CHAPTER 3

DLMNN-BASED PROFICIENT SENTIMENT ANALYSIS

3.1 INTRODUCTION

The rapid enhancement in social networking services over the internet generates massive information in real-time scenarios which has a striking impact on big data analysis. It resulted in the elevated usage of emotions and sentiments in social media. This works proffers a proficient sentiment analysis technique in Twitter data. Primarily, the data as of the Twitter database is preprocessed, tokenization, stemming, stop word removal and number removal are done. The preprocessed words are then passed into the HDFS (‘Hadoop Distributed File System’) to reduce the repeated words and also that are eliminated using the MapReduce technique. Then, the emoticons and the non-emoticons are extorted as features. The resulted features are ranked centered on their meaning. Then, the classification is performed by utilizing the DLMNN (Deep Learning Modified Neural Network). Also the optimization is done using the PSO (‘Particle Swarm Optimization’). Finally, the attained results are validated utilizing K-fold cross-validation methodology. The results are evaluated and compared with the existing works.

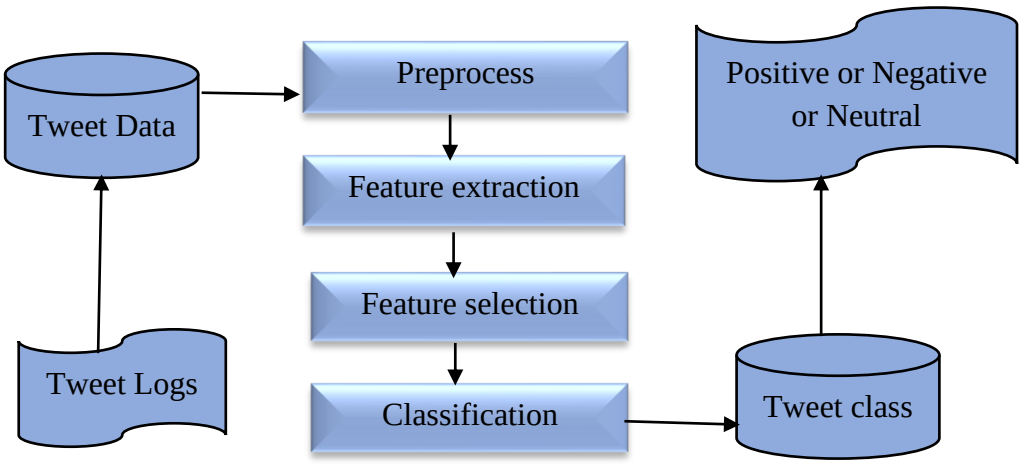

Figure 3.1 Process of Tweet Dataset and Classification

Figure 3.1 shows the Process of tweet dataset and progress classification. This is pertinent to design an accurate and efficient emotional classification system that is not determined by the very individual functions or intended categorization processes of past research regarding emotional classification. The proposed system combines many feature extraction techniques with emoticons such as exclamation marks, word place names, and unigrams to design the most accurate emotional classification system. This study provides an empirical comparison of emotional classification mechanisms.

3.1.1 Big Data

Big Data indicates the datasets which are too large and intricate to handle. Those applications require application software to execute different analysis. Big data indicates the process that is utilized when traditional handling and data mining methodologies could not un-cover the insights and meaning of the provided data. The data which is employed is time sensitive, unstructured, simple or extremely large and it could not be processed with relational database engines. Immense parallelism on readily available hardware is requisite for executing voluminous computations. Big data is available on social platforms. The quantity of content created by users is too vast for a normal user to analyze. So it is needed to be automated. The automation is coupled with Machine Learning (ML) algorithms for the augmented analysis. The emotions of individuals are often expressed in different forms in social platforms. Evaluating these opinions is crucial. With this intention, disparate SA (‘Sentiment Analysis’) methodologies are broadly used.

3.1.2 Sentiment Analysis

This analysis is executed on twitter data for examining the information that is existent in the tweets. In tweets, the opinions are positive/negative, heterogeneous and unstructured. SA is perceived as a process that mines the views, opinions, emotions, and attitudes from speech, text, tweets together with database sources via natural language processing. The SA of disparate data may encompass numerous tasks like i) sentiment extraction, ii) sentiment classification, iii) subjectivity classification, iv) summarization of opinions v) opinion spam discovery. It targets to examine the emotions, people's sentiments, attitudes and opinions towards an individual, product, topic, service or organization.

3.1.3 Tweet Reflect Sentiment

Twitter data comprises a striking predictive influence. The data that is attained as of social media is in the forms of news, forums, product reviews or even blogs. These data can also contain sentiment-based sentences. The sentiment is delineated as a personal opinion or view that is held or expressed. The twitter microblog service comprises a large column of frequently self-standing short textual tweets that are made open to the research community. Emotions could be articulated by tweets. The six fundamental human emotions are fear, disgust, surprise, anger, and sadness. Tweets mirror such emotions precisely. This work copes with a proficient SA on twitter data utilizing DLMNN.

3.2 PROBLEM IDENTIFICATION FACTORS

Numerous works are being carried out in the domain of SA of twitter data. Many research works are surveyed in this section. Yili Wang *et al.* (2018) suggested a methodology for twitter SA. A 'modified

Chi-square' centered feature clustering & weighting strategy was employed on the twitter message. Together with the 'part of speech' tagging, the dependency, and discriminability of the words in the tagged training dataset, were regarded. The multi-nomial Naïve Bayes prototype was utilized for handling undesirable features. The influence of the emotional words was augmented for raising the accuracy. Sven Rill *et al.* (2014) presented a structure to spot the budding political theme in twitter before it attains the other standard data channels. It was also elucidated that how those topics were utilized to widen the knowledge bases which are requisite for conceptual level SA. This work employed sentiment hash tags that were attained as of a German society amid a parliamentary election. Soujanya *et al.* (2016) propounded a deep learning strategy to mine the opinion. This extortion is a sub-task of SA that comprises recognizing opinion targets. A seven layered DCNN ('Deep Convolutional Neural Network') was utilized to tag all words as either facet or non-faceted word. An assortment of linguistic models for the same specific purpose was introduced and joined with the NN (neural network). The resulting group classifier was integrated with a word-embedded design for SA. Pandey *et al.* (2017) formulated the twitter SA utilizing hybrid cuckoo-search methodology. This methodology was utilized to ascertain the optimal cluster head as of the sentimental subjects of the Twitter dataset. This methodology had generalized inferences for the system model that could proffer conclusive surveys for whatsoever social issue. The statistical examination was executed for validating the algorithm's performance.
Hassan *et al.* (2016) recommended a lexicon centered approach for SA on a Twitter dataset. This methodology permitted the sentiment detection at the tweet level and entity level. This methodology was assessed on three twitter datasets utilizing three disparate sentiment lexicons to attain word centered sentiments. This methodology displayed augmented performance on considering the other prevailing methodologies in the initial two datasets, but it was 1% marginal in F-measure on the 3rd dataset. David Gonzalez-Marron

et al. (2017) propounded a methodology to pick the utmost relevant algorithms for examining tweets for specific words written mostly in Spanish. Certain resources were chosen and those were prioritized to that which could analyze Spanish sentences and those which could interact with the python language. It was perceived that merely 85% of the outcomes were satisfactory. The error that transpired in the analysis was owing to slang use, regionalisms and sarcasm in Spanish. Zhao *et al.* (2018) introduced a word embeddings methodology that was attained by unsupervised learning grounded on large twitter corpora. This methodology utilized the latent contextual semantic relationship along with co-occurrence statistic characteristics betwixt the words available in tweets. The Word Embeddings (WE) were incorporated with word-sentimental polarity score feature and n-gram features to develop a sentiment features assortment of tweets. The features' set was integrated to a DCNN for training and also predicting the existent sentiment classification labels. Shufeng *et al.* (2017) developed a multi-leveled sentiment enrich we are learning methodology. This methodology employed a parallel asymmetrical NN that designed the n-gram, word-leveled sentiment together with the tweet level sentiment amid the learning. Numerous experimentations were executed on benchmarks.

3.2.1 Problem Definition

Tweet analysis contains sentimental terms in the form of multi-dimensional way to represent real term opinion values which can represent many real-world problems mainly taken from the feedback tweet data's from online. It is often a difficult task to select essential feature variables for classification or regression problem which is difficult on classification. The twitter classification of information has turned into an undeniably difficult issue, because of ongoing advances in mining innovation. Feature selection or extraction process is a noteworthy piece of information classification. As sentimental analysis feature selection process, the calculation cost diminishes

and furthermore, the classification execution can also increase. A reasonable portrayal of information from all features is an essential issue in machine learning and information mining issues. The major problem of reducing classification accuracy is occurred by a grouping of non-related features formed by training feature learning and transmission purpose for classifying the statistics sets with emergent number of features and classified dataset. The classification technique doesn't require feature requirements to make class by reference of the categorized object. The opinion is generally in the form of sentiment analysis, which may not satisfy the informational needs. The sentiments remain extremely shapeless, varied, and may be positive or negative.

3.2.2 Methodology

The existing SA on twitter dataset requires much time to train the dataset. Moreover, the feed-forward methodology in the prevailing work requires the transmutation of the dataset to a window centered model and a vector table. The extraction process of sentiment features still rests as a chief challenge for data miners. Also, the existing sentiment analysis techniques provide less accuracy. The proposed system tackles all such problems in a convenient manner. In the proposed work, proficient sentiment analysis is performed. As the chief contribution, proposed system performs six stages like pre-processing, map reducing, feature extraction, ranking, classification, and validation. Originally, the input twitter data is preprocessed using tokenization, stemming, removal of stop words, and removal of numbers. Next, the repeated data are removed using HDFS (i.e., map () and reduce ()). Then, Emotion and Non-emotion features are extracted. Next, the extorted features are ranked, and the ranked values are given as input to the DLMNN classifier. Here, the modification process is done by using PSO algorithm for optimizing the weight

value of the ranked values. Each step is precisely elucidated. Figure 3.2 proffers the detailed delineation of this proposed work.

3.2.3 Novelty of the Proposed System

This works proffers a proficient sentiment analysis technique in Twitter data. Primarily, the data as of the Twitter database is preprocessed, tokenization, stemming, stop word removal and number removal are done. The preprocessed words are then passed into the HDFS ('Hadoop Distributed File System') to reduce the repeated words and also that are eliminated using the MapReduce technique. Then, the emoticons and the non-emoticons are extorted as features. The resulted features are ranked centered on their meaning. Then, the classification is performed by utilizing the DLMNN (Deep Learning Modified Neural Network). Also the optimization is done using the PSO ('Particle Swarm Optimization'). Finally, the attained results are validated utilizing K-fold cross-validation methodology. The results are evaluated and compared with the existing works.

3.3 PREPROCESSING

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.

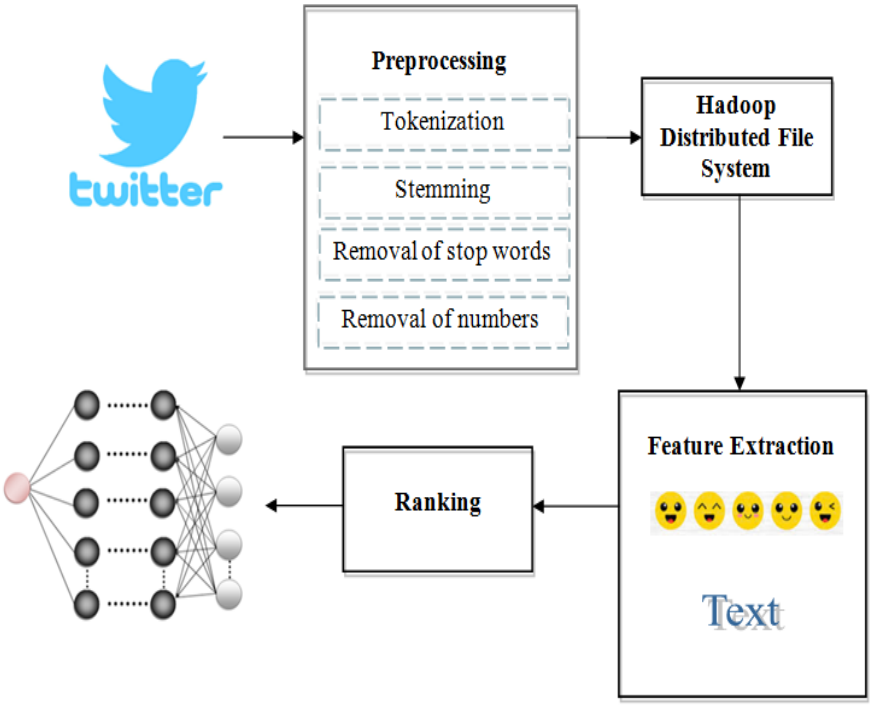

Figure 3.2 System Architecture of the proposed DLMNN.

Figure 3.2 shows the Architecture diagram of DLMNN. The stages of making great contributions and suggestions in deep learning neural network are pre-implementation, minimization map, feature extraction, classification validation. First, the included Twitter data tokenization begins to work using Tokenization i.e. the elimination of stop words and the reduced impact of numbers. Again, the data is deleted using the Hadoop distributed file system called Map Reducer. Emotional and non-emotional aspects have been obtained. Extracted features occupy space and the standardized values are entered as a DLMNN classifier. Here, the reform process is performed for integrating rank value and weight value.

The various stages in the proposed method are

- Preprocessing
- Feature Selection
- Classification using PSO- DLMNN Classifier.

3.3.1 Tokenization

Initially, the values are read from the Twitter Dataset. Then the tokenization process replaces a sensitive data to a non-sensitive one termed token. A token has no extraneous meaning. Tokenization is done to part the values into words. Then, unwanted words are wiped out.

3.3.2 Stemming

It is done utilizing SentiWordNet. The SentiWordNet is endowed with a web-centered GUI. This step removes the suffix like ‘ing’ and ‘ed’ from words. This process is normally done so that the SA on the twitter datasets is made more effectual and efficient.

3.3.3 Removal of Stop words

A list of commonly utilized words in any language is termed as Stop Words (SWs). The utmost commonly seen SW is “the”. Some other SWs 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 SA of twitter data, the SWs do not comprise any information related to emotions. Therefore, the stop words must be removed. The proposed system is programmed to disregard these SWs

3.3.4 Removal of Numbers

The sentences that are present in the database contain numbers. The numbers in sentences have no great significance in the SA. Numbers possesses no information related to emotions or sentiments. Hence, during preprocessing, the numbers are also eliminated.

3.4 HADOOP DISTRIBUTED FILE SYSTEM (HDFS)

After preprocessing, the words that are attained are built in a structured form utilizing HDFS. It proffers a consistent way of handling big data pools. It also upholds speedy data transfer betwixt nodes. At its onset, it is closely coupled with 'MapReduce'. MapReduce is basically a programmatic structure for data processing. In the proposed system, it augments the elimination of repeated words. MapReduce is concerned as a program prototype and a processing methodology for distributed computing centered on Java. The MapReduce algorithm comprises two notable tasks like Map & Reduce. The Map takes an assortment of data and transmutes it to another set of data. Here, the elements are then broken to tuples individually. To reduce task, it considers the output of a map as an input. It also integrates such data tuples to a tiny tuple set. The chief benefit is that it is simpler for scaling data processing over numerous computing nodes.

3.5 FEATURE EXTRACTION

The subsequent feature extortion phase diminishes the resources that are requisite to represent a particular dataset. Redundant data is transformed to an assortment of a reduced data set. The emoticon and non-emoticon features are precisely extracted as of the Twitter dataset.

3.5.1 Emoticon Features

The Emoticon features signify the emotion symbols which are present in the specified dataset. Emoticons are ASCII art. These emoticons are often indicated as "smileys". Emoticons are formed by the creative utilization of numbers, letters as well as punctuation symbols. Mostly, emoticons represent different facial features. Emoticons are modeled for adding emotional flavors to normal messages with plain texts. A number of emoticons exist in tweets. These emoticons are considered as important features which are extracted in the sentiment analysis. This emoticon is categorized as positive/negative. The

subsequent Table 3.1 gives certain emoticon feature that was extracted in the proposed work.

Table 3.1 List of sample Emoticons

Emoticon	Meaning	Polarity	Strength
	Big grin	Extremely positive	1
	Big grin with glasses	Extremely positive	1
	Laughing	Extremely positive	1
	Hi 5	Extremely positive	1
	Happy, Smile	Positive	0.5
	Kiss	Positive	0.5
	Straight face	Neutral	0
	Undecided	Neutral	0
	Sad	Negative	-0.5
	Broken heart	Negative	-0.5
	Sad with glasses	Negative	-0.5
	Crying	Extremely negative	-1
	Angry	Extremely negative	-1

3.5.2 Non-Emoticon Features

The Non-Emoticon features denoted the non-emotion symbols that exist in the Twitter dataset. Each of those features has a specific meaning. These features are the icons excluding the emotion icon. This includes the verbal information that is present in different sentences in the Twitter dataset.

3.5.3 Ranking of Features

The process of allotting a rank for each feature is the next task to be done. Ranking involves creating a hierarchy centered on some characteristics.

In this proposed system, the words are ranked grounded on their meaning. This step is executed using the SentiWordNet. The features are generally ranked grounded on the meaning that is provided in this dictionary. After ranking the dataset, classification is executed. The high-ranked data are classified first.

3.6 CLASSIFICATION

The ranked values are then classified using the DLMNN. To optimize its weights, DLMNN utilizes the PSO algorithm. The classified output indicates if the sentiment is positive/negative.

3.6.1 Deep Learning Modified Neural Network (DLMNN)

The classification of the sentiments in the proposed work is carried out utilizing the DLMNN. Each input is given to a discrete node prevailing in the input region of a classifier. The weights are arbitrarily assigned values and are linked with each input. The subsequent layer is regarded as the hidden layer. Moreover, the nodes in this layer are termed hidden nodes. These nodes perform the function of adding the product values of the input and the weight vector of all the input nodes that are linked to it. In DLMNN, the weight value betwixt Input and Hidden layers and also between Hidden and Output layers were optimized using PSO. Random weight value gives more Back Propagation (BP) process to achieve the result. To solve this, an optimized weight value was generated. The activation operation is then applied and the resulting output as of this layer is transported to the consecutive layer. These weights have an utmost impact on the classifier's output. The general depiction of the DLMNN is displayed in Figure 3.3.

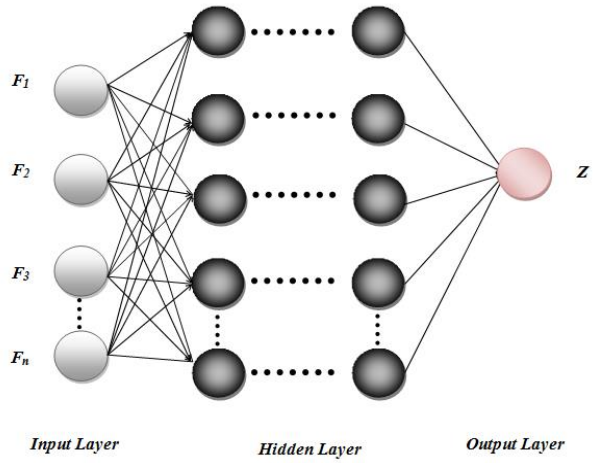

Figure 3.3 Structure of DLMNN

The algorithmic steps that are embraced in the DLMNN design are displayed here.

Step 1: Let the ranked values of different words and their equivalent weights be expressed as given below.

$$R_i = \{R_1, R_2, R_3 .... R_n\} \tag{3.1}$$

$$W_i = \{W_1, W_2, W_3 .... W_n\} \tag{3.2}$$

Step 2: Here, the inputs are then multiplied with the weight vectors that are arbitrarily selected and then they are totally summed up.

$$S = \sum_{i=1}^{n} R_i W_i \tag{3.3}$$

Here, S denotes the summed value, R_i signifies the input values, whereas W_i represents the weight values.

Step 3 : The 'Activation Function' (AF) is determined. This function is mathematically depicted as,

$$Z_i = f(\sum_{i=1}^{n} R_i W_i) \tag{3.4}$$

The category of AF that is utilized in this proposed system is the Gaussian function.

$$f(R) = e^{-R_i^2} = A_i \tag{3.5}$$

Step 4: Here, the output of the next hidden layer is stated as,

$$Y_i = b_1 + \sum A_i W_i \tag{3.6}$$

Where, b_1 denotes the bias value, W_i specifies the weight between the input & the hidden layers and A_i are values that are modified by the application of the AF.

Step 5: Here, the above 3 steps are performed for each layer in the DLMNN. Finally, evaluate the output unit by adding up all the weights of the input signals to attain the output layer neurons' value

$$Z_i = b_1 + \sum P_i W_j \tag{3.7}$$

Where, P_i denotes the value of the layer that precedes the output one, W_j specifies the weights of the hidden layer and Z_i indicates the output unit.

Step 6: This step contrasts the network output with the target value of output. The difference betwixt these 2 values is termed the error signal. This value is mathematically represented as,

$$er = D_i - Z_i \tag{3.8}$$

Where, er denotes the error signal, D_i specifies the aimed target output and Z_i denotes the current output of the classifier.

Step 7: Here, the output unit is contrasted with the targeted value. The related error is determined. Grounded on this error, a value δ_i is computed and it is utilized to allocate the error at the output back to all other units in the network.

$$\delta_i = er[f(Z_i)] \tag{3.9}$$

Step 8: The weight correction is assessed utilizing the BP methodology. This relation is proffered as,

$$wc_i = \alpha\delta_i(R_i) \tag{3.10}$$

Here, wc_i indicates the weight correction, α specifies the momentum term, (R_i) signifies the input vector, whereas δ_I denotes the error that is distributed in the network.

3.6.2 Particle Swarm Optimization Algorithm (PSO)

Swarm intelligence is a category of Artificial Intelligence (AI) that is grounded on the communal behavior of the decentralized & self-organized systems. It is commonly made of a population of simpler agents that communicate locally among themselves and with their environment. PSO algorithm indicates an AI method that ascertains the relevant solutions for extremely intricate problems. This PSO was modeled grounded on the societal conduct of a group of birds. In PSO, all particles fly on the searching space in a velocity adjusted by its personal flying memories and companions' flying experiences. All particles possess its own objective function value which is ascertained by an FF (fitness function). Disparate steps embraced in executing the PSO is delineated below:

Step 1: Initialize a swarm with the value of the position pos_i together with velocity vel_i chosen randomly for n-variable in the problem arena.

Step 2: For each of those arbitrarily generated particles evaluate the optimization FF in n-variables.

Step 3: Assess the fitness value with the particles' p_b value. If the present fitness value is best on considering the p_b then choose the current fitness value as p_b for further processing.

Step 4: This fitness value is contrasted to all previous best values. Then a condition is checked. If the current value is best on considering the previous value, then, update the g_b for the present particles array index and then value as the new g_b.

Step 5: Pick the particle which possesses the finest fitness value and then reinitialize its position. Along with this, assess the particle with the worst fitness value that whether its new position is satisfactory. If it is in an acceptable range then renew its position or else arbitrarily allot a new position to the particle in its neighborhood. Then, renew the velocity along with position of other particles utilizing the below expression,

$$vel_i(t+1) = wvel_i(t) + d_1a_1(g_i(t) - wc_i) + d_2a_2(g_b'(t) - wc_i) \tag{3.11}$$

$$pos_i(t+1) = pos_i(t) + vel_i(t+1) \tag{3.12}$$

Where d_1 and d_2 indicates the acceleration coefficients, $g_i(t)$ is the particle's position at a time t, a_1 and a_2 signifies the random values, w is the inertia co-efficient, wc_i is the corrected weight that denotes the particles' individual finest solution and g_b (t) is the best solution

of swarm at a time t. The illustrations are displayed as pseudo code using Figure 3.4.

Begin
for each particle
Initialize the position and velocity of the particle in the swarm.
End for
do
for each particle ($X=C_1,C_2,C_3\ldots\ldots C_i.\ C_k$)
Calculate fitness value of the particle
if fitness value is better than P_b
Set P_b = current fitness value
end if
if P_b is better than g_b
Set g_b = Best fitness value of all particles
end if
for each particle
Calculate particle velocity according eq.(11)
Update particle position according eq.(12)
end for
end for
end

Figure 3.4 Pseudo Code of PSO Algorithm

After pre-processing of tweets, PSO algorithm for clustering. The first step of PSO is initializing the number of particles. A particle is nothing but one of the possible solution for clustering the streaming tweets. Therefore, a swarm consists of collection of candidate clustering solutions of streaming tweets. Each particle is represented as X = (C1, C2, C3…Ci. Ck), where Ci represents the cluster centroid vector and k is the number of clusters. After initialization of the particles, for each particle, assign

each tweet to its closest centroid vector. The fitness of each particle is computed by considering the average similarity between the cluster centroid and a tweet in the document vector space, belonging to that cluster using cosine correlation measure. Experimental results show that PSO clustering out performs over hierarchical and partitioning clustering techniques.

3.6.3 K Fold Cross Validation (KFCV)

The very last step that is involved in the implementation of proposed system is the ‘k-fold cross validation’. This methodology is done for evaluating the results on a statistic manner. This step determines the degree of accurateness of the predictive model. Its’ target is to explore the proposed system’s ability to forecast the new data that are not employed in the estimation process. A single round of this methodology comprises the subsequent steps that are i) partition of the sampled data to complementary subsets, ii) executing the analysis on one sub-set and validating it on another subset. To diminish the variability, the proposed system utilizes a ‘10-fold cross validation’. The validation outcomes are integrated after every round. This is executed to attain an estimation of the analytical performance of the proposed design.

3.7 PERFORMANCE ANALYSIS

The proposed system copes with the efficient SA of twitter data. This is done utilizing the DLMNN. The elucidation of the utilized database, the performance together with comparative analysis is elaborated here. The sample input data and the relevant positive and negative sentiment score of the emoticons and non-emoticons features are shown in Tables 3.2 and 3.3.

Table 3.2 Emoticon Features

Words	Positive Score from Sentiwordnet dictionary	Negative Score from Sentiwordnet dictionary
☺	0.5	0

Table 3.3 Non - Emoticon Features

Words	Positive Score from Sentiwordnet dictionary	Negative Score from Sentiwordnet dictionary
Just	0.625	0
got	0	0
Home	0.25	0
Got	0.234	0
Friend	0	0
Zahra	0	0
haven't	0	0.211
Seen	0.322	0
Graduate	0	0
Year	0	0
Make	0	0
Happy	0.75	0
Total	2.181	0.211

3.7.1 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.

3.7.2 Performance Evaluation Metrics

It is executed by assessing numerous performance metrics. Precision ascertains the accurateness of the classifier in respect of all classes. Precision is a performance metric which is ascertained to observe the performances of this proposed work. This metric specifies the count of instances that are categorized correctly to the total count of True Positives as well as True Negatives. The mathematical depiction of precision is displayed in the succeeding equation.

$$precision = \frac{TP}{TP + FP} \tag{3.13}$$

where, TP specifies the True Positives &FP denotes the False Positives.

The next performance metric that is determined for performance analysis and it is named as Recall. The recall signifies the completeness of all classifiers in respect of each class. The value of Recall is calculated as exhibited in the mathematical equation.

$$recall = \frac{TP}{TP + FN} \tag{3.14}$$

Where, FN depicts the False Negatives.

Generally, F-score is a performance measure that takes the recall and the precision into consideration. Its maximal value is 1 whereas the minimal value is 0. It is evaluated as the harmonic average of the recall as well as precision values. The F-score is computed by,

$$F\_score = 2.\frac{(precision)(recall)}{precision + recall} \tag{3.15}$$

Accuracy is a performance gauge that denotes how closer the proposed system is to the target value. It is a gauge that ascertains the count of predictions that are made to the total count of predictions that are made. The system accuracy is computed utilizing the subsequent mathematical depiction.

$$accuracy = \frac{TP + TN}{TP + FP + TN + FN} \tag{3.16}$$

The Time Complexity is regarded as the amounts of time in which a CPU processes the instructions on a computer. This is a metric that possesses an imperative role in finding the proposed work's performances. The 'Average Sentiment Score' (ASS) is the mean value of the sentiment score. This value is an utmost precise numerical depiction of the sentiments polarity.

3.8 RESULT AND DISCUSSION OF DLMNN

3.8.1 Comparative Analysis

The proposed DLMNN is contrasted with the prevailing approaches like the DCNN and the K-Means Algorithm (KMA). However, K-means indicates the unsupervised learning algorithm that resolves the clustering problem. The only clusters formed by K-Means are utilized to train a classification model. DCNN is also 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.

3.8.2 Precision

Figure 3.5 depicts the precision values for disparate quantities of data. When the data count is 100, the precision of KMA, DCNN, and DLMNN are 89.13, 89.01 and 88.88 respectively. A stable decrease is perceived in the precision value in the proposed system. But when the count of data was elevated to 200, the precision was increased to 90.71 whereas the precision of KMA and DCNN were 88.63 and 89.74 respectively.

Table 3.4 Precision for different techniques

Method	100	200	300	400	500
K-Means	89.13	88.63	87.59	88.06	90.10
DCNN	89.01	89.94	90.87	90.65	91.30
DLMNN	88.88	90.71	91.91	93.33	95.78

For enhanced visualization, the Table 3.4 is plotted in the form of a graph as in Figure 3.5. The higher value of precision 95.78 is attained when the total data is 500. For 300 data, the proposed system achieves 91.91% precision, but the existing DCNN and K-Means achieves 90.87% and 87.59% precision. Next, for the data count of 400, the proposed DLMNN obtained 93.33% precision but the existing DCNN and K-Means obtained 90.65% and 88.06% precision. The analysis displays that this proposed system exhibits augmented performance on considering the prevailing techniques.

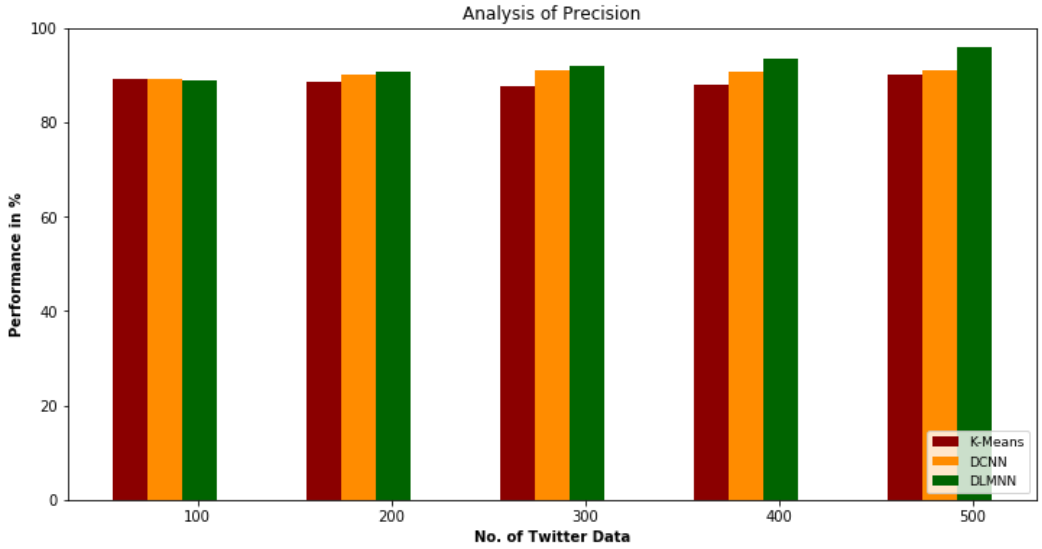

Figure 3.5 Comparative analysis of precision values

3.8.3 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. In the prevailing K-Means, recall values are the least when the data count is 400. The recall decreases when 200 data are utilized and then it gradually rises as the data count increases. From Figure 3.6, it is perceived that the proposed DLMNN acquired better outcomes when weighed against the prevailing KMA and DCNN algorithm.

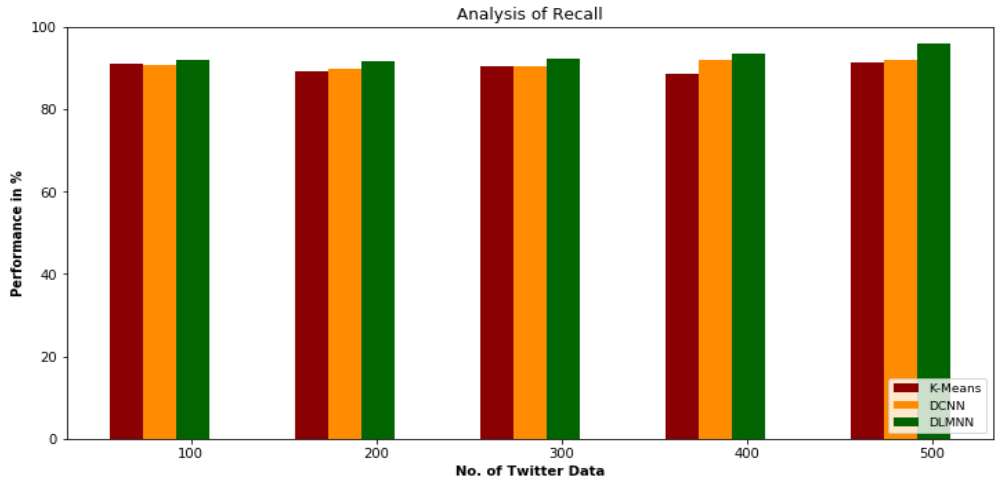

Figure 3.6 Recall Performance of K-Means, DCNN and DLMNN

3.8.4 F-Score

The next performance gauge that is utilized for the comparative examination is the F-score. The F-score of this proposed DLMNN, K-Means and DCNN are calculated and compared. The attained F-score showed a stable increase as the data count increased. The KMA witnessed an unequal increase and decrease on the value of F-score. This variation is elucidated using Figure 3.7.

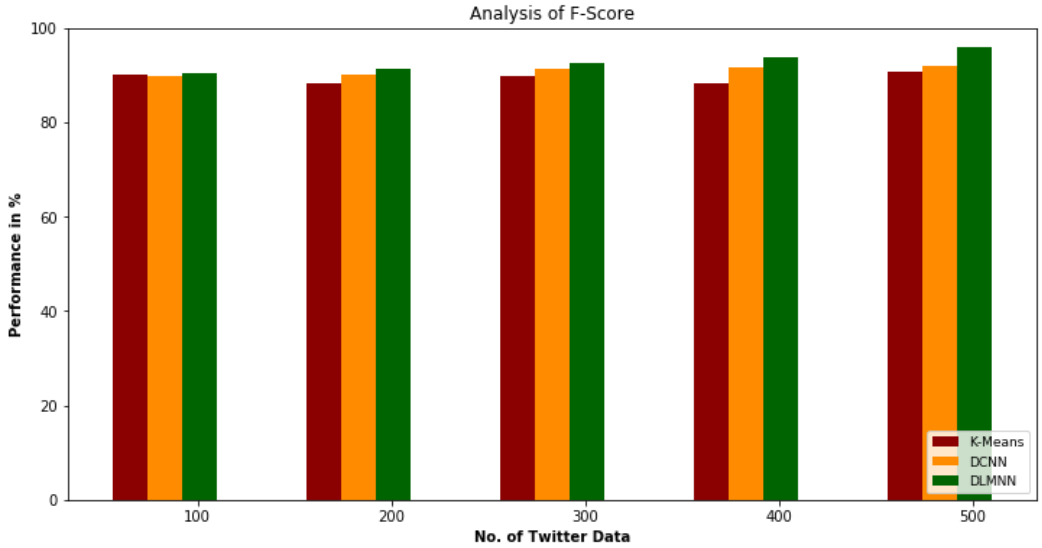

Figure 3.7 F-score of various quantities of data in K-Means, DCNN and DLMNN

Figure 3.7 shows the performances proffered by the proposed DLMNN classifier and the prevailing DCNN and K-Means algorithms centered on F-Score measure. The F-Score performance varies based on the number of data. The data value starts from 100 and end with 500 data. When the data count is 200, the proposed DLMNN achieves above 90% F-score, but the prevailing DCNN and KMA achieves below 90% F-Score. When the data count is 500, the proposed classifier has obtained above 95% F-score but the existing DCNN and K-Means obtained below 90% F-score performance. Similarly, the performance of the system varies for the remaining data counts. Thus, it is deduced that the proposed DLMNN proffers higher performance on considering the prevailing systems.

3.8.5 Time Complexity

It is gauged in milli seconds in this analysis. The proposed DLMNN has an ET of 20.61, 83.71, 238.02, 464.86, 1112.21 for 100, 200, 300, 400 and 500 data, respectively. Whereas DCNN has 22.11, 86.82, 256.23, 502.64, 1260.22 and K-Means has 29.51, 92.11, 292.46, 678.45, 1382.67 respectively. The system's ET with KMA is increased with the elevation in the number of data.

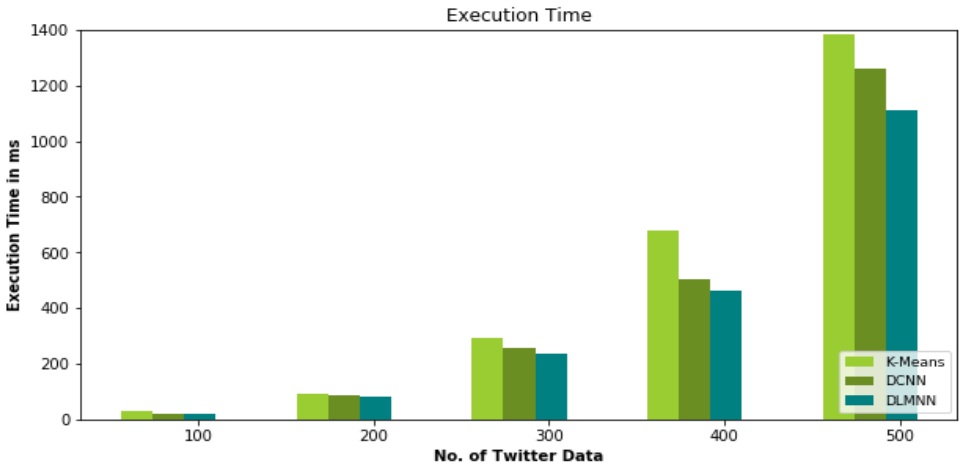

Figure 3.8 Time Complexity of K-Means, DCNN and DLMNN

Figure 3.8, it is inferred that the proposed system exhibited a lower rate of rise in the ET as the data count increased. The Time Complexity is the presented time in seconds. From the observation of this figure, the K-Means takes much time to execute the result than the DCNN and proposed DLMNN. When the data count is 500, the proposed system takes below 1500ms time but the prevailing KMA and DCNN takes above 1500ms time for accomplishing the task. This affirms that this proposed system shows noteworthy efficiency when contrasted to the prevailing works.

3.8.6 Accuracy

The proposed work's accuracy witnessed an increase with the increase in the data count that was taken for analysis. For analyzing 100, 200, 500 data, the proposed DLMNN yielded an accuracy of 83, 84, 86, 88 and 91 percent respectively. When the same specified data was scrutinized utilizing the KMA, the accuracy attained was 82% for 100, 80.5% for 200, 80.33% for 300, 79.5% for 400 and 83% for 500 data. The DCNN algorithm provided an accuracy of 81% for 100, 82% for 200, 83% for 300, 84% for 400 and 85% for 500 data. Figure 3.9, visually elucidates such observations.

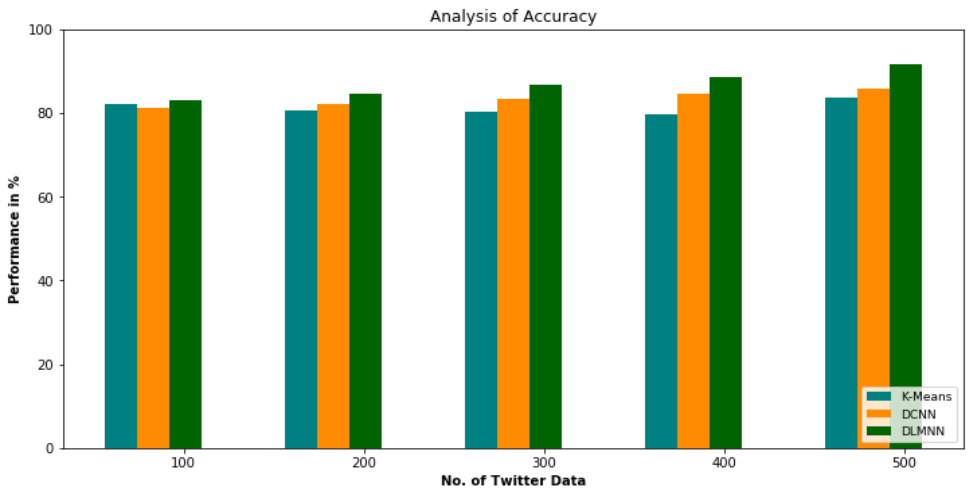

Figure 3.9 Accuracy of the proposed and the existing systems

3.8.7 Average Sentiment Score

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

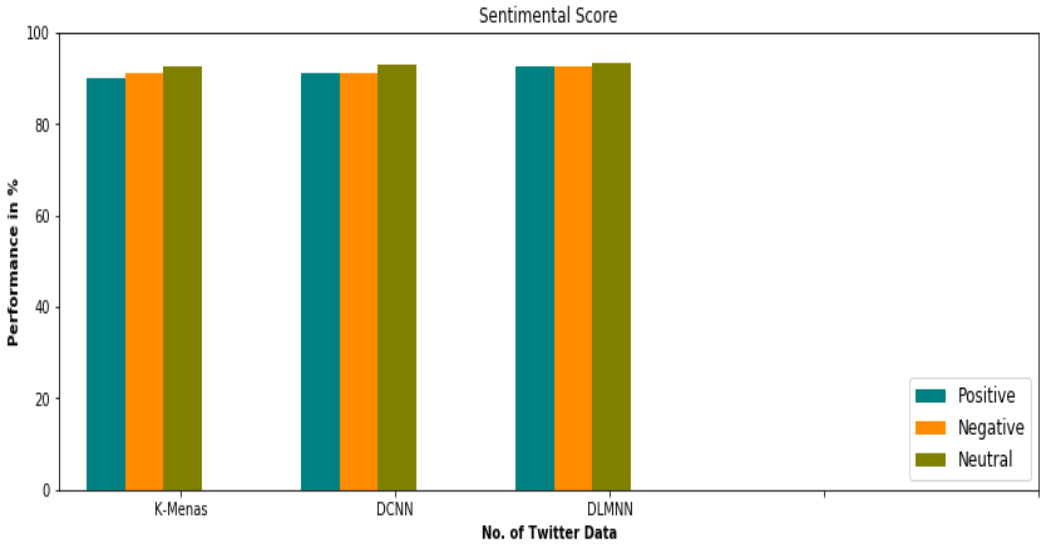

Figure 3.10 Comparative analysis of average sentiment score

Figure 3.10 delineates the comparative examination of the ASS for the proposed and the prevailing works. The average sentiment score performance varies based on the data count. The KMA 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 DLMNN has very few classifications; hence, the ASS is lower on considering other methods that are taken for comparison.

3.9 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 four processes namely tokenization, stemming, removal of stop words and removal of numbers. 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 DLMNN 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 propitious results when contrasted to the KMA and DCNN algorithm. The simulation results demonstrate that the proposed classification structure has proved to be having better classification for online tweet dataset when compared with the existing K-means and DCNN. From the results, it is inferred that the proposed approach of PSO-DLMNN classifier scores over the existing methodologies. Therefore, all the necessary factors like Precision, Recall, F-Score, Accuracy and sentiment score discussed so far give a complete scenario of what is required to improve the Sentimental Analysis of Twitter data by using DLMNN.