# DECISION TREE BASED FEATURE SELECTION AND MULTILAYER PERCEPTRON FOR SENTIMENT ANALYSIS

Jeevanandam Jotheeswaran[1] and S. Koteeswaran[2]
[1]Vel Tech Dr. RR and Dr. SR Technical University, Chennai, Tamil Nadu, India
[2]Department of Computer Science Engineering, Vel Tech Dr. RR & Dr. SR Technical University, Chennai, Tamil Nadu, India

## ABSTRACT

Sentiment analysis plays a big role in brand and product positioning, consumer attitude detection, market research and customer relationship management. Essential part of information-gathering for market research is to find the opinion of people about the product. With availability and popularity of like online review sites and personal blogs, more chances and challenges arise as people now can, and do use information technologies to understand others opinions. In this paper, a Multi-Layer Perceptron (MLP) is used to classify the features extracted from the movie reviews. A Decision Tree-based Feature Ranking is proposed for feature selection. The ranking is based on Manhattan Hierarchical Cluster Criterion In the proposed feature selection; a decision tree induction selects relevant features. Decision tree induction constructs a tree structure with internal nodes denoting an attribute test with the branch representing test outcome and external node denotes class prediction. In this paper, a hybrid algorithm based on Differential Evolution (DE) and Genetic Algorithm (GA) for weight optimization algorithm to optimize MLPNN is proposed. IMDb dataset is used to evaluate the proposed method. Experimental results showed that the MLP with proposed feature selection improves the performance of MLP significantly by 3.96% to 6.56%. Classification accuracy of 81.25% was achieved when 70 or 90 features were selected.

**Keywords:** multi-layer perceptron (MLP), opinion mining (OM), IMDb, inverse document frequency (IDF), principal component analysis (PCA), differential evolution (DE), genetic algorithm (GA).

## 1. INTRODUCTION

Opinion Mining (OM) identifies author’s viewpoint on a subject instead of identifying subject itself. Due to the automatically extracted usable knowledge’s from customer feedback data on Web, OM has become a widely researched subject in data mining. OM’s ultimate goal is to extract customer opinions on products and to present it in an effective way to serve certain objectives. The steps and techniques will differ based on presentation of the summarized information. In case of negative and positive reviews on a given product are provided, classifying each review based on its polarity (positive/negative) is required. But, if we were to show customer feedback on a product’s features, it is necessary to extract product features and analyze each feature’s [2] sentiment.

Also known as sentiment analysis or sentiment classification, OM focuses on an author’s attitude to a topic rather than the topic. OM is applied to movie reviews, commercial products and services reviews to Weblogs, and News. Following advances spearheaded by Pang and Lee, OM sub-tasks have evolved over years. Subtasks include:

a) Subjectivity Analysis –is a binary classification task determining whether a given text is objective (neutral in sentiment) or subjective (expressing a positive/negative sentiment).

b) Polarity Analysis - prediction of whether an established text which is subjective is positive/negative in polarity.

c) Polarity Degree - measures a subjective text’s [3] polarity degree as positive or negative.

Sentiment analysis [4] is natural language processing to track public mood about a specific product or topic. Sentiment analysis builds a system to collect reviews or opinions about product in the web expressed in blog posts, comments or tweets and examine its polarity. Sentiment analysis finds application in many domains. In marketing, it judges an ad campaign or new product launch’s success, determines which product or service versions are popular and identifies which demographics like/dislike particular features. Literature surveys indicate two popular techniques including machine learning and semantic orientation with regard to sentiment analysis.

Machine learning applicable to sentiment analysis belongs to supervised classification in general. Two sets of documents, training and test set are required in machine learning based classification: Training set is used by classifiers to learn documents differentiating characteristics; it is thus called supervised learning. And test sets validate the classifier’s performance. Semantic orientation approach to sentiment analysis is unsupervised learning as it needs no prior training to mine data. It measures how far a word is either positive or negative.

Sentiment classification [5] is considered as a two-class, positive and negative, classification problem. Training/testing data consists of reviews. As online reviews include rating scores by reviewers, e.g., 1-5 stars, ratings determine positive/negative classes; a review with 4 or 5 stars is considered positive; and that with 1 to 2 stars negative. Research papers do not use neutral class, which makes classification issues easier, but it is possible to assign 3-star reviews as neutral class. Sentiment classification is a text classification problem. Conventional text classification classifies different topics documents e.g., sciences, politics, sports and so on where topic related words act as key features. The sentiment classification does not concern about the topic but of the sentiment or opinion words indicating positive or negative opinions. Thus words like great, excellent, amazing, horrible, bad, worst, etc are the key for classifying the polarity. Classification performed is based on fixed syntactic patterns likely to express opinions.

Inverse Document Frequency (IDF) is an important concept in information retrieval and it aims to improve automatic indexing and retrieval systems. IDF is a standard way to measure global importance or discriminative power of textual terms [6]. IDF when combined with Term Frequency (TF), results in a highly effective term weighting scheme applied across a range of application areas, including knowledge management, databases, natural language processing, text classification and information retrieval. The IDF based on global term frequency of empirical observations, where highly frequent terms are given less weight than less frequent terms as they are common and less discriminative.

In information retrieval, there were many attempts to refine TF component. IDF term weighting is computed; but, there were few attempts to improve limited number of "classical" IDF formulations. This may be due to being non-trivial to change standard IDF formulation in a meaningful way when improving effectiveness. Though there are heuristic ways to alter IDF formulation, doing so results in limited understanding on why things improved.

Term specificity measure became IDF based on counting documents number in a collection being searched which has the term in question. The idea was that a query term in many documents was not a good discriminator, and thus be given less weight than one which occurs in few documents. Intuition and measure associated with it changed the look of information retrieval. Together with TF (frequency of term in document, here, the more the better), it is used in almost every term weighting scheme [7]. The weighting schemes class generically known as TF*IDF, involves multiplying IDF measure (one of a number of variants) by TF measure proved robust and hard to beat, even by more carefully worked out models and theories. It spread outside text retrieval into other media's retrieval methods and into other purposes language processing techniques.

This work uses Multi-Layer Perceptron (MLP) to classify features extracted for OM using decision tree based feature selection. The rest of the paper is organized as follows: section 2: literature survey, section 3: Methodology, Section 4: results and discussion and section 5: conclusion.

## 2. LITERATURE REVIEW

An opinion-tree based flexible Sentiment Analysis model proposed by Ding, et al., [8]. A new tree type opinion tree was proposed and defined. Opinion tree based flexible Sentiment Analysis model was created and was coarse grained, medium-sized and fine-grained. OM was realized in one unified, flexible model. The flexible OM procedure was set for internet public opinions. Finally, an experiment on building an opinion tree was finished and overall opinions about hot topics on the internet was formed in this opinion tree.

Sentiment Analysis based on feature-level was proposed by Liu, et al., [9], where explicit and implicit features were used. Opinion words were divided into two categories, vague and clear opinion words, to identify implicit features and feature clusters. Feature clustering was based on 3 aspects: corresponding opinion words, feature similarity and features structures. Context information was also used to enhance clustering in the method, which was useful. The experiment demonstrated the good performance of the proposed method.

Comparison of model-based learning methods for feature-level Sentiment Analysis was proposed by Qi and Chen [10] where the authors adopted Conditional Random Field (CRF) model to perform OM tasks. It not only highlighted algorithm's ability in mining intensifiers, phrases and infrequent entities, but integrated additional elements into the model to optimize its training, decoding processes. It was compared to Lexicalized Hidden Markov Models (L-HMMs) based OM in experiments, which proved it to have better accuracy from various aspects.

A feature dependent method for OM and classification was proposed by Balahur, *et al*., [11], which presented a feature driven opinion summarization method, where the term ldquo driven rdquo described the concept to detail the look. The proposed method improved over baseline and a discussion on the method's strong and weak points was presented.

An approach based on Tree Kernels for Online Product Reviews OM was proposed by Jiang, *et al*., [12], which defined many tree kernels for sentiment expression extraction and sentiment classification, OM's subtasks. Tree kernels encoded syntactic structure information and sentiment related information like sentiment boundary and sentiment polarity, which were important OM features. Experiments on a benchmark data set indicated that tree kernels significantly improved sentiment expression extraction and sentiment classification performance. Besides, the proposed tree kernels' linear combination; traditional feature vector kernel achieved best performances using benchmark data set.

A feature based OM online free format for customer reviews using frequency distribution and Bayesian statistics was proposed by Anwer, *et al*., [13], which read reviews word by word to finally summarize results in terms of frequency and opinions probability. Bayesian probability was useful for accurate results and true predictions. As frequency results were in graphics, its use by new customers could lead to a decision to buy the displayed product. Frequency based results were understood by consumer and Bayesian probability results verified frequency results.

A product feature grouping for OM proposed by Zhai, *et al*., [14], reviewed aggregators and e-commerce sites which were examples of businesses that were OM dependent to produce feature based products quality summaries. This model identifies product features and collects their positive and negative opinions to produce good and bad point's summary.

New Avenues in OM and Sentiment Analysis, which have valuable, vast, and unstructured information about public opinion was proposed by Cambria, *et al.,* [15] where history, current use, and future of OM and sentiment analysis were discussed, with techniques and tools.

A state of the art OM and its application domains was proposed by Binali, *et al*., [16], which critically evaluated existing work, presented an OM framework and exposed new research areas. Individuals, businesses and government could know a product's general opinion as also that of a company or public policy. At its core was the subjective terms, semantic orientation in documents or reviews which seek to establish contextual connotation through OM. This leads framework motivation for OM. It categorized present literature to ensure clear, research opportunities.

Mining product features and opinions were proposed by Pan and Wang [17], based on pattern matching for mining features and opinions according to Chinese reviews characteristics. Reviews were split into simple structure fragments, and different patterns were adopted to match fragments with different structures to mine review features and opinions. Then, a feature grouping based method was used to prune infrequent features and mining results comprehensiveness. Experiments proved the method to be effective.

Conditional Random Fields model based product features was proposed by Xu, *et al.,* [18], to present a Chinese product features identification approach, integrating chunk features and heuristic position information to word features, part of speech features and context features. Experiments revealed the proposed techniques improved product OM performance.

Extracting opinion features in Sentiment Patterns was proposed by Zhai, *et al*., [19]. This work proposed a new OFESP approach which considered reviews structure characteristics for higher precision and recall values. With a sentiment patterns self-constructed database, OFESP matched each review sentence to obtain features, followed by filter redundant features regarding domain, statistics and semantic similarity relevance. Experiments on real world data showed that compared to a window mechanism based traditional method, OFESP outperformed it on F-score, precision and recall. Compared to syntactic analysis based approach, OFESP performed better on recall and F-score

Tasks are extracted by OM from documents opinions as expressed by sources. A comparative study was undertaken on methods/resources used for OM from newspaper article quotations. Balahur, et al., [20] presented problems in being motivated by possible targets and the variety offered by quotes. It evaluated annotated quotations from news from an EMM news engine. Generic OM needs large lexicons, and specialized training/testing data.

Researchers developed large feature selection algorithms for other purposes in the past with each model having its own advantages/disadvantages. Principal Component Analysis (PCA) is a statistical tool to reduce data set dimensionality. It is popular due to its simplicity as regards computational and understanding what's happening [27]. PCA's goal is revealing data set's hidden structure. By doing so, it may be able to

- identify how different variables work together to create system dynamics
- reduce data dimensionality
- decrease data redundancy
- filter data noise
- compress data
- prepare data for analysis through other techniques

Though efforts attempted to survey existing feature selection algorithms, what is needed is a repository which collects representative feature selection algorithms to ensure comparison/joint study. To offset this, a feature selection repository to collect popular algorithms developed in feature selection research was presented by Zhao, *et al*., [21]. This was to be a platform to ensure an application/comparison/joint study. The repository assists researchers achieve reliable evaluation to develop new feature selection algorithms.

Kim and Hovy [22] proposed a novel technique which generalized the n-gram feature patterns. Crystal an election prediction system was presented for which classified users' opinions posted. The proposed method was implemented on an election prediction website. The past election prediction messages was collected from the Web and the lexical patterns frequently used by people to express their predictive opinions was concentrated upon. The proposed n-gram feature pattern was applied and SVM was used to predict election results. Experimental results show that Crystal out performances non-generalized n-gram approach and predicted future elections with 81.68% accuracy.

Abbasi *et al* [23] integrated particular feature extraction components to the linguistic characteristics of Arabic. An Entropy Weighted Genetic Algorithm (EWGA) was developed for feature selection. The experimental results indicate high performance levels using EWGA with SVM. EWGA has proven to improve performance and get a better assessment of the key features.

Shein *et al* [24] proposed an ontology based combination approach for sentiment classification. The proposed method combined natural language processing techniques, ontology based on Formal Concept Analysis (FCA) design, and SVM for classifying the software reviews are positive, negative or neutral.

## 3. METHODOLOGY

This work focuses on feature selection for Sentiment Analysis using decision tree based feature selection and classifying the feature using MLP. The movie dataset is used to evaluate the proposed method. The various techniques involved are:

### 3.1 Inverse Document Frequency (IDF)

Inverse Document Frequency (IDF) is used in Information Retrieval [25]. IDF is defined as$-log2dfw/D$, where D is number of documents in a collection and $dfw$ the document frequency, the number of documents containing w. naturally, there is a strong relationship between document frequency$dfw$, and word frequency, $fw$

If there are N documents in a collection, and term $t_i$ occurs in $n_i$ of them (It is assumed that terms are words, or word stems). Then the measure as a weight is applied to term ti, essentially as

$$idf\ (ti)\ =\ log\ \frac{N}{n_i}$$

### 3.2 Principal Component Analysis (PCA)

PCA is mathematically defined as an orthogonal linear transformation transforming data to a new coordinate system so that any data projection's greatest variance comes to lie on first coordinate, second greatest variance on second coordinate, and so on [28].

If $\boldsymbol{X^T}$ with zero empirical mean, where each $n$ rows represents differing repetition of an experiment, each $m$ columns gives a specific datum. Singular value decomposition of X is $X = W\Sigma V^T$, where $m \times m$ matrix $\mathbf{W}$ is matrix of eigenvectors of covariance matrix $\mathbf{XX}^T$, the matrix $\boldsymbol{\Sigma}$ is $m \times n$ rectangular diagonal matrix with nonnegative real numbers on diagonal, and $n \times n$ matrix $\mathbf{V}$ is matrix of eigenvectors of $\mathbf{X}^T\mathbf{X}$.

### 3.3 Proposed decision tree-based feature ranking

The decision trees are used as embedded method of feature selection. In the proposed decision tree-based feature ranking, a Decision tree induction selects relevant features and ranks the features. Decision tree induction is decision tree classifiers learning, constructing a tree structure with internal nodes (non-leaf node) denoting an attribute test. Each branch represents test outcome and external node (leaf node) denotes class prediction [26]. The algorithm at each node chooses best attribute to partition data into individual classes. Information gain measure is used to choose the best partitioning attribute by attribute selection. Attribute with highest information gain splits the attribute. The attribute's information gain is found by

$$info(D) = -\sum_{i=1}^{m} P_i log_2 P$$

where $p_i$ is probability that an arbitrary vector in D belongs to class $c_i$. A log function to base 2 is resorted to as information is encoded in bits. Info (D) is average information needed to identify vector D class label. Before constructing trees, base cases are considered with the following points:

- A leaf node is created if all samples belong to same class.
- When no features provide information gain, it creates a decision node higher up the tree using the expected class value.

The decision tree induction algorithm in general checks for base cases and/or each attribute (a), locates information gain of each attribute for splitting. Let a-best be attributing with highest information gain. Create decision node that splits a-best. Return with sub lists obtained by splitting a-best, adding nodes as children for tree.

The proposed method defines a threshold measure to choose relevant features. The threshold measure is based on the information gain value and the proposed Manhattan distance for selecting of the features. The proposed decision tree method searches heuristically for relevant features. The features are ranked by computing the distance between the hierarchical clusters. The proposed Manhattan distance for n number of clusters is given as:

$$MDist = \sum_{i=1}^{n} \left( a_i - b_i \right)$$

A cubic polynomial equation is derived using the Manhattan values and the threshold criterion is determined from the slope of the polynomial equation. The features are assumed to be irrelevant for classifying if the slope is zero or negative and relevant when the slope is positive.

### 3.4 Multilayer Perceptron (MLP) network

Neural Networks (NN) are parallel computing systems having a large number of simple processors with interconnections. NN models use organizational principles in a weighted and directed graphs network where nodes are artificial neurons and directed edges connections between neuron outputs and neuron inputs [29]. NN include many interconnected processing elements that operate simultaneously. Pattern recognition data processing is bulky and recognition in conventional NN is slow as propagation takes place in multiplication and addition calculation required for data processing.

A Multilayer Perceptron (MLP) is a feed forward Artificial Neural Network (ANN) model that maps input data sets onto appropriate output [29-31] sets. An MLP has many node layers in a directed graph, with each layer being connected to the next. Each node is a neuron (processing element) with nonlinear activation function in layers other than the input layer. Supervised learning technique is used to train a network. The widely used learning technique is called back propagation. MLP is modified standard linear perceptron that differentiates data not linearly separable. Figure-1 shows a MLP structure with input layer, one hidden layer and output layer.

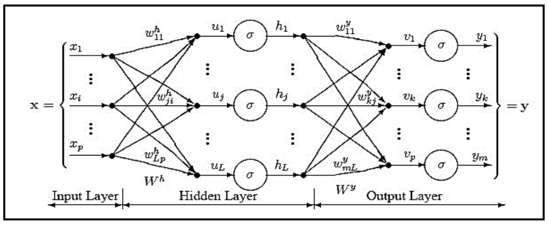

**Figure-1.** A structure for MLP layer.

In the input layer, (x1.., xp) are inputs; wh11 through whlp are weights of x1 through xp; h1 and hL are hidden layers; y1 through ym are outputs.

$u_i = \sum ui * whi$

$v_i = \sum hi * wyi$

MLP has linear activation function in neurons, which is a simple on-off mechanism to determine whether a neuron fires. What makes a MLP different is that each neuron uses nonlinear activation developed to model action potentials frequency, or firing. This function can be modeled in many ways, but must be normal and differentiable. The two activation functions in current applications are sigmoid and described by

$$\phi(y_i) = \tanh(v_i) \text{ and } \phi(y_i) = (1 + e^{-v_i})^{-1}$$

in which the former function is a hyperbolic tangent ranging from -1 to 1, and the sigmoid is a logistic function, similar in shape but which ranges from 0 to 1. Other specialized activation functions include radial basis functions used in another class of supervised NN models.

Back-propagation is generally used to train the network where training is performed for single pattern at a time. A training set consists of a collection of input-output examples. Each training instance is fed to train the network for different classes. Back-propagation training is said to be gradient descent algorithm which improves the performance of the neural net by reducing its error along its gradient. The error is expressed by the Root-Mean-Square (RMS) error, which can be calculated by:

$$E = \frac{1}{2} \sum_{p} \| t_p - o_p \|^2$$

The error (E) is computed based on the sum of the geometric averages of the difference between projected target (t) and the actual output (o) vector over all patterns (p). In each training step, the weights (w) are adjusted to decrease error, scaled by learning rate lambda.

$$\nabla E = \left( \frac{\delta E}{\delta w_1}, \frac{\delta E}{\delta w_2}, \ldots\ldots, \frac{\delta E}{\delta w_n} \right)$$

$$w_{new} = w_{old} - \lambda \nabla E$$

The sigmoid function has the property

$$\frac{d}{dx} \sigma(x) = \sigma(x)(1 - \sigma(x))$$

Simple multiplication and subtraction operators are required to compute the derivative of the sigmoid function simplifying the computational effort of the back-propagation algorithm. The equations for weight changes are reduced to:

$$\nabla w_{from,to} = -\lambda o_{from} \delta_{to}$$

$$\Delta w_{from,to} = -\lambda o_{from} \delta_{to}$$

$$\delta_{output} = -(t_{output} - o_{output})$$

$$\delta_{hidden} = \sigma'(s_{hidden}) \sum_{i} \delta_i w_{hidden,i}$$

Different functions are available for connecting hidden and output nodes. Back-propagation has difficulty with local optima. It also requires many repeated presentations of the input patterns, so that the weights can be adjusted before the network settles down into an optimal solution.

### 3.5 Proposed hybrid genetic algorithm and differential evolution algorithm for MLPNN training

The main drawbacks of backpropagation are performance degradation as the dimensionality and complexity of the data increases and getting trapped in the local minima. Genetic Algorithm (GA) is a popular alternative learning technique replacing gradient descent methods like error Backpropagation in MLPNN. GA avoids local minima entrapment in instances where a backpropagation algorithm converges prematurely. With GA locating a region of optimal performance of learning and gradient descent, backpropagation is applied to this region. Evolutionary Algorithms work on candidate solutions populations; they represent a basic framework for multi objective optimization. Similar to GA, Differential Evolution (DE) is a population based algorithm, a stochastic optimization procedure to reduce an objective function modelling problem's objectives while including constraints.

### 3.5.1Genetic Algorithm (GA)

A GA is an iterative procedure that structures a chromosomal population which represents the candidate solutions for the specific domain. The population are rated for effectiveness as solutions based on fitness. New candidate solutions (populations) are formed with genetic operators like reproduction, crossover, and mutation [32]. The basic steps in GA are shown in Figure-2.

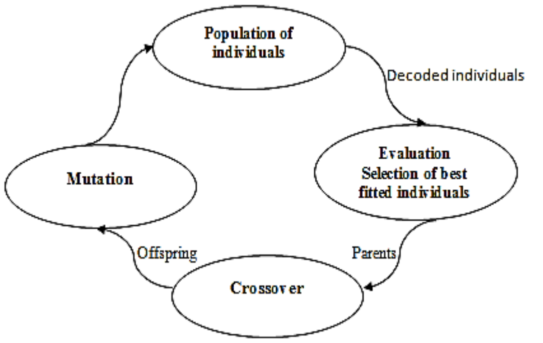
Population of individuals
Decoded individuals
Evaluation Selection of best fitted individuals
Parents
Crossover
Offspring
Mutation

**Figure-2.** Basic GA scheme.

**Population of individuals:** Generally, the initial population are initiated randomly. Chromosomes represent a gene set which are code independent variables to represent a problem solution. An offspring population is created through operators like selection, recombination and mutation. In this work, the weights of the MLPNN are encoded as a list of real numbers.

**Selection:** In selection process is based on 'survival of the fittest' principle. The fitness of chromosomes evaluates the quality of the solutions. Chromosomes with higher fitness survive and are used as parents to create next generation of population.

**Crossover:** Crossover is a reproductive step where parent genes form a new chromosome. The GA recombines two parent gene into 2 children using single point or two point crossover.

**Mutation:** It introduces random gene to introduce variety in the gene pool. It is regulated by the mutation probability.

The MLPNN weights are encoded into chromosomes as a list of real numbers. Run the network using training instances which returns sum of the squares of errors. The fitness objective is to reduce the error. Initial population weights (real numbers) are chosen randomly with probability distribution and vary from back propagation where the weights are in uniform distribution between -1.0 and 1.0.

The steps of the proposed method for training weights for the MLPNN are:

```
Create initial population chromosomes randomly
for (all training data)
for (all weights)
for (i = 1 : 50)
- Evaluate the fitness of the population.
- Selection: The best chromosomes to reproduce,
- Crossover and Mutation.
- A new generation is created from the fittest of the previous generation.
end for
Evaluate the fitness.
The fittest chromosome of the population is assigned as the new weight.
end for
end for
```

### 3.5.2 Differential evolution

Differential Evolution (DE) like GA has a population of candidate solutions, which recombine and mutate to produce new individuals which are chosen based on function performance. DE is a parallel direct search method using NP D-dimensional parameter vectors [33]. Initial vector population is chosen randomly to cover all parameter space. A uniform distribution of probability for all random decisions is assumed unless otherwise stated. New parameter vectors are generated by mutation process where the weighted difference between two population vector is added to a third vector. The mutated vectors parameters are combined with target vector, to yield the trial vector. In selection step, the trial vector replaces the target vector if it has a lower cost function value. Each population vector acts as the target vector once so that NP competitions take place in one generation. Basic strategy of DEs is:

- Initialization: Population is generated randomly with a distribution uniform [34].

**Mutation:** Randomly select three vectors, differences of two vectors is added to the third. For each target vector $x_{i,G,}i$ =xi, G; i = 1; 2; 3; . . . ; NP, a mutant vector is generated according to random indexes $r_1 r_2 r_3 \in \{1.2....\}$ integer, mutually different and F>F > 0.

$$v_{i,G+1} = x_{r1} + F.\{x_{r2,G} - v_{i,G+1} = x$$

Recombination: If the child has generated a higher value of the objective function than the primary parent, then it replaces the trial vector [35]:

$$u_{i,G+1} = \{u1_{i,G}, u2_{i,G+1}, .....u.$$

$$u_{i,G+1} = \begin{Bmatrix} u_{i,G+1} if\{randb(j) \le CR\} orj = rn \\ u_{i,G} if\{randb(j) > CR\} orj = rnb \end{Bmatrix} \quad j = 1, 2......D$$

- Selection: All vectors are selected once as primary parent to check whether the selected parent is better that their child [36]. The next generation trial vector $u_{i,G+1}$ is compared to target vector $x_{i,G}$ using greedy criterion. If vector $u_{i,G+1}$ ui;G+1 yields a smaller cost function value than $x_{i,G}$ xi then xi; $x_{i,G+1}$ is set to ui; $u_{i,G+1}$; otherwise, old value $x_{i,G+1}$ xi, G is retained [37].

### 3.5.3 Proposed hybrid genetic algorithm and differential evolution algorithm

The evolutionary algorithms find globally satisfactory, which may not be optimal, solutions to the optimization problem. When applied to large real problems they may become too slow. To overcome this difficulty parallelization methods have been proposed. In the proposed hybrid GA-DE, the GA and DE algorithms are run in parallel. Two main reasons for parallelizing these algorithms are to achieve time savings by distributing the computational effort and to benefit from the algorithmic point of view.

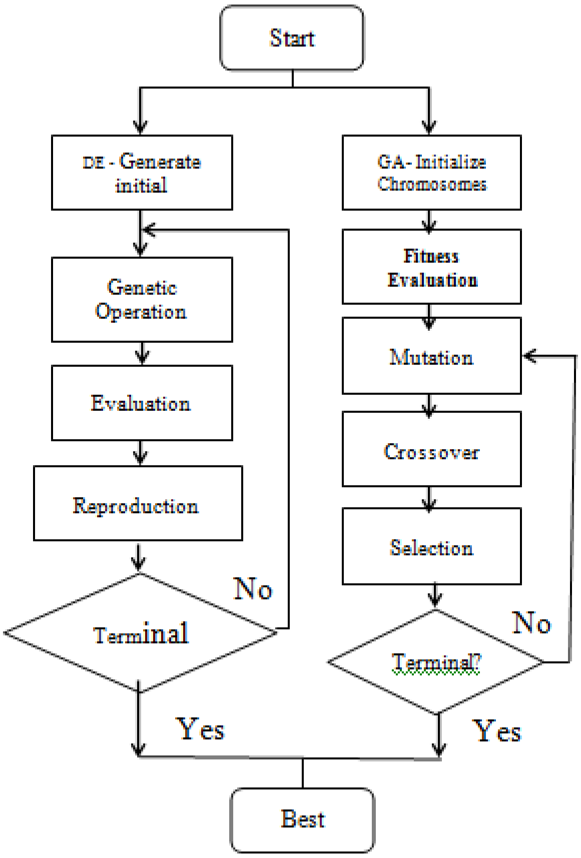

**Figure-3.** Proposed hybrid genetic algorithm and differential evolution algorithm.

Parallelization also assures extension of search space that yields to improvement or degradation of final solution quality. So, final solution quality should be considered as a parameter of parallelization strategy performance. Consequently, combination of gains is expected: parallel execution enables efficient search of different regions in solution space yielding to improved final solution quality in smaller execution time. The flowchart of the proposed hybrid is depicted in Figure-3.

## 4. RESULTS AND DISCUSSIONS

This paper focuses on feature selection and classification for Sentiment Analysis using decision tree based feature selection. For classifying the movie reviews, features are extracted and 30, 50, 70 and 90 features are selected. The selected features are classified using MLP. MLP with one hidden layer and tanh activation function are used as the classification algorithm. The results obtained are tabulated in Table-1 and are shown in Figures 4-7.

**Table-1.** Results achieved for MLP.

| **No of Features** | **30** | **50** | **70** | **90** |
|---|---|---|---|---|
| **Classification accuracy** | | | | |
| MLP NN with PCA | 70.75 | 74.75 | 76.25 | 76.25 |
| MLP with proposed feature extraction | 75.75 | 78.75 | 81.25 | 81.25 |
| MLP with proposed feature extraction and Proposed weight training | 78.25 | 79.5 | 83 | 83.25 |
| **Average precision** | | | | |
| MLP NN with PCA | 0.7063 | 0.7475 | 0.7625 | 0.7625 |
| MLP with proposed feature extraction | 0.7575 | 0.7878 | 0.8125 | 0.8125 |
| MLP with proposed feature extraction and Proposed weight training | 0.78265 | 0.7951 | 0.8301 | 0.8326 |
| **Average recall** | | | | |
| MLP NN with PCA | 0.6982 | 0.7429 | 0.7682 | 0.7682 |
| MLP with proposed feature extraction | 0.7594 | 0.7739 | 0.81455 | 0.81455 |
| MLP with proposed feature extraction and Proposed weight training | 0.7728 | 0.7871 | 0.82175 | 0.8263 |
| **F measure** | | | | |
| MLP NN with PCA | 0.7138 | 0.7447 | 0.7654 | 0.7654 |
| MLP with proposed feature extraction | 0.7584 | 0.7808 | 0.8135 | 0.8135 |
| MLP with proposed feature extraction and Proposed weight training | 0.7777 | 0.7911 | 0.8259 | 0.8294 |

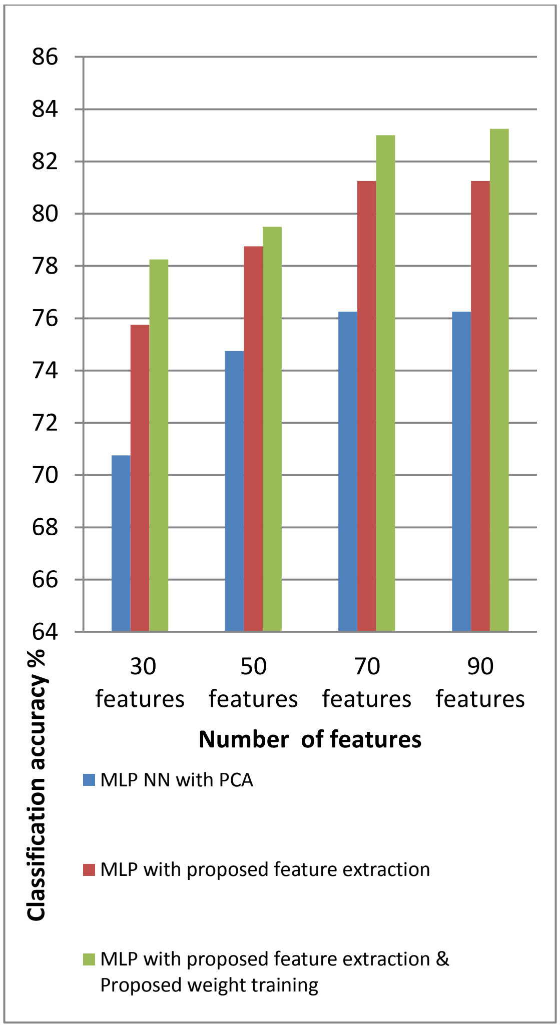

**Figure-4.** Classification accuracy.

The Figure-4 shows the classification accuracy obtained by different methods. It is observed that the MLP with proposed feature selection achieves observed that the proposed feature selection improves the classification the best results of 81.25% when 70 or 90 features were selected. It is also increases accuracy of MLP significantly by 3.96% to 6.56%. The proposed MLP with proposed feature extraction and proposed weight training achieves the best classification accuracy of 83.25% for 90 features.

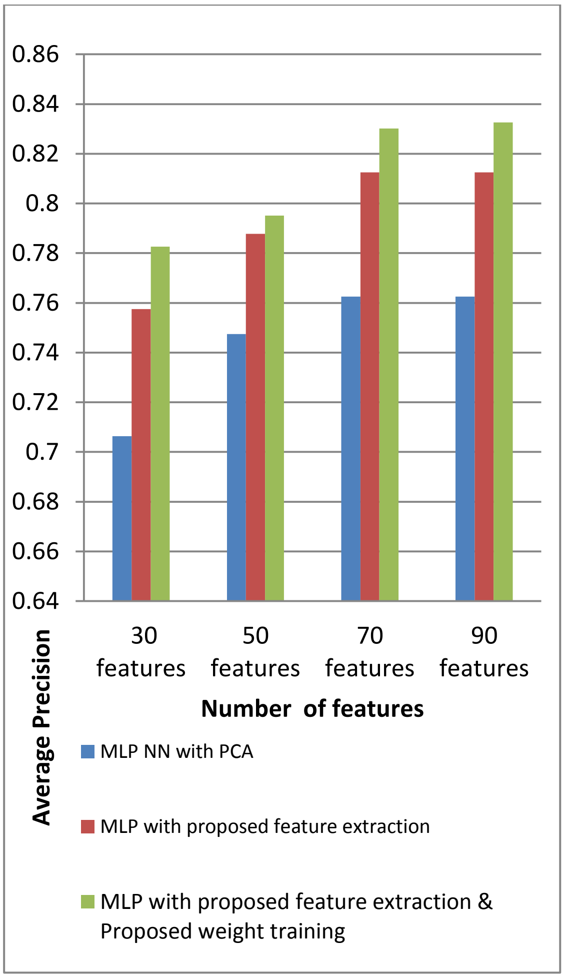

**Figure-5.** Average precision.

Figure-5 shows that the proposed selection method with MLP obtains an average precision of 0.8125 when 70 or 90 features are used. The proposed MLP with proposed feature extraction and proposed weight training achieves the best precision of 0.8326 for 90 features.

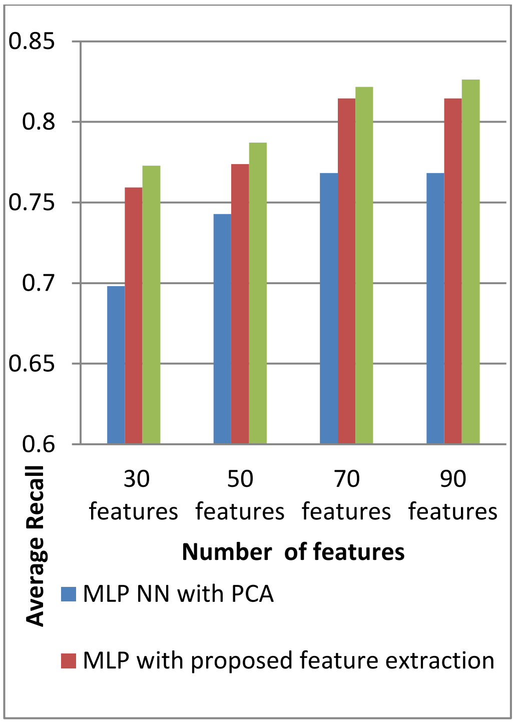

**Figure-6.** Average Recall.

Figure-6 shows the proposed MLP with proposed feature extraction and proposed weight training achieves the best recall of 0.8263 for 90 features.

Figure-7 shows that the proposed selection method with MLP obtains F-measure of 0.8135 when 70 or 90 features are used. The proposed feature selection method improves the f measure by 4.85% to 6.28% when compared to PCA. The proposed MLP with proposed feature extraction and proposed weight training achieves the best recall of 0.8294 for 90 features which is higher by 8.03% compared to MLPNN with PCA and by 1.94% compared to MLPNN with proposed feature extraction.

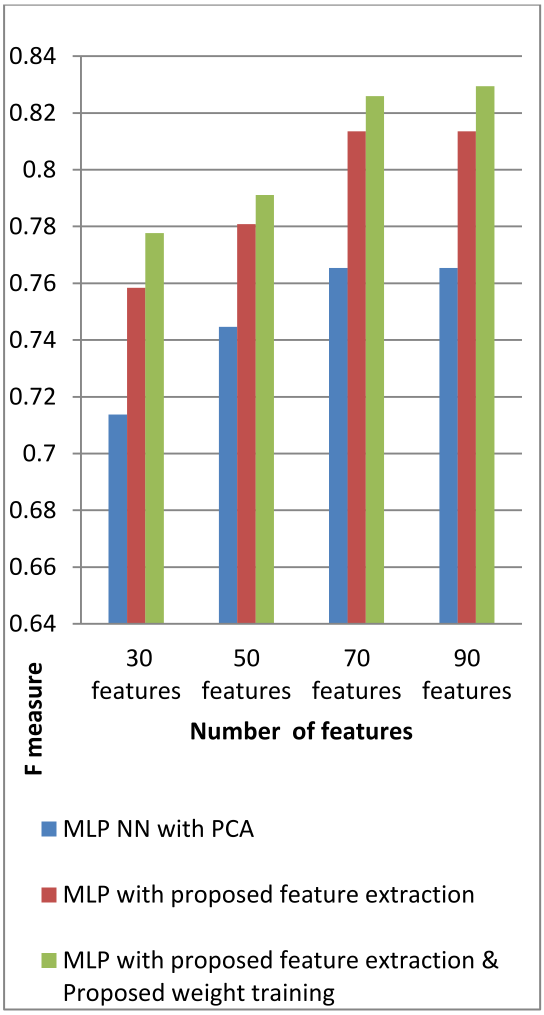

**Figure-7.** F-measure.

## 5. CONCLUSIONS

Feature selection is required for successful data mining, as it lowers data dimensionality and removes irrelevant features. This work proposes a feature selection for OM using decision trees and classification by MLP. The feature selection is an extension of Decision Tree-based Feature Ranking using Manhattan Hierarchical Cluster Criterion focusing on OM feature selection using decision tree. Movie review features from IMDb was extracted by using IDF. PCA was used for feature selection based on work importance regarding the entire document. Experiments were conducted using different number of features. The proposed MLP with proposed feature extraction and proposed weight training method with 90 features obtains 83.25% classification accuracy.

## REFERENCES

1. Balahur A., Steinberger R., Goot E. V. D., Pouliquen B. and Kabadjov M. 2009, September. Opinion mining on newspaper quotations. In Web Intelligence and Intelligent Agent Technologies, 2009. WI-IAT'09. IEEE/WIC/ACM International Joint Conferences on (3: 523-526). IET.

2. Lee D., Jeong O. R. and Lee S. G. 2008, January. Opinion mining of customer feedback data on the web. In: Proceedings of the 2nd international conference on Ubiquitous information management and communication (pp. 230-235). ACM.

3. Conrad J. G. and Schilder F. 2007, June. Opinion mining in legal blogs. In: Proceedings of the 11th international conference on Artificial intelligence and law (pp. 231-236). ACM.

4. Vinodhini G. and Chandrasekaran R. M. 2012. Sentiment Analysis and Opinion Mining: A Survey. International Journal. 2(6).

5. Liu B. 2012. Sentiment analysis and opinion mining. Synthesis Lectures on Human Language Technologies. 5(1): 1-167.

6. Metzler D. 2008, October. Generalized inverse document frequency. In: Proceedings of the 17th ACM conference on Information and knowledge management (pp. 399-408). ACM.

7. Robertson S. 2004. Understanding inverse document frequency: on theoretical arguments for IDF. Journal of documentation. 60(5): 503-520.

8. Ding J., Le Z., Zhou P., Wang G. and Shu W. 2009, November. An Opinion-Tree Based Flexible Opinion Mining Model. In Web Information Systems and Mining, 2009. WISM 2009. International Conference on (pp. 149-152). IEEE.

9. Liu L., Lv Z. and Wang H. 2012, October. Opinion mining based on feature-level. In Image and Signal Processing (CISP), 2012 5th International Congress on (pp. 1596-1600). IEEE.

10. Qi L. and Chen L. 2011, August. Comparison of model-based learning methods for feature-level opinion mining. In Proceedings of the 2011 IEEE/WIC/ACM International Conferences on Web Intelligence and Intelligent Agent Technology-Volume 01 (pp. 265-273). IEEE Computer Society.

11. Balahur A., and Montoyo A. 2008, October. A feature dependent method for opinion mining and classification. In Natural Language Processing and Knowledge Engineering, 2008. NLP-KE'08. International Conference on (pp. 1-7). IEEE.

12. Jiang P., Zhang C., Fu H., Niu Z. and Yang Q. 2010, December. An approach based on tree kernels for opinion mining of online product reviews. InData Mining (ICDM), 2010 IEEE 10th International Conference on (pp. 256-265). IEEE.

13. Anwer N., Rashid A. and Hassan S. 2010, August. Feature based opinion mining of online free format customer reviews using frequency distribution and Bayesian statistics. In Networked Computing and Advanced Information Management (NCM), 2010 Sixth International Conference on (pp. 57-62). IEEE.

14. Zhai Z., Liu B., Wang J., Xu H. and Jia P. 2012. Product Feature Grouping for Opinion Mining. IEEE Intelligent Systems. 27(4): 0037-44.

15. Cambria E., Schuller B., Xia Y. and Havasi C. 2013. New avenues in opinion mining and sentiment analysis.

16. Binali H., Potdar V. and Wu C. 2009, February. A state of the art opinion mining and its application domains. In Industrial Technology, 2009. ICIT 2009. IEEE International Conference on (pp. 1-6). IEEE.

17. Pan Y. and Wang Y. 2011, December. Mining product features and opinions based on pattern matching. In Computer Science and Network Technology (ICCSNT), 2011 International Conference on (3: 1901-1905). IEEE.

18. Xu B., Zhao T. J., Zheng D. Q. and Wang S. Y. 2010, July. Product features mining based on

conditional random fields model. In Machine Learning and Cybernetics (ICMLC), 2010 International Conference on (6: 3353-3357). IEEE.

19. Zhai Y., Chen Y., Hu X., Li P. and Wu X. 2010, October. Extracting Opinion Features in Sentiment Patterns. In Information Networking and Automation (ICINA), 2010 International Conference on (1: V1-115). IEEE.

20. Balahur A., Steinberger R., Goot E. V. D., Pouliquen B. and Kabadjov M. 2009, September. Opinion mining on newspaper quotations. In Web Intelligence and Intelligent Agent Technologies, 2009. WI-IAT'09. IEEE/WIC/ACM International Joint Conferences on (3: 523-526). IET.

21. Zhao Z., Morstatter F., Sharma S., Alelyani S., Anand A. and Liu H. 2010. Advancing feature selection research. ASU Feature Selection Repository.

22. Kim S. M. and Hovy E. H. 2007. Crystal: Analyzing Predictive Opinions on the Web. In EMNLP-CoNLL (pp. 1056-1064).

23. Abbasi A., Chen H. and Salem A. 2008. Sentiment analysis in multiple languages: Feature selection for opinion classification in Web forums. ACM Transactions on Information Systems (TOIS), 26(3): 12.

24. Shein K.P.P., Nyunt T.T.S. 2010. Sentiment Classification Based on Ontology and SVM Classifier. Communication Software and Networks, 2010. ICCSN '10. Second International Conference on. vol., no., pp. 169, 172, 26-28.

25. Church K. and Gale W. 1999. Inverse document frequency (idf): A measure of deviations from poisson. In Natural language processing using very large corpora (pp. 283-295). Springer Netherlands.

26. Gayatri N. Nickolas S. and Reddy A. V. 2010. Feature selection using decision tree induction in class level metrics dataset for software defect predictions. In Proceedings of the World Congress on Engineering and Computer Science. 1: 124-129.

27. Friston K. J., Frith C. D., Liddle P. F. and Frackowiak R. S. J. 1993. Functional connectivity: the principal-component analysis of large (PET) data sets. Journal of cerebral blood flow and metabolism, 13, 5-5.

28. MEI M. 2009. Principal component analysis.

29. Kröse B., Krose B., van der Smagt P. and Smagt P. 1996. An introduction to neural networks.

30. Ruck D. W., Rogers S. K. and Kabrisky M. 1990. Feature selection using a multilayer perceptron. Journal of Neural Network Computing. 2(2): 40-48.

31. Principe J. C., Euliano N. R. and Lefebvre W. C. 1999. Neural and adaptive systems: fundamentals through simulations with CD-ROM. John Wiley and Sons, Inc.

32. Blanton Jr, J. L. and Wainwright R. L. 1993, June. Multiple vehicles routing with time and capacity constraints using genetic algorithms. In: Proceedings of the 5th International Conference on Genetic Algorithms (pp. 452-459). Morgan Kaufmann Publishers Inc.

33. Holland J. H. 1992. Genetic algorithms. Scientific American 267(1): 66-72.

34. L. V. Santana Quintero and C. A. Coello Coello. 2004. Un Algoritmo Basado en Evoluci´on Differential para Resolver Problemas Multiobjetivo. Master’s thesis, IPN.

35. R. d. C. Gomez Ramon. 2001. Estudio emp´ırico de variantes de Evoluci´on Diferencial en optimization con restrictions. Master’s thesis, Laboratorio Nacional de Inform´atica Avanzada.

36. Botía J. A. and Charitos D. 2013, July. Genetic Algorithms and Differential Evolution Algorithms Applied to Cyclic Instability Problem in Intelligent Environments with Nomadics Agents. In Workshop Proceedings of the 9th International Conference on Intelligent Environments. 17: 222. IOS Press.