[1] **Ajay Rastogi**

[2] **Ravendra Singh**

[3] **Mohammad Zubair Khan**

Radicalization Detection Using Hybrid Deep Learning with Whale Optimization Technique

Journal of Electrical Systems

***Abstract*—**Microblogging and Social media platforms like twitter, Facebook, etc. are very much popular among the youth. One can easily post any thought anytime using these platforms. Many times these posts belong to radical messages. These radical posts are one of the major social issues. This problem affects people everywhere in the world. This fosters a hostile, contentious, and discouraging atmosphere, which easily impacts the youth. Social media is the primary platform for radical people. They are using this as a weapon to spread their propaganda. It is important to quickly find and stop these radical messages on social media. In this article we proposed hybrid deep learning model DCLSNet using whale optimization technique for timely detection of radical message. We compared the performance of different baseline deep learning models with this model. This model outperforms than the baseline deep learning models. The F1-Score is 0.96 of DCLSNet. Further we used BERT, DistilBERT and RoBERTa transformer. BERT, RoBERTa and DistilBERT F1-Scre is 0.94, 0.95 and 0.92 respectively. These transformers have to be fine-tuned on the training data and then their performance is almost as good as DCLSNet in term of accuracy and F1-Score. But the complexities of these transformers are very higher than the proposed model. The proposed hybrid model is consuming low computing resources. It can be used by the administrators to detect the radicalization timely.

***Keywords*—**Radicalization, sentiment analysis, Deep learning, Twitter, social media, terrorism, extremism, DCLSNet-Deep CNN LSTM Network.

I. INTRODUCTION

Social media enjoys immense popularity among people. Facebook witnesses a staggering more than two lakh status alterations every minute, and Twitter registers over half a million tweets posted within the same timespan[1][2]. With billions of users, social media provides a platform to reach a vast audience quickly, making it the preferred choice for radicalization. Radical person aims to promote their agenda through social media by disseminating extremist content and misleading young individuals. These posts on social media affects people sentiments. Social media users often use abbreviations in their posts to spread their ideas, making it hard to understand what they mean. Many radical groups use social media to push their agendas, and they have a lot of followers. These followers regularly post extreme messages to influence others. When these groups take action, their supporters flood social media with thousands of posts all at once. They propagate hatred by disseminating their content through platforms like twitter, Facebook, etc. Additionally, various online blogs [3] run by these organizations are dedicated to promoting hatred [4]. They employ various tactics to attract young individuals.

Initial efforts to combat online radicalization began by influencing people. [5]. Early detection methods relied on manual techniques, as advanced technologies were not readily available. Automated identification methods were introduced after 2006 [6]. By 2013, Machine Learning techniques were used for automatic detection, coinciding with the rapid increase in social media users. Data from platforms such as YouTube, Facebook, and Twitter were collected for training ML models. SVM and Naïve Bayes algorithms demonstrated promising performance [7]. Concurrently, deep learning techniques gained popularity in various NLP process due to their effective feature extraction capabilities. Today, transformer models like BERT and RoBERTa have become the state-of-the-art solutions for numerous NLP tasks.

Sentiment analysis and opinion mining is one of the most used technique to get the sentiment of the people. Sentiment analysis classifies the sentiments of the people in different classes. These classes can be categorized in binary, ternary or multiple for example positive/negative, positive/neutral/negative or highly-positive/highly-

[1] *Corresponding author: Department of Computer Science and Information Technology, MJP Rohilkhand University, Bareilly 243006, India

[2] Department of Computer Science and Information Technology, MJP Rohilkhand University, Bareilly 243006, India

[3] Department of Computer Science and information, Applied College, Taibah University, Madinah 42353, Saudi Arabia

negative/ positive/neutral/negative. Sentiment analysis is used in many fields like politics, product reviews, healthcare, drug reviews etc. Many organizations specially have the procedure to get the review from their consumers to know their sentiments. So, that these companies can correct the product or services as per the consumer needs. This data is personal to them. But people have another place—social media—where they can post their thoughts about products, services, and the situations surrounding them.

Identifying radicalization is a challenging task. There are very few datasets available to help with this, and they often lack labels. To create a system for spotting radical behavior online, we first need to mark and categorize these datasets. To stop radical users and their posts on social media, we need a fast and accurate system that can quickly recognize and report them. There should be an AI technique that can detect and prevent these kind of messages timely. The efforts to detect and prevent such messages were initiated but couldn't provide state-of-the-art solution. In this article we develop a deep learning model DCLSNet. It is one of the finest model that compete transformers performance with a few resource usages. This model can detect and classify the radical messages in to positive, negative, and neutral classes. Positive class confirms that the message is not radical and does not harm the society. Neutral class identifies that the message is not harming the society however the message can be radical. Negative class identifies that the message is radical and such kind of messages must be identified and removed.

A. Contribution of the research

- The purpose of this research is to develop an efficient deep learning model that consume very few resources and detect the radical messages from social media.
- This study proposes DCLSNet hybrid deep learning model using whale optimization) [8]. This technique provides best hyper-parameters. The layered architecture of DCLSNet is shown in Fig. 1.
- This study explores the different embeddings techniques of NLP for example –Trainable-Embedding-layer (WE-1) and Glove(WE-2).
- This study compares the proposed model with other baseline deep learning models (LSTM, GRU, Bi-LSTM) as well as state of the art transformers (BERT, RoBERTa, DistilBERT).

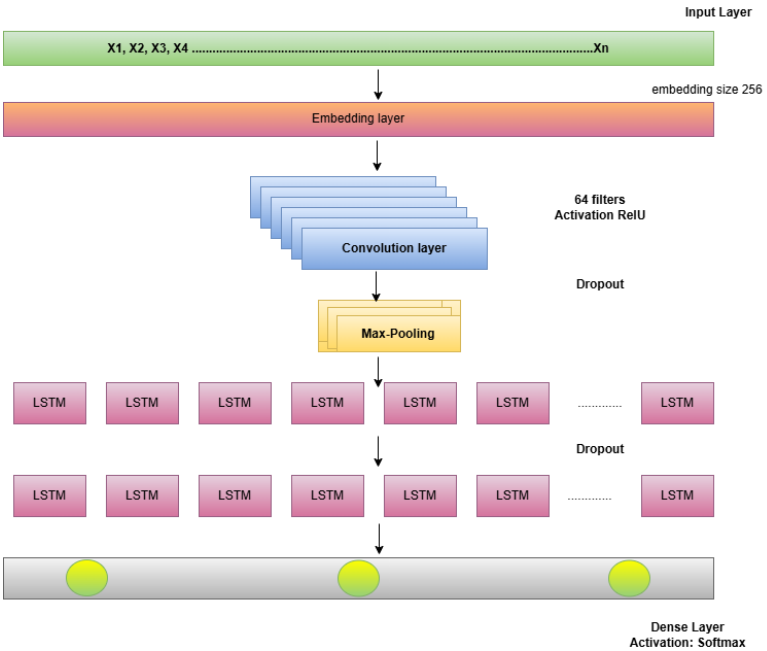

Fig. 1. DCLSNet Layered Architecture

II. Related Work

Research on online radicalization can be categorized into three main areas, as identified by [9]: analysis, detection, and prediction. Additionally, the authors explore the nuances of automatic extremist detection and content prediction. This analysis underscores the deficiencies in online radicalization monitoring, which include a lack of validated data, limited researcher collaboration, the changing landscape of extremist language, and

ethical concerns.

A critical foundation for radicalization research is the analysis of hate speech. Consequently, conducting a comprehensive literature review becomes imperative when identifying hate speech, as emphasized by [10]. This study delves into the characteristics of hate speech and proposes a new definition. Research efforts are divided into distinct categories, such as racism, sexism, and prejudice against refugees. However, the limited availability of publicly accessible datasets poses a challenge. Researchers also face difficulty in determining the most effective categorization techniques, as different scholars employ diverse measures and datasets. As highlighted by Fortuna et al., for scholars engaged in the field of online radicalization, comparing datasets, methodologies, and metrics is of paramount importance.

Furthermore, Al-Hassan et al. (2019) raised fundamental questions regarding hate speech and the identification of hate speech[11]. Unfortunately, they did not provide the dataset or any validation method for their dataset. Their study primarily focuses on the challenges of detecting Arabic hate speech within the Arabic context. Notably, previous research has not adequately concentrated on dataset analysis or performance metrics, as evidenced in references [9]-[11].

Correa and Sureka highlight the objective of online radicalization studies, which is to provide valuable data for enhancing the decision-making processes of law enforcement agencies (LEAs). These studies primarily encompass two categories of analysis: content-based and network-based methods [12].

To categorize publications related to detection, techniques such as web mining and text mining can be employed [12]. While text classification strategies aim to construct classification models [13], often combined with additional factors like social dynamics [13],[14] web mining studies focus on the identification of radical online content, using methods like targeted scanning [15].

To gain a deeper understanding of online radicalization, various forms of analysis, including content-based and network-based approaches, have been recommended for detection. Content-based analysis scrutinizes multiple facets of radical texts, encompassing emotions, themes, and aesthetic attributes, while network analysis delves into social connections.

Numerous studies have delved into prediction. For instance, Ferrara et al. propose a machine learning approach for detecting extremist supporters, predicting the adoption of extremist content, and forecasting communication with terrorists (direct message replies) [16].

This framework considers three feature categories: temporal, network-related, and user activity. Agarwal and Sureka focus on two aspects: the automated detection of web-based radicalization and the prediction of events linked to violent incidents[17]. Most studies leverage spatiotemporal variables as discriminative factors for event prediction. López-Sáncez et al. introduce a method for predicting the likelihood of radicalization, suggesting the creation of alerts based on users' observed propensity for radicalization and the emotional impact of the re-tweets they receive [6].

Sentiment analysis has yielded valuable insights, particularly in understanding the radicalization process. For example, in the process of radicalization, users tend to discuss political topics before becoming active, frequently using terms with negative connotations. Once they become active, there is a shift towards using more religious terminology [13].

Furthermore, the public's response to a terrorist attack is a significant factor. Dewan et al. conducted an analysis of sentiment in Facebook posts, encompassing both text and image analysis[18]. They observed that the sentiment in textual posts initially started as negative but eventually shifted towards positive. In contrast, the sentiment in shared images began positively but turned negative within a few hours.

While many studies emphasize the significance of sentiment features [19], [20], and [21], other experiments [22] indicate that sentiment features, including word unigrams with sentiment, do not outperform the use of unigrams alone for classification.

A. Comparison

We compared our findings with prior research as shown in Table 1. Most previous studies employed traditional machine learning classifiers like SVM, MaxEnt, and NB, known for their faster training times. These models often achieve lower performance metrics (accuracy, F1-score, precision, recall) compared to recent deep learning approaches. in the above table we can see that deep learning models as well as transformer models were used by many authors recently, but they could not reach towards the correct hyper-parameters so their performance was low. In proposed model we use a different technique, which is whale optimization to identify

the correct hyper-parameters and achieve state-of-the-art performance.

TABLE I. : COMPARISON OF PROPOSED APPROACH WITH PREVIOUS APPROACHES

Author	Method	Description/ Findings	Application	Dataset	Performance
A. A. Ahmed et al. (2023) [23]	NB, SVM, KNN, DT, RF, ANN with Unigram, bi-gram and trigram	SVM with Unigram produced highest accuracy: 81.097 and NB is with Bi-gram produced second highest accuracy:78.048 and show TF-IDF is the best feature extractor b/w TF & TF-IDF.	extremism detection	ALSA- Arabic tweets	ACC:81.09
M. Gaikwad et al. (2022) [24]	BERT, RoBERTa, and DistilBERT	DistilBERT - F1 SCORE 0.72 , Accuracy - 0.72, RoBERTa - F1 SCORE 0.71 , Accuracy - 0.68	Radical Detection	MWS-Merged ISIS & White supermasist	Highest F1 SCORE 72
Saini. et al. (2021) [25]	Machine Learning : SVM, LBoosting, RF, MaxEnt were used	Online conversations on terrorism recruitment or creating new links for recruitment .	Radical Recruitment detection	Five dark web discussion forums	
S. R. Muramudalige et al. (2021) [26]	Graph Search Algo similarity based user and group matching. Created PINGS open source library	Algo are used on three datasets and results are accurate.	Radical Detection	Radicalization Dataset, Mimic dataset, crime dataset	
J.-d.-J. Rocha-Salazar et al. (2021) [27]	Phase-1: fuzzy logic is applied. Phase-2: unsupervised clustering is applied. Phase-3 introduce abnormality indicator applied to the riskiest cluster.	prediction cost, humor effort cost and research cost is reduced.	predicting terrorism funding & money laundering	Data collected from Mexico Financial Institution	
Kaur et al. (2019) [28]	SVM, RF, MaxEnt, LSTM	classified into three classes Radical(R), Non-Radical (NR) and Irrelevant (I), LSTM achieved best precision: 85.90%	Radical Detection	news, articles and blogs	PRECISION: 85
Fernandez et al. (2018) [14]	propose a computational approach for detecting radicalization, used J48, NB, LogR, CF algorithm	classified 112 pro-ISIS vs.112 “general" Twitter users. Performance of classifiers is: f1 score is 0.9 and Precision is in between 0.7 to 0.8	Radical Detection	tweets	F1- 90
Barhamgi et al. (2018) [29]	semantic web and domain ontologies	Messages and posts on social networks can be automatically mined for radicalization signs using semantic web and domain ontologies.	Radical Detection		
Proposed Model	**DCLSNet with Glove and Trainable Embeddings**	**Radical Messages and Post on Twitter. CNN + LSTM with trainable Word embedding ACC: 94 and CNN + LSTM with Glove embedding ACC: 96**	**Radical Detection**	**tweets**	**ACC: 96 F1-Score: 96**

III. MATERIALS AND METHODS

Our main focus lies in understanding the sources of radical messages and the language employed within these messages. Specifically, we aim to pinpoint web blogs and social media posts that disseminate radical content and identify phrases that are more strongly associated with one category than another. The primary aim of this study is to detect radicalization and provide timely information to relevant authorities.

We used RNN, LSTM, Bi-LSTM, GRU and hybrid model DCLSNet using whale optimization and test the model performance on the radicalization dataset. We are planning to predict the radical messages so that it could be blocked by the social media platforms. We also used transformers like BERT, RoBERTa, DistilBERT on the same dataset and compare the performance with aforesaid models.

A. Dataset Preparation

In this field, the availability of datasets is quite limited, and many existing datasets lack annotations. Some of the datasets we've compiled include the ISIS dataset, which was sourced from Kaggle. It contains 17,410 Twitter messages from various pro-ISIS users and their followers. However, this dataset lacks class labels, so it necessitates the assignment of classes for computational purposes.

Another dataset is the Terrorism Incidents available on kaggle, comprising 38998 records of various activities. To create a comprehensive dataset, we merged all the datasets. Subsequently, we kept common attributes. Then we have performed data-pre-processing to clean the text messages. we have used python- TextBlob[30], and VADAR[31] library to annotated all the text, messages, and tweets into three classes: Positive, Negative, and Neutral. Dataset preparation steps are show in the Fig. 2. Final distribution of the classes is represented in the Table II.

TABLE II. LABEL COUNT OF THE DATASET

Label	Description	Count	Ratio
Positive	Indicate the Non-violent Radical text.	17026	30.18
Negative	Indicate the Violent Radical text.	20449	36.25
Neutral	Indicate the text is not Radical.	18934	33.56

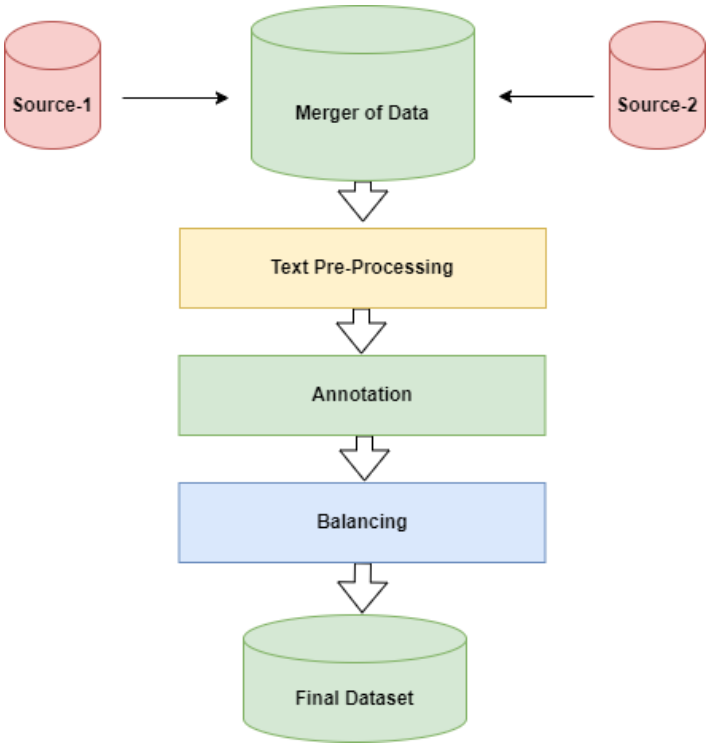

Fig. 2. : Dataset Preparation Steps

a. Text Pre-processing

This data is imbalanced for training purposes. First we applied text pre-processing steps to the dataset described below. In text processing we have removed duplicate messages. After removal of duplicate review, we got 16741 positive labels. Then we apply the random removal method on the negative and neutral class balance. After removing extra records, we have 16741 records in each class. Now our data is well balanced.

In the Fig. 3 we have plotted a graph in which we have shown the top radical words and non-radical words with their frequencies. We can see some of the words comes under both category, so we can easily understand that we cannot apply any method that is calculating polarity by looking words only. we have to capture the entire sequence context so that we could get more accurate classification. In the Fig. 4 we have created word-cloud. Word-cloud help us to identify key terms and their importance in the dataset. In word cloud one can easily understand the nature of words in the dataset.

In our model, we employ a sophisticated approach to harness data features, combining a 1-D Convolutional Neural Network (CNN) with Long Short-Term Memory (LSTM) for a substantial enhancement in various performance metrics, including accuracy, F1-score, precision, and recall.

B. Proposed Model - DCLSNet

In our model, we employ a sophisticated approach to harness data features, combining a 1-D Convolutional Neural Network (CNN) with Long Short-Term Memory (LSTM) for a substantial enhancement in various performance metrics, including accuracy, F1-score, precision, and recall.

Specifically, our model features a 1-D CNN layer configured with a kernel size of 3 and 32 filters, complemented by a Rectified Linear Unit (ReLU) activation function. This powerful combination excels at extracting meaningful patterns from the data. The resulting feature map from the CNN is then subjected to a

pooling layer, with a pooling size of 2, to further distill the most salient information.

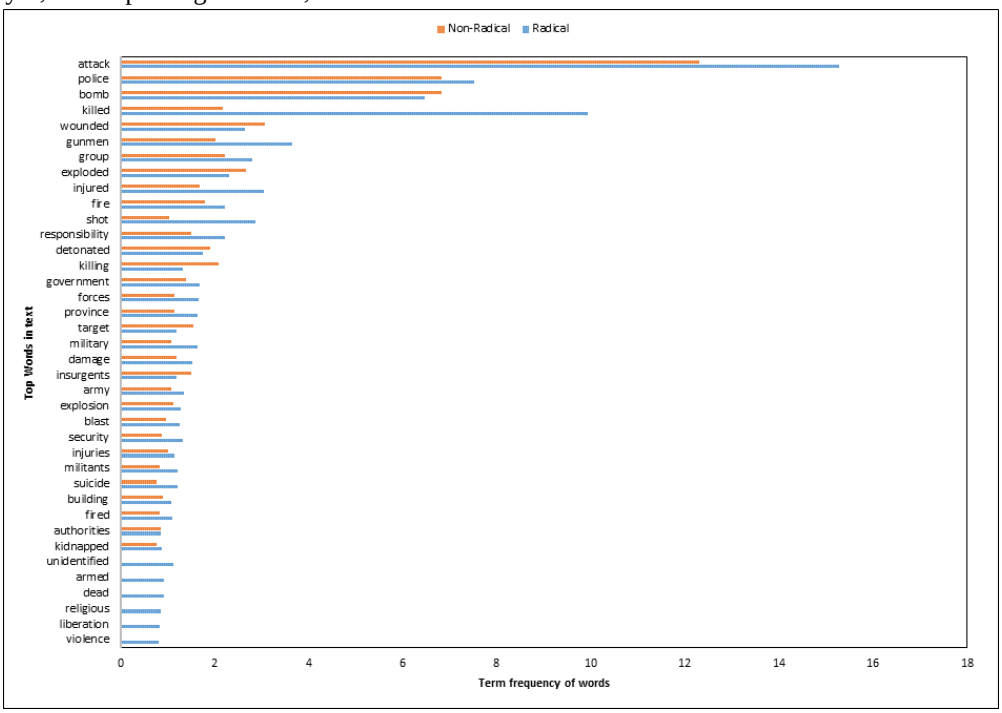

Fig. 3. : Frequency of radical and non-radical words from the dataset

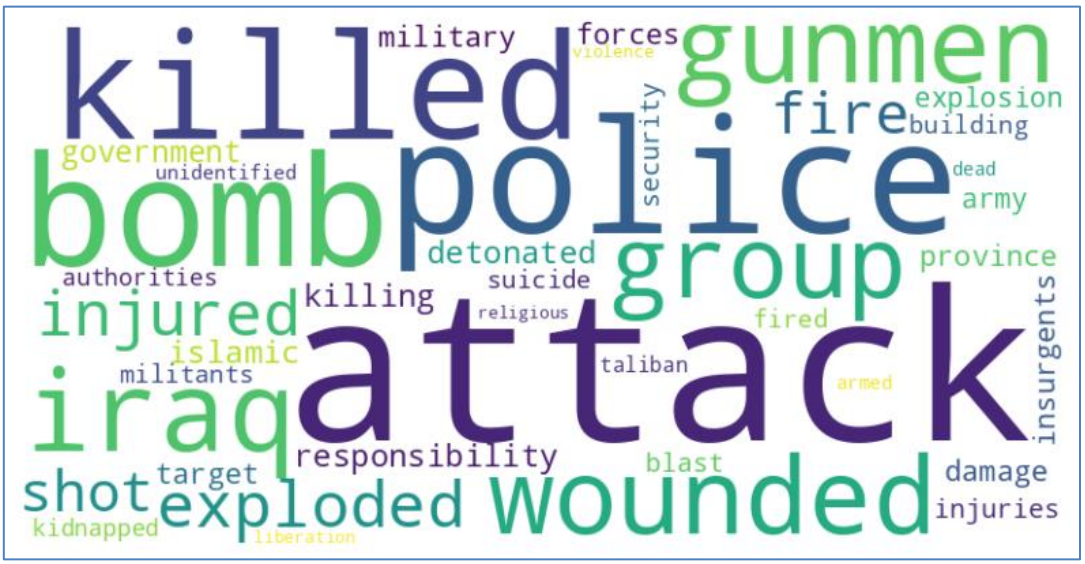

Fig. 4. : Word Cloud of the top words from the dataset

The convolution operation for each filter f (where f = 1, 2, ..., 32) can be expressed as:

$$Z_f(t) = ReLU\left(\sum_{i=0}^{k-1} W_f(i) \cdot X(t+i) + b_f\right) \quad (1)$$

Here, W_f represents the weights of the filter f, b_f is the bias term, and t is the time step. The result is a feature map Z with 32 channels.

The pooling operation typically max-pooling can be represented as:

$$P_f(t) = max\left(Z_f(2t), Z_f(2t+1)\right) \quad (2)$$

Here P_f is the pooled feature map for the filter f.

Subsequently, the output of the pooling layer is directed to the LSTM layer, enabling the model to effectively capture and utilize both short-term and long-term dependencies within the data, a critical aspect of improving performance in tasks that benefit from sequential information processing. This architecture has proven to be

highly effective, yielding superior results across accuracy, F1-score, precision, and recall, making it a valuable asset in a wide range of applications. The operation of passing pooled features in LSTM can be represented as:

$$f_t = \sigma\left(W_f . [h_{t-1}, P_t] + b_f\right) \quad (3)$$

$$i_t = \sigma(W_i . [h_{t-1}, P_t] + b_i) \quad (4)$$

$$\widetilde{C_t} = tanh(W_c . [h_{t-1}, P_t] + b_c) \quad (5)$$

$$C_t = \left(f_c . C_{t-1} + i_t . \widetilde{C_t}\right) \quad (6)$$

$$O_t = \sigma(W_o . [h_{t-1}, P_t] + b_o) \quad (7)$$

$$h_t = O_t . tanh(C_t) \quad (8)$$

Where f_t represent the output of forget gate, i_t represent the output of input gate, $\widetilde{C_t}$represent the output of cell candidate, C_t represent the output of cell state, O_t represent the output of output gate, and h_t represent the output of the hidden state. The output h_t from the LSTM layer effectively captures both short-term and long-term dependencies in the data. we have passed the output h_t on the next LSTM layer which produce the final output *H.*

On the last the network has a dense layer. This layer has a Soft-max activation function to classify the text into three classes. This can be represented as:

$$y = \sigma(W_d . H_T + b_d) \quad (9)$$

Where W_d is the matrix of weights H_T is the output of final LSTM layer and $b_d$is bias vector. The output of dense layer y has the scores of each class *i.* the soft-max function converts these scores into probability of these class. This can be represented as:

$$\hat{y}_i = \frac{exp(y_i)}{\sum_{J=1}^{3} \exp(y_j)} \quad (10)$$

This sequence of operations completes the process of classifying the input data into one of three classes using the final dense layer with soft-max activation. The soft-max output $\hat{y}_i$ provides the probability distribution over the three classes, allowing the model to make a final classification decision based on the highest probability.

C. Whale optimization technique

Whale optimization technique(WOT) is inspired by nature. As the whale finds prey, it encircles and attack it. Whale always explore the location of their prey. The Algorithm is also start with different parameters initialization for many whales. It tries to compare the performance outcomes using different whale parameters. It continuously updates the location of the best whale based on performance. The algorithm iterates until it finds the best parameters for the optimal whale. Finally, we can use these parameters in the models [8] [32].

Whale optimization technique

$best\_performance = 0$
$Initialize\ whale\_population\ with\ different\ parameters\ initialized\ with\ different\ values$
$For\ loop\ to\ iterate\ upto\ whale\_population$:
$Select\ one\ whale$
$Create\ a\ model\ with\ whale\ parameters$
$train\ the\ model$
$Evaluate\ the\ performance\ on\ test\ set$
$If\ performance > best\_performance$:
$best\_whale = selected\_whale$

$if\ best\_whale\ found$:
$update\ numerical\ parameters$

$Train\ the\ final\_model\ with\ best\_whale\ parameters$

D. BERT Bidirectional Encoder Representations from Transformers

Bidirectional Encoder Representations for Transformers (BERT) use the transformer attention model and employs an encoder-decoder architecture [33]. BERT processes text sequences in both left-right and right-left

ways. This makes BERT efficient and accurate. BERT's achieved state-of-the-art performance in wide range of NLP problems. It has been trained on huge corpus of text and we can use it to fine tune on smaller dataset so that we can achieve the better performance in the domain of radicalisation.

E. Feature Extraction

Deep learning techniques have introduced a powerful approach for embedding, with word embedding being one of the standout methods. This technique has gained widespread favour among practitioners in the field of deep learning, including our own work, where we leverage it to extract essential features. In addition to word embedding, we have harnessed another word vectorization method, Glove, to further enhance our feature extraction capabilities. Remarkably, we've trained our model to effectively utilize both of these input sources.

Glove(WE-2) and trainable word embedding(WE-1) stand as prominent techniques for feature extraction from textual data. They excel in capturing the contextual nuances of sentences and transforming them into meaningful feature vectors. It's important to note that deep learning algorithms, as described earlier, aren't inherently compatible with textual inputs. To bridge this gap, we've undertaken the crucial task of converting these textual inputs into numerical vectors. Our approach involves establishing a vocabulary based on the dataset and ensuring uniform sentence lengths through padding. Subsequently, we've applied both Glove and word embedding, yielding two distinct feature vector representations.

a. Conv-1D feature Extraction

Further the deep learning models have the capacity to generate the features itself. The 1D Convolutional neural network is extracting features from the textual data where the temporal and spatial relationships between elements are important. Let the input sequence X with length L and feature dimension F:

$X = [x_1, x_2, x_3 \ldots . x_L] \ where \ x_i \ \in \mathbb{R}^F$ (11)

A 1D convolution operation involves a kernel (or filter) that slides over the input sequence. The kernel has a size *K* and the same feature dimension *F*:

$K = [k_1, k_2, k_3 \ldots . k_L] \ where \ k_i \ \in \mathbb{R}^F$ (12)

The convolution operation for a given position t can be represented as:

$(\mathrm{X} * \mathrm{K})_t = ReLU(\sum_{i=1}^{K} \mathrm{X}_{t+1-1} \cdot K_i)$ (13)

As the kernel slides across the input sequence, it generates a feature map *F* After the convolution activation function ReLU is applied to introduce non-linearity.

1D convolutional layer extracts features from the input sequence by applying a kernel across the sequence. The result of the convolution operation is passed through an activation function to introduce non-linearity, and optionally, a pooling layer can be applied to reduce the dimensionality of the feature map while retaining the most significant features. This process allows the CNN to learn and extract important patterns from sequential data.

b. LSTM feature extraction

The mathematical operations as shown in the equation number (3) to (8) within the LSTM cell enable it to learn and remember important patterns over long sequences, making it highly effective for tasks involving sequential data.

c. BERT feature extraction

BERT is based on the Transformer architecture, specifically the encoder part of the Transformer. It leverages self-attention mechanisms to learn contextual relationships between words in a text. Following is the example of BERT tokenization and embedding-

Given an input sequence, e.g., "I love machine learning":

Tokenization:

$$\text{Tokens} = \left[[\text{CLS}], \text{I}, \text{love}, \text{machine}, \text{learning}, [\text{SEP}]\right]$$

Embedding:

Combine token, segment, and position embeddings.

$$E = [\, E_{[CLS]}, E_I, E_{love}, E_{machine}, E_{learning}, E_{[SEP]} \,]$$

$E_{[CLS]}$ is the starting and $E_{[SEP]}$ is the ending of each sentence.

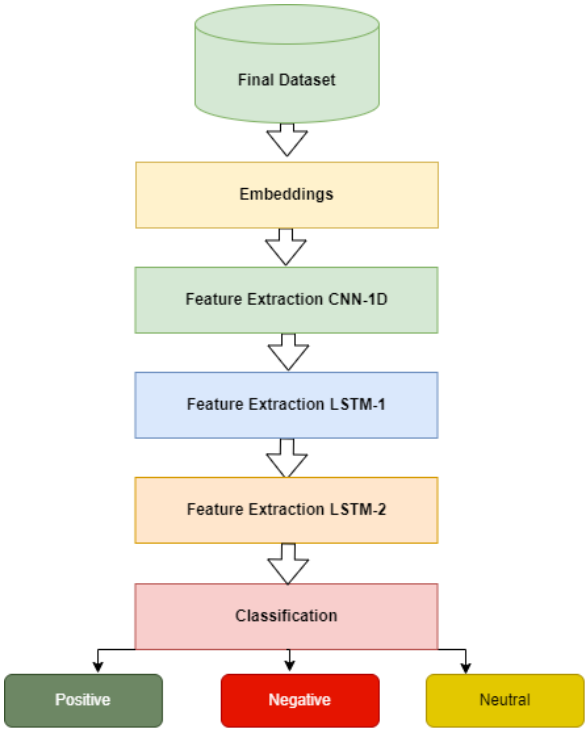

Fig. 5. : Data Flow into DCLSNet

Transformer Encoder Layers:
Apply multiple layers of self-attention and feed-forward networks. Its Output is contextualized embeddings for each token. The final hidden state for each token.

$$BERT\ Output = [H_{[CLS]}, H_I, H_{love}, H_{machine}, H_{learning}, H_{[SEP]}]$$

$H_{[CLS]}$ can be used for sequence-level tasks (e.g., classification). $H_{[token]}$ can be used for token-level tasks (e.g., NER).
BERT extracts features by leveraging its deep bidirectional Transformer encoder. Through multiple layers of self-attention and feed-forward networks, BERT learns rich contextual representations of the input text, capturing both the left and right context for each token. These features can be used for various downstream NLP tasks, often leading to state-of-the-art performance.

F. Data Splitting

It is necessary to assess the performance of the trained model. We employed an 80% data for the training and 20% data for testing. In the training data we have again split data 80% for training and 20% validation purpose.

G. Performance Measure and Evaluation Metrics

Performance assessment is carried out after model training, during which we retain the predictions generated for the test dataset for each of the models being evaluated. Following this step, a confusion matrix is meticulously constructed. The resulting equations are employed as metrics to measure performance, encompassing factors such as accuracy, F1-score, recall, and precision.

$$\text{Accuracy}=\frac{TP+TN}{TP+TN+FP+FN} \quad (14)$$

$$\text{Precision} =\frac{TP}{TP+FP} \quad (15)$$

$$\text{Recall}= \frac{TP}{TP+FN} \quad (16)$$

$$\text{F1-Score} =2 * \frac{Precision*Recall}{Precision+Recall} \quad (17)$$

IV. Results and Discussion

We have used several baseline deep learning algorithms like RNN, LSTM, Bi-LSTM, and GRU for classification with WE-1 and WE-2 word-embedding. We have provided the comparison in baseline deep learning models with the performance matrix - Accuracy, F1-Score, Recall and precision in following Table III. In this table we can observe that all the performance metrics having same results. The reason of this similarity is the balanced dataset. It can be clearly observed that proposed hybrid model is performing better than the other baseline models. WE-2 provides the best results for the proposed model. We have plotted a performance graph using F1-Score in Fig. 6 and Fig. 7. Fig. 6 shows the performance using WE-1 and Fig. 7 shows the performance using WE-2.: Comparison of proposed DCLSNet with Baseline Models.

TABLE III. Comparision of Proposed Model Performance with baseline deep learning models using WE-1 and WE-2

	WE-1				**WE-2**			
Model	**Accuracy**	**Precision**	**Recall**	**F1-Score**	**Accuracy**	**Precision**	**Recall**	**F1-Score**
RNN	77	77	77	77	79	79	79	79
Bi-LSTM	87	87	87	87	88	88	88	88
GRU	84	84	84	84	85	85	85	85
LSTM	85	85	85	85	86	86	86	86
Proposed Model (DCLSNet)	**94**	**94**	**94**	**94**	**96**	**96**	**96**	**96**

TABLE IV. Comparision of parameters, size, epochs and f1-score with proposed model

Model	**No. of parameters**	**size**	**epoch**	**F1-Score**
RNN	12849667	49.02	500	79
GRU	12948611	49.4	500	85
LSTM	12997507	49.58	500	86
Bi-LSTM	13195011	50.33	500	88
Proposed Model (DCLSNet)	**30746739**	**147**	**500**	**96**
DistilBERT	66365187	317.29	500	92
BERT	109484547	523.44	500	94
RoBERTa	124647939	595.94	500	95

We can see the comparison on the basis of parameters, size and F1-Score in TABLE IV. It is clearly observed that the number of parameters and size in RoBERTa is very high in transformers. BERT is having the second highest parameters and DIstilBERT is having the least parameters in the given transformers. This is a clear indication of the resource usage would be the highest in RoBERTa, second highest in BERT, and the third highest in DistilBERT. Still these transformers are performing similar to the proposed hybrid model. Fig. 8 shows the performance of proposed model with transformers.

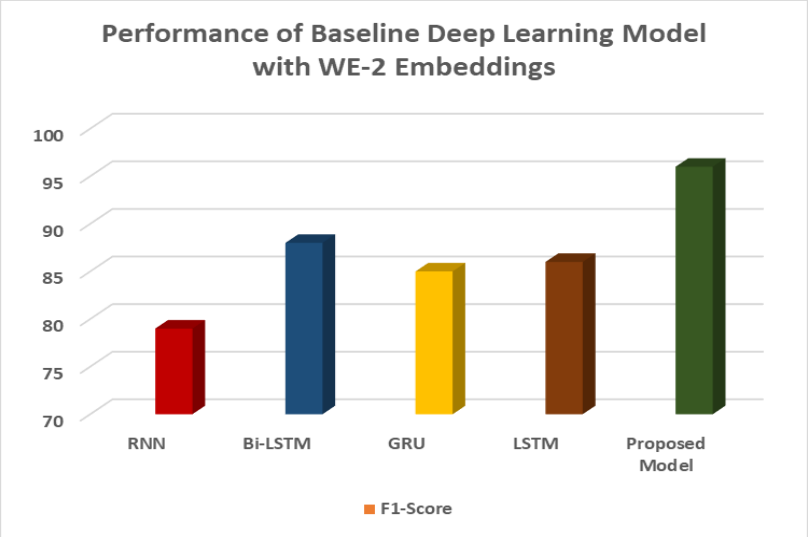

Fig. 6. : Experimental Results of Baseline deep learning models with proposed Model using WE-2

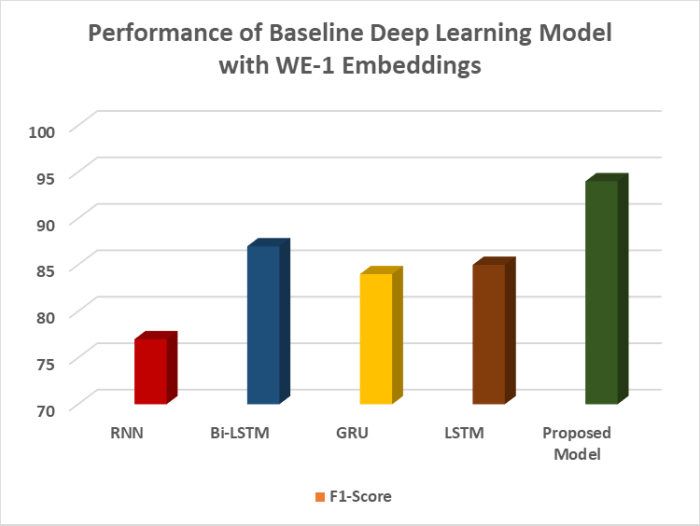

Fig. 7. : Experimental Results of Baseline deep learning models with proposed Model using WE-1

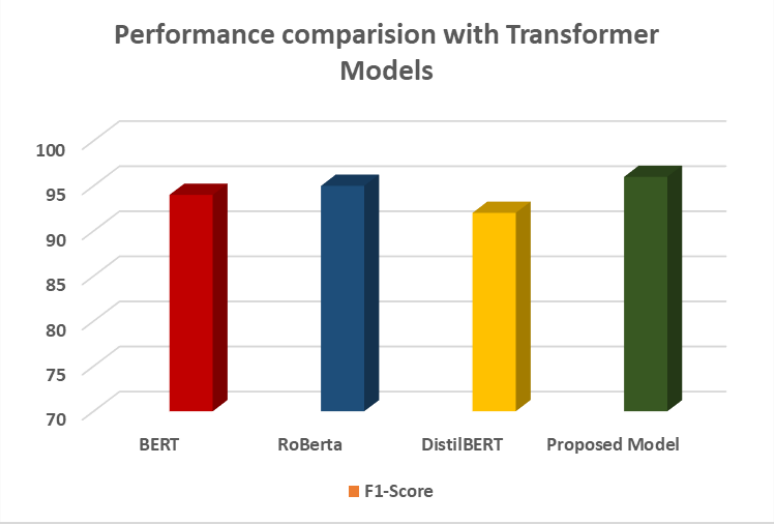

Fig. 8. : Experimental Results of Transformer models with proposed model

A. Comparison with transformers BERT, RoBERTa, and DistilBERT model

We have used three transformers. RoBERTa led in accuracy - 95% however it is one of the complex model. It has 12,46,47,939 parameters that is the highest parameter count in all transformers. This feature shows its complexity is highest. Second transformer is BERT have the accuracy of 94% which is slightly 1% lesser than RoBERTa. But number of parameters are reduced by 1,51,63,392. And DistilBERT having 92% accuracy which is less than other two transformers. DistilBERT has only 6,63,65,187 parameters which is approximately half from the RoBERTa but provide much better accuracy.

Overall, the transformer model is providing the performance nearby proposed model. But in term of complexity and resource requirement of transformers is huge. The proposed modal outperforms from all other models including transformers in terms of performance metrics for the given task.

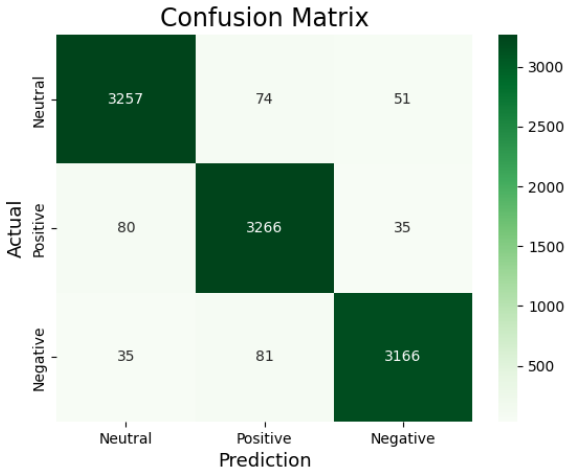

Fig. 9. :Confusion Matrix of proposed model

V.CONCLUSION

Humans are frequently described as sensitive and vulnerable to emotional effects. Unfortunately, some individuals exploit this vulnerability and use emotional manipulation for malicious purposes. To combat this issue, we propose to identify and address harmful messages based of hybrid deep learning using whale optimization technique, thereby safeguarding humans from emotional exploitation. In our experiments, the proposed hybrid model-DCLSNet consistently outperformed others and offering a potent tool for understanding human sentiments. Additionally, the BERT, RoBERTa, and DistilBERT transformer models stand as a benchmark, setting the standard for state-of-the-art performance in text classification, sentiment analysis and intent detection. The proposed model is a light-weight hybrid model. It consumes less computation resources than transformer models. This model achieves 0.96 F1-Score. This model is performing equivalent to BERT that achieved F1-score of 0.94 and RoBERTa that achieved F1-score of 0.95. Optimization techniques can help to provide the best hyper-parameters. In future work we will work upon other optimization techniques. Through these efforts, we aim to protect and empower young individuals by ensuring their emotions are respected and shielded from malicious manipulation. It will help administration to timely detect and prevent such harmful messages.

ACKNOWLEDGMENT

I am thankful to my research guide Prof. Ravendra Singh. He gives me the path towards my research journey. I am also thankful to Mohammad Zubair Khan. He enlightens my path at every stage. He is very humble and intelligent researcher. This work is completed under the support and guidance of these two prominent researchers.

REFERENCES

[1] J. Schultz, “How Much Data is Created on the Internet Each Day?,” *Micro Focus Blog*, no. 09.08.2016, p. Stories and updates from our team, partners and su, 2016, [Online]. Available: https://blog.microfocus.com/how-much-data-is-created-on-the-internet-each-day/%0Ahttps://blog.microfocus.com/how-much-data-is-created-on-the-internet-each-day/#

[2] A. Chavan, "Twitter data storage and processing," 2020, [Online]. Available: https://ankush-chavan.medium.com/twitter-data-storage-and-processing-dd13fd0fdb30

[3] J. F. Binder and J. Kenyon, "Terrorism and the internet: How dangerous is online radicalization?," *Front. Psychol.*, vol. 13, p. 997390, 2022, doi: 10.3389/fpsyg.2022.997390.

[4] J. Wilson, "Leak from neo-Nazi site could identify hundreds of extremists worldwide," *Guard.*, 2019, [Online]. Available: https://www.theguardian.com/us-news/2019/nov/07/neo-nazisite-iron-march-materials-leak

[5] E. Tolis, "Investigating the influence of ISIS radicalisation on the recruitment process: a critical analysis," *J. Policing, Intell. Count. Terror.*, vol. 14, no. 2, pp. 129–146, 2019, doi: 10.1080/18335330.2019.1572910.

[6] D. López-Sáncez, J. Revuelta, F. de la Prieta, and J. M. Corchado, "Towards the Automatic Identification and Monitoring of Radicalization Activities in Twitter," in *Knowledge Management in Organizations*, L. Uden, B. Hadzima, and I.-H. Ting, Eds., Cham: Springer International Publishing, 2018, pp. 589–599.

[7] A. Rastogi, R. Singh, and D. Ather, "Sentiment Analysis Methods and Applications-A Review," in *Proceedings of the 2021 10th International Conference on System Modeling and Advancement in Research Trends, SMART 2021*, 2021, pp. 391–395. doi: 10.1109/SMART52563.2021.9676260.

[8] S. Mirjalili and A. Lewis, "The Whale Optimization Algorithm," *Adv. Eng. Softw.*, vol. 95, pp. 51–67, 2016, doi: https://doi.org/10.1016/j.advengsoft.2016.01.008.

[9] M. Fernandez and H. Alani, *Artificial intelligence and online extremism*. Routledge Frontiers of Criminal Justice. Abingdon: Routledge, 2021. doi: 10.4324/9780429265365-7.

[10] P. Fortuna and S. Nunes, "A Survey on Automatic Detection of Hate Speech in Text," *ACM Comput. Surv.*, vol. 51, no. 4, Jul. 2018, doi: 10.1145/3232676.

[11] A. Al-Hassan and H. Al-Dossari, "DETECTION OF HATE SPEECH IN SOCIAL NETWORKS: A SURVEY ON MULTILINGUAL CORPUS," 2019, pp. 83–100. doi: 10.5121/csit.2019.90208.

[12] D. Correa and A. Sureka, "Solutions to Detect and Analyze Online Radicalization : A Survey," vol. V, no. January, pp. 1–30, 2013, [Online]. Available: http://arxiv.org/abs/1301.4916

[13] M. Rowe and H. Saif, "Mining pro-ISIS radicalisation signals from social media users," *Proc. 10th Int. Conf. Web Soc. Media, ICWSM 2016*, no. Icwsm, pp. 329–338, 2016, doi: 10.1609/icwsm.v10i1.14716.

[14] M. Fernandez, M. Asif, and H. Alani, "Understanding the roots of radicalisation on twitter," *WebSci 2018 - Proc. 10th ACM Conf. Web Sci.*, no. May, pp. 1–10, 2018, doi: 10.1145/3201064.3201082.

[15] S. Agarwal and A. Sureka, "Topic-specific youtube crawling to detect online radicalization," *Lect. Notes Comput. Sci. (including Subser. Lect. Notes Artif. Intell. Lect. Notes Bioinformatics)*, vol. 8999, pp. 133–151, 2015, doi: 10.1007/978-3-319-16313-0_10.

[16] E. Ferrara, W. Q. Wang, O. Varol, A. Flammini, and A. Galstyan, "Predicting online extremism, content adopters, and interaction reciprocity," *Lect. Notes Comput. Sci. (including Subser. Lect. Notes Artif. Intell. Lect. Notes Bioinformatics)*, vol. 10047 LNCS, pp. 22–39, 2016, doi: 10.1007/978-3-319-47874-6_3.

[17] S. Agarwal and A. Sureka, "Applying Social Media Intelligence for Predicting and Identifying On-line Radicalization and Civil Unrest Oriented Threats," pp. 1–18, 2015, [Online]. Available: http://arxiv.org/abs/1511.06858

[18] P. Dewan, A. Suri, V. Bharadhwaj, A. Mithal, and P. Kumaraguru, "Towards Understanding Crisis Events On Online Social Networks Through Pictures," in *2017 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining (ASONAM)*, 2017, pp. 439–446.

[19] T. Chalothorn and J. Ellman, "Affect Analysis of Radical Contents on Web Forums Using SentiWordNet," *Int. J. Innov. Manag. Technol.*, vol. 4, no. 1, pp. 122–124, 2013, doi: 10.7763/IJIMT.2013.V4.373.

[20] M. Ashcroft, A. Fisher, L. Kaati, E. Omer, and N. Prucha, "Detecting Jihadist Messages on Twitter," in *2015 European Intelligence and Security Informatics Conference*, 2015, pp. 161–164. doi: 10.1109/EISIC.2015.27.

[21] S. Agarwal and A. Sureka, "Using KNN and SVM Based One-Class Classifier for Detecting Online Radicalization on Twitter," in *Proceedings of the 11th International Conference on Distributed Computing and Internet Technology - Volume 8956*, in ICDCIT 2015. Berlin, Heidelberg: Springer-Verlag, 2015, pp. 431–442. doi: 10.1007/978-3-319-14977-6_47.

[22] H. Saif, T. Dickinson, L. Kastler, M. Fernandez, and H. Alani, "A Semantic Graph-Based Approach for Radicalisation Detection on Social Media," in *The Semantic Web*, E. Blomqvist, D. Maynard, A. Gangemi, R. Hoekstra, P. Hitzler, and O. Hartig, Eds., Cham: Springer International Publishing, 2017, pp. 571–587.

[23] A. A. Ahmed *et al.*, "Arabic Text Detection Using Rough Set Theory: Designing a Novel Approach," *IEEE Access*, vol. 11, pp. 68428–68438, 2023, doi: 10.1109/ACCESS.2023.3278272.

[24] M. Gaikwad, S. Ahirrao, K. Kotecha, and A. Abraham, "Multi-Ideology Multi-Class Extremism Classification Using Deep Learning Techniques," *IEEE Access*, vol. 10, no. October, pp. 104829–104843, 2022, doi: 10.1109/ACCESS.2022.3205744.

[25] J. K. Saini and D. Bansal, "Detecting online recruitment of terrorists: towards smarter solutions to counter terrorism," *Int. J. Inf. Technol.*, vol. 13, no. 2, pp. 697–702, 2021, doi: 10.1007/s41870-021-00620-2.

[26] S. R. Muramudalige, B. W. K. Hung, A. P. Jayasumana, I. Ray, and J. Klausen, "Enhancing Investigative Pattern Detection via Inexact Matching and Graph Databases," *IEEE Trans. Serv. Comput.*, vol. 15, no. 5, pp. 2780–2794, 2022, doi: 10.1109/TSC.2021.3073145.

[27] J. de J. Rocha-Salazar, M. J. Segovia-Vargas, and M. del M. Camacho-Miñano, "Money laundering and terrorism financing detection using neural networks and an abnormality indicator," *Expert Syst. Appl.*, vol. 169, 2021, doi: 10.1016/j.eswa.2020.114470.

[28] A. Kaur, J. K. Saini, and D. Bansal, "Detecting Radical Text over Online Media using Deep Learning," 2019, [Online]. Available: http://arxiv.org/abs/1907.12368

[29] M. Barhamgi, R. Lara-Cabrera, D. Benslimane, and D. Camacho, "Ontology Uses for Radicalisation Detection on Social Networks," in *Intelligent Data Engineering and Automated Learning -- IDEAL 2018*, H. Yin, D. Camacho, P. Novais, and A. J. Tallón-Ballesteros, Eds., Cham: Springer International Publishing, 2018, pp. 3–8.

[30] Lorla dan Steven, "TextBlob Documentation Release 0.16.0," *TextBlob*, 2020, [Online]. Available: https://textblob.readthedocs.io/en/dev/

[31] D. Marutho, S. Rustad, and others, "Sentiment Analysis Optimization Using Vader Lexicon on Machine Learning Approach," in *2022 International Seminar on Intelligent Technology and Its Applications (ISITIA)*, 2022, pp. 98–103.

[32] M. O. Okwu and L. K. Tartibu, "Whale Optimization Algorithm (WOA)," in *Metaheuristic Optimization: Nature-Inspired Algorithms Swarm and Computational Intelligence, Theory and Applications*, Cham: Springer International Publishing, 2021, pp. 53–60. doi: 10.1007/978-3-030-61111-8_6.

[33] J. Devlin, M. W. Chang, K. Lee, and K. Toutanova, "BERT: Pre-training of deep bidirectional transformers for language understanding," *NAACL HLT 2019 - 2019 Conf. North Am. Chapter Assoc. Comput. Linguist. Hum. Lang. Technol. - Proc. Conf.*, vol. 1, no. Mlm, pp. 4171–4186, 2019.