The Journal of Supercomputing
https://doi.org/10.1007/s11227-021-03709-4

Intelligent sentinet-based lexicon for context-aware sentiment analysis: optimized neural network for sentiment classification on social media

K. E. Naresh Kumar[1] · V. Uma[1]

Accepted: 23 February 2021

Abstract

In the modern era, lack of adequate training data requires lexicon-based models. The lexicon scoring model was extensively deployed as an effective and convenient substitute by the majority of practitioners and researchers. Usually, the entire sentiment of the document is portrayed by leading polarity (i.e., negative or positive) among the indicators. The efficiency of the conventional lexicons is however quite imperfect when employed to novel issues. This paper intends to propose a new "Intelligent Senti-net based lexicon generation method," which guarantees the classification of sentiments in social media. The proposed sentiment classification model is performed through certain steps (i) pre-processing (ii) keyword extraction (iii) holoentropy-based lexicon construction (iv) feature extraction (semantic similarity) (v) classification (vi) feedback process and (vii) sentiment classification from Int SentiNet. For classification purposes, neural network (NN) is used. To make the classification more accurate, the training of NN is carried out using a new Improved Sealion (SLnO) algorithm named Self-Improved SLnO via optimizing the weights. In the end, simulation is done to validate the enhancement of the presented scheme over traditional schemes.

Keywords Sentiment · Holoentropy · Feedback · Feature sets · Neural network · SI-SLnO algorithm

Abbreviations

NN	Neural network
DNN	Deep neural networks
CNN	Convolution NN
TGFS	Twitter generic feature set
SLnO	Sea lion optimization

✉ K. E. Naresh Kumar
nareshforgovt@gmail.com

[1] Department of Computer Science, Pondicherry University, Puducherry, India

SI-SLnO	Self-improved SLnO
SA	Sentiment analysis
SVM	Support vector machine
GA	Genetic algorithm
LSTM	Long short-term memory
FOR	False omission rate
FDR	False discovery rate
FNR	False-negative rate
FPR	False-positive rate

1 Introduction

In the past few years, social media has experienced tremendous growth. One of the most trending activities on social media websites is posting messages [1, 2]. Due to the immense count of user-generated content, social media has become the biggest data source of public opinion. These data sources are invaluable for business analytics and intelligence as opinion is the major predictor of human behavior. Despite the incredible effort in manipulating customers via marketing operation on social media [3, 4], the extraction of public opinion remains at the infant stage. Both researchers and business practitioners are yet searching for more effective tools to derive value from social media data including pictures, videos, profiles, and textual content as well. Accordingly, text data have great prospects to maintain business information in real-time scenarios [5, 6], if proper models are employed.

Sentiment analysis aids to accomplish the targets such as "predicting the market business intelligence, customer satisfaction, public awareness, and opinion about products" [7, 8]. Accordingly, SA is categorized into "document-level, sentence-level, and entity-level." The sentiments contained in a document are predicted by document-level analysis [9]. Different phases are included in extracting the views from user-made content; since the text arrives from diverse sources in varied formats. Data acquirement and pre-processing are the main tasks in SA [10, 11].

Sentiment lexicons via the word or phrases labeling with their sentiment polarization are more significant for sentiment analysis. Further, the lexicon-based models include evaluating orientations from the semantic polarities of phrases or words in documents. Recently, researchers have introduced many approaches including linguistic rule-based models, corpora-based models, and dictionary-based models for automated construction of sentiment lexicons. Many domains existed on the internet and high-quality domain-specific sentiment lexicons could enhance the efficiency of fine-grained sentiment analysis. Additionally, machine learning-oriented schemes were further developed for classification purpose that is trained using pre-labeled dataset of "positive, negative, neutral content" [12, 13]. The major contributions of the paper are as follows:

- Introduces a new "IntelligentSentiNet based Lexicon generation method" that includes seven major phases.

- Optimization assisted NN is introduced in this work for classification purpose, where the training will be carried out by a new improved optimization algorithm via tuning the optimal weights.
- Introduces a novel SI-SLnO algorithm, which is the modified version of the sea lion optimization algorithm with a new updating evaluation. This paves the way for precise evaluation.

The rest of the paper is organized as: The reviews are presented in Section 2. Section 3 depicts about proposed feedback-based sentiment classification model. Section 4 about the proposed lexicon: intelligent sentiment lexicon network. Section 5 portrays the proposed Sentinet construction via optimization-assisted NN-based classification and Sect. 6 describes the feedback and sentiment classification. The resultants are presented in Sect. 7, and the paper is concluded in Sect. 8.

2 Literature review

2.1 Related works

In 2019, Nguyen and Le [14] have established effectual several attention techniques (interactive and intra attention techniques) incorporated with sentiment lexicon information for forming an aspect level of two phases: Aggregation-level and phrase-level information. This permitted the model for incorporating the aspect data into the DNN, and it focused on the exact sentimental context words trained on the instructive aspect words. Finally, the investigational results have pointed out that the aspect-level sentiment classification has been improved by the adopted model.

In 2017, Deng et al. [15] have introduced a method for adapting the prevailing sentiment lexicons for "domain-specific sentiment classification" using a dictionary and unannotated corpus. The implemented technique was computed through two large corpora that contained one million tweets associated with political issues and 743,069 tweets associated with the stock market, correspondingly. Besides, five sentiment lexicons exist as baselines and seeds. As a final point, the outcomes have revealed the effectiveness of the adopted technique by showing enhancement on the performance of sentiment classification.

In 2020, Akshi et al [16] has introduced a novel hybridized deep learning approach for predicting fine-grained sentiment in real-time data. Also, Google Lens was exploited by discretization module for separating the text from the image that was further processed and delivered to the relevant image and text analytics modules. Accordingly, the text analytics portrayed the sentiment using CNN framework. In the end, the accuracy attained by the adopted scheme was almost 91% attained by the image and text modules individually.

In 2019, Zhao et al [17] have established a novel context propagation sentiment model for extracting Chinese micro blog-specific implicit and explicit sentiment features. During the selection process, the seed sentiment elements with a higher standard degree of centrality were chosen. Further, experimentations on two datasets have demonstrated that the adopted technique has produced sentiment lexicons efficiently.

Also, the accuracies of sentiment classification have extensively outperformed the conventional baselines.

In 2017, Keshavarz and Mohammad [18] have enhanced the polarity categorization of sentiments by constructing adaptive sentiment lexicons in microblogs. As per the adopted approach, lexicon-oriented and corpora-oriented techniques were united and lexicons were produced. Moreover, the classification of sentiments was modeled as an optimization issue, where the intention was to discover optimal sentiment lexicons. For solving the optimization issues, GA was adopted, which discovered lexicons for classifying text. The analysis outcomes have exposed better F-measure and accuracy when compared to the existing approaches.

In 2018, Dey et al [19]. have established a new methodology for creating a lexicon known as "Senti-N-Gram." The implemented rule-oriented scheme has extracted the n-gram sentiment scores from an arbitrary corpus that included related numeric ratings and product reviews on a 5-point scale. This approach has presented a sentiment classification technique using a ratio-oriented model depending on counts of negative and positive sentences of a text.

In 2018, Ghiassi and Lee [20] have hierarchically reduced the feature set to a minute set of 7 "meta-features" for reducing sparsity. Also, TSA depending on these features has generated many exact outcomes via SVM and a dynamic approach for NN as calculated by F1 metrics, precision, and recall (the mean of recall and precision). Finally, the transferability and effectiveness of the TGFS were evaluated across distinct and three novel domain.

In 2019, Alharbi and Elise [21] have established a NN approach, which incorporated the user behavioral data in a specified tweet (document). The NN exploited in this work was a CNN. The adopted scheme was computed on two datasets offered by "SemEval-2016 Workshop." The implemented scheme has outperformed the existing schemes as it offered the classifier with a deep knowledge of the task.

2.2 Review

Table 1 shows the reviews on lexicon-based sentiment analysis. At first, the DNN method was introduced in [14], which maximizes the improved accuracy and it also offers better prediction. However, the LSTM model has to be focused more. The average-sense method was exploited in [15] that offers optimal F-measure and it also provides improved sensitivity, but it has to focus more on the recognition of neutral words. SVM model was used in [16] that offers high accuracy and it also offers better precision. However, it needs an analysis on deep learning features. Also, the Markov chain model was implemented in [17, 22]that is highly accurate and it offers better recall; anyhow, it has to be executed using more datasets. GA was presented in [18] that offers improved accuracy with optimal F- measure, but it needs consideration on time consumption.

Moreover, rule-based methods were implemented in [19] that provide enhanced recall along with improved sensitivity. Anyhow, more number of unigram should be discovered. Based on literature work, neural networks perform text classification with a great advantage of performance without adjusting hyper and learn parameters

Table 1 Reviews on conventional sentiment analysis models

Author [citation]	Adopted scheme	Features	Challenges
Nguyen and Le [1]	DNN	Improved accuracy Better prediction	LSTM model have to be considered more
Deng et al.[2]	Average-sense method	Optimal F- measure High sensitivity	Have to focus more on the recognition of neutral words
Akshi et al [3]	SVM model	Highly accurate Better precision	Needs analysis on deep learning features
Zhao et al. [4]	Markov chain model	Highly accurate Better recall	Have to be executed using more datasets
Keshavarz and Mohammad [5]	GA	Improved accuracy Optimal F- measure	Needs consideration on time consumption
Dey et al. [6]	Rule-based methods	Enhanced recall Improves the sensitivity	More number of unigrams could be discovered
Ghiassi and Lee [7]	SVM method	Highly accurate Better recall	Complex owing to manual operations
Alharbi and Elise [8]	DNN	Optimal F1-score Offers better precision	More learning approaches have to be concerned

or features automatically for the given task and do not require prior knowledge of the linguistic structure of target language. Besides, the SVM method was suggested in [20] which offers high accuracy and it also offers better recall. However, the process is complex owing to manual operations. DNN was introduced in [21] which provides an F1-score and it also offers better precision. However, more learning approaches have to be concerned.

3 Proposed feedback-based sentiment analysis: architectural description

The architecture of the proposed Int SentiNet-based sentiment classification using SI-SLnO algorithm is illustrated in Fig. 1.

The presented work is carried out under major phases like (i) pre-processing, (ii) keyword extraction, (iii) holoentropy-based lexicon construction, (iv) feature

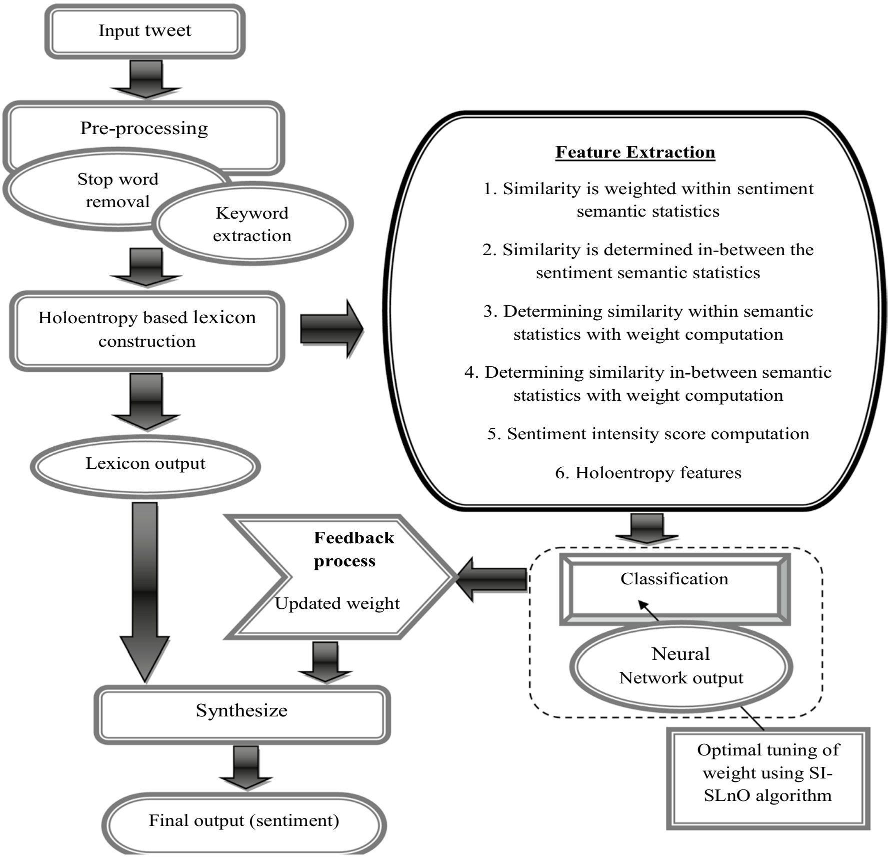

Fig. 1 Architecture of proposed framework

extraction, (semantic similarity), (v) classification (vi) feedback process, and (vii) sentiment classification from Int SentiNet.

Pre-processing is the initial step, where stop word removal takes place and the keywords are extracted. After the keyword extraction, holoentropy-based lexicon construction is carried out. The holoentropy is defined as the sum of the entropy and the total correlation of the random vector and can be expressed by the sum of the entropies on all attributes. It will be used to analyze the relationship between attributes and cluster structure. It is determined to perform semantic similarity of words in the tweet during lexicon construction. As the next process, feature extraction takes place, by which 16 features are extracted from six feature sets. The extracted features are then subjected to the classification process. This work deploys optimized NN for classifying the sentences, where the training is carried out by a new SI-SLnO algorithm via selecting the optimal weights. The outputs attained from NN are subjected to feedback process, where NN outputs are compared with actual output to get the misclassified class. Finally, during the sentiment classification from Int SentiNet, the lexicon output and NN output are synthesized to attain the final sentiment.

4 Proposed lexicon: intelligent sentiment lexicon network

4.1 Pre-processing

This is the preliminary step; where the keywords are extracted from each tweet that is carried out using stop word removal.

4.1.1 Stop word removal and keyword extraction

The stop words are the division of natural languages that offer better knowledge regarding the chats if they are illegal or legal. Usually, the stop words are articles and pronouns that do not offer to mean to the tweet. When the stop words are recognized to be malicious, they have to be eliminated. This removal minimizes the term space dimensionality. Thereby, the keywords are extracted.

4.2 Proposed lexicon construction

From the extracted key words, 90% of data is taken for training purpose, while, 10% of data is taken for testing purpose. The training data contains both positive (+ve) as well as negative (−ve) words. Accordingly, the sentiment of each word is determined as either positive or negative using "sentiword net" S^{net} [23] and they are grouped as defined in Eq. (1), where d indicates the tweets and $d = d_1, d_2 \ldots d_r$, here r denotes the total count of tweets, $S^{\mathrm{net}}_{\mathrm{op}}$ indicates the output of sentinet and w denotes the word. Accordingly, the sentiment of tweet that contains the particular word are denoted as lexicon output indicated by L^{op}.

$$w = \begin{cases} +ve;\text{if } S_{op}^{net} \text{ is} + \text{ve } \& \text{ d } is + ve \\ -ve;\text{if } S_{op}^{net} \text{is} - \text{ ve } \& \text{ d } is - ve \end{cases} \tag{1}$$

4.2.1 Determination of holoentropy-based lexicon

In the proposed holoentropy-based lexicon, the holoentropy of both positive, as well as negative words from the tweets, are determined. Additionally, the weights (w^t) are assigned as one for all words. However, later on, the weights get varied based on the feedback. The feedback process and the weight updating of words are given in the subsequent section. Based on this, the misclassified words are removed and the needed words are moved to the lexicon. The major advantage of the proposed lexicon model is: it is structured with different particulars including words, sentiment, holoentropy, and weights. However, the conventional lexicons are modeled with only the sentiments. The structure of the proposed lexicon is illustrated in Fig. 2.

The holoentropy [24] of each word is computed as per Eq. (2), where l point outs the nonlinear function and E denotes the entropy. The entropy of each word should be calculated within the group, i.e., it should be computed within all the words of $+ve$ sentiment and it is determined using Eq. (3).

In Eq. (3), P_i indicates the semantic similarity between a pair of word, i.e., $\begin{pmatrix} w1(d_1), w2(d_2) \\ w1(d_1), w3(d_2) \\ w1(d_1), wn(d_n) \end{pmatrix}$.

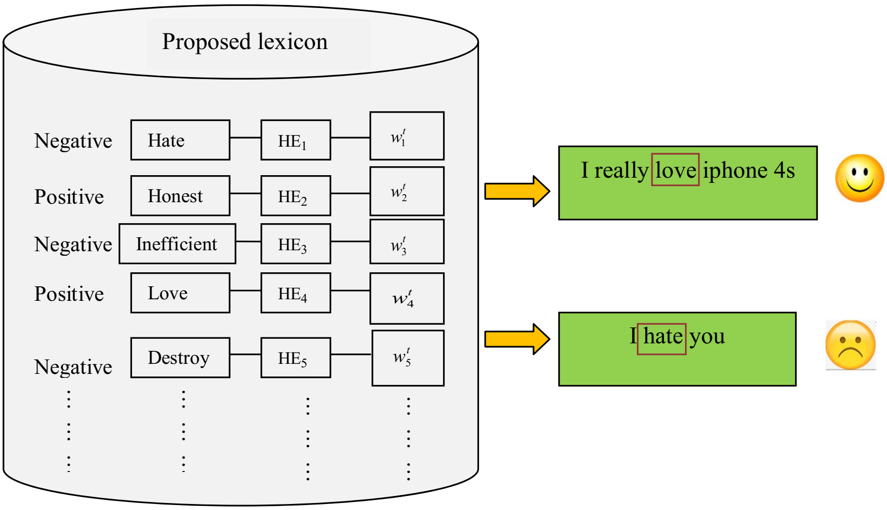

Fig. 2 Diagrammatic representation of proposed Holoentropy-based lexicon

Accordingly, the nonlinear function l is determined as per Eq. (4).

$$\mathrm{HE}(w) = l(w).E(w) \tag{2}$$

$$E(w) = -\sum_{i=1}^{n} P_i \log P_i \tag{3}$$

$$l(w) = 2\left(1 - \frac{1}{1 + \exp\left(-E(w)\right)}\right) \tag{4}$$

5 Proposed sentinet construction via optimization assisted neural network

5.1 Feature extraction process

The proposed sentinet construction model involves six feature sets that are explained below.

5.1.1 Feature set 1

It involves the finding of semantic similarity of every word in a group with other words in the group. That is, the similarity is weighted *within sentiment semantic statistics.*

For e.g., on considering $w1\left(d_1\right)$, the semantic similarity denoted by S can be determined as per Eq. (5), where n denotes the number of words in that group.

$$\begin{aligned} S_1 &= S\left(w1\left(d_1\right), w2\left(d_2\right)\right) \\ S_2 &= S\left(w1\left(d_1\right), w3\left(d_2\right)\right) \\ &\vdots \\ Sn &= S\left(w1\left(d_1\right), wn\left(d_n\right)\right) \end{aligned} \tag{5}$$

Consequently, the statistical features namely, maximum, minimum, mean, median, and standard deviation (SD) are derived from the semantic similarities as shown in Eq. (6).

$$\begin{aligned} s_1 &= \max\left(S_1, S_2, \ldots S_n\right) \\ s_2 &= \min\left(S_1, S_2, \ldots S_n\right) \\ s_3 &= \mathrm{mean}\left(S_1, S_2, \ldots S_n\right) \\ s_4 &= \mathrm{median}\left(S_1, S_2, \ldots S_n\right) \\ s_5 &= \mathrm{SD}\left(S_1, S_2, \ldots S_n\right) \end{aligned} \tag{6}$$

Thus, five features are obtained from feature set 1 denoted by Fe_1, i.e. $Fe_1 = s_1, s_2, s_3, s_4, s_5$.

5.1.2 Feature set 2

Here, the feature extraction process is the same as feature set 1; however, the similarity is determined *in-between the sentiment semantic statistics*. That is, the similarity is determined among the positive as well as negative words.

Accordingly, the statistical features s_1, s_2, s_3, s_4, s_5 are derived from the semantic similarities as similar to the feature set 1. Therefore, five features are obtained from feature set 2 denoted by Fe_2, i.e., $Fe_2 = s_1, s_2, s_3, s_4, s_5$.

5.1.3 Feature set 3

Here, the similarity is *weighted within semantic statistics*. Here, two semantic words are considered for each word and statistical features are derived for each semantic word as similar to feature set 1. For e.g., if we take the first semantic word the semantic similarity $S_1, S_2, \ldots, S_n$ has to be computed with other words in the group.

The computation of weighted function F_j is given by Eq. (7), where $j = 1, 2 \ldots \mathrm{nu}$, where nu denotes the number of semantic words and normalized similarity score $\overline{f}_j$ is computed as per Eq. (8), where f_j denotes the current similarity score and $\max\left(f_1, f_2 \ldots f_n\right)$ denotes the maximal similarity value. Accordingly, the weighted computation for maximum function Ff_1 is expressed as in Eq. (9).

$$F_j = \frac{\overline{f}_j}{\sum_j \overline{f}_j} \text{ Here, } \begin{array}{c} f_1 \rightarrow f_1\left(S_1\right) \\ f_2 \rightarrow f_1\left(S_2\right) \\ \vdots \\ f_n \rightarrow f_1\left(S_n\right) \end{array} \tag{7}$$

$$\text{Normalized } f_j = \overline{f}_j = \frac{f_j}{\max\left(f_1, f_2 \ldots f_n\right)} \tag{8}$$

$$Ff_1 = F_1 f_1\left(S_1\right) + F_2 f_1\left(S_2\right) + \cdots F_n f_1\left(S_n\right) \tag{9}$$

Similarly, Ff_2 is determined within the class as given in Eq. (10).

$$Ff_2 = F_1 f_2\left(S_1\right) + F_2 f_2\left(S_2\right) + \cdots F_n f_2\left(S_n\right) \tag{10}$$

As a result, two features are obtained from feature set 3, denoted by Fe_3, i.e., $Fe_3 = Ff_1 + Ff_2$.

5.1.4 Feature set 4

The feature extraction process is similar to the feature set 3, however, the similarity is determined *in-between semantic statistics*. That is, the similarity is determined among the positive as well as negative words.

Thus, two features are obtained from the feature set 4 that is denoted by Fe_4, i.e., $Fe_4 = Ff_1 + Ff_2$.

5.1.5 Feature set 5

Here, sentiment intensity score denoted by C is computed based on the presence of intensifiers Int as given in Eq. (11), where, N_{int} denotes the number of intensifiers.

$$C = \begin{cases} N_{\text{int}} + 1; & \text{if no negation} \\ -\left(N_{\text{int}} + 1\right); & \text{if negation} \\ 0; & \text{if no Int} \end{cases} \tag{11}$$

For e.g., the word "very good" represents 1 intensifier and there is no negation. Therefore, it is assigned a value of 2. On the other hand, the word "not very good" represents negation, and therefore, it is assigned a value of -2.

Thus, the fifth set feature denoted by Fe_5 is computed as given in Eq. (12), where S^{wo} denotes the sentiment of the word.

$$Fe_5 = C \times S^{\text{wo}} \left\{ \begin{array}{l} +1;\ \text{if} + \text{ve statement} \\ -1;\ \text{if} - \text{ve statement} \end{array} \right\} \tag{12}$$

Therefore, one feature is obtained from Fe_5.

5.1.6 Feature set 6

The holoentropy features that were computed as given in Table 2 constitutes feature set 6. As a result, one feature is obtained from the feature set 6 that is denoted by Fe_6 .

Thus, the totals of 16 features are attained from the six feature sets and are entirely denoted by $Fe = Fe_1 + Fe_2 + Fe_3 + Fe_4 + Fe_5 + Fe_6$. These features with the size of 5000×16 (since this work considered the positive words as 5000) are given as input to NN for training and here, the sentiment of the document containing the positive word is considered as the target. Based on the specification of context, the holoentropy-based lexicon is constructed for context-related sentiment bearing terms. The context is defined as a framework that provides relevant resources for the implementation of an operation corresponding to events or concepts. The context in sentiment analysis is defined as any complementary source of evidence that can either intensify or flip the polarity of content.

Finally, the lexicon is used to perform context-aware sentiment analysis by performing neural network-based sentiment classification.

Table 2 Exemplary representation of misclassified words attained after training

Actual	NN output
$w3$	
$+ve$	$-ve$
$w2$	
$-ve$	$+ve$

5.2 Optimized neural network

NN [25] considers the features Fe as inputs as shown in Eq. (13), in which nu signifies the total feature's count ($nu=16$).

$$Fe = \left\{ Fe_1, Fe_{2.}, \dots Fe_{\mathrm{nu}} \right\} \tag{13}$$

The model includes input, output, and hidden layers. The output of the hidden layer $e^{(H)}$ is portrayed by Eq. (14), here $nf(.)$ symbolizes "activation function," $\widehat{i}$ and j refers to the neurons of hidden and input layers correspondingly, $W^{(H)}_{\left(B\widehat{i}\right)}$ denotes bias weight to $\widehat{i}^{th}$ hidden neuron, input neuron's count is symbolized by $n_{\widehat{i}}$ and $W^{(H)}_{\left(j\widehat{i}\right)}$ denotes the weight from jth input neuron to $\widehat{i}th$ hidden neuron. The output of the network $\hat{G}_O$ is determined as in Eq. (15), where $\hat{O}$ denote output neurons, n_h denotes the hidden neurons' count, $W^{(G)}_{(B\hat{o})}$ indicates output bias weight to the $\hat{o}^{\mathrm{th}}$ output layer, and $W^{(G)}_{\left(\widehat{i}\hat{o}\right)}$ specifies the weight from $\widehat{i}^{\mathrm{th}}$ hidden layer to $\hat{o}^{\mathrm{th}}$ output layer. Consequently, the error among the predicted and actual values is computed as per Eq. (16) which should be reduced (objective). In Eq. (16), n_G symbolizes the output neuron count, $G_{\hat{o}}$ and $\hat{G}_{\hat{o}}$ refers to the actual and predicted output, respectively.

$$e^{(H)} = nf\left(W^{(H)}_{\left(B\widehat{i}\right)} + \sum_{j=1}^{n_{\widehat{i}}} W^{(H)}_{\left(j\widehat{i}\right)} Fe \right) \tag{14}$$

$$\hat{G}_{\hat{o}} = nf\left(W^{(G)}_{(B\hat{o})} + \sum_{\widehat{i}=1}^{n_h} W^{(G)}_{\left(\widehat{i}\hat{o}\right)} e^{(H)} \right) \tag{15}$$

$$Er^* = \underset{\left\{ W^{(H)}_{\left(B\widehat{i}\right)}, W^{(H)}_{\left(j\widehat{i}\right)}, W^{(G)}_{(B\hat{o})}, W^{(G)}_{\left(\widehat{i}\hat{o}\right)} \right\}}{\arg\min} \sum_{=1}^{n_G} \left| G_{\hat{o}} - \hat{G}_{\hat{o}} \right| \tag{16}$$

As mentioned above, the training of NN model is carried out using a new SI-SLnO algorithm via optimizing the weights, $W = W^{(H)}_{\left(B\widehat{i}\right)}, W^{(H)}_{\left(j\widehat{i}\right)}, W^{(G)}_{(B\hat{o})}$ and $W^{(G)}_{\left(\widehat{i}\hat{o}\right)}$. The output attained from NN is denoted as $\hat{G}_{\hat{o}}$.

5.3 Proposed SI-SLnO algorithm

SLnO [26] is are re reowned optimization scheme that portrays the hunting nature of sea lions. To make the algorithm more effective with better convergence, it is planned to make some improvements in the algorithm. Self-improvement is proven to be promising in traditional optimization algorithms [27, 28]. The sea lions consist of a sensitive feature known as "Whiskers" that aids them to find out the precise prey positions.

Four phases of the algorithm are (i) tracking, (ii) social hierarchy, (iii) attacking, and (iv) encircling prey. The detailed description of its phase is as follows:

5.3.1 Detecting and tracking phase

The tracking mechanism is specified as per Eq. (17), which D denotes the distance among sea lion and target prey, the vector position of sea lion and targeted prey is given by $\vec{X}(t)$ and $\vec{M}(t)$, in that order, t denotes the present iteration and the arbitrary vector is pointed out by $\vec{V}$.

$$D = \left|2\vec{V}.\vec{M}(t) - \vec{X}(t)\right| \tag{17}$$

The proposed contribution is given as follows: Conventionally, at successive iterations, the sea lion moves towards the target prey based on $\vec{M}(t)$, subsequent iteration $(t+1)$ and random factor $\vec{H}$ which is steadily minimized over the iterations from 2 to 0. In the proposed context, the movement of sea lion towards the target prey depends on the above constraints along with the "degree of distance, δ" as given in Eq. (18). The computation of δ takes place as per Eq. (19), where ND points out the normalized distance and G^M indicates the geometric mean. ND can be evaluated as shown in Eq. (20), where D denotes the distance and Max^D denotes the maximum distance.

$$\vec{X}(t+1) = \vec{M}(t) - D.\vec{H} \times \delta \tag{18}$$

$$\delta = G^M(ND) \tag{19}$$

$$ND = \frac{D}{Max^D}. \tag{20}$$

5.3.2 Vocalization phase

When sea lions discover a prey, it calls other sea lions for attacking as shown in Eqs. (21), (22) and (23), where the speed of a leader's sound is indicated as $\overrightarrow{X_{leader}}$, the sound's speed in air and water is denoted by $\vec{P_2}$ and $\vec{P_1}$.

$$\overrightarrow{X_{leader}} = \left|\left(\vec{P_1}(1+\vec{P_2})\right)/\vec{P_2}\right| \tag{21}$$

$$\overrightarrow{P_1} = \sin\theta \tag{22}$$

$$\overrightarrow{P_2} = \sin\varphi \tag{23}$$

The pseudocode of the proposed model is portrayed in Algorithm 1, and the flow-chart is illustrated in the following Fig. 3.

Algorithm 1:Proposed SI-SLnOscheme

Initialization of population

Choose $\overrightarrow{X_{rnd}}$

Estimate the fitness function for every search agent

if $(i < \max iter)$

    Compute $\overrightarrow{X_{leader}}$ as per Eq. (21)

    If($\overrightarrow{SP_{leader}}$ <0..25)

        If ($H < 1$)

            Update the current search agent location using Eq. (17)

            Compute the movement of sea lion towards the prey as per Eq. (18)

        else

            Select a random search agent $\overrightarrow{X_{rnd}}$

            Update the current search agent location using Eq. (26)

        End if

    else

        Update the current search agent location using Eq. (24)

    End if

    If the search agent do not belong to any $\overrightarrow{X_{leader}}$

        Go to the first if condition

    else

        Compute the fitness function for every search agent

        Update $\overrightarrow{X}$ as per the better solution

        Return $\overrightarrow{X}$, which is the best solution

    End if

End if

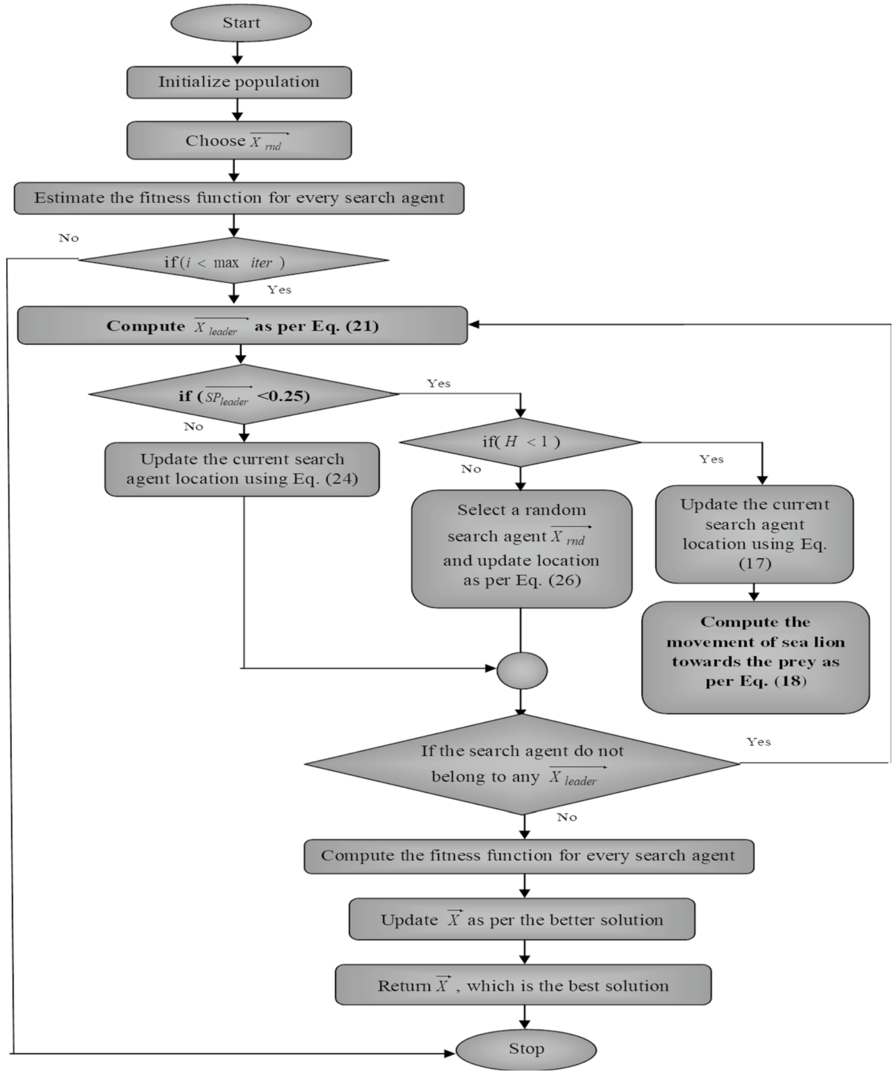

Fig. 3 Flowchart of SI-SLnO model

5.3.3 Attacking phase

The attacking process of the sea lion is specified in Eq. (24), where the distance among the sea lion and target prey is denoted by $\vec{M}(t) - \vec{X}(t)$, the absolute value is denoted by ||, and the random number among to 1 is referred by l.

$$\vec{X}(t+1) = \left|\vec{M}(t) - \vec{X}(t).\cos(2\pi l)\right| + \vec{M}(t) \tag{24}$$

5.3.4 Prey Searching

The SLnO model carries out a global search when $\vec{H}$ is higher than 1. This is exhibited by Eqs. (25) and (26).

$$D = \left| 2\vec{B}.\overrightarrow{X_{rnd}}(t) - \vec{X}(t) \right| \tag{25}$$

$$\vec{X}(t+1) = \overrightarrow{X_{rnd}}(t) - D\vec{H} \tag{26}$$

6 Feedback and sentiment classification from Int SentiNet

6.1 Feedback process

The sentiment of words with variation in the actual output and NN output is said to be the misclassified words. Table 2 shows the exemplary representation of misclassified words attained after the training process.

Under such conditions, if the word in the lexicon column is equal to the actual sentiment of the document, then w^t of a particular word (as given in Table 2) will be incremented by one and the word is removed from the misclassified set. In the reverse condition, i.e., if the word in the lexicon column is not equal to the actual sentiment of the document, then the word will be moved to the respective column of the lexicon and the weight of that word will be incremented by one. Thus, the lexicon weights of all words are updated as given in Eq. (27), where N^{we} denotes the new weight of words, w^t denotes the weight of word, and $\sum w^t$ denotes the weight of all words. Algorithm 2 depicts the pseudocode of the feedback process.

$$N^{we} = \frac{w^t}{\sum w^t} \tag{27}$$

Algorithm 2: Feedback process		
For every word in misclassified set		
	if word is in lexicon column= actual sentiment of document	
		remove the word from misclassified set
	else	
		move the word to the respective column of lexicon
	end if	
		$w^t = w^t + 1$
end for		

6.2 Sentiment classification from Int SentiNet

During the sentiment classification process, the lexicon output L^{op} and $\hat{G}_{\hat{o}}$ are fused to attain the final sentiment. The parameter L^{op} indicates the sentiment of the tweet that is taken from the respective word. The lexicon word that is highly similar to each word of each sentiment group of the lexicon of testing data is computed as per Eq. (28), where L^{w*} denotes the maximum similarity of lexicon word and L^{w} denotes the words in the lexicon.

$$L^{w*} = \underset{L^{w}}{\arg\max}\, S(L^{w}, w1) \tag{28}$$

In this way, the sentiment of the lexicon words and the NN output of that word are considered as L^{op} and $\hat{G}_{\hat{o}}$. Thus, the sentiment of word 1 is computed as shown in Eq. (29), where S^{op} indicates the sentiment output,F^{Lw} denotes the weight of the lexicon word and accordingly, the final sentiment (FS) is evaluated as per Eq. (30), which Av denotes the average.

$$S^{op}(w1) = \begin{cases} 0.5 \times L^{op} + 0.5 \times \hat{G}_{\hat{o}}; & if\ F^{Lw} = 1 \\ F^{Lw} \times L^{op} + \left(1 - F^{Lw}\right) \times \hat{G}_{\hat{o}}; & else \end{cases} \tag{29}$$

$$FS = Av(\,(w1)^{op},\ (w2)^{op}\,) \tag{30}$$

7 Results and discussion

7.1 Simulation procedure

The proposed lexicon-based sentiment classification model was implemented in JAVA and the resultants were observed. Here, an analysis was done on two datasets namely, "judge and twcs.csv" and they were downloaded from "https://www.figure-eight.com/data-for-everyone/ and https://www.kaggle.com/thoughtvector/customer-support-on-twitter#twcs.csv," respectively. In the description section, the judge was represented as 1st dataset, and twcs were represented as 2nd dataset. Accordingly, the betterment of the proposed SI-SLnO model was evaluated by comparing it over the traditional models like GA [29], SVM [30], CNN [31], LSTM [32], NN [25], and SLnO + NN [26] with respect to "accuracy, sensitivity, specificity, precision, FPR, FNR, FDR, and FOR". The analysis was carried out with respect to varied training data that ranges from 60%, 70%, 80%, and 90%. In addition, convergence analysis was carried out with respect to MSE measure for varied iterations.

Table 3 Performance analysis of proposed model over existing models in terms of positive measures using judge dataset

Training data in %	GA[30]	SVM [31]	CNN [32]	LSTM [33]	NN [27]	SLnO + NN [26]	SI-SLnO
Accuracy							
60	0.49085	0.500538	0.582347	0.703108	0.715823	0.765339	0.847147
70	0.471475	0.474704	0.593111	0.703108	0.726588	0.83423	0.847147
80	0.487621	0.510226	0.569429	0.703108	0.709365	0.843918	0.88267
90	0.485468	0.516685	0.590958	0.703108	0.724435	0.841765	0.905274
Sensitivity							
60	0.465625	0.475	0.665625	0.78125	0.81875	0.859375	0.996894
70	0.46875	0.521875	0.6125	0.746875	0.840625	0.8625	0.996894
80	0.521875	0.53125	0.609375	0.75	0.8375	0.9	0.996894
90	0.496875	0.521875	0.565625	0.725	0.840625	0.91875	0.996894
Specificity							
60	0.499179	0.518883	0.538588	0.548282	0.681445	0.737274	0.840722
70	0.444992	0.477833	0.548282	0.582923	0.715928	0.83087	0.83908
80	0.469622	0.499179	0.548282	0.54844	0.688013	0.847291	0.873563
90	0.479475	0.513957	0.548282	0.604269	0.724138	0.842365	0.898194
Precision							
60	0.332604	0.337104	0.431174	0.537688	0.563063	0.620853	0.739247
70	0.320513	0.330693	0.435556	0.537688	0.580097	0.723118	0.737968
80	0.340816	0.357895	0.414894	0.537688	0.55814	0.742382	0.789041
90	0.334034	0.360691	0.42891	0.537688	0.58	0.736986	0.825843

Table 4 Performance analysis of proposed model over existing models in terms of negative measures using judge dataset

Training data in %	GA[30]	SVM [31]	CNN [32]	LSTM [33]	NN [27]	SLnO+NN [26]	SI-SLnO
FPR							
60	0.500821	0.481117	0.461412	0.451718	0.318555	0.262726	0.159278
70	0.555008	0.522167	0.451718	0.417077	0.284072	0.16913	0.16092
80	0.530378	0.500821	0.451718	0.45156	0.311987	0.152709	0.126437
90	0.520525	0.486043	0.451718	0.395731	0.275862	0.157635	0.101806
FNR							
60	0.534375	0.525	0.334375	0.21875	0.18125	0.140625	0.003106
70	0.53125	0.478125	0.3875	0.253125	0.159375	0.1375	0.003106
80	0.478125	0.46875	0.390625	0.25	0.1625	0.1	0.003106
90	0.503125	0.478125	0.434375	0.275	0.159375	0.08125	0.003106
FDR							
60	0.667396	0.662896	0.568826	0.462312	0.436937	0.379147	0.260753
70	0.679487	0.669307	0.564444	0.462312	0.419903	0.276882	0.262032
80	0.659184	0.642105	0.585106	0.462312	0.44186	0.257618	0.210959
90	0.665966	0.639309	0.57109	0.462312	0.42	0.263014	0.174157
FOR							
60	0.355932	0.351129	0.245977	0.14433	0.114398	0.08079	0.002976
70	0.368764	0.360849	0.258873	0.156673	0.091562	0.079279	0.002976
80	0.348519	0.330396	0.272331	0.160321	0.091549	0.056738	0.002976
90	0.355408	0.328326	0.274162	0.166352	0.090426	0.045375	0.002976

7.2 Performance analysis

The performance of proposed model over the conventional model with respect to varied measures for judge dataset is represented in Tables 3 and 4, whereas the analysis on the twcs.csv dataset is represented in Tables 5 and 6, respectively. The analysis was carried out by varying the training data to validate the betterment of the presented scheme. On noticing the tables, the presented SI-SLnO model has accomplished better accuracy and precision when compared over the compared models. Here, from Table 3, the presented method has achieved an accuracy of 0.847 at 60% of training data; however, as the percentage of training data increases, the presented method has achieved a higher accuracy up to 0.9. Similarly, the specificity of the adopted scheme at 60% of training data is 0.84, while at 90% of training data; a higher value of 0.89 has been achieved. Thereby, the adopted method has accomplished better outcomes with the increase in the % of training data. However, the existing algorithms have attained poor results when compared to the proposed model.

Table 5 revealed the analysis on positive measures, while the negative measures are revealed by Table 6. Specifically, from Table 5, the accuracy of the proposed

Table 5 Performance analysis of proposed model over existing models in terms of positive measures using twcs.csv dataset

Training data in %	GA [30]	SVM [31]	CNN [32]	LSTM [33]	NN [27]	SLnO + NN [26]	SI-SLnO
Accuracy							
60	0.487736	0.509434	0.592453	0.737736	0.791509	0.851504	0.857547
70	0.479245	0.510377	0.584906	0.727358	0.832075	0.851504	0.869811
80	0.500943	0.500943	0.586792	0.731132	0.84434	0.851504	0.890566
90	0.479245	0.488679	0.586792	0.731132	0.833962	0.851504	0.904717
Sensitivity							
60	0.5	0.508065	0.508065	0.704301	0.768817	0.884409	0.997326
70	0.451613	0.486559	0.497312	0.709677	0.819892	0.86828	0.997326
80	0.462366	0.502688	0.526882	0.709677	0.865591	0.892473	0.997326
90	0.448925	0.478495	0.502688	0.709677	0.846774	0.908602	0.997326
Specificity							
60	0.481105	0.510174	0.638081	0.755814	0.772464	0.803779	0.843023
70	0.494186	0.523256	0.632267	0.736919	0.772464	0.838663	0.87064
80	0.486919	0.521802	0.632267	0.742733	0.772464	0.832849	0.889535
90	0.494186	0.49564	0.632267	0.742733	0.772464	0.827035	0.902616
Precision							
60	0.342541	0.359316	0.431507	0.609302	0.679335	0.703774	0.75286
70	0.325581	0.355599	0.422374	0.593258	0.703774	0.733173	0.783981
80	0.343313	0.357013	0.425	0.598639	0.703774	0.736842	0.813725
90	0.324903	0.338403	0.425	0.598639	0.703774	0.725806	0.834568

SI-SLnO method at90% of training data is 86.47%, 75.21%, 18.57%, 28.75%, 24.96% and 7.54% superior to traditional methods such as GA, SVM, CNN, LSTM, NN and SLnO + NN, respectively. In addition, the sensitivity of the proposed SI-SLnO algorithm at 80% of training data is 86.47%, 75.21%, 18.57%, 28.75%, 24.96%, and 7.54% superior to GA, SVM, CNN, LSTM, NN, and SLnO + NN correspondingly. Similarly, the specificity of SI-SLnO method at 90% of training data is 87.33%, 74.76%, 63.82%, 17.76%, 24.04%, and 6.63% superior to GA, SVM, CNN, LSTM, NN and SLnO + NN, respectively. On analyzing the precision, the presented scheme has attained a minimal value of 0.7392473, whereas the existing schemes, namely GA, SVM, CNN, LSTM, NN, and SLnO + NN models has achieved comparatively higher values of 0.3326039, 0.337104072, 0.4311740, 0.537688442, 0.5630630, and 0.6208530, respectively, at 60% of training data.

Likewise, on observing the negative measures, the presented model has revealed a minimal value over the compared models. More particularly, from Table 6, the FPR measure of the presented SI-SLnO scheme has attained a minimal value at 90% of training data, which is 80.75%, 80.69%, 73.52%, 62.15%, 57.2%, and 43.69% superior to GA, SVM, CNN, LSTM, NN, and SLnO + NN models. Moreover, the FNR measure at 60% of training data is minimal that is 99.46%, 99.46%, 99.46%, 99.09%,

Table 6 Performance analysis of proposed model over existing models in terms of negative measures using twcs.csv dataset

Training data in %	GA [30]	SVM [31]	CNN [32]	LSTM [33]	NN [27]	SLnO + NN [26]	SI-SLnO
FPR							
60	0.518895	0.489826	0.361919	0.244186	0.227536	0.196221	0.156977
70	0.505814	0.476744	0.367733	0.263081	0.227536	0.161337	0.12936
80	0.513081	0.478198	0.367733	0.257267	0.227536	0.167151	0.110465
90	0.505814	0.50436	0.367733	0.257267	0.227536	0.172965	0.097384
FNR							
60	0.5	0.491935	0.491935	0.295699	0.231183	0.115591	0.002674
70	0.548387	0.513441	0.502688	0.290323	0.180108	0.13172	0.002674
80	0.537634	0.497312	0.473118	0.290323	0.134409	0.107527	0.002674
90	0.551075	0.521505	0.497312	0.290323	0.153226	0.091398	0.002674
FDR							
60	0.657459	0.640684	0.568493	0.390698	0.320665	0.296226	0.24714
70	0.674419	0.644401	0.577626	0.406742	0.296226	0.266827	0.216019
80	0.656687	0.642987	0.575	0.401361	0.296226	0.263158	0.186275
90	0.675097	0.661597	0.575	0.401361	0.296226	0.274194	0.165432
FOR							
60	0.359768	0.342697	0.294212	0.174603	0.134585	0.069021	0.001873
70	0.375	0.346642	0.300643	0.17561	0.104037	0.075617	0.001873
80	0.357782	0.344423	0.298387	0.174475	0.080257	0.06135	0.001873
90	0.375458	0.363296	0.298387	0.174475	0.091054	0.051908	0.001873

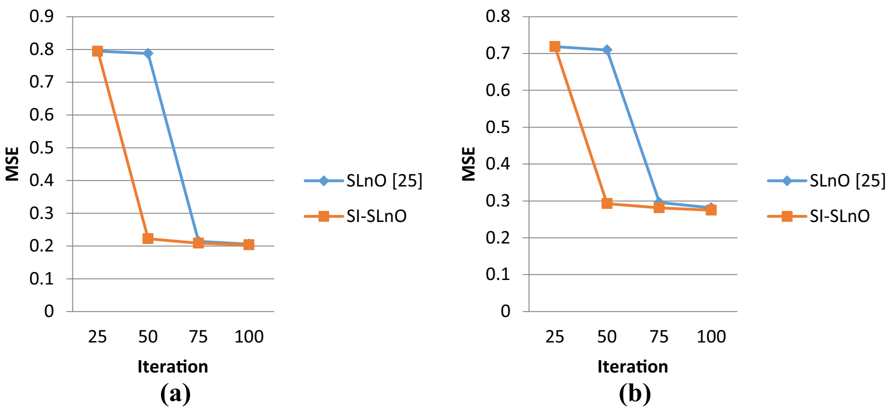

Fig. 4 Convergence analysis of adopted model over traditional SLnO scheme regarding MSE using (**a**) Judge (**b**) twcs.csv

98.84%, and 97.69% superior to GA, SVM, CNN, LSTM, NN and SLnO + NN models. The FOR measure of the adopted scheme, which has to be minimal for better performance of the system is 99.48%, 99.46%, 99.37%, 98.93%, 97.67%, and

Table 7 Overall analysis of proposed model over existing models using judge dataset

Measures	SI-SLnO	SLnO [26]	NN [27]
Accuracy	0.88266954	0.84391819	0.70936491
Sensitivity	0.99689441	0.9	0.8375
Specificity	0.87356322	0.84729064	0.68801314
Precision	0.7890411	0.74238227	0.55813953
FPR	0.12643678	0.15270936	0.31198686
FNR	0.00310559	0.1	0.1625
FDR	0.2109589	0.25761773	0.44186047
FOR	0.00297619	0.05673759	0.0915493

Table 8 Performance analysis of proposed model over existing models using twcs.csv dataset

Measures	SI-SLnO	SLnO [26]	NN [27]
Accuracy	0.890566	0.851504	0.84434
Sensitivity	0.997326	0.892473	0.865591
Specificity	0.889535	0.832849	0.772464
Precision	0.813725	0.736842	0.703774
FPR	0.110465	0.167151	0.227536
FNR	0.002674	0.107527	0.134409
FDR	0.186275	0.263158	0.296226
FOR	0.001873	0.06135	0.080257

96.95% superior to GA, SVM, CNN, LSTM, NN, and SLnO + NN models at 80% of training data. This shows the enhancement of the presented SI-SLnO framework over the existing models.

7.3 Convergence analysis

Figure 4a, b show the convergence analysis of the adopted model over the conventional SLnO model in terms of MSE for Judge and twcs.csv datasets, respectively. Here, the analysis was done with respect to MSE by varying the iterations from 25, 50, 75, and 100. On observing the outcomes, the presented model has revealed more effective results when compared to the SLnO models. More specifically, from Fig. 3a, the MSE of the presented model at the 25th iteration is 0.8, whereas, at the 100th iteration, a much lower error of 0.2 has been reported. Hence, as the count of iteration increases, the MSE of the presented scheme has been reduced. This minimization of error indicates the raise of accuracy, i.e., it ensures accurate prediction outputs. In Fig. 3a, the MSE of adopted scheme is 0.8% better than the existing SLnO model at the 100th iteration. On observing the 50th iteration using Judge dataset, the MSE of the adopted scheme is 71.76% superior to the SLnO model.

Similarly, from Fig. 3b, the MSE of offered model is 58.75% better than the existing SLnO model at the 50th iteration. In addition, at the100th iteration, the MSE of the adopted scheme is 2.31% superior to the compared SLnO model. Thus, the development of the presented SI-SLnO model has been verified from the investigation outcomes in terms of MSE.

7.4 Overall analysis

This section outlines the overall review of the proposed sentiment analysis model over current models with regard to different success metrics. Tables 7 and 8 depict the overall performance for judge and twcs.csv dataset, respectively. On examining the attained outputs, higher accuracy, sensitivity, and precision are attained by the presented scheme over the traditional schemes. Also, the false rates of the adopted model such as FPR, FNR, FOR, and FDR has accomplished minimal values, thus guaranteeing a minimal error. Particularly, higher accuracy of 0.88266 has been attained by the implemented SI-SLnO scheme whereas the compared NN and SLnO schemes have obtained comparatively lower accuracy values of 0.7093649 and 0.843918.

Moreover, the sensitivity of the suggested method is 19.03% and 10.77% superior to traditional NN and SLnO models. Furthermore, the accuracy of implemented SI-SLnO scheme has obtained a higher value of 0.88266; whereas, the compared NN and SLnO schemes has obtained comparatively lower accuracy values of 0.7093649 and 0.843918. Thus, the development of the presented SI-SLnO model is established from the examined outcomes.

8 Conclusion

This paper has developed a new "Intelligent Senti-Net based Lexicon generation method," that included seven major phases. In addition, optimization-assisted NN-based classification was introduced in this work that defined precise or optimal selection of weights using a new improved algorithm termed as SI-SLnO algorithm. Finally, the simulation was carried out to validating the enhancement of the presented scheme. On observing the outcomes, the specificity of SI-SLnO method at the 90th iteration was 87.33%, 74.76%, 63.82%, 17.76%, 24.04%, and 6.63% superior to GA, SVM, CNN, LSTM, NN and SLnO + NN, respectively. On analyzing the precision, the presented scheme has attained a minimal value of 0.7392473, whereas, the existing schemes namely, GA, SVM, CNN, LSTM, NN, and SLnO + NN models has achieved comparatively higher values of 0.3326039, 0.337104072, 0.4311740, 0.537688442, 0.5630630, and 0.6208530, respectively, at 60th iteration. Thus, the superiority of the developed model has been verified successfully.

References

1. Mukhtar N, Khan MA, Chiragh N (2018) Lexicon-based approach outperforms Supervised Machine Learning approach for Urdu Sentiment Analysis in multiple domains. Telemat Inform 35(8):2173–2183
2. Araque O, Zhu G, Iglesias CA (2019) A semantic similarity-based perspective of affect lexicons for sentiment analysis. Knowl Based Syst 165:346–359
3. Al-Rowaily K, Abulaish M, Haldar NAH, Al-Rubaian M (2015) BiSAL – A bilingual sentiment analysis lexicon to analyze Dark Web forums for cyber security. Digit Investig 14:53–62
4. Mowlaei ME, Abadeh MS, Keshavarz H (2020) Aspect-based sentiment analysis using adaptive aspect-based lexicons. Expert Syst Appl. https://doi.org/10.1016/j.eswa.2020.113234
5. Sixing Wu, Fangzhao Wu, Chang Y, Chuhan Wu, Huang Y (2019) Automatic construction of target-specific sentiment lexicon. Expert Syst Appl 116:285–298
6. Al-Thubaity A, Alqahtani Q, Aljandal A (2018) Sentiment lexicon for sentiment analysis of Saudi dialect tweets. Procedia Comput Sci 142:301–307
7. Vashishtha S, Susan S (2019) Fuzzy rule based unsupervised sentiment analysis from social media posts. Expert Syst Appl. https://doi.org/10.1016/j.eswa.2019.112834
8. Hogenboom A, Heerschop B, Frasincar F, Kaymak U, de Jong F (2014) Multi-lingual support for lexicon-based sentiment analysis guided by semantics. Decis Support Syst 62:43–53
9. Youssef M, El-Beltagy SR (2018) MoArLex: an arabic sentiment lexicon built through automatic lexicon expansion. Procedia Comput Sci 142:94–103
10. Tan S, Wu Q (2011) A random walk algorithm for automatic construction of domain-oriented sentiment lexicon. Expert Syst Appl 38(10):12094–12100
11. Itani M, Roast C, Al-Khayatt S (2017) Developing resources for sentiment analysis of informal arabic text in social media. Procedia Comput Sci 117:129–136
12. Zhang Z, Zou Y, Gan C (2018) Textual sentiment analysis via three different attention convolutional neural networks and cross-modality consistent regression. Neurocomputing 275:1407–1415
13. Hassan SU, Aljohani NR, Idrees N, Sarwar R, Herrera F (2019) Predicting literature's early impact with sentiment analysis in Twitter, Knowledge-Based Systems, In press, corrected proof, Available online 14 December 2019, Article 105383
14. Nguyen HT, Nguyen LM (2019) ILWAANet: an interactive lexicon-aware word-aspect attention network for aspect-level sentiment classification on social networking. Expert Syst Appl. https://doi.org/10.1016/j.eswa.2019.113065
15. Deng S, Sinha AP, Zhao H (2017) Adapting sentiment lexicons to domain-specific social media texts. Decis Support Syst 94:65–76
16. Kumar A, Srinivasan K, Cheng WH, Zomaya AY (2020) Hybrid context enriched deep learning model for fine-grained sentiment analysis in textual and visual semiotic modality social data. Inform Process Manag. https://doi.org/10.1016/j.ipm.2019.102141
17. Zhao C, Wang S, Li D (2019) Exploiting social and local contexts propagation for inducing Chinese microblog-specific sentiment lexicons. Comput Speech Lang 55:57–81
18. Keshavarz H, Abadeh MS (2017) ALGA: adaptive lexicon learning using genetic algorithm for sentiment analysis of microblogs. Knowl Based Syst 122:1–16
19. Dey A, Jenamani M, Thakkar JJ (2018) Senti-N-Gram: an n-gram lexicon for sentiment analysis. Expert Syst Appl 103:92–105
20. Ghiassi M, Lee S (2018) A domain transferable lexicon set for Twitter sentiment analysis using a supervised machine learning approach. Expert Syst Appl 106:197–216
21. Alharbi ASM, de Doncker E (2019) Twitter sentiment analysis with a deep neural network: an enhanced approach using user behavioral information. Cogn Syst Res 54:50–61
22. Fu X, Liu W, Xu Y, Cui L (2017) Combine HowNet lexicon to train phrase recursive autoencoder for sentence-level sentiment analysis. Neurocomputing 241:18–27
23. Khan FH, Qamar U, Bashir S (2016) SentiMI: introducing point-wise mutual information with SentiWordNet to improve sentiment polarity detection. Appl Soft Comput 39:140–153
24. Bhutada D, Balaram VVSSS, Bulusu VV (2016) holoentropy based dynamic semantic latent dirichilet allocation for topic extraction. J Inf Optim Sci. https://doi.org/10.1080/02522667.2016.1165000

25. Mohan Y, Chee SS, Xin DKP, Foong LP, (2016) Artificial neural network for classification of depressive and normal in EEG, 2016 IEEE EMBS Conference on Biomedical Engineering and Sciences (IECBES)
26. Masadeh R, Mahafzah B, Sharieh A (2019) Sea lion optimization algorithm. Int J Adv Comput Sci Appl 10:388–395
27. Rajakumar BR (2013) Impact of static and adaptive mutation techniques on genetic algorithm. Int J Hybrid Intell Syst 10(1):11–22. https://doi.org/10.3233/HIS-120161
28. BR Rajakumar, A George, (2012) A new adaptive mutation technique for genetic algorithm, In: proceedings of IEEE International Conference on Computational Intelligence and Computing Research (ICCIC), pp. 1–7, December 18–20, Coimbatore, India. https://doi.org/10.1109/ICCIC.2012.6510293
29. McCall J (2005) Genetic algorithms for modelling and optimisation. J Comput Appl Math 184(1):205–222
30. Almansour NA, Syed HF, Khayat NR, Altheeb RK, Olatunji SO (2016) Neural network and support vector machine for the prediction of chronic kidney disease: a comparative study. Comput Biol Med 109:101–111
31. LeCun Y, Kavukvuoglu K, Farabet C (2010) Convolutional networks and applications in vision, In: Circuits and Systems, International Symposium on, pp. 253–256
32. Song M, Park H, Shin K-S (2019) Attention-based long short-term memory network using sentiment lexicon embedding for aspect-level sentiment analysis in Korean. Inf Process Manage 56(3):637–653
33. Thompson JJ, Leung BHM, Blair MR, Taboada M (2017) Sentiment analysis of player chat messaging in the video game StarCraft 2: extending a lexicon-based model. Knowl Based Syst 137:149–162