

# Rider Driven African Vulture Optimization with Multi Kernel Structured Text Convolutional Neural Network for Classifying e-Commerce Reviews

H. Mohamed Zakir<sup>1\*</sup>

Department of Computer Science  
Noorul Islam Centre for Higher Education  
Kumaracoil, Thuckalay, India

Dr.S.Vinila Jinny<sup>2</sup>

School of Computer Science & Engineering  
Vellore Institute of Technology  
Vellore, India

**Abstract**—Opinion mining is a natural language processing based on sentiment classification technique to determine the sentiment of the reviews. The major existing text Convolutional Neural Network (CNN) algorithms are derived based on  $3 \times 3$  size kernels which extract ineffective review text-features and lead to less classification accuracy. Moreover, most of the traditional CNN versions output three classes such as positive, negative, and neutral as their classification results. Hence, a novel algorithm namely 'RAVO driven Multi-Size Kernel structured Text CNN for classifying ecommerce reviews (MSK-TCNN-RAVO)' is proposed in this work. This proposed approach utilizes five multi-size kernels ( $3 \times 7, 5 \times 7, 1 \times 3, 1 \times 5, 1 \times 7$ ), multi-dimensional kernels (1D & 2D), and integrates varying size kernels to extract text-features effectively. In addition, the performance of multi-kernel CNN is highly enhanced by RAVO algorithm based on rider optimization. Moreover, the proposed approach is highly effective to process 'review-stop-words removal' that decrease the complexity and time consumption of the opinion mining process. Most existing systems use single pooling operations which reduce feature map processing performance, hence, dual pooling operations (both Max and Average pooling) are employed in this research. Furthermore, it is configured to generate five classification outputs such as bad, fair, neutral, good, and excellent to support better decision-making with 95.5% accuracy. This method is evaluated using different quality metrics and five review-databases to measure the performance, and the results reveal that the proposed method outperforms the other existing review classification algorithms.

**Keywords**—Natural language processing; opinion mining; convolutional neural network; text sentiment classification; ecommerce review

## I. INTRODUCTION

Social media plays a very important role in almost everybody's day to day life. It allows the people to convey what they think and feel about the products in the e-commerce website. This is called an opinion or review. Online shopping is a form of e-commerce which allows consumers to directly buy goods or services from a seller over the internet [1]. Most of the e-commerce websites incorporate provisions which enable the consumer to post their opinions about the product, companies, or their experiences etc.

Online shopping is being performed by millions of users every day, as a result of this, a huge amount of reviews are

being generated constantly. Handling these reviews manually to extract knowledge is a tedious task and hence, companies use classification tools to categorize the customer reviews to understand the customer's mindset. Opinion Mining is also known as Sentiment Analysis (SA) is the automated process of identifying opinions in text, and labeling them as positive, negative or neutral, based on the emotions expressed by the customer. It includes text analysis, computational linguistics to identify and extract subjective information in source materials [2].

The rest of the article is structured as follows: A summary of traditional methodologies for review classification is provided in Section II. Section III covers the proposed approach in detail with diagrammatic representations. Results and discussions are included in Section IV. The work has been concluded in Section V.

## II. RELATED WORK

The existing review classification algorithms discussed in various works of literature are summarized in this section. Anam et al. [3] proposed a voting classifier (LR-SGD) model for textual-tweets classification as happy or unhappy for emotion recognition. The dataset contains a lot of contrary tweets which are used for evaluation. The weakness of this work is that, the combination of different models should be employed to increase the performance. Ren and Wu [4] proposed a sentence-based analysis model to identify investor herd behaviour. The data is taken from blue-chip stocks in Chinese stock market. The work's limitation is that it uses only a small amount of data for experimentation, arising doubt on the algorithm's efficacy.

Sasikala and Sheela [5] expressed a sentiment analysis technique called Deep learning modified neural network to process the online product reviews. The food review dataset has been taken as input for the proposed technique. The drawback is that, the keyword processing only detects the sentiment expressed in a single word, and frequently fails to provide all information needed to interpret the context. Zeeshan et al. [6] proposed an opinion mining methodology using lexicon and neural networks for classifying online movie reviews. The reviews are collected from the IMDB database. The disadvantage is that, the resultant vectors will be in larger dimension and contain a large number of null

\*Corresponding Author.

values, resulting in sparse vectors. Jia et al. [7] proposed a hierarchical gated deep memory network with position-aware for aspect-based sentiment analysis. The algorithm is evaluated on laptop and restaurant datasets. The limitation of the work is that the accuracy is very low while predicting the neutral polarity.

Cheng et al. [8] proposed a multi-channel model that combines the CNN and the bidirectional gated recurrent unit network with attention mechanism for text sentiment orientation analysis. Two public datasets such as IMDB and Yelp 2015 review dataset is used to evaluate the algorithm. It does not include any syntactic structure features, hence, this model cannot be used for sentiment classification task. Dong et al. [9] proposed a CNN based on multiple convolutions and pooling for text sentiment classification. The reviews from English and Chinese datasets are used as input for evaluation. The four convolution operations proposed in this work perform low on English emotions.

Gupta et al. [10] proposed a feature-based supervised model to identify the extremist reviewers who target whole brand. The datasets are created by crawling reviews from Amazon website. The weakness of the work is that it requires larger training time for even smaller datasets. Madbouly et al. [11] proposed a hybrid classification approach of tweets based on user ranking for online social networks. The dataset consists of tweets with their feature used as input. Tweets were filtered manually to exclude non-English tweets which will increase the execution time of the algorithm. Bhalla and Bagga [12] proposed an RB-Bayes method based on Naive Bayes's theorem for prediction to remove problem of zero likelihood. The algorithm is evaluated on a small dataset which contains text data. The efficiency of the model is not proved in large-scale databases. Zhang and Zhong et al. [13] proposed an e-commerce reviews mining method called sentiment similarity analysis to explore user's similarity and trust. The experimental dataset is collected from Amazon.com. The essential parameters involved in calculating the trust between users is not considered in this work which leads to less accuracy.

Aziz et al. [14] proposed the Contextual analysis mechanism to find the relationship between words and sources to predict Supervised machine learning model performance. The experiment is conducted on four different domain datasets collected from Amazon. The result of the prediction algorithm is less while performing real time analysis for individual changes of dataset. Iqbal et al. [15] proposed a hybrid framework for sentiment analysis which bridges the gap between lexicon-based and machine learning approaches. The reviews dataset is collected from UCI ML repository. The proposed framework works only for specific domain and does not support other domains like cyber-intelligence, law-enforcement sector, etc.

Liu et al. [16] proposed a modified fuzzy approach for cyber hate classification. This model uses four datasets collected from Twitter regarding four types of hate speech. Because the intersectionality of different types of hate speech is not addressed, more diversified features are extracted for hate speech detection. Fang et al. [17] proposed a multi-

strategy sentiment analysis method with semantic fuzziness to extract the customer's opinions expressed in sentiment Chinese phrases. The input reviews are taken from search forum website. Since the emotional phrases are not included in this work, calculation errors arise which results in lower accuracy.

#### A. Problem Statement

In the field of SA, machine learning methods such as Decision Tree [18], Logistic Regression [19], Support Vector Machine [20], Naive Bayes [21] and others have shown promising results, but most of them overly rely on hand-crafted features and necessitate a lot of manual design and adjustment, which is time-consuming and costly. Deep learning methods have achieved excellent performance with the help of large-scale corpus in many research fields, and become a research hotspot in SA [22]. However, in the majority of traditional text sentiment classification algorithms, the sentence-level sentiment classification remains difficult for various reasons, including a lack of semantic understanding and low classification accuracy. Moreover, the major existing methodology uses only  $3 \times 3$  and 2D kernels which cannot be able to extract the review-text features effectively. Most of the current classification method commonly preprocesses only the basic stop-words, numbers, & symbols, and maintains the words which does not reveal any sentiment information (Ex: around, surround, door etc.). Keeping those words in the classification phase increases the complexity and time consumption of the method. In some existing literatures, when the length of a word in an input review is less than the length of the sentence matrix, the padding operation is used to pad with sufficient amount of NULL data in order to meet the original length of the sentence matrix. This padded NULL data reduces the impact of real data and dominates over it which results in less effective feature representation and negatively impacts the classification accuracy. Substantially, many classification methods classifies the reviews into two (happy, unhappy) or three classes (positive, negative, neutral) which results in an ineffective decision making process.

The main contribution of this paper is to propose a novel e-commerce review classification method namely RAVO based Multi-Size Kernel structured Text CNN based Review Classification (MSK-TCNN-RAVO) based on a new variant of CNN. This method utilizes a convolutional operation that employs multi-size, multi-dimensional kernels to generate enriched features which yields high review classification accuracy. This research demonstrates the concept of varying kernels applied on 1D kernels in order to extract features exclusively from accurate data rather than padded data, which is one of the key processes of this work. This work introduces a new concept called review-stop-words removal, in which, the words that do not convey any sentiment information are eliminated. Keeping these review-stop-words in classification increase the complexity and the time-consumption of both training and testing process. To our knowledge, there are no studies in the available literatures that address the removal of non-sentimental words during the classification stage. Subsequently, it is configured to generate five classification outcomes which will help companies make better business decisions. Finally, to improve the performance of the

classifier, the proposed work is driven by a Rider driven African vulture optimization algorithm based on the overtaker's strategy of the Rider optimization algorithm.

### III. METHODOLOGY

The proposed review-classifier network is designed based on Convolutional Neural Network (CNN). The design of this new MSK-TCNN-RAVO method is one-of-a-kind, allowing for correct classification of review data. The following are the unique aspects of the proposed work:

- Input processing for  $7 \times 7$  size sentence matrix rather than a traditional  $5 \times 5$  size.
- Design of multi-size Convolutional kernels.
- Integration of both 1D and 2D kernels during convolution process.
- Introducing the concept of varying kernel based convolution process to extract features exclusively from accurate data rather than padded data.
- Pooling layer constructed by dual pooling operations
- Configured to generate five classification results viz. Bad, Fair, Neutral, Good, and Excellent.

The aforementioned modifications make this MSK-TCNN-RAVO method as a novel work and more effective than existing Text-based CNN classifiers to classify the review-data. The proposed work is mainly divided into three sections such as: MSK-TCNN-RAVO review training process, MSK-TCNN-RAVO review testing process, and Optimization using Modified African Vulture Algorithm.

The workflow of the proposed work is shown in Fig. 1. The input reviews from three domains such as Laptop, Camera, Mobile is taken from Amazon website. The dataset as a whole has a lot of long sentences, with an average of 11.44 sentences per review. The proposed method has both training and testing sections. The basic stopwords, numbers, and symbols are removed from the input reviews in the preprocessing section and words which do not reveal any sentiment information is removed in review-stop-words removal section. The sentence matrix (SM) is generated by extracting each word from the enhanced reviews and placing it in matrix form. The above mentioned process is common for both training and testing section. Each review from the SM is fed as input to the proposed algorithm's training process and trained network is created. Similarly, in the testing section, each review from SM is fed as input to the proposed algorithm's testing process and it continues until all the reviews are processed. Finally, the review classification report is produced from the testing section.

#### A. MSK-TCNN-RAVO Review Training Process

1) *Pre-processing*: Reviews are usually composed of incomplete expression, a variety of noise and poorly structured sentences. Noise and unstructured Twitter data will affect the performance of tweet sentiment classification [23], [24]. Prior to feature selection, a series of preprocessing steps

are performed on reviews to reduce the meaningless data in the review sentences.

Stop-words are words which have no value (positive or negative) in a sentiment analysis system that are meaningless in information retrieval, hence, they must be eliminated from the data set. For example stop words include "the", "as", "of", "and", "or", "to", etc. Stop word removing is substantial in the preprocessing, it has some advantages like reducing the size of stored data set and it improves the overall efficiency and effectiveness of the analysis system [25]. The proposed system uses a list of stop words obtained from Onix Text Retrieval Toolkit website [26].

In this proposed research work, three crucial pre-processing steps are carried out using text processing functions of MATLAB research tool to remove the (i) basic stop-words (ii) numbers, and (iii) symbols. The resultant pre-processed review is stored in the review-array  $R_{pp}$ .

2) *Review-stop-words removal*: The input of this section is the pre-processed review data. In the actual pre-processing section, the basic stop-words, numbers, and symbols are removed. But the remaining enhanced review contains the word which does not reveal any sentiment information, and those words are called as "Review-Stop-words". Keeping those words will increase time consumption and the complexity of the classification process. A tool is developed to assist for the generation of review-stop-words. Any sample review dataset can be fed as input to this generalized tool which comprises two steps, viz. basic stop-words removal and unique words extraction. Afterwards, the review-stop-words are extracted manually and the review-stop-word list can be generated.

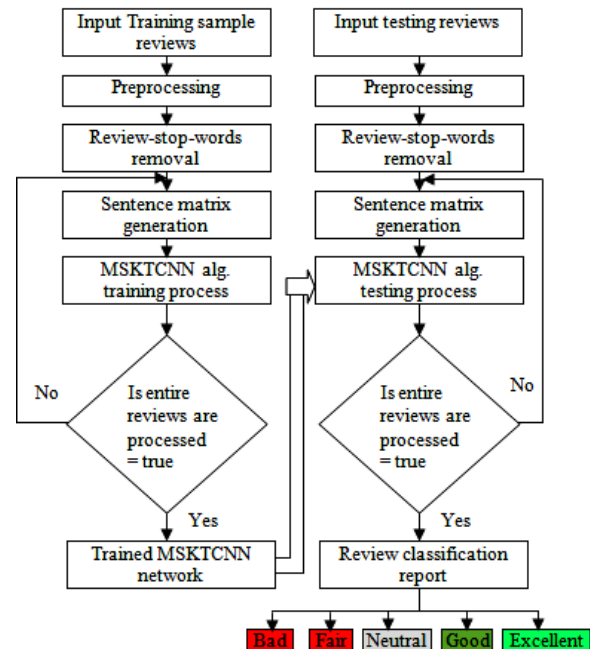


Fig. 1. Workflow of the Proposed MSK-TCNN-RAVO Classification Process.

SI No	Stop Words
1	today
2	seems
3	kind
4	something
5	after
6	around
7	across

Fig. 2. Sample List of Review-Stop-Words.

The sample review-stop-words are shown in Fig. 2. The survey-stop-words are loaded in the survey-stop-words-array  $L_{SSW}$ . In the pre-processed review, the survey-stop-words are removed based on the  $L_{SSW}$  list. The resultant survey after this removal process is stored in the review-array  $R_{SSW}$ .

3) *Sentence matrix generation*: The Sentence Matrix  $SM$  represents a sentence in the matrix form. The dimension of  $SM$  is fixed as  $7 \times 7$ . It is fixed by maintaining a trade-off between complexity and accuracy. A sentence of the enhanced review (after removing basic stop-words, numbers, symbols, and review-stop-words)  $R_{SSW}$  is extracted and each word of it is placed in the matrix form.

In this research work, the convolution process is carried out by taking  $SM$  as input using the combination of both 2D kernels such as  $3 \times 7$ ,  $5 \times 7$  and 1D kernel such as  $1 \times 3$ ,  $1 \times 5$ , and  $1 \times 7$  in order to extract the review text feature representation effectively.

4) *Training of MSK-TCNN-RAVO Algorithm*: This algorithm is comprised of multi-size, multi-dimensional kernels. The varying kernel adopted in this technique retrieves features only from the exact data and not from the padded data. The dual pooling operations applied in this algorithm increase the efficiency of the feature map processing. The training process is carried out in order to produce five classification results. The architecture of the algorithm is depicted in Fig. 3.

a)  *$3 \times 7$  Size Kernel based Convolution*: This convolution process is performed using six  $3 \times 7$  size kernels such as  $k_0, k_1, k_2, k_3, k_4$ , and  $k_5$  which is represented in Fig. 4. The column width of these kernels is fixed as seven, because the length of Sentence Matrix  $SM$  is seven. First the kernel  $k_0$  is used to convolute the sentence matrix  $SM$  to produce convoluted matrix  $CM_{k_0}$ . This process is performed using (1) to (5).

$$k_0 = \begin{bmatrix} 1 & 1 & 1 & 0 & 1 & 1 & 1 \\ -1 & 1 & 1 & 0 & 1 & 1 & -1 \\ -1 & -1 & -1 & 0 & -1 & -1 & -1 \end{bmatrix} \quad (1)$$

$$A = \sum_{m=0}^{3-1} \sum_{n=0}^{7-1} SM(i+m, j+n) * k_0(m, n), i=0, j=0 \quad (2)$$

$$B = \sum_{m=0}^{3-1} \sum_{n=0}^{7-1} SM(i+m, j+n) * k_0(m, n), i=2, j=0 \quad (3)$$

$$C = \sum_{m=0}^{3-1} \sum_{n=0}^{7-1} SM(i+m, j+n) * k_0(m, n), i=4, j=0 \quad (4)$$

$$CM_{k_0} = \begin{bmatrix} A \\ B \\ C \end{bmatrix} \quad (5)$$

Herein,  $A, B, C$  represents the convoluted result by applying convolution on row of 0, 2, and 4 respectively in  $SM$ , and  $CM_{k_0}$  indicates convoluted matrix by  $k_0$  kernel.

Equation (1) defines the kernel  $k_0$ . Equation (2) computes the convoluted value by projecting the  $k_0$  over the element of  $SM(0,0)$ . It means the convolution process performed on 0<sup>th</sup> row and 0<sup>th</sup> column in  $SM$ , and stored in A. Likewise the Equation (3) computes the convoluted result by considering the position  $SM(2,0)$  and the output is assigned in B. The Equation (4) computes the convoluted result by considering the position  $SM(4,0)$ . Finally, the convoluted matrix  $CM_{k_0}$  with size  $3 \times 1$  is constructed by placing the A, B, and C values in the order using (5). Similarly the other  $3 \times 7$  size kernels such as  $k_1, k_2, k_3, k_4$ , and  $k_5$  is convoluted to find the convoluted matrix  $CM_{k_1}, CM_{k_2}, CM_{k_3}, CM_{k_4}$ , and  $CM_{k_5}$  respectively.

b)  *$5 \times 7$  Size Kernel based Convolution*: Here, the convolution process is performed using four  $5 \times 7$  size kernels such as  $k_6, k_7, k_8$ , and  $k_9$  to generate four convoluted matrices such as  $CM_{k_6}, CM_{k_7}, CM_{k_8}$ , and  $CM_{k_9}$ . The four  $5 \times 7$  size kernels are depicted in Fig. 5.

The kernel  $k_6$  is used to compute the convoluted matrix  $CM_{k_6}$  with size  $6 \times 1$  using (6) and (7).

$$k_6 = \begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & -1 & -1 & -1 & -1 & -1 & 1 \\ 1 & -1 & 0 & 0 & 0 & -1 & 1 \\ 1 & -1 & -1 & -1 & -1 & -1 & 1 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 \end{bmatrix} \quad (6)$$

$$CM_{k_6} = \begin{bmatrix} \sum_{m=0}^{5-1} \sum_{n=0}^{7-1} SM(i+m, j+n) * k_6(m, n), i=0, j=0 \\ \sum_{m=0}^{5-1} \sum_{n=0}^{7-1} SM(i+m, j+n) * k_6(m, n), i=1, j=0 \\ \sum_{m=0}^{5-1} \sum_{n=0}^{7-1} SM(i+m, j+n) * k_6(m, n), i=2, j=0 \\ \sum_{m=0}^{5-1} \sum_{n=0}^{7-1} SM(i+m, j+n) * k_6(m, n), i=3, j=0 \\ \sum_{m=0}^{5-1} \sum_{n=0}^{7-1} SM(i+m, j+n) * k_6(m, n), i=4, j=0 \\ \sum_{m=0}^{5-1} \sum_{n=0}^{7-1} SM(i+m, j+n) * k_6(m, n), i=5, j=0 \end{bmatrix} \quad (7)$$

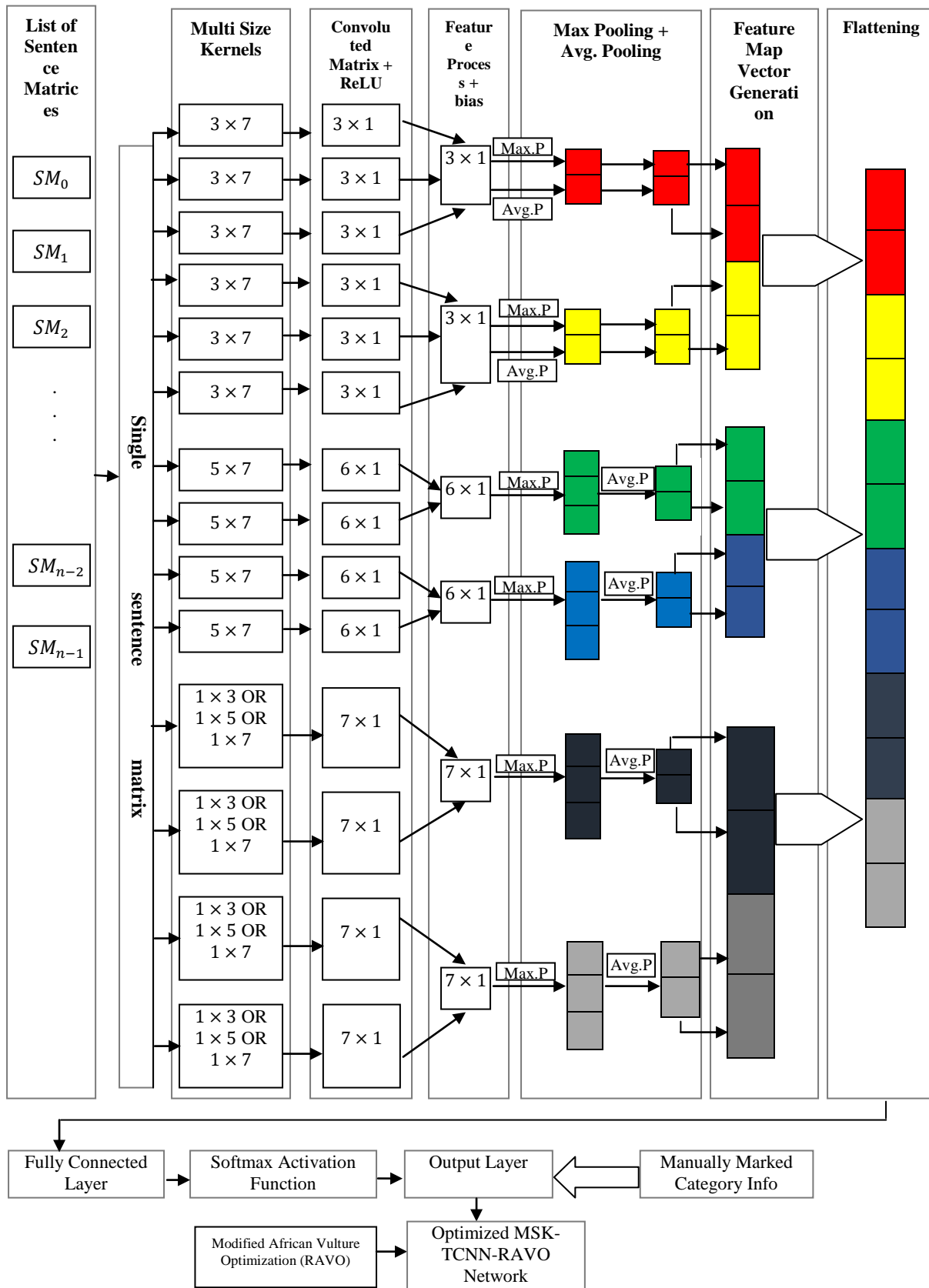


Fig. 3. Architecture Diagram of the Training Section of the Proposed MSK-TCNN-RAVO Method.

1 1 1 0 1 1 1	0 0 0 -1 -1 -1 -1
-1 1 1 0 1 1 -1	-1 -1 -1 1 1 1 0
-1 -1 -1 0 -1 -1 -1	1 1 1 -1 0 0 0
$k_0$	$k_3$
-1 -1 -1 -1 0 0 0	-1 1 -1 0 -1 1 -1
1 -1 0 1 0 -1 1	0 0 0 1 1 0 1
1 1 1 0 0 0 0	1 -1 -1 1 -1 -1 0
$k_1$	$k_4$
0 0 0 -1 1 1 1	0 1 1 0 -1 -1 0
-1 1 -1 0 -1 1 -1	-1 0 1 -1 1 0 1
1 1 1 0 0 0 0	0 -1 -1 1 1 1 -1
$k_2$	$k_5$

Fig. 4. Representation of Six Kernel of  $3 \times 7$  Size.

1 1 1 1 1 1 1	1 -1 1 -1 1 -1 1
1 -1 -1 -1 -1 -1 1	1 -1 1 -1 1 -1 1
1 -1 0 0 0 -1 1	1 -1 -1 0 -1 -1 1
1 -1 -1 -1 -1 -1 1	1 -1 -1 -1 -1 -1 1
1 1 1 1 1 1 1	1 -1 -1 -1 -1 -1 1
$k_6$	$k_8$
1 1 1 1 1 1 1	-1 0 0 1 0 0 1
-1 -1 -1 -1 -1 -1 -1	0 -1 -1 1 1 1 0
1 1 1 0 -1 -1 -1	1 1 1 0 -1 -1 -1
-1 -1 -1 -1 -1 -1 -1	0 1 1 -1 -1 -1 0
1 1 1 1 1 1 1	1 0 0 -1 0 0 -1
$k_7$	$k_9$

Fig. 5. Representation of  $5 \times 7$  Size Kernels  $k_6, k_7, k_8,$  and  $k_9$ .

In Equation (7), the kernel  $k_6$  is projected over the rows of padded-SM from row = 0 to row = 5, and the convolution process is performed. Similarly the other convoluted matrix such as  $CM_{k_7}, CM_{k_8},$  and  $CM_{k_9}$  is convoluted.

c) One Dimensional Varying Kernel based Convolution:

In Sentence matrix  $SM$ , the length of a word may be less than the matrix length 7. In this case, padding operation pads the necessary quantity of NULL data to meet the length of the  $SM$ . However, the padded data diminishes the effect of real data in feature representation, lowering review classification accuracy. Hence, this research employs the concept of 1D varying kernels such as  $1 \times 3, 1 \times 5, 1 \times 7$  which have the length bounded in the range of  $1 \times 7$  as shown in Fig. 6. The words which have length less than 7 in  $SM$  undergone the convolution operation with the aid of either the varying kernel  $1 \times 3$  or  $1 \times 5$ . The words which have length equal to 7 in  $SM$  undergone the convolution process by using the kernel of  $1 \times 7$  size. This concept extracts the features from the exact data and not from the padded data, which is one of the key processes of this research.

1 1 1	1 1 1 1 1	1 1 1 1 1 1 1
$k_{10}^{1 \times 3}$	$k_{10}^{1 \times 5}$	$k_{10}^{1 \times 7}$
1 -1 1	1 1 -1 1 1	1 1 -1 -1 1 1 1
$k_{11}^{1 \times 3}$	$k_{11}^{1 \times 5}$	$k_{11}^{1 \times 7}$
1 1 -1	1 1 1 -1 -1	1 1 1 1 -1 -1 -1
$k_{12}^{1 \times 3}$	$k_{12}^{1 \times 5}$	$k_{12}^{1 \times 7}$
-1 1 1	-1 -1 1 1 1	-1 -1 1 1 1 1 1
$k_{13}^{1 \times 3}$	$k_{13}^{1 \times 5}$	$k_{13}^{1 \times 7}$

Fig. 6. Illustration of Varying Kernels.

The convoluted matrix computation for the kernel  $k_{10}$  is performed using (8) to (14).

$$k_{10}^{1 \times 3} = [1 \ 1 \ 1] \tag{8}$$

$$k_{10}^{1 \times 5} = [1 \ 1 \ 1 \ 1 \ 1] \tag{9}$$

$$k_{10}^{1 \times 7} = [1 \ 1 \ 1 \ 1 \ 1 \ 1 \ 1] \tag{10}$$

$$H_{SM} = \text{HeightOfSentenceMatrix}(SM) \tag{11}$$

$$L_i = \text{Word Length}(SM_i) \tag{12}$$

$$i \in [0, H_{SM} - 1] \tag{12}$$

$$C_i = \begin{cases} \sum_{n=0}^{3-1} SM(i, n) * k_{10}^{1 \times 3}(0, n), & \text{if } L_i \leq 3 \\ \sum_{n=0}^{5-1} SM(i, n) * k_{10}^{1 \times 5}(0, n), & \text{else if } L_i \leq 5 \\ \sum_{n=0}^{7-1} SM(i, n) * k_{10}^{1 \times 7}(0, n), & \text{else if } L_i \leq 7 \end{cases}$$

$$i \in [0, H_{SM} - 1] \tag{13}$$

$$CM_{k_{10}} = \begin{bmatrix} C_0 \\ C_1 \\ C_2 \\ C_3 \\ C_4 \\ C_5 \\ C_6 \end{bmatrix} \tag{14}$$

where

$H_{SM}$  - Height of the Sentence Matrix  $SM$

Height Of Sentence Matrix – Function to find the height of sentence matrix  $SM$

$L_i$  - Length of  $i^{\text{th}}$  word in  $SM$

Word Length – Function to compute length of the specific word in  $SM$

$C_i$  - Convoluted value of  $10^{\text{th}}$  varying kernel corresponding to  $i^{\text{th}}$  row of Sentence Matrix  $SM$

$n$  - Column movement index

$CM_{k_{10}}$  - Convoluted matrix related to  $k_{10}$  kernel.

Equation (8) to Equation (10) defines the 10<sup>th</sup> varying kernel's variation such as  $k_{10}^{1 \times 3}$ ,  $k_{10}^{1 \times 5}$  and  $k_{10}^{1 \times 7}$ . Equation (11) computes the height of the Sentence Matrix  $SM$  and stores in the term  $H_{SM}$ . Equation (12) computes the length of each word of  $SM$ . Equation (13) computes the convoluted value of each word of  $SM$  using 10<sup>th</sup> varying kernel, by using the kernels  $k_{10}^{1 \times 3}$  or  $k_{10}^{1 \times 5}$  or  $k_{10}^{1 \times 7}$  depending on the length of word  $L_i$ . The Equation (14) shows the convoluted matrix by merging the seven values of  $C_i$ . Similarly, the convoluted matrix such as  $CM_{k11}$ ,  $CM_{k12}$ , and  $CM_{k13}$  will be computed.

d) *ReLU Operation and Feature Matrix Generation:* In this research, the ReLU function is used as activation function after convolution process. The function returns 0 if it receives any negative input, but for any positive value  $x$ , it returns the same value [27]. The ReLU activation function is designed for the convolution matrix  $CM_{k10}$  is based on Equation (15).

$$CM_{k0}^i = \begin{cases} CM_{k0}^i, & \text{if } CM_{k0}^i \geq 0 \\ 0, & \text{else} \end{cases} \quad (15)$$

where  $i \in [0, n - 1]$

$n$  – Height of the convoluted matrix  $CM_{k0}$  (Let it be 3 for  $3 \times 7$  size kernels). Similarly the other convolution matrixes ranging  $CM_{k1}$  to  $CM_{k13}$  are computed based on (15).

The feature matrix is generated in order to reduce the quantity of convoluted matrix from 14 to 6. This helps to improve the speed of the review-analysis. Herein, the bias value participates its part similar with the traditional neural network. The bias value  $b1$  is set as 1 and bias value  $b2$  is set as 0. The illustration of feature matrix generation on the Sentence Matrix is shown in Fig. 7.

e) *Pooling Operations:* The pooling operation reduces the size of the feature map by keeping the key information. This research applies two types of pooling operations such as maximum pooling and Average pooling which essentially extracts the essence of the feature map. The feature matrix  $FM_0, FM_1$  with  $3 \times 1$  dimension is undergone the Max pooling and Avg. pooling operation which results a individual single value by computing the maximum value and average value respectively. The feature matrix  $FM_2, FM_3$  with  $6 \times 1$  dimension is undergone both the pooling operation based on Fig. 8.

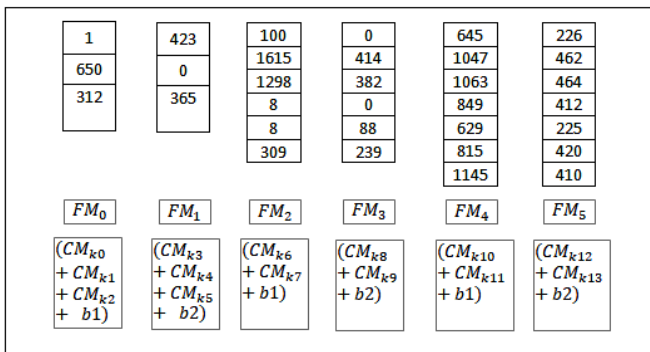


Fig. 7. Illustration of Feature Matrix Generation on the Example Sentence Matrix.

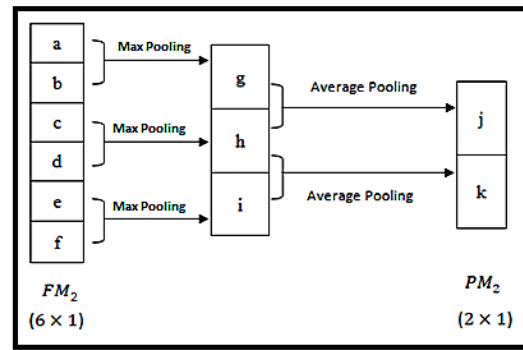


Fig. 8. Pooling Operation Performed in the Feature Map  $FM_2$ .

Similarly the feature matrix  $FM_4, FM_5$  is undergone both the pooling operations as shown in Fig. 9. The illustration of pooling layer computation is depicted in Fig. 10. The  $2 \times 1$  dimension is the resultant matrix after the pooling operations.

f) *Feature Map Vector Generation:* The feature map vector generation is performed to collect the same features of same size kernels into small groups. The size reduced  $3 \times 7$  kernels are grouped in order to generate feature map vector  $FV_0$  using (16).

$$FV_0 = \{ PM_0^0, PM_0^1, PM_1^0, PM_1^1 \} \quad (16)$$

Similarly, the size reduced  $5 \times 7$ , and one dimensional kernel oriented pooled feature maps are grouped to generate the feature map vector  $FV_1, FV_2$  respectively using (17) and (18).

$$FV_1 = \{ PM_2^0, PM_2^1, PM_3^0, PM_3^1 \} \quad (17)$$

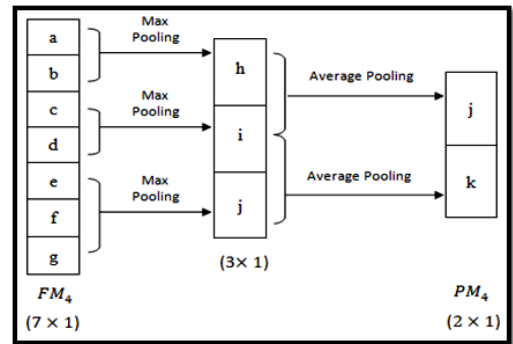


Fig. 9. Pooling Operation Performed in the Feature Map  $FM_4$ .

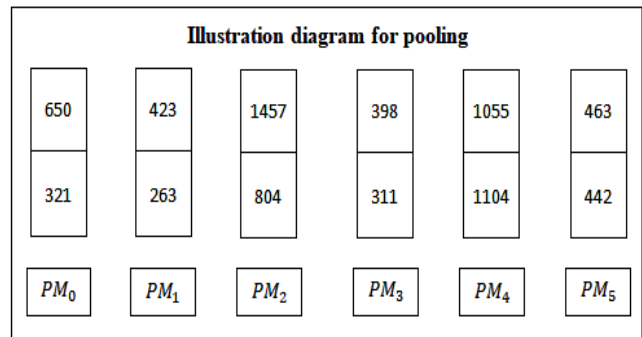


Fig. 10. Illustration of Pooling Layer Computation.

Likewise, the feature map vector  $FV_2$  is originated using the one dimensional kernels oriented pooled feature maps such as  $PM_4$  and  $PM_5$  based on (18).

$$FV_2 = \{PM_4^0, PM_4^1, PM_5^0, PM_5^1\} \quad (18)$$

g) *Softmax Function*: The resultant feature map vectors  $FV_0$ ,  $FV_1$ , and  $FV_2$  holds four elements each. The output of the convolution layer is flattened into a 1D array called as F, which is given as input to the fully connected layer. The output layer in an artificial neural network produces the given outputs for the program. In the training section of the proposed RAVO-MSKTCNN algorithm, the manually marked category information such as BD, FR, NL, GD, EX with target values 1, 2, 3, 4 and 5 respectively are fed to support the learning process. The extracted flattened features of the given training review sample is converted into probabilistic distributions form by softmax function. This data is compiled by the target values of training categories until it converges by the back propagation concept. Thus the training of a review data is progressed. Multi reviews are trained using the MSK-TCNN-RAVO algorithm, and the trained network of the same is generated.

#### B. MSK-TCNN-RAVO Review Testing Process

The testing process of proposed algorithm is used to classify the review data with the aid of the trained network. The Fig. 11 depicts the architecture of the testing process of the proposed MSK-TCNN-RAVO network. The basic stop words, numbers, and symbols are removed from the test reviews in the preprocessing section and words which do not reveal any sentiment information are removed in review-stop-words removal section.

The enhanced reviews  $SM$  is fed as input to the testing section of the MSK-TCNN-RAVO algorithm and the convolution operations such as  $3 \times 7$ ,  $5 \times 7$ , and one dimensional varying kernel based convolution are performed on  $SM$  to generate the corresponding convoluted matrices. The ReLU function is used as an activation function after convolution process, and feature matrix is generated in order to reduce the quantity of convolution matrices from 14 to 6 as shown in Fig. 7. The dual pooling operation such as Max and Average pooling is applied and finally the feature map vector is generated. The extracted feature map is given to the MSK-TCNN-RAVO network as a feature which is then classified into five classes as shown in Fig. 11. This process is repeated until all the test reviews are processed. The classification report generated by the proposed algorithm can be used for better decision making process.

---

#### Algorithm -1 of review testing process

---

- Step 1:** The test review is given as input  
**Step 2:** Remove basic stop words from the given test review  
**Step 3:** Remove the survey-stop-words from the pre-processed test reviews  
**Step 4:** Convert the processed test review into a Sentence Matrix (SM) of size  $7 \times 7$   
**Step 5:** Design six  $3 \times 7$  size kernels such as  $k_0, k_1, k_2, k_3, k_4$ , and  $k_5$ .  
**Step 6:** Convolute the Sentence Matrix (SM) by  $k_0$  kernel and find the Convolution Matrix  $CM_{k_0}$   
**Step 7:** Repeat **Step 6** with the other kernels such as  $k_1, k_2, k_3, k_4$ , and  $k_5$  to obtain the Convolution Matrix  $CM_{k_1}, CM_{k_2}, CM_{k_3}, CM_{k_4}$ , and  $CM_{k_5}$  respectively.  
**Step 8:** Design four  $5 \times 7$  size kernels such as  $k_6, k_7, k_8$ , and  $k_9$ .  
**Step 9:** Convolute the Sentence Matrix (SM) by  $k_6$  kernel and find the Convolution Matrix  $CM_{k_6}$ .  
**Step 10:** Repeat **Step 9** with the other kernels such as  $k_7, k_8$ , and  $k_9$  to obtain the Convolution Matrix  $CM_{k_7}, CM_{k_8}$ , and  $CM_{k_9}$  respectively.  
**Step 11:** Construct four one dimensional varying kernels of size  $1 \times 3$ ,  $1 \times 5$ ,  $1 \times 7$ . Name the four kernels as  $k_{10}, k_{11}, k_{12}$ , and  $k_{13}$ .  
**Step 12:** Compute Convolution matrix  $CM_{k_{10}}$  via one dimensional varying kernels  $k_{10}^{1 \times 3}, k_{10}^{1 \times 5}$ , and  $k_{10}^{1 \times 7}$ .  
**Step 13:** Compute convoluted matrix  $CM_{k_{11}}$  using kernels  $k_{11}^{1 \times 3}, k_{11}^{1 \times 5}$ , and  $k_{11}^{1 \times 7}$   
**Step 14:** Compute Convolutional matrix  $CM_{k_{12}}$  using kernels like  $k_{12}^{1 \times 3}, k_{12}^{1 \times 5}$  and  $k_{12}^{1 \times 7}$   
**Step 15:** Compute convoluted matrix  $CM_{k_{13}}$  using kernels such as  $k_{13}^{1 \times 3}, k_{13}^{1 \times 5}$ , and  $k_{13}^{1 \times 7}$ .  
**Step 16:** Activate the convoluted matrices such as  $CM_{k_0}$  to  $CM_{k_{13}}$  using the ReLU activation function.  
**Step 17:** Construct the feature matrix  $FM_0$  to  $FM_5$  using the convoluted matrices  $CM_{k_0}$  to  $CM_{k_{13}}$ .  
**Step 18:** Apply the max pooling and average pooling operations on the feature matrix  $FM_0$  to  $FM_5$  and compute the corresponding pooled matrix  $PM_0$  to  $PM_5$  of  $2 \times 1$  size.  
**Step 19:** Construct  $3 \times 7$  kernel based feature map vector  $FV_0$  to  $FV_2$ .  
**Step 20:** Feed the output of fully connected layer to Softmax activation function.  
**Step 21:** Output classification results for test reviews based on RAVO driven MK-TCNN network.
-



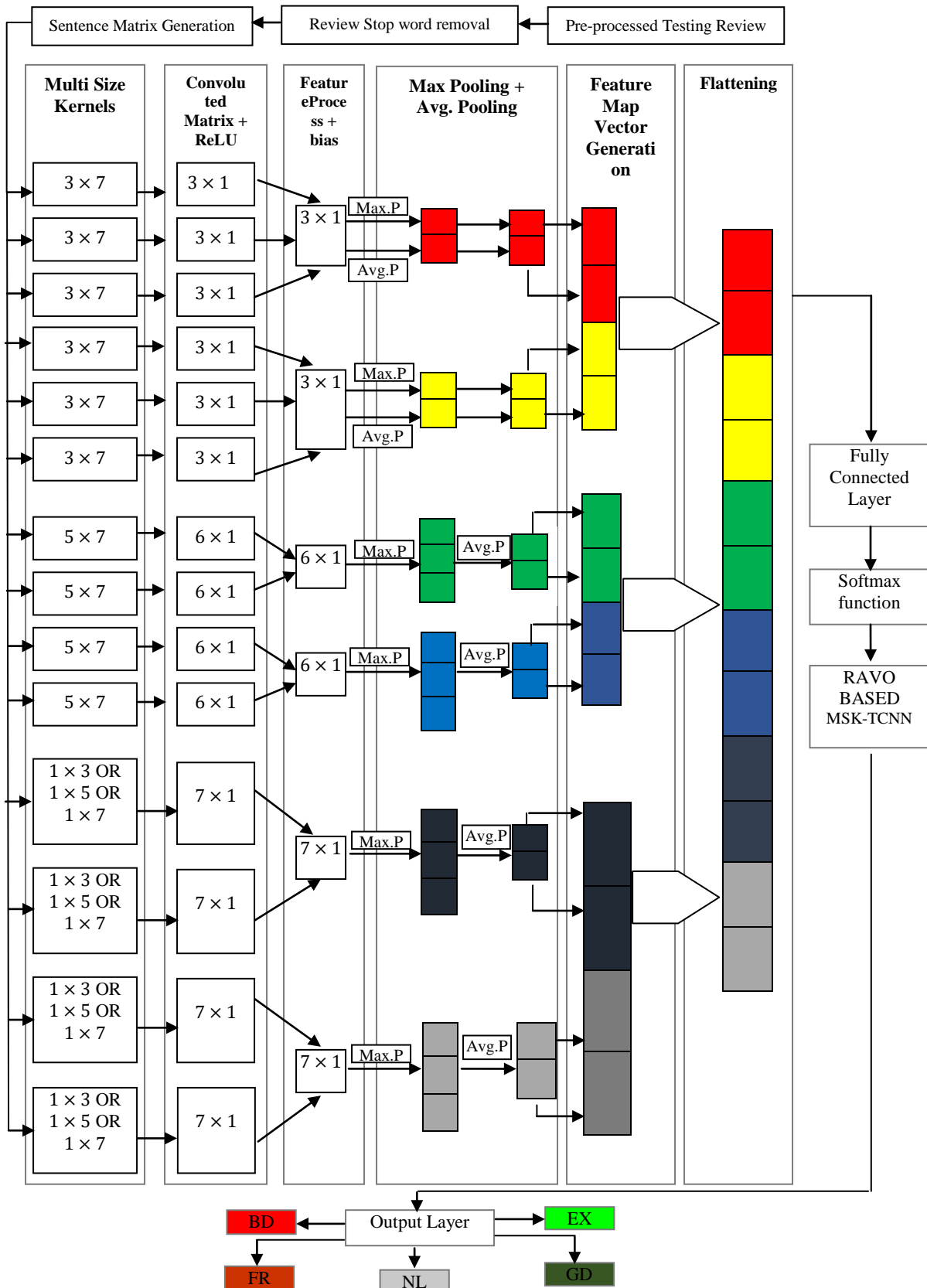


Fig. 11. Architecture Diagram of the Testing Section of the Proposed MSK-TCNN-RAVO Method.

### C. RAVO Driven MSK-TCNN

The proposed MSK-TCNN-RAVO network is driven by Rider driven African vulture optimization algorithm based on over taker's strategy of rider optimization algorithm. The parameters of the MSK-TCNN-RAVO is tuned by considering the significant parameters of RAVO such as multi kernel size ( $M_{ks}$ ), Pooling type ( $P_o$ ), and Feature map Vector ( $F_{mv}$ ). These parameters are analyzed using (19).

$$T_p = \{M_{ks=1 \text{ to } t} || F_{mv=1 \text{ to } t} || P_{o=1 \text{ to } t}\} \quad (19)$$

Where,  $T_p$  denotes the hidden layers to be optimized in which  $t$  represents the initialization parameters of RAVO from MSK-TCNN-RAVO. In addition to the hidden layers, number of epoch, learning rate, batch size are considered as the hyper parameters for obtaining maximum efficiency in terms of performance matrices such as accuracy, precision, recall, F-Score, Time taken, MSE, Misclassification ratio, ranking index, etc. Moreover, these parameters are taken from MSK-TCNN to evaluate the performance based on Mean Square Error (MSE) and Logarithmic Loss. The MSE is considered as a fitness value of MSK-TCNN-RAVO that is processed until it reaches the minimal MSE. Equation (20) is considered as fitness evaluation.

$$F_{(T_p)} = \min(MSE, lss) \quad (20)$$

After the initialization of the fitness value the following equation is utilized for determining the best solution.

$$R_i = \begin{cases} Best_{sol1} & \text{if } p_i = L_1 \\ Best_{sol2} & \text{if } p_i = L_2 \end{cases} \quad (21)$$

The selection of best solution is obtained by Roulette wheel selection strategy as shown in Equation (22).

$$p_i = \frac{F_{(T_p)}}{\sum_{iter=1}^n F_{(T_p)}} \quad (22)$$

Where,  $R_i$  and  $F_{(T_p)}$  are, respectively, the probability solution and the fitness solutions of each individual as  $i$ . This process is repeated until reaches the  $n$  number of iterations. This strategy enhances the diversity in AVOA. Then the second strategy of AVOA is rate of starvation which is determined using (23) and (24).

$$s = t \times \left( \sin^w \left( \frac{\pi}{2} \times \frac{iter}{maxiters} \right) + \cos \left( \frac{iter}{maxiter} \right) - 1 \right) \quad (23)$$

$$V_s = (2 \times rand_1 + 1) \times z \times \left( 1 - \frac{iter}{maxiter} \right) + s \quad (24)$$

Where,  $V_s$  denotes the satiated vultures,  $i$  denotes the iteration,  $maxiter$  denotes the maximum iteration,  $z$  denotes the random number in range (-1 and 1),  $t$  represents the random value in range (0 and 1). Here, the starvation of vulture is determined by  $z < 0$  and  $z > 0$  conditions, that show vulture is starved and not starved, respectively. According to this RAVO is executed exploration strategy.

$$P(i+1) = \begin{cases} R(i) - D(i) \times V_s & \text{if } P_1 \geq rand_{p_1} \\ X \times R(i) - P(i) & \text{if } P_1 < rand_{p_2} \end{cases} \quad (25)$$

Where,  $P(i+1)$  is new position vector,  $V_s$  denotes the number of vulture gratified using (24) in the current iteration. Moreover, the variables in the following denotes

$X \rightarrow$  Coefficient vector to maximize the random motion that is measured using  $X = 2 \times rand$  in which  $rand$  is random number in range of 0 and 1.

$P(i) \rightarrow$  The position vectors of the current iteration.

$$P(i+1) = R(i) - V_s + rand_2 \times ((up_b - lw_b) \times rand_3 + lw_b) \quad (26)$$

In aforementioned Equation (26),  $rand_2$  is the random value in range of 0 and 1.  $lw_b$ , and  $up_b$  are the variable that lower bound and upper bound, respectively. The Equation (26) is utilized to generate random solution based on the range of  $lw_b$ , and  $up_b$ . The “ $rand_3$ ”, increases the coefficient of random. Here,  $rand_3$  reaches nearly 1, it disseminates the solution with similar order that increases random motion based  $lw_b$ . Therefore, the diversity can be increased and explored maximum search space solution. Then, the model is moving on exploitation strategy if the  $|F_{(T_p)}|$  has obtained below 1.

### D. Exploitation

The exploitation has incorporated with two strategies where  $P_2$  and  $P_3$  are the parameters to determine the strategy to be executed. The ranges of these parameters are initialized at 0 and 1 before executing the searching process.

(i) Phase 1

In the first phase, the exploitation is obtained only when the value of  $|F_{(T_p)}|$  reaches between the range of 1 and 0.5. Here, determining the  $P_2$  value is highly significant before the execution of searching process with the range between 0 and 1. Here, the “ $rand_{p_2}$ ”, generates random number between 0 and 1. If  $rand_{p_2} \geq P_2$ , the Siege-fight strategy is executed. Conversely, if  $rand_{p_2} < P_2$ , rotation flight is processed. The mathematical formula of this is shown in (27).

$$P(i+1) = \begin{cases} D(i) \times (F_{(T_p)} + rand_4) - d(t) & \text{if } P_2 \geq rand_{p_2} \\ d(t) = R(i) - P(i) & \text{if } P_2 \geq rand_{p_2} \end{cases} \quad (27)$$

1) *Competition for food*: If the condition has obtained  $|F_{(T_p)}| \geq 0.5$ , means that the vultures in the environment are highly satiated and they have considerable energy for surviving. This can be determined by Equation (28).

$$D(i) \times (V_s + rand) - d(t) \\ d(t) = R(i) - P(i) \quad (28)$$

Where,  $D(i)$  is evaluated using (27),  $V_s$  denotes the rate of vulture satiated,  $rand_4$  denotes the random number (0 and 1) helps for generating random coefficient.

Then the vulture nest strategy is rotating flight that is based on spiraling between whole vulture and one of the two

best vultures. The mathematical formulation of these strategy is shown in (29) and (30).

$$S_1 = R(i) \times \left( \frac{rand_5 \times p(i)}{2\pi} \right) \times \cos(p(i)) \quad (29)$$

$$S_2 = R(i) \times \left( \frac{rand_6 \times p(i)}{2\pi} \right) \times \sin(p(i)) \quad (30)$$

$$P(i + 1) = R(i) - (S_1 + S_2) \quad (31)$$

Where,

$R(i)$  is executed based on same strategy as earlier mentioned. Cos and Sin are utilized in this work based on sine and cosine function.  $rand_5$  and  $rand_6$  are random number generated in the range of 0 and 1.  $S_1$ , and  $S_2$  are generated using (29) and (30). Equation (31) to update the newly generated solution.

(ii) Phase 2:

In the second phase of exploitation, the two vultures' movements accumulate several types of vultures over the food source, and the siege and aggressive strive to find food are carried out. If  $|F_{(T_p)}| \leq 0.5$ , this phase is relying on  $rand_{p3}$  in the range (0 and 1). If  $rand_{p3} \geq p_3$ , various vultures are accumulating over the food. Otherwise, if  $rand_{p3} < p_3$ , the aggressive siege flight strategy is executed. The mathematical formulation of this is shown in equation (32).

$$P(i + 1) = \begin{cases} equ (31) & \text{if } rand_{p3} \geq P_3 \\ equ (32) & \text{if } rand_{p3} < P_3 \end{cases}$$

$$A_1 = Best_{vult_1}(i) - \frac{Best_{vult_1}(i) \times p(i)}{Best_{vult_1}(i) \times p(i)^2} \times F$$

$$A_2 = Best_{vult_2}(i) - \frac{Best_{vult_2}(i) \times p(i)}{Best_{vult_2}(i) \times p(i)^2} \times F$$

$$(i + 1) = \frac{A_1 + A_2}{2} \quad (32)$$

a) Aggressive Competition for Food

If  $|F_{(T_p)}| < 0.5$  occurs, repositioning the vultures using equation (33)

$$P(i + 1) = R(i) - |d(t)| \times over_t(d) \quad (33)$$

Here, the updating strategy of Rider Optimization is executed especially based on over taker's strategy. This strategy helps to enhance the performance of AVOA. Therefore, Rider based enhanced AVOA can make highly effective performance.

$$over_{t+1}^0(\text{iter}, k) = over_t(i, k) + [d^Z(t)] * X^{hv}(hv, k)$$

Where,  $over_t(i, k)$ , represents the  $i_{th}$  vulture in the  $k_{th}$  coordinates,  $d^Z(t)$ , represents the direction of  $i_{th}$  rider at the time  $t$ , and in the range (-1 and 1)  $X^{hv}(hv, k)$ , represents the head vulture in the  $k_{th}$  coordinates. Moreover, the computation of  $d^Z(t)$  is achieved by using (34) to (36).

$$d^Z(t) = \left[ \frac{2}{1 - \log(S_t^R(\text{iter}))} \right] - 1 \quad (34)$$

$$S_t^R(i) = \frac{r_t(i)}{\max_{i=2}^R r_t(i)} \quad (35)$$

$$r_t(i) = \frac{1}{\|X_i - L_T\|} \quad (36)$$

$S_t^R(i)$  represents the relative success rate of the  $i_{th}$  vulture at the time  $t$  and it will be in the interval of 0 and 1. Finally, determine the coordinate selection based on normalized distance vector that obtained by gauging the difference of  $i_{th}$  vulture and head vulture using equation (37).

$$l(i, j) = |X_t(i, j) - X^h(h, j)| \quad (37)$$

The value of  $j$  is selected for  $l(i, j)$  values are less than its fitness value. Therefore, the Rider's over taking strategy improves the performance by replacing Levy fight strategy. The RAVO can provide high performance than traditional African Vulture Optimization algorithm.

---

### Pseudo Code: RAVO driven MSKTCNN

---

Start:

Define the parameters of the RAVO-MSKTCNN

I/P

Five sets of Review datasets are divided into 70% for Training and 30% for Testing

O/P

Optimized RAVO-MSKTCNN  
Minimized MSE, & Cross Entropy and Maximized accuracy, precision, recall, Fscore

Initialize the parameters of RAVO-MSKTCNN

RAVO Parameter

Number of population, Number of iteration,

MSKTCNN Parameters:

Batch Size, Number of Epoch, Learning Rate, Multi-Size Kernel, Pooling Size

Determine the MSKTCNN parameter using equation (19)

$$T_p = \{M_{ks=1 \text{ to } t} \parallel F_{mv=1 \text{ to } t} \parallel P_{o=1 \text{ to } t}\}$$

While ((MSE & Loss)  $\leq$  termination Criteria ) do

Best solutions are obtained using  $R_i$

For (each solution ( $p_i$ )) do

Equation (21) to obtain  $R_i$

Update the solution using (23) to (35)

Rider's Overstake's strategy to improve AVOA using (34) to (37)

Return

Optimized Parameters of RAVO based MSKTCNN obtained

End

---

## IV. ANALYSIS AND DISCUSSION

This section is very important in any research paper which portrays the efficiency of the proposed or the other existing methods. The proposed MSK-TCNN-RAVO classification task is compared with three existing methods mentioned below:

1) Weakly supervised deep embedding for review Sentence sentiment classification (WS\_RSSC) [28].

- 2) Sentiment similarity analysis based mining of user’s trust from e-commerce reviews (SSA\_TER) [29].
- 3) CNN based multiple convolution and pooling for text sentiment classification (CNN\_MCP\_TSC) [30].

In order to test the proposed MSK-TCNN-RAVO classifier, the training and testing reviews are taken from five benchmark databases. The product reviews from different categories such as Mobile, Laptop, and Camera are taken from Amazon website. The names of the different datasets are given below:

- 1) Dataset of Laptop Product Review (DS1\_LPR) [28].
- 2) Dataset of Camera Product Review (DS2\_CPR) [28].
- 3) Dataset of Mobile Product Review for NOKIA 3310 (DS3\_MPR3310) [28].
- 4) Dataset of Mobile Product Review for SAMSUNG EPIC 4G (DS4\_MPRSE4G) [31].
- 5) Dataset of Mobile Product Review for NOKIA 105 (DS5\_MPR105) [32].

From each dataset, 7000 reviews are extracted, out of that, 4000 reviews are utilized for training and the remaining 3000 reviews which are completely new to the MSK-TCNN-RAVO classifier is used for testing purpose to ensure the classifier’s efficiency. Following that, 2000 more reviews are extracted from the training samples to act as test reviews, totaling 5000 test reviews for each dataset.

**A. Classification Accuracy**

The Accuracy of the classifier is computed using (38) and its values are represented in Table I.

$$Accuracy = \frac{TP+TN}{(TP+FP+FN+TN)} \tag{38}$$

A true positive (TP) is an outcome where the model correctly predicts the positive class. Similarly, a true negative (TN) is an outcome where the model correctly predicts the negative class. A false positive (FP) is an outcome where the model incorrectly predicts the positive class. And a false negative (FN) is an outcome where the model incorrectly predicts the negative class.

TABLE I. CLASSIFICATION ACCURACY ANALYSIS FOR FIVE DATABASES AND FOUR METHODS

Database Name	Classification Accuracy (%)				
	WS_RSSC Method	SSA_TER Method	CNN_MCP_TSC Method	Proposed MSK-TCNN-RAVO Method (Before RAVO)	Proposed MSK-TCNN-RAVO Method (After RAVO)
DS1_LPR	78.2	79	80.5	<b>86.7</b>	<b>90.6%</b>
DS2_CPR	77.6	77.3	78.9	<b>84</b>	<b>89.6%</b>
DS3_MPR3310	80.1	81.5	82.4	<b>89.6</b>	<b>93.2%</b>
DS4_MPRSE4G	81.8	82.6	84	<b>90.5</b>	<b>95.5%</b>
DS5_MPR105	79.3	80.4	81.4	<b>88.9</b>	<b>92.4%</b>

The accuracy of the proposed model is high in all datasets when compared to other models. The highest accuracy is produced for the DS4\_MPRSE4G dataset which is 95.5% after performing optimization via RAVO. The result shows that the proportion of classified true results produced by the proposed work is high when compared to other models.

**B. Precision**

Precision measures the exactness of a classifier. It can be calculated using (39).

$$Precision = \frac{TP}{TP+FP} \tag{39}$$

The average precision values are represented in Table II. The precision is computed for each method for all the five datasets. Average precision of a specific method is computed by averaging the precision value of five datasets. The precision value for a specific dataset is computed by testing 5000 test review samples. The proposed methodology has a higher precision value, indicating that the proposed model produces less false positives, implying that the proposed model’s prediction is more reliable. The average value of the proposed method is 0.89

TABLE II. AVERAGE PRECISION ANALYSIS FOR FOUR METHODS

Method	Average Precision
WS_RSSC Method	0.74
SSA_TER Method	0.80
CNN_MCP_TSC Method	0.79
Proposed MSK-TCNN-RAVO Method	<b>0.89</b>

**C. Recall**

Recall measures the completeness, or sensitivity, of a classifier. Higher recall means less false negatives, while lower recall means more false negatives. The Recall can be calculated using (40).

$$Recall = \frac{TP}{(TP+FN)} \tag{40}$$

The average recall values are depicted in Fig. 12. The MSK-TCNN-RAVO method achieves the higher recall value of 0.895 which indicates that the percentage of true positives correctly classified by the proposed model is high compared to other existing models.

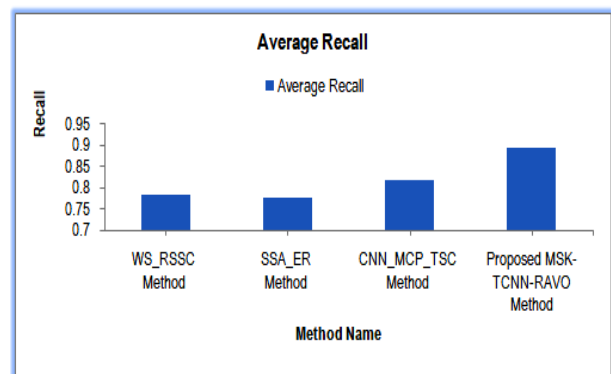


Fig. 12. Average Recall Analysis for Four Methods.

D. F-Score

In F-Score, precision and recall can be combined to produce a single metric known as F-measure, which is the weighted harmonic mean of precision and recall. The F-score can be calculated using (41).

$$Fscore = \frac{2*(precision*Recall)}{(Precision+Recall)} \quad (41)$$

The F-Score analysis is represented in Fig. 13. The F-Score value corresponding to a specific method and particular database is computed by processing the 5000 test review samples. Highest F-score value 0.947 is achieved by the MSK-TCNN-RAVO method for the DS4\_MPRSE4G dataset.

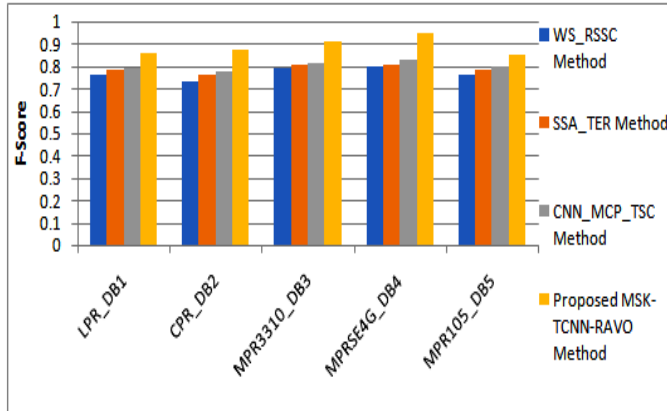


Fig. 13. F-Score Analysis for Five Database and Four Methods.

E. Mean Square Error

In Machine Learning, main goal is to minimize the error which is defined by the Loss Function. It can be computed using (42). The MSE analysis values are listed in Table III.

$$MSE = \frac{1}{N} \sum_{i=1}^N (PR_i - GT_i)^2 \quad (42)$$

Here, N denotes the No. of samples tested,  $PR_i$  indicates the predicted class for  $i^{th}$  sample, and  $GT_i$  denotes the ground truth class for  $i^{th}$  sample.

It can be observed from the Table III that the MSE value for the proposed method is drastically minimized after the RAVO optimization. The least error value after optimization is 426, which indicates that the proposed method's prediction error rate is very low when compared to the other three current models. The WS\_RSSC method produces the highest error value of 1090 in this analysis for the DS1\_LPR dataset.

F. Cross Entropy

This metric is used in neural networks when a classifier's output is multiclass prediction probabilities. In general, minimizing categorical cross-entropy gives greater accuracy for the classifier [33]. It can be computed using (43), and its computed values are depicted in Fig. 14.

$$LogarithmicLoss (lls) = -\frac{1}{N} \sum_{i=1}^N \sum_{j=1}^M y_{ij} * \log(p_{ij}) \quad (43)$$

TABLE III. MSE ANALYSIS FOR FIVE DATABASES AND FOUR METHODS

Database Name	MSE				
	WS_R SSC Method	SSA_T ER Method	CNN_MCP_TSC Method	Proposed MSK-TCNN-RAVO Method (Before RAVO)	Proposed MSK-TCNN-RAVO Method (After RAVO)
DS1_LPR	1090	1050	975	665	615
DS2_CPR	1120	1135	1055	800	745
DS3_MPR3310	995	925	880	520	473
DS4_MPRSE4G	910	870	800	475	426
DS5_MPR105	1035	980	930	555	503

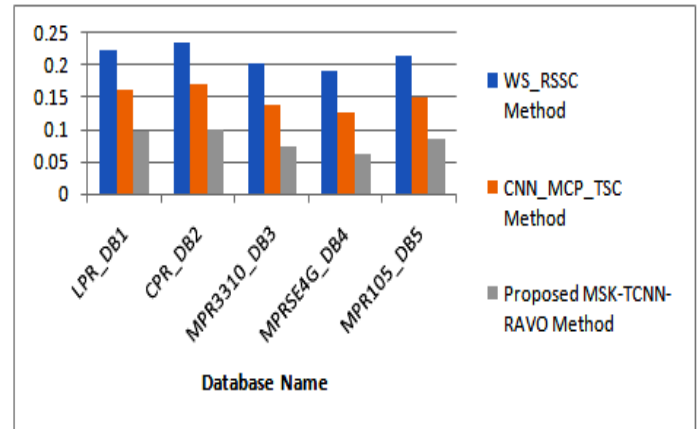


Fig. 14. Cross Entropy Analysis for Five Databases and Three Methods.

Here,  $y_{ij} = 1$ , if the sample  $i$  belongs to class  $j$  else 0.  $p_{ij}$  is the probability of the classifier that predicts of sample  $i$  belonging to class  $j$ . Since, the SSA\_TER method is not coded based on neural network, this analysis is performed for the other three methods and all datasets. The proposed method gives a least value of 0.0619 after RAVO optimization on DS4\_MPRSE4G dataset which means it gives high probability for classified data with more stabilized value.

G. Time Taken

In time taken analysis, the time consumed by the methods to produce the classification results is evaluated. The time taken by all methods is listed in Table IV.

In this analysis, WS\_RSSC method takes the least execution time for all datasets. But, when considering the classification accuracy of the proposed method, the excess time taken by the proposed model is acceptable. The time consumption difference between WS\_RSSC and the MSK-TCNN-RAVO classifier is very less i.e., 1.26 seconds. Similarly, the time consumption difference between CNN\_MCP\_TSC and the proposed classifier is only 0.76 seconds.

TABLE IV. TIME TAKEN ANALYSIS FOR FIVE DATABASES AND FOUR METHODS

Database Name	Time Taken (in seconds)			
	WS_RSSC Method	SSA_TER Method	CNN_MCP_TSC Method	Proposed MSK-TCNN-RAVO Method
DS1_LPR	5.472	5.892	7.513	<b>6.726</b>
DS2_CPR	5.593	5.986	7.742	<b>6.995</b>
DS3_MPR3310	5.401	5.802	7.298	<b>6.497</b>
DS4_MPRSE4G	5.512	5.923	7.632	<b>6.883</b>
DS5_MPR105	5.456	5.841	7.396	<b>6.654</b>

H. Misclassification Ratio

This analysis is carried out to determine the ratio of incorrect classification results generated by the classifier. It can be computed using (44). The ratios of misclassified results generated by all the methods are depicted in Fig. 15.

$$Misclassification = \frac{FP+FN}{Total\ Data} \tag{44}$$

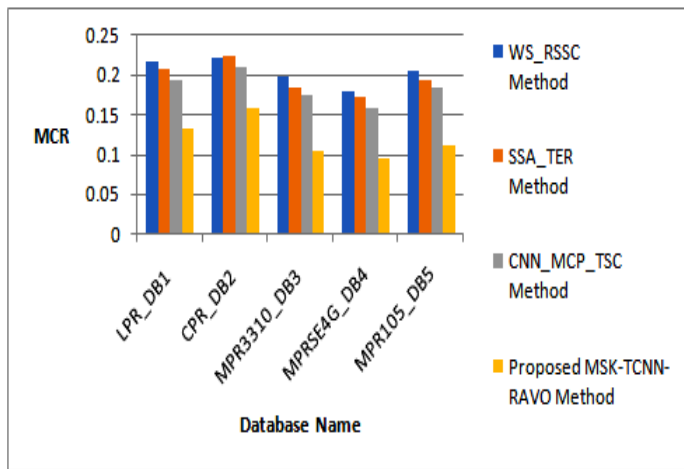


Fig. 15. Misclassification Ratio Analysis for Five Databases and Four Methods.

From the analysis, the misclassification ratio produced by the proposed method is very low for DS4\_MPRSE4G dataset which is 0.095. It shows that the proposed classifier produces very low incorrect classification results. The second best method which has less misclassification ratio is CNN\_MCP\_TSC which has 0.160 for the same dataset.

I. Performance based Ranking Index

This analysis ranks the review classification methods based on the ranking index. The ranked values are depicted in Fig. 16.

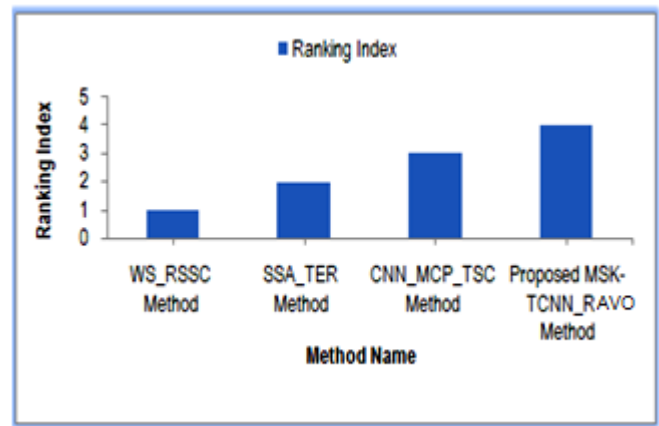


Fig. 16. Ranking Index Analysis for the Four Methods.

According to classification accuracy, precision, f-score, recall, mean square error, cross entropy, misclassification results, and the ranking index of each method is decided. The best method among the four methods is set with rank 4, i.e., the highest rank. The second best method is ranked by 3 and so on. Hence, from the above Fig. 16, the proposed method achieves the highest fourth rank which shows it is the best method for review classification than the other existing methods.

V. CONCLUSION

In this work, a classification method namely MSK-TCNN-RAVO is proposed for e-commerce reviews. The proposed algorithm is executed on five different datasets from three different domains utilized for experimentation and efficient analysis, with 7000 review samples per database being used for training and testing the proposed classifier. The MSK-TCNN-RAVO classifier achieves the highest classification accuracy of 95.5% when compared to other three existing algorithms such as WS\_SSC, SSA\_TER, and CNN\_MCP\_TSC which achieves the accuracy of 81.8%, 82.6%, and 84% respectively. The highest F-Score value of 0.947 is achieved by the proposed method when compared to other three methods. The proposed MSK-TCNN-RAVO classifier outperforms the other three approaches in all metrics, and it consistently performs well on the DS4\_MPRSE4G dataset. When compared to the WS\_RSSC technique, the proposed classifier improves classification accuracy by 8.54% before RAVO and 12.86% after RAVO optimization. Furthermore, the MSK-TCNN-RAVO classifier achieves the highest values on all evaluation metrics such as Classification accuracy, Precision, Recall, F-Score, MSE, Cross entropy, and Misclassification ratio, which proves that the proposed method is well suited for opinion mining. When processing the emoticons in the input reviews, the computing complexity of the proposed work grows. Additionally, a slight tweak to the method will be planned to address the increased CPU utilization observed when the emojis are being trained into the classifier. In future work, it is intended to modify the coding to extend the MSK-TCNN-RAVO classifier to produce hepta classifier which classifies the reviews into seven mode classification results. Furthermore, in order to further enhance the classification accuracy and reduce processing time, an

extended stop words list will be employed as a future improvement to eliminate basic stop words and survey stop words in the preprocessing phase.

#### DECLARATION

##### Competing Interests

- There are no conflicts of interest.

##### Funding

- There is no funding available.

##### Authors' contributions

- Methodology design, analysis, dataset selection, experiment evaluation is performed by the first author
- Correction is performed by the second author

#### ACKNOWLEDGMENT

I would like to express my heartfelt gratitude to our university for offering unwavering support in all aspects to complete our research project successfully.

#### REFERENCES

- [1] "Wikipedia". [Online].Found at: [https://en.wikipedia.org/wiki/Online\\_shopping](https://en.wikipedia.org/wiki/Online_shopping) [Accessed On: 26-November-2021].
- [2] "IGIGlobal".[Online].Found at: <https://www.igi-global.com/dictionary/using-the-flipped-classroom-to-improve-knowledge-creation-of-masters-level-students-in-engineering/21327> [Accessed On: 12-December-2021].
- [3] A. Yousaf, "Emotion recognition by textual tweets classification using voting classifier (LR-SGD)", *IEEE Access*, 9, pp. 6286–6295, 2021.
- [4] R. Ren and D. Wu, "An innovative sentiment analysis to measure herd behavior", *IEEE Trans. Syst. Man, Cybern.Syst.*, 50(10), pp. 3841–3851, 2020.
- [5] P. Sasikala and L. Mary Immaculate Sheela, "Sentiment analysis of online product reviews using DLMNN and future prediction of online product using IANFIS", *J. Big Data*, 7(1), 2020.
- [6] Z. Shaukat, A. A. Zulfiqar, C. Xiao, M. Azeem, and T. Mahmood, "Sentiment analysis on IMDB using lexicon and neural networks", *SN Appl. Sci.*, 2(2), 2020.
- [7] Z. Jia, X. Bai, and S. Pang, "Hierarchical gated deep memory network with position-aware for aspect-based sentiment analysis", *IEEE Access*, 8, pp. 136340–136347, 2020.
- [8] Y. Cheng, L. Yao, G. Xiang, G. Zhang, T. Tang, and L. Zhong, "Text sentiment orientation analysis based on multi-channel CNN and bidirectional GRU with attention mechanism", *IEEE Access*, 8, pp. 134964–134975, 2020.
- [9] M. Dong, Y. Li, X. Tang, J. Xu, S. Bi, and Y. Cai, "Variable convolution and pooling convolutional neural network for text sentiment classification", *IEEE Access*, 8, pp. 16174–16186, 2020.
- [10] V. Gupta, A. Aggarwal, and T. Chakraborty, "Detecting and characterizing extremist reviewer groups in online product reviews", *IEEE Trans. Comput. Soc. Syst.*, 7(3), pp. 741–750, 2020.
- [11] M. M. Madbouly, S. M. Darwish, and R. Essameldin, "Modified fuzzy sentiment analysis approach based on user ranking suitable for online social networks", *IET Softw.*, 14( 3), pp. 300–307, 2020.
- [12] R. Bhalla and A. Bagga, "Opinion mining framework using proposed rbayes model for text classification", *Int. J. Electr.Comput.Eng.*, 9(1), pp. 477–484, 2019.
- [13] S. Zhang and H. Zhong, "Mining users trust from E-Commerce reviews based on sentiment similarity analysis", *IEEE Access*, 7, pp. 13523–13535, 2019.
- [14] A. Abdul Aziz and A. Starkey, "Predicting supervise machine learning performances for sentiment analysis using contextual-based approaches", *IEEE Access*, 8, pp. 17722–17733, 2020.
- [15] F. Iqbal, "A Hybrid framework for Sentiment analysis using genetic algorithm based feature reduction", *IEEE Access*, 7, pp. 14637–14652, 2019.
- [16] H. Liu, P. Burnap, W. Alorainy, and M. L. Williams, "A fuzzy approach to text classification with two-Stage training for ambiguous instances", *IEEE Trans. Comput. Soc. Syst.*, 6(2), pp. 227–240, 2019.
- [17] Y. Fang, H. Tan, and J. Zhang, "Multi-strategy sentiment analysis of consumer reviews based on semantic fuzziness", *IEEE Access*, 6, pp. 20625–20631, 2018.
- [18] H. Zou, X. Tang, B. Xie, and B. Liu, "Sentiment classification using machine learning techniques with syntax features", *Proc. Int. Conf. Comput. Sci. Comput. Intell.* 2015, 5(4), pp. 175–176, 2016.
- [19] A. L. Maas, R. E. Daly, P. T. Pham, D. Huang, A. Y. Ng, and C. Potts, "Learning word vectors for sentiment analysis", *ACL-HLT-Proc. 49th Annu. Meet. Assoc. Comput. Linguist. Hum. Lang. Technol.*, 1, pp. 142–150, 2011.
- [20] T. Mullen and N. Collier, "Sentiment analysis using support vector machines with diverse information sources", *Proc. Conf. Empir.Methods Nat. Lang. Process.* pp. 412–418, 2004.
- [21] S. Tan, X. Cheng, Y. Wang, and H. Xu, "Adapting naive bayes to domain adaptation for sentiment analysis", *Lect. Notes Comput. Sci*, 5478, pp. 362–374, 2009.
- [22] K. Zhang, M. Jiao, X. Chen, Z. Wang, B. Liu, and L. Liu, "SC-BiCapsNet: A Sentiment classification model based on bi-channel capsule network", *IEEE Access*, 7, pp. 171801–171813, 2019.
- [23] J. Zhao, "Pre-processing boosting twitter sentiment analysis?", *Proc. - IEEE Int. Conf. Smart City, SmartCity*, pp. 748–753, 2018.
- [24] Z. Jianqiang and G. Xiaolin, "Comparison research on text pre-processing methods on twitter sentiment analysis", *IEEE Access*, 5, pp. 2870–2879, 2017.
- [25] V. Singh and B. Saini, "An effective tokenization algorithm for Information", pp. 109–119, 2014.
- [26] "Onix Text Retrieval Toolkit API". [Online].Found at: <http://www.lextek.com/manuals/onix/stopwords1.html> [Accessed On: 30-October-2021].
- [27] "DeepAI".[Online].Found at: <https://deepai.org/machine-learning-glossary-and-terms/relu> [Accessed On: 13-November-2021].
- [28] W. Zhao., "Weakly-supervised deep embedding for product review sentiment analysis", *IEEE Trans. Knowl.Data Eng.*, vol. 30(1), pp. 185–197, 2018.
- [29] S. Zhang, H. Zhong, "Mining Users Trust From E-Commerce Reviews Based on Sentiment Similarity Analysis", *IEEE Access*, vol. 7, pp. 13523–13535, 2019.
- [30] M. Dong, Y. Li, X. Tang, J. Xu, S. Bi, Y. Cai, "Variable Convolution and Pooling Convolutional Neural Network for Text Sentiment Classification", *IEEE Access*, vol. 8, pp. 16174–16186, 2020.
- [31] Kaggle, "Amazon Reviews: Unlocked Mobile Phones", 2021. [Available: <https://www.kaggle.com/PromptCloudHQ/amazon-reviews-unlocked-mobile-phones>. [Accessed On: 14-January-2021].
- [32] Kaggle, "Amazon Mobile Reviews", 2021. [Available: <https://www.kaggle.com/binaryjoker/amazon-mobile-reviews>. [Accessed On: 26-February-2021].
- [33] "TowardsDataScience". [Online].Found at: <https://towardsdatascience.com/the-5-classification-evaluation-metrics-you-must-know-aa97784ff226> [Accessed On: 26-December-2021].