applied sciences

MDPI

Article

Hosted Cuckoo Optimization Algorithm with Stacked Autoencoder-Enabled Sarcasm Detection in Online Social Networks

Dalia H. Elkamchouchi [1], Jaber S. Alzahrani [2], Mashael M. Asiri [3], Mesfer Al Duhayyim [4,*], Heba Mohsen [5], Abdelwahed Motwakel [6], Abu Sarwar Zamani [6] and Ishfaq Yaseen [6]

1 Department of Information Technology, College of Computer and Information Sciences, Princess Nourah Bint Abdulrahman University, P.O. Box 84428, Riyadh 11671, Saudi Arabia; dhelkamchouchi@pnu.edu.sa
2 Department of Industrial Engineering, College of Engineering at Alqunfudah, Umm Al-Qura University, Mecca 24382, Saudi Arabia; Jszahrani@uqu.edu.sa
3 Department of Computer Science, College of Science & Art at Mahayil, King Khalid University, Abha 62529, Saudi Arabia; abusharara@kku.edu.sa
4 Department of Computer Science, College of Sciences and Humanities-Aflaj, Prince Sattam bin Abdulaziz University, Al-Kharj 16278, Saudi Arabia
5 Department of Computer Science, Faculty of Computers and Information Technology, Future University in Egypt, New Cairo 11835, Egypt; hmohsen@fue.edu.eg
6 Department of Computer and Self Development, Preparatory Year Deanship, Prince Sattam bin Abdulaziz University, Al-Kharj 16278, Saudi Arabia; aismaeil@psau.edu.sa (A.M.); AZamani@psau.edu.sa (A.S.Z.); iyaseen@psau.edu.sa (I.Y.)
* Correspondence: m.alduhayyim@psau.edu.sa

Abstract: Sarcasm detection has received considerable interest in online social media networks due to the dramatic expansion in Internet usage. Sarcasm is a linguistic expression of dislikes or negative emotions by using overstated language constructs. Recently, detecting sarcastic posts on social networking platforms has gained popularity, especially since sarcastic comments in the form of tweets typically involve positive words that describe undesirable or negative characteristics. Simultaneously, the emergence of machine learning (ML) algorithms has made it easier to design efficacious sarcasm detection techniques. This study introduces a new Hosted Cuckoo Optimization Algorithm with Stacked Autoencoder-Enabled Sarcasm Detection and Classification (HCOA-SACDC) model. The presented HCOA-SACDC model predominantly focuses on the detection and classification of sarcasm in the OSN environment. To achieve this, the HCOA-SACDC model pre-processes input data to make them compatible for further processing. Furthermore, the term frequency-inverse document frequency (TF-IDF) model is employed for the useful extraction of features. Moreover, the stacked autoencoder (SAE) model is utilized for the recognition and categorization of sarcasm. Since the parameters related to the SAE model considerably affect the overall classification performance, the HCO algorithm is exploited to fine-tune the parameters involved in the SAE, showing the novelty of the work. A comprehensive experimental analysis of a benchmark dataset is performed to highlight the superior outcomes of the HCOA-SACDC model. The simulation results indicate that the HCOA-SACDC model accomplished enhanced performance over other techniques.

Keywords: sarcasm; Internet of Things; stacked autoencoder; hosted cuckoo; data classification

Citation: Elkamchouchi, D.H.; Alzahrani, J.S.; Asiri, M.M.; Al Duhayyim, M.; Mohsen, H.; Motwakel, A.; Zamani, A.S.; Yaseen, I. Hosted Cuckoo Optimization Algorithm with Stacked Autoencoder-Enabled Sarcasm Detection in Online Social Networks. *Appl. Sci.* **2022**, *12*, 7119. https://doi.org/10.3390/app12147119

Academic Editors: Tuan Anh Nguyen and Francisco Airton Silva

Received: 16 April 2022
Accepted: 12 July 2022
Published: 14 July 2022

1. Introduction

The emergence of web 2.0 and Online Social Networking (OSN) sites has provided new dimensions to the communication world and has given ample opportunity to extract provable and countable patterns from public opinions [1]. Therefore, these networks are utilized as powerful methods to identify popularity and trends in various topics, such as politics, entertainment, social or economic problems, and the environment [2]. Not

only do people use standard languages, such as German, Spanish, and English, but they also try to be more advanced by using emotion icons otherwise called hashtags #, URLs, emoticons, etc. [3].

With the huge volume of content being generated on social networking platforms and the necessity to evaluate it carefully, text classification techniques have been presented in order to handle this sophisticated emergence [4]. In text classification, sarcasm recognition is an important tool that has several implications for numerous fields, including sales, security, and health [5]. Sarcasm means conveying negative opinions through positive words or intensified positive words. On social media, people frequently use sarcasm to express their opinions, and it is inherently tough to analyze not only a machine but also humans [6]. The existence of sarcastic comments has had a crucial impact on sentiment analysis (SA) tasks. For instance, "It is a great feeling to bring a smartphone which has short battery life." is a sarcastic sentence stating negative sentiment regarding battery life utilizing positive words such as "great feeling" [7]. Thus, sarcasm detection is an important tool used to enhance SA task performances. Sarcasm detection has been devised as a binary classifier task for the prediction of whether sentences are non-sarcastic or sarcastic.

Sarcasm is a widely used, well-studied, and well-known topic in linguistics. Despite being part of our speech and so commonly used, it is integrally very difficult for humans and machines to identify sarcasm in text [8]. Since the length of text messages is gradually becoming shorter, the challenge of recognizing sarcasm poses real threats to the efficacy of machine learning algorithms. Hence, it is not important but rather essential to resolve the challenge of sarcasm in text datasets for the refinement and further evolution of different systems applied for sentimental analyses.

Earlier studies on forecasting sarcastic sentences predominantly concentrated on statistical and rule-based methods, utilizing (1) pragmatic and lexical features and (2) the presence of sentiment shifts, punctuation, interjections, etc. [8]. A deep neural network (DNN) grants a technique the ability to study essential features automatically rather than utilizing handcrafted features [9]. Deep learning (DL) techniques are used in numerous natural language processing (NLP) methods, namely, machine translation, question answering, and text summarization [10]. DL methods have been explored in sarcasm detection, resulting in interesting outcomes.

This study introduces a new Hosted Cuckoo Optimization Algorithm with Stacked Autoencoder-Enabled Sarcasm Detection and Classification (HCOA-SACDC) model in the OSN environment. The objective of the HCOA-SACDC method is to determine the existence of sarcasm. To achieve this, the HCOA-SACDC technique pre-processes the input data to make them compatible for further processing. Furthermore, the term frequency-inverse document frequency (TF-IDF) methodology is employed for effective feature extraction. Moreover, the stacked autoencoder (SAE) technique is utilized for the recognition and categorization of sarcasm. Lastly, the HCO approach is exploited to adjust the parameter involved in the SAE, thus increasing detection performance. In the HCO algorithm, significant solutions are created as nests, and the eggs are placed in three varying nests. A comprehensive experimental analysis of a benchmark dataset is performed to highlight the superior outcomes of the HCOA-SACDC model.

2. Literature Review

This section provides a comprehensive study of the present sarcasm detection approaches. Potamias et al. [11] presented advanced DL techniques to tackle the issue of the detection of figurative language (FL) forms. Expanding on earlier work, this work proposed a neural network (NN) approach, building on a recently devised pretrained transformer-related network infrastructure, which was further enriched with the employment and formulation of a recurrent CNN. Hence, data pre-processing is minimal. Pan et al. [12] proposed a BERT architecture-related method, which focuses on intra- and inter-modality incongruities for multimodal sarcasm detection. This was based on the ideology of designing a self-attention mechanism and inter-modality attention to capture inter-modality

incongruity. Moreover, this co-attention system can be employed to model contradictions within text. The incongruity data are utilized for prediction purposes.

Cai et al. [13] concentrated on multimodal sarcasm recognition for Twitter, comprising images and text on Twitter. It considers image attributes, text features, and image features as three modalities and models a multimodal hierarchical fusion technique to address this task. This method initially extracts attribute features and image features, and it uses the bidirectional LSTM network and attribute features to extract text features. Then, the features of the three approaches were rebuilt and merged into a single feature vector for estimation. Reference [14] mainly focuses on recognizing sarcasm in textual conversation from social media platforms and websites. As a result, an interpretable DL technique utilizing gated recurrent units (GRUs) and multi-head self-attention modules was developed. The multi-head self-attention system helps to detect sarcastic cue-words in input data, and the recurrent unit learns long-range dependencies among such cue-words for superior classification of the input data.

Du et al. [15] emphasized examining the content of sarcastic text by making use of several natural language processing (NLP) methods. The argument made here is to detect sarcasm by analyzing the context, which includes the sentiments of the texts that respond to the target text and the expression habits of users. A dual-channel CNN is devised, which scrutinizes not only the semantics of the targeted text but also its sentimental context. Furthermore, SenticNet can be leveraged to include common sense in the LSTM method. The attention system is implemented afterward to consider the expression habits of users. Kamal and Abulaish [16] modeled a new Convolutional and Attention with Bi-directional GRU (CAT-BiGRU) method, which has an input layer, embedded layer, convolution layer, Bi-directional GRU (BiGRU) layer, and two attention layers [17]. The convolution layer extracts SDS-related semantic and syntactic characteristics from the embedded layer; the BiGRU layer retrieves contextual data from the features, which are extracted in succeeding and preceding directions; and the attention layers retrieve SDS-related complete context representation from the input text [18].

3. Design of HCOA-SACDC Model

In this study, a new HCOA-SACDC model was developed to determine the existence of sarcasm in the OSN environment. Firstly, the HCOA-SACDC model pre-processes input data to make them compatible for further processing. Next, the preprocessed data are passed into the TF-IDF technique for effective feature extraction. This is followed by the use of HOC with the SAE model, which is utilized for the recognition and categorization of sarcasm. Figure 1 illustrates the overall process of the proposed HCOA-SACDC technique.

3.1. SAE-Based Classification

The SAE model is utilized for the recognition and categorization of sarcasm [19,20]. Normally, SAE is a type of unsupervised deep learning (DL) method that is organized by a dissimilar autoencoder (AE). The AE comprises a decoder and an encoder. Initially, the encoder layer is useful to translate the input x to a hidden illustration h, viz., defined by $h = f(wx + b)$, where f, w, and b define the activation function, the weighting matrices, and the bias of the existing encoder layer, respectively. Next, the decoding layer is useful to reconstruct x from h, viz., represented by $x' = g(w'h + b')$, where x', g, w', and b' signify the weighting matrix, the simulation outcome, the bias of the existing decoder layer, and the activation function, respectively. Moreover, the wide-ranging training technique of the AE comprises a pretraining stage and a finetuning stage [17]. Firstly, AE minimizes the cost function as follows:

$$L(x) = \frac{1}{2m} \sum_{i=1}^{m} \|x_i - x_i^{'}\|^2 \quad (1)$$

where x_i describes the AE input, which characterizes the ith samples, and m denotes the sample count. $x_i^{'}$ indicates the AE outcomes, and the regeneration of the ith sample and m determines the quantity of samples. SAE can be recognized by adding hidden states.

Consequently, the implementation of a hidden encoder of an existing AE is assumed as the input of the upcoming AE. When the pretraining layer of SAE is completed, the decoding AE is released. Then, the encoder weight of the SAE is interconnected and finetuned using softmax regression. Figure 2 depicts the framework of SAE.

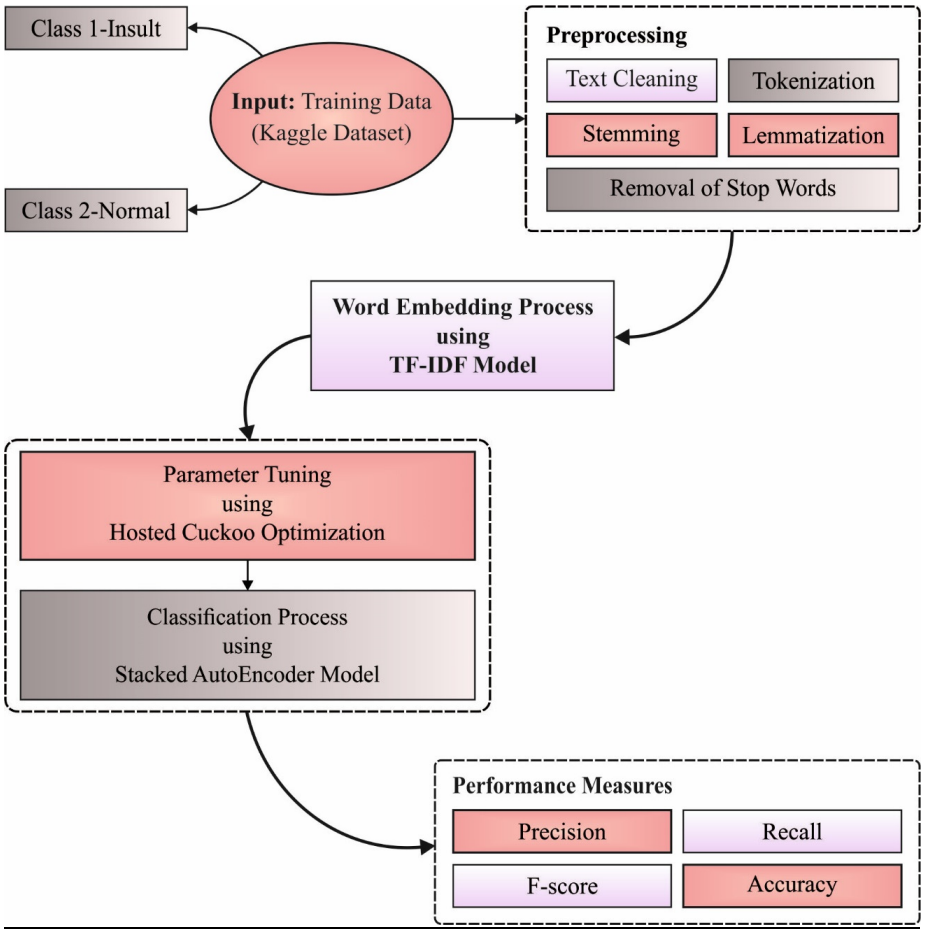

Figure 1. The overall process of HCOA-SACDC technique.

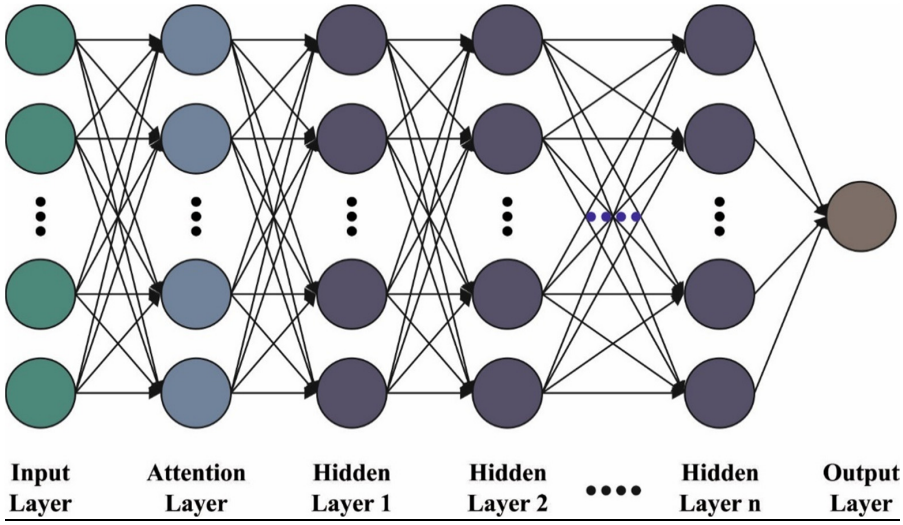

Figure 2. Framework of SAE.

Furthermore, a restricted term is involved in the loss function of the AE. The prolonged loss function is as follows:

$$L(x) = \frac{1}{2m}\sum_{i=1}^{m}\|x_i - x_i^{'}\|^2 + \lambda\|w\|_1 \quad (2)$$

where w represents the weight matrix, and λ is the balance factor. When the quantity of hidden states is high in comparison with the input layer from the input neuron, it usually increases Kullback-Leibler (***KL***) divergence to the loss function of the AE. Consequently, the adopted loss function is as follows:

$$L(x) = \frac{1}{2m}\sum_{i=1}^{m}\|x_i - x_i^{'}\|^2 + \lambda\sum_{j=1}^{H} KL(\rho||\hat{\rho}_j) \quad (3)$$

where H indicates the quantity of concealed states, and ***KL*** shows the ***KL*** discrepancy. ρ describes the sparsity variable quantity, and $\hat{\rho}_j$ defines the activation of the input instance of the jth concealed neuron. Obvious guiding normalization, applicable to some functions, includes the cross-entropy loss function, which is suggested to be included in the loss function of the AE. Thus, the adopted loss function can be expressed as follows:

$$L(x) = \frac{1}{2m}\sum_{i=1}^{m}\|x_i - x_i^{'}\|^2 + \frac{\lambda}{m}\sum_{i=1}^{m}\sum_{j=1}^{c} label(i,j)\cdot log\ pred(i,j) \quad (4)$$

where C indicates the quantity of classes. $label(i,j)$ represents the correct possibility from the jth class of the ith sample. $log\ pred(i,j)$ represents the surveyed probability from the jth class of the ith sample. At that moment, a dissimilarity of SAE is applied in the newly designed technique. Now, the weight and bias values of the SAE are carefully chosen using the suggested technique.

3.2. HCO-Based Parameter Optimization

In the final phase, the HCO algorithm is exploited to adjust the parameters included in the SAE, thus increasing detection performance [21]. The cuckoo optimization algorithm (COA) is a popular optimization method, and it is the strongest one. It is inspired by the actions of a cuckoo bird. They can lay their eggs in the nests of other birds. Some limits are described, and it is optimized to manage various problems, for example, energy dispatch, controller parameters, job shops, cluster computing, system cost, and obtainability. In our work, the COA is improved to resolve system consistency optimization using heterogeneous components and is renamed the HCO algorithm. The probable solution can be created as nests, and the eggs are laid in the nests of different species. This is described below.

Step 1. Initialize the parameters, including the input of the highest cuckoo generation N_{gen} and the number of nests M to be considered.

Step 2. Create the nest. The nest can be created as follows:

$$\begin{cases} Nest_1(r,\ n) = (r_1,\ r_2,\ \ldots, r_m,\ n_1,\ n_2,\ \ldots, n_m) \\ Nest_2(r,\ n) = (r_1,\ r_2,\ \ldots, r_m,\ n_1,\ n_2,\ \ldots, n_m) \\ \qquad\qquad \vdots \\ Nest_M(r,\ n) = (r_1,\ r_2,\ \ldots, r_m,\ n_1,\ n_2,\ \ldots, n_m) \end{cases} \quad (5)$$

where $Nest(n,\ r)$ represents a collection of probable solutions.

Step 3. The limitation is accomplished through the succeeding penalty function:

$$\widetilde{R}_s(r,\ n) = R_s(r,\ n) + \varphi_1 \cdot \text{Max}\ \{0,\ g_1(r,\ n) - V\} +]\varphi_2$$

$$\text{Max}\ \{0,\ g_2(r,\ n) - C\} + \varphi_3 \cdot \text{Max}\ \{0,\ g_3(r,\ n) - W\} \quad (6)$$

Step 4. The cuckoo's egg can be placed according to the novel COA:

$$ELR = \alpha \times \frac{Number\ of\ current\ cuckoo's\ eggs}{Total\ number\ of\ eggs} \times (V_{hi} - V_{low}) \quad (7)$$

where ELR symbolizes the laying radius; α represents an integer value; and V_{hi} and V_{low} represent the upper and lower limits, respectively.

Step 5. The cuckoo's egg is introduced by 3 dissimilar hosts and dissimilar possibilities. Consequently, the cuckoo egg contains 3 dissimilar possibilities to successfully develop and characterize σ_1, σ_2, and σ_3 [0%, 100%], called host quality. This value is subjectively completed by every single generation. Henceforth, the nest is detached into 3 groups, namely, M_1, M_2, and M_3, and these values are subjective. Host quality can be described as follows:

$$\begin{cases} M_1\ nests\ with\ \sigma_1, & where\ M_1 \in \{M\} \\ M_2\ nests\ with\ \sigma_2, & where\ M_2 \in \{M - M_1\} \\ M_3\ nests\ with\ \sigma_3, & where\ M_3 \in \{M - M_2 - M_1\} \end{cases} \quad (8)$$

Step 6. The optimal generation of cuckoo travels to alternative habitats; viz., the optimum solution existing in the forthcoming generation is used to enhance the search solution.

Step 7. Iterate Steps 2–6 until the number of generations $\left(N_{gen}\right)$ is accomplished.

The HCO algorithm computes a fitness function to accomplish better classification performance. It derives a positive integer to characterize the solution candidate. The minimization of the classification error rate is considered as the fitness function, as given in Equation (9). The best solution has a reduced error rate, and the worst solution has an improved error rate (Algorithm 1).

$$fitness(x_i) = ClassifierErrorRate(x_i) = \frac{number\ of\ misclassified\ samples}{Total\ number\ of\ samples} \times 100 \quad (9)$$

Algorithm 1: Pseudocode of HCO algorithm

Input: Parameter initialization: M, N_{gen}
Begin
While $z \leq N_{gen}$
Produce the nests using Equation (5)
Determine the fitness value
Carry out the egg laying
Carry out the chick stage
Migrate cuckoos
End while
Output: Report optimal solutions
End

4. Experimental Validation

In this section, the outcomes of the HCOA-SACDC model are tested using benchmark datasets from the Kaggle repository [22]. The dataset holds 1049 samples under insult class and 2898 samples under normal class. This is a single-class classification problem. The label 0 implies a neutral comment, and 1 implies an insulting comment (neutral is regarded as not belonging to the insult class). The prediction should be a real number in the range from zero to one, where one is a 100% confident prediction that the comment is an insult.

Figure 3 illustrates the confusion matrices presented by the HCOA-SACDC method with dissimilar training/testing (TR/TS) dataset sizes. With a TR/TS dataset of 90:10, the HCOA-SACDC method identified 85 instances of insult and 290 instances of normal. Moreover, with a TR/TS dataset of 80:20, the HCOA-SACDC approach identified 178 instances of insult and 554 instances of normal. Simultaneously, with a TR/TS dataset of 70:30, the HCOA-SACDC approach identified 233 instances of insult and 861 instances of

normal. Concurrently, with a TR/TS dataset of 60:40, the HCOA-SACDC system identified 267 instances of insult and 1170 instances of normal.

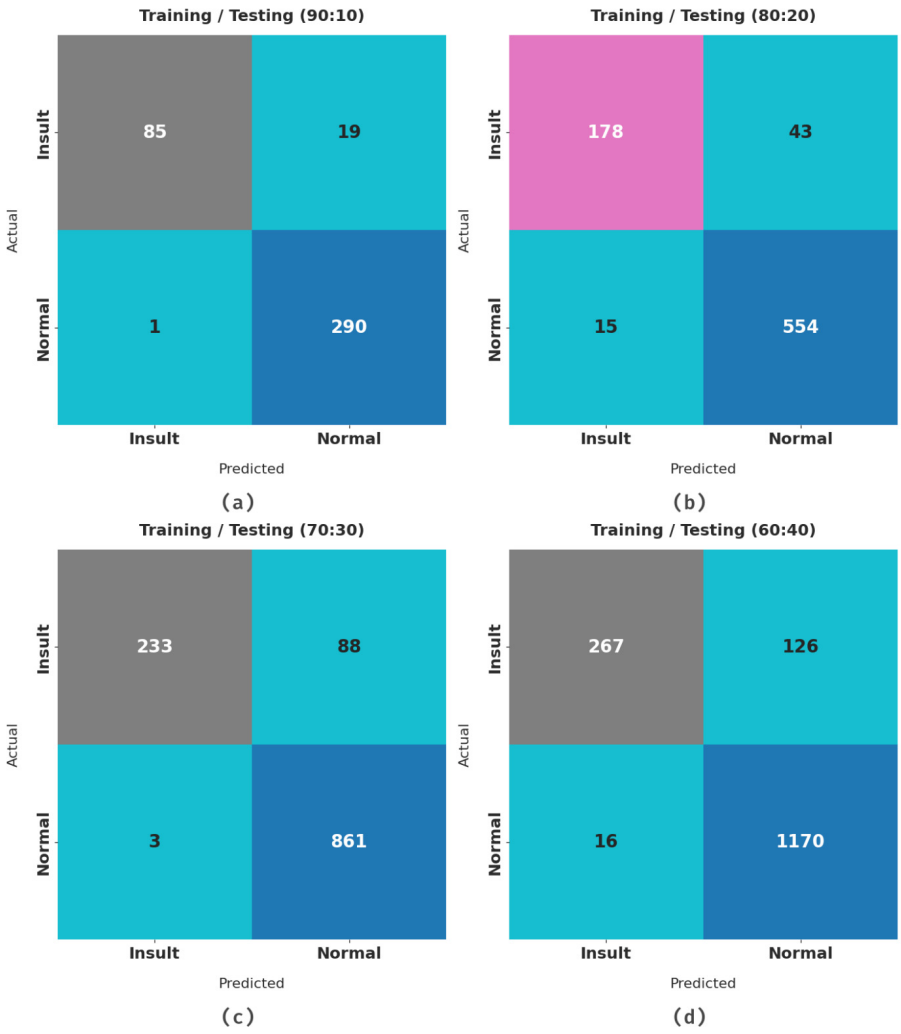

Figure 3. Confusion matrix of HCOA-SACDC system under distinct TR/TS dataset sizes.

Table 1 provides a detailed classification outcome of the HCOA-SACDC technique with various sizes of data. The stimulation results indicate that the HCOA-SACDC system obtained the highest outcome in all aspects. Figure 4 reports a brief $prec_n$ and $reca_l$ inspection of the HCOA-SACDC method with dissimilar TR/TS dataset sizes. The results indicate that the HCOA-SACDC technique accomplishes increasing values of $prec_n$ and $reca_l$. For example, with a TR/TS of 90:10, the HCOA-SACDC method provided $prec_n$ and $reca_l$ values of 96.94% and 90.69%, respectively. Simultaneously, with a TR/TS of 70:30, the HCOA-SACDC system provided $prec_n$ and $reca_l$ values of 94.73% and 86.12%, respectively. Additionally, with a TR/TS of 60:40, the HCOA-SACDC methodology provided $prec_n$ and $reca_l$ values of 92.31% and 83.29%, respectively.

Table 1. Result analysis of HCOA-SACDC approach with distinct measures and TR/TS datasets.

Class Labels	Accuracy	Precision	Recall	Specificity	F-Score
Training/Testing (90:10)					
Insult	94.94	98.84	81.73	99.66	89.47
Normal	94.94	93.85	99.66	81.73	96.67
Average	**94.94**	**96.34**	**90.69**	**90.69**	**93.07**

Table 1. *Cont.*

Class Labels	Accuracy	Precision	Recall	Specificity	F-Score
Training/Testing (80:20)					
Insult	92.66	92.23	80.54	97.36	85.99
Normal	92.66	92.80	97.36	80.54	95.03
Average	**92.66**	**92.51**	**88.95**	**88.95**	**90.51**
Training/Testing (70:30)					
Insult	92.32	98.73	72.59	99.65	83.66
Normal	92.32	90.73	99.65	72.59	94.98
Average	**92.32**	**94.73**	**86.12**	**86.12**	**89.32**
Training/Testing (60:40)					
Insult	91.01	94.35	67.94	98.65	78.99
Normal	91.01	90.28	98.65	67.94	94.28
Average	**91.01**	**92.31**	**83.29**	**83.29**	**86.64**

Figure 5 reports a brief $spec_y$ and F_{score} examination of the HCOA-SACDC method on distinct TR/TS dataset sizes. The results indicate that the HCOA-SACDC system accomplishes increasing values of $spec_y$ and F_{score}. For example, with a TR/TS of 90:10, the HCOA-SACDC approach provided $spec_y$ and F_{score} values of 90.69% and 93.07%, respectively. With a TR/TS of 70:30, the HCOA-SACDC algorithm provided $spec_y$ and F_{score} values of 86.12% and 89.32%, respectively. Finally, with a TR/TS of 60:40, the HCOA-SACDC method provided $spec_y$ and F_{score} values of 83.29% and 86.64%, respectively.

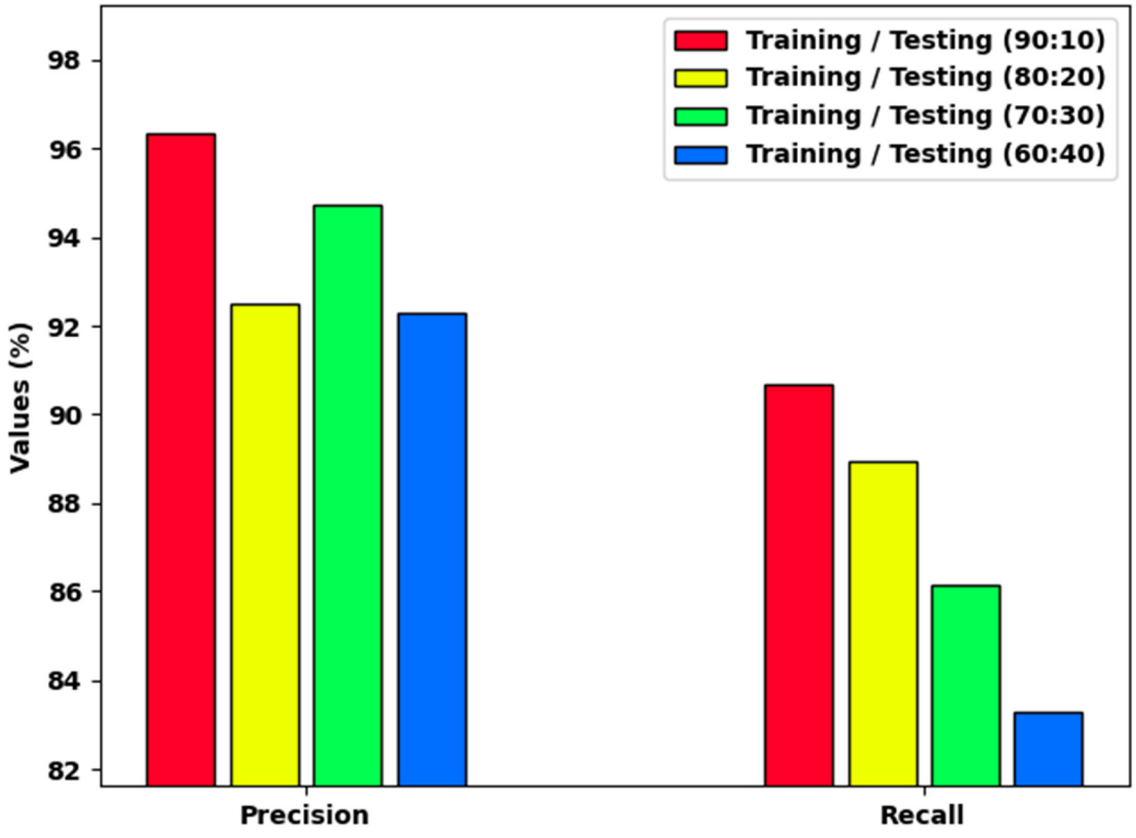

Figure 4. $Prec_n$ and $reca_l$ results of HCOA-SACDC technique using distinct TR/TS dataset sizes.

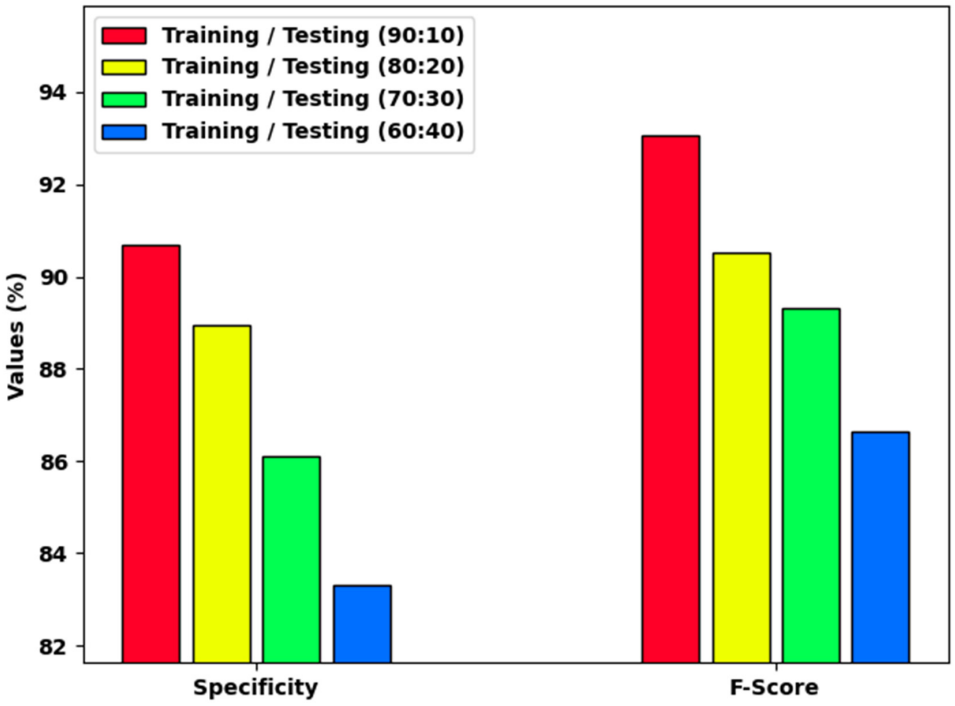

Figure 5. $Spec_y$ and F_{score} results of HCOA-SACDC technique using distinct TR/TS dataset sizes.

Figure 6 demonstrates a detailed acc_y analysis of the HCOA-SACDC system on distinct TR/TS dataset sizes. The results indicate that the HCOA-SACDC method accomplished increasing values of acc_y. For instance, with a TR/TS of 90:10, the HCOA-SACDC approach provided an acc_y of 94.94%. Simultaneously, with a TR/TS of 70:30, the HCOA-SACDC system provided an acc_y of 92.32%. Moreover, with a TR/TS of 60:40, the HCOA-SACDC method provided an acc_y of 91.01%.

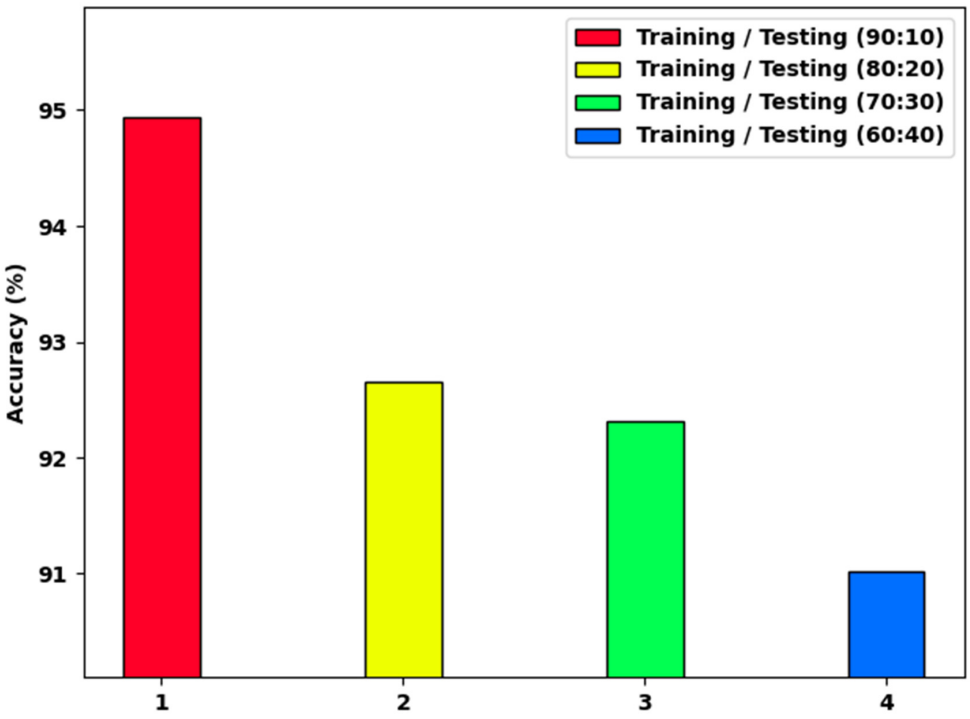

Figure 6. Acc_y analysis of HCOA-SACDC approach using dissimilar TR/TS dataset sizes.

A detailed precision-recall inspection of the HCOA-SACDC method on various forms of datasets is described in Figure 7. It can be observed that the HCOA-SACDC approach obtained maximal precision-recall performance with all datasets.

Next, a comprehensive ROC study of the HCOA-SACDC method using the distinct datasets is described in Figure 8. The results indicate that the HCOA-SACDC approach successfully categorized two different classes, namely, insult and normal, within the test dataset.

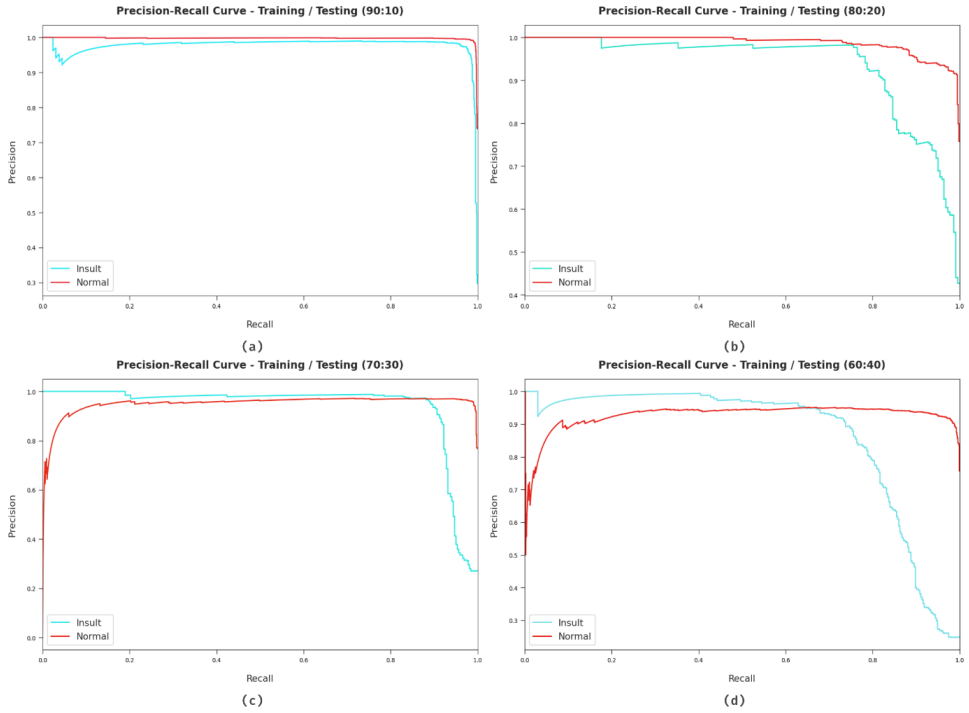

Figure 7. Precision-recall analysis of HCOA-SACDC method using dissimilar TR/TS dataset sizes.

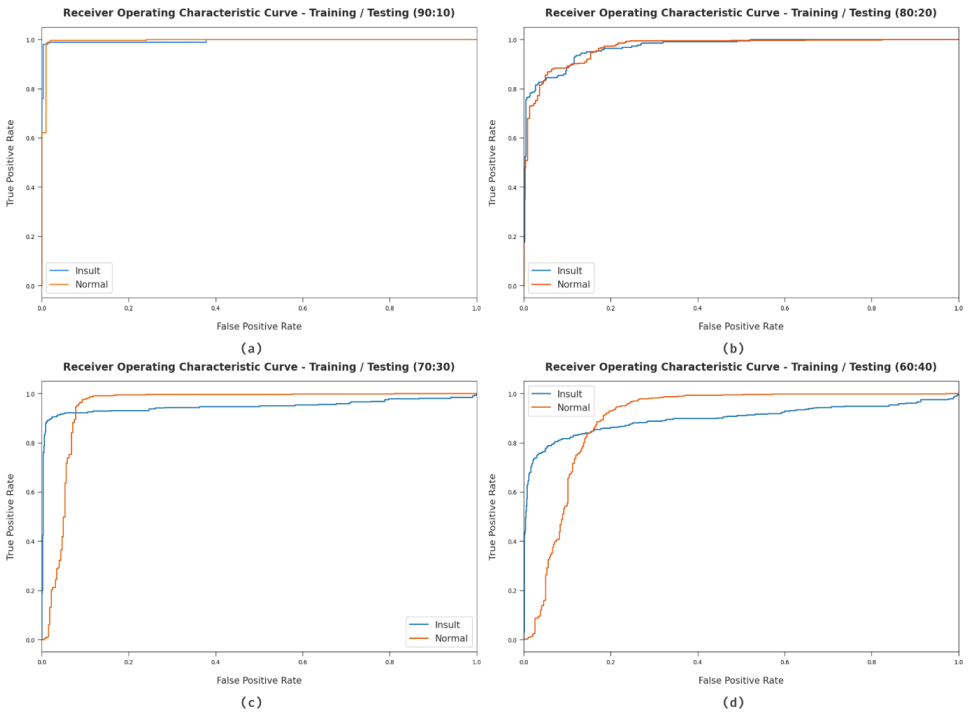

Figure 8. ROC analysis of HCOA-SACDC approach using dissimilar TR/TS dataset sizes.

Figure 9 demonstrates the training and validation accuracy examination of the HCOA-SACDC algorithm using dissimilar TR/TS dataset sizes. The figure shows that the HCOA-SACDC system has maximum training/validation accuracy in the classification of the test dataset. It also shows that the HCOA-SACDC system has low training/accuracy loss in the classification of the test dataset.

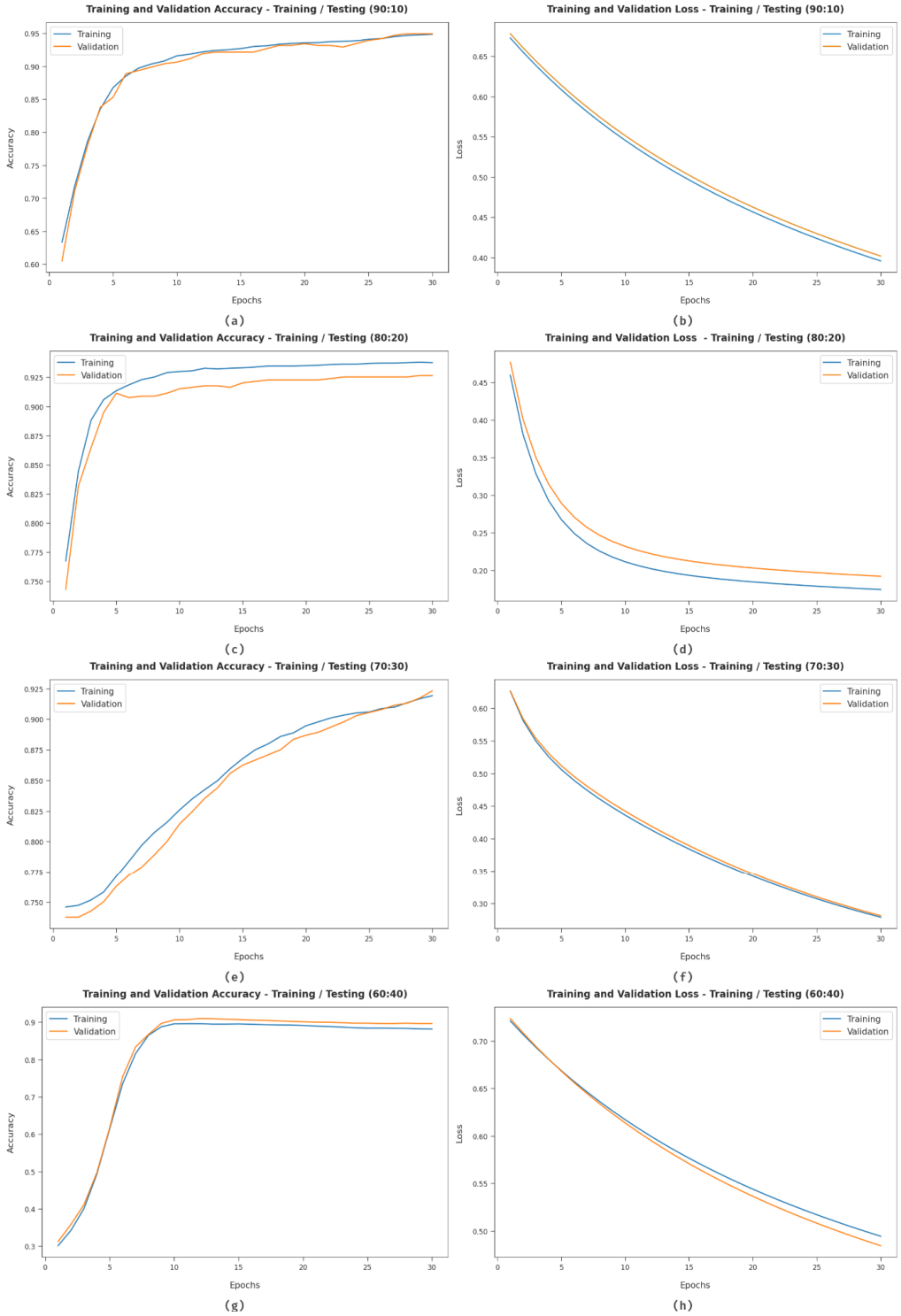

Figure 9. Accuracy and loss graphs of HCOA-SACDC technique using different TR/TS dataset sizes.

To highlight the enhanced outcomes of the HCOA-SACDC method, a brief accuracy analysis with recent methods was conducted, and the results are presented in Table 2 and

Figure 10 [23]. The results indicate that the LSTM and RNN models obtained low accuracies of 81.66% and 81.84%, respectively. This is followed by the B-LSTM and GRU models, which had moderately improved accuracies of 81.66% and 83.36%, respectively.

Table 2. Testing accuracy analysis of HCOA-SACDC technique with recent algorithms.

Methods	Testing Accuracy
B-LSTM Model	83.89
Bi-GRNN Model	93.33
LSTM Model	81.66
RNN Model	81.84
ODLCDC Model	93.76
GRU Model	83.36
HCOA-SACDC	94.94

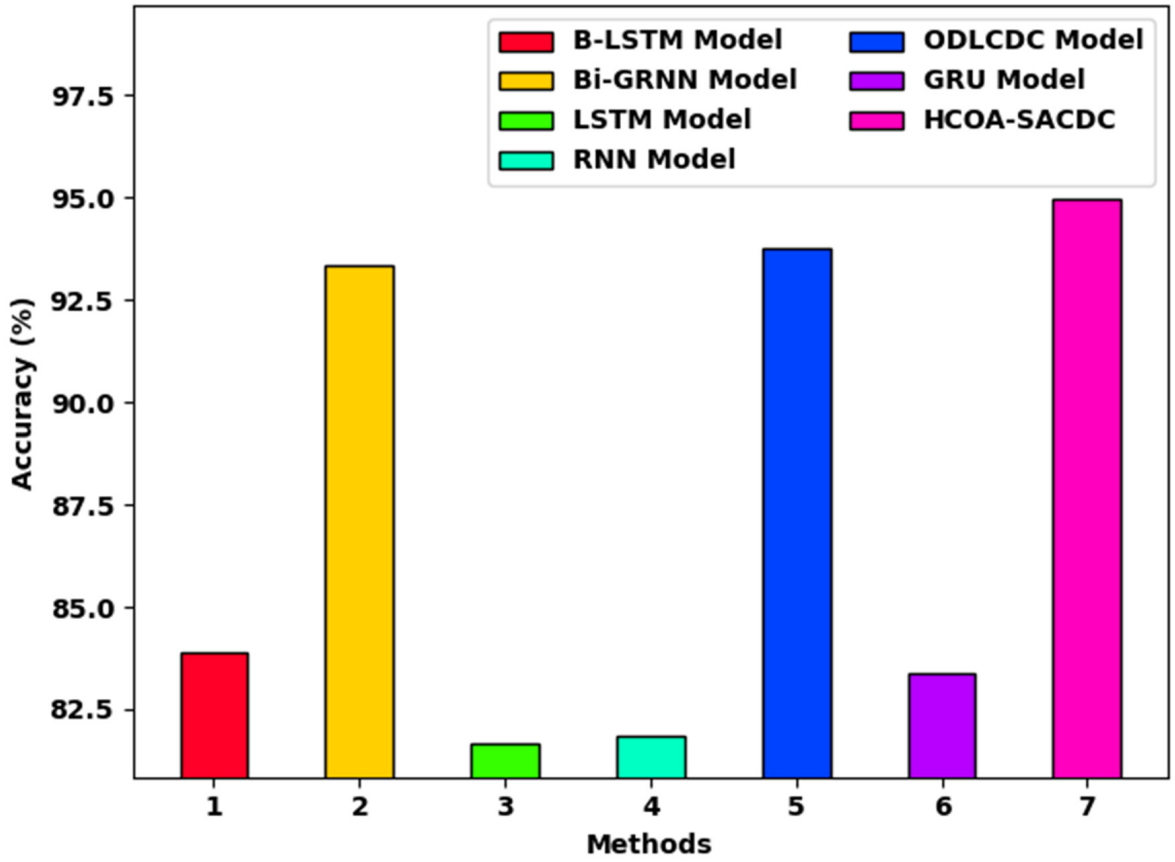

Figure 10. Testing accuracy analysis of HCOA-SACDC technique with recent algorithms.

Additionally, the BiGRNN and ODLCDC models accomplished reasonable accuracies of 93.33% and 93.76%, respectively. However, the HCOA-SACDC model achieved the maximum value with an accuracy of 94.94%. Therefore, the experimental results show that the HCOA-SACDC method has effectual outcomes in comparison to the other methods. The enhanced performance of the proposed method is mainly due to the inclusion of the HCO algorithm, which can optimally select SAE parameters. This helps to considerably reduce computation complexity and to improve the performance of the classification. Thus, the proposed method can be employed for the classification of sarcasm and to ensure security in the OSN environment.

5. Conclusions

In this study, a new HCOA-SACDC model is developed to determine the existence of sarcasm in the OSN environment. The HCOA-SACDC model pre-processes input data to make them compatible for further processing. Furthermore, the TF-IDF method is employed for effective feature extraction. Moreover, the SAE model is utilized for the recognition and categorization of sarcasm. Finally, the HCO approach is exploited to adjust the parameters included in the SAE, thus increasing the detection performance. A comprehensive experimental analysis of a benchmark dataset is carried out to highlight the superior outcomes of the HCOA-SACDC method. The simulation results indicate that the HCOA-SACDC model accomplished enhanced performance over the other methods, with a maximum accuracy of 94.94%. In the future, advanced DL techniques can be utilized to boost the classification results of the HCOA-SACDC model. Additionally, outlier detection and clustering approaches can also be included to further enhance the overall sarcasm detection and classification performance.

Author Contributions: Conceptualization, D.H.E. and J.S.A.; methodology, M.M.A.; software, A.S.Z.; validation, I.Y., M.A.D. and H.M.; formal analysis, A.M.; investigation, A.M.; resources, M.A.D.; data curation, I.Y.; writing—original draft preparation, D.H.E.; J.S.A. and M.A.D.; writing—review and editing, A.M.; visualization, A.S.Z.; supervision, M.A.D.; project administration, J.S.A.; funding acquisition, D.H.E. All authors have read and agreed to the published version of the manuscript.

Funding: The authors extend their appreciation to the Deanship of Scientific Research at King Khalid University for funding this work through the Large Groups Project under grant number (45/43). Princess Nourah bint Abdulrahman University Researchers Supporting Project number (PNURSP2022R238), 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 (Grant Code: 22UQU4340237DSR29).

Institutional Review Board Statement: Not Applicable.

Informed Consent Statement: Not Applicable.

Data Availability Statement: Data sharing is not applicable to this article, as no datasets were generated during the current study.

Conflicts of Interest: The authors declare that they have no conflict of interest. The manuscript was written through contributions of all authors. All authors have given approval to the final version of the manuscript.

References

1. Sarsam, S.M.; Al-Samarraie, H.; Alzahrani, A.I.; Wright, B. Sarcasm detection using machine learning algorithms in Twitter: A systematic review. *Int. J. Mark. Res.* **2020**, *62*, 578–598. [CrossRef]
2. Kumar, A.; Narapareddy, V.T.; Srikanth, V.A.; Malapati, A.; Neti, L.B.M. Sarcasm Detection Using Multi-Head Attention Based Bidirectional LSTM. *IEEE Access* **2020**, *8*, 6388–6397. [CrossRef]
3. Muaad, A.Y.; Davanagere, H.J.; Benifa, J.V.B.; Alabrah, A.; Saif, M.A.N.; Pushpa, D.; Al-Antari, M.A.; Alfakih, T.M. Artificial Intelligence-Based Approach for Misogyny and Sarcasm Detection from Arabic Texts. *Comput. Intell. Neurosci.* **2022**, *2022*, 7937667. [CrossRef] [PubMed]
4. Banerjee, A.; Bhattacharjee, M.; Ghosh, K.; Chatterjee, S. Synthetic minority oversampling in addressing imbalanced sarcasm detection in social media. *Multimed. Tools Appl.* **2020**, *79*, 35995–36031. [CrossRef]
5. Jaiswal, N. Neural sarcasm detection using conversation context. In Proceedings of the Second Workshop on Figurative Language Processing, Seattle, WA, USA, 9 July 2020; pp. 77–82.
6. Dong, X.; Li, C.; Choi, J.D. Transformer-based context-aware sarcasm detection in conversation threads from social media. *arXiv* **2020**, arXiv:2005.11424.
7. Shrivastava, M.; Kumar, S. A pragmatic and intelligent model for sarcasm detection in social media text. *Technol. Soc.* **2020**, *64*, 101489. [CrossRef]
8. Lou, C.; Liang, B.; Gui, L.; He, Y.; Dang, Y.; Xu, R. Affective dependency graph for sarcasm detection. In Proceedings of the 44th International ACM SIGIR Conference on Research and Development in Information Retrieval, New York, NY, USA, 11–15 July 2021; pp. 1844–1849.
9. Gupta, R.; Kumar, J.; Agrawal, H. A statistical approach for sarcasm detection using Twitter data. In Proceedings of the 4th International Conference on Intelligent Computing and Control Systems (ICICCS), Madurai, India, 13–15 May 2020; pp. 633–638.

10. Yao, F.; Sun, X.; Yu, H.; Zhang, W.; Liang, W.; Fu, K. Mimicking the Brain's Cognition of Sarcasm from Multidisciplines for Twitter Sarcasm Detection. *IEEE Trans. Neural Netw. Learn. Syst.* **2021**, 1–15. [CrossRef] [PubMed]
11. Potamias, R.A.; Siolas, G.; Stafylopatis, A.G. A transformer-based approach to irony and sarcasm detection. *Neural Comput. Appl.* **2020**, *32*, 17309–17320. [CrossRef]
12. Pan, H.; Lin, Z.; Fu, P.; Qi, Y.; Wang, W. Modeling Intra and Inter-modality Incongruity for Multi-Modal Sarcasm Detection. In *Findings of the Association for Computational Linguistics*; Association for Computational Linguistics: Stroudsburg, PA, USA, 2020; pp. 1383–1392.
13. Cai, Y.; Cai, H.; Wan, X. Multi-modal sarcasm detection in twitter with hierarchical fusion model. In Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics, Florence, Italy, 28 July–2 August 2019; pp. 2506–2515.
14. Akula, R.; Garibay, I. Interpretable Multi-Head Self-Attention Architecture for Sarcasm Detection in Social Media. *Entropy* **2021**, *23*, 394. [CrossRef] [PubMed]
15. Du, Y.; Li, T.; Pathan, M.S.; Teklehaimanot, H.K.; Yang, Z. An Effective Sarcasm Detection Approach Based on Sentimental Context and Individual Expression Habits. *Cogn. Comput.* **2021**, *14*, 78–90. [CrossRef]
16. Kamal, A.; Abulaish, M. Cat-bigru: Convolution and attention with bi-directional gated recurrent unit for self-deprecating sarcasm detection. *Cogn. Comput.* **2022**, *14*, 91–109. [CrossRef]
17. Sultana, A.; Bardalai, A.; Sarma, K.K. Salp Swarm-Artificial Neural Network Based Cyber-Attack Detection in Smart Grid. *Neural Process. Lett.* **2022**, 1–23. [CrossRef]
18. Soleymanzadeh, R.; Aljasim, M.; Qadeer, M.W.; Kashef, R. Cyberattack and Fraud Detection Using Ensemble Stacking. *AI* **2022**, *3*, 22-36. [CrossRef]
19. Sagheer, A.; Kotb, M. Unsupervised Pre-training of a Deep LSTM-based Stacked Autoencoder for Multivariate Time Series Forecasting Problems. *Sci. Rep.* **2019**, *9*, 19038. [CrossRef] [PubMed]
20. Yu, M.; Quan, T.; Peng, Q.; Yu, X.; Liu, L. A model-based collaborate filtering algorithm based on stacked AutoEncoder. *Neural Comput. Appl.* **2021**, *34*, 2503–2511. [CrossRef]
21. Mellal, M.A.; Al-Dahidi, S.; Williams, E.J. System reliability optimization with heterogeneous components using hosted cuckoo optimization algorithm. *Reliab. Eng. Syst. Saf.* **2020**, *203*, 107110. [CrossRef]
22. Available online: https://www.kaggle.com/c/detecting-insults-in-social-commentary/data (accessed on 12 March 2022).
23. Albraikan, A.A.; Hassine, S.B.H.; Fati, S.M.; Al-Wesabi, F.N.; Hilal, A.M.; Motwakel, A.; Hamza, M.A.; Al Duhayyim, M. Optimal Deep Learning-based Cyberattack Detection and Classification Technique on Social Networks. *Comput. Mater. Contin.* **2022**, *72*, 907–923. [CrossRef]