ISSN 1330-3651 (Print), ISSN 1848-6339 (Online)
https://doi.org/10.17559/TV-20240426001497
Original scientific paper

Chaotic Sparrow Search Algorithm with Deep Learning for Event Detection and Classification in Social Media Environment

Ramya R.*, Kannan S.

Abstract: Event detection is a computational process that enables the automatic identification of major events by analyzing media data. An event refers to a significant occurrence that takes place at a specific time and location. Many researchers have focused on predicting certain events through social media analysis. These events include disease outbreaks, election results, stock market trends, the frequency of article citations, product sales, and sports competition outcomes. The noisy nature of media content requires innovative semantic techniques to ensure accurate analysis of media streams. Consequently, improving the accuracy of event detection methods by addressing the noisy characteristics of media content is critical. Recent studies have explored event detection and recognition through social media analysis. This paper introduces a Chaotic Sparrow Search Algorithm with Deep Learning for Event Detection and Classification (CSSA-DLEDC) in social media. The primary goal of the CSSA-DLEDC technique is to identify and classify the presence of events and non-events. To achieve this, the technique involves data pre-processing and the TF-IDF word embedding process. The detection of events in social media is then performed using a Deep Belief Network (DBN). Finally, the hyperparameters of the DBN model are tuned using the Chaotic Sparrow Search Algorithm (CSSA).To validate the improved performance of the CSSA-DLEDC technique, extensive experimental simulations were conducted. The results demonstrated the superior effectiveness of the CSSA-DLEDC technique in the event detection process on social media.

Keywords: applied linguisti; chaotic sparrow search algorithm; deep learning; event detection; social media

1 INTRODUCTION

Social media platforms produce high volumes of information with their developing popularity and extensive user community [1]. These data consist of a large quantity of information collected with people's opinions significant for various groups like the corporate world and scientific community to take exhilarating open challenges and acquire marketing decisions [2]. Due to the massive amount and great dynamicity of information, it is impossible for analysing them manually to extract important events: activities or incidents that occur at a specific time and are reported or discussed substantially in social media or its sentiments [3]. Therefore, several research workers concentrated on automatic mechanisms for extracting sentiments and events from social media data streams. In microblogging, an event is described as a real-world time that takes place, which produces the interest of persons and can be detected depending on temporal and spatial characteristics, i.e. the events are a particular geographical coordinate and time [4]. Any kinds of events that have been reviewed in the literature comprise natural disasters (for example, typhoons, fires, floods, and earthquakes), exhibitions, conferences, festivals, traffic incidents, weather updates, infectious diseases outbreaks, sports events, riots, terrorist attacks, and so on [5].

The two events are namely known and unknown. Known events are events whose previous knowledge about the location and time of an event is known (e.g. festivals) [6]. Unknown events are events whose previous knowledge about the position and time of events is not known (for instance, terrorist attacks). Detecting the time and location of events is required as it permits local authorities to acquire early reactions for dealing with the events and notify the people. Recognizing events also increases the level of people's awareness and provides a snapshot of real-time activities worldwide [7]. The inspiration is referred to event forecasts, which can be done automatically from an openly accessible and useful source like Twitter. This process has numerous applications, for instance, in novel interventions to automatically detect and target upcoming events, control social crises and later make for relief, utilize in recommendation systems, and rank forthcoming events by their significance [8]. Recently, several researches are implemented facing event recognition and detection by exploiting social media reviews. Deep learning (DL) techniques are employed in current natural language processing (NLP), speech and image identification, and intelligent gamification and can be utilized in the domain of short-range traffic and disaster forecast [9]. However, these methods concentrate on the information and particular kinds of events; our techniques consider many tweets as features and events in general [10].

This study presents a Chaotic Sparrow Search Algorithm with Deep Learning for Event Detection and Classification (CSSA-DLEDC) in Social Media. The goal of the CSSA-DLEDC technique is to recognize and classify the presence of events/non-events. To achieve this, the CSSA-DLEDC technique primarily undergoes data pre-processing and the TF-IDF word embedding process. For the detection of events in social media, the CSSA-DLEDC technique uses a deep belief network (DBN) approach. At last, the hyperparameter tuning of the DBN model takes place utilizing the Chaotic Sparrow Search Algorithm (CSSA). To demonstrate the improved detection outcomes of the CSSA-DLEDC technique, a widespread experimental value was carried out.

2 RELATED WORKS

In [11], the authors proposed an optimum DL-based flood prediction method with big data analytics (ODLFF-BDA) that depends on Twitter statistics. After pre-processing, a BERT model is used for generating context embedding from Twitter. In addition, a GRU with Multilayer CNN (MLCNN) is implemented for extracting local data. Lastly, an Equilibrium Optimizer (EO) is used for fine tuning the hyperparameter of the GRU and MLCNN techniques. Yavari et al. [12] introduced a technique that pre-processed the Tweets in successive fixed-length time windows. Then, the Tweets are classified by the distance-dependent incremental clustering and the non-negative matrix factorization analysis. Lastly, a description of the event is developed by some recurrent words in all the clusters. Essien et al. [13] present a DL urban traffic prediction method which combines the data extracted from Twitter with traffic and weather statistics. The predictive method adopted a deep BiLSTM-SAE structure for multi-step traffic flow forecasting and was trained by the tweets, weather, and traffic data.

Uthirapathy and Sandanam [14] introduced a new architecture to analyze data diffusion and opinion evolution. A fuzzy c-means clustering, forest fire algorithm, and cuckoo search are used to present a model for forecasting data diffusion and opinion analysis in social networking platforms. In [15], a new DNN-based architecture named event prediction with feedback model (EPFM) is introduced. Particularly, the proposed method exploits a feedback model based on emerging event recognition for enhancing the outcome. The feedback model ensembles 3 outlier recognition methods and returns a list of new events. Then, a few events are selected by analysts to flow into the finetuning method for updating the prediction method. Aum et al. [16] inspected artist-fan interaction on social networking platforms as the potential indicator.

In [17] is proposed an effective multi-modal technique that identifies fake images of microblogging platforms. The presented architecture exploits CNN for images and sentence transformers for the analysis of texts. Chaudhary et al. [18] introduced a big data concept to analyze and process information to forecast consumer behaviours on social media platforms. Consumer behavior was analyzed on social networking platforms based on specific criteria and parameters. Mathematical modeling using ML was developed for predicting consumer behaviors on social media platforms.

3 THE PROPOSED METHOD

In this study, we have introduced an automated event detection approach named CSSA-DLEDC technique in Social Media. The main intention of the CSSA-DLEDC technique is to recognize and classify the presence of events/non-events.

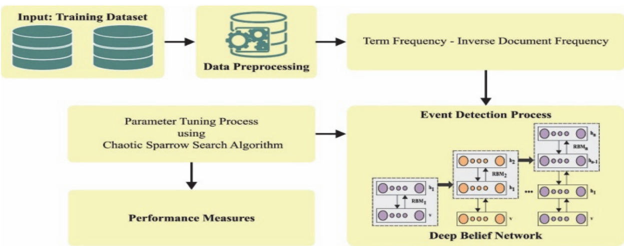

Figure 1 Overall flow of CSSA-DLEDC algorithm

To achieve this, the CSSA-DLEDC technique involves data pre-processing, TF-IDF word embedding process, DBN classification, and CSSA-based hyperparameter tuning. Fig. 1 depicts the entire flow of the CSSA-DLEDC algorithm.

3.1 Data Pre-Processing

Generally, Tweets are not in a usable format; for instance, they involve emoticons, characters, or symbols. Thus, it is essential to format them in a suitable procedure that is capable to extract meaningful ideas from them [19]. Tokenization can be described as a type of lexical analysis that breaks a stream of text up into symbols, phrases, words, or other meaningful components named tokens. Tokenization is a process used to separate the text string into a list of separate words. The tokenization is used for the transformation of NLP in preprocessing. For the pre-processing step, different techniques were introduced and exploited for data cleaning. The data pre-processing is given in the following:

Elimination of Non-English Tweets: Social media platforms have a worldwide reach, and tweets can be written in many languages. To ensure consistency and facilitate analysis, tweets that are not in English are excluded. The process involves utilizing language identification algorithms to determine the language of each tweet and subsequently excluding those that are not written in English. Username and external link removal: Tweets frequently include usernames (indicated by '@') and external URLs. These factors are often irrelevant to the process of content analysis and can create unwanted interference. Hence, any content that starts with '@' and URLs (often commencing with 'http' or 'www') is detected and eliminated.

Hashtag Removal: Hashtags, indicated by the symbol '#' before a word or phrase, are frequently employed to label tweets with particular subjects. Although they might be beneficial for classification purposes, they are eliminated at this stage to concentrate on the actual text of the tweet. Hashtags are removed by detecting the '#' sign and deleting the complete hashtag content. Stopword elimination refers to the process of removing frequent words, known as stopwords, from a text. Stopwords, such as 'and', 'the', and 'is', are words that do not add substantial meaning to the text. Eliminating these factors aids in decreasing the dimensionality of the data. A predetermined set of stopwords is employed to exclude certain words from the tweets.

Emoticon Elimination: Emoticons and emojis express emotions but might disrupt the processing of text. These emoticons are detected and eliminated by utilizing regular expressions that correspond to typical emoticon patterns.

3.2 TF-IDF Approach

Feature extraction in data mining is a method that contains phases for decreasing the count of data accessible to define massive datasets [20]. If examining the mood of a difficult text, major issues rise from the variable counts. Generally, to analyze difficult and huge text, huge counts of memory and processing power are essential. It creates the classification method more valuable to trained instances and leads to worse generalized with novel instances. The researchers mentioned that in applications containing several features, the extracted feature is related to size reduction.

It establishes a novel Element Extract system to remove elements in a provided text. Utilizing context words for representing or extracting meaning in a big body of text is identified as "counter vectorization". Many mutual limitations are executed on all the words for determining the feasible word correspondences. Text data is also employed for deriving suitable functions like Term Frequency (TF) and Inverse Document Frequency (IDF). Eq. (1) and Eq. (2) are utilized by TF and IDF for determining the frequency that a phrase performs in a document.

$$\text{Term frequency}= \\ =\frac{\text{No. of times term}(\text{t})\text{appears in document}}{\text{Total no. of terms in a document}} \tag{1}$$

$$\text{Inverse document frequency}= \\ =\log\frac{\text{Total no. of documents}}{\text{No. of documents term}(\text{t})} \tag{2}$$

The TF-IDF approach can be employed for extracting suitable data in tweets. During this case, 6288 features can be extracted as document-term matrix TF-IDF features.

3.3 Event Detection using DBN Model

In this study, the DBN model is applied to the classification of events. The DBN comprises a single MLP layer and two RBM layers [21]. In DBN, there is no connectivity availability between the visible and hidden neurons. The input to the initial RBM layer of DBN is the segmented index S_p and the feature vector F_p together with the ground truth fed into the visible layer. Consequently, the outcome from the initial layer is fed into the second one and its resultant outcomes are eventually used in the MLP.

The feature vectors are regarded as an input to the hidden units of the initial RBM and it is given as follows:

$$U^1=\left\{U_1^1, U_2^1, ..., U_m^1, ..., U_{10}^1\right\}; 1\le m\le 10 \tag{3}$$

$$V^1=\left\{V_1^1, V_2^1, ..., V_l^1, ..., V_n\right\}; l\le 1\le n \tag{4}$$

where U_m^1 denotes the m-rh visible neuron in the initial RBM, V_l^1 indicates the l-th hidden neurons and n denotes the overall amount of the hidden neurons. The visible and hidden layers are encompassed by neurons and all the neurons have their own bias. Assume x and y denote the bias and it is represented as follows:

$$x^1=\left\{x_1^1, x_2^1, ..., x_m^1, ..., x_{10}^1\right\} \tag{5}$$

$$y^1=\left\{y_1^1, y_2^1, y_l^1, ..., y_n^1\right\} \tag{6}$$

Now, x_m^1 denotes the bias with respect to m-rh m$^{\text{rh}}$ visible neurons and the bias with respect to l-th hidden neurons is indicated by y_l^1. The weight in the initial RBM is represented as follows:

$$W^1=\left\{W_{ml}^1\right\}; 1\le m\le 10; l\le 1\le n \tag{7}$$

In Eq. (7), W_{ml}^1 shows the weights between m-th visible and l-th hidden neurons. Therefore, the output of the hidden layer in the first RBM is formulated based on the weight corresponding to a single visible neuron.

$$V_l^1=\eta\left[y_l^1+\sum_m U_m^1 W_{ml}^1\right] \tag{8}$$

In Eq. (8), η shows the activation parameter, and therefore, the outcome obtained in the first RBM is calculated by the following expression:

$$V^1=\left\{V_l^1\right\}; l\le 1\le n \tag{9}$$

The outcome of the initial RBM is exploited to the visible layer of the next RBM. Therefore, the quantity of visible neurons is the same as the quantity of hidden neurons in the first RBM,

$$U^2=\left\{U_1^2, U_2^2, ..., U_n^2\right\}=\left\{V_l^1\right\}; l\le 1\le n \tag{10}$$

In Eq. (10), V_l^1 represents the output representation of the first RBM. The vector form of the HL in the next RBM is formulated as follows:

$$V^2=\left\{V_1^2, V_2^2, ..., V_l^2, ..., V_n^2\right\}; l\le 1\le n \tag{11}$$

The weight vector of the next RBM is given below,

$$W^2=\left\{W_{ll}^2\right\}; l\le 1\le q \tag{12}$$

In Eq. (12), the weight between l-th visible and hidden neurons is characterized by W_{ll}^2. The outcome of a hidden neuron is provided as:

$$V_l^2=\eta\left[y_l^2+\sum_m U_m^2 W_{ll}^2\right]\forall U_m^2=V_l^1 \tag{13}$$

In Eq. (13), y_l^2 refers to the bias corresponding to the l-th neuron and the outcome of HL is defined below:

$$V^2=\left\{V_l^2\right\}; l\le 1\le q \tag{14}$$

The abovementioned formula generates the input to the MLP and it can be mathematically modelled as follows:

$$P=\left\{P_1, P_2, ..., P_l, ..., P_n\right\}=\left\{V_l^2\right\}; l\le 1\le n \tag{15}$$

In Eq. (15), n shows the total amount of neurons in MLP is expressed as follows:

$$\lambda=\left\{\lambda_1, \lambda_2, ..., \lambda_{kk}, ..., \lambda_{mm}\right\};\ 1 \le kk \le mm \tag{16}$$

In Eq. (16), mm and kk indicate the total amount of hidden neurons and kk-rh hidden neurons, correspondingly.

$$Out=\left\{Out_1, \mathrm{Out}_2, ..., \mathrm{Out}_{aa}, ..., \mathrm{Out}_{bb}\right\};\ 1 \le aa \le bb \tag{17}$$

In Eq. (17), bb denotes the overall amount of neurons in the output layer. Therefore, the resulting vector is defined according to W^{Hh} and it is calculated as follows:

$$Out_{aa}=\sum_{kk=1}^{mm} W_{kkbb}^{Hh} \cdot Our_{aa} \tag{18}$$

In Eq. (18), W_{kkbb}^{Hh} indicates the weight amongst bb-th output neuron and the kk-th hidden neuron. Furthermore, the outcome of HL is represented by Out_{aa}. Therefore, the pixel-wise classified output obtained from DBN is represented by K_p with a size of [256 × 256 × 9].

3.3 Hyperparameter Tuning using CSSA

Chaotic Sparrow Search (CSSA) is an optimization approach inspired by sparrow foraging and anti-predation. It uses chaotic maps to improve convergence and exploration. CSSA optimises DBN hyper parameters such as layer count, neuron count per layer, learning rate, batch size, and training epochs.

CSSA creates a sparrow population using randomly supplied hyperparameters. Every hyperparameter group trains a Deep Belief Network (DBN), and a validation dataset evaluates its performance. Chaotic maps create random sequences that diversify sparrow populations and avoid early convergence. The system uses producer sparrows to discover food (optimal solutions) in promising places, while scrounger sparrows exploit their finds. To avoid predators, sparrows might hunt in new areas. This process continues until convergence or a halting condition is met.

Finally, the hyperparameters related to the DBN model can be chosen by the CSSA. The SSA is a new SI optimization technique stimulated by the sparrow foraging process developed by Xue et al., to search for the optimum solution [22]. Nonetheless, SSA might easily get stuck into the local optima which decreases the optimization efficacy. Based on SSA, the CSSA technique presents Tent chaotic search and Gaussian mutation to resolve problems of SSA. Tent chaotic sequence has unstable period points and a small period, for which the parameter $\mathrm{rand}(0, 1) \times \frac{1}{N_T}$ is used to enhance the global search ability of the algorithm and improve the quality of the initial solution without damaging the features of chaotic variables. $\mathrm{rand}(0, 1)$ shows a random integer within [0, 1], and N_T refers to the amount of particles in the chaotic sequence. During the population initialization, the Tent chaotic sequence was presented for initializing the population, and ND-dimensional vector is produced. All the components are carried to the value of the original problem space using the following expression:

$$N_{\mathrm{new}}^d=d_{\min}+\left(d_{\max}-d_{\min}\right) B_i \tag{19}$$

In Eq. (19), $d_{\max}$ and $d_{\min}$ correspondingly denote the maximal and minimal values of the d-th dimensional parameter N_{new}^d. Next, according to Eq. (20), chaotic disturbance is performed.

$$N_{\mathrm{new}}=\frac{\left(N+N_{\mathrm{new}}\right)}{2} \tag{20}$$

In Eq. (20), N shows the individual that requires chaotic disturbance, N_{new} represents the individual after chaotic disturbance, and N_{new} denotes the produced chaotic disturbance.

The Gaussian difference is based on the Gaussian distribution. The values of the original parameter were exchanged by the uniform distribution random integer with mean μ and standard deviation of σ^2.

$$\mathrm{mutation}(x)=x \times (1+N(0, 1)) \tag{21}$$

In Eq. (21), mutation(x) indicates the value after Gaussian variation, x refers to the values of an original parameter, and N(0, 1) shows the uniformly distributed random integer with mean 0 and variance of 1.

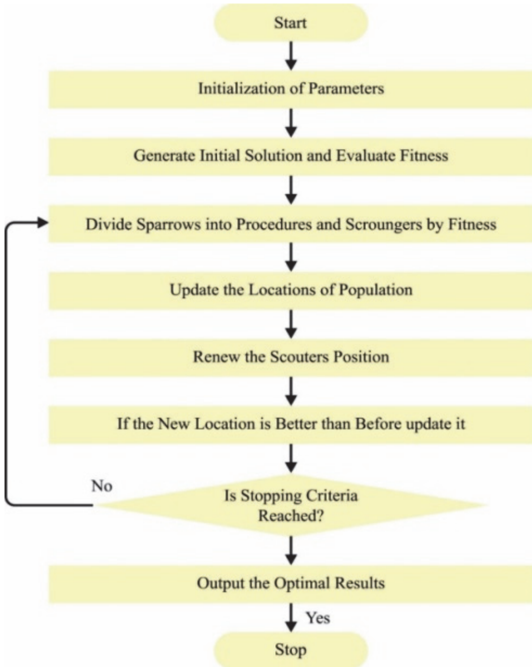

Figure 2 Flowchart of SSA

The average fitness values A and B of the sparrow population are re-evaluated after one iteration. If A is less than B, then an individual is in the center location. According to Eq. (21), the Gaussian mutation is implemented.

When the mutated individual is better, then the original value can be replaced; or else, it remains the same. Especially, if A is greater than B, then individuals are at the edge, and the chaotic perturbation was carried out based on Eq. (19) and Eq. (20). When the performance of disturbed individuals is more effective, then it is replaced with the disturbed individuals; or else, the original value remains the same. Fig. 2 depicts the flowchart of SSA.

The fitness choice is a key feature of the CSSA method. An encoded performance has been deployed for developing a better solution for candidate outcomes. At present, the accuracy value is the major condition employed to design a FF.

$$\text{Fitness} = \max(P) \tag{22}$$

$$P = \frac{TP}{TP + FP} \tag{23}$$

In which, FP and TP denote the false and true positive values.

4 RESULT AND DISCUSSION

In this section, the experimental validation of the CSSA-DLEDC technique is tested using a dataset available in (Kalyanam et al. 2016). It comprises Tweets taken from familiar news media such as CNN, BreakNews, and BBC on Twitter for a duration of 6 months. The results are examined a number of days before the event occurrence.

Table 1 $F1_{\text{score}}$ outcome of CSSA-DLEDC algorithm under ten runs

$F1_{\text{score}}$							
	Number of days before events occurrence						
No. of Runs	1	2 - 7	8 - 7	15 - 21	22 - 28	29 - 35	36 +
Run 1	98.58	96.56	94.33	80.18	60.34	31.04	31.25
Run 2	98.13	96.60	94.40	80.18	60.24	37.96	36.41
Run 3	98.41	96.18	94.57	80.23	60.46	37.14	30.56
Run 4	98.32	96.36	94.59	80.36	60.58	31.92	35.20
Run 5	98.39	96.22	94.19	80.12	60.33	38.66	33.85
Run 6	98.41	96.18	94.64	80.40	60.23	33.67	35.45
Run 7	98.59	96.07	94.47	80.43	60.52	30.97	38.52
Run 8	98.13	96.19	94.01	80.67	60.17	36.35	36.32
Run 9	98.51	96.53	94.49	80.19	60.34	37.63	35.44
Run 10	98.32	96.49	94.30	80.47	60.57	33.92	35.80
Average	98.38	96.34	94.40	80.32	60.38	34.93	34.88

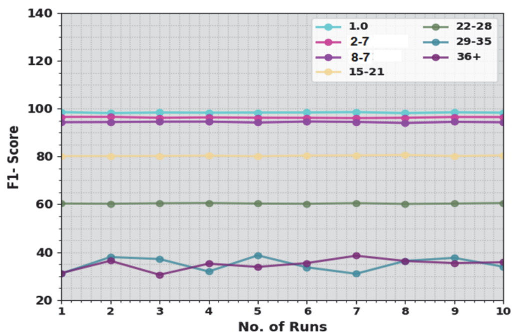

Figure 3 $F1_{\text{score}}$ outcome of CSSA-DLEDC algorithm under ten runs

Tab. 1 and Fig. 3 represent the $F1_{\text{score}}$ results of the CSSA-DLEDC technique under ten runs. The results show that the CSSA-DLEDC technique exhibits effectual $F1_{\text{score}}$ values under all days and runs. For instance, with run-1, the CSSA-DLEDC technique offers $F1_{\text{score}}$ of 98.58%, 96.56%, 94.33%, 80.18%, 60.34%, 31.04%, and 31.25% under 1, 2 - 7, 8 - 7, 15 - 21, 22 - 28, 29 - 35, and 36 + days respectively. In addition, with run-10, the CSSA-DLEDC approach attains $F1_{\text{score}}$ of 98.32%, 96.49%, 94.30%, 80.47%, 60.57%, 33.92%, and 35.80% under 1, 2 - 7, 8 - 7, 15 - 21, 22 - 28, 29 - 35, and 36 + days correspondingly.

Tab. 2 and Fig. 4 represent a comparative study of the CSSA-DLEDC technique with recent models in terms of $F1_{\text{score}}$ [12]. The comparison results ensured that the CSSA-DLEDC technique reaches improved $F1_{\text{score}}$ values over other models.

Table 2 $F1_{\text{score}}$ outcome of the CSSA-DLEDC algorithm with other systems

$F1_{\text{score}}$					
Events occurrence	CSSA-DLEDC	EPSNTTMA	MABED	FSD	Embed2etect
36 +	34.88	1.514	0.015	0.015	0.015
29 - 35	34.92	13.739	7.626	0.015	3.806
22 - 28	60.37	32.077	11.447	0.749	8.390
15 - 21	80.32	64.933	22.908	5.334	35.898
8 - 7	94.40	76.395	53.472	9.919	38.954
2 - 7	96.34	82.507	64.169	37.426	47.359
1	98.38	86.328	74.866	84.036	61.877

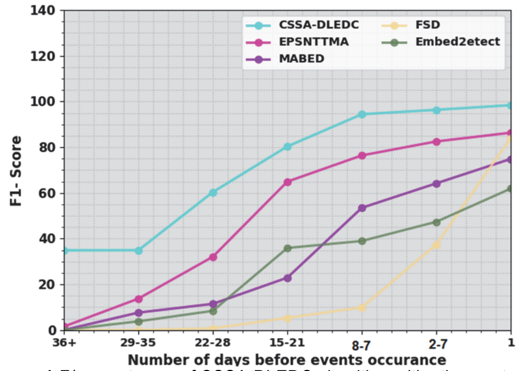

Figure 4 $F1_{\text{score}}$ outcome of CSSA-DLEDC algorithm with other systems

Tab. 3 and Fig. 5 demonstrate the $prec_n$ outcome of the CSSA-DLEDC system on ten runs. The outcome depicted that the CSSA-DLEDC technique displays effective $prec_n$ values under all days and runs.

Table 3 $prec_n$ outcome of CSSA-DLEDC algorithm under ten runs

Precision							
No. of Runs	Number of days before events occurrence						
	1	2 - 7	8 - 7	15 - 21	22 - 28	29 - 35	36 +
Run 1	97.02	97.94	97.33	87.57	69.31	48.84	39.00
Run 2	97.19	97.92	97.33	87.40	69.26	49.47	39.04
Run 3	97.17	97.95	97.22	88.38	69.27	48.64	39.12
Run 4	97.00	97.99	97.33	88.05	69.03	48.05	39.10
Run 5	97.13	97.98	97.05	87.41	69.37	48.30	39.07
Run 6	97.33	97.95	97.11	87.85	69.26	47.69	39.40
Run 7	97.05	97.97	97.25	87.36	69.30	47.84	39.50
Run 8	97.07	97.96	97.00	88.48	69.42	48.63	39.27
Run 9	97.30	97.96	97.50	87.00	69.16	47.44	39.20
Run 10	97.02	97.93	97.50	87.55	69.13	48.48	39.48
Average	97.13	97.96	97.26	87.71	69.25	48.34	39.22

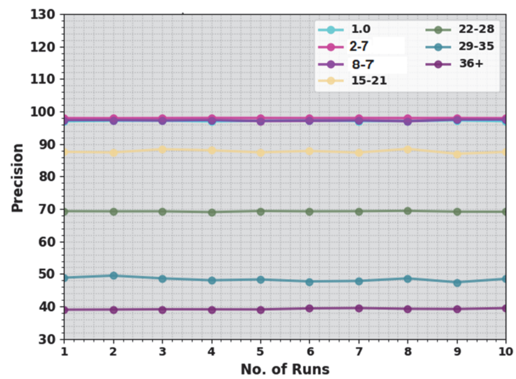

Figure 5 $prec_n$ outcome of CSSA-DLEDC algorithm under ten runs

Tab. 4 and Fig. 6 define the comparative analysis of the CSSA-DLEDC algorithm with recent models with respect to $prec_n$. The comparison outcomes make sure that the CSSA-DLEDC approach achieves higher $prec_n$ values over other systems. For instance, on 2 - 7 days, the CSSA-DLEDC technique obtains a higher $prec_n$ of 97.96%. On the other hand, the EPSNTTMA, MABED, FSD, and Embed2etect systems achieve minimal $prec_n$ of 82.206%, 32.202%, 63.560%, and 44.067% correspondingly. In addition, on 1 day, the CSSA-DLEDC technique obtains a higher $prec_n$ of 97.13%. On the other hand, the EPSNTTMA, MABED, FSD, and Embed2etect approaches achieve lesser $\mathrm{prec_n}$ of 88.139%, 99.986%, 77.968%, and 63.560% respectively.

Table 4 *$prec_n$* outcome of CSSA-DLEDC algorithm with other systems

Precision					
Number of days before events occurrence	CSSA-DLEDC	EPSNTTMA	MABED	FSD	Embed 2etect
36 +	39.22	1.691	0.851	0.844	0.004
29 - 35	48.34	12.709	0.851	4.234	0.844
22 - 28	69.25	32.202	0.851	11.014	6.776
15 - 21	87.71	69.493	6.776	20.337	37.287
8 - 7	97.26	81.358	5.081	53.390	38.135
2 - 7	97.96	82.206	32.202	63.560	44.067
1	97.13	88.139	88.986	77.968	63.560

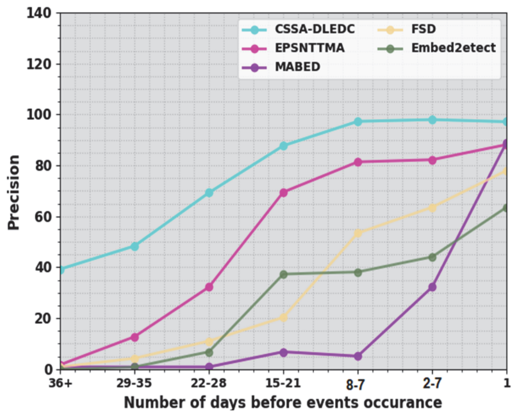

Figure 6 *$prec_n$* outcome of CSSA-DLEDC algorithm with other systems

Tab. 5 and Fig. 7 define the $\mathrm{reca_l}$ outcomes of the CSSA-DLEDC approach under ten runs. The results show that the CSSA-DLEDC system reveals efficient $\mathrm{reca_l}$ values under all days and runs.

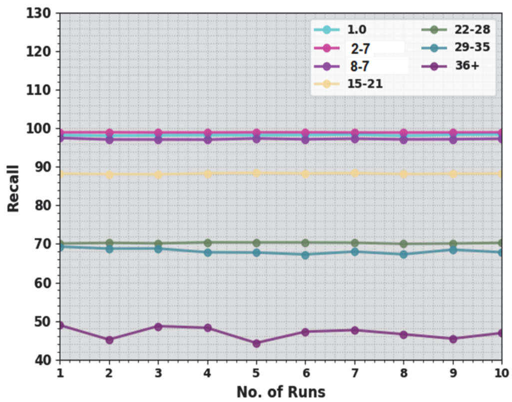

Figure 7 *$Reca_l$* outcome of CSSA-DLEDC algorithm under ten runs

Table 5 *$Reca_l$* outcome of CSSA-DLEDC algorithm under ten runs

Recall							
	Number of days before events occurrence						
No. of Runs	1	2 - 7	8 - 7	15 - 21	22 - 28	29 - 35	36 +
Run 1	98.27	98.94	97.49	88.28	70.11	69.29	48.94
Run 2	98.08	98.96	97.10	88.12	70.29	68.79	45.18
Run 3	98.21	98.92	97.10	88.06	70.15	68.82	48.67
Run 4	98.26	98.91	97.07	88.34	70.41	67.83	48.24
Run 5	98.26	98.95	97.37	88.44	70.39	67.77	44.30
Run 6	98.27	98.95	97.16	88.29	70.38	67.27	47.23
Run 7	98.43	98.90	97.31	88.39	70.32	68.00	47.65
Run 8	98.06	98.90	97.13	88.15	70.02	67.30	46.57
Run 9	98.37	98.92	97.17	88.26	70.12	68.53	45.43
Run 10	98.44	98.95	97.30	88.28	70.29	67.83	46.88
Average	98.27	98.93	97.22	88.26	70.25	68.14	46.91

Tab. 6 and Fig. 8 demonstrate a comparative analysis of the CSSA-DLEDC technique with recent models in terms of *$reca_l$*. The comparison outcome makes sure that the CSSA-DLEDC algorithm reaches improved *$reca_l$* values over other models.

Table 6 *$reca_l$* outcome of CSSA-DLEDC algorithm with other systems

Recall					
Number of days before events occurrence	CSSA-DLEDC	EPSNTTMA	MABED	FSD	Embed2etect
36 +	46.91	5.855	0.073	2.891	2.150
29 - 35	68.14	12.524	2.891	6.596	2.891
22 - 28	70.25	31.791	2.150	11.783	10.301
15 - 21	88.26	70.323	8.819	22.898	36.978
8 - 7	97.22	80.697	9.560	56.985	41.424
2 - 7	98.93	84.402	38.460	65.136	48.834
1	98.27	91.071	89.589	76.251	62.913

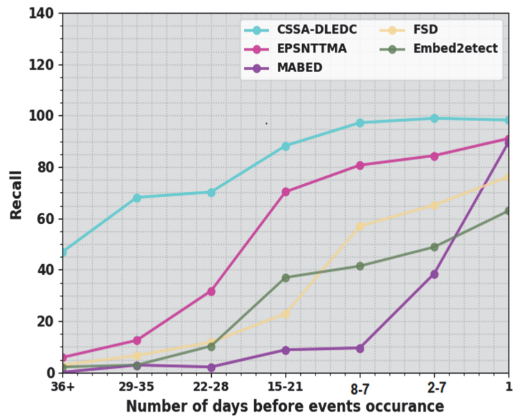

Figure 8 *$prec_n$* outcome of CSSA-DLEDC algorithm with other systems

These results stated the better performance of the CSSA-DLEDC technique in the event detection process.

5 CONCLUSION

In this study, we have introduced an automated event detection approach named CSSA-DLEDC system in Social Media. The main intention of the CSSA-DLEDC technique is to recognize and classify the presence of events/non-events. To achieve this, the CSSA-DLEDC technique involves data pre-processing, TF-IDF word embedding process, DBN classification, and CSSA-based hyperparameter tuning. For the detection of events in social media, the CSSA-DLEDC technique uses the DBN model. At last, the hyperparameter tuning of the DBN model takes place utilizing the CSSA, which helps in accomplishing an enhanced event detection rate. To demonstrate the improved detection outcomes of the CSSA-DLEDC technique, a comprehensive experimental outcome was carried out. The simulation values highlighted the promising performance of the CSSA-DLEDC technique in the event detection process in social media.

6 REFERENCES

[1] Yavari, A., & Hassanpour, H. (2023). Election Prediction Based on Messages Feature Analysis in Twitter Social Network. *International Journal of Engineering, 36*(6), 1179-1184. https://doi.org/10.5829/ije.2023.36.06c.16

[2] Park, Y. E. (2022). Developing a COVID-19 crisis management strategy using news media and social media in big data analytics. *Social Science Computer Review, 40*(6), 1358-1375. https://doi.org/10.1177/08944393211007314

[3] Hagemann, L. & Abramova, O. (2023). Sentiment, we-talk and engagement on social media: Insights from Twitter data mining on the US presidential elections 2020. *Internet Research, 33*(6), 2058-2085. https://doi.org/10.1108/INTR-12-2021-0885

[4] Ali, H., Farman, H., Yar, H., Khan, Z., Habib, S., & Ammar, A. (2022). Deep learning-based election results prediction using Twitter activity. *Soft Computing, 26*(16), 7535-7543. https://doi.org/10.1007/s00500-021-06569-5

[5] Sampath, K. K. & Supriya, M. (2023). Traffic Prediction in Indian Cities from Twitter Data Using Deep Learning and Word Embedding Models. *International Conference on Multi-disciplinary Trends in Artificial Intelligence, Cham: Springer Nature Switzerland,* 671-682. https://doi.org/10.1007/978-3-031-36402-0_62

[6] Yavari, A., Hassanpour, H., Rahimpour Cami, B., & Mahdavi, M. (2022). Election prediction based on sentiment analysis using twitter data. *International Journal of Engineering, 35*(2), 372-379. https://doi.org/10.5829/ije.2022.35.02b.13

[7] Azhar, A., Rubab, S., Khan, M. M., Bangash, Y. A., Alshehri, M. D., Illahi, F., & Bashir, A. K. (2023). Detection and prediction of traffic accidents using deep learning techniques. *Cluster Computing, 26*(1), 477-493. https://doi.org/10.1007/s10586-021-03502-1

[8] Yao, W. & Qian, S. (2021). From Twitter to traffic predictor: Next-day morning traffic prediction using social media data. *Transportation research part C: emerging technologies, 124*, 102938. https://doi.org/10.1016/j.trc.2020.102938

[9] Hodorog, A., Petri, I., & Rezgui, Y., (2022). Machine learning and Natural Language Processing of social media data for event detection in smart cities. *Sustainable Cities and Society, 85*, 104026. https://doi.org/10.1016/j.scs.2022.104026

[10] Duraisamy, P., Duraisamy, M., Periyanayaki, M., & Natarajan, Y. (2023). Predicting Disaster Tweets using Enhanced BERT Model. *2023 7th International Conference on Intelligent Computing and Control Systems (ICICCS),* 1745-1749. https://doi.org/10.1109/ICICCS56967.2023.10142660

[11] Indra, G. & Duraipandian, N. (2023). Modeling of Optimal Deep Learning Based Flood Forecasting Model Using Twitter Data. *Intelligent Automation and Soft Computing, 35*(2), 1455-1470. https://doi.org/10.32604/iasc.2023.027703

[12] Yavari, A., Hassanpour, H., Rahimpour Cami, B., & Mahdavi, M. (2022). Event prediction in social network through Twitter messages analysis. *Social Network Analysis and Mining, 12*(1), 78. https://doi.org/10.1007/s13278-022-00911-x

[13] Essien, A., Petrounias, I., Sampaio, P., & Sampaio, S. (2021). A deep-learning model for urban traffic flow prediction with traffic events mined from twitter. *World Wide Web, 24*(4), 1345-1368. https://doi.org/10.1007/s11280-020-00800-3

[14] Uthirapathy, S. E. & Sandanam, D. (2023). Predicting opinion evolution based on information diffusion in social networks using a hybrid fuzzy based approach. *International Journal of Information Technology, 15*(1), 87-100. https://doi.org/10.1007/s41870-022-01109-2

[15] Ma, W., Hu, X., Chen, C., Wen, S., Choo, K. R., & Xiang, Y. (2022). Social media event prediction using DNN with feedback mechanism. *ACM Transactions on Management Information Systems (TMIS), 13*(3), 1-24. https://doi.org/10.1145/3522759

[16] Aum, J., Kim, J., & Park, E. (2023). Can we predict the Billboard music chart winner? Machine learning prediction based on Twitter artist-fan interactions. *Behaviour & Information Technology, 42*(6), 775-788. https://doi.org/10.1080/0144929X.2022.2042737

[17] Singh, B. & Sharma, D. K. (2022). Predicting image credibility in fake news over social media using multi-modal approach. *Neural Computing and Applications, 34*(24), 21503-21517. https://doi.org/10.1007/s00521-021-06086-4

[18] Chaudhary, K., Alam, M., Al-Rakhami, M. S., & Gumaei, A. (2021). Machine learning-based mathematical modelling for prediction of social media consumer behavior using big data analytics. *Journal of Big Data, 8*(1), 1-20. https://doi.org/10.1186/s40537-021-00466-2

[19] Vasiu, M. A. & Potolea, R. (2020). Enhancing tokenization by embedding romanian language specific morphology. *2020 IEEE 16th International Conference on Intelligent Computer Communication and Processing (ICCP),* 243-250. https://doi.org/10.1109/ICCP51029.2020.9266140

[20] Saranya, S. & Usha, G. (2023). A Machine Learning-Based Technique with IntelligentWordNet Lemmatize for Twitter Sentiment Analysis. *Intelligent Automation & Soft Computing, 36*(1), 339-352. https://doi.org/10.32604/iasc.2023.031987

[21] Subba Reddy, T., Krishna Reddy, V. V., Vijaya Kumar Reddy, R., Kolli, C. S., Sitharamulu, V., & Chandrababu, M. (2023). SHBO-based U-Net for image segmentation and FSHBO-enabled DBN for classification using hyperspectral image. *The Imaging Science Journal*, 1-20. https://doi.org/10.1080/13682199.2023.2208927

[22] Wen, J. & Wang, Z. (2023). Short-Term Power Load Forecasting with Hybrid TPA-BiLSTM Prediction Model Based on CSSA. *CMES-Computer Modeling in Engineering & Sciences, 136*(1), 749-765. https://doi.org/10.32604/cmes.2023.023865

Contact information:

Ramya R., Assistant Professor
(Corresponding author)
Department of Computer Science and Engineering,
A.V.C. College of Engineering, Mannampandal,
Mayiladuthurai 609305, India
E-mail: ramyar_cse@outlook.com

Kannan S., PhD Professor
Department of Electronics and Communication Engineering,
Kings College of Engineering, Pudukottai, India