to locate the identifying features

that could separate the information into positive, neutral or negative classes and create improved Sentimental Classification precision.

It helps in determining the opinions of the various individuals by using the different models say a) Negative emoticon count, b) Positive emoticon count, c) exclamation mark count, d) question mark count, e) positive gazetteer words occurrence count, e) negative gazetteer words occurrence count, f) unigrams, g) bigrams, h) trigrams, i) n-grams along with j) Part of Speech (PoS) tag.

Towards the proficiency of created Gradient boosted decision rules, they are connected with some benchmark informational indexes and the outcomes are contrasted with some different techniques in the writing. The test results legitimize the attainability of our method to deal with work with high-dimensional information and its satisfactory execution regarding outlining time complexity and classification precision.

Input: *featured attributes*

Output: *Gradient boosted decision ruled out feature selected dataset values*

Start

Step 1: initialize *the term tweet Ts from preprocessed Ps.*

Step 2*: compute the tags as Tweet Tgs*

Step 3: *Process the feature occurrence attributes Ts, K.*

//transfer terms, k times selection

Step 4: Compute *relevance measure of*

$$Ps = (Ts)^n = \sum_{k=0}^{n} \binom{n}{k} Initilazed\ feature^k K$$

// total terms feature by extracted average

Step 5: K- *Specific features values.*

Step6: *Process for All Tweet tags--Tg*

Extract the terms Et ←Tg

Compute all the positive negative selected tweet tags Pt, Nt

Check if (Tg as Et)

Return all tag Selected feature Tg

End.

End

Step7: *return feature Tgs*

Step8: *End.*

The use of a significant factor is considered to be equivalent to the subgroup classification algorithm, which is suitable for analysis using a filter for the mean-mean value as classification and is equivalent to a subset classification algorithm.

The extracts and selected features are prioritized to create records set as in positive and negative terms. Positive and Negative Emoticon - For this feature, the counts of positive- emoticons' and negative- emoticons in the Twitter dataset are determined.

- Exclamation Mark For this feature, the count of exclamation marks (!) in the Twitter data is determined.

- Question Mark Count For this question mark feature, the count of question mark (?) in the Twitter data is regarded.

- Positive and Negative Gazetteer Words Occurrence For this feature, the positive-gazetteer words' and negative-gazetteer words' occurrence counts in the Twitter data are determined.

- The terms unigram, bigram, trigram, and n-gram are- used to determine the opinions. Unigram words one by one. Here, there is one gram of each word. Bigram (picture) refers to two words simultaneously. In this, the word next to each "2" creates a bigram. N-grams are basically fragments of words within the group. It once represents the number of grams N. Area of Speech Tag

- POS tag helps to analyze intuition and emotion in sentences. This feature detects easy identification of words and adjectives that are significant identifiers of content input. A POS tag refers special label allotted to every token, and a group of other grammatical types, verbs, such as tenses, quick, number (s), cases, adjectives

4.3.5 Feature Selection

The actual group of features which forms patterns at a thought dataset. It's done in order to decrease the size of this problem for learning algorithms that might augment classification accuracy because of a decrease in computation requisites. It elevates the classification accuracy as the data dimensions to train the classifier is diminished. It's done using the I-EHO technique.

Algorithm:

As tags are formed q_i a Tweet query as a collection $Q=\{q_1, q_2, \ldots, q_k\}$, where k is the number of Tags preferred as tweet sets. X={xi1, xi2,…, xil} denote the list of features retrieved for qi. Each feature is selected as max relevance measure for retained Y, where $Y \in \{0,1,2\ldots\}$ is the proper decision. F={f1, f2,.., fn} denote the feature set. 'n' used to denote the total number of features and 'm' represent the selective tags feature which emotions that should

select. ‘S’ denotes the group of tweet tags. to calculate the maximum features using the following equation,

$$M = \text{Max } (f1 \rightarrow S_{m-1}) \sum_{f1}^{fn} - \frac{1}{m-1} i(f1 - fn) \ldots \quad (4.1)$$

where, M is a class label. The algorithm selects one feature that maximizes each iteration.

Step 1: Construct a set feature T to contain the selected features. Initially $T_0 = \emptyset$ (n features).

Step 2: representation the tweet tags for selected data

Step 3: Select greatest crucial feature as first designated feature T1=max M

Step 4: For all Max m features as i=0 to select all cases

Add features to T then

Return redundant feature T

Step 5: Return term T

The feature selection remains as the weight age to form marginal relevance which is additional to the use of two such methods. One is measuring both continuous and continuous data interactions. Calculations therefore require alternative mutual information.

4.4 IMPROVED ELEPHANT HERD OPTIMIZATION(I-EHO)

The EHO algorithm was propounded by Wang and Essentially a swarm intelligence algorithm along with meta-heuristic hunt methodology. It appears from the modeling of overall herding behaviors of elephants and this behavior is expounded as follows. The elephant populace comprises countless

subgroups each with a number of elephants, termed as clans. Every clan moves under the direction of a oldest female (matriarch), whilst a number of ME (Male Elephants) that reached adulthood leave their clan and live alone. Other members at a clan are generally females and their calves because Male Elephants following their entire growth leave the clan to reside independently. Albeit they live, male elephants communicate with all others as of their clan through low-frequency vibrations. This structural freedom together with societal communication in elephant herd may well be represented in 'two' different surroundings: the primary environment where the whole elephants live beneath the matriarch's influence, and the next surroundings, in which male elephants live autonomously but still have connections with the clan. These surroundings are modeled as upgrading and dividing operators. In respects of EHO, these actions are often designed with 'two' operators: clan upgrade (wherein upgrade the elephants Together with matriarch present positions in each clan) along with a separation (which enriches the people diversity at the subsequent search phase).

I-EHO believes the following assumptions.

- The populace of elephants is divided into clans; every clan encircles a predetermined number of elephants.
- A predetermined variety of Male Elephants moves from the clan and live independently.
- The group of matriarch encircles the best Alternative in the herd of elephants while the worse alternative would be decoded in the job of the bunch of elephants.

Begin
Initialization MaxGen, Size
Initialize the Population
Evaluate fitness for each Population
Set Gen_count a=1
While a<MaxGen
Sort all according to their fitness
for all clans do
for all solutions in the clan **do**
Generate $C_{new, e_{m, n}}$ and Update $Ce_{m, n}$
Select and retain efficient solution between $Ce_{m, n}$ and $C_{new, e_{m, n}}$
Update $C_{best, e_{m, n}}$ and generate $C_{new, e_{m, n}}$
Select and retain efficient solution between $Ce_{m, n}$ and $C_{new, e_{m, n}}$
End for
end for
for all clans in the population do
Replace the worst solution in the clans
end for
Evaluate the Population and calculate fitness
end while
end

Figure 4.3 Pseudocode for I-EHO

4.4.1 Algorithm

Step 1 : The Entire elephant Populace is divided into j class. Each 'n' of the clan 'm' moves as per matriarch, in which matriarch implies the elephant e_m with the best fitness value in the generation.

Step 2: The new position of elephant 'n' in the clan *m*, $c_{e_m,n}$ implies the old position, c_{best,e_m} implies the best solution of clan $e_m, \alpha \in [0,1]$, is parameter which ascertains female elephants and 'r' is that the arbitrary number used to enhance the diversity of their population

Step 3: Position of the best elephant on the clan c_{best,e_m} is updated. $\beta \in [0,1]$ is the next parameter of the algorithm that controls the influence of the a_{center,e_m}.

Step 4: Consider $1 \le d \le D$ is d^{th} Dimension and D is the entire dimension of this distance and nem suggests the number of elephants around the clan m. Elephants off that back as of the clan are used to simulate exploration.

Step 5: In every clan u, and some elephants with the worst case of the objective function are shifted to the next positions.

Step 6: Evaluate $c_{\min}$ and $c_{\max}$ and its an lower and also upper bound of the find space respectively. A parameter $rand \in [0,1]$ is a random number selected from a uniform distribution.

Step 7: Next the positions of this Elephants have been calculated, crossover and mutation is done in order to make the optimization more successful. The Two-point crossover has been chosen as of disparate kinds of crossovers. The genes at betwixt the two points and therefore are interchanged betwixt the parent chromosomes and consequently kids' ones have been attained.

Step 8: Replacing a variety of genes included in of each chromosome with new genes. The swapped genes would be the naturally occurring genes with no any recurrence inside the chromosome. The practice

is re-executed before the solution with greater fitness value is attained.

Step 9: End

After the autonomy of this structure in conjunction with the social interaction of the elephant herd, sufficient clans can represent different environments. In this case, the model is made in separate updates from the operators. In EHO's terms, these actions can be well designed using the clans operator: clan renewal (where each clan adds another to the elephant's inclination to the current status of the clan), and (the following hypotheses, which upsurge the assortment of the populace in the quest after separation I-EHO).

One of the heirs of the elephant population, Clan is divided by a fixed number of elephants. The number of MEs living outside their own family or living alone is constant.

All families of elephants come under the patriarchal leadership. The patriarchal group, the elephant herd, embraces the best solution while the remedy is coded from the position of the male elephant group.

A complete elephant habitat is divided into J species. It is the generation. Each member "N" is the best fit value. Mutations are then made by mutating a large number of genes into a new gene, like each chromosome. Swap genes are randomly generated on the chromosome without causing it to recur. The procedure is restored until a solution is obtained for the best exercise. I-EHO's pseudo-code can be unleashed using figure 3

4.4.2 Calculation of Special Significance

In calculation, the score value is assigned to each function. The score value is created based on some properties. In the proposed system, the words have occupied their meaning using the Seniti word net. Features have taken up space around the material provided. After steps to separate the quality from the dataset, the classification is performed. Here, a score value is assigned for each feature. On the base of that score value, a hierarchy is created based on some characteristics. The number of occurrences of positive and negative words in each document was counted to determine the document's sentiment score. To calculate the document sentiment score, each positive word counts as + 1 and each negative word as − 1.

4.5 GRADIENT BOOSTING DECISION TREE CLASSIFICATION (GBDT)

The Selected features are supplied as input into the GBDT classifier. GBDT is the extreme propitious machine learning model for classification and regression issues which creates a forecast frame at the type of integration of feeble forecast frameworks, and decision trees. GBDT implements a model in a stage wise model as other boosting techniques do, and it allows optimization of a subjective differentiable loss function. The primary aim of boosting is to attach new one to the ensemble successively. At every one of these iterations, a novel weak-base layout is trained in regard to this error of the whole ensemble learned. GB trains several techniques, particularly in a continuous, sequential, and additive method. In case of GB, the learning strategy sequentially matches new models for transporting a more exact evaluation of a class variable. Every new model is connected to some negative gradient of the system loss function and tends to reduce it. The Loss function acts as a step specifying how great the model's coefficients have been in matching the basic data. The most significant motivation for using GBDT is that it enables you to optimize a user-specified

cost feature, instead of loss functions that typically provide less control and does not correspond with applications that are instantaneous. GB adopts Decision Tree (DT) as foundation feature, and it is essentially simple yet very effective technique for learning linear in addition to non-linear function with a linear combination of a succession of DTs.

Adopting a GBDT Basic Function as a Decision Tree (TT is simple, but it is a very effective method of combining linear learning as well as nonlinear functions into a series of DTS.

4.5.1 Algorithm

Step 1: F(u) is an approximation part of the response based ‘v’ centered on a collection of predictor variables ‘u’. The squared error function is applied as the loss function L to evaluate the approximation function

Step 2: If the number of breaks S for each regression tree, hence each tree divides the input space ’S’ into disjoint areas $D_{1n},...,D_{kn}$ ’and then envisages a continuous value c_{kn} for the region.

Step 3: Using training data $\{vi,ui\}_{i=1}^{M}$, the GBDT iteratively assembles N different individual regression trees $a_1(u),...,a_N(u)$. The upgrading approximation function F(u) and operate along with gradient descent step size ρ_n.

Step 4: Utilizing a separate optimal γ_{kn} for every region D_{kn}, c_{kn} can then be discarded.

Step 5: GBDT is Stage-wise style in addition to upgrade the version by decreasing the expected price of a specific loss function. To stop more fitting and improve classification precision, the GBDT implements a shrinkage strategy.

Step 6 : End

Implementing the shrinkage strategy, the overfitting issue can well be shunned by shrinking the effect of each extra tree. Small shrinkage values can better lessen the loss function. Nevertheless, a bigger number of trees may be inputted to the model. Another parameter, tree complexity alludes to the number of splits (or else the nodes) and is utilized for fitting each DT. It signifies the depth of diverse interactions on a tree. Increasing the tree intricacy can capture more intricate interactions amongst variables and utilize the strength of GBDT. Relying on the shrinkage as well as tree complexity value, the optimum number of trees could well be found via checking to what extent the model fits on the test data-set.

The GBDT algorithm is built to analyze the significance and bias of the results. Further, the decision-making survey, in which the confidence and disadvantage of lower and slope upper approximations are described as classes.

However, trees can enter into a large number of samples. Other parameters are used to it each DT, inferring the number of complex partitions (or terminals) in the tree. Meaning that the depth of the various interactions in the tree. Increasing the complexity of the tree allows you to capture more complex interactions that are both variable and positive classes of the GBDT. Depending on the tree, complexity will retain the abstract value. The Trees are optimal and cannot properly test the test dataset via a model fit test. Finally, GBDT classifies positive and negative as well as neutral Twitter data.

Input: $D = \{(u_1, v_1), \ldots, (u_n, v_n)\}$, γ

Output: $F(u)$

Begin

$$F_0(u) \leftarrow \arg\min_{\rho} \sum_{i=1}^{N} F_n(u)$$

For $k = 1$ to K

$$\bar{v}_i = -\left[\frac{\partial L(v_i, F(u_i))}{\partial F(u_i)}\right]$$

$$\gamma_{kn} = \arg\min_{\gamma} \sum_{u_i \in D_{kn}} (\bar{v}_i - \gamma)^2$$

$$\rho_n = \arg\min_{\rho} \sum_{i=1}^{M} L\left(v_i, F_{n-1}(u_i) + \rho \sum_{k=1}^{K} c_{kn} I(u \in D_{kn})\right)$$

$$F_n(u) = F_{n-1}(u) + \tau \cdot \sum_{k=1}^{K} \gamma_{kn} I(u \in D_{kn}), \qquad 0 \prec \tau \leq 1$$

End For

End

Figure 4.4 Pseudocode for GBDT

4.6 PERFORMANCE ANALYSIS

The proposed system copes with the efficient Sentimental Analysis of twitter data. This is done utilizing the GBDT. The elucidation of the utilized database, the performance together with comparative analysis is elaborated here.

4.6.1 Database Description

A dataset centered on Twitter data is taken (Twitter Sentiment Analysis Training Corpus (Dataset)). This dataset comprises loads of formerly categorized tweets in respect of sentiments. The dataset is grounded on data that is attained from two sources. The primary source is the 'University of Michigan' SA competition on Kaggle. Every document (a line in the data file) is a sentence extracted from social media (blogs). The secondary source is the 'Twitter Sentiment Corpus' by Niek Sanders. It consists of 5513 hand classified

tweets. These tweets were classified grounded on one of the four different topics. It contains positive, negative and neutral labeled data. The Twitter SA Dataset comprises 896886 categorized tweets; each row contains ItemID, Sentiment, SentimentSource, SentimentText is marked as '1' for positive sentiment and '0' for negative sentiment. Here, 1/10 of the corpus is utilized for testing, while the rest could be contributed for training to classify sentiment.

4.7 RESULT AND DISCUSSION OF GBDT

4.7.1 Comparative Analysis

The proposed GBDT is contrasted with the prevailing approaches like the DCNN, ANN, DLNN and DLMNN However, DCNN is an unsupervised classification technique having Deep learning process. So, the proposed work was contrasted with disparate techniques say clustering, classification and also deep learning process. The disparate parameters are assessed and contrasted as delineated here. The proposed GBDT performance is analyzed and contrasted with the Deep CNN (DCNN), Artificial Neural Network (ANN), Deep Learning Neural Network (DLNN), and Deep Learning Modified NN (DLMNN) techniques in respect of metrics namely a) precision, b) recall, c) F-score, d) Time Complexity, e) accuracy and f) average sentiment score.

Table 4.1 Comparison of Existing techniques and the Proposed GBDT for 100 Numbers of Data

Metrics	Existing DCNN	Existing ANN	Existing DLNN	Existing DLMNN	Proposed GBDT
Precision	89.01	87.23	87.95	88.88	90.45
Recall	90.95	90.45	90.65	92.12	93.11
F-Score	89.65	88.23	89.02	90.23	92.04

Table 4.1 compares the existing techniques such as ANN, DCNN, DLNN, DLMNN and proposed GBDT model in respect of metrics say Precision, Recall, and F-score for 100 data. The Precision, Recall, and F-Score value for the proposed GBDT for 100 data are 90.45, 93.11 and 92.04 respectively. It is inferred from the table that the proposed GBDT shows efficient improved results when contrasted with the prevailing approaches.

4.7.2 Precision

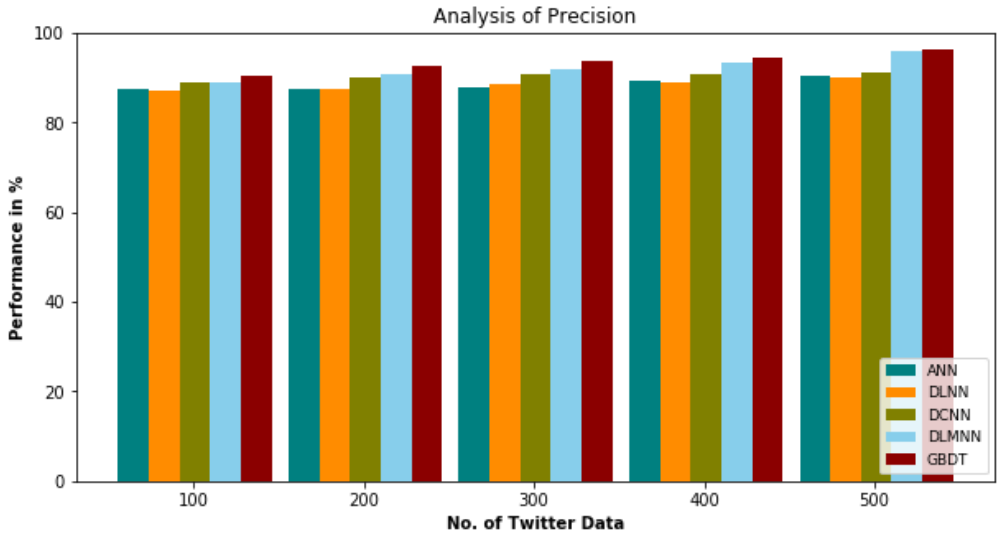

Figure 4.5 Performance Analysis of Proposed GBDT in terms of Precision

Figure 4.5 depicts the precision values for disparate quantities of data. It analyzes the performance of existing ANN, DLNN, DCNN, DLMNN and Proposed GBDT. It is observed from the precision graph that the precision increases as the number of data (N) increases in the case of both the proposed GBDT and the existing models. For N=100, the precision for the proposed GBDT is 90.45 and for N= 500, the precision for the proposed GBDT is 96.45, which is 6.01% greater for N= 100. similarly, for all compared N values, the proposed GBDT proffered the greatest performance.

4.7.3 Recall

Recall denotes the TP rate. This value is an imperative metric to ascertain the system's performance. Here, the values of recall are computed for disparate values of data. The recall value increases as the data count increases. Figure 4.6 analyzes the performances of the prevailing ANN, DCNN, DLNN, DLMNN and proposed GBDT model in respect of Recall.

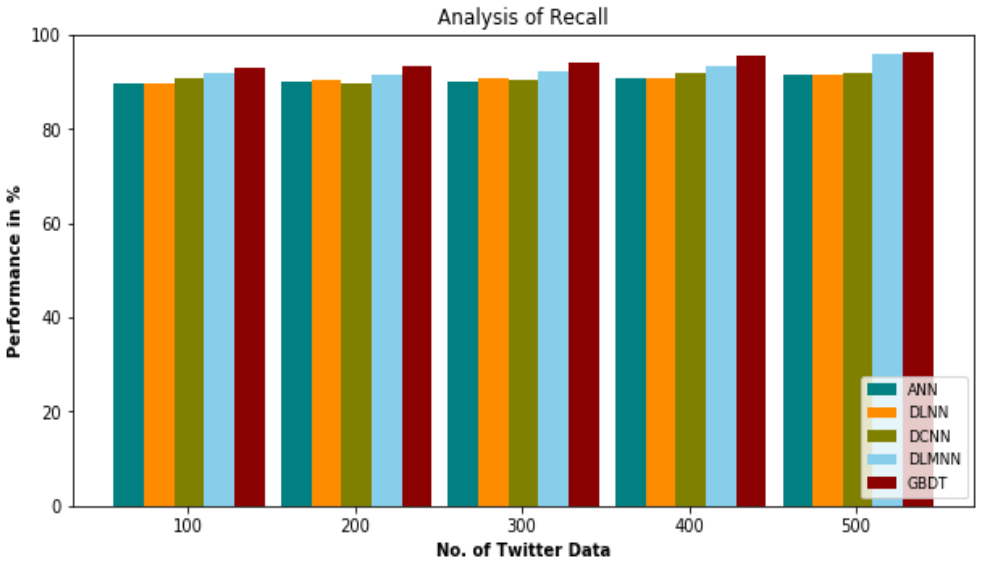

Figure 4.6 Performance analysis of Proposed GBDT in terms of Recall

Recall, for N= 100, 200 and 300, the recall value is approximately similar for the proposed and prevailing techniques. As the N values are increased further, the recall value is also augmented. For N= 500, the recall value for the proposed GBDT is 96.46, which is higher on considering the existing DLMNN and DCNN and in turn, it shows the greatest performance.

4.7.4 F-Score

The next performance gauge that is utilized for the comparative examination is the F-score. The F-score of this proposed GBDT, ANN, DCNN, DLNN and DLMNN are calculated and compared. The attained

F-score showed a stable increase as the data count increased. The DLNN witnessed an unequal increase and decrease on the value of F-score.

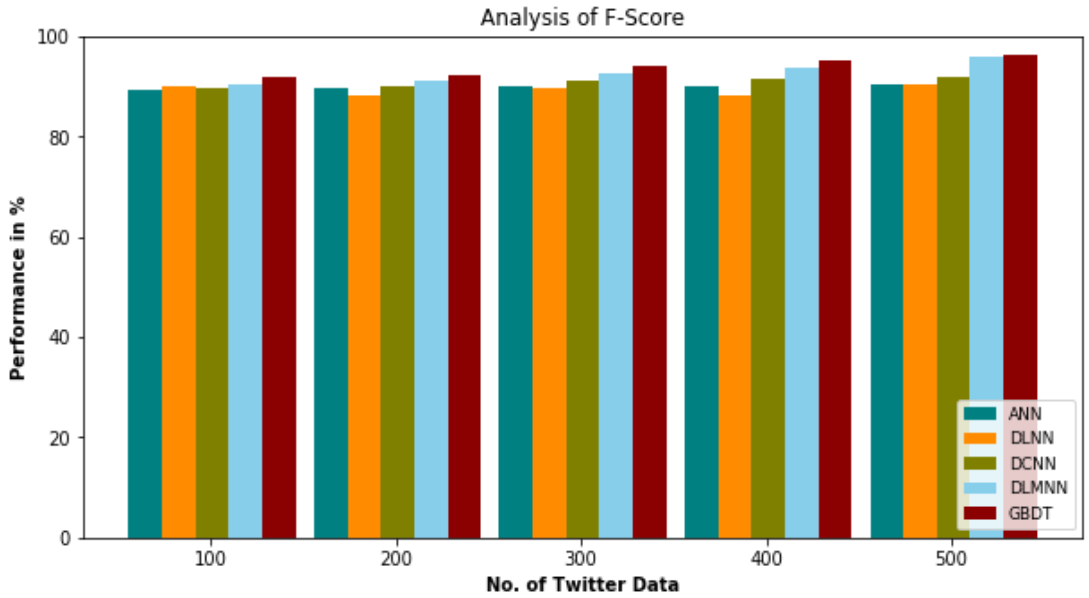

Figure 4.7 Performance Analysis of proposed GBDT in terms of F-Score

Figure 4.7 analyzes the performances of the existing techniques and proposed GBDT model in respect of F-Score. It is inferred that as the N value increases, the F-Score value also increased. For N= 100, F-Score value for the existing ANN, DCNN, DLNN, and DLMNN are 89.45, 89.83, 90.91 and 90.27 respectively and for the proposed model GBDT is 91.89. For N= 500, the proposed GBDT shows 96.36 F-Score value, which is higher on considering the existing techniques and in turn, it shows the highest performance

4.7.5 Time Complexity

It is gauged in seconds in this analysis. The proposed GBDT has an Time Complexity of 18.91, 60.11, 144.89, 351.68, 856.17 and 1052 for 100, 200, 300, 400 and 500 data respectively. The system's Time Complexity with ANN, DLNN and DCNN is increased with the elevation in the number of data.

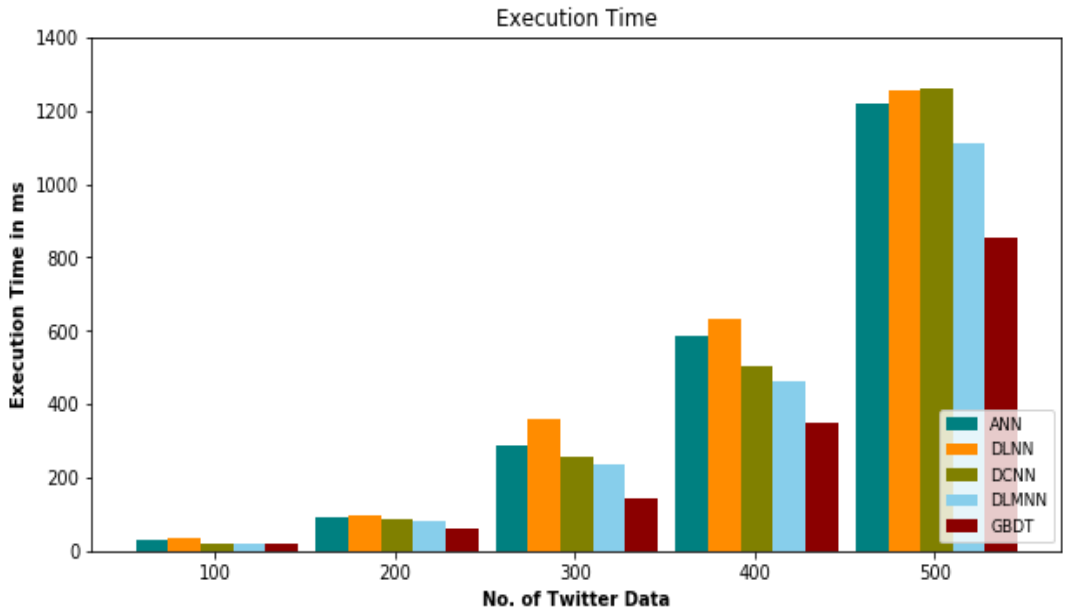

Figure 4.8 Performance analysis of proposed GBDT in terms of Time Complexity(ms).

Figure 4.8 analyzes the performances of the Existing ANN, DCNN, DLNN, DLMNN, and proposed GBDT model in respect of Time Complexity. It is inferred that as the N value increases, the Time Complexity also increase. For N= 100 and 200, the time taken for the execution is very low for the proposed model and the existing techniques. But amongst them, the proposed GBDT has the lowest Time Complexity which in turn implies the greatest performance. For N= 500, the existing DCNN technique has the highest Time Complexity of 1260.22 milli seconds and the existing DLNN technique has the highest Time Complexity of 1258.11 milli seconds, but the proposed GBDT model one takes 856.11 milli seconds, which is lowest when compared with the existing ones. Hence, the proposed GBDT model implies greatest performance.

4.7.6 Accuracy

The proposed work's accuracy witnessed an increase with the increase in the data count that was taken for analysis. For analyzing 100 and

500 data, the proposed GBDT yielded an accuracy of 84.78 and 93.86 percent respectively. Figure 4.9, visually elucidates such observations.

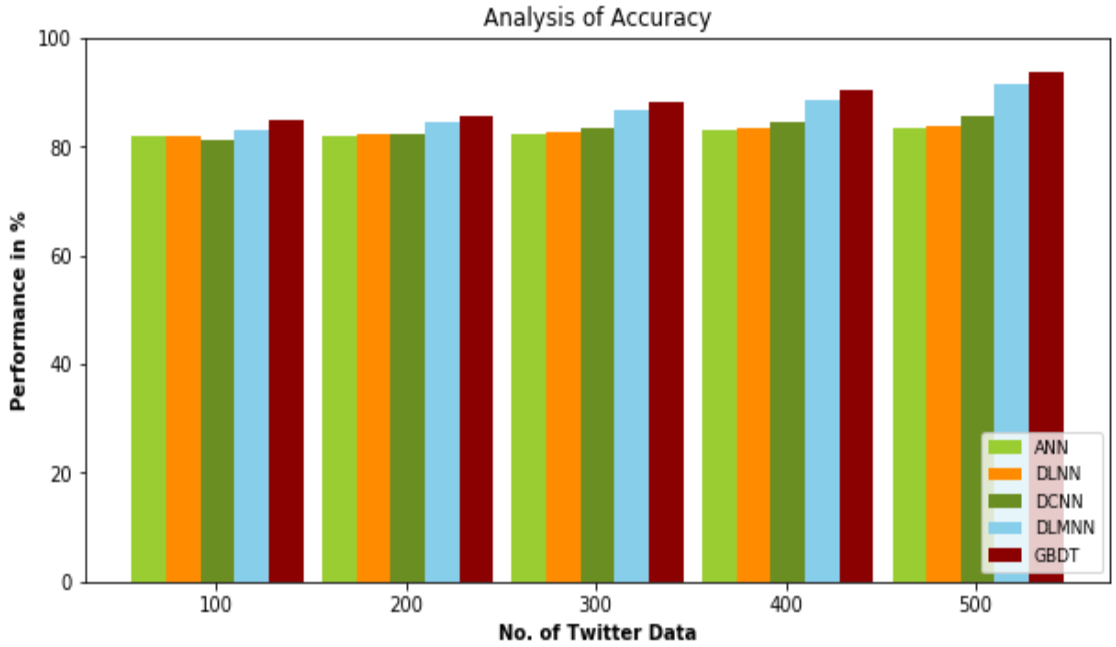

Figure 4.9 Performance Analysis of proposed GBDT techniques in terms of Accuracy

Figure 4.9 analyzes the performances of the existing model with proposed GBDT centered on Accuracy. It is observed from the graph that the accuracy value increases as the N values augment in the case of both the proposed GBDT and existing techniques. For N= 100, the accuracy for the proposed GBDT is 84.78 and for N= 500, the accuracy for the proposed GBDT is 93.86, which is 8.88% greater from N= 100. Likewise, for all compared N values, the proposed GBDT model proffers the greatest performance.

4.7.7 Average Sentiment Score

It is a performance measure that is evaluated for the proposed and prevailing approaches. This value represents the sentiments' polarity.

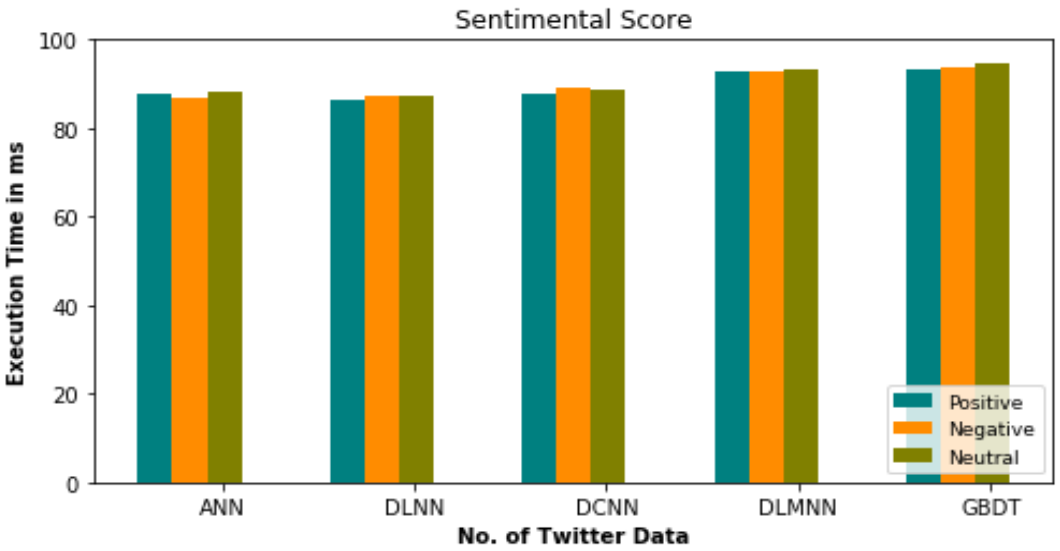

Figure 4.10 Performance analysis of proposed GBDT in terms of Average Sentiment Score

Figure 4.10 delineates the comparative examination of the ASS for the proposed GBDT techniques and the prevailing works. The average sentiment score performance varies based on the data count. The ANN shows a steady increase in the average score. This is because of the misclassification in sentiment analysis. The DCNN also yields certain misclassifications. The proposed GBDT has very few classifications; hence, the ASS is lower on considering other methods that are taken for comparison. For N= 100, the average sentiment score is high for both the proposed GBDT and the DLMNN techniques. But amid them, the proposed GBDT technique has the highest average sentiment score. It is observed from the graph that the accuracy value increases as the N value augments in terms of both the proposed model and the existing techniques. Compared with the existing ones, the proposed GBDT technique proffers the highest performance.

4.8 SUMMARY

From the experiential result, it is apparent that the proposed SA utilizing Twitter data is a highly proficient approach in the big data domain.

The steps that were performed in the proposed system are preprocessing, map reduction, feature extraction, ranking, classification, and validation. In preprocessing, it executes six processes namely tokenization, stemming, removal of stop words, Lemmatization, Filtering and removal of Hash tags. The HDFS provided a structured representation of data. The relevant emoticon and the non-emoticon features were extracted. These features were appropriately ranked centered on their characteristics. The classification technique that was employed in the proposed work was the GBDT algorithm. The performance metrics including recall, precision, F-score, accuracy, computation time and average sentiment score were evaluated and compared with the existing techniques. It was affirmed that the proposed system had greatest results when contrasted to the ANN, DCNN, DLNN and DLMNN algorithm.

An anticipated feature selection and feature classification methodology is proposed. Here Hadoop framework with I-EHO algorithm estimation is related with the high dimensional twitter dataset to pick the ideal features. The arrangement is done through GBDT classifier with feature selection effective approach decision classifier. Feature selection using I-EHO, demonstrates the proposed grouping system has defeated the previous approaches by having better accuracy for five benchmark datasets. Accuracy of 93.86% is reported in lung dataset. From the outcome, the proposed approach of GBDT classifier-based system has higher accuracy, higher affectability and higher specificity than the existing procedures to classify the tweet dataset. In this manner, our proposed GBDT technique serves the best information arrangement structure to categorize the result. In future, the analyst will have discriminating chances to perform with different feature selection strategy and make more dynamic statures of best performance in execution.