International Journal of Advanced Science and Technology
Vol. 29, No.8, (2020), pp. 6283-6300

Feature Selection And Optimized Weight Based Multi-Tier Stacked Ensemble (Owmtse) Classification For Twitter Sentiment Analysis

[1] Dr.Antony selvadoss Thanamani, Head of the department in

computer science (Aided), NGM College, Pollachi, TamilNadu .
[2] Padmapriya P , PhD Research scholar, Department of Computer Science, NGM College, Pollachi, TamilNadu .

Abstract

In recent years, a huge number of people have been attracted to social-networking platforms like Facebook, Twitter and Instagram. Most use social sites to express their emotions, beliefs or opinions about things, places or personalities. Sentiment Analysis (SA) is important to improve particular products (or) topics. There has been lot of work in the field of sentiment analysis of twitter data. The previous system designed an Optimized Weight based Multi-Tier Stacked Ensemble (OWMTSE) framework for twitter sentiment analysis. However, the extracted features may cause a complex computation problem due to overfitting. To deal with such problem, optimal feature selection techniques are required. To solve this problem the proposed system designed a Binary Swallow Swarm Optimization (BSSO) based feature selection and Optimized Weight based Multi-Tier Stacked Ensemble (OWMTSE) classification for Twitter Sentiment analysis. Initially, each tweet is represented by a vector of numbers based on the extracted features. Then the optimal features are selected by using Binary Swallow Swarm Optimization (BSSO) algorithm. Finally, the OWMTSE learning is designed for sentiment classification. In the proposed OWMTSE system, Weighted Majority Voting (WMV) ensemble classifier with Support Vector Machine (SVM), Convolutional Neural Network (CNN) and Passive Aggressive Classifier (PAC) in as base learners. This novel algorithm uses the incremental learners to predict the results that get combined by the classifiers in the next tier. The meta-learning in the next tier generalizes the output from the classifiers to give the final prediction. Results of these classifiers are tuned via the optimized weight via Genetic Algorithm (GA). The voting ensemble model will consider this tweet as positive one because this is the majority decision. The experimental results shows that the proposed system attains higher performance compared with the previous methods interms of accuracy, precision, recall, f-measure and error rate.

Key words: Sentiment Analysis (SA), Binary Swallow Swarm Optimization (BSSO), Weighted Majority Voting (WMV) and Genetic Algorithm (GA).

1. INTRODUCTION

Now days the social networking sites like Twitter, Facebook, MySpace, and YouTube have gained so much popularity. They allow people to build connection networks with other people in an easy and timely way and allow them to share various kinds of information and to use a set of services like picture sharing, blogs, etc. Twitter is the most popular social networking platform over which people express their views and opinions about various trending topics with the help of short messages called as tweets. According to statistics, there are approximately 317 million Twitter users worldwide [1-3]. On an average, around 6000 tweets are tweeted and around 500 million tweets per day. This signifies a massive generation of data in social media. These contents through social media can be beneficial in knowing the opinion and for today's business. These tweets are text messages with maximum length

ISSN: 2005-4238 IJAST

of 140 characters, because of this short message service people make the use of acronyms, emoticons and other special characters with special meanings.

Sentiment analysis also referred to as Opinion mining, is the field of study that analyses people‟s opinion, sentiments, attitudes, evaluations, and emotions through social media data [4-5]. Sentiment analysis involves classifying opinions in text into categories like "positive" or "negative" or "neutral. There are various approaches available for sentiment analysis (SA), only two main groups are widespread. The first group solves the problems of SA by implementing the machine learning approach [6-7]. In this group, multiple techniques are employed in a bid to extract salient features that more accurately give information about the polarity of sentiments. The technique used is constantly monitored, as the process requires manually annotated corpus. The second group uses a linguistically inclined method called the lexicon-based approach. According to reference [8], the investigation is initiated with words or sentences showing characteristics of semantic polarity.

A number of machine learning techniques have been formulated to classify the tweets into classes. Some well-known machine learning techniques include Maximum Entropy, Stochastic Gradient Descent (SGD), Random Forest (RF), SailAil Sentiment Analyzer (SASA), Multi-Layer Perceptron (MLP), Naïve Bayes (NB) [9], Multinomial Naïve Bayes (MNB) and Support Vector Machine (SVM) [11-13] are utilized for sentiment classification. However, SVM does not perform very well when the data set has more noise and the Naive Bayes has issue with zero-frequency problem. The Random Forest requires much time for training. The Multi-Layer Perceptron (MLP) is inefficient because there is redundancy in such high dimensions.

The proposed research work concentrates on both feature selection and classification methods for utilizing twitter data. The extracted features are selected by using Binary Swallow Swarm Optimization (BSSO) and the classification is done by using Optimized Weight based Multi-Tier Stacked Ensemble (OWMTSE) learning. These methods are implemented via the collected dataset of sentiment140 dataset. The tweets have been annotated (0 = negative, 2 = neutral, 4 = positive) and they can be used to detect sentiment.

2. LITERATURE REVIEW

Saad and Yang (2019) presented ordinal regression methods to perform a detailed sentiment analysis of tweets. The designed approach consists of first pre-processing tweets and using a feature extraction method that creates an efficient feature. Then, under several classes, these features scoring and balancing. Multinomial logistic regression (SoftMax), Support Vector Regression (SVR), Decision Trees (DTs), and Random Forest (RF) algorithms are used for sentiment analysis classification in the presented framework. Experimental findings reveal that the designed approach can detect ordinal regression using machine learning methods with good accuracy. Moreover, results indicate that Decision Trees obtains the best results outperforming all the other algorithms [14].

Arora and Kansal (2019) designed a text normalization with deep convolutional character level embedding (Conv-char-Emb) neural network model for SA of unstructured data. This model can tackle the problems: (1) processing the noisy sentence for sentiment detection (2) handling small memory space in word level embedded learning (3) accurate sentiment analysis of the unstructured data. The initial preprocessing stage for performing text normalization includes the following steps: tokenization, Out Of Vocabulary (OOV) detection and its replacement, lemmatization and stemming. A character-based embedding in Convolutional Neural Network (CNN) is an effective and efficient technique for SA that uses less learnable parameters in feature representation. The experimental results are evaluated in the Twitter dataset by a different point polarity (positive, negative and neutral).

ISSN: 2005-4238 IJAST

As a result, our model performs well in normalization and sentiment analysis of the raw Twitter data enriched with hidden information [15].

Kumar et al (2019) designed the Bidirectional Long Short Term Memory (Bi-LSTM) for tweets classification. Usually, the noises such as URLs, positive and negative emoji, stop words are reduced from raw tweets in the pre-processing stage. The Twitter Sentiment Analysis (TSA) is done by feature extraction approach to pre-processed twitter data for extracting useful information. The Bi-LSTM obtained the three types of results such as positive, negative and neutral to validate the TSA for Sanders dataset. The presented Bi-LSTM approach achieved 90.04% of accuracy, 88.12% of precision, 92.31% of recall and 90.17% of F-Measure which are higher than the existing methods such as Support Vector Machine (SVM), Neural Network (NN) [16].

Pethalakshmi (2020) presented a new method for twitter sentiment analysis using dempster shafer algorithm and multiclass SVM Classifier. The designed system consists of Data cleaning, Preprocessing, Feature extraction, Feature Selection and Classification phases. In the data cleaning stage, perform four major processes: URL removal, Username Removal, Punctuation Removal and Spell Correction. To increase the accuracy, adopt preprocessing where tokenization, stop word removal, lemmatization and stemming, acronyms expansion, slangs correction, split attached word, and POS tagging. In order to improve the classification accuracy, execute the feature extraction process where eight features are extracted. Then select the best features from the extracted features using the Dempster Shafer algorithm. Finally, Sentiments are classified using the One against All-multiclass Support Vector Machine (OA2 -SVM) algorithm. It classifies sentiments into five classes: Strongly Positive, Strongly Negative, Positive, Negative and Neutral. From the comparison results, it perceived that the designed method enhances accuracy and precision, recall and F-Measure [17].

Xia et al (2011) presented an ensemble framework to sentiment classification tasks, with the aim of efficiently integrating different feature sets and classification algorithms to synthesize a more accurate classification procedure. First, two types of feature sets are designed for sentiment classification, namely the part-of-speech based feature sets and the word-relation based feature sets. Second, three well-known text classification algorithms, namely Naive Bayes (NBs), Maximum Entropy (ME) and Support Vector Machines (SVMs), are employed as base-classifiers for each of the feature sets. Third, three types of ensemble methods, namely the fixed combination, weighted combination and meta-classifier combination, are evaluated for three ensemble strategies. Finally, some in-depth discussion is presented and conclusions are drawn about the effectiveness of ensemble technique for sentiment classification [18].

3. PROPOSED METHODOLOGY

In this proposed research work, introduced an Optimized Weight based Multi-Tier Stacked Ensemble (OWMTSE) classification for twitter sentiment analysis. The flow diagram of the proposed work is shown in figure 1.

ISSN: 2005-4238 IJAST

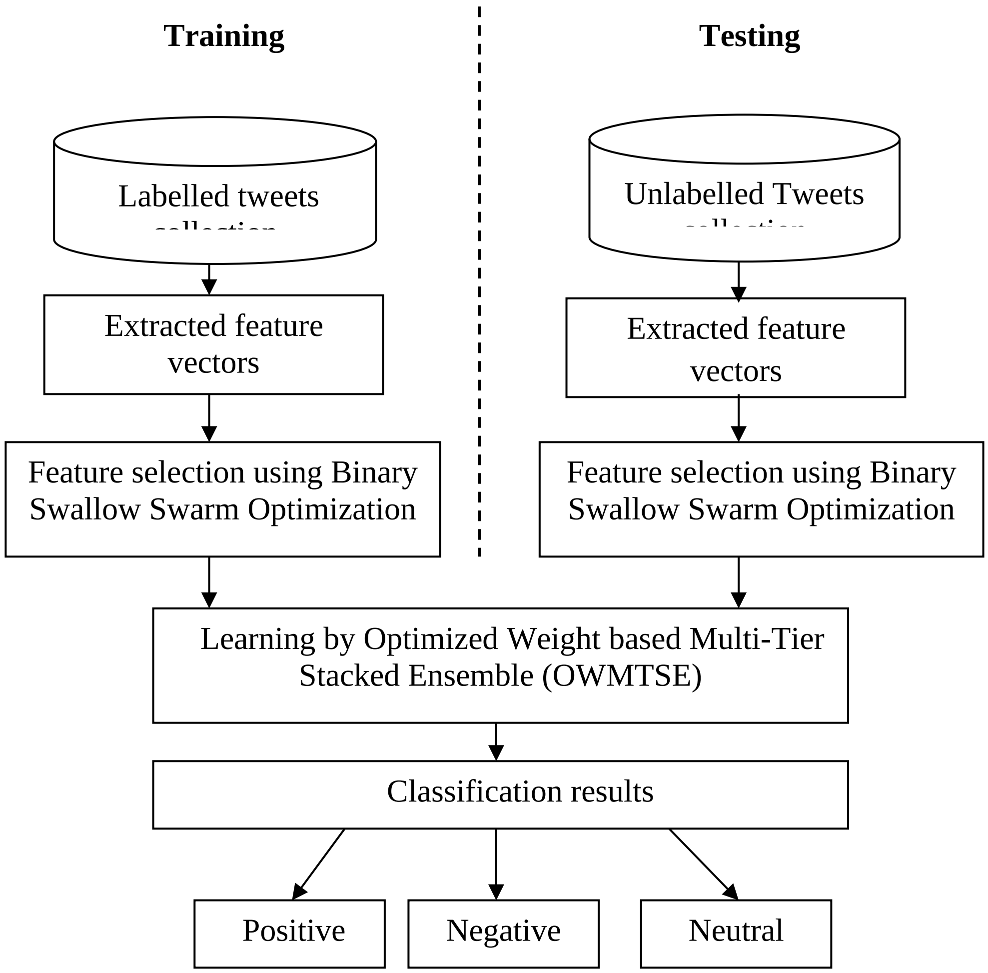

Figure 1: Flow diagram of the proposed work

3.1 Input

Let $D = \{(x_1, y_1), (x_2, y_2), \cdots, (x_N, y_N)\}$ be dataset, where $x_i \in X, y_i \in Y = \{c_1, c_2, \ldots, c_l\}$ and N is the complete number of occurrences in dataset and $c_1, c_2, \ldots, c_l$ are class labels. Regularly, input space X comprises of numerous features and henceforth its components are represented by N-tuple of 'd' measurements.

$$X = \{(x_{11}, x_{12}, \cdots, x_{1d}), \cdots, (x_{N1}, x_{N2}, \cdots, x_{Nd})\} \ldots . (1)$$

The features extracted from the dataset. The extracted features are considered as an input.

3.2 Feature selection using Binary Swallow Swarm Optimization (BSSO)

In this work, Binary Swallow Swarm Optimization (BSSO) technique is used for feature selection. The Swallow swarm optimization is a population-based metaheuristic algorithm. At the beginning of each iteration, the population is sorted based on the value of the objective function. Then, the following roles are assigned:

1. Head leader is a particle with the best value of the objective function;

ISSN: 2005-4238 IJAST

2. Local leaders are l particles that follow the head leader in accordance with the value of the objective function;

3. Aimless particles are k particles with the worst value of the objective function;

4. Explorers are all other particles.

This algorithm represents solutions as vectors S that encode features. At the first step of the algorithm, a population (number of features) S is generated. The number of vectors in the population is a preset integer, which is also referred to as the population size. For each feature vector, a measure of classification accuracy E is computed. On each iteration, all the feature vectors are sorted in descending order of accuracy. The first element becomes the head leader. The next l solutions are local leaders, n worst vectors are aimless particles, and all other feature vectors are explorer particles. The integer variables *l* and *n* are also specified beforehand. Explorer particles change their positions based on the positions of the leaders, while aimless particles do so randomly. Below are the corresponding formulas for explorers:

$$S_e\ (t+1) = \text{merge}\ (V(t+1), S_e(t), p_{ve}),$$

$$V(t+1) = \text{merge}\ (VHL(t+1), VLL\ (t+1), p_{vhl}),$$

$$VHL(t+1) = \text{merge}\ (\text{merge}\ (SHL(t), S_e(t), p_{he}), \text{rand}\{0,1\}^D, p_{her}),$$

$$VLL\ (t+1) = \text{merge}\ (\text{merge}(SLL\ (t), S_e(t), p_{le}), \text{rand}\{0,1\}^D, p_{ler}) \quad (2)$$

Where, SHL is the position of the head leader, SLL is the position of the local leader, S_e is the position of the explorer, VHL is the velocity vector with respect to the head leader, VLL is the velocity vector with respect to the nearest local leader, V is the common velocity vector, p_{ve} is the effect of the velocity vector on the position of the explorer, p_{vhl} is the effect of the velocity vector with respect to the head leader on the velocity vector with respect to the local leader, p_{he} is the effect of the head leader's position on the position of the explorer, p_{her} is the combined effect of the head leader and explorer on a random vector, p_{le} is the effect of the local leader's position on the position of the explorer, and p_{ler} is the combined effect of the local leader and explorer on a random vector. A pseudo code of the BSSO is shown in algorithm 1.

Algorithm 1: Binary Swallow Swarm Optimization (BSSO)

Input: Number of feature vectors

Output: Selected features

Parameters: iterations -maximum number of cycles, N – population size, D – dimension, p_{vs}, p_{vhl}, p_{he}, p_{her}, p_{le} pler, l – local leaders, k – aimless particles.

1. begin
2. for i ← to N do
3. Initialize feature vectors S^i
4. Select j-th (j = 1,2,…,D) feature for subset
5. Evaluate fitness value ($fitness_i$) of Subset;

ISSN: 2005-4238 IJAST

6. end
7. SHL ← S^k, where k= $\min_{i=1,2,..,N} fitness_i$
8. for iter←1 to iterations do
9. for i ← to N do
10. Find nearest local leader SLL among feature vectors $\{S^1, S^2, \ldots, S^N\}$;
11. Evolve a new solution $S_e=\{f_1, f_2, \ldots, f_D\}$ using Equation (2);
12. Select j-th feature (j = 1,2,…,D) for subset Sub, where Se_j= 1;
13. Build reduced dataset based on Subset;
14. Evaluate fitness value ($fitness_i$) of Sub: $fitness_i$ ← errorrate;
15. if $fitness_{se} < fitness_i$ then
16. S^i ← S_e;
17. $fitness_i$ ←$fitness_{se}$;
18. end
19. end
20. SHL ← S^k, where k= $\min_{i=1,2,..,N} fitness_i$
21. end
22. end

3.3 Classification using Optimized Weight based Multi-Tier Stacked Ensemble (OWMTSE)

In this work, classification is performed by using Optimized Weight based Multi- Tier Stacked Ensemble (OWMTSE) algorithm. The proposed OWMTSE consists of three tiers such as base tier, ensemble tier and incremental learners. The base tier is the first tier. It uses a set of incremental classifiers with the partial fit as weak learners. These learners produce the cross-validated predictions. The process of producing the cross validated predictions using the Support Vector Machine (SVM), Convolutional Neural Network (CNN) and Passive Aggressive Classifier (PAC) classifiers. The ensemble tier uses Plural Voting, Majority Voting, Weighted Majority Voting (WMV) and Confidence-Based Voting scheme. The WMV assigns the weights based on the class probabilities of the incremental learners. The optimized weights are achieved via Genetic Algorithm (GA).

ISSN: 2005-4238 IJAST

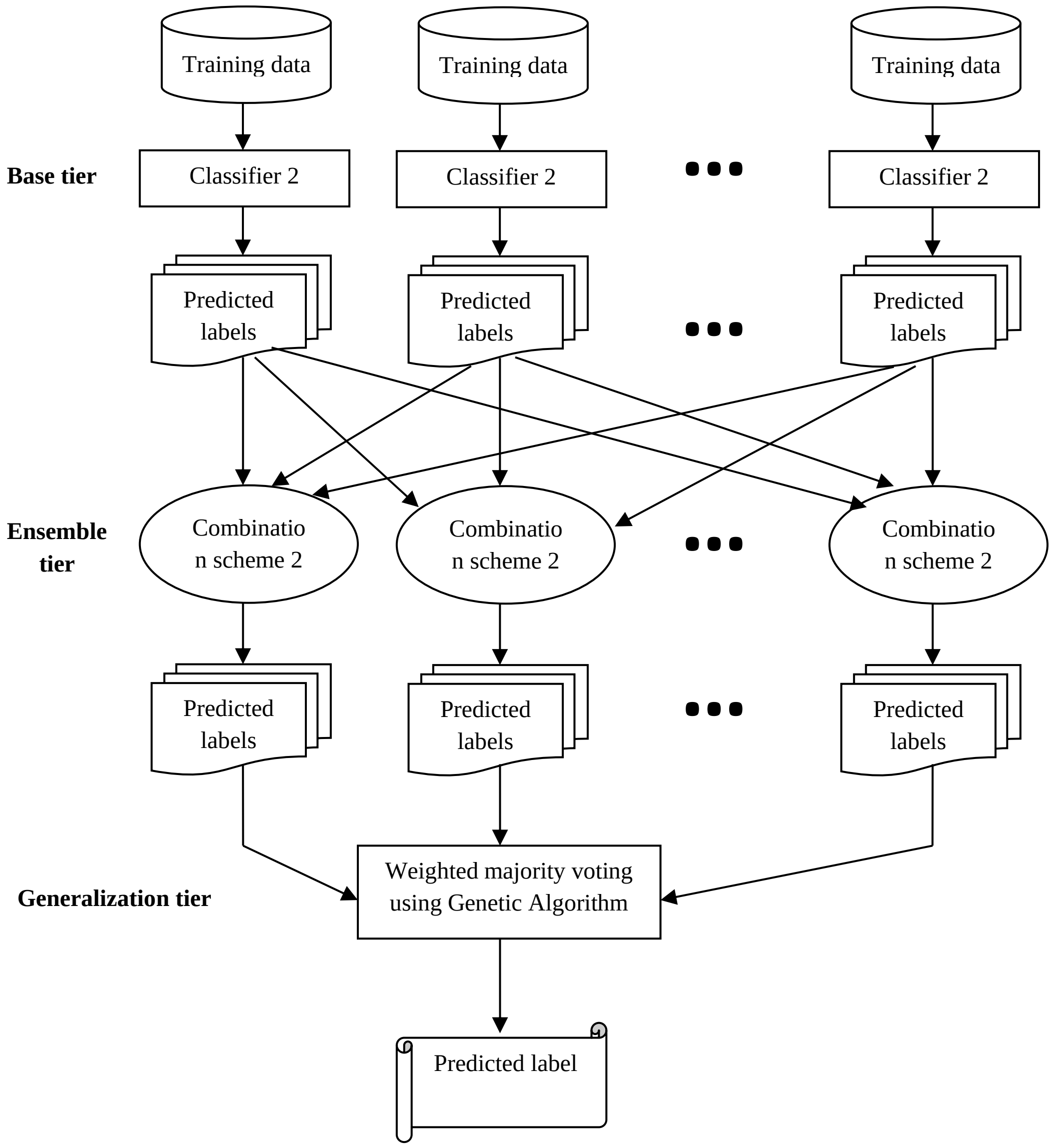

Figure 2: Depiction of data processing at different tiers of Optimized Weight based Multi- Tier Stacked Ensemble (OWMTSE)

A. Support Vector Machine (SVM)

The Support Vector Machine (SVM) is a supervised machine learning algorithm used for data classification. The basic principle of SVM is to search for optimal hyperplane with maximal distance of the nearest samples from each class. In this proposed system, sentiment 140 dataset is taken as an input. Given a set of tweets $(x_1, y_1), (x_2, y_2) \dots, (x_l, y_l)$ where $x_i(x_i \in R^N)$ is the input vector of N dimension and y_i is its label (positive, negative and neutral) for a classification problem. Decision function for SVM is

$$D(x) = w \ x + b \qquad (3)$$

Where, w is weighted vector, b is a scalar. The optimal classifying plane and the support vectors are depicted in Figure 3. The separating hyper plane satisfies:

ISSN: 2005-4238 IJAST

$$y_i(w\, x_i + b) \geq 1 \text{ for } i=1,\ldots,M \quad (4)$$

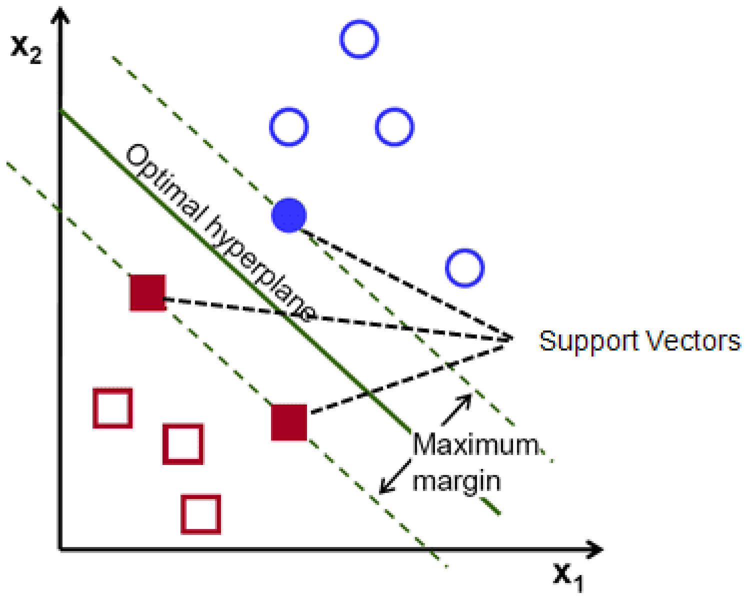

Figure 3 : Separating hyperplane

SVMs solve the optimization problem

$$\min_{w,b} \frac{1}{2}||w||^2 + C\sum_{i=1}^{m} \xi_i \qquad (5)$$

$$\text{subject to } y_i(w\, \varphi(x_i) + b) \geq 1 - \xi_i \text{ , i=1,,…,m}$$

C - Penalty parameter for the error

ξ_i – Slack variable

If the data points are too close, indeed, if it is difficult to separate them directly, it is possible to use a *kernel* function *K* to separate them. That is,

$$\mathrm{F}(\alpha) = \sum_{i=1}^{m} \alpha_i - \frac{1}{2}\sum_{j,k=1}^{m} \alpha_j \alpha_k y_j y_k K(x_j, x_k) \quad (6)$$

subject to

$$\sum_{i=1}^{m} y_i \alpha_i = 0, C \geq \alpha_i \geq 0, \quad \text{i=1,…,m}$$

Where, $K(x_j, x_k)$ is the kernel function.

The hyperplane is used to separating the tweets as negative, neutral, positive they can be used to detect sentiment

B. Convolutional Neural Network (CNN)

ISSN: 2005-4238 IJAST

In this proposed research work, Convolutional Neural Network (CNN) is utilized for sentiment analysis. Basically, this type of networks mainly consists of three layers: convolution layers, sub sampling or pooling layers, and fully connected layers. The network has input layer which takes selected features as the input, output layer from where the system get the trained output and the intermediate layers called as the hidden layers which is shown in figure 4.

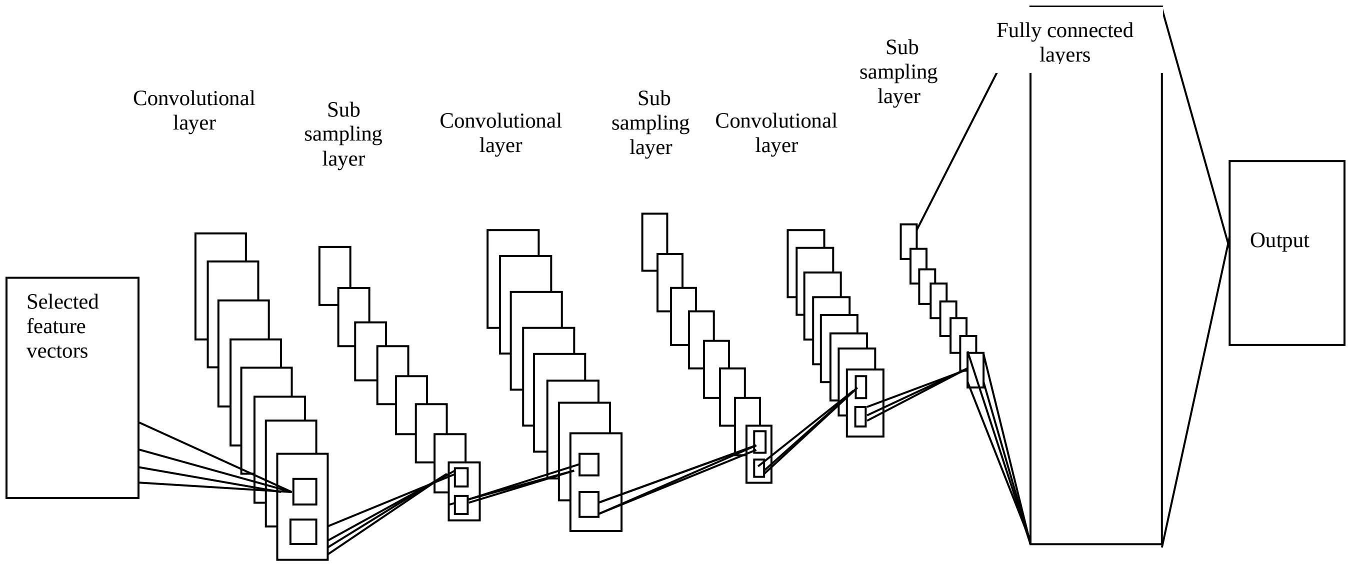

Figure 4: Convolutional Neural Network

Convolution layer

Convolutional layer has 16 kernels. Each vector of the input matrix is independently convolved with the kernel and generated the output. The result of the convolution of the input and kernel is used to generate n output. Generally, a kernel of the convolution matrix is referred to as a filter while the output obtained by convolving kernel and the input are referred to as maps.

The CNN can include three convolutional layers, the inputs and outputs of next convolutional layers are the feature maps. There is a bunch of n filters in each convolution layer. These filters are convolved with the input, and the depth of the generated maps (n∗) is equivalent to the number of filters applied in the convolution operation. Note that each filter map is considered as a specific features at a certain location of the input.

The output of the l-th convolution layer, denoted as $C_j^{(l)}$, consists of maps. It is computed as

$$C_i^{(l)} = B_i^{(l)} + \sum_{j=1}^{a_i^{(l-1)}} K_{i,j}^{(l-1)} * C_j^{(l)} \quad (7)$$

Where, $B_i^{(l)}$ is the bias matrix and $K_{i,j}^{(l-1)}$ is convolution filter that connects the j-th feature map in layer $(l-1)$ with the i-th feature map in the same layer. The output $C_i^{(l)}$ layer consists of feature maps. In (7), the first convolutional layer $C_i^{(l-1)}$ is input space, that is, $C_i^{(0)} = X_i$.

ISSN: 2005-4238 IJAST

The kernel generates frame map. After the convolution layer, the activation function can be applied for nonlinear transformation of the outputs of the convolutional layer:

$$Y_i^{(l)} = Y(C_i^{(l)}) \quad (8)$$

Where, $Y_i^{(l)}$ is the output of the activation function and $C_i^{(l)}$is the input that it receives.

Typically used activation functions are sigmoid, tanh, and rectified linear units (ReLUs). In this work, ReLUs which is denoted as $Y_i^{(l)} = \max(0, Y_i^{(l)})$ are used. This function is popularly used in deep learning models due to its help in reducing the interaction and nonlinear effects. ReLU converts the output to 0 if it receives a negative input, while it returns the same input value if it is positive. The advantage of this activation function over other functions is the faster training because of the error derivative, which becomes very small in the saturating region; therefore, the updates of the weights almost vanish. This is called the vanishing gradient problem.

Sub sampling or pooling Layer

The sub sampling layer comes after the convolutional layer. The main aim of this layer is to spatially reduce the dimensionality of the frame maps extracted from the previous convolution layer. The outputs of the previous convolutional layer have 120 feature maps with 1x1 size. Global pooling layer plays a very important role in deep convolutional neural networks. .

Fully Connected layer

The output layer uses Softmax activation function:

$$Y_i^{(l)} = f(z_i^{(l)}), \text{ where } z_i^{(l)} = \sum_{i=1}^{m_i^{(l-1)}} w_H y_i^{(l-1)} \quad (9)$$

where w_H are the weight value that should be tuned by the complete fully connected layer in order to form the representation of each class and f is the transfer function which represents the nonlinearity.

C. Passive Aggressive Classifier (PAC)

The Passive Aggressive Algorithm is an online algorithm; ideal for classifying massive streams of data (e.g. twitter). It is easy to implement and very fast. Its purpose is to make updates that correct the loss, causing very little change in the norm of the weight vector**.**

Weighted Majority Voting (WMV)

The Weighted Majority Voting (WMV) is a decision-making method retrieved from several classifiers that are separated and independent by giving each classifier more abilities. Let us consider χ a set of N examples and C a set of Q classes. Let us define an algorithm set $S = \{A_1, A_2, A_M\}$ which contains the M classifiers used for the voting. Each example $x \in \chi$ is assigned to have one of the Q classes. Each classifier will have its prediction for each example. The final class assigned to each example is the class predicted by the majority of classifiers (gaining the majority votes) for this example. In WMV, each vote is weighted by the prediction accuracy value of the classifier that is denote as Acc. The count of total votes for a class c_k can then be defined as:

ISSN: 2005-4238 IJAST

$$T_k = \sum_{l=1}^{M} Acc\ (A_l) \times F_k(c_l) \qquad (10)$$

$$F_k(c_l) = \begin{cases} 1 & c_l = c_k \\ 0 & c_l \neq c_k \end{cases} \quad (11)$$

where c_l and c_k are the classes of C. The class receiving the greatest total weight is chosen. In general all classifiers are trained on different independent training sets and weights are assigned which leads to highest classification rate to classify the tweets as positive, negative and neutral. The weights of Weighted Majority Voting (WMV) ensemble are optimized through GA so that the system could predict the correct sentiment.

GENETIC ALGORITHM (GA)

In this proposed research work, the weights of the classifiers are optimized by using Genetic algorithms (GA) to classify the sentiment. The GAs are a robust and efficient optimization technique based on natural evolutionary theory. In GAs the parameters of the search space are encoded in the form of strings called chromosomes. A collection of such chromosomes is called a population. Initially, a random population is created, which represents different points in the search space. An objective or fitness function is associated with each string that represents the degree of goodness of the string. Based on the principle of survival of the fittest, a few of the strings are selected and each is assigned a number of copies that go into the mating pool. Biologically inspired operators like crossover and mutation are applied on these strings to yield a new generation of strings. The processes of selection, crossover and mutation continue for a fixed number of generations or till a termination condition is satisfied.

1. Initial Population

Initially, a random population are initialized. In this work, weights of all classifiers are normalized between [0-1] based on the actual and predicted results and length of chromosome m equals to number of classifiers l. So that l= m.

2. Fitness Evaluation

In this work randomly initialized population of chromosomes is evaluated according to fitness function which is the accuracy of classifiers.

3. Reproduction

Tournament selection is used to generate the new population. In each new generation elitism policy is adopted where the numbers of best chromosomes (weights of the classifier) are taken to the next generation without any competition, this will help maintain the best accuracy achieved throughout the evolution process.

4. Crossover

Then cross-over and mutation operations are applied to evolve the weights of ensemble classifiers. Once the partner of a chromosome is identified, the two chromosomes perform crossover at a random location to obtain a new chromosome, which is stored in the next generation.

5. Mutation

ISSN: 2005-4238 IJAST

Here, perform the new weight value obtained from the crossover of two classifiers usually has some duplicate indices.

Algorithm 2: GA

Step 1: Begin

Step 2: t=0

Step 3: Initialize population (weights of classifiers) P(t)

Step 4: for i=1 to popsize

Step 5: Compute the classification accuracy

Step 6: t=t+1

Step 7: if termination criterion achieved go to step 13

Step 8: Select (P)

Step 9: Crossover (p)

Step 10: Mutation (P)

Step 11: Go to step 3

Step 12: Obtain best weight values and stop

Step 13: End

4. EXPERIMENTAL RESULTS

This section shows the performance comparison results of various methods. The proposed system is implemented and its metrics is measured against dataset in the area of twitter sentiment analysis. All the experiments were conducted using a machine with IntelR CoreTM i7-3770 CPU @ 3.40 GHz and 8.00 GB memory, running 64-bit Windows 7. The dataset used for implementation is sentiment 140 dataset. It contains 1,600,000 tweets extracted using the twitter API. The tweets have been annotated (0 = negative, 4 = positive) and they can be used to detect sentiment. It contains the following 6 fields: target: the polarity of the tweet (0 = negative, 2 = neutral, 4 = positive), ids: The id of the tweet (2087), date: the date of the tweet (Sat May 16 23:58:44 UTC 2009), flag: The query (lyx). If there is no query, then this value is NO_QUERY, user: the user that tweeted (robotickilldozr) and text: the text of the tweet (Lyx is cool).

4.1 Results comparison

The performance of the proposed OWMTSE classifier is compared with the existing

classifiers such as Navie Bayes (NB), K Nearest Neighbour(KNN), Multi Layer Perceptron (MLP), Weighting Naive Bayes (WNB), MTSE interms of accuracy, precision, recall, f-measure and error. In order to measure the

ISSN: 2005-4238 IJAST

classifiers results, confusion matrix is formed with four categories such as True Positive (TP), True Negative (TN), False Positive (FP) and False Negative (FN).

Table1: Performance comparison

Metrics	Methods					
	KNN	**NB**	**MLP**	**WNB**	**MTSE**	**OWMTSE**
Accuracy	83.79	85.22	87.37	88.1	91.85	94.5
Precision	82.54	87.14	89.14	92.35	94.85	95.5
Recall	84.54	89.14	90.14	93.25	95.2	96.5
F-Measure	86.2	89.26	91.14	94.25	96.8	97.25
Error	16.21	14.78	12.63	11.9	8.15	5.5

1. Accuracy

Accuracy is defined as the proportion of correctly classified instances over the whole set of instances.

$$\text{Accuracy} = \frac{TP+TN}{TP+FP+FN+TN} \quad (12)$$

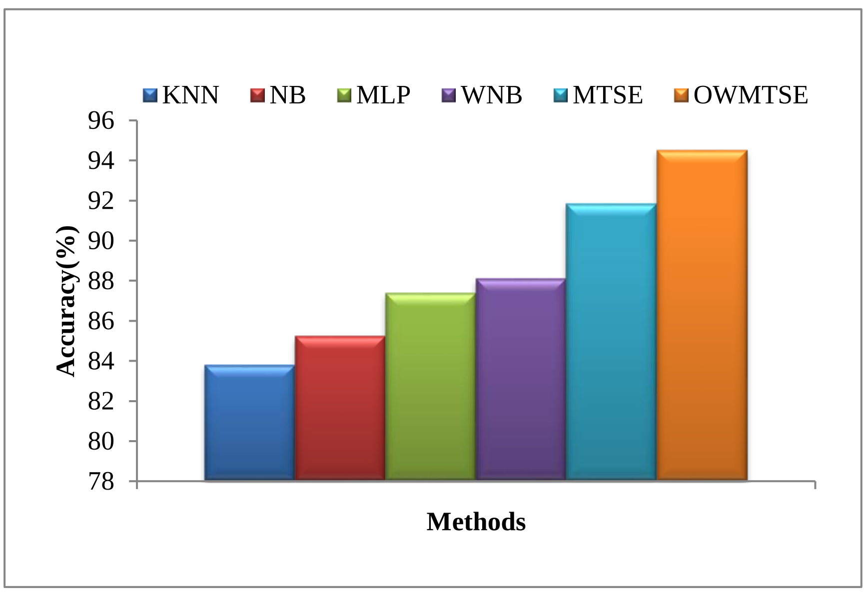

FIGURE 5. ACCURACY COMPARISON VS. CLASSIFIERS

Accuracy comparison results of various classifiers such as KNN, NB, MLP, WNB , MTSE and proposed OWMTSE classifier are shown in the figure 5. It shows that the proposed OWMTSE classifier has gives higher accuracy results of 94.5% which is 10.71%, 9.28%, 7.13%, 6.4%, and 2.65% higher when compared to KNN, NB, MLP, WNB, and MTSE methods respectively.

ISSN: 2005-4238 IJAST

2. Precision

Precision is the ratio of correctly predicted positive observations to the total predicted positive observations.

$$\text{Precision} = \frac{\text{TP}}{\text{TP+FP}} \quad (13)$$

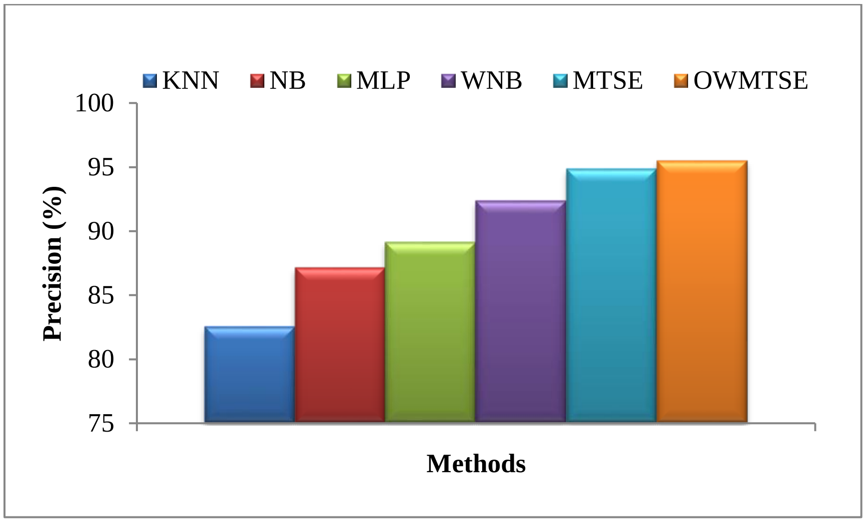

FIGURE 6. PRECISION COMPARISON VS. CLASSIFIERS

Precision of the proposed OWMTSE classifier is compared with the previous classifiers such as KNN, NB, MLP, WNB, MTSE which is shown in the figure 6. From the experimental results, it can be concluded that the proposed system attains 95.5% of precision whereas KNN, NB, MLP, WNB, MTSE attains 82.54 %, 87.14%, 89.14%, 92.35% and 94.85% respectively.

3. Recall

Recall is the ratio of correctly predicted positive observations to the all observations in actual class - yes.

$$\text{Recall} = \frac{\text{TP}}{\text{TP+FN}} \quad (14)$$

ISSN: 2005-4238 IJAST

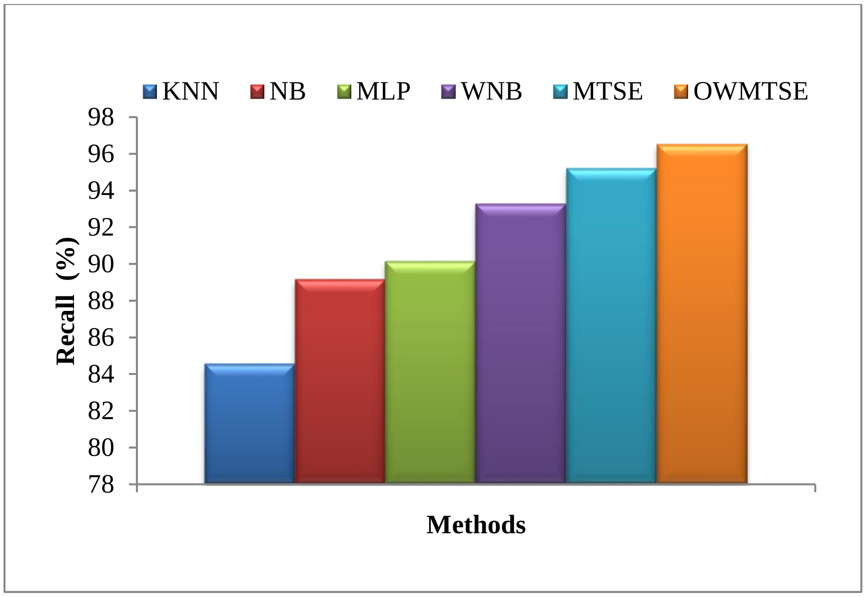

FIGURE 7. RECALL COMPARISON VS. CLASSIFIERS

Recall of the proposed OWMTSE classifier is compared with the previous classifiers such as KNN, NB, MLP, WNB, MTSE which is shown in the figure 7. From the experimental results, it can be concluded that the proposed system attains 96.5% of precision whereas KNN, NB, MLP, WNB, MTSE attains 84.54%, 89.14%, 90.14%, 93.25% and 95.2% respectively.

4. F-measure

F-measure is defined as the harmonic mean of precision and recall.

$$\text{F-measure} = 2*\frac{(Recall*Precision)}{(Recall+Precision)} \quad (15)$$

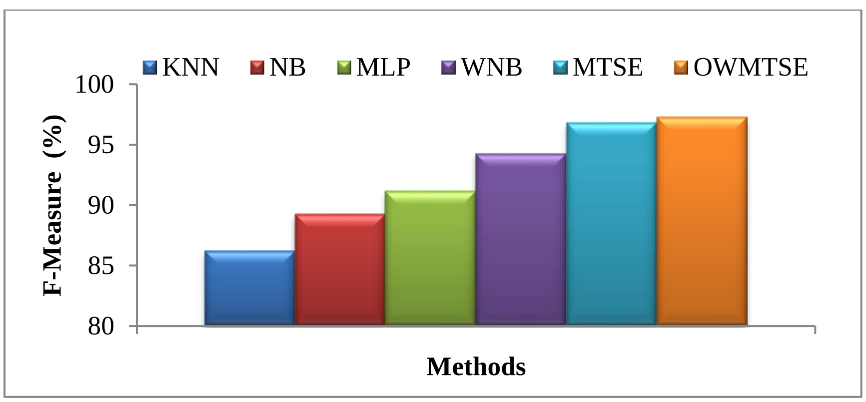

FIGURE 8. RECALL COMPARISON VS. CLASSIFIERS

Figure 8 represents the f-measure proposed OWMTSE classifier is compared with the previous classifiers such as KNN, NB, MLP, WNB, MTSE. From the experimental results, it can be concluded that the proposed system

ISSN: 2005-4238 IJAST

attains 97.25% of precision whereas KNN, NB, MLP, WNB, MTSE attains 86.2%, 89.26%, 91.14%, 94.25% and 96.8% respectively.

5. Error

Error is defined as the proportion of incorrectly classified instances over the whole set of instances.

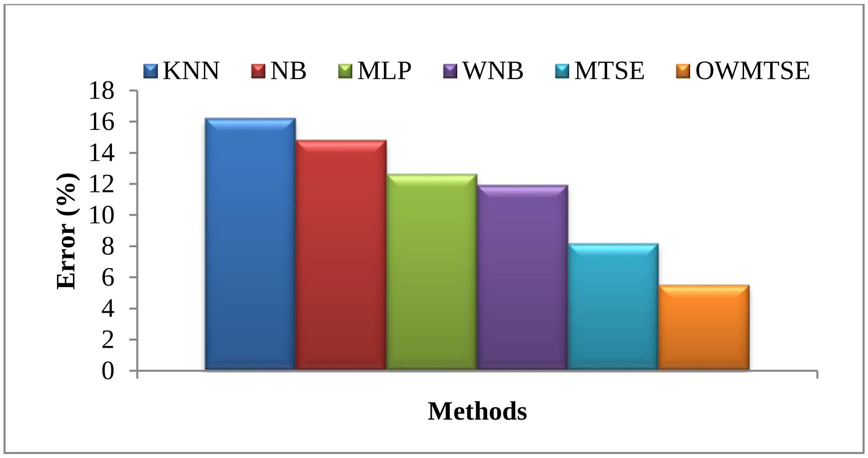

FIGURE 9. ERROR COMPARISON VS. CLASSIFIERS

Error comparison results of various classifiers such as KNN, NB, MLP, WNB, MTSE and proposed OWMTSE classifier are shown in the figure 9. It shows that the proposed OWMTSE classifier has gives lesser error results of 5.5% which is 16.21%, 14.78%, 12.63%, 11.9%, and 8.15% higher error when compared to KNN, NB, MLP, WNB, and MTSE methods respectively.

5. CONCLUSION

The proposed system designed Binary Swallow Swarm Optimization (BSSO) based feature selection and Optimized Weight based Multi-Tier Stacked Ensemble (OWMTSE) sentiment classification in Twitter. In order to achieve higher classification accuracy, the optimal features are selected by using Binary Swallow Swarm Optimization (BSSO) algorithm. Then the Optimized Weight based Multi-Tier Stacked Ensemble (OWMTSE) learning is utilized for classifying the tweets into positive, negative and neutral label sentiment. The proposed OWMTSE includes Support Vector Machine (SVM), Convolutional Neural Network (CNN) and Passive Aggressive Classifier (PAC). The Weighted Majority Voting (WMV) assigns the weights based on the class probabilities of the incremental learners. Results of these classifiers are tuned via the optimized weight using Genetic Algorithm (GA). In the classification phase, the trained classification model will assign positive, negative and neutral label to the new unlabeled tweets. The experimental results shows that the proposed system attains higher performance compared with the previous methods interms of accuracy, precision, recall, f-measure and error rate.

References

[1] Han, K. X., Chien, W., Chiu, C. C., & Cheng, Y. T. (2020). Application of Support Vector Machine (SVM) in the Sentiment Analysis of Twitter DataSet. *Applied Sciences*, *10*(3), 1125.

ISSN: 2005-4238 IJAST

[2] Le, B., & Nguyen, H. (2015). Twitter sentiment analysis using machine learning techniques. In *Advanced Computational Methods for Knowledge Engineering* (pp. 279-289). Springer, Cham.

[3] Ahmad, M., Aftab, S., & Ali, I. (2017). Sentiment analysis of tweets using svm. *Int. J. Comput. Appl*, *177*(5), 25-29.

[4] Neethu, M. S., & Rajasree, R. (2013, July). Sentiment analysis in twitter using machine learning techniques. In *2013 Fourth International Conference on Computing, Communications and Networking Technologies (ICCCNT)* (pp. 1-5). IEEE.

[5] Saif, H., He, Y., Fernandez, M., & Alani, H. (2016). Contextual semantics for sentiment analysis of Twitter. *Information Processing & Management*, *52*(1), 5-19.

[6] Ramadhani, A. M., & Goo, H. S. (2017, August). Twitter sentiment analysis using deep learning methods. In *2017 7th International annual engineering seminar (InAES)* (pp. 1-4). IEEE.

[7] Jianqiang, Z., Xiaolin, G., & Xuejun, Z. (2018). Deep convolution neural networks for twitter sentiment analysis. *IEEE Access*, *6*,pp. 23253-23260.

[8] P. Arora, A. Bakliwal and V. Varma "Hindi subjective lexicon generation using WordNet graph traversal". International Journal of Computational Linguistics and Applications, 3(1): 25-39, 2012.

[9] M. Taboada, J. Brooke, M. Tofiloski, K. Voll and M. Stede Lexiconbased methods for sentiment analysis. Computational linguistics, 37(2): 267-307, 2011

[10] Zainuddin, N., & Selamat, A. (2014, September). Sentiment analysis using support vector machine. In *2014 international conference on computer, communications, and control technology (I4CT)* (pp. 333-337). IEEE.

[11] Varghese, R., & Jayasree, M. (2013, August). Aspect based sentiment analysis using support vector machine classifier. In *2013 international conference on advances in computing, communications and informatics (ICACCI)* (pp. 1581-1586). IEEE.

[12] BholaneSavita, D., & Gore, D. (2016). Sentiment analysis on twitter data using support vector machine. *International Journal of Computer Science Trends and Technology (IJCST)–Volume*, *4*, 365.

[13] Celikyilmaz, A., Hakkani-Tür, D., & Feng, J. (2010, December). Probabilistic model-based sentiment analysis of twitter messages. In *2010 IEEE Spoken Language Technology Workshop* (pp. 79-84). IEEE.

[14] Saad, S. E., & Yang, J. (2019). Twitter sentiment analysis based on ordinal regression. IEEE Access, 7, pp.163677-163685.

[15] Arora, M., & Kansal, V. (2019). Character level embedding with deep convolutional neural network for text normalization of unstructured data for Twitter sentiment analysis. Social Network Analysis and Mining, 9(1), 12.

[16] Kumar, D. R., & Rao, S. K. M. (2019, August). A Sentiment Analysis of Twitter Data Using Bi-Directional Long Short Term Memory. In *International Conference on Application of Robotics in Industry using Advanced Mechanisms* (pp. 161-175). Springer, Cham.

ISSN: 2005-4238 IJAST

[17] Pethalakshmi D. A. (2020). Twitter Sentiment Analysis Using Dempster Shafer Algorithm Based Feature Selection and One Against All Multiclass SVM Classifier. International Journal of Advanced Research in Engineering and Technology (IJARET), 11 (2), 2020, pp 163-185.

[18] R. Xia, C. Zong, and S. Li, "Ensemble of feature sets and classification algorithms for sentiment classification," Information Sciences: an International Journal, vol. 181, no. 6, pp. 1138–1152, 2011 .

ISSN: 2005-4238 IJAST