An Ensemble Approach to Enhance the Efficacy of Sentiment Prediction

Monalisha Ghosh*, Anirban Chakraborty, and Indrajit Pal

Future Institute of Engineering and Management (FIEM), Makaut University, India
E-mail: mghoshnit2019@gmail.com (M.G.); chakraborty.anirban@rediffmail.com (A.C.); imindramc@gmail.com (I.P.)
*Corresponding author

Manuscript received March 3, 2023; revised April 20, 2023; accepted July 10, 2023; published May 23, 2024

***Abstract*—Sentiment Analysis (SA) has recently been considered as the most active research field in the Natural Language Processing (NLP) domain. Deep Learning (DL) is a subset of the large family of Machine Learning (ML) and becoming a growing trend due to its automatic learning capability with impressive consequences across different NLP tasks. Hence, a fusion-based machine learning framework has been attempted by merging the traditional machine learning method with deep learning techniques to tackle the challenge of sentiment prediction for a massive amount of unstructured review dataset. The proposed architecture aims to utilize the Convolutional Neural Network (CNN) with a backpropagation algorithm to extract embedded feature vectors from the top hidden layer. Thereafter, these vectors were augmented with an optimized feature set generated from the Binary Particle Swarm Optimisation (BPSO) method. Finally, a traditional Support Vector Machine (SVM) classifier is trained with this extended feature set to determine the optimal hyper-plane for separating two classes of review datasets. The evaluation of this research work has been carried out on two benchmark movie review datasets: IMDb, SST-2. Experimental results with comparative studies based on performance accuracy and F-Score value are reported to highlight the benefits of the developed frameworks.**

***Keywords*—deep learning framework, sentiment analysis, Binary Particle Swarm Optimisation (BPSO), Convolutional Neural Network (CNN), Support Vector Machine (SVM)**

I. Introduction

In this study, Sentiment Analysis (SA) represents a rich set of interrelated sub-problems rather than a single problem to solve, which makes this field more challenging to the researchers [1, 2]. The task of sentiment analysis poses two main challenges. The initial challenge is the recognition of relevant bits of information that may not be directly present in the concentrated text. Another is related to the analysis of relevant textual data to detect the exact sentiment, as the same terms may convey different sentiments in different contexts. The problem statement of this research can be represented as follows: suppose a movie review with multiple features has been given, the purpose is to classify this review as positive or negative.

The maximum number of research works related to sentiment classification has been approached through either Lexicon-based methods or Machine Learning (ML) methods [3, 4]. The former category doesn't require any prior training dataset; it performs the task of identifying a list of words, phrases that havea semantic value. It mainly concentrates on patterns of unseen data. The second category, or Machine learning method involves the training of models on specific documents by employing the supervised and unsupervised methods. The performance of machine learning algorithms is determined by estimating how accurately the features are recognized and pulled out [5, 6]. Feature engineering is the task of placing domain knowledge to transform raw data into a feature vector. Therefore, Deep Learning (DL) strategy has been utilized in this paper to address the issue of the future engineering process [7]. Deep learning method needs very less engineering by hand, so it can easily take advantage of the increasing amount of available data and achieve impressive classification accuracy in comparison with traditional machine learning approaches [8, 9].

Actually, in case of traditional machine learning models, several domain experts are needed for identification, selection, and extraction of relevant features to minimize the complexity as well as the dimension of the data.Though the traditional approach with the support of feature engineering achieved quite impressive results in sentiment classification, several domain experts are needed for the identification, selection, and extraction of relevant features to minimize the complexity as well as the dimension of the data. These handcrafted features are basically domain-specific and need to be altered from being applied to a different domain.

However, we have overcome such issues by utilizing deep learning techniques to automatically learn a complex feature representation from big data, rather than accepting manually crafted features.

The deep learning approach, especially Convolutional Neural Network (CNN), has been achieving a lot of success in the domain of image processing [10]. The topology provided by the image dataset is specifically constructed with the lattice of pixels and deep learning is suitable on that framework. On the other hand, very few researchers in the text review domain have portrayed CNN architecture as the most relevant feature extractor. The previous work [11–13] mainly exploited traditional CNN as a sentiment classifier for polarity detection. In this research, we have implemented two different approaches to sentiment analysis. Firstly, the employment of CNN as a feature extractor is encouraging for the visualization of features within a large dataset, since the CNN architecture is quite adept in learning the hidden semantics from a large corpus of review sets. CNN-based architecture, mainly utilizes convolutional filters to appear as a computationally captivating model, but the memory requirements and computational overhead due to the use of huge parameters may be a big issue. Therefore, in this paper, we have proposed the CNN-SVM model to connect an Support Vector Machine (SVM) classifier which helps CNN to reduce the computational burden.

Researches on sentiment analysis are growing to a great extent and attracting a wide range of attention from academics and industries as well. Understanding emotions, analyzing situations, and sentiments to them, is a human's natural ability. Sentiment analysis offers a huge scope of ineffective analysis of the attitude, behaviour, likes, and dislikes of an individual's towards entities such as products, services, events and topics, and their features. This field is widely appreciated by researchers due to its dynamic range of applications in numerous fields. There are several areas such as marketing, politics, news, analytics, etc. are benefiting from the results of Sentiment Analysis.

Our contribution mainly pointed at the following:

- To explore the feasibility of a convolutional neural network as a feature extractor for text classification of review dataset while most of the researchers utilized the feature learning capability of CNN for image classification.
- To improve the performance of deep learning techniques, we integrated the deep features with optimized Binary Particle Swarm Optimisation (BPSO) features based on traditional surface approaches. The method binary PSO possesses immense abilities to explore a large search space and rarely falls into local optima thus making it a nice choice for feature selection.
- We implemented a hybrid model of CNN with traditional classifiers SVM, which learns CNN features including word embedded and optimized feature set from binary PSO. Next, it switches from softmax to a stronger and more reliable classifier SVM [14], which is simple but convenient for classification tasks by considering negative log-likelihood as the loss function. The effectiveness of our proposed models, i.e., CNN-SVM is evaluated on the datasets of the movie domain.

Traditional CNN architecture is a combination of one input layer, convolution layer as a feature extractor. Next, the dimensions of features are reduced by a max-pooling layer, a fully connected layer, and finally an output layer. Most of the research with CNN typically employs the method of using a softmax function as the last layer activation function. However, we proposed SVM as a more powerful and secure classifier (rather than softmax function) to perform the sentiment classification task [15, 16]. In this research, we exhibit a trivial but stable improvement in resulting accuracy by replacing a softmax function with a non-linear SVM classifier. The proposed approach to incorporating an additional optimized feature vector with CNN features is more vigorous than applying CNN features or optimized feature set alone.

The rest of the paper is constructed as the following: Section II consists of the existing literature that can be connected to our approach. Section III explains the methodology in detail regarding the implementation of the proposed classification algorithm. This section also presents the details of text representation "Word Embedding" techniques and describes the training procedures of the Convolutional Neural Network (CNN) and classification model, respectively. The particulars about experimental results are expounded in Section IV. And finally, Section V concludes with a discussion of the proposed method with ideas for future steps.

II. Literature Survey

After the tremendous success in the field of computer vision, image classification, pattern recognition, deep neural models also gradually changed the scenario of the research field based on the review dataset. Sentiment analysis of review data on multiple domains aims to understand the sentiment of documents automatically using a deep learning framework.

Kim [17] presented a method to catch the non-linear structure of data for sentence classification using the CNN classifier. The proposed architecture consisted of the layers including pre-trained word embedding, convolutional with *n* filters, max-pooling, and a fully connected softmax layer. The author describes a simple modification to the architecture with "multi-channel" of word embedding, which enables us to use both task-specific vectors as well as a pre-trained vector. In this case, the word vectors in only one channel are fine-tuned at once by back-propagation training while the other is kept static. The experimental outcomes reported the potentiality of the method on the multi-domain dataset (movie reviews and product reviews). This existing work suggested pre-trained vectors can be used as "universal" feature extractors in other NLP related tasks. They have tested their model with various standard datasets on different domains of the movie (MR, SST-1, SST-2), product (customer review), and question-answering (TREC, MPQA).

Nogueira and Santos *et al*. [18] proposed a new deep convolutional neural network, which performs sentiment analysis tasks by utilizing the information from the character level to the sentence level. They considered two different domain datasets to evaluate their approach which is Stanford Sentiment Treebank (SSTb) and Stanford Twitter Sentiment (STS), respectively. The highest prediction accuracy of 86.5% was acquired with an STS corpus. Gupta *et al*. [19] performed sentiment classification by selecting the based on Particle-Swarm Optimization (PSO) within the framework of Conditional Random Field (CRF). They conduct the experiments on the SemEval-2014 Aspect based Sentiment Analysis Shared Task on review dataset from the product (laptop) and Hotel (restaurant) domain. They achieved the highest F-score of 81.96% for the laptop and 72.42% on the restaurant dataset.

Sun and Li *et al*. [20] have suggested a novel architecture by employing a convolutional neural network for identifying contextual information. The outcome of this investigation has defeated the traditional SVM and Naive Bayes models with an accuracy of 83.1%, 63.9%, and 67.8%, consecutively.

Yang *et al*. [21] compared the performance of three Machine Learning models, namely: Convolutional Neural Networks (CNN), Support Vector Machine (SVM) and Random Forest (RF) by utilizing pre-trained word embedding for feature representation in the clinical domain. They applied rule-based NLP algorithms to automatically generate weak labels and create large training datasets. The outcome indicates that CNN outperformed all other classifiers.

According to the investigation by Poria *et al*. [22], the CNN model has been employed to extract sentiment, emotion, and personality features for sarcasm detection.

They have proposed a CNN-SVM hybrid scheme, where they feed the extracted features of the fully connected layer of CNN to the classifier SVM for the endmost classification result.

Uisal and Murphey [23] proposed a comparative study between feature selection based approach and deep learning method based approach to perform document-level sentiment classification. The investigation is conducted based on combined feature sets including word embedding features and selected Bag of Words (BOW) features. Huang *et al.* [24] evaluates their approach with one traditional classifier SVM and three different deep learning models such as s CNN, LSTM, and long-term RCNN on four different datasets. The result indicates a deep learning model that performed best in three out of four datasets. They also reported that deep learning models are more suitable with fine-tuned word embedded features. Table 1 summarizes some important articles relevant to our research. It includes the year of publication, authors' names, research work, methods, datasets, and the study target.

Table 1. Summary of deep-learning-based sentiment analysis

No	Year	Study	Method	Dataset	Research Target
1.	2019	Mohamed *et al.* [25]	They have proposed a CNN model that considered the word level as well as character level embedding.	Movie reviews IMDb dataset	Proposed deep learning models have also outperformed SVM, Naïve Bayes and RNTN that were considered as standard methods in other research.
2.	2018	Chen *et al.* [26]	CNN architecture has employed as a feature extractor.	Corpus of product review dataset	Text Sentiment Analysis based on CNNs and SVM.
3.	2018	Li *et al.* [27]	Estimate the effects of textual quality of online Reviews on classification performance.	Movie reviews IMDb	Study the Impact of two main textual Features the word count and review readability.
4.	2018	Qian *et al.* [28]	Sentiment analysis based on DNN and CNN architecture.	Weather-related tweets	Feature extraction.
5.	2023	Başarslan *et al.* [29]	The presented efficiency of ensemble learning method Stacking and Voting.	Movie review dataset (Rotten tomatoes).	Proposed a fusion based model to analyze sentiments of multi domain dataset.
6.	2020	Mitra [30]	Sentiment analysis of short texts CNN, LSTM, on top of pre-trained word vectors	Movie Reviews IMDb	Rule based approach for sentiment classification Using Lexicon method.
7.	2022	Ramadhan *et al.* [31]	SVM classifier trained with Tf-IDf features filtering method	IMDb movie dataset	The authors have performed Emotion Recognition from Movie dataset.
8.	2018	Keerthi *et al.* [32]	Sentiment classification has been performed by applying Hybrid Feature Extraction Method (HFEM).	Movie Dataset IMDb	Combining machine learning features with the lexicon features to train the classification model SVM, KNN, Maximum Entropy and Naive Bayes.

III. Methodology

The main focus of this section is to investigate the performance of the deep learning method with a feature set generated from Deep Neural Network (DNN). Thus, the properties of CNN and SVM complement each other in such a way that their advantages are combined. A general overview of the proposed framework is summarized into several steps with Fig. 1 and the following subsections consist of detailed description (Algorithm 1) about each preliminary function mentioned (steps a–d) in our proposed approach.

- Data Collection: We used two publicly available datasets that provide user reviews and rating information. The datasets used for Classifications are the movie review datasets Stanford Sentiment Treebank SST-2 and IMDb have been used to evaluate the performance of the
- The review dataset is then esteemed for the process of feature selection and extraction. We train CNN by using SGD with a back-propagation algorithm to extract high-level feature vectors from different layers.
- We have carried out further experiments with a set of merged features. We employed the BPSO feature selection method to generate the most relevant optimized feature vector.
- Training of the SVM classifier with the output of the top hidden layer of CNN along with optimized binary PSO features. In our approach, CNN hasn't performed both feature extraction and classification within a single network structure. We have applied the SVM classifier to perform sentiment prediction or classification based on training with the above mentioned combined feature set.

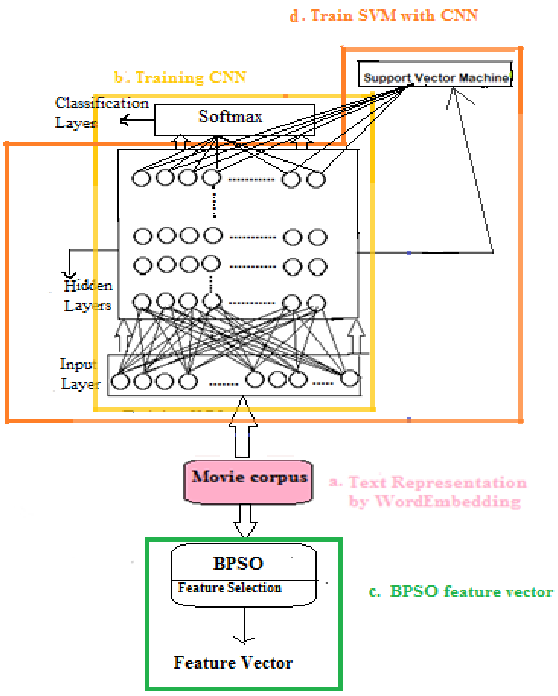

Fig. 1. The architecture of proposed framework for the sentiment classification task.

Algorithm 1. Algorithm for the proposed approach

Input: Two Movie datasets IMDb and SST-2: Training, validity, Testing; Consider stopping criteria: 0

Output: Classification result with **Prediction** and achieved **F-score** based on two polarities i.e., positive and negative with achieved accuracy.

1. *Train* ← TrainCNN with review text

2. *CNNnet*← BuildNetwork()

3. Start to train the network (*CNNnet*)
4. while stopping criteria not met
5. do *Train Network* (*CNNnet, Train*) /* train the network with SGD backpropagation/
6.end while
7. /* Training complete*/
8. $CNN_{Features}$← GetTopLayer(*CNNnet, Train*) features from a fully connected layer.
9. Applying BPSO(*Train)* on training samples to derive optimized features set.
10. $BPSO_{Opt\ Features}$← optimized features vector from BPSO.
11. $Combined_{Features\ Set}$← $CNN_{Features}$U$BPSO_{Opt\ Features}$
12. *Test* ← TestCNN with data portion allocated for testing from review text
13. $Test_{Features}$← GetTopLayer(*CNNnet,Test)*
14. Model 1:*CNN-SVM model1* ← $\mathrm{SVM}_{Combined_{FeaturesSet}}$
15. $Test_{ComFeaturesSet}$← $Test_{Features}$U$BPSO_{Opt\ Features}$
16. *PredictionResult*SVM← $\mathrm{TEST}_{\mathrm{SVM}}$(*CNN-SVM model1*, $Test_{ComFeaturesSet}$)
17. *F-scoreResultSVM*← Evaluation(MDDS, *PredictionResult*SVM)

A. Time Complexity

This research work exhibits that the CNN-SVM model is able to achieve a higher test accuracy of 93.53 % using the movie review dataset. CNN is composed of multiple layers such as Input layer, Convolutional layer, pooling layer, fully connected layer, and the classification layer with several shared parameters. In the architecture of CNN-SVM, the classification layer was changed from Softmax to SVM. According to Ref. [33], the time complexity of all conventional layers can be estimated as follows:

The total time complexity of all convolutional layers is:

$$O\left(\sum_{l=1}^{d} n_{l-1} \cdot s_l^2 \cdot n_l \cdot m_l^2\right) \tag{1}$$

where d indicates the number of convolutional layers or depth. The number of filters and the number of input channels in the l^{th} layer can be identified by n_l and n_{l-1} individually. s_l defines the filter length and m_l is the length (spatial size) of the output feature map. This time complexity can be estimated for both training as well astesting time, but the scale must be varied. The time complexity of pooling and fully connected layers and pooling layers is not included in the above formulation as these layers hardly consume 5–10% computational time.

The theoretical time and space complexities of the SVM model defined here based on various sources [34, 35]. The complexity of SVM mainly depends on the amounts of sampling data. In the case of computing the Time complexity, it can be assumed that n is the number of training samples and d is the feature dimension of a sample. The number of support vectors can be identified by n_{sv} to estimate the computational time of SVM.

The time complexity of SVM during training: $O(n^2d+n^3)$ and the prediction complexity of SVM with RBF kernel is $O(n_{sv}d)$.

B. Word Embedding Model

Word embedding techniques are a popular choice adapted from the Deep Learning model for learning the document representation directly from the data. In connection with NLP, the researchers Bengio *et al.* [36] firstproposed the terminology "word embedding" to train an Artificial Neural Network (ANN) model. The "Word2vec" is introduced by Mikolov *et al.* [37] as the most efficient prediction based model for learning word embedding.

Two most prosperous text representation models based on the word embedding techniques are a Continuous Bag of Words (CBOW) and the Skip-gram model. In this research, we have applied the CBOW Word2vec model to learn vector representations [38].

The CBOW model aims to predict the current word concerning the average of the context word vectors. But in the case of the Skip-gram model, the prediction of the surrounding words in a sentence is based on the given current word. The optimal word embedding with the highest probability of prediction accuracy for a given window size is proposed by these two Word2vec models. In comparison with the skip-gram model, the CBOW model is several times faster, which has allowed training models on large corpusquickly with better accuracy for the more repeated terms [39]. On the other hand, the skip-gram model works well with a small amount of training data. As mentioned earlier, the CBOW model is used to predict the current word vectors. Each word or term of text is mapped to a corresponding vector that is placed as a column in a matrix. The input layer of the proposed model is embedding of neighboring words and further these words are concatenated at a hidden layer to be used as features for determining the target word in a given sentence.

We have implemented the word embedding models by applying the Word2vec algorithm with a CBOW model (Fig. 2) on a large dataset composed of movie reviews [40].

We investigate the effect of using word embedding with CNN models. For an individual term or word in a sentence of input, we acquired a vector representation of a particular dimension that have fixed to 200.

The following notations are going to use in developing the word embedding model.

$D = \{D_l, ..., D_n\}$: a training corpus of size n, where each document D_l consists of a sequence of words $w_l,...,w_t$ of variable length. V: the vocabulary used in the training corpus.

The CBOW model trains two metrics.

(Input word matrix) $M_i \in R^{V\times N}$: The weight matrix from the input layer to the hidden layer of size N, we use Mi^w to represent the column (input vector) for word w.

(Output word matrix) $M_j \in R^{N\times V}$: The weight matrix from the hidden layer to the output layer, we use Mj^w to represent the column (output vector) for word w.

Initially, we need to generate word vectors as the representation of input context words ((w(c − x),...,w(c − 1), w(c + 1), ..., w(c + x)) of size x. Next, CBOW model attempts to predict w(c) as the center or target word. One hot vector for each word is needed to embed input context words. We have one hot word vector C (= 2x) and now the size of the input layer becomes [C×V]. Thereafter, we multiply them with input matrix *Mi* for getting embedded words of dimension N [V×N]. Now we can estimate the hidden layer output by multiplying hidden layer input with matrix *Mj* and the size of the score vector we obtain is [N×V]. The new score vector is denoted as "*s*". Finally, this new score vector "*s*" is extracted from the fully connected layer.

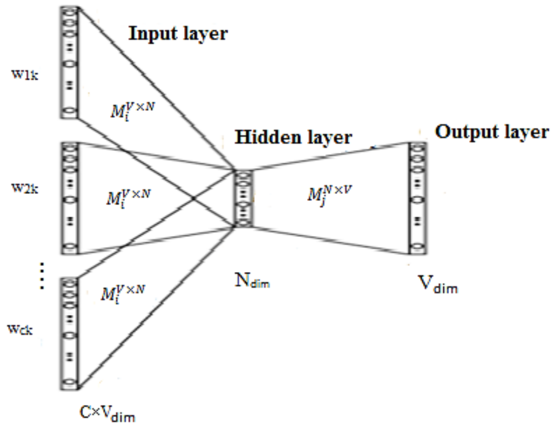

Fig. 2. CBOW: Continuous bag of words model.

C. An Ensemble of Features

This work aims to apply a straightforward method to combine deep features as well as an optimized feature set at the classification level. Hence, the composition of two feature sets carried out more knowledge which helped in achieving better performance than an individual can obtain.

1) CNN features

Proper feature selection is an essential task to enhance the result of the sentiment classification process. The performance of the classifier mostly depends on the feature set, if features selection performs well then the simplest classifier may also give a good F-Score through training. CNN performs reasonably well in capturing the relevant lexical and syntactic features on its own. We briefly describe the set of features that we use for building different models as follows.

- Word Embeddings: The embedded features have been captured through the CBOW model at the word embedding layer on CNN. The resulting representation of CNN architecture is then used to feed the classification model. According to the word embedding dictionary, each word has been encoded as a 200-dimensional vector.

2) BPSO features

We obtained another optimal subset of features by applying BPSO [41] a meta-heuristic algorithm. The binary PSO is the extended version of the stochastic optimization technique Particle Swarm Optimization (PSO). The primitive PSO technique [42] is chiefly prototyped for solving the global optimization problems by finding a good set of solutions in continuous space. PSO is established with a crowd or a swarm of random particles, where each particle is associated with a velocity. Every time the particles have a tendency to search for the best global optima by updating the velocity of each particle based on their position moves towards the best search space. Every particle keeps a record of two best positions, i.e., one best (p_{Best}) position is that the particle ever traversed and another is the global best position (g_{Best}), i.e., the position encountered by any particle so far in the whole population. Hence, a particle's velocity and position were updated as follows:

$$V_i^{New} \leftarrow w \times V_i^{Current} + c1 \times Rand \times (P^{Best} - X_i^{Current}) + c2 \times Rand \times (G^{Best} - X_i^{Current}) \quad (2)$$

$$X_i^{New} \leftarrow X_i^{Current} + V_i^{New} \quad (3)$$

where $V_i^{Current}$ and $X_i^{Current}$ are indicating the current velocity and position of particle *i*, respectively. *Rand* returns the randomly generated numbers. W represents the "inertia weight" and c1, c2 are the learning parameters.

The traditional PSO is designed to solve the real-number optimization problem. Therefore, Kennedy and Eberhart have proposed BPSO by extending the original PSO to tackle the binary numbers or discrete problems. In the case of BPSO, the velocity becomes a probability to determine whether the particle's position component is in state 1 or state 0. The normalized function which we used here is a sigmoid function *S* to transform all the values of velocity in real numbers in a binary range.

$$S(V_i) = \frac{1}{1+exp\,(-V_i)} \quad (4)$$

The aforementioned Eq. (2) that we used to compute the velocity of particles and the positions of a particle can be updated as follows:

$$X_i^{New} = \begin{cases} 0, & Rand < S(V_i) \; 1 \\ & \text{Otherwise} \end{cases} \quad (5)$$

where the "Rand" numbers for BPSO belong to {1, 0}, if the "Rand" is less than $S(V_i)$ of velocity, then X_i is set to be 1, otherwise, it's set to be 0.

D. Training of CNN as Feature Extractor

The proposed Convolutional Neural Network architecture has been exploited as a feature extractor to extract more meaningful, generalizable, and abstract features. The following subsections have explained this extraction process.

E. Architecture of CNN

Convolutional Neural Network is the extension of the traditionalMLP network, which is more relevant with a combination of convolutional filter layer; pooling layers and classification layer, etc. CNN is composed of multiple layers with several shared parameters [43]. Each layer performs a specific task of alternating its input into useful representation. In this proposed architecture (Fig. 3), CNN computed word embedding and sentence feature extraction concurrently.

We consider the sentence "The cast in the movie is also fantastic," as an example to explain how the architecture works to extract the features from review sentences.

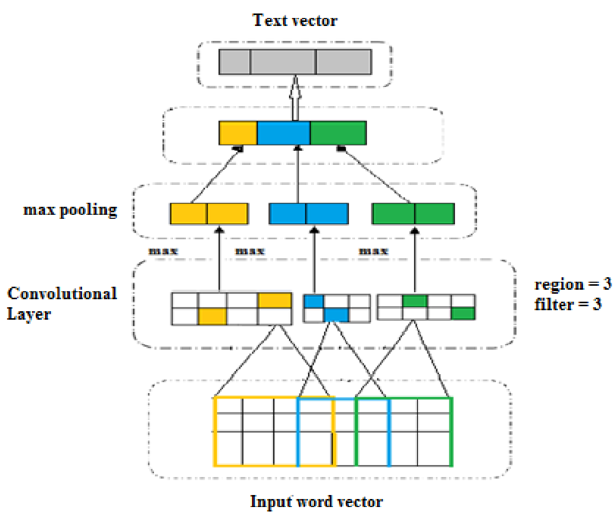

Fig. 3. The proposed CNN architecture for feature extraction.

- The first layer is called the embedding layer, every word in the sentence is encoded as a word vector, which has already been defined in the previous section. Suppose, we represent a vocabulary size |V| with word dimension N and then the embedding matrix can be defined as $W_E \in R^{V\times N}$. According to our example "The cast in movie fantastic", the correlating indexes, such as 1st, 6th, 10th, 13th, and 15th are selected for sentence representation.
- The first layer is called the embedding layer, every word in the sentence is encoded as a word vector, which has already been defined in the previous section. Suppose, we represent a vocabulary size |V| with word dimension N and then the embedding matrix can be defined as $W_E \in R^{V\times N}$. According to our example "The cast in movie fantastic", the correlating indexes, such as 1st, 6th, 10th, 13th, and 15th are selected for sentence representation.
- The convolutional layer consists of multiple learnable filters utilized to extract local features from input layers and builds a feature map. The idea behind convolution is to take the dot product of a vector of specific weights. Convolution operation involves a filter of $W_i \in R^{4\times d}$ employed with window size $x = 4$ as follows:

$$f_j^i = g(w_{j:j+4} \cdot W_i), \quad j = 1, 2, \dots, n, i = 1, 2, \dots, k \qquad (6)$$

where, $W_j$indicates the d-dimension of a word and n represents the size of the sentence. W_i $(i = 1, 2, \dots, k)$ defines the weight of the convolution mask and k denotes the number of filters. $g(x)$ is considered as the non-linear activation function. To produce the feature map as an output of each convolution hidden layer, a non-linear function "ReLU" has been used for this purpose. This function is applied with each possible window of words in the sentence $\{w_{1:x}, w_{2:x+4}, \dots, w_{n-x+1:n}\}$ to generate a feature map.

Therefore, we obtain k feature maps of n values at this convolutional layer. The convolution operation over the sentence pursues abstract representations of the phrases equipped with implicit semantic information. At each successive layer, this information spans due to an increasing number of words and the entire sentence. CNN combines the words using several hidden layers to extract the global features of the whole text.

- Next, we employed a max-pooling over the feature map to capture the most important feature with the highest value and the maximum value for each particular filter that can be considered as

$$F = [\max(f_j^i), \dots, \max(f_j^i)], \quad j=1, 2, \dots, n \qquad (7)$$

This layer passes the most significant features with max value to the next layer by applying a maximum pooling operation. Thus, pool layer helps in reducing the feature size as well as trainable parameters (downsampling) within the model.

- After conventional and pooling layers, we must add one fully-connected hidden layer with a set of neurons. These neurons have a full connection with all neurons in the proceeding layers. These hidden layers determine a specific output by summarizing the weight of previous layer features.

F. Training of Classification Models

In this study, we present a fusion model CNN-SVM to perform Sentiment Classification. The supremacy of this approach is to utilize the merits of two classifiers to the fullest by merging the possibilities of the Convolutional Neural Network (CNN) with superior classifiers Support Vector Machine (SVM).

In this section, it has been propounded to present CNN as a trainable feature extractor instead of a sentiment classifier. By having filters with a confined size, CNNs implement so-called local receptive fields. Contrarily, in simple dense layer architecture, all data points would be treated on the same footing and the spatial information would be lost. Creating a network that can take into account the spatial structure of the input data was the main reason behind the invention of CNNs. The feature extraction process has generated a long vector that is fed into a supervised Machine Learning classifier SVM. The softmax function is a generalization of the binary form of Logistic function which makes a big difference with the SVM classifier.The reasons to select SVM than typical CNN for classification are as follows:

- SVM classifier mainly focuses on binary classification (full separation between two classes), while Softmax assigns the decimal probabilities to each class.
- SVM tries to find the best accessible surface to make a separation between positive and negative training samples based on empirical risk (training set and test set error) minimization principle, whereas Softmax function is used for tuning the parameters by minimizing cross-entropy or maximizing the log-likelihood.
- SVM classifiers can avoid the overfitting issue by including regularization parameters, cross-validation methods and increasing the training data also helps in preventing overfitting in SVM.

IV. EXPERIMENTATION

This section exhibits the results of the experiments that have worked out to evaluate the performance of the proposed hybrid models. The experimental outcomes have reported that the hybrid model makes the best achievements on both feature extraction and polarity detection performance in comparison with some previous literature.

A. Experimental Datasets

In the previous section, we have already provided a detailed description of the dataset, which is utilized to judge the efficiency of the proposed model. These datasets belong to the movie domain, namely IMDb and SSTb. The review data of the movie domain are considered as an ideal large scale dataset to train the Neural Network model.

- **Internet Movie review Database (IMDb):** The IMDb dataset was first introduced by Mass *et al*. [44] as a benchmark movie domain database to perform Sentiment Analysis. It is a document-level Text Classification dataset http://www.cs.cornell.edu/people/pabo/ consisting of 50,000 binary (positive and negative) labeled movie reviews, which are equally separated into training and testing sets. The class distribution with each subset of data is balanced. In the case of testing, 20% labeled data has been utilized [45, 46]. The key feature of this review dataset is that each review is composed of several sentences.

- **Stanford Sentiment Treebank (SSTb):** The SSTbcorpus was utilized for the first time by Pang *et al*. [47] for Sentiment Analysis research. The extension of this SSTb as a benchmark dataset was used by Sochar *et al*. [48]. It composed of 11,855 samples from the movie review site Rotten Tomatoes, where each review holds a single sentence https://nlp.stanford.edu/ sentiment/. The database stop was separated into three different sets, specific sentences 8544, 2210, 1101 are utilized for training, testing, and validating respectively. An individual sentence is explained by the following 5 sentiment scales: negative—0, somewhat negative—1, neutral—2, somewhat positive—3, and positive—4. The extended version SST-2 dataset considered only binary labeled reviews (positive and negative).

B. Experimental Setup

We implemented our proposed model based on the Python module, which is a generic toolkit consisting of a large number of the most efficient and useful tools for high-performance deep learning analysis. We applied Keras 2.0.4 with TensorFlow backend (version 1.3.0) for the training and classification of CNN. Keras, running on top of Tensorflow, is a high-level open-source Neural Networks API compiled through Python. In our research, this deep learning library wrapped with TensorFlow is well suited to the training and testing of CNN since the main focus of Keras is to allow fast prototyping and experimentation. Keras has the potential to execute on both CPUs and GPUs without any restrictions. We installed Keras in Windows 7 using pip after the installation of the TensorFlow open-source library.

Traditional machine learning classifiers SVM has evaluated in this work. After the implementation of CNN, we have a large feature space from movie review datasets, which are incorporatedin the python package. In Particular, we have used the scikit-learn module with NumPy 1.8.1, SciPy 0.14.0 python library to intensify and enlarge the core python potentiality.

In this work, we first trained the CNN classifier for a limited number of epochs such as from 0 to 50. The result shows that CNN generally extracts better features at increasing epochs. The performance of CNN is booming according to the epoch. CNN with 0 epoch indicates that CNN has not been trained yet. During the training phase of CNN, the optimization function normally computes the gradient value, which is the partial derivative of the loss function in respect of weight. The weight value must be revised in the opposite direction of the gradient.

C. Result Analysis and Discussion

All the results regarding the experiments that have been carried out to demonstrate the efficiency of the proposed model are visualized by plotting the accuracy and F-score value.

1) Parameter (filter size and hidden units) optimizing for training CNN

The selection of the optimal number of units in the hidden layer is very important because it impacts the performance of the proposed models. The selection of the optimal number of hidden units mainly depends on the type of neural network model, the complications of the learning task, and up to a certain extent the number of training and testing samples. But, there are no exact rules to decide on the number of hidden layers and the number of neurons in each layer should be present.

Therefore, we have to try different combinations. We have finely tuned the parameters *Filter* and *Hidden unit* (the product of these two represents the size of the feature vector as an outcome of the pooling layer) through this experiment. The number of hidden neurons in each layer is gradually increased until performance saturates due to over-fitting. In this experiment, each hidden layer, we consider a different number of hidden units (i.e., n = 100/150/200/250/300). Table 2. Exhibits that the classification accuracy of the movie review dataset (validation) obtains the maximum accuracy of 93.4 when *Filter* is set to (4,5,6) with hidden unit 250.

Table 2. Parameter optimization of CNN on the validation dataset

Filter	Hidden unit				
	100	150	200	250	300
(1,2,3)	85.1	85.8	86.3	86.5	87.2
(2,3,4)	87.7	86.0	86.6	88.3	87.5
(3,4,5)	87.9	87.3	88.2	88.6	89.1
(4,5,6)	86.4	88.0	90.8	93.4	91.6
(5,6,7)	88.7	87.1	88.3	87.2	87.5

2) Performance on CNN during training and validation

Training loss (*Train_loss*), training accuracy (*Train_acc*), validation loss (*val_loss*) and validation accuracy (*val_acc*) are plotted in Fig. 4. The accuracy and loss curves are set to 50 with the training epoch while training. An under-fitting model usually performs competently on the training data set and poorly on the test dataset. Over-fitting occurs when you achieve a good fit of your model on the training data, while it does not generalize well on new, unseen data.

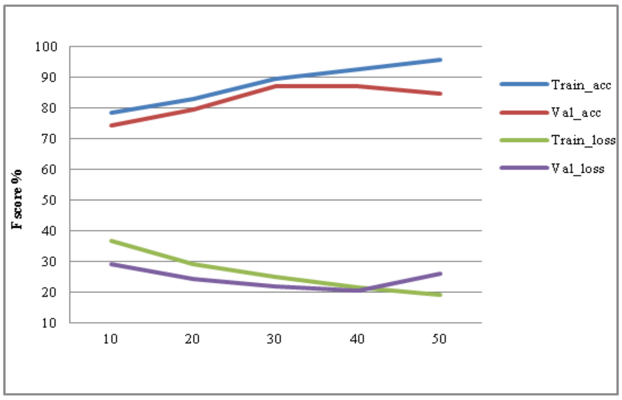

Fig. 4. Accuracy and Loss of CNN during training and validation.

In this experiment, a large gap between training and validation accuracy occurred after 40 epochs and that may indicate the presence of over-fitting. The highest accuracy of validation data was found to be 89.1%, corresponding to a training accuracy of 95.4%.

3) Performance of different base models

We compare our proposed model with the following baseline methods.

- *SVM*^{BASE}: A traditional SVM based model that can absorb all kinds of features.

- CNN_{WE}^{BASE} :This is a fundamental CNN based model. Word embedded features have been utilized for training and evaluating the proposed framework.
- $CNN\text{-}SVM_{WE}^{BASE}$: This is an SVM based model fed with word embedding features extracted from the basic CNN model.
- CNN_{WE+OP}^{BASE}: This consists quite similar to the basic CNN except this model of an optimized feature set, that extracts from the BPSO feature selection method.

The experiments which we have organized produce the individual performance regarding Sentiment Analysis of each base classifier on two different domain datasets and all the results of the proposed approach are exhibited in Fig. 5. The outcomes of the proposed model CNN-SVM with the IMDb dataset beat the performance of six baseline methods.

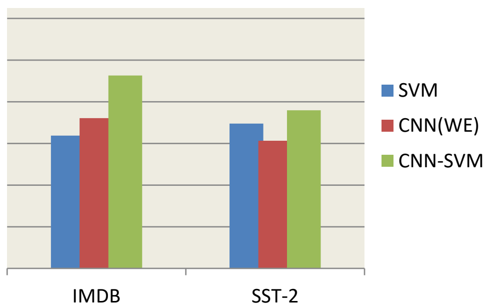

Fig. 5. Experimental results (% Accuracy) of the proposed model for movie review data set.

The model CNN-SVM with the combined feature vectors achieves an accuracy of 93.53% and 90.18% for the datasets IMDb and SST-2, respectively. In the case of the movie domain, the results impressively outperformed the baseline method SVM and CNN by +8.74% and +5.62%, respectively. The result indicates positive in favor of the CNN feature set. However,when we use the combined features set, CNN alone can't beat the proposed model with reduced accuracy –2.56% for IMDb and –4.28% for the SST-2 dataset.

4) Generalization of the model

Generalization is the potential of a machine learning trained model to correctly predict data samples that were not exploited for training. A worthwhile performance of the generalized method has been considered as a key target of any learning algorithm.To test the generalizability of the models, we have trained our framework on a complete IMDb dataset and tested on the SST-2 dataset (Fig. 6). Then we train SVM binary classifiers with the extracted features based on training and testing splits to demonstrate the generalizability of the model concerning the movie genre. If a classifier trained on one of the genres performs poorly on the others, then there might be a reason to believe that sentiment can have genre-specific qualities.

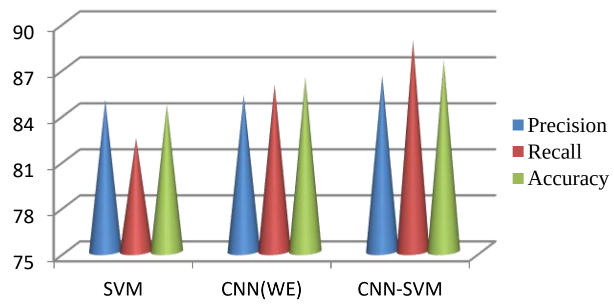

Fig. 6. Cross-dataset (IMDb and SST-2) comparison through classification accuracy.

This generalization better indicates the differences across the two datasets by showing how well a model trained on one data set can predict the positivity or negativity in reviews from other datasets. However, results in Tables 3 and 4 exhibit that, at least for movie genres, there are slight differences in performance on the different review datasets.

Table 3. Experimental results of the proposed model for IMDb datase

Classifier	IMDb (Movie domain)			Precision	Recall	F-Measure
	Confusion matrix					
		Positive	Negative			
SVM	Positive	15564 (TP)	1676 (FN)	0.920	0.908	0.91
	Negative	1511 (FP)	15729 (TN)			

Table 4. Experimental results of the proposed model for the SST-2 movie review

Classifier	SST-2 (Movie domain)			Precision	Recall	F-Measure
	Confusion matrix					
		Positive	Negative			
SVM	Positive	7193 (TP)	821 (FN)	0.904	0.897	0.898
	Negative	756 (FP)	7258 (TN)			

D. Comparison with Base Models and Existing Approaches

This section compares between the accuracy of the proposed approach with other existing approaches considering the datasets from the movie review domain. There are two main key aspects of our proposed architecture which make these models very potential to outperform state-of-the-art approaches. First, we applied Convolutional Neural Network (CNN) architecture which consists of linear and non-linear filters that can fit the data in a better way than a linear model. Second, the concatenation of CNN word embedded features along with negation features can beat the performance of a state-of-the-art method that uses either word embedded features or handcrafted features.

It is generally difficult to directly compare with the related studies, because the processing techniques, selection methods, and classifiers applied are different for most of the cases in different domains. Therefore, we select all those studies related to the same domain and other domains also for comparison. In particular, those state-of-the-art methods selectedfor evaluation must be considered Deep Neural Networks (DNN) with traditional ML classifiers for their experiments. The trained CNN model as a feature extractor performed very well on a variety of classification tasks. In Ref. [17], the researchers have reported that simple CNN with static vectors and minimum tuning of hyper-parameters achieved better performance on multiple data sets. They added learning task-specific vectors and improved the results with high accuracy of 89.6%. In this research, we have received the best F-Score value 93.53% for IMDb movie review datasets.

As shown in Table 5, the methods by Liu *et al*. [49] performed with the highest accuracy 91.2% for the IMDb dataset. On the contrary, our proposed models produce a quite impressive performance on this dataset by achieving the highest accuracy of 93.53%.

Table 5. Performance evaluation of the proposed approach with a state-of-the-art method on the movie database

Reference	Model	Datasets	
		IMDb	SST-2
Zhang *et al*. 2017[47]	CNN^{WE}	91.3%	88.4%
Gao *et al*. 2016 [14]	CNN	89.5%	-
Tay *et al.* 2019 [48]	LSTM-CNN	90.0%	80.5%
Liu *et al.* 2016 [49]	CNN	91.2%	87.8%
Proposed Model	CNN-SVM	93.53%	90.18%

The performance of the proposed approach has been compared in Table 6 with previous research works that have used the same method on a different domain. Akhtar *et al.* [16] proposed hybrid deep learning architecture CNN-SVM for sentiment analysis on four Hindi review datasets on various domains. They utilized combined features set of automatic CNN features and handcrafted Multi-Objective Genetic (MOO) algorithm based features. At the final stage of classification, they replaced the SoftMax layer by the SVM classifier. The highest accuracy they obtained 65.96% with the Hindi review dataset. The deep convolutional neural network was very appreciated by the researchers who have performed sentiment classification of images. Our work is philosophically similar to the work presented in [51] on the image dataset. They employed a pre-trained CNN classifier on the image dataset as a feature extractor and the extracted features are incorporated as training input to SVM and RF classifier. In contrast to other existing literature, they considered lower layer features instead of using features from a fully connected layer alone. This classifier has been chosen due to its superior performance over a single decision tree with the ability to rank the attributes according to their predictive importance. Table 7 compares the performance of the proposed approach with other existing approaches, especially in terms of F-Measure.

Table 6. Performance evaluation of the proposed approach on different domain datasets

Reference	Model	Dataset	Accuracy
Akhtar *et al.* 2016 [16]	CNN-SVM	Hindi Text Review Data	86.7%
Liao *et al.* 2017 [50]	CNN-SVM	STS Gold twitter dataset	75.39%
Athiwaratkun *et al.* 2015 [51]	CNN-SVM	ImageNet	72.13%
Proposed Model	CNN-SVM	IMDb, SST-2	93.53%, 90.18%

Table 7. Comparative results obtained by existing works using different ensemble feature selection methods

Author	Dataset	Feature Selection	Classifier	Performance
Alnashwan *et al*. [52]	Movie	Ensemble (voting)	SVM MNB	86.52% (F-Measure) 87.94%
Ali *et al.* [53]	Amazon product review	U-EFS (IG, CHI)	RF LR	87.23% (F-Measure) 89.15%
Dwivedi *et al*. [54]	Cornell movie review dataset	Sentiment Lexicon Features	Rule based model	83.0% (F-Measure)
Tripathy*et al.* [5]	Movie (IMDb)	N-Gram features	SVM ME NB SGD	88.94% (F-Measure) 88.48% 86.23% 85.11%
Pham *et al*. [55]	Movie (IMDb)	N-gram features andnrating based features	Lib LINEAR SVM	89.87% (Accuracy)
Proposed Model	CNN-SVM	IMDb, SST-2	CNN-SVM	91.0% (F-Measure)

A comparative report in Table 7 has considered only those works that utilized different ensemble methods for feature identification. The classification task of all those studies mentioned in Table 6 has been performed by traditional machine learning classifiers. In comparison with Table 5, the resulting values clearly indicates [45, 46] that the Deep Learning techniques achieve impressive classification accuracy in comparison with traditional Machine Learning approaches.

V. Conclusion

In this chapter, we conclude this proposition by summarizing the contributions and discussing the limitations with future research work.

A. Contributions

This paper aims to explore the effectiveness of our proposed hybrid deep learning framework to perform movie and product reviews based on sentiment classification. We have first utilized Convolutional neural network CNN to learn word embedded feature vectors and after that combine these CNN features with an external optimized feature vector. This work exploits the effectiveness of the ensemble of traditional sentiment classifiers and the combination of sentiment-embedded feature vectors along with handcrafted optimized features. To generate an external feature set, we applied a feature selection wrapper method, BPSO. Finally, the combined feature set fed to the SVM classifier and our proposed models detects the polarity of sentiments for various datasets from two different domains such as movie and product reviews. Furthermore, we defined various deep learning baseline models to evaluate our proposed models. For evaluation purposes, we considered two movie domain datasets. The outcomes of our proposed model provide competing accuracy and F-Score in comparison with the baseline model as well as a state-of-the-art model. According to our knowledge, very limited approaches exist in the domain of sentiment analysis, considered to combine the traditional and deep learning methods into a similar frame. Therefore, our proposed novel ensemble learning methodology of aggregating the feature learning capability of CNN and the merits of traditional classifiers is an important contribution to the area of empirical natural language processing.

B. Limitations

The combined approach of CNN with SVM outperforms the base classifiers, while some limitations were also found in applying the CNN classifier as follows.

- To reduce overfitting caused by the small amount of data, the architecture of CNN needs a huge dataset for training purposes. However, sometimes it is very difficult to find large labeled datasets on the internet.
- Training the CNN model with a large number of parameters increases thecomputationand memorycost.
- The proposed model achieves a great accuracy of binary classification where strong clues are present, but it fails to detect the neutral sentiment. The comment in text format contains sarcasm, linguistic problems, etc. To predict the sentiment of that comment we have to understand the nature and the ambiance of the comment thoroughly, as a single word can create a contradiction with the polarity of the comment. However, this aspect is also not considered in this paper.

C. Future Work

In future work, we would like to address the above-mentioned shortcomings of our present work as well as planning to design a novel neural network to perform multimodal emotion recognition and sentiment analysis by extracting multimodal data such as audio, video, and text. Furthermore, we desire to expand the domain of the proposed models to other languages, such as Bengali, Hindi, and others.

CONFLICT OF INTEREST

The authors declare no conflict of interest.

AUTHOR CONTRIBUTIONS

All mentioned authors contribute in the elaboration of the article. The main contributions of AC are involved in data collection, analyzing those data and also provided valuable inputs in writing the paper. MG conducted the research; IP contributed in the experimentation and resulting section. All authors have read and approved the final manuscript.

REFERENCE

[1] L. Zhang and B. Liu, "Sentiment analysis and opinion mining," in *Encyclopedia of Machine Learning and Data Mining*, Springer, Boston, MA, 2017.

[2] M. Mostafa, "More than words: Social networks text mining for consumer brand sentiments," *Expert Systems with Applications*, vol. 40, pp. 4241–4251, 2013. https://doi.org/10.1016/j.eswa.2013.01.019

[3] D. M. E. M. Hussein, "A survey on sentiment analysis challenges," *Journal of King Saud University—Engineering Sciences*, vol. 34, no. 4, pp. 330–338, 2018. https://doi.org/10.1016/j.jksues.2016.04.002

[4] B. Ghaddar and J. Naoum-Sawaya, "High dimensional data classification and feature selection using support vector machines," *European Journal of Operational Research*, vol. 265, no. 3, pp. 993–1004, 2018. https://doi.org/10.1016/j.ejor.2017.08.040

[5] A.Tripathy, A. Agrawal, and S. K. Rath, "Classification of sentiment reviews using n-gram machine learning approach," *Expert Systems with Applications*, vol. 57, pp. 117–126, 2016. https://doi.org/10.1016/j.eswa.2016.03.028

[6] X. Sun, C. Li, and F. Ren, "Sentiment analysis for chinese microblog based on deep neural networks with convolutional extension features," *Neurocomputing*, vol. 210, pp. 227–236, 2016. https://doi.org/10.1016/j.neucom.2016.02.077

[7] Q. T. Ain, M. Ali, A. Riaz, A. Noureen, M. Kamran, B. Hayat, and A. Rehman, "Sentiment analysis using deep learning techniques: A review," *International Journal of Advance Computer Science Application*, vol. 8, no. 6, pp. 424–437, 2016.

[8] S. Dara and H. Banka, "Annavarapu CSR rough based hybrid binary PSOalgorithm for flat feature selection and classification in gene expression data," *Annals of Data Science*, pp. 341–360, 2017. https://doi.org/10.1007/s40745-017-0106-3

[9] L. Zhang, S. Wang, and B. Liu, "Deep learning for sentiment analysis: A survey," in *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery*, e1253, 2018.

[10] K. Zhang, W. Zuo, Y. Chen, D. Meng, and L. Zhang, "Beyond a Gaussian denoiser: Residual learning of deep CNN for image denoising," *IEEE Transactions on Image Processing*, vol. 26, no. 7, pp. 3142–3155, 2017. doi: 10.1109/TIP.2017.2662206

[11] G. Vinodhini and R. M. Chandrasekaran, "A comparative performance evaluation of neural network based approach for sentiment classification of online reviews," *Journal of King Saud University-Computer and Information Sciences*, vol. 28, no. 1, pp. 2–12, 2016. https://doi.org/10.1016/j.jksuci.2014.03.024

[12] S. Lai, L. Xu, K. Liu, and J. Zhao, "Recurrent convolutional neural networks for text classification," in *Proc. 29th AAAI Conference on Artificial Intelligence*, 2015.

[13] A. Severyn and A. Moschitti, "Twitter sentiment analysis with deep convolutional neural networks," in *Proc. the 38th International ACM SIGIR Conference on Research and Development in Information Retrieva*l, Santiago, 2015, pp. 959–962.

[14] Y. Chen and Z. Zhang, "Research on text sentiment analysis based on CNNs and SVM," in *Proc. 13th IEEE Conference on Industrial Electronics and Applications (ICIEA)*, 2018. doi: 10.1109/ICIEA.2018.8398173

[15] K. Zhao and Y. Jin, "A hybrid method for sentiment classification in Chinese movie reviews based on sentiment labels," in *Proc. IEEE 2015 International Conference on Asian Language Processing (IALP)*, 2015. doi: 10.1109/IALP.2015.7451538

[16] M. S. Akhtar, A. Kumar, A. Ekbal, and P. Bhattacharyya, "A hybrid deep learning architecture for sentiment analysis," in *Proc. 26th International Conference on Computational Linguistics, COLING*, Japan, 2016, pp. 482–493.

[17] Y. Kim, "Convolutionalneural networks for sentence classification," in *Proc. the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, Qatar, 2014, pp. 1746–1751.

[18] C. Nogueira, D. Santos, and M. Gatti, "Deep convolutional neural networks for sentiment analysis of short texts," in *Proc. 25th International Conference on Computational Linguistics COLING*, Dublin, 2014, pp. 69–78.

[19] D. Gupta, R. S. Reddy, and A. Ekbal,"PSO-ASent: Feature selection using particle swarm optimization for aspect based sentiment analysis," in *Proc. International Conference on Applications of Natural Language Processing to Information Systems*, 2015, pp. 220–233.

[20] X. Sun, C. Li, and F. Ren, "Sentiment analysis for Chinese microblog based on deep neural networks with convolutional extension features," *Neurocomputing*, vol. 210, pp. 227–236, 2016.

[21] Y. Yang *et al.*, "Adeep representation empowered distant supervision paradigm for clinical information extraction," arXiv preprint, arXiv:1804.07814v1, 2018.

[22] S. Poria, E. Cambria, D. Hazarika, and P. Vij, "A deeper look into sarcastic tweets using deep convolutional neural networks," arXiv preprint, arXiv:1610.08815v2, 2016.

[23] A. Uisal and Y. Murphey, "Sentiment classification: Feature selection based approaches versus deep learning," in *Proc. IEEE International*

Conference on Computer and Information Technology (CIT), 2017. doi: 10.1109/CIT.2017.53

[24] Q. Huang, R. Chen, X. Zheng, and Z. Dong, "Deep sentiment representation based on CNN and LSTM," in *Proc. IEEE International Conference on Green Informatics (ICGI)*, 2017. doi: 10.1109/ICGI.2017.45

[25] A. N. Mohamed, A. Mostafa, E. Hamid, and A. Youssif, "Sentiment analysis for movies reviews dataset using deep learning models," *International Journal of Data Mining & Knowledge Management Process (IJDKP)*, vol. 9, no. 3, 2019. doi: 10.5121/ijdkp.2019.9302

[26] Y. Chen and Z. Zhang, "Research on text sentiment analysis based on CNNs and SVM," in *Proc. 13th IEEE Conference on Industrial Electronics and Applications (ICIEA)*, 2018. doi: 10.1109/ICIEA.2018.8398173

[27] L. Li, T. T. Goh, and D. Jin, "How textual quality of online reviews affect classification performance: A case of deep learning sentiment analysis," *Neural Computing Application*, pp. 1–29, 2018.

[28] J. Qian, Z. Niu, and C. Shi, "Sentimentanalysis model on weather related tweets with deep neural network," in *Proc. the 10th International Conference on Machine Learning and Computing*, Macau, China, pp. 31–35, 2018.

[29] M. S. Başarslan and F. Kayaalp, "Sentiment analysis with ensemble and machine learning methods in multi-domain datasets," *Turkish Journal of Engineering*, vol. 7, no. 2, pp. 141–148, 2023.

[30] A. Mitra, "Sentiment analysis using machine learning approaches (Lexicon based on movie review dataset)," *Journal of Ubiquitous Computing and Communication Technologies (UCCT)*, vol. 2, no. 3, pp. 145–152, 2020.

[31] A. Rahman and M. D. Hossen, "Sentiment analysis on movie review data using machine learning approach," in *Proc. 2019 International Conference on Bangla Speech and Language Processing (ICBSLP)*, IEEE, 2019, pp. 1–4.

[32] B. S. Harish, K. Kumar, and H. K. Darshan, "Sentiment analysis on IMDb movie reviews using hybrid feature extraction method," *International Journal of Interactive Multimedia and Artificial Intelligence*, vol. 5, no. 5, pp. 109–116,2019.

[33] K. He and J. Sun, "Convolutional neural networks at constrained time cost," in *Proc. IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2015, pp. 5353–5360.

[34] A. P. Gopi, R. N. S. Jyothi *et al.*, "Classification of tweets data based on polarity using improved RBF kernel of SVM," *International Journal of Information Technology*, 2020. https://doi.org/10.1007/s41870-019-00409-4

[35] J. Kim, S. Jang, S. Choi, and E. Park, "Text classification using capsules," *Neurocomputing*, vol. 376, pp. 214–212, 2020.

[36] Y. Bengio, A. Courville, and P. Vincent, "Representation learning: A review and new perspectives," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 35, no. 8, pp. 1798–1828, 2017.

[37] T. Mikolov, I. Sutskever, K. Chen, G. Corrado, and J. Dean, "Distributed representations of words and phrases and their compositionality," in *Proc. 26th Advances in Neural Information Processing Systems*, Lake Tahoe, Nevada, 2013, pp. 3111–3119.

[38] N. Majumder, S. Poria, A. Gelbukh, and E. Cambria, "Deep learning-based document modeling for personality detection from text," *IEEE Intelligent Systems*, vol. 32, no. 2, pp. 74–79, 2018.

[39] H. Zhang and G. Zhong, "Improving short text classification by learning vector representations of both words and hidden topics," *Knowledge-Based Systems*, vol. 102, pp. 76–86, 2016.

[40] M. Giatsoglou, M. G. Vozalis, K. Diamantaras, A. Vakali, G. Sarigiannidis, and K. C. Chatzisavvas, "Sentiment analysis leveraging emotions and word embeddings," *Expert Systems with Applications*, vol. 69, pp. 214–224, 2017.

[41] H. N. Pour, M. R. Shahrbabaki, and M. M. Farsangi, "Binary particle swarm optimization: Challenges and new solutions," *The Journal of Computer Society of Iran on Computer Science and Engineering*, vol. 6, no. 1, pp. 1–32, 2008.

[42] M. A. Ahmed, M. Hassan, and A. Hassan, "PSO-based feature selection for Arabic text summarization," *Journal of Universal Computer Science*, vol. 21, no.11, pp. 1454–1469, 2008.

[43] Y. LeCun, Y. Bengio, and G. Hinton,"Deep learning," *Nature*, vol. 521, pp. 436–444, 2015.

[44] A. L. Maas, R. E. Daly, P. T. Pham, D. Huang, A. Y. Ng, and C. Potts, "Learning word vectors for sentiment analysis," in *Proc. the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, 2011, pp. 142–150.

[45] B. Pang, L. Lee, and S. Vaithyanathan, "Thumbs up? Sentiment classification using machine learning techniques," in *Proc.* the *Conference on Empirical Methods in Natural Language Processing (EMNLP 2002)*, 2016, pp. 79–86.

[46] R. Socher, C. C. Lin, A. Y. Ng, and C. D. Manning, "Parsing natural scenes and natural language with recursive neural networks," in *Proc. the 26th International Conference on Machine Learning (ICML 2011)*, 2011.

[47] H. Zhang, L. Xiao, W. Chen, Y. Wang, and J. Yaohui, "Multitask label embedding for text classification," arXiv preprint, arXiv:1710.07210, 2017.

[48] T. Yi, T. Anh, S. Luu, and H. Cheung, "Recurrently controlled recurrent networks," *Advances in Neural Information Processing Systems*, pp. 4731–4743, 2019.

[49] P. Liu, X. Qiu, and X. Huang, "Deep multi-task learning with shared memory," in *Proc. 2016 Conference on Empirical Methods in Natural Language Processing (ACL 2016)*, 2016, pp. 118–127.

[50] S. Liao, J. Wang, R. Yu, K. Sato, and Z. Cheng, "CNN for situations understanding based on sentiment analysis of twitter data," *Procedia Computer Science*, vol. 111, pp. 376–381, 2017.

[51] B. Athiwaratkun and K. Kang, "Feature representation in convolutional neural networks," arXiv preprint, arXiv:1507.02313, 2015.

[52] R. Alnashwan, A. P. O'Riordan, and H. Sorensen, "Improving sentiment analysis through ensemble learning of meta-level features," in *Proc. KDWEB 2016: 2nd International Workshop on Knowledge Discovery on the Web*. *Sun SITE Central Europe (CEUR)/RWTH Aachen University*, Aachen, 2016.

[53] M. Ali, S. I. Ali, D. Kim, T. Hur, J. Bang, S. Lee, B. H. Kang, and M. Hussain, "uEFS: An efficient and comprehensive ensemble-based feature selection methodology to select informative features," *PloS One*, pp. 1–28, 2018. https://doi.org/10.1371/journal.pone.0202705

[54] R. K. Dwivedi, M. Aggarwal, S. K. Keshari, and A. Kumar, "Sentiment analysis and feature extraction using Rule-Based Model (RBM)," in *Proc. International Conference on Innovative Computing and Communications. Lecture Notes in Networks and System*, 2019.

[55] S. B. Pham *et al.*, "Sentimentclassification on polarity reviews: An empirical study using ratingbased features," in *Proc. the 5th Workshop on Computational Approaches to Subjectivity, Sentiment and Social Media Analysis*, 2014, pp. 128–135.

[56] O. Dele, T. Michael, and F. E. Ayetiran, "Social unrest prediction through sentiment analysis on twitter using support vector machine," *Experimental Study on Nigeria's Open Information Science*, vol. 7, no. 1, 2023. https://doi.org/10.1515/opis-2022-0141

[57] E. F. Ayetiran, "Attention-based aspect sentiment classification using enhanced learning through cnn-Bilstm networks," *Knowledge-Based Systems*, vol. 252, 109409, 2022

[58] A. Gupta, P. Tyagi, T. Choudhury, and M. Shamoon, "Sentiment analysis using support vector machine," in *Proc. 2019 International Conference on contemporary Computing and Informatics (IC3I)*, Singapore, 2019, pp. 49–53. doi: 10.1109/IC3I46837.2019.9055645