Intelligent Automation & Soft Computing

Tech Science Press

DOI: 10.32604/iasc.2023.035334
Article

Modified Sine Cosine Optimization with Adaptive Deep Belief Network for Movie Review Classification

Hala J. Alshahrani[1], Abdulbaset Gaddah[2], Ehab S. Alnuzaili[3], Mesfer Al Duhayyim[4,*], Heba Mohsen[5], Ishfaq Yaseen[6], Amgad Atta Abdelmageed[6] and Gouse Pasha Mohammed[6]

[1]Department of Applied Linguistics, College of Languages, Princess Nourah bint Abdulrahman University, P.O. Box 84428, Riyadh, 11671, Saudi Arabia
[2]Department of Computer Sciences, College of Computing and Information System, Umm Al-Qura University, Saudi Arabia
[3]Department of English, College of Science & Art at Mahayil, King Khalid University, Muhayel Aseer, 63311, Saudi Arabia
[4]Department of Computer Science, College of Sciences and Humanities-Aflaj, Prince Sattam bin Abdulaziz University, Al-Aflaj, 16828, Saudi Arabia
[5]Department of Computer Science, Faculty of Computers and Information Technology, Future University in Egypt, New Cairo, 11835, Egypt
[6]Department of Computer and Self Development, Preparatory Year Deanship, Prince Sattam bin Abdulaziz University, AlKharj, Saudi Arabia

*Corresponding Author: Mesfer Al Duhayyim. Email: m.alduhayyim@psau.edu.sa
Received: 16 August 2022; Accepted: 23 November 2022

Abstract: Sentiment analysis (SA) is a growing field at the intersection of computer science and computational linguistics that endeavors to automatically identify the sentiment presented in text. Computational linguistics aims to describe the fundamental methods utilized in the formation of computer methods for understanding natural language. Sentiment is classified as a negative or positive assessment articulated through language. SA can be commonly used for the movie review classification that involves the automatic determination that a review posted online (of a movie) can be negative or positive toward the thing that has been reviewed. Deep learning (DL) is becoming a powerful machine learning (ML) method for dealing with the increasing demand for precise SA. With this motivation, this study designs a computational intelligence enabled modified sine cosine optimization with a adaptive deep belief network for movie review classification (MSCADBN-MVC) technique. The major intention of the MSCADBN-MVC technique is focused on the identification of sentiments that exist in the movie review data. Primarily, the MSCADBN-MVC model follows data pre-processing and the word2vec word embedding process. For the classification of sentiments that exist in the movie reviews, the ADBN model is utilized in this work. At last, the hyperparameter tuning of the ADBN model is carried out using the MSCA technique, which integrates the Levy flight concepts into the standard sine cosine algorithm (SCA). In order to demonstrate the significant performance of the MSCADBN-MVC model, a wide-ranging experimental analysis is performed on three different datasets. The comprehensive study highlighted the enhancements of the MSCADBN-MVC model in the movie review classification process with maximum accuracy of 88.93%.

Keywords: Computational linguistics; movie review analysis; sentiment analysis; sentiment classification; deep learning

1 Introduction

Computational linguists create resources and tools for significant practical tasks like information extraction from text, machine translation, speech synthesis, text mining, grammar checking, and speech recognition. Computational linguistics refers to the study of computer systems to understand Natural language generation [1]. Tools that operate in computational linguistics utilize artificial intelligence (AI), that is, formal methods, Algorithms, data structures to express knowledge, methods for inference process, etc. Recent advancement in the web has impacted the everyday life of humans, and the necessity of user view analysis has seen exponential growth [2]. The flow of massive volumes of data can affect the decision-making process in organizations. Analyzing emotions, aspects, and reactions of people with regard to entities like issues, services, events, products, and their attributes by the feedback from Web pages is known as opinion mining [3]. The improvement in the domain of web technology made changes in the way of expressing people's opinions. People depend on the user perspective information to analyze the substances for online purchases or booking movie tickets to watch movies [4]. The users will interface by tweets on Twitter, posts on Facebook, and so on. The measure of data can be huge to highlight that it becomes a burden for a common man to examine and come to a conclusion.

Sentiment analysis (SA) can be broadly organized into 2 types they are classification methods and information-oriented method. An information-oriented method needs an extensive database of predefined feelings and an effective information portrayal to recognize sentiments [5]. There exists a massive volume of opinionated data accessible in digital formats, for example, microblogs, reviews, social networks, Twitter forum discussions, and blogs [6]. Therefore, studying SA has an overwhelming effect on economics, natural language processing (NLP), political science, social sciences, and management sciences since they are influenced by people's opinions. A sentiment can be a negative or positive emotion, opinion, feeling, or valuation regarding a term, feature, or attribute from a sentiment holder. Neutral, Positive, and negative views are known as sentiment orientations (also termed semantic polarities, orientations, or opinion orientations) [7].

Owing to the evolution of massive data and the volume of data that has been produced and exchanged every minute, analyzing, mining, and comprehending this data has increased remarkably [8]. And then, the regular Neural Networks (NN) and machine learning (ML) methods are inadequate to gain big data; deep learning (DL) has become the key to the big data period. DL refers to a subfield of ML and an alternative to NNs. Simply, regular NN was a single network having output and input layers along with the hidden layers simultaneously performing computation. Deep Neural Networks (DNN) have many NNs in which the output of a single network was input to the next network, and so on [9]. This has overcome the limitation of the number of hidden layers in NNs and works with big data feasibly. DL networks will learn the attributes on their own, i.e.,; it is a robust ML method that studies multiple layers of features and induces outcomes of prediction [10]. DL is currently utilized in several applications in the domain of information processing and signal, particularly with the development of big data. Moreover, DL networks were utilized in opinion mining and SA.

This study develops a computational intelligence-enabled modified sine cosine optimization with adaptive deep belief network for movie review classification (MSCADBN-MVC) technique. The major intention of the MSCADBN-MVC technique is focused on the identification of sentiments thatthat exist in the movie review data. Primarily, the MSCADBN-MVC model follows data pre-processing and word2vec word embedding process. For the classification of sentiments exist in the movie reviews, the ADBN model is utilized in this work. At last, the hyperparameter tuning of the ADBN model is carried out using the MSCA technique. In order to demonstrate the significant performance of the MSCADBN-MVC model, a wide-ranging experimental analysis is performed on three different datasets.

2 Related Works

Rehman et al. [11] project a hybrid method utilizing long short-term memory (LSTM) and deep convolutional neural network (CNN) method called Hybrid CNN-LSTM technique to address the SA problem. Firstly, the author employs the Word to Vector (Word2Vec) technique for training primary word embedding. The Word2Vec will translate the text strings to a vector of numerical values, calculates the distance in words, and creates groups of comparable words related to their meanings. And then, embedding can be executed where the presented method integrates a group of features that can be derived by global max-pooling, and convolution layers have long-term dependences. Abidin et al. [12] intend to categorize the SA of movie reviews gained from the IMDb website. The support vector machine (SVM) technique has been used for classifying the movie review's sentiments. At the same time, the information gain (IG) and radial basis function (RBF) kernel are utilized for enhancing classification. Feature selection has been made through the removal of unrelated attributes and choosing attributes with a strong correlation for classifying purposes. The IG method has been employed for selecting features.

In [13], the author makes use of LSTMs, a variant of recurrent neural network (RNN) for predicting the SA for the mission of movie review analysis. LSTMs seem to be excellent in devising very long sequential data. The issue was imposed as a binary classifier task in which the review is negative or positive. Sentence vectorizing techniques were utilized to deal with the variability of the sentence length. In this study, the author tries to examine the effect of hyperparameters such as activation functions, dropout, and the number of layers. Ullah et al. [14] devises a DNN having 7 layers for movie reviews' SA. The method comprises input layers named an embedding layer that denotes the data as a number of sequences named vectors and 2 consecutive layers of 1D-CNN (1-dimensional CNN) to extract features. A global max-pooling layer can be used for dimension reduction. A dense layer for classifying purposes and a dropout layer were utilized to reduce overfitting and enhance generalization error in the NN. An FC layer was the final layer for predicting 2 classes.

Ramadhan et al. [15] focused on analyzing the sentiment of opinions from numerous comments from IMDB site users utilizing the star rating aspect and categorized utilizing the SVM method. SA was a classifying process for understanding the emotions, opinions, and interactions of a text or document. SVM can be fruitful for several applications in engineering and science, particularly for classification (pattern identification) issues. Along with the SVM technique, the term frequency-inverse document frequency (TF-IDF) method was utilized for changing the document shape into numerous words. In [16], the LSTM classifier can be utilized to analyze the sentiments of the IMDb movie reviews. It depends on the RNN method. The data can be efficiently pre-processed and divided to enhance the post-classifier performance. In [17], the author performs SA in 2 different movie review datasets utilizing several ML approaches. The author devises some structures of sentiment classification utilizing these methods on the given dataset.

3 The Proposed Model

In this study, a new MSCADBN-MVC technique has been developed for the identification and classification of movie reviews on social media. The major intention of the MSCADBN-MVC technique is focused on the identification of sentiments that exist in the movie review data. Fig. 1 demonstrates the overall process of the MSCADBN-MVC approach.

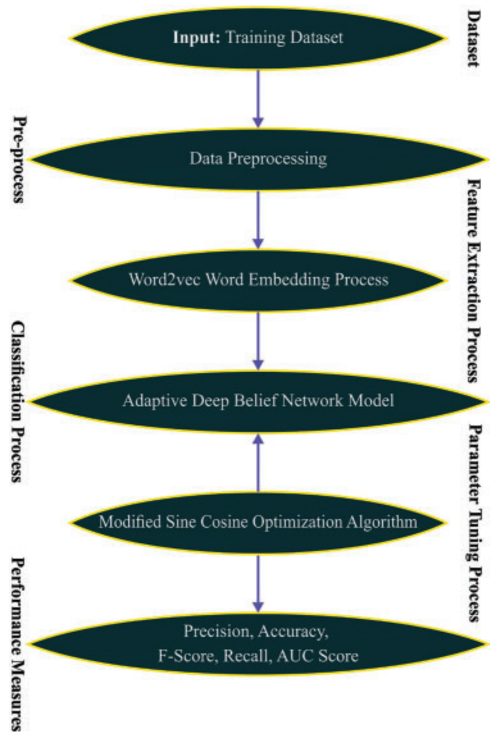

Figure 1: Overall process of MSCADBN-MVC approach

3.1 Data Pre-Processing and Word Embedding

Primarily, the MSCADBN-MVC model follows data pre-processing and the word2vec word embedding process. The pre-processing not only focus on cleaning the text but is also helpful in extracting the text features such as numerous words, symbols, and URLs which is not helpful for classifying purposes [18]. In the pre-processing stage, it involves various tasks for converting unstructured text files into a word vector as listed below:

- Stemming
- Tokenization
- Short-word removal
- Stop-word removal

Stemming: it aims to discover the root word through the removal of suffixes. In the feature space, various tokens which share a similar root-word are recognized as the same token.

Tokenization will chunk text sentences as meaningful words named tokens. With the help of the whitespaces, the text documents were chunked out of a long paragraph. Additionally, the special characters, HTML tags, punctuation, and XML scripts presented in text documents have not affected the performance. So, eliminating them will be helpful in reducing the feature count in the classification stage.

Short-word Removal: A short word with a length of 1 to 2 characters can be removed to reduce the feature or number count. Such short words were often extracted from short-form words, like ‘ok’ (read: okay) and ‘tq’ (thank you). It can be taken as noise due to the spelling and grammatical errors that are repeated in an informal text. Therefore, to reduce the feature count, it has to be eliminated from the text documents.

Stop-word Removal intends to omit terms or tokens in the text files which were commonly known as ‘functional words’ because it does not have any meaning, for example ‘but’, ‘this’, and ‘is’. The stop-words were repetitive, and it does not affect the classifying process; conversely, it minimizes computational difficulty.

Word2Vec was an open-source tool related to DL. In recent times, it is gaining more popularity due to its maximum accuracy in examining semantic similarities between 2 words and has a less computational cost. It comprises 2 modes they are Skip-Gram and Continuous bag of words (CBOW), which is applied quickly to study word embedding out of the original text and captures word relations with the built vector representing method (NN method). In this study, the Skip-Gram method has been chosen for training word embedding. Then in the text pre-processing stage, the Skip-Gram model learns the word vector representations of every word in the document and extracts the vector representation for every word encountered in the input. After that, the representation of every sentence is got through the average on the vectors of all its encompassing words. During the sentiment prediction stage, the vector representation of the novel document can be derived in the same way.

3.2 Movie Review Sentiment Classification

At this stage, the ADBN model is used for the classification of sentiments that exist in movie reviews. DBN is a model made up of more than one Restricted Boltzmann Machine (RBM) [19]. The feature extracted using the initial RBM is transferred to the upper RBM, whereas the features extracted by the latter layer of RBM are transferred to the backpropagation neural network (BPNN). According to the structural properties of multilayer, it is simple to achieve the compression coding of the data, therefore accomplishing the best feature depiction.

RBM is a building block of DBN, and it is extensively employed in pattern recognition, data reconstruction, and classification. RBM is an undirected graph mechanism that involves hidden and visible layers without linking among nodes in a similar layer, and different layers are connected. The architecture is the same as the two-layer fully convolutional network (FCN) while considering each visible and hidden layers are binary distributed. RBM is an energy mechanism where the energy function is utilized originally. At the same time, the neural network uses RBM, with a vector v and h signifying neuron in the visible and hidden units as follows:

$$E(v,\ h|\theta)_{RBM} = -\sum\nolimits_{i=1}^{I}\sum\nolimits_{j=1}^{J} v_j\omega_{ij}h_j - \sum\nolimits_{i=1}^{I} a_i v_i - \sum\nolimits_{j=1}^{J} b_j h_j \tag{1}$$

Now v_i and h_j indicate the binary state of the i and j visible layers., a_i, and b_j denote the biases of visible and hidden units correspondingly. $\omega_j i$ shows the weight connecting between v_i and h_j, and $\theta = (\omega_{ij},\ a_i,\ b_j)$ are variables for RBM. The hidden and visible units of the RBM are the Bernoulli distribution.

The marginal distribution of RBM is:

$$P\left(v|\theta\right)_{RBM} = \frac{\sum_{j=1}^{J} e^{-E(v,h|\theta)_{RBM}}}{z\left(\theta\right)} \tag{2}$$

Here $z\left(\theta\right)$ indicates a partition function, viz., $z\left(\theta\right) = \sum_v \sum_h e^{-E(v,h|\theta)}$.

Assume that the amount of instances is I and the present instance is i. The variable θ is attained by finding the maximal likelihood function of the trained instance:

$$\begin{aligned} \theta^*_{RBM} &= maximize_{\omega_{ij},a_i,b_j} l(\theta)_{RBM} \\ &= maximize_{\omega_{ij},a_i,b_j} \left(\frac{1}{I}\sum_{i=1}^{I} lnP\left(v^{(i)}|\theta\right)_{RBM}\right) \end{aligned} \tag{3}$$

To attain the optimum value θ^*_{RBM} of θ, the maximal value of $l(\theta)_{RBM}$ is attained through the stochastic gradient ascent methodology in RBM.

$$\begin{cases} \dfrac{\partial l\left(\theta\right)}{\partial \omega_{ij}} = \langle v_i h_j \rangle_{data} - \langle vh \rangle_{recon} \\ \dfrac{\partial l\left(\theta\right)}{\partial a_i} = \langle v_i \rangle \, data - \langle v_i \rangle_{recon} \\ \dfrac{\partial 1\left(\theta\right)}{\partial b_j} = \left\langle h_{jdata} \right\rangle - \langle h_j \rangle_{recon} \end{cases} \tag{4}$$

Now, $\langle \cdot \rangle_{data}$ and $\langle \cdot \rangle_{recon}$ denotes the mathematical expectation of the distribution described as the trained data and reconstructed model. Fig. 2 depicts the infrastructure of the DBN technique.

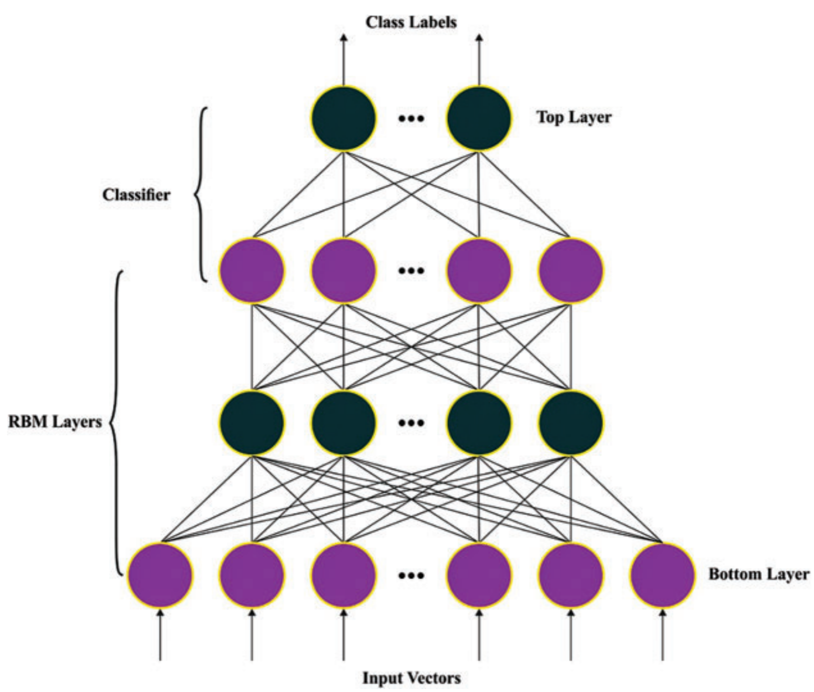

Figure 2: Structure of DBN

By utilizing the Contrastive Divergence (CD) approach for training RBM and different from Gibbs sampling, the CD approach needs k steps of Gibbs sampling to retrieve a better fit. The CD method comprises a recursive step of Gibbs sampling that estimates the data distribution and model by means of a larger amount of continuous updates, which improves the computation speed of learning and ensures the computation performance.

Stochastic gradient descent (SGD) and variant might be the increasingly common optimization algorithm in ML; however, the learning method is sometimes slower, and the update direction is dependent wholly on the current batch. Therefore, its update is extremely unstable.

The momentum model, known as classical momentum (CM), is a method to go faster gradient descent that collects velocity vector in the direction wherein the objective lens continues to reduce during the procedure of iteration. CM updates the direction of the previous renewal and exploits the gradient of the current batch to finetune the last update direction. Therefore, stability is improved to a range such that learning is faster, and it is evaluated in the following:

$$v_t = \gamma v_{t-1} + \eta \nabla_\theta J\left(\theta\_(\mathrm{t}-1)\right) \tag{5}$$

$$\theta_i = \theta_{t-1} - v_t$$

In Eq. (5), θ characterizes the DBN model parameter, v_t and v_{t-1} signify the gradient afterwards at the end of t and t-l cycles, η indicates the learning rate of RBM, $I(\theta)$ and ∇_θ characterize the error function and partial derivative of error function than that of parameter model, correspondingly. $\gamma \in [0, 1]$ represents the momentum coefficients viz., generally fixed as 0.9, and $\eta > 0$ refers to the learning rate.

The stochastic gradient descent (SGD) optimizer with momentum causes the gradient in a similar direction to accumulate, as different directions cancel one another, which might accelerate the method to the optimum point. Nevertheless, the momentum gradient is sightless. The gradient drops faster; however, it could judge where the parameter will fall. Consequently, the variable still drops faster once it is nearer the optimum solution that it might miss it. In such cases, Nelder mead (NM) could be a possible solution. The dissimilarity between momentum and NM is considered in the gradient computation. In NM, the gradient is computed afterwards the existing speed is employed. Consequently, NM is described by adding a correction factor to the momentum model that evaluates the gradient of $(\theta - \gamma v_{t-1})$, for calculating the position of the parameter next drop. These situations slow down the drop beforehand the parameter reaches its optimum point, such that missing these points might be prevented as presented in Eq. (6) below.

$$v_t = \gamma v_{t-1} + \eta \nabla_\theta J\left(\theta_{t-1} - \gamma v_{t-1}\right) \tag{6}$$

$$\theta_t = \theta_{t-1} - v_t$$

But sometimes, the NM approach is very conservative. For the RBM training to accomplish adequate training speed and classification, an independent adaptive learning rate is presented. $_{Wij}$ is upgraded as follows:

$$h_{ij}(k) = \begin{cases} h_{ij}(k-1) + \alpha & if \ \left(grad_{ij}^{k} grad_{ij}^{k-1}\right) > 0 \\ h_{ij}(k-1) \times (1-\alpha) & if \ \left(grad_{ij}^{k} grad_{ij}^{k-1}\right) < 0 \end{cases} \tag{7}$$

$$\Delta w_{ij}^{k} = \eta \cdot h_{ij}^{k} \cdot \nabla_\theta J\left(\theta_{t-1} - \gamma v_{t-1}\right) \tag{8}$$

In Eq. (7), $grad_{ij}^{k}$ specifies the gradient of w_{ij} afterward the $k\text{-}th$ training, $h_{ij}(0) = 1$. Once the gradient direction of contemporary weight keeps the same as the preceding one, $h_{ij}(t)$ rises α times. On the contrary, $h_{ij}(t)$ is reduced $1 - \alpha$ times. The variable α is fixed to 0.1, so it is smaller enough to ensure that the learning rate won't increase faster, which might cause the miss of the optimum point. The range of $h_{ij}(t)$ must be a constraint to [0.01, 100] for preventing gradient vanishing. The presented training model could avoid the problems of negative acceleration of the training speed. This compensates the gradient update to a specific range by improving the falling step size to ensure that the parameter gets high training speed without any loss of the optimum solution once the NM module incorrectly judges as presented in Eq. (8).

3.3 Design of MSCA for Hyperparameter Optimization

Finally, the hyperparameter tuning of the ADBN model is carried out using the MSCA technique. SCA draws motivation from the mathematical modelling of the sine and cosine trigonometric functions [20]. The solution position in the population is upgraded according to the sine and cosine functions output, which makes them oscillate near the optimum solution. The return value of a function is -1 and $+1$, which keeps the solution changeable. The process initiates by producing a sequence of arbitrary candidate solutions within the limits of the problem in the initialization stage. Exploration and exploitation are controlled inversely all over the implementation by arbitrary adaptive variables.

The location updating method is accomplished in all the iterations through Eqs. (9) and (10), X_i^t and X_i^{t+1} indicate the existing location of the solution in the $i-th$ parameter at $t-th$ and $i+1$ iterations; correspondingly, r_{1-3} are pseudo-randomly produced numbers, and the P_i^* represents the endpoint location (present optimum approximation) in the $i-th$ parameter, whereas || characterizes the absolute values. A similar notation in the original manuscript whereby the technique was originally projected is utilized.

$$X_i^{t+1} = X_i^t + r_1 \cdot sin\,(r_2) \cdot \left| r_3 \cdot P_i^{*t} - X_i^t \right| \tag{9}$$

$$X_i^{t+1} = X_i^t + r_1 \cdot cos\,(r_2) \cdot \left| r_3 \cdot P_i^{*t} - X_i^t \right| \tag{10}$$

The abovementioned two formulas are utilized as control parameter r_4:

$$X_i^{(t+1)} = \begin{cases} X_i^{(t+1)} = X_i^t + r_1 \cdot sin(r_2) \cdot |r_3 \cdot P_i^{*t} - X_i^t|, & r_4 < 0.5 \\ X_i^{(t+1)} = X_i^t + r_1 \cdot cos(r_2) \cdot |r_3 \cdot P_i^{*t} - X_i^t|, & r_4 \geq 0.5, \end{cases} \tag{11}$$

In Eq. (11), r_4 characterizes an arbitrarily produced integer within 0 and 1. Note that, for each constituent of every solution in the population, a novel value for pseudo-random parameter r_{1-4} is produced.

The search procedure can be controlled using four arbitrary variables, and they affect the existing and the optimal solution position. The balance between solutions is required for the optimal global convergence. It can be accomplished by altering the range of based functions in the adhoc method. Exploitation is assured through the fact that the sine and cosine function exhibits a cyclic pattern that allows for relocation nearby the solution. Change in the range of sine and cosine functions allows the procedure to search the outer of the respective destination. Moreover, the solution needs the location not to overlap with the area of another solution.

For good quality of randomness, the value for variable r_2 is provided in [0, 2Π] and that assurances exploration. The control of the balance between exploitation and diversification is given below.

$$r_1 = a - t\frac{a}{T}, \tag{12}$$

In Eq. (12), t indicates the present iteration, T characterizes the maximal amount of iterations in a run, and a shows a constant.

The MSCA technique integrates the Levy flight (LF) concepts into the standard SCA. LF is a type of chaotic system, whereas the magnitude of the leap has defined by the possibility function [21]. If the higher fly recognizes a prey area, Aquila defines the land and then strikes. It is recognized as a contour flight with rapid glide invasion. During this work, Aquila optimized a detailed examination of the target prey in certain areas from the arranging for the assault. This performance was formulated properly as follows.

$$x_{new} = x_{prey} \times Levy\ (D) + X_R(t) + (y - x) * rand, \tag{13}$$

whereas $_{new}$ implies the novel place, which is created by the searching approach (x). The dimensionality space was represented by D, and the LF distribution was represented by Levy (D), which is developed utilizing Eq. (14). At the i^{th} cycle, $X(t)$ represents the arbitrary value selected in the range [1N].

$$Levy\ (D) = s \times \frac{u \times \sigma}{|r|^{1/p}}, \tag{14}$$

In which s represents the constant fixed to 0.01, u signifies the randomized value between zero and one, and r denotes the randomized number between *zero* and one. Eq. (15) has been utilized for computing σ.

$$\sigma = \left(\frac{\Gamma(1+\beta) \times \text{sine}\left(\frac{\pi\beta}{2}\right)}{\Gamma\left(1+\frac{\beta}{2}\right) \times \beta \times 2^{\left(\beta - \frac{1}{2}\right)}} \right), \tag{15}$$

whereas β denotes the constant fixed to 1.5. y and x are being utilized for displaying the circular procedure from the seek, and it can be provided as the subsequent formula.

$$\begin{aligned} y &= r \times \cos(\theta), \\ x &= r \times \sin(\theta), \\ r &= r_1 + U \times D_1, \\ \theta &= -w \times D_1 + \theta_1, \\ \theta_1 &= \frac{3 \times \pi}{2}. \end{aligned} \tag{16}$$

In order to provide the number of search iterations, r_1 indicates the value from 1 to 20, and U stands for the smaller value fixed to 0.00565. D_1 defines the integers ranging from 1 to Dim (that is, the length of searching spaces), and w demonstrates the smaller value fixed to 0.005.

The MSCA method will derive a fitness function (FF) for achieving enhanced classifier outcomes. It specifies a positive value for denoting superior outcomes of the candidate solutions. In this article, the reduction of the classifier error rate can be regarded as the FF, which is presented below in Eq. (17).

$$fitness\,(x_i) = ClassifierErrorRate\,(x_i)$$

$$= \frac{number\ of\ misclassified\ samples}{total\ number\ of\ samples} * 100 \tag{17}$$

4 Experimental Validation

The proposed model is simulated using Python 3.6.5 tool on PC i5-8600 k, GeForce 1050Ti 4 GB, 16 GB RAM, 250 GB SSD, and 1 TB HDD. The parameter settings are given as follows: learning rate: 0.01, dropout: 0.5, batch size: 5, epoch count: 50, and activation: ReLU.

The experimental validation of the presented MSCADBN-MVC model takes place. The dataset comprises 4929 samples in the positive class and 5071 samples into the negative class, as illustrated in Table 1.

Table 1: Dataset details

Class	No. of samples
Positive	4929
Negative	5071
Total number of samples	**1000**

The confusion matrices of the MSCADBN-MVC model on the movie review classification process are given in Fig. 3. On run-1, the MSCADBN-MVC model has recognized 4814 samples into positive class and 3955 samples into negative class. In the meantime, on run-2, the MSCADBN-MVC approach has recognized 3696 samples into the positive class and 5040 samples into negative class. In parallel, on run-3, the MSCADBN-MVC method has recognized 3952 samples into positive class and 4383 samples into negative class. Also, on run-4, the MSCADBN-MVC algorithm has recognized 4534 samples into positive class and 4359 samples into negative class. Finally, on run-5, the MSCADBN-MVC methodology has recognized 3589 samples into positive class and 4785 samples into negative class.

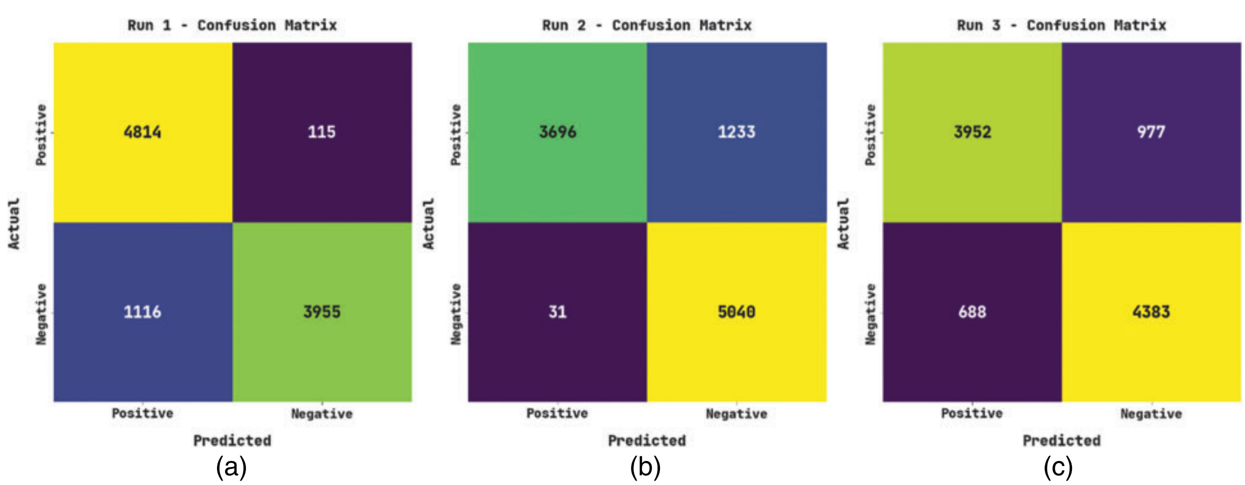

Figure 3: (Continued)

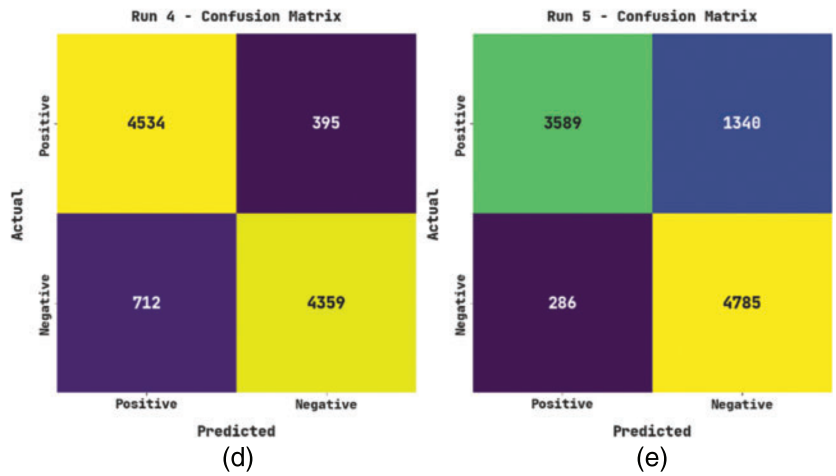

Figure 3: Confusion matrices of MSCADBN-MVC approach (a) Run1, (b) Run2, (c) Run3, (d) Run4, and (e) Run5

Table 2 provides overall classification outcomes of the MSCADBN-MVC model under five distinct runs. Fig. 4 reports a brief movie review classification results of the MSCADBN-MVC model under test run-1. The presented MSCADBN-MVC model has categorized positive movie reviews with $accu_y$ of 87.69%, $prec_n$ of 81.18%, $reca_l$ of 97.67%, F_{score} of 88.66%, and AUC_{score} of 87.83%. Likewise, the MSCADBN-MVC model has classified negative movie reviews with $accu_y$ of 87.36%, $prec_n$ of 89.76%, $reca_l$ of 87.19%, F_{score} of 87.13%, and AUC_{score} of 87.19%.

Table 2: Result analysis of MSCADBN-MVC approach with distinct measures and runs

Class	Accuracy	Precision	Recall	F-score	AUC score
Run-1					
Positive	87.69	81.18	97.67	88.66	87.83
Negative	87.69	97.17	77.99	86.53	87.83
Average	**87.69**	**87.69**	**87.69**	**87.69**	**87.69**
Run-2					
Positive	87.36	99.17	74.98	85.40	87.19
Negative	87.36	80.34	99.39	88.86	87.19
Average	**87.36**	**87.36**	**87.36**	**87.36**	**87.36**
Run-3					
Positive	83.35	85.17	80.18	82.60	83.31
Negative	83.35	81.77	86.43	84.04	83.31
Average	**83.35**	**83.35**	**83.35**	**83.35**	**83.35**
Run-4					
Positive	88.93	86.43	91.99	89.12	88.97

(Continued)

Table 2: Continued

Class	Accuracy	Precision	Recall	F-score	AUC score
Negative	88.93	91.69	85.96	88.73	88.97
Average	**88.93**	**88.93**	**88.93**	**88.93**	**88.93**
Run-5					
Positive	83.74	92.62	72.81	81.53	83.59
Negative	83.74	78.12	94.36	85.48	83.59
Average	**83.74**	**83.74**	**83.74**	**83.74**	**83.74**

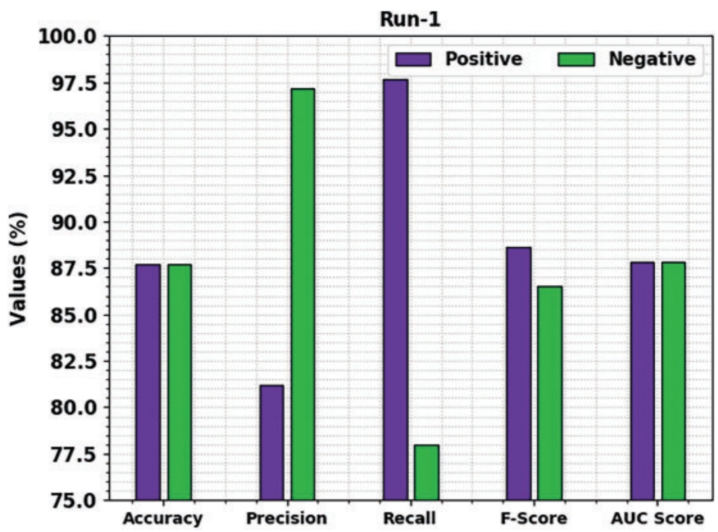

Figure 4: Result analysis of MSCADBN-MVC approach under Run-1

Fig. 5 reports a brief movie review classification results of the MSCADBN-MVC model under test run-2. The presented MSCADBN-MVC model has categorized positive movie reviews with $accu_y$ of 87.36%, $prec_n$ of 99.17%, $reca_l$ of 74.98%, F_{score} of 85.40%, and AUC_{score} of 87.19%. Likewise, the MSCADBN-MVC model has classified negative movie reviews with $accu_y$ of 87.36%, $prec_n$ of 80.34%, $reca_l$ of 99.39%, F_{score} of 88.86%, and AUC_{score} of 87.19%.

Fig. 6 reports a brief movie review classification results of the MSCADBN-MVC model under test run-3. The presented MSCADBN-MVC model has categorized positive movie reviews with $accu_y$ of 83.35%, $prec_n$ of 85.17%, $reca_l$ of 80.18%, F_{score} of 82.60%, and AUC_{score} of 83.31%. Likewise, the MSCADBN-MVC model has classified negative movie reviews with $accu_y$ of 83.35%, $prec_n$ of 81.77%, $reca_l$ of 86.43%, F_{score} of 84.04%, and AUC_{score} of 83.31%.

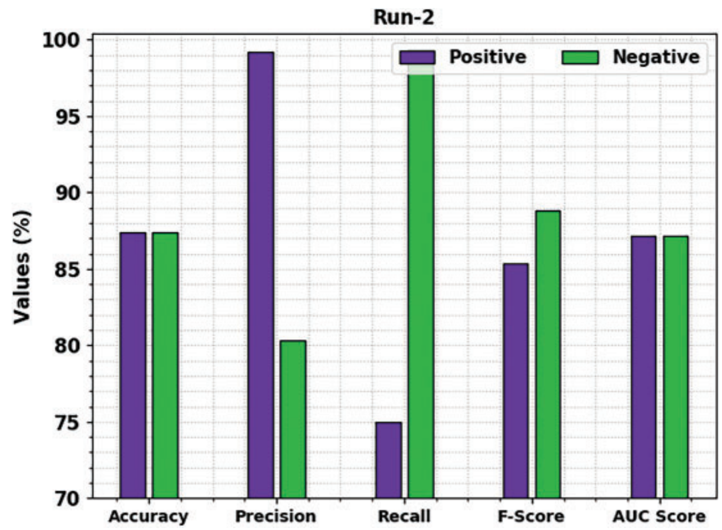

Figure 5: Result analysis of MSCADBN-MVC approach under Run-2

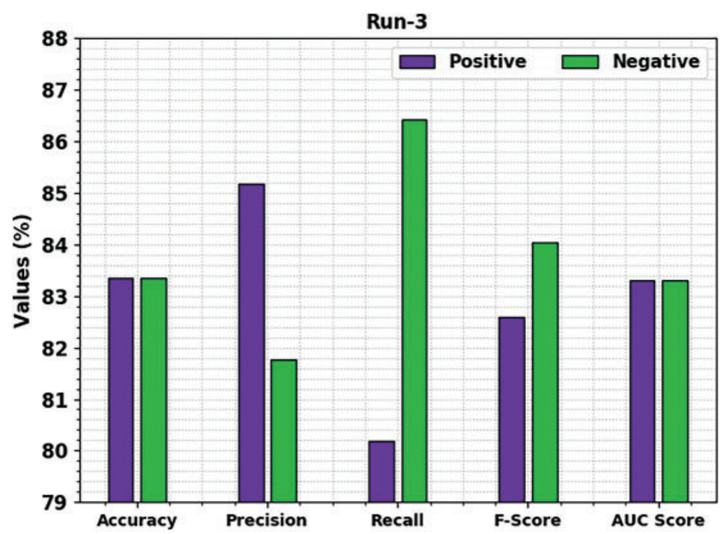

Figure 6: Result analysis of MSCADBN-MVC approach under Run-3

Fig. 7 reports a brief movie review classification results of the MSCADBN-MVC model under test run-4. The presented MSCADBN-MVC model has categorized positive movie reviews with $accu_y$ of 88.93%, $prec_n$ of 86.43%, $reca_l$ of 91.99%, F_{score} of 89.12%, and AUC_{score} of 88.91%. Likewise, the MSCADBN-MVC model has classified negative movie reviews with $accu_y$ of 88.93%, $prec_n$ of 91.69%, $reca_l$ of 85.96%, F_{score} of 88.73%, and AUC_{score} of 88.97%.

Fig. 8 reports a brief movie review classification results of the MSCADBN-MVC model under test run-5. The presented MSCADBN-MVC model has categorized positive movie reviews with $accu_y$ of 83.74%, $prec_n$ of 92.62%, $reca_l$ of 72.81%, F_{score} of 81.53%, and AUC_{score} of 83.59%. Likewise, the MSCADBN-MVC model has classified negative movie reviews with $accu_y$ of 83.74%, $prec_n$ of 78.12%, $reca_l$ of 94.36%, F_{score} of 85.48%, and AUC_{score} of 83.59%.

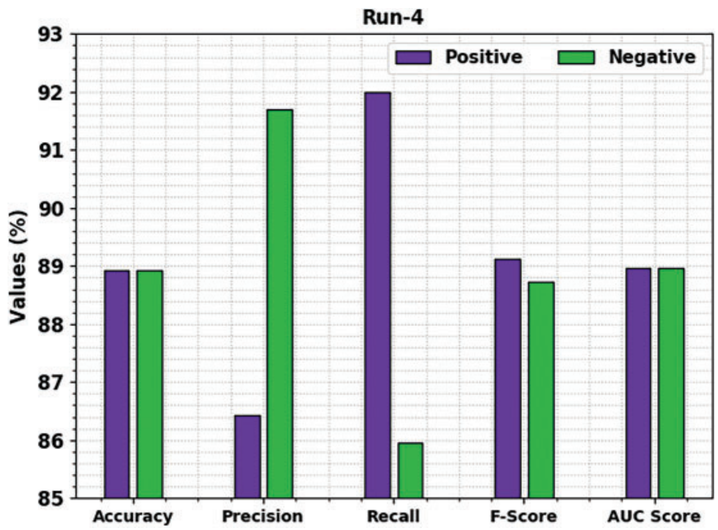

Figure 7: Result analysis of MSCADBN-MVC approach under Run-4

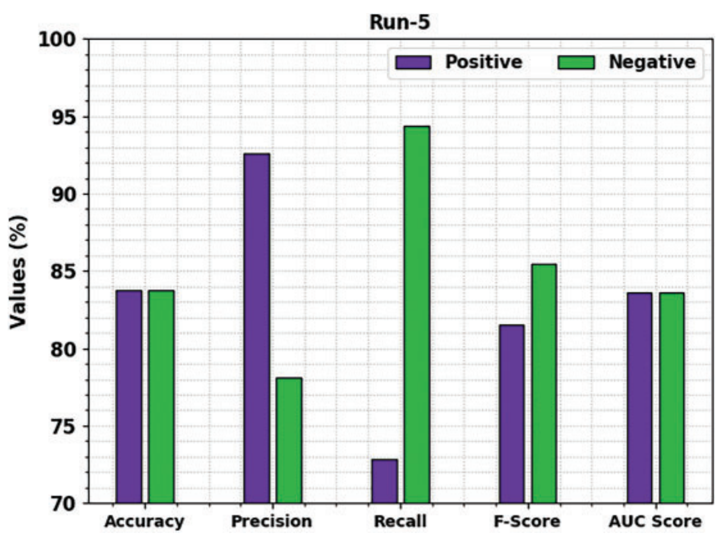

Figure 8: Result analysis of MSCADBN-MVC approach under Run-5

The training accuracy (TRA) and validation accuracy (VLA) acquired by the MSCADBN-MVC technique on test dataset is shown in Fig. 9. The experimental outcome implicit the MSCADBN-MVC algorithm has gained maximal values of TRA and VLA. Seemingly, the VLA is greater than TRA.

The training loss (TRL) and validation loss (VLL) gained by the MSCADBN-MVC technique on test dataset are exhibited in Fig. 10. The experimental outcome denoted the MSCADBN-MVC method has displayed least values of TRL and VLL. Particularly, the VLL is lesser than TRL.

A clear precision-recall investigation of the MSCADBN-MVC algorithm on test dataset is represented in Fig. 11. The figure represented the MSCADBN-MVC approach has resulted in enhanced values of precision-recall values under all classes.

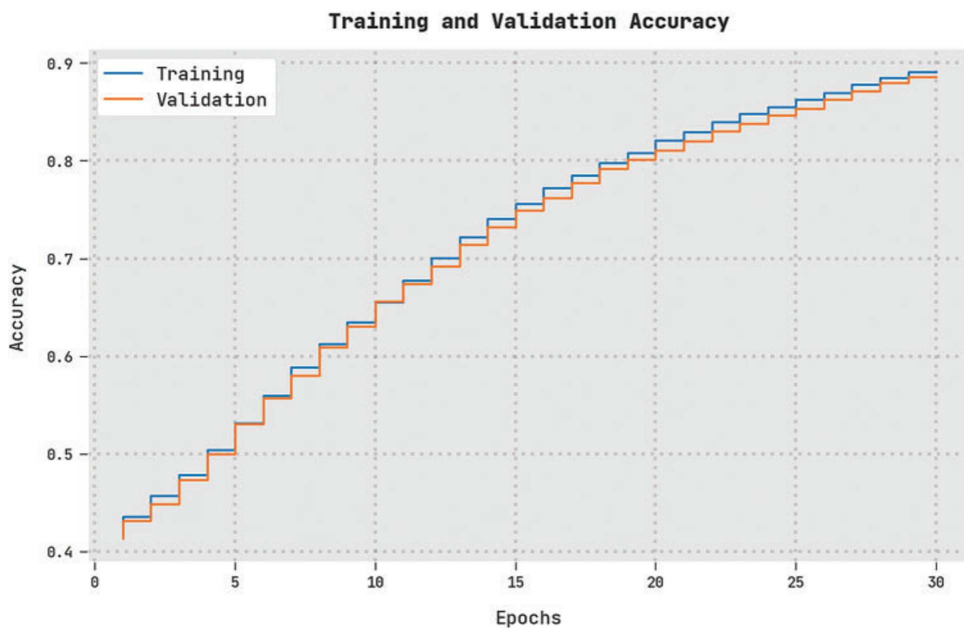

Figure 9: TRA and VLA analysis of MSCADBN-MVC approach

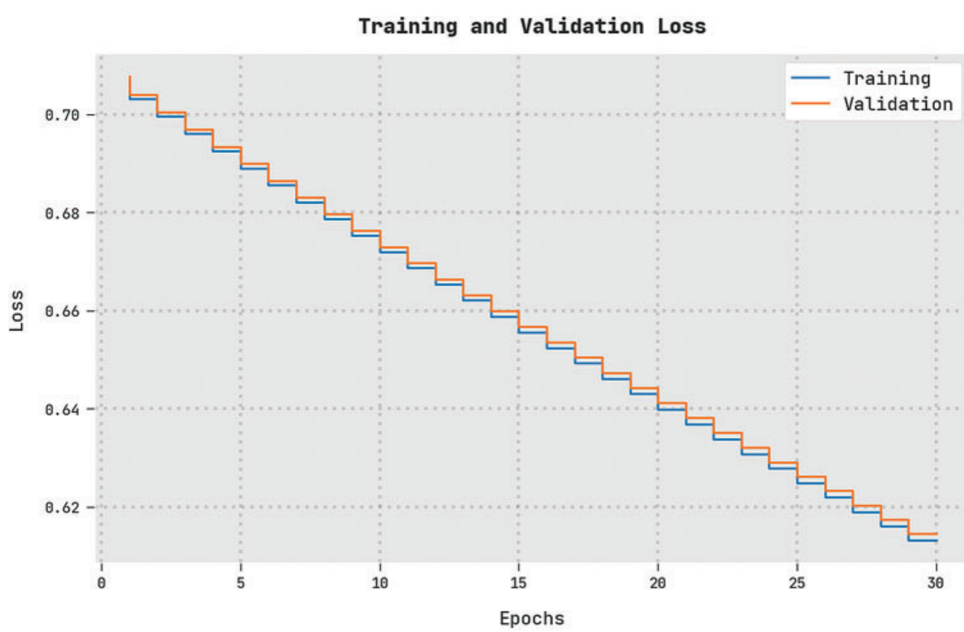

Figure 10: TRL and VLL analysis of MSCADBN-MVC approach

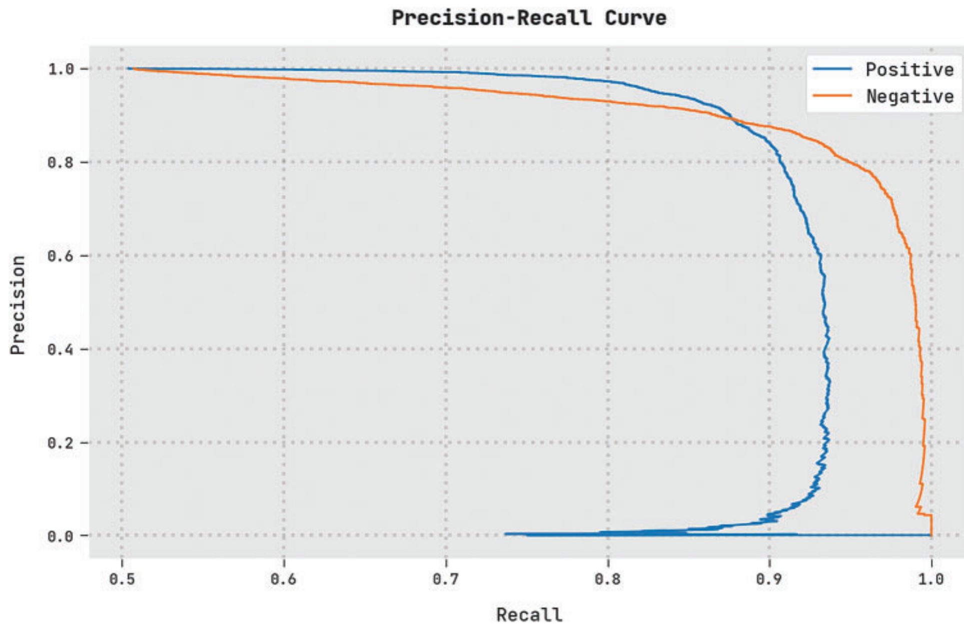

Figure 11: Precision-recall analysis of MSCADBN-MVC approach

The movie review classification results of the MSCADBN-MVC model are compared with existing ML models in Table 3 [22]. The results implied that the MSCADBN-MVC model has shown enhanced outcomes over other models. Based on $accu_y$, the MSCADBN-MVC model has gained increased $accu_y$ of 88.93% whereas the Dynamic feature selection (DFS), SVM-Chi Square, SVM-RFE, decision tree (DT), and NN models have obtained reduced $accu_y$ of 76.67%, 80.20%, 82.31%, 81.99%, and 81.51% respectively. Simultaneously, based on F_{score}, the MSCADBN-MVC approach has obtained increased F_{score} of 88.93% whereas the DFS, SVM-Chi Square, SVM-RFE, DT, and NN algorithms have attained reduced F_{score} of 78.12%, 81.71%, 82.13%, 80.19%, and 81.88% correspondingly.

Table 3: Comparative analysis of MSCADBN-MVC approach with existing algorithms

Methods	Accuracy	Precision	Recall	F-score
MSCADBN-MVC	88.93	89.06	88.97	88.93
DFS	76.67	77.34	76.96	78.12
SVM-Chi square	80.20	81.26	81.15	81.71
SVM-RFE	82.31	80.11	84.27	82.13
DT	81.99	80.70	80.65	80.19
NN	81.51	80.32	81.14	81.88

5 Conclusion

In this study, a new MSCADBN-MVC technique has been developed for the identification and classification of movie reviews on social media. The major intention of the MSCADBN-MVC technique is focused on the identification of sentiments that exist in the movie review data. Primarily, the MSCADBN-MVC model follows data pre-processing and word2vec word embedding process. For the classification of sentiments exist in the movie reviews, the ADBN model is utilized in this work. At last, the hyperparameter tuning of the ADBN model is carried out using the MSCA technique, which integrates the LF concepts into the standard SCA. In order to demonstrate the significant performance of the MSCADBN-MVC model, a wide ranging experimental analysis is performed on three different datasets. The comprehensive study highlighted the enhancements of the MSCADBN-MVC model in movie review classification process. In future, the performance of the MSCADBN-MVC model can be boosted using advanced DL models.

Funding Statement: Princess Nourah bint Abdulrahman University Researchers Supporting Project Number (PNURSP2022R281), Princess Nourah bint Abdulrahman University, Riyadh, Saudi Arabia. The authors would like to thank the Deanship of Scientific Research at Umm Al-Qura University for supporting this work by Grant Code: (22UQU4320484DSR08).

Conflicts of Interest: The authors declare that they have no conflicts of interest to report regarding the present study.

References

[1] B. L. Devi, V. V. Bai, S. Ramasubbareddy and K. Govinda, "Sentiment analysis on movie reviews," in *Emerging Research in Data Engineering Systems and Computer Communications, Advances in Intelligent Systems and Computing*, vol. 1054, Singapore: Springer, pp. 321–328, 2020.
[2] M. Y. A. Salmony and A. R. Faridi, "Bert distillation to enhance the performance of machine learning models for sentiment analysis on movie review data," in *2022 9th Int. Conf. on Computing for Sustainable Global Development (INDIACom)*, New Delhi, India, pp. 400–405, 2022.
[3] A. S. Abdullah, K. Akash, J. S. Thres and S. Selvakumar, "Sentiment analysis of movie reviews using support vector machine classifier with linear kernel function," in *Evolution in Computational Intelligence, Advances in Intelligent Systems and Computing Book Series*, vol. 1176, Singapore: Springer, pp. 345–354, 2021.
[4] M. Yasen and S. Tedmori, "Movies reviews sentiment analysis and classification," in *2019 IEEE Jordan Int. Joint Conf. on Electrical Engineering and Information Technology (JEEIT)*, Amman, Jordan, pp. 860–865, 2019.
[5] R. Maulana, P. A. Rahayuningsih, W. Irmayani, D. Saputra and W. E. Jayanti, "Improved accuracy of sentiment analysis movie review using support vector machine based information gain," *Journal of Physics: Conference Series*, vol. 1641, no. 1, pp. 012060, 2020.
[6] M. Hossen and N. R. Dev, "An improved lexicon based model for efficient sentiment analysis on movie review data," *Wireless Personal Communications*, vol. 120, no. 1, pp. 535–544, 2021.
[7] N. O. F. Daeli and A. Adiwijaya, "Sentiment analysis on movie reviews using information gain and K-nearest neighbor," *Journal of Data Science and Its Applications*, vol. 3, no. 1, pp. 1–7, 2020.
[8] N. M. Ali, M. M. A. E. Hamid and A. Youssif, "Sentiment analysis for movies reviews dataset using deep learning models," *International Journal of Data Mining & Knowledge Management Process*, vol. 9, no. 2/3, pp. 21–27, 2019.
[9] R. S. Kumar, A. F. S. Devaraj, M. Rajeswari, E. G. Julie, Y. H. Robinson *et al.*, "Exploration of sentiment analysis and legitimate artistry for opinion mining," *Multimedia Tools and Applications*, vol. 81, no. 9, pp. 11989–12004, 2022.
[10] S. Onalaja, E. Romero and B. Yun, "Aspect-based sentiment analysis of movie reviews," *SMU Data Science Review*, vol. 5, no. 3, pp. 10, 2021.
[11] A. U. Rehman, A. K. Malik, B. Raza and W. Ali, "A hybrid CNN-LSTM model for improving accuracy of movie reviews sentiment analysis," *Multimedia Tools and Applications*, vol. 78, no. 18, pp. 26597–26613, 2019.

[12] Z. Abidin, W. Destian and R. Umer, "Combining support vector machine with radial basis function kernel and information gain for sentiment analysis of movie reviews," *Journal of Physics: Conference Series*, vol. 1918, no. 4, pp. 042157, 2021.
[13] J. D. Bodapati, N. Veeranjaneyulu and S. Shaik, "Sentiment analysis from movie reviews using lstms," *Ingenierie des Systemes D'Information*, vol. 24, no. 1, pp. 125–129, 2019.
[14] K. K. Ullah, A. Rashad, M. Khan, Y. Ghadi, H. Aljuaid *et al.* "A deep neural network-based approach for sentiment analysis of movie reviews," *Complexity*, vol. 2022, pp. 1–9, 2022.
[15] N. G. Ramadhan and T. I. Ramadhan, "Analysis sentiment based on imdb aspects from movie reviews using svm," *Sinkron: Jurnal dan Penelitian Teknik Informatika*, vol. 7, no. 1, pp. 39–45, 2022.
[16] S. M. Qaisar, "Sentiment analysis of imdb movie reviews using long short-term memory," in *2nd Int. Conf. on Computer and Information Sciences (ICCIS)*, Sakaka, Saudi Arabia, pp. 1–4, 2020.
[17] D. D. Tran, T. T. S. Nguyen and T. H. C. Dao, "Sentiment analysis of movie reviews using machine learning techniques," in *Proc. of Sixth Int. Congress on Information and Communication Technology, Lecture Notes in Networks and Systems Book Series*, Singapore, Springer, vol. 235, pp. 361–369, 2022.
[18] S. H. Kumhar, M. M. Kirmani, J. Sheetlani and M. Hassan, "Word embedding generation for Urdu language using word2vec model," *Materials Today: Proceedings*, pp. S2214785320394426, 2021. https://doi.org/10.1016/j.matpr.2020.11.766
[19] J. Xie, G. Du, C. Shen, N. Chen, L. Chen *et al.,* "An end-to-end model based on improved adaptive deep belief network and its application to bearing fault diagnosis," *IEEE Access*, vol. 6, pp. 63584–63596, 2018.
[20] A. B. S. Yıldız, N. Pholdee, S. Bureerat, A. R. Yıldız and S. M. Sait, "Sine-cosine optimization algorithm for the conceptual design of automobile components," *Materials Testing*, vol. 62, no. 7, pp. 744–748, 2020.
[21] M. A. Elaziz, A. Mabrouk, A. Dahou and S. A. Chelloug, "Medical image classification utilizing ensemble learning and levy flight-based honey badger algorithm on 6G-enabled internet of things," *Computational Intelligence and Neuroscience*, vol. 2022, pp. 1–17, 2022.
[22] N. S. M. Nafis and S. Awang, "An enhanced hybrid feature selection technique using term frequency-inverse document frequency and support vector machine-recursive feature elimination for sentiment classification," *IEEE Access*, vol. 9, pp. 52177–52192, 2021.