Computers, Materials & Continua

Tech Science Press

DOI: 10.32604/cmc.2023.035237
Article

Computational Linguistics with Optimal Deep Belief Network Based Irony Detection in Social Media

Manar Ahmed Hamza[1,*], Hala J. Alshahrani[2], Abdulkhaleq Q. A. Hassan[3], Abdulbaset Gaddah[4], Nasser Allheeib[5], Suleiman Ali Alsaif[6], Badriyya B. Al-onazi[7] and Heba Mohsen[8]

[1]Department of Computer and Self Development, Preparatory Year Deanship, Prince Sattam bin Abdulaziz University, AlKharj, Saudi Arabia
[2]Department of Applied Linguistics, College of Languages, Princess Nourah bint Abdulrahman University, P. O. Box 84428, Riyadh, 11671, Saudi Arabia
[3]Department of English, College of Science and Arts at Mahayil, King Khalid University, Abha, 62217, Saudi Arabia
[4]Department of Computer Sciences, College of Computing and Information System, Umm Al-Qura University, Makkah, 24211, Saudi Arabia
[5]Department of Information Systems, College of Computer and Information Sciences, King Saud University, Riyadh, Saudi Arabia
[6]Department of Computer, Deanship of Preparatory Year and Supporting Studies, Imam Abdulrahman Bin Faisal University, P. O. Box 1982, Dammam, 31441, Saudi Arabia
[7]Department of Language Preparation, Arabic Language Teaching Institute, Princess Nourah bint Abdulrahman University, P. O. Box 84428, Riyadh, 11671, Saudi Arabia
[8]Department of Computer Science, Faculty of Computers and Information Technology, Future University in Egypt, New Cairo, 11835, Egypt
*Corresponding Author: Manar Ahmed Hamza. Email: ma.hamza@psau.edu.sa

Received: 13 August 2022; Accepted: 14 October 2022

Abstract: Computational linguistics refers to an interdisciplinary field associated with the computational modelling of natural language and studying appropriate computational methods for linguistic questions. The number of social media users has been increasing over the last few years, which have allured researchers' interest in scrutinizing the new kind of creative language utilized on the Internet to explore communication and human opinions in a better way. Irony and sarcasm detection is a complex task in Natural Language Processing (NLP). Irony detection has inferences in advertising, sentiment analysis (SA), and opinion mining. For the last few years, irony-aware SA has gained significant computational treatment owing to the prevalence of irony in web content. Therefore, this study develops Computational Linguistics with Optimal Deep Belief Network based Irony Detection and Classification (CLODBN-IRC) model on social media. The presented CLODBN-IRC model mainly focuses on the identification and classification of irony that exists in social media. To attain this, the presented CLODBN-IRC model performs different stages of pre-processing and TF-IDF feature extraction. For irony detection and classification, the DBN model is exploited in this work. At last, the hyperparameters of the DBN model are optimally modified by improved artificial bee colony optimization (IABC) algorithm. The experimental validation of the presented CLODBN-IRC method can be tested by making use of benchmark dataset. The simulation outcomes highlight the superior outcomes of the presented CLODBN-IRC model over other approaches.

Keywords: Computational linguistics; natural language processing; deep learning; irony detection; social media

1 Introduction

The advancement of information technologies has allowed new goals of intelligence via human-centric meaning, and it is simple as a question-and-answer or a forum customer review [1]. The rapid growth rate of huge amounts of data produces several subjective information sources. Sentiment analysis (SA) is considered an active study topic in information retrieval, natural language processing (NLP), and data mining [2]. SA monitors and provides the understanding of information relevant to customer feedback and public views on various attributes and entities, like business perspectives, social networks, and many services or products [3]. It even assesses people's attitudes, opinions, and emotions against attributes and entities classified as neutral, negative, or positive. SA was broadly studied in many fields, which encompass various features like n-gram features [4]. The terms sarcasm and irony were frequently utilized interchangeably in spite of their delicate variances in meaning. Precise irony recognition becomes significant for mass media analysis. For instance, existing models fail to identify irony results in the low act for SA, as the existence of irony frequently causes polarity reversal [5]. Additionally, irony detection becomes significant for security services for discriminating effective menaces from ironic comments. Conversely many text classification errands, irony detection becomes a difficult task that needs to conclude the hidden, ironic intent, which is not attained by syntactic or SA of the textual contents [6].

As per linguistics research, the irony was the incongruity articulated among statements conveyed in a piece of text and the context. Sentiment polarity contrast was a usually realized type of irony over Twitter [7]. Irony detection means classifying a part of the text as non-ironic or ironic [8]. Prevailing studies express irony detection as a common supervised learning text classifier issue. Methods for irony detection on Twitter are coarsely 35, categorized into 3 classes: rule-related techniques, deep neural network (DNN) techniques, and classical feature-oriented machine learning approaches. In work, rule-related and classical feature-related machine learning (ML) techniques were presented for irony detection. Recently, specific methods for analysing sentiment were provided for fundamental vocabulary analysis methods [9]. ML methods have reached appropriate results. Later, applying deep learning (DL) as the base to classify and identify the automated SA has the greatest overall effect because of DL's multilayer techniques with a sensitive procedure to classify the data [10]. Now, DL can be implemented in several NLP areas like speech, text, and voice and offers precise outcomes. Text classification becomes an important issue in social media and NLP.

This study develops Computational Linguistics with Optimal Deep Belief Network based Irony Detection and Classification (CLODBN-IRC) model on Social Media. The presented CLODBN-IRC model mainly focuses on the identification and classification of irony that exists in social media. To attain this, the presented CLODBN-IRC model performs different stages of pre-processing and TF-IDF feature extraction. For irony detection and classification, the DBN model is exploited in this work. At last, the hyperparameters of the DBN model are optimally modified by improved artificial bee colony optimization (IABC) algorithm. The experimental validation of the presented CLODBN-IRC method can be tested by making use of benchmark dataset.

2 Literature Review

In [11], the main aim was to address irony detection (IRD) in Twitter concerning many degrees of imbalanced distribution among classes. The author depends on the emotIDM IRD method. The author assessed it against skewed Twitter datasets and benchmark corpora accumulated for simulating a real-time dispersal of ironic tweets. The author performed a set of classifier experimentations intended to fix the effect of class imbalance on identifying irony. The author assessed the act of IRD when diverse scenarios were taken into account. Tasneem et al. [12] present a technique for IRD in tweets. The author emphasizes an ensemble of preprocessing approaches for solving the idiosyncratic natures and noisy tweet characteristics. Also, the author used a fruitful amalgamation of various hand-crafted features and n-gram features in a suitable supervised classifier technique. The author uses opinionated lexicons, lexical syntax, and parts-of-speech (POS) to extract the handcrafted features and uses a feature-selecting method to select optimal combinations. In [13], the author uses advanced DL techniques for managing the issue of classifying the FL as mentioned above forms. Suggestively extending our earlier work, the author devises a NN technique that creates a recently devised pre-trained transformer-related network structure that can be further improved with the utility and develop a recurrent CNN. Hence, data preprocessing can be kept to a minimum.

In [14], the author proposes IRD as an alternative to a transfer learning (TL) task in which supervised learning on irony-labelled text can be improved with knowledge transported from exterior SA sources. The author devises 3 TL-related techniques to use sentiment knowledge for enhancing the attention system of RNNs techniques to capture hidden paradigms for incongruity. The core discoveries were initially identifying implicit incongruity; transporting deep sentiment features is considered the possible way. Utilizing sentiment knowledge from exterior sources becomes a very active method for enhancing IRD. Hee et al. [15] introduce the structure of a manually annotated irony corpus related to a finely-grained annotation method that enables the detection of several irony kinds. The author carried out a sequence of binary classifier experimentations for automatic IRD utilizing an SVM that uses various sets of features and compares this approach to a DL method that depends on pre-trained word embeddings and an LSTM network. Assessment of a held-out corpus displays that the SVM method outpaces the NN method and aids in joining syntactic, lexical, and semantic information sources.

Naseem et al. [16] introduce the detection of sarcasm and irony in mass media posts via transformer-related deep, intelligent contextual embedding (T-DICE) that enhances noise in contexts. It resolves the language ambiguities like word sentiments, polysemy, syntax, and semantics by compiling embeddings. T-DICE was sent to attention-related Bidirectional LSTM (BiLSTM) to find the sentiment of a post. Buzea et al. [17] formulate a new corpus and method for evaluating and detecting the irony in online news in Romanian. The author presents a supervised ML mechanism related to a Romanian non-ironic and ironic news corpus, physically marked and a Romanian phrasebook of commonly utilized words.

3 The Proposed Model

In this study, a new CLODBN-IRC technique was formulated to identify and classify irony in social media. To attain this, the presented CLODBN-IRC model performs different stages of pre-processing and TF-IDF feature extraction. For irony detection and classification, the DBN model is exploited in this work. At last, the hyperparameters of the DBN method are modified optimally by improved artificial bee colony optimization (IABC) algorithm. Fig. 1 depicts the block diagram of the CLODBN-IRC approach.

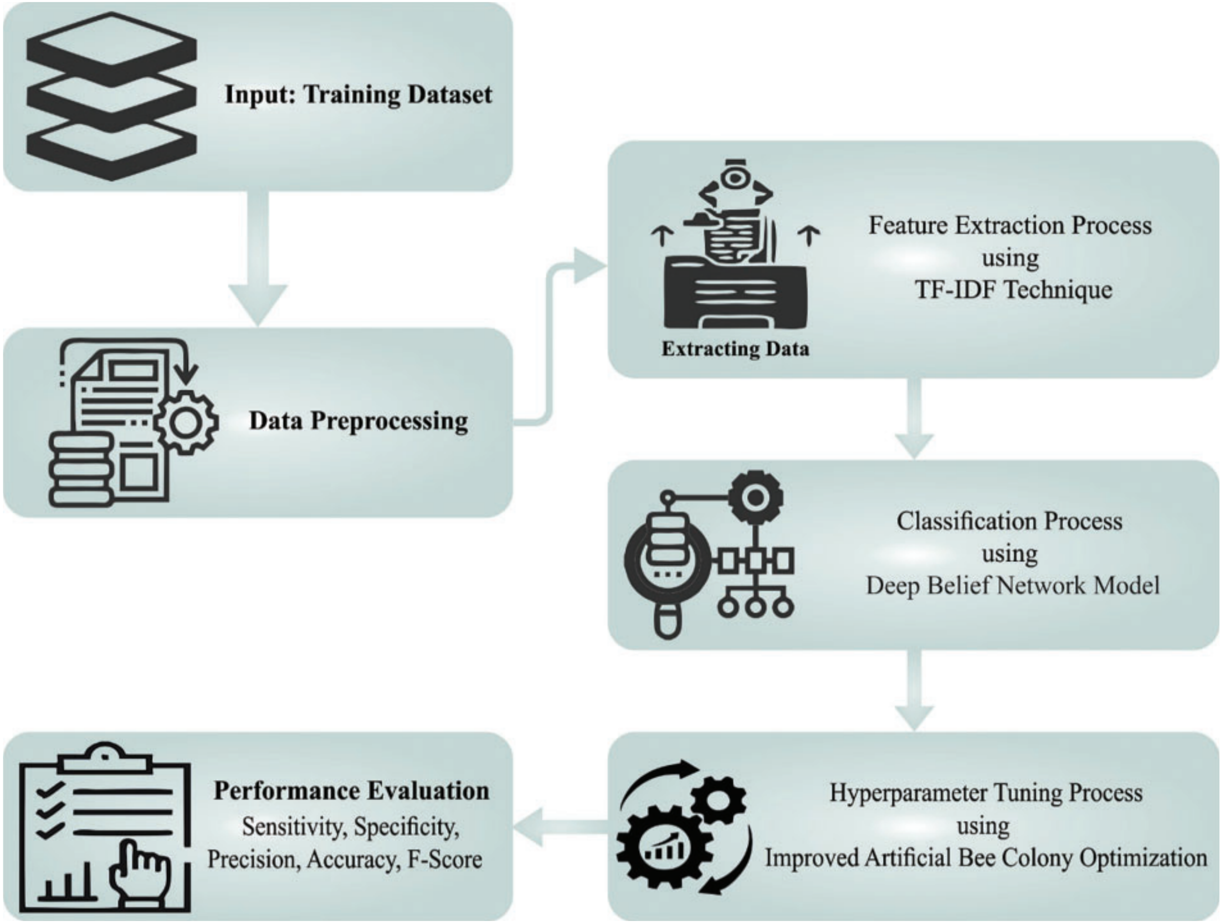

Input: Training Dataset
Data Preprocessing
Extracting Data
Feature Extraction Process
using
TF-IDF Technique
Classification Process
using
Deep Belief Network Model
Hyperparameter Tuning Process
using
Improved Artificial Bee Colony Optimization
Performance Evaluation
Sensitivity, Specificity,
Precision, Accuracy, F-Score

Figure 1: Block diagram of CLODBN-IRC approach

3.1 Data Pre-processing

The data preprocessing stage comprises NLP-based phases that normalize the text and prepare them for analysis. It encompasses different phases, as illustrated in the following.

- Tokenization: Dividing the original texts into elements pieces is the tokenization stage of NLP. There exists a pre-determined rule for the tokenization of the document into word. The tokenization stage can be executed in Python with the help of the SpaCy library.
- Stop Words Removal: Words such as “a” and “the” that appear very commonly were not pertinent to the context of the E-mail and made noise in the textual dataset. Such words are termed stop words, and they are filtered from the text that is treated. We exploited the “NLTK” Python library to remove the stop words from the texts.
- Punctuation Removal: Punctuation involves (comma (,), full stop (.), and brackets) to clarify meaning and separate sentences. We employ the “NLTK” library for punctuation removal.

3.2 TF-IDF Feature Extraction

Afterwards, eliminate irrelevant data, and the explained list of words is changed to numbers. The TF-IDF technique was executed for accomplishing this work [18]. Term Frequency has many incidences of words from the document, and IDF refers to the ratio of the entire amount of documents and the document count comprising the term. The famous and straightforward approach to extracting features, including text data termed as bag-of-words (BoW) technique of text. A BoW technique is a process of feature extraction in the text to utilize from the modeling, namely ML techniques. A BoW was a demonstration of text which explains the event of words in the document. It includes 2 things such as (i) A measure of the occurrence of identified words and (ii) A vocabulary of identified words.

It can be extracting features on the fundamental of Eqs. (1)–(6). At this point, *tf* denotes the term frequency and *df* implies the document frequency.

$$TFIDF = tf * \left(\frac{1}{df}\right) \tag{1}$$

$$TFIDF = tf * Inverse\,(df) \tag{2}$$

$$TFIDF\,(t, d, D) = TF\,(t, d)\,.IDF\,(t, D) \tag{3}$$

$$TFIDF\,(t, d) = log\frac{N}{|d\varepsilon Dt\varepsilon D|} \tag{4}$$

Extracting features in DL with context of words becomes crucial. This approach utilized for this drive is word2vec NN-based technique. Eq. (5) provided under depicts that word2vec achieves the word context with utilize of probability measured. The D denotes the pair-wise design of a group of words, and (w, c) refers word context pair drawn in the huge set D.

$$P\,(D = 1|w, c1: k) = \frac{1}{1 + e^{-\left(w\cdot c_1 + w\cdot c_2 + \ldots + w\cdot c\right)}k} \tag{5}$$

The multi-word context was different from word2vec, as demonstrated in Eq. (6). The variable-length context can be measured by provided under mathematics.

$$P\,(D = 1|w, c) = \frac{1}{1 + e^{-s(w,c)}} \tag{6}$$

3.3 DBN Based Irony Detection

For irony detection and classification, the DBN model is exploited in this work. DBN was stacked through a sequence of RBM layer-wise. RBM can be classified into 2 layers (hidden and visible) [19]. Amid the two layers, all the hidden and visible neurons are interconnected with connection weight. However, the neuron in the layer is not interconnected. A RBM contains three learning hyperparameters $\theta = \{b, c, w\}$. b_i refers to the visible layer bias. Hidden layer offset represents c_j. wij, signifying the connection weight among hidden and visible units. The amount of neurons in hidden and visual layers represents j and i.

RBM refers to an energy-oriented mechanism, and the energy function is described using the following equation:

$$E\,(v, h; \theta) = -\sum\nolimits_i b_i v_i - \sum\nolimits_j c_j h_j - \sum\nolimits_{ij} v_i w_{ij} h_j \tag{7}$$

The joint likelihood distribution of RBM depends on the energy function as follows:

$$Z\,(\theta) = \sum\nolimits_{vh} exp\,(-E\,(v, h; \theta)) \tag{8}$$

$$p(v, h; \theta) = \frac{1}{Z\,(\theta)} exp\,(-E\,(v, h; \theta)) \tag{9}$$

The marginal likelihood distribution of RBM is demonstrated below:

$$p(v;\theta) = -\sum\nolimits_h p(v, h; \theta) \tag{10}$$

Consequently, the conditional probability of RBM is attained:

$$p(h_j|v;\theta) = \frac{p(v, h; \theta)}{p(v;\theta)} \tag{11}$$

$$p(v_i|h;\theta) = \frac{p(v, h; \theta)}{p(h;\theta)} \tag{12}$$

$$p(h_j = 1|v;\theta) = sigmoid\left(\sum\nolimits_{i=1}^{n} v_i w_{ij} + c_j\right) \tag{13}$$

$$p(v_i = 1|h;\theta) = sigmoid\left(\sum\nolimits_{j=1}^{m} w_{ij} h_j + b_i\right) \tag{14}$$

Loss function of RBM:

$$J(\theta) = \langle\frac{1}{N}\sum\nolimits_i logp\,(x_i;\theta)\rangle_{data} - \langle logZ\,(\theta)\rangle_{model} \tag{15}$$

K-step Gibbs sampling (CD-K algorithm) was applied for initializing the network weight. For obtaining the optimum network super parameter, viz., the distribution $\langle logZ(\theta)\rangle_{model}$ of the model is considerably closer to the distribution $\langle\frac{1}{N}\sum_i logp(x_i;\theta)\rangle_{data}$ of the sample dataset, the maximal likelihood estimate of the model is performed. The SGD algorithm is used for updating the hyperparameter, as follows, in which η refers to the learning rate in the SGD approach, θ^t indicates the gradient of RBM at the t-th training:

$$\theta^{t+1} = \theta^t - \nabla_\theta J(\theta^t) \tag{16}$$

During real-time enforcement, the capability of a single RBM to characterize complex raw datasets was not frequently optimistic that needed more than one RBM for stacking to a depth confidence network for extracting features layer-wise for stimulating the original distribution. The architecture of DBN has been demonstrated in Fig. 2. The training procedure of DBN is categorized into reverse finetuning and forward unsupervised pre-learning. we adopted hierarchical greedy learning in the forward unsupervised pre-learning phase. Afterwards, the pretraining adds a classification layer to the final hidden layer, and the weights of DBN are finetuned by minimalizing the errors among the labeled dataset and the estimated output value via the backpropagation model. At last, the training is done, and the performance of the network is tested through a test dataset.

3.4 Hyperparameter Optimization

Here, the hyperparameters of the DBN model are modified optimally by the IABC method. ABC approach is a SI-based stochastic and metaheuristic optimization technique [20]. This approach is motivated by the natural behaviors of honey bees in colonies that search for food sources. The artificial bees of the colony were classified into three major groups: onlooker bees, scout bees, employed bees, and. The major part of the ABC approach is shown below:

Initialization: The presented model involves a population of food sources or candidate solutions. Every food source comprises of multi-dimension vector. Firstly, the ABC initiates the search procedure with an arbitrarily distributed population, and it is given as follows:

$$x_{ij} = x_j^{\min} + rand \times (x_j^{\max} - x_j^{\min}) \tag{17}$$

In Eq. (17), i and j are carefully chosen within $[1, SN]$ and $[1, D]$ correspondingly. SN denotes the food source count, D denotes the dimension count, rand indicates an arbitrary uniform value within [0, 1], and $x_j^{\min}$ and x_j^{max} show the lower and upper bounds, correspondingly.

Employed bee stage: here, every bee is related to one certain food source (x_i) and later generates novel food sources (v_i) via altering the food sources (x_i) in the memory based on neighboring search, and it is evaluated in the following:

$$v_{ij} = x_{ij} + \varphi_{ij} \times (x_{ij} - x_{kj}) \tag{18}$$

In Eq. (18), x_i (or v_i) indicates the J^{th} component of x_i (or v_i), j shows an arbitrary index, x_{kj} denotes the J^{th} component of food source (x_k) chosen arbitrarily from the population, and φ_{ij} specifies an arbitrary uniform value within $[-1, 1]$. Here, food sources (x_i) and (v_i) are compared and later employed, the best ones. Thus, they return to their hive and share the data on novel food sources with onlooker bees via waggle dances.

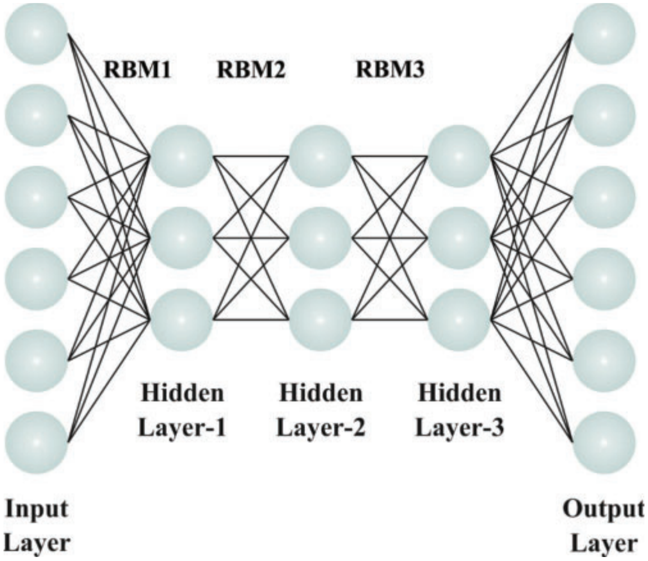


Figure 2: Structure of DBN

Onlooker bee stage: it estimates the nectar data obtained from each employed bee and select a food source with probability proportional to its nectar amount. The values of probabilistic selection p_i are evaluated for an l^{th} food source in the following:

$$p_i = \frac{f^{it_i}}{\Sigma_j fit_j} j = 1, 2, \ldots, SN \tag{19}$$

$$fit_i = \begin{cases} \dfrac{1}{1 + f_i} & f_i \geq 0 \\ 1 + abs(f_i) & f_i < 0 \end{cases} \tag{20}$$

Now, fit_i and f_i indicate the fitness and objective values of food source i, correspondingly. Therefore, the best food source is more probable to be chosen and upgraded in the onlooker bee stage. When the onlooker bee chooses a food source and produces a novel food source related to Eq. (18), then same as the employed bee stage, it implements a greedy selection among x_i and v_i. As illustrated, the update search formula in the employed bee is similar to the one in the onlooker bee stage. The major variation among themselves is that in the employed bee stage same as each food source undergoes the update procedure, whereas, in the onlooker bee stage, only the solution has the opportunity to be chosen according to the probability values and has the chance to be upgraded.

Scout bee stage: it is potential that the onlooker and employed bees visit a food source that they could not optimize. As a result, for a predefined number of cycles (determined using a limited variable), a food source could not be optimized, later food sources are abandoned, and the employed bee related to it becomes scout bees. Then, 138, this scout bee generates a novel solution based on Eq. (17) and transformed into employed bees. According to the abovementioned explanation, the pseudocode of the ABC approach is demonstrated in Algorithm 1.

Algorithm 1: Pseudocode of ABC Algorithm

```
Set parameters. Maximum iteration, SN, limit (maximal amount of trials)
Initializing the population of food source $x_i$ whereby $i = 1, SN$ and estimate their qualities
for every food source, $i$ do set $Trial_i = 0$ end, whereas $i = 1, \ldots, SN$
Set iteration = 1 /*iteration count ABC iteration*/
Repeat
/* Employed bee stage */
For $i = 1$ to $SN$, do
        Produce a new food source $v_i$ for the employed bee stage based on Eq. (18) and estimate its quality
        Employ greedy selection mechanisms among $x_i$ and $v_i$ to select the best one
        If food source $x_i$ didn't enhance, then $trial_i = trial_i + 1$ or else $trial_i = 0$
End
Calculate the likelihood value of $p_i$ based on Eq. (19) for the food source according to the fitness value
    /* Onlooker bee stage */
    $T = 0, i = 0$
    Repeat
        If random $< p_i$
            Produce a novel food source $v_i$ for the onlooker bee based on Eq. (18) and estimate the fitness
            Employ greedy mechanisms among $x_i$ and $v_i$ for selecting the best one
              If food source $x_i$ didn't enhance, then $trial_i = trial_i + 1$ or else $trial_i = 0$
              $T = T + 1$
        End
      Until ($T = SN$)
        / * $s_{coutbee}$ stage */
  $If_{\max(trial_i)>\text{limit},then}$
    Replace_(x_i) with a newly produced food source based on Eq. (17)
  End
  Store the better food source
  iteration = iteration + 1
  Until (iteration = iteration + 1)
```

In this study, the IABC algorithm is formed by the use of opposition-Based Learning (OBL) [21]. It is utilized to optimize the Convergence curve by attaining the opposite solution, later compararing the 2 solutions, and taking the best one.

Definition: Opposite number when x refers to a real number ∈ [lb, ub], then $\overline{x}$ can be shown as

$$\overline{x} = ub + lb - x \tag{21}$$

In Eq. (21), *lb* and *ub* indicate the lower and upper bounds of the problem.

Definition: Opposite Vector when x refers to a multi-dimension vector, then x is generalized:

$$\overline{x_j} = ub_j + lb_j - x_j \tag{22}$$

In Eq. (22), lb_j, ub_j are the lower and upper limits to the *j-th* component of x.

4 Results and Discussion

In this study, the ironic detection performance of the CLODBN-IRC method is simulated using a dataset comprising 4618 samples. The dataset holds 2222 ironic samples and 2396 non-ironic samples, as given in Table 1.

Table 1: Dataset details

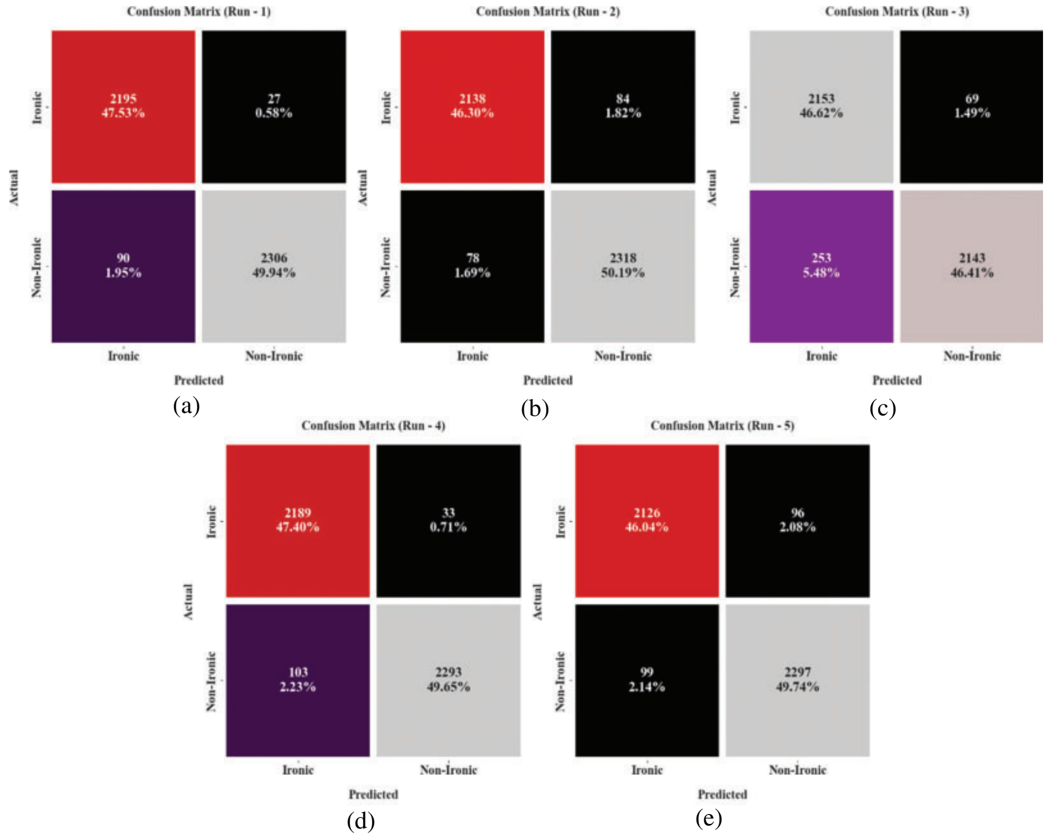
Class	No. of samples
Ironic	2222
Non-ironic	2396
Total number of samples	**4618**

Fig. 3 signifies the confusion matrices generated by the CLODBN-IRC model in five runs. With run-1, the CLODBN-IRC model has recognized 2195 samples into the ironic class and 2306 samples into the non-ironic class. In addition, with run-2, the CLODBN-IRC technique has recognized 2138 samples into the ironic class and 2318 samples into the non-ironic class. Concurrently, with run-3, the CLODBN-IRC approach has recognized 2153 samples into the ironic class and 2143 samples into the non-ironic class. Finally, with run-4, the CLODBN-IRC algorithm has recognized 2189 samples into the ironic class and 2293 samples into the non-ironic class.

Table 2 provides overall ironic detection results of the CLODBN-IRC model under distinct runs.

Fig. 4 highlights the detection outcomes of the CLODBN-IRC model under run-1. The CLODBN-IRC model has identified ironic samples with $accu_y$ of 97.47%, $prec_n$ of 96.06%, $reca_l$ of 98.78%, F_{score} of 97.40%, and AUC_{score} of 97.51%. Also, The CLODBN-IRC approach has identified non-ironic samples with $accu_y$ of 97.47%, $prec_n$ of 98.84%, $reca_l$ of 96.24%, F_{score} of 97.53%, and AUC_{score} of 97.51%. Additionally, The CLODBN-IRC method has identified average samples with $accu_y$ of 97.47%, $prec_n$ of 97.45%, $reca_l$ of 97.51%, F_{score} of 97.46%, and AUC_{score} of 97.51%.

Fig. 5 highlights the detection outcomes of the CLODBN-IRC method under run-2. The CLODBN-IRC approach has identified ironic samples with $accu_y$ of 96.49%, $prec_n$ of 96.48%, $reca_l$ of 96.22%, F_{score} of 96.35%, and AUC_{score} of 96.48%. Further, The CLODBN-IRC approach has identified non-ironic samples with $accu_y$ of 96.49%, $prec_n$ of 96.50%, $reca_l$ of 96.74%, F_{score} of 96.62%, and AUC_{score} of 96.48%. Besides, The CLODBN-IRC methodology has identified average samples with $accu_y$ of 96.49%, $prec_n$ of 96.49%, $reca_l$ of 96.48%, F_{score} of 96.49%, and AUC_{score} of 96.48%.



(a) (b) (c) (d) (e)

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

Table 2: Result analysis of CLODBN-IRC approach with distinct measures and runs

Labels	Accuracy	Precision	Recall	F-Score	AUC score
Run-1					
Ironic	97.47	96.06	98.78	97.40	97.51
Non-Ironic	97.47	98.84	96.24	97.53	97.51
Average	**97.47**	**97.45**	**97.51**	**97.46**	**97.51**
Run-2					
Ironic	96.49	96.48	96.22	96.35	96.48
Non-Ironic	96.49	96.50	96.74	96.62	96.48
Average	**96.49**	**96.49**	**96.48**	**96.49**	**96.48**

(Continued)

Table 2: Continued

Labels	Accuracy	Precision	Recall	F-Score	AUC score
Run-3					
Ironic	93.03	89.48	96.89	93.04	93.17
Non-Ironic	93.03	96.88	89.44	93.01	93.17
Average	**93.03**	**93.18**	**93.17**	**93.03**	**93.17**
Run-4					
Ironic	97.06	95.51	98.51	96.99	97.11
Non-Ironic	97.06	98.58	95.70	97.12	97.11
Average	**97.06**	**97.04**	**97.11**	**97.05**	**97.11**
Run-5					
Ironic	95.78	95.55	95.68	95.62	95.77
Non-Ironic	95.78	95.99	95.87	95.93	95.77
Average	**95.78**	**95.77**	**95.77**	**95.77**	**95.77**

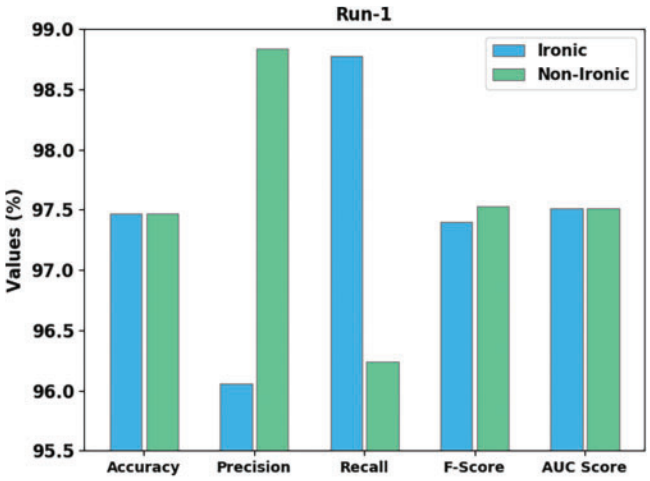


Figure 4: Result analysis of CLODBN-IRC approach under run1

Fig. 6 portrays the detection outcomes of the CLODBN-IRC algorithm under run-3. The CLODBN-IRC method has identified ironic samples with $accu_y$ of 93.03%, $prec_n$ of 89.48%, $reca_l$ of 96.89%, F_{score} of 93.04%, and AUC_{score} of 93.17%. Also, The CLODBN-IRC approach has identified non-ironic samples with $accu_y$ of 93.03%, $prec_n$ of 96.88%, $reca_l$ of 89.44%, F_{score} of 93.01%, and AUC_{score} of 93.17%. Furthermore, The CLODBN-IRC method has identified average samples with $accu_y$ of 93.03%, $prec_n$ of 93.18%, $reca_l$ of 93.17%, F_{score} of 93.03%, and AUC_{score} of 93.17%.

Fig. 7 illustrates the detection outcomes of the CLODBN-IRC approach under run-4. The CLODBN-IRC algorithm has identified ironic samples with $accu_y$ of 97.06%, $prec_n$ of 95.51, $reca_l$ of 98.51%, F_{score} of 96.99%, and AUC_{score} of 97.11%. Also, The CLODBN-IRC method has identified non-ironic samples with $accu_y$ of 97.06%, $prec_n$ of 98.58%, $reca_l$ of 95.70%, F_{score} of 97.12%, and AUC_{score}

of 97.11%. Also, The CLODBN-IRC technique has identified average samples with $accu_y$ of 97.06%, $prec_n$ of 97.04%, $reca_l$ of 97.11%, F_{score} of 97.05%, and AUC_{score} of 97.11%.

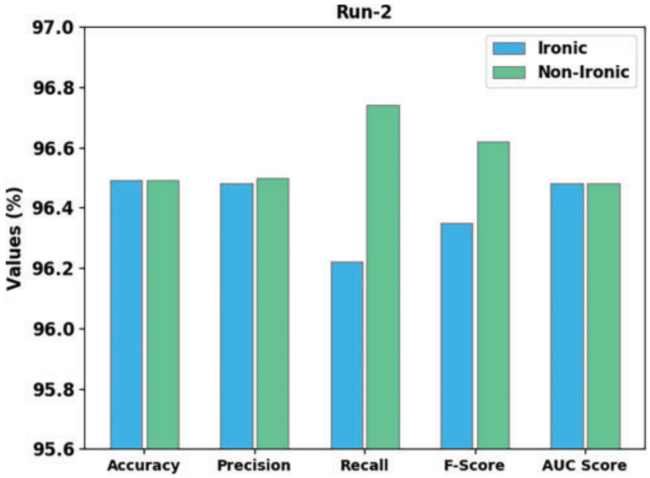


Figure 5: Result analysis of CLODBN-IRC approach under run2

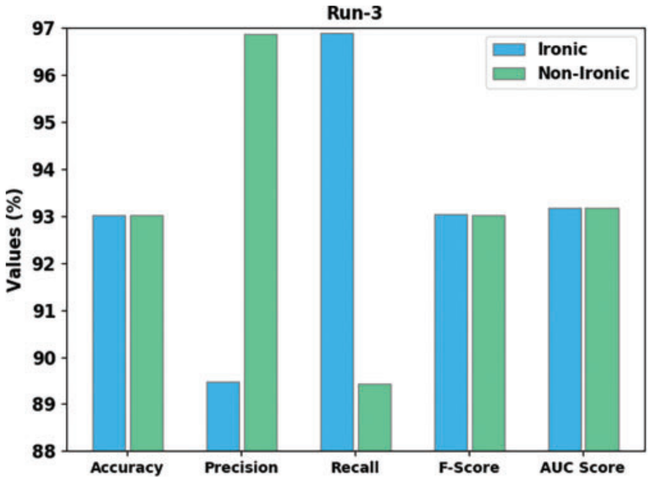


Figure 6: Result analysis of CLODBN-IRC approach under run3

Fig. 8 demonstrates the detection outcomes of the CLODBN-IRC algorithm under run-5. The CLODBN-IRC technique has identified ironic samples with $accu_y$ of 95.78%, $prec_n$ of 95.55, $reca_l$ of 95.68%, F_{score} of 95.62%, and AUC_{score} of 95.77%. Also, The CLODBN-IRC method has identified non-ironic samples with $accu_y$ of 95.78%, $prec_n$ of 95.99%, $reca_l$ of 95.87%, F_{score} of 95.93%, and AUC_{score} of 95.77%. Furthermore, The CLODBN-IRC approach has identified average samples with $accu_y$ of 95.78%, $prec_n$ of 95.77%, $reca_l$ of 95.77%, F_{score} of 95.77%, and AUC_{score} of 95.77%.

The training accuracy (TRA) and validation accuracy (VLA) gained by the CLODBN-IRC technique on the test dataset is shown in Fig. 9. The experimental result denotes the CLODBN-IRC approach maximal values of TRA and VLA. Seemingly the VLA is greater than TRA.

The training loss (TRL) and validation loss (VLL) reached by the CLODBN-IRC algorithm on the test dataset are given in Fig. 10. The experimental result indicates the CLODBN-IRC method has established the least values of TRL and VLL. Particularly, the VLL is lesser than TRL.

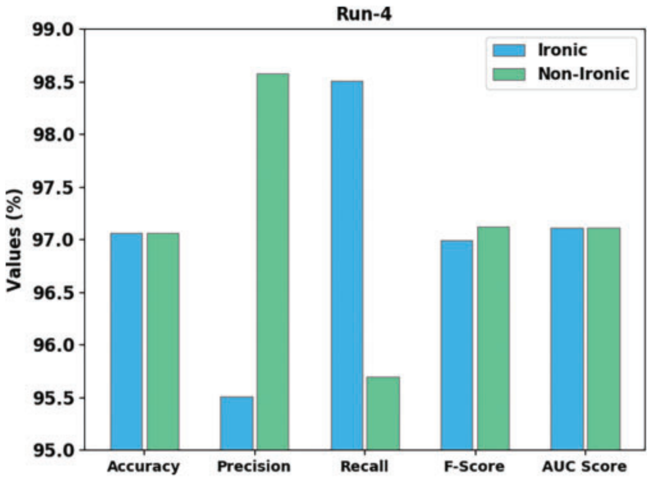


Figure 7: Result analysis of CLODBN-IRC approach under run4

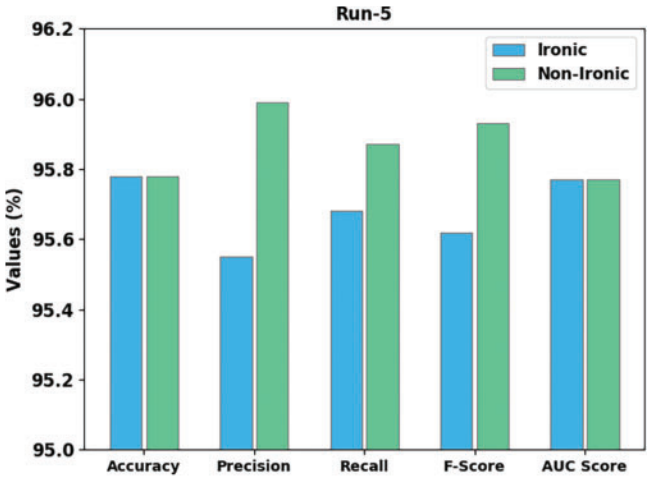


Figure 8: Result analysis of CLODBN-IRC approach under run5

A clear precision-recall inspection of the CLODBN-IRC method in the test dataset is depicted in Fig. 11. The figure is implicit that the CLODBN-IRC approach has resulted in enhanced precision-recall values in every class label.

A brief ROC study of the CLODBN-IRC methodology on the test dataset is depicted in Fig. 12. The outcomes exemplified the CLODBN-IRC algorithm has displayed its capability in classifying different class labels in the test dataset.

To ensure the enhanced performance of the CLODBN-IRC method, a comparative study is made in Table 3 and Fig. 13 [10,14]. The experimental outcomes implied that the CLODBN-IRC model had improved performance over other existing techniques. For example, in terms of $accu_y$, the CLODBN-IRC method has attained increased $accu_y$ of 97.47%, whereas the Bi-LSTM, CNN-LSTM, LSTM, NB, SVM-linear, and CNN models have obtained reduced $accu_y$ of 87.48%, 85.84%, 84.78%, 88.75%, 86.63%, and 84.74% respectively.

Besides, with respect to $pren_c$, the CLODBN-IRC method has gained increased $pren_c$ of 97.45%, whereas the Bi-LSTM, CNN-LSTM, LSTM, NB, SVM-linear, and CNN methodologies have acquired reduced $pren_c$ of 84.90%, 84.06%, 84.22%, 89.22%, 89.91%, and 87.69% correspondingly. Lastly, with respect to $reca_l$, the CLODBN-IRC method has reached an increased $reca_l$ of 97.51%, whereas the Bi-LSTM, CNN-LSTM, LSTM, NB, SVM-linear, and CNN techniques have exhibited reduced $reca_l$ of 90.99%, 84.45%, 92.91%, 86.70%, 91.36%, and 86.08% correspondingly. Therefore, the CLODBN-IRC model can be employed for productive and automated ironic classification.

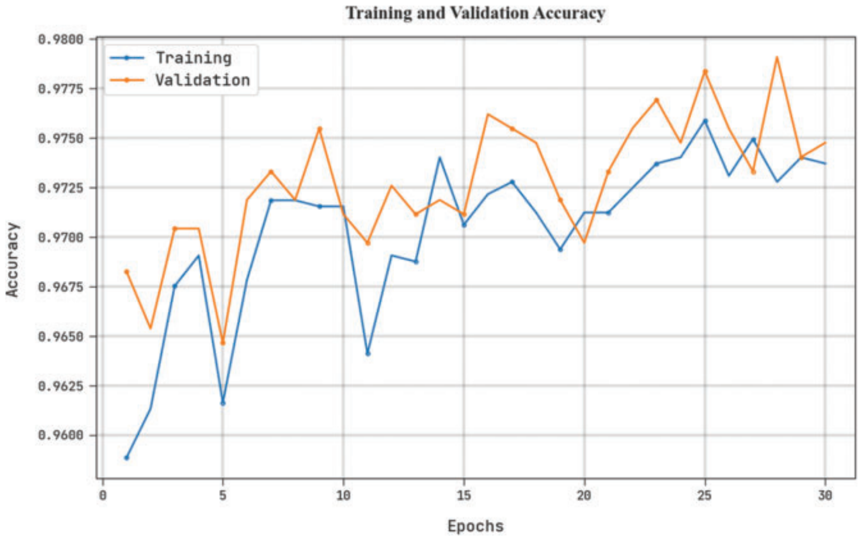


Figure 9: TRA and VLA analysis of CLODBN-IRC approach

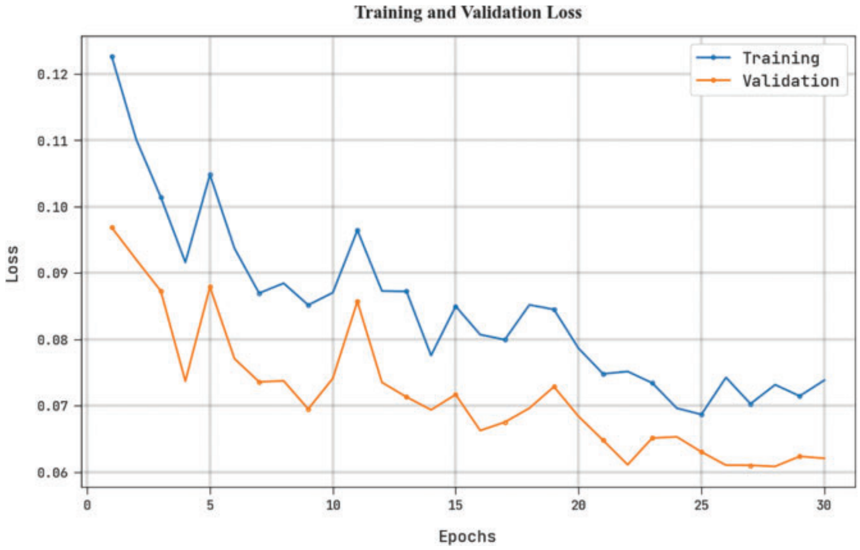


Figure 10: TRL and VLL analysis of the CLODBN-IRC approach

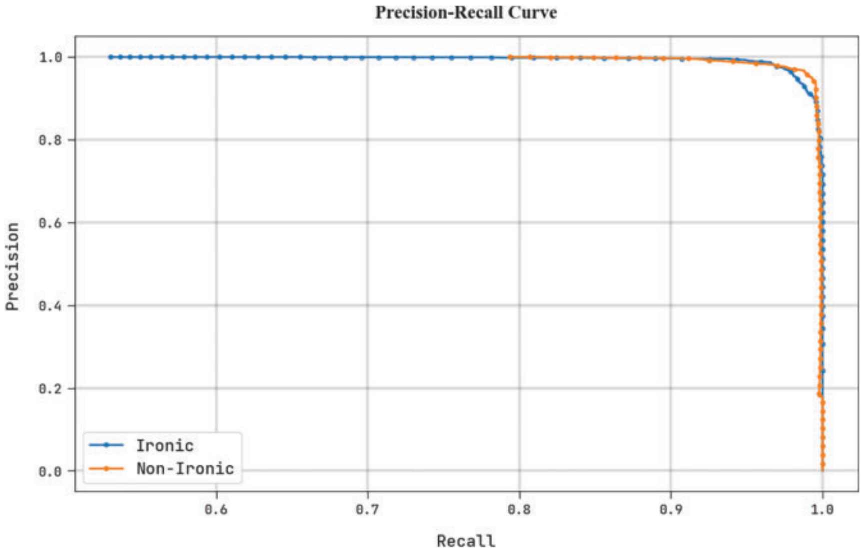


Figure 11: Precision-recall analysis of the CLODBN-IRC approach

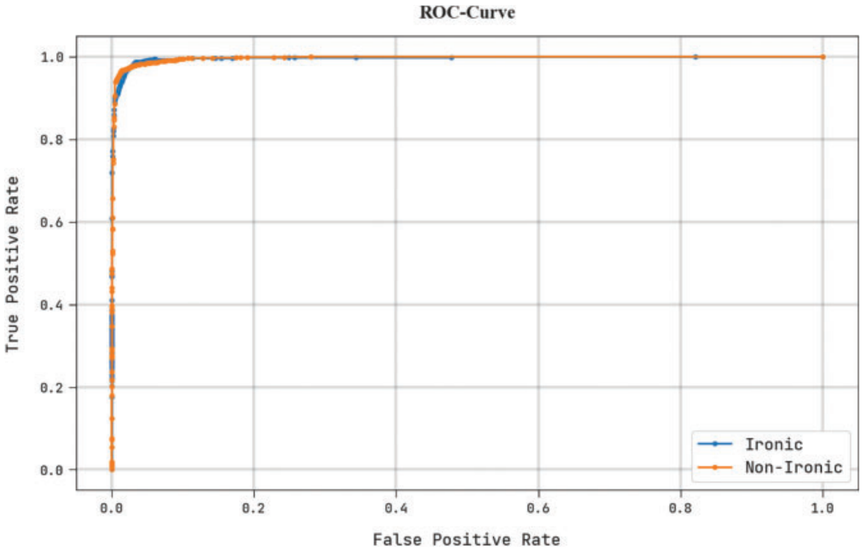


Figure 12: ROC curve analysis of CLODBN-IRC approach

Table 3: Comparative analysis of CLODBN-IRC technique with existing algorithms

Methods	Accuracy	Precision	Recall	F-score
CLODBN-IRC	97.47	97.45	97.51	97.46
Bi-LSTM	87.48	84.90	90.99	88.80
CNN-LSTM	85.84	84.06	84.45	90.76
LSTM	84.78	84.22	92.91	92.56

(Continued)

Table 3: Continued

Methods	Accuracy	Precision	Recall	F-score
Naïve Bayes	88.75	89.22	86.70	86.26
SVM-linear	86.63	89.91	91.36	89.14
CNN	84.74	87.69	86.08	87.91

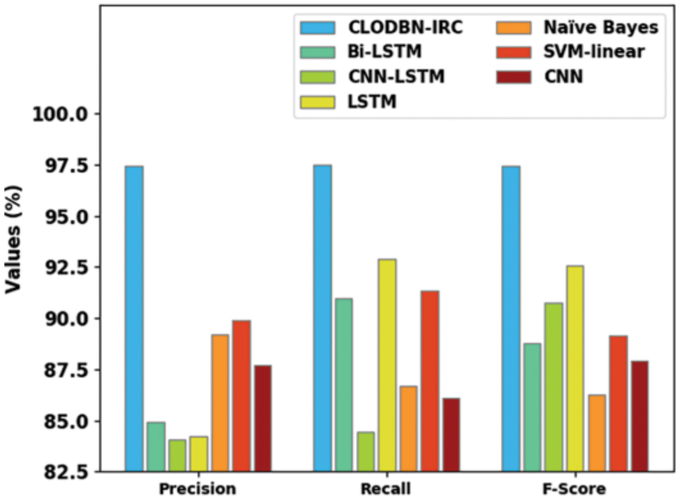


Figure 13: Comparative analysis of CLODBN-IRC approach with existing algorithms

5 Conclusion

In this study, a new CLODBN-IRC algorithm was projected for the identification and classification of irony exists in social media. To attain this, the presented CLODBN-IRC model performs different stages of pre-processing and TF-IDF feature extraction. For irony detection and classification, the DBN model is exploited in this work. At last, the hyperparameters of the DBN method were optimally modified by improved artificial bee colony optimization (IABC) algorithm. The presented CLODBN-IRC method's experimental validation can be tested using a benchmark dataset. The simulation outcomes highlight the superior outcomes of the presented CLODBN-IRC method over other methods. Thus, the CLODBN-IRC method can be employed for real time irony detection in social media content. In future, a hybrid DL model can be applied to boost the detection efficacy of the CLODBN-IRC method.

Funding Statement: The authors extend their appreciation to the Deanship of Scientific Research at King Khalid University for funding this work through Small Groups Project under Grant Number (120/43). 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: (22UQU4320484DSR33).

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

References

[1] H. Calvo, O. J. Gambino and C. V. G. Mendoza, "Irony detection using emotion cues," *Computación y Sistemas*, vol. 24, no. 3, pp. 1281–1287, 2020.

[2] J. Á. González, L. F. Hurtado and F. Pla, "Transformer based contextualization of pre-trained word embeddings for irony detection in Twitter," *Information Processing & Management*, vol. 57, no. 4, pp. 102262, 2020.

[3] M. Sykora, S. Elayan and T. W. Jackson, "A qualitative analysis of sarcasm, irony and related# hashtags on Twitter," *Big Data & Society*, vol. 7, no. 2, pp. 2053951720972735, 2020.

[4] R. Akula and I. Garibay, "Interpretable multi-head self-attention architecture for sarcasm detection in social media," *Entropy*, vol. 23, no. 4, pp. 394, 2021.

[5] B. Ghanem, P. Rosso and F. Rangel, "An emotional analysis of false information in social media and news articles," *ACM Transactions on Internet Technology (TOIT)*, vol. 20, no. 2, pp. 1–18, 2020.

[6] C. V. Hee, E. Lefever and V. Hoste, "We usually don't like going to the dentist: Using common sense to detect irony on Twitter," *Computational Linguistics*, vol. 44, no. 4, pp. 793–832, 2018.

[7] M. C. Buzea, S. T. Matu and T. Rebedea, "Automatic fake news detection for romanian online news," *Information—An International Interdisciplinary Journal*, vol. 13, no. 3, pp. 151, 2022.

[8] K. Sundararajan and A. Palanisamy, "Probabilistic model based context augmented deep learning approach for sarcasm detection in social media," *International Journal of Advanced Science and Technology*, vol. 29, no. 6, pp. 8461–8479, 2020.

[9] P. Rosso, F. Rangel, I. H. Farías, L. Cagnina, W. Zaghouani *et al.,* "A survey on author profiling, deception, and irony detection for the Arabic language," *Language and Linguistics Compass*, vol. 12, no. 4, pp. e12275, 2018.

[10] Z. L. Chia, M. Ptaszynski, F. Masui, G. Leliwa and M. Wroczynski, "Machine learning and feature engineering-based study into sarcasm and irony classification with application to cyberbullying detection," *Information Processing & Management*, vol. 58, no. 4, pp. 102600, 2021.

[11] D. I. H. Farías, R. Prati, F. Herrera and P. Rosso, "Irony detection in Twitter with imbalanced class distributions," *Journal of Intelligent & Fuzzy Systems*, vol. 39, no. 2, pp. 2147–2163, 2020.

[12] F. Tasneem, J. Naim and A. N. Chy, "Harnessing ensemble of data preprocessing and hand-crafted features for irony detection in tweets," in *23rd Int. Conf. on Computer and Information Technology (ICCIT)*, Dhaka, Bangladesh, pp. 1–6, 2020.

[13] R. A. Potamias, G. Siolas and A. G. Stafylopatis, "A transformer-based approach to irony and sarcasm detection," *Neural Computing and Applications*, vol. 32, no. 23, pp. 17309–17320, 2020.

[14] S. Zhang, X. Zhang, J. Chan and P. Rosso, "Irony detection via sentiment-based transfer learning," *Information Processing & Management*, vol. 56, no. 5, pp. 1633–1644, 2019.

[15] C. V. Hee, E. Lefever and V. Hoste, "Exploring the fine-grained analysis and automatic detection of irony on Twitter," *Language Resources and Evaluation*, vol. 52, no. 3, pp. 707–731, 2018.

[16] U. Naseem, I. Razzak, P. Eklund and K. Musial, "Towards improved deep contextual embedding for the identification of irony and sarcasm," in *Int. Joint Conf. on Neural Networks (IJCNN)*, Glasgow, UK, pp. 1–7, 2020.

[17] M. C. Buzea, S. T. Matu and T. Rebedea, "Automatic irony detection for romanian online news," in *24th Int. Conf. on System Theory, Control and Computing (ICSTCC)*, Sinaia, Romania, pp. 72–77, 2020.

[18] M. Moussa and I. I. Măndoiu, "Single cell RNA-seq data clustering using TF-IDF based methods," *BMC Genomics*, vol. 19, no. 6, pp. 31–45, 2018.

[19] B. H. Patil, "Effect of optimized deep belief network to patch-based image inpainting forensics," *International Journal of Swarm Intelligence Research (IJSIR)*, vol. 13, no. 3, pp. 1–21, 2022.

[20] A. M. Abdulazeez, D. M. Hajy, D. Q. Zeebaree and D. A. Zebari, "Robust watermarking scheme based LWT and SVD using artificial bee colony optimization," *Indonesian Journal of Electrical Engineering and Computer Science*, vol. 21, no. 2, pp. 1218–1229, 2021.

[21] A. G. Hussien, "An enhanced opposition-based salp swarm algorithm for global optimization and engineering problems," *Journal of Ambient Intelligence and Humanized Computing*, vol. 13, no. 1, pp. 129–150, 2022.