©2024 International Transaction Journal of Engineering, Management, & Applied Sciences & Technologies

ISSN 2228-9860 eISSN 1906-9642 CODEN: ITJEA8

International Transaction Journal of Engineering, Management, & Applied Sciences & Technologies

http://TuEngr.com

An Automated Hate Speech Detection Using Deep Learning on Arabic Corpus

Abdulrahman Alzahrani[1]*

[1]*Department of Computer Science and Engineering, College of Computer Science and Engineering, University of Hafr Al Batin, SAUDI ARABIA.*

**Corresponding Author (Tel: +966500558900, Email: aalzahrani @uhb.edu.sa).*

Paper ID: 15A4F

Volume 15 Issue 4

Received 15 July 2024
Received in revised form 26 August 2024
Accepted 02 September 2024
Available online 09 September 2024

Keywords:
Arabic Hate Speech; Offensive Language; Deep Learning; BiGRU; Arithmetic Optimization Algorithm; AraBERT; AHOSD-DLAC; Machine Learning; AOA; Natural Language Processing; Speech Detection; NLP; MOA function.

Abstract

Hate speech on social media, particularly in the Arabic language take witnessed a dramatic increase and caused significant destruction. Several researches have been undertaken on social media adopted by people to share their thoughts. Due to the diverse and broad linguistic terrain, Hate Speech Detection in Arabic brings about substantial hurdles. With its rich cultural subtleties and multiple dialects, Arabic requires certain measures to successfully address hate speech online. To overcome these issues, developers have applied ML algorithms and NLP systems adapted to the difficulties of Arabic text. This work offers the design of Automated Hate and Offensive Speech Detection using Deep Learning on Arabic Corpus (AHOSD-DLAC) technique. The objective of AHOSD-DLAC is to design an approach that is capable of handling the recognition and classification of offensive language and Arabic hate speech. The AHOSD-DLAC is composed of pre-processing, AraBERT-based feature extraction, and bidirectional gated recurrent unit (BiGRU) classification to identify hate speech. An arithmetic optimizer algorithm (AOA) is used for hyperparameter tuning to improve the performance of the AHOSD-DLAC. We accompanied a set of experiments on X data in order to evaluate the AHOSD-DLAC. The AHOSD-DLAC shows potential outcomes on hate speech recognition than preliminary works on Arabic hate-speech recognition.

Discipline: Computer Science and Engineering..

©2024 INT TRANS J ENG MANAG SCI TECH.

Cite This Article:

Alzahrani, A. (2024). An Automated Hate Speech Detection using Deep Learning on Arabic Corpus.. *International Transaction Journal of Engineering, Management, & Applied Sciences & Technologies, 15*(4), 15A4F: 1-14. http://TUENGR.COM/V15/15A4F.pdf DOI: 10.14456/ITJEMAST.2024.25

1 Introduction

In the present scenario, social media platforms like Twitter, Facebook, and WhatsApp are commonly employed for making and exchanging content [1]. They deliver consumers with an easy

and convenient method to share the data effectively and quickly, creating them a valued resource of data. Social media platforms are the main technique for distributing harmful and offensive content [2]. The spread of bad and toxic content on social media sites has a main harmful effect on the capability of consumers and the complete excellence of online shared content [3]. In the past few years, the systematic recognition of hate speech and other violent speech has affected the notice from several researchers owing to the rising propagation of offensive and harmful content on social network platforms [4]. These sites have hate speech strategies, which exclude any publishes that threaten or attack people or groups depending on their defended features like ethnicity, race, gender, religion, and nationality. However, filtering hate speech depends upon reporting bad behavior, and observing by agents [5]. Physically monitoring the massive number of content on social media platforms is a difficult job. So, developing technologies will mechanically identify hate speech, which has become very dangerous [6]. In the domains of computer science and ML, many researchers have established techniques, which can repeatedly recognize hate speech on the platforms of social media. These methods can aid in easing the transmission of this kind of hate content [7]. On the other hand, most of the models mainly concentrated on the recognition of harmful speech in English; there is an absence of analyses on the recognition of Arabic harmful speech owing to the nature of the language's complexity.

Past research on the content of Arabic social media has been greatly determined by both identifying offensive language and the recognition of disliked speech, which can be recognized [8]. The Arabic language is one of the foremost languages with the greatest social hostilities index that estimates offenses inspired in segments by race or religion. It aims at the significance of harmful speech in Arabic, as this kind of content contains dangerous negative outcomes in groups. Social network strategies generally exclude consumers from broadcasting any wrong or illegal content [9]. Owing to this, most of the users post data that looks real but slightly increases the boundaries of harmful speech. As an outcome, it is complex to acquire devices that can discover dislike speech mechanically [10]. There are numerous ML methods to classify harmful speech on platforms of social media and in other online groups.

This work offers the design of Automated Hate and Offensive Speech Detection using the DL on Arabic Corpus (AHOSD-DLAC) technique. The objective of the AHOSD-DLAC technique is to design an approach that is capable of handling the recognition and classification of offensive language and Arabic hate speech. The AHOSD-DLAC technique is composed of pre-processing, AraBERT-based feature extraction, and bidirectional gated recurrent unit (BiGRU) classification to identify hate speech. An arithmetic optimizer algorithm (AOA) can be used for hyperparameter tuning to improve the performance of the AHOSD-DLAC technique. We accompanied a set of experiments on Twitter data in order to evaluate the AHOSD-DLAC technique.

2 Related Works

Saleh et al. [11] investigated the probability of leveraging domain-specific word embedding as a feature and a BiLSTM-based deep method as a classification method in order to certainly

identify harmful speech. This technique ensures that the term is allocated its negative value, which is a useful method for detecting coded words. Likewise, the authors examine the usage of the TL language method (BERT) on the harmful speech issue as a dual task of identification as it offers higher-performance outcomes for numerous NLP tasks. Motwakel et al. [12] presented an Improved ALO with DL Dirven Offensive and Hate Speech Detection (IALODL-OHSD) approach. The projected IALODL-OHSD approach generally objective is to discover and categorize harmful speech communicated on platforms of social media. It contains a 3-phase procedure implemented, such as word embedding, pre-processing, and identification. Mainly, data pre-processing is achieved to modify the text of Arabic social media into a beneficial layout. The attention-based cascaded LSTM (ACLSTM) technique is employed for the process of classification. At last, the IALO technique is utilized as a hyperparameter optimizer for boosting the outcomes of the classifier. Alsafari et al. [13] main objective is to improve an efficient Arabic offensive and hate speech recognition architecture in order to find out this vital problem. Initially, we construct a trustworthy Arabic textual amount by scrambling data from Twitter utilizing 4 strong extraction plans that we execute depending on 4 kinds of hate such as ethnicity, nationality, religion, and gender. Then, the authors identify the corpus depending upon a 3-hierarchical annotation plan in which we validate the inter-annotation compact to confirm filed truth at every level.

Aldjanabi et al. [14] examine hate and offensive speech on media of Arab social to construct a perfect aggressive and hate speech recognition method. More exactly, the author projected an identification method for defining harmful speech utilizing a multi-task learning (MTL) method constructed on the leading of a pre-trained Arabic language method. In [15], only the mechanical features of designing an automatic method could observe and perceive harmful speech in the Arabic language, which several firms, like Facebook, Twitter, and many more, could utilize to stop harmful speech and cyberbullying. Furthermore, the usage of deep RNN has been developed for the identification and recognition of hate speech. Another technique comprising 5 hidden layers, termed DRNN 1 that utilized only for dual identification.

Alatawi et al. [16] investigate the potential for automatically identifying white supremacist harmful speech on Twitter utilizing DL and NLP approaches. Dual DL models are researched in this study. The initial technique uses a BiLSTM technique beside domain-specific word embedding isolated from white supremacist corpus to acquire the semantics. The next method uses the most current language technique called BERT. Boulouard et al. [17] projected a TL resolution to identify hateful and violent speech on platforms of social media. This research work will evaluate the performance of distinctive BERT-based techniques trained to categorize commentaries as both offensive and unbiased.

3 Method

3.1 The Proposed Model

This work offers the design of the AHOSD-DLAC technique. The objective of the AHOSD-DLAC technique is to design an approach that is capable of handling the recognition and

classification of offensive language and Arabic hate speech. Figure 1 depicts the workflow of the AHOSD-DLAC technique.

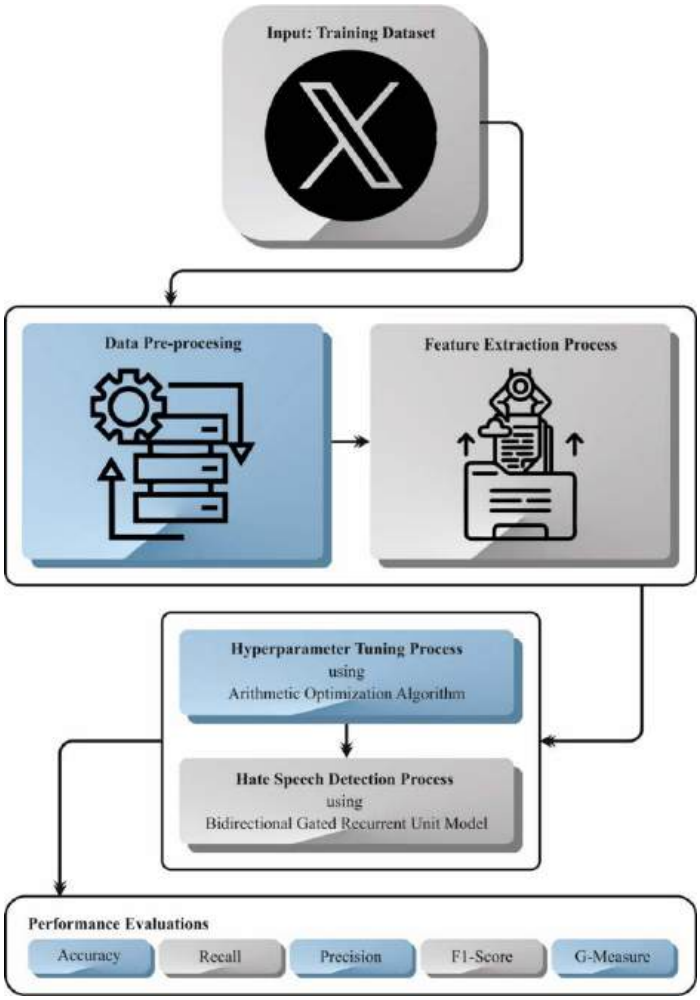

Figure 1: Workflow of AHOSD-DLAC technique

3.2 Preprocessing

Data preprocessing and cleaning steps have been conducted to remove noise in the dataset [18]. Firstly, delete each duplicate and "retweeted" tweets. Then, remove the non-Arabic tweet from the data. Later, remove redundant tokens like URLs, user tags, numbers, emails, hashtags, and HTML tags because they could decrease the classier performance.

Even though emoji express emotions, they are eliminated from the data for keeping the emojis from the dialect Arabic Twitter decreases the classier performance, and because of the way Arabic word is written from right to left, resulting in the emoticons reversal, and misunderstanding between brackets in the emoticons and quote. Afterward, remove whitespaces including newlines, tabs, and duplicate spaces from the data.

Lastly, remove the shorter tweets with very small sentences from the dataset. It is noteworthy that the stemming algorithm is not used for the data as it is required to perform better with Arabic dialect words. Now, the dataset has 1,824,220 tweets after using the pre-processing and cleaning steps.

3.3 AraBERT-based Feature Extraction

AraBERT is a pre-trained language method specially tailored for Arabic NLP tasks [19]. It is based on the BERT model that has been fine-tuned on the considerable amount of Arabic text for capturing linguistic nuances of the language.

The training data of AraBERT involves various sources like social media posts, web pages, and news articles, which ensures a thorough knowledge of Arabic usage. This model is especially effective for tasks like sentiment analysis, text classification, machine translation, and entity detection in Arabic, which provide powerful mechanisms for developers and researchers working in the Arabic NLP space. By leveraging the strengths of BERT and adapting them to Arabic, AraBERT has considerably increased cutting-edge Arabic language processing

3.4 BiGRU-based Classification Process

To identify hate speech, the BiGRU classification can be employed. GRU is similar to the LSTM model, which was projected to overcome the long-term memory and gradient issues in classical RNN during BP [20]. This unit is intended for sequential data processing, especially in capturing long-term dependency. GRU is used to control the data flow by presenting a gating module. For a series = $(x_1, x_2, x_3, \ldots, x_t)$, where x_t refers to the input at t moment, the update gate z_t defines how far the hidden layer (HL) is from the prior moment h_{t-1} is preserved in the existing moment:

$$z_t = \sigma(W_z \cdot [h_{t-1'} x_t] + b_z) \quad (1).$$

The reset gate r_t is used to control the influence of a prior moment h_{t-1} on computing the candidate HL $\tilde{h}_{t-1}$ at the existing moment:

$$r_t = \sigma(W_r \cdot [h_{t-1}, x_t] + b_r) \quad (2).$$

Based on the present input and reset prior HL, the candidate HL h_t is computed, which provides candidate values for the novel HL:

$$\tilde{h}_t = tanh(W \cdot [r_t \odot h_{t-1}, x_t] + b) \quad (3).$$

The last h_t HL is defined while interacting with the update gate, representing the amount of the prior HL to be preserved:

$$h_t = (1 - z_t) \odot h_{t-1} + z_t \odot \tilde{h}_t \quad (4).$$

In Eq. (4), σ and $tanh$ are the sigmoid and hyperbolic tangent activation functions; $\odot$ indicates the Hadamard product; $\mathcal{W}_z, \mathcal{W}_r$, and $\mathcal{W}$ are the weighted matrices; and b_z, b_r, and b are the bias vectors.

BiGRU is used to deploy two independent GRUs at the moment, viz., processing the forward and the backward data flow, which enables it to simultaneously encode backward and forward information of a series. This enables us to recognize the data from 2 directions, offering a complete overview of this sequence. The HLs of forward and backward directions are evaluated as follows:

$$h_t^{BiGRu} = \left[h_t^{forward}, h_t^{backward}\right] \quad (5).$$

In Eq. (5), $h_t^{forward}$ and $h_t^{backward}$ are the forward and backward HL at the t moment. In contrast to the LSTM network structure, GRU only has update and reset gates, whereas LSTM contains an output gate, a forget gate, and an input gate. It implies that GRU has a more streamlined network structure and fewer model parameters under the premise of a similar amount of hidden units. In addition, the model uses long-term dependency and can remember data in the input sequence. Figure 2 illustrates the structure of BiGRU.

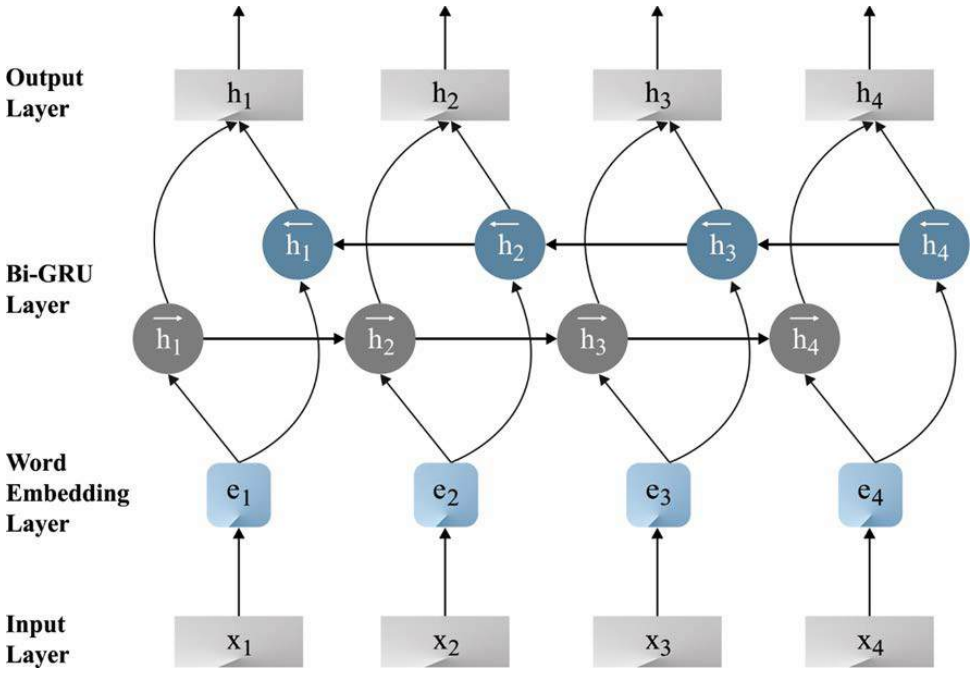

Figure 2: Architecture of BiGRU

3.5 AOA-based Parameter Tuning

Finally, an AOA can be utilized for hyperparameter tuning to improve the performance of the AHOSD-DLAC methodology. AOA is a mathematical model that efficiently implements optimization procedures through different search fields [21]. This technique aims at determining the ideal values for decision parameters within the given problem or function either by minimizing or maximizing the objective function. AOA must determine the appropriate search phase before beginning the optimization procedure wherein the Math Optimized Accelerated (MOA) function acts as a coefficient. This significantly improves the effectiveness of search phases within AOA.

A. Initialization Stage

In AOA, the optimizer process begins by arbitrarily creating a possible candidate solution. At the iteration, the best solution is detected as the closest to the optimum solution or most optimal attained. The search process is defined by the MOA function that depends on a condition that compares a random number to the MOA function. The MOA function increases linearly from 0.2 to 0.9 within the search range.

$$MOA(C_{Iter}) = \text{Min} + C_{Iter} \times \left(\frac{\text{Max}-\text{Min}}{M_{Iter}}\right) \quad (6).$$

Moreover, the Math Optimizer Probability (MOP) acts as a coefficient, and function value at t^{th}iteration is represented as $MOP\ (C\_Iter)$. The α parameter defines the exploitation accuracy at the iteration and remains constant at a 5 value based on the outcomes within AOA. $r1, r2$, and r3 are random numbers.

$$MOP(C_{Iter}) = 1 - \frac{C_{Iter}^{1/a}}{M_{Iter}^{1/a}} \quad (7).$$

B. Exploration Stage

The exploration operator utilized in AOA plays an important part in randomly and thoroughly exploring various regions in the search range. This operator aims to determine a better solution by performing two search processes namely Division (D) and the Multiplication (M) operators. The usage of the MOA function conditions the exploration phase, particularly when the random value ($r1$) surpasses the threshold value.

$$x_{i,j} = \begin{Bmatrix} \frac{best\_x_j}{(M0P+\in)} \times \begin{pmatrix} (UB_j - LB_j) \\ \times \mu + LB_j \end{pmatrix}, r2 > 0.5 \\ best\_x_j \times M0P \times \begin{pmatrix} (UB_j - LB_j) \\ \times \mu + LB_j \end{pmatrix}, else \end{Bmatrix} \quad (8).$$

In the exploration stage, the D operator is activated selectively if another random value (r2) is lesser than 0.5. The initial operator thoroughly carries out its task until the end, whereupon the M operator remains idle. On the other hand, if the activation condition is not satisfied, the M operator controls the ongoing task rather than the D operator. Here, $\in$ is a parameter with a small integer, whereas $UB_j$and LB_j are the upper and lower boundaries correspondingly related to the j^{th} location. Additionally, the μ parameter controls the search process with a 0.5 value.

C. Exploitation Stage

This section shows the robust exploitation stage applied by the AOA. Subtraction (S) and Addition (A) operators in AOA accomplish remarkable results by harnessing the arithmetic abilities, considered as their density and precision. The search process within AOA is contingent and orchestrated on the presence of the MOA function. The MOA function directs the exploration stage, ensuring effective and efficient exploitation of the search range. During this vital stage, the initial operator (S) is activated selectively based on the condition that a random number (r3) falls below the threshold of 0.5. This operator diligently performs its task assigned, focusing fully on extracting meaningful data. Meanwhile, another operator (A) patiently awaits its turn, abstaining from engagement until the initial operator finishes its existing objective. If the condition is not satisfied, the operator (A) readily steps in, taking over the ongoing task from the initial operator (S).

$$x_{i,j} = \begin{Bmatrix} \begin{pmatrix} best\_x_j \\ -MOP \end{pmatrix} \times \begin{pmatrix} (UB_j - LB_j) \\ \times \mu + LB_j \end{pmatrix}, r3 > 0.5 \\ best\_x_j + M0P \times \begin{pmatrix} (UB_j - LB_j) \\ \times \mu + LB_j \end{pmatrix}, else \end{Bmatrix} \quad (9).$$

The fitness choice is a main feature adjusting the effectiveness of AOA. The hyperparameter choice method comprises the encoder model to measure the result of candidate efficacies. In this effort, the AOA assumes that accuracy is a main condition to the proposal of the FF that is written as

$$Fitness = \max(P) \quad (10),$$

$$P = \frac{TP}{TP+FP} \quad (11).$$

In which, TP and FP stand for the true and false positive rates.

4 Result Analysis

This section discusses the experimental validation analysis of the AHOSD-DLAC approach using a speech dataset [22]. The dataset contains 900 samples with three classes are represented in Table 1. The suggested technique is simulated by employing the Python 3.6.5 tool on PC i5-8600k, 250GB SSD, GeForce 1050Ti 4GB, 16GB RAM, and 1TB HDD. The parameter settings are provided in the following: learning rate: 0.01, activation: ReLU, epoch count: 50, dropout: 0.5, and batch size: 5.

Table 1: Details of the dataset.

Class	No. of Samples
Normal	300
Abusive	300
Hate	300
Total Samples	900

Figure 3 defines the classifier performances of the AHOSD-DLAC algorithm at the test database. Figures. 3a-3b displays the confusion matrices produced by the AHOSD-DLAC system at 70%TRAS and 30%TESS. The outcome stated that the AHOSD-DLAC system has recognized 3 classes properly. Next, Figure 3c reveals the PR analysis of the AHOSD-DLAC methodology. The outcome implied that the AHOSD-DLAC technique has accomplished a better solution of PR in 3 classes. However, Figure 3d portrays the ROC result of the AHOSD-DLAC technique. The outcome depicted that the AHOSD-DLAC algorithm has resulted in capable performances with superior values of ROC at 3 classes.

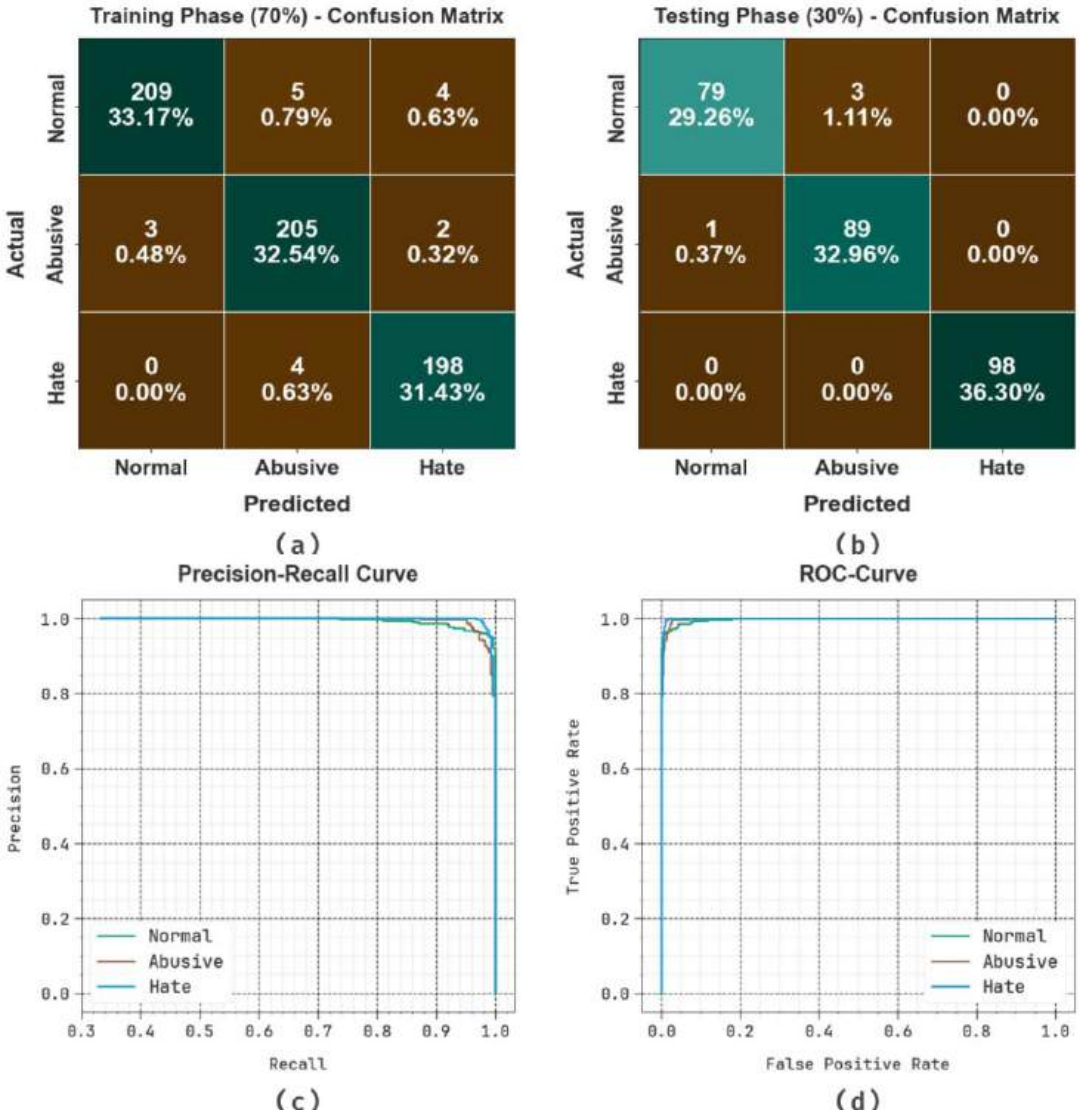

Figure 3: Classifier outcome of (a-b) Confusion matrices and (c-d) PR and ROC curves

The detection outcomes of the AHOSD-DLAC methodology under several speech classes are depicted in Table 2. In Figure 4, the overall performances of the AHOSD-DLAC algorithm at several classes on 70%TRAS are given. The outcomes stated that the AHOSD-DLAC methodology reaches enhanced performance. With normal class, the AHOSD-DLAC method gains $accu_y$ of 98.10%, $prec_n$ of 98.58%, $reca_l$ of 95.87%, $F1_{score}$ of 97.21%, and $G_{measure}$ of 97.22%.

Table 2: Detection outcome of AHOSD-DLAC system at 70%TRAS and 30%TESS.

Classes	$Accu_y$	$Prec_n$	$Reca_l$	$F1_{Score}$	$G_{Measure}$
TRAS (70%)					
Normal	98.10	98.58	95.87	97.21	97.22
Abusive	97.78	95.79	97.62	96.70	96.70
Hate	98.41	97.06	98.02	97.54	97.54
Average	98.10	97.15	97.17	97.15	97.15
TESS (30%)					
Normal	98.52	98.75	96.34	97.53	97.54
Abusive	98.52	96.74	98.89	97.80	97.81
Hate	100.00	100.00	100.00	100.00	100.00
Average	99.01	98.50	98.41	98.44	98.45

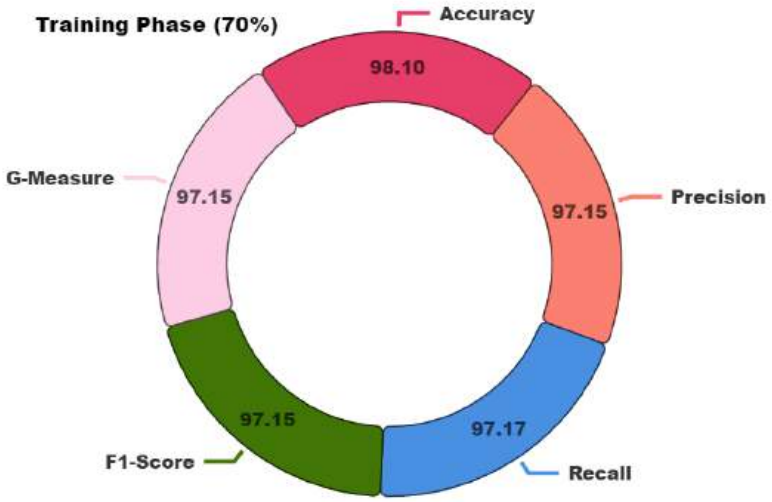

Figure 4: Average of AHOSD-DLAC technique under 70%TRAS

Followed by, abusive class, the AHOSD-DLAC system reaches $accu_y$ of 97.78%, $prec_n$ of 95.79%, $reca_l$ of 97.62%, $F1_{score}$ of 96.70%, and $G_{measure}$ of 96.70%. Moreover, with hate class, the AHOSD-DLAC system offers $accu_y$ of 98.41%, $prec_n$ of 97.06%, $reca_l$ of 98.02%, $F1_{score}$ of 97.54%, and $G_{measure}$ of 97.54%.

In Figure 5, the overall outcomes of the AHOSD-DLAC methodology at many classes on 30%TESS are given. The outcomes inferred that the AHOSD-DLAC algorithm reaches higher performance. With normal class, the AHOSD-DLAC technique reaches $accu_y$ of 98.52%, $prec_n$ of 98.75%, $reca_l$ of 96.34%, $F1_{score}$ of 97.53%, and $G_{measure}$ of 97.54%. Afterward, with abusive class, the AHOSD-DLAC technique achieves $accu_y$ of 98.52%, $prec_n$ of 96.74%, $reca_l$ of 98.89%, $F1_{score}$ of 97.80%, and $G_{measure}$ of 97.81%. Finally, with hate class, the AHOSD-DLAC methodology accomplishes $accu_y$ of 100.00%, $prec_n$ of 100.00%, $reca_l$ of 100.00%, $F1_{score}$ of 100.00%, and $G_{measure}$ of 100.00%.

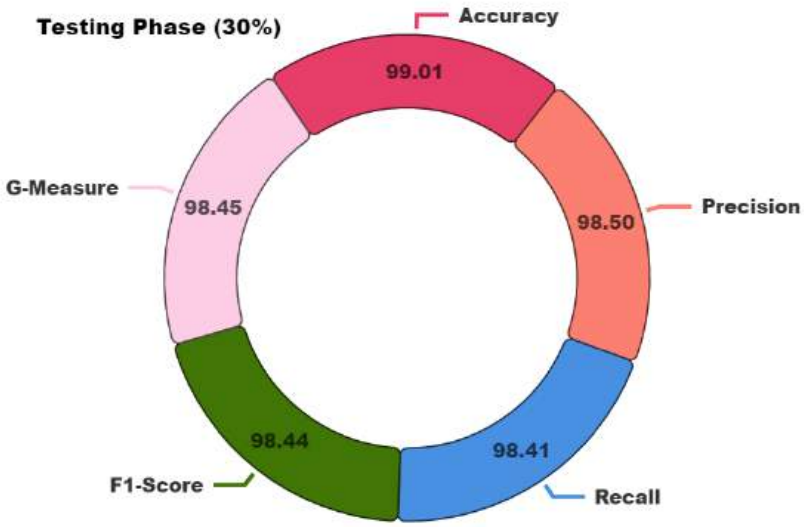

Figure 5. Average of AHOSD-DLAC technique under 30%TESS

In Figure 6, the training and validation accuracy outcomes of the AHOSD-DLAC approach are shown. The accuracy outcomes are estimated over an interval of 0-25 epochs. The figure emphasized that the training and validation accuracy values reveal an increasing tendency that notified the ability of the AHOSD-DLAC system with improved performance under many iterations. Furthermore, the training accuracy and validation accuracy remain closer over the epochs, which indicates low lesser overfitting and exhibits enhanced performance of the AHOSD-DLAC algorithm, ensuring reliable prediction on unnoticed samples.

Figure 6: $Accu_y$ the curve of the AHOSD-DLAC methodology.

In Figure 7, the training and validation loss graph of the AHOSD-DLAC method is depicted. The loss rates are calculated on an interval of 0-25 epochs. It is defined that the training and validation accuracy outcome depict a reducing tendency, reporting the ability that notified the capability of the AHOSD-DLAC methodology in balancing a trade-off among data fitting and generalized. The constant reduction in loss outcomes further guarantees the superior performance of the AHOSD-DLAC methodology and adjusts the prediction outcomes over time.

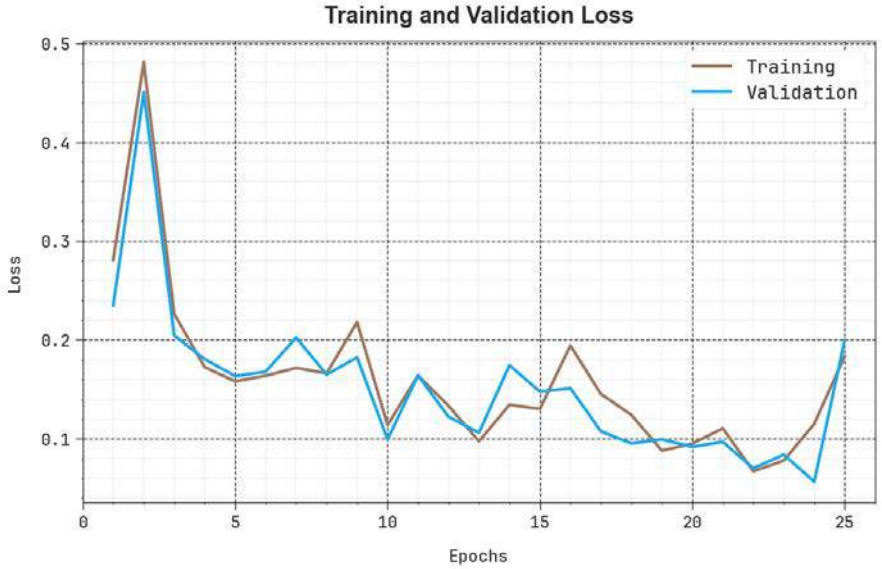

Figure 7: Loss curve of the AHOSD-DLAC technique.

In Table 3 and Figure 8, the comparative outcomes of the AHOSD-DLAC system with other systems are demonstrated [23]. The outcomes are proficient in that the CNN+LTSM and KNN systems have resulted in minimal performances. Similarly, the Bernoulli NB, Multinomial NB, and LSTM methodologies are somewhat improved results. Although the SVC, liner SVC, and ABMM algorithms have gained considerable outcomes, the AHOSD-DLAC methodology reaches superior $accu_y$ of 99.01%, $prec_n$ of 98.50%, and $reca_l$ of 98.41%. Therefore, the AHOSD-DLAC methodology has been executed for speech recognition.

Table 3: Detection outcome of AHOSD-DLAC system at 70%TRAS and 30%TESS.

Classifier	$Accu_y$	$Prec_n$	$Reca_l$
AHOSD-DLAC	99.01	98.50	98.41
ABMM Algorithm	98.60	98.30	98.30
LSTM Model	95.10	95.10	95.10
Linear SVC	96.00	96.00	96.00
SVC Model	95.30	95.30	95.30
Multinomial NB	95.10	95.10	95.10
Bernoulli NB	93.60	93.70	93.60
K-Nearest Neighbors	93.40	93.40	93.40
CNN+LTSM	75.00	72.00	75.00

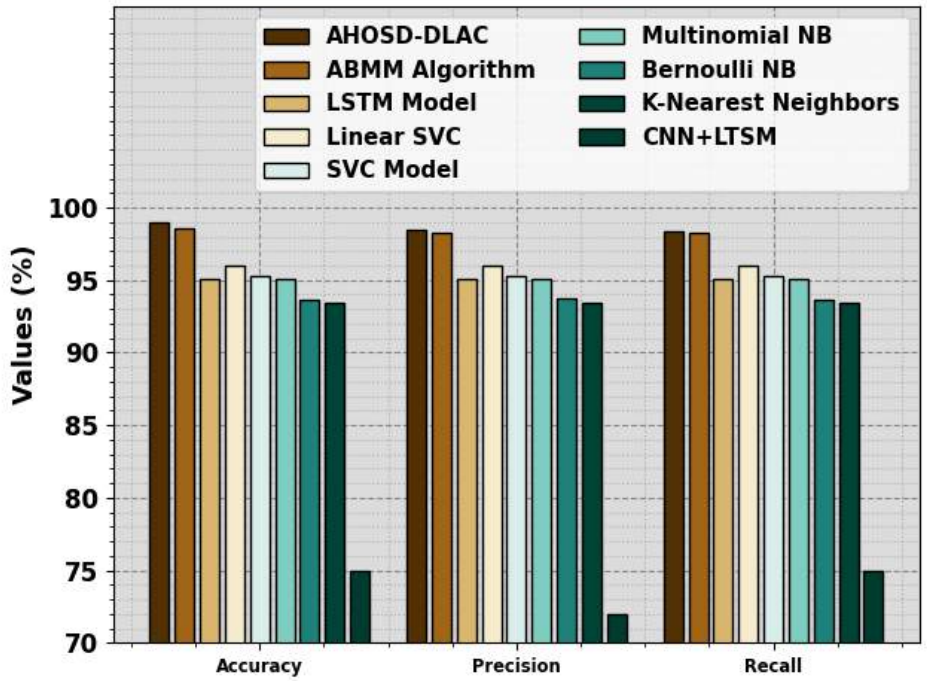

Figure 8: Comparative outcome of AHOSD-DLAC algorithm with recent systems.

5 Conclusion

In this work, we offer the design of AHOSD-DLAC technique. The objective of the AHOSD-DLAC technique is to design an approach that is capable of handling the recognition and classification of offensive language and Arabic hate speech. The AHOSD-DLAC technique comprises pre-processing, AraBERT-based feature extractor, and BiGRU classification to recognize

hate speech. An AOA can be employed for hyperparameter tuning to improve the performance of the AHOSD-DLAC methodology. We accompanied a set of experiments on Twitter data in order to evaluate the AHOSD-DLAC technique. The AHOSD-DLAC method shows potential outcomes on hate speech recognition than preliminary works on Arabic hate-speech recognition. Limitations of the AHOSD-DLAC method comprise the need for comprehensive evaluation across various datasets to validate robustness, and future work could explore enhancing the AOA-based hyperparameter tuning process for enhanced model performance and efficiency in intrusion detection applications.

6 Availability of Data and Material

Data can be made available by contacting the corresponding author.

7 References

[1] Awane, W., Lahmar, E.H.B. and Falaki, A.E., 2021. Hate Speech in the Arab Electronic Press and Social Networks. Revue d'Intelligence Artificielle, 35(6).

[2] Davidson, T., Warmsley, D., Macy, M., Weber, I. 2017. Automated Hate Speech Detection and the Problem of Offensive Language. In Proceedings of the International AAAI Conference on Web and Social Media, Montreal, QC, Canada, 15–18 July 2017, pp. 88–93.

[3] Fortuna, P., Nunes, S. A Survey on Automatic Detection of Hate Speech in Text. ACM Comput. Surv. 2018, 51, 1–30.

[4] Al-Hassan, A., Al-Dossari, H. Detection of hate speech in Arabic tweets using deep learning. In Multimedia Systems, Springer Nature: Cham, Switzerland, 2021.

[5] Gitari, N.D., Zhang, Z., Damien, H., Long, J. A Lexicon-based Approach for Hate Speech Detection. Int. J. Multimed. Ubiquitous Eng. 2015, 10, 215–230.

[6] Silva, L., Mondal, M., Correa, D., Benevenuto, F., Weber, I. Analyzing the Targets of Hate in Online Social Media. Proc. Int. AAAI Conf. Web Soc. Media 2021, 10, 687–690.

[7] Kwok, I., Wang, Y. Locate the Hate: Detecting Tweets against Blacks. Proc. AAAI Conf. Artif. Intell. 2013, 27, 1621–1622.

[8] Mercan, V., Jamil, A., Hameed, A.A., Magsi, I.A., Bazai, S. and Shah, S.A., 2021, October. Hate speech and offensive language detection from social media. In 2021 International Conference on Computing, Electronic and Electrical Engineering (ICE Cube) (pp. 1-5). IEEE.

[9] Alshalan, R., Al-Khalifa, H. A Deep Learning Approach for Automatic Hate Speech Detection in the Saudi Twittersphere. Appl. Sci. 2020, 10, 8614.

[10] Burnap, P., Williams, M.L. Cyber Hate Speech on Twitter: An Application of Machine Classification and Statistical Modeling for Policy and Decision Making. Policy Internet 2015, 7, 223–242.

[11] Saleh, H., Alhothali, A. and Moria, K., 2023. Detection of hate speech using bert and hate speech word embedding with deep model. Applied Artificial Intelligence, 37(1), p.2166719.

[12] Motwakel, A., Al-onazi, B.B., Alzahrani, J.S., Alazwari, S., Othman, M., Zamani, A.S., Yaseen, I. and Abdelmageed, A.A., 2023. Improved Ant Lion Optimizer with Deep Learning Driven Arabic Hate Speech Detection. Computer Systems Science & Engineering, 46(3).

[13] Alsafari, S., Sadaoui, S. and Mouhoub, M., 2020. Hate and offensive speech detection on Arabic social media. Online Social Networks and Media, 19, p.100096.

[14] Aldjanabi, W., Dahou, A., Al-qaness, M.A., Elaziz, M.A., Helmi, A.M. and Damaševičius, R., 2021. Arabic offensive and hate speech detection using a cross-corpora multi-task learning model. In Informatics (Vol. 8, No. 4, p. 69). MDPI.

[15] Anezi, F.Y.A., 2022. Arabic hate speech detection using deep recurrent neural networks. Applied Sciences, 12(12), p.6010.

[16] Alatawi, H.S., Alhothali, A.M. and Moria, K.M., 2021. Detecting white supremacist hate speech using domain specific word embedding with deep learning and BERT. IEEE Access, 9, pp.106363-106374.

[17] Boulouard, Z., Ouaissa, M., Ouaissa, M., Krichen, M., Almutiq, M. and Gasmi, K., 2022. Detecting hateful and offensive speech in arabic social media using transfer learning. Applied Sciences, 12(24), p.12823.

[18] Ahmad, A., Azzeh, M., Alnagi, E., Abu Al-Haija, Q., Halabi, D., Aref, A. and AbuHour, Y., 2024. Hate speech detection in the Arabic language: corpus design, construction, and evaluation. Frontiers in Artificial Intelligence, 7, p.1345445.

[19] Abo-Elghit, A.H., Hamza, T. and Al-Zoghby, A., 2022. Embedding Extraction for Arabic Text Using the AraBERT Model. Computers, Materials & Continua, 72(1).

[20] Guo, L. and Wang, Y., 2024. Predicting Tool Wear with ParaCRN-AMResNet: A Hybrid Deep Learning Approach. Machines, 12(5), p.341.

[21] Asghar, A. and Ashraf, M.M., Estimation of Power System Harmonics Using Least Square Based Arithmetic Optimization Algorithm.

[22] https://www.kaggle.com/datasets/haithemhermessi/arabic-levantine-hate-speech-detection

[23] Almaliki, M., Almars, A.M., Gad, I. and Atlam, E.S., 2023. Abmm: Arabic bert-mini model for hate-speech detection on social media. Electronics, 12(4), p.1048.

Dr.Abdulrahman Alzahrani is an Assistant Professor at Department of Computer Science and Engineering in the College of Computer Science and Engineering at University of Hafr Al Batin, Saudi Arabia. He got his Master's and PhD degrees in Computer Science from University of Bridgeport, Bridgeport, CT, USA and Oakland University, Rochester, MI, USA respectively. His research focuses on Artificial Intelligence Applications, IoT, and Cybersecurity.