Deep Learning-Driven Detection of Terrorism Threats from Tweets Using DistilBERT and DNN

Divya S[1]*, B Ben Sujitha[2]
Research Scholar-Department of Computer Science and Engineering,
Noorul Islam Centre for Higher Education, Kumaracoil, Thuckalay, Tamil Nadu, India[1]
Professor-Department of Computer Science and Engineering,
Noorul Islam Centre for Higher Education, Kumaracoil, Thuckalay, Tamil Nadu, India[2]

Abstract—**As globalization accelerates, the threat of terrorist attacks poses serious challenges to national security and public safety. Traditional detection methods rely heavily on manual monitoring and rule-based surveillance, which lack scalability, adaptability, and efficiency in handling large volumes of real-time social media data. These approaches often struggle with identifying evolving threats, processing unstructured text, and distinguishing between genuine threats and misleading information, leading to delays in response and potential security lapses. To address these challenges, this study presents an advanced terrorism threat detection model that leverages DistilBERT with a Deep Neural Network (DNN) to classify Twitter data. The proposed approach efficiently extracts contextual and semantic information from textual content, enhancing the identification of potential terrorist threats. DistilBERT, a lightweight variant of BERT, is employed for its ability to process large volumes of text while maintaining high accuracy. The extracted embeddings are further analyzed using a Dense Neural Network, which excels at recognizing complex patterns. The model was trained and evaluated on a labeled dataset of tweets, achieving an impressive 93% accuracy. Experimental results demonstrate the model's reliability in distinguishing between threatening and non-threatening tweets, making it an effective tool for early detection and real-time surveillance of terrorism-related content on social media. The findings highlight the potential of deep learning and natural language processing (NLP) in automated threat identification, surpassing traditional machine learning approaches. By integrating advanced NLP techniques, this model contributes to enhancing public safety, national security, and counter-terrorism efforts.**

Keywords—Terrorism; global safety; terrorist attacks; data mining; artificial intelligence; natural language processing; DistilBERT; deep neural network

I. Introduction

An act of violence against individuals committed to further political or ideological goals is frequently referred to as terrorism [1]. Terrorism aims to erode the rule of law, democracy, and human rights. Terrorism has a direct influence on several human rights, including life, liberty, and bodily integrity [2]. The study of terrorism does not have strict boundaries. It covers the full spectrum of human emotions, attitudes, and behaviours, incorporating a diversity of perspectives along with the associated fears and inclinations.

There has been terrorism- related fatalities in almost every part of the world. Terrorism may harm civil society, impair social and economic advancements, destroy governments, and threaten peace and security [3]. Each of these elements greatly impacts the enjoyment of human rights. Recently, states have implemented counterterrorism measures that have presented significant challenges to human rights and the rule of law [4]. As a counterterrorism strategy, certain regimes have employed torture along with other cruel treatment, often ignoring the legal and practical protections against assaults, like regular, unbiased inspections of detection facilities. Certain states have violated the international legal obligations of non-refoulement by returning suspected terrorists to countries where they truly face the risk of torture or other grave violations of human rights. In certain situations, the use of special courts to try civilians has undermined the independence of the judiciary and reduced the efficacy of traditional court systems. Repressive measures have suppressed the voices of journalists, human rights advocates, indigenous populations, minorities, and civil society. The economic, social, and cultural rights of many people have suffered as a result of the security sector receiving funds that were frequently intended for social initiatives and development assistance.

The adversary does not publicly announce or carry out the attack, and no enemy aircraft fly over the targets. Instead, all that is left are the horrific images of building collapses, plane hijacking, and innocent people murdered or wounded in bombings and shootings that have caused fear and panic around the world. Attacks of this kind have been referred to as "terrorism" [5]. This is not a struggle between states: rather it is a new kind of terror warfare. The enemy may not be a nation-state but rather a small terrorist group based mostly in a developing country. For the sake of their cause, they are willing to give their lives. Thus, it is clear that they are inaccessible to the police and too elusive for the military to hunt down. Conventional forces are designed to fight, they are not designed for unexpected peaceful situations.

Terrorist attack detection has become essential in the contemporary global context due to the growing frequency, sophistication, and unpredictability of terrorist attacks, which pose major threats to national security, public safety, and economic stability. Rapidly developing communication technologies, like the internet and social media have provided terrorists with platforms to plot and execute attacks, therefore, early discovery is crucial to reducing such dangers. Traditional methods of detecting terrorists' activity, such as surveillance and intelligence gathering often fall short because of the volume of information, the complexity of attack preparation, and the secretive character of modern terrorist activities. Real-time analysis of large datasets, such as text, images, videos and audio has demonstrated extraordinary proficiency in identifying patterns suggestive of terrorist operations using Deep Learning (DL) algorithms [6]. DL is capable of efficiently analyzing unstructured data and producing insights that were previously unavailable using traditional techniques. This aids security forces in stopping attacks before they are carried out and makes it easier to identify terrorists' activities quickly. DL can manage complex, multidimensional data and provide immediate practical insights in a constantly shifting danger scenario, making it an essential tool for terrorist attack detection in the current era.

The detection of terrorism threats from tweets has been transformed by the combination of DL and Natural Language Processing (NLP), which allows for the accurate and efficient analysis of large volumes of unstructured text data [7]. NLP methods enable the extraction of context, sentiment, and linguistic patterns, whereas DL models are excellent at identifying subtleties and intricate relationships in text. These methods are essential for recognizing keywords, spotting extreme language, and quickly detecting hidden risks. Their capacity to adjust and pick up on changing linguistic patterns improves prediction skills, which is crucial for counterterrorism operations and public safety. This paper proposes an effective terrorism threat detection model utilizing the advantages of DL and NLP. The major contributions of the paper include:

- To prepare tweet data using efficient text preprocessing methods, ensuring high- quality input for the detection algorithm.
- To effectively tokenize and extract features from tweet data by utilizing the transformer- based architecture of DistilBERT tokenizer, which captures contextual meaning and relationships.
- To design a system for detecting terrorism threats by combining DistilBERT embeddings with a Deep Neural Network (DNN) model, with the goal of accurately classifying tweets as either non- threats or threats.

The remaining sections of the research are arranged as follows: Section II gives a summary of the current studies, highlighting areas that need more investigation. Section III offers a comprehensive explanation of the methodology. Section IV offers the findings derived from the suggested methodology in detail. A discussion is provided in Section V and finally, a summary of the findings is included in Section VI, which gives a conclusion to the paper.

II. Literature Review

Shinde et al. [8] employed data from the Global Terrorism Database (GTD) to create a model for displaying the aspects of terrorist acts. The study used two different graph embedding methods and seven Machine Learning (ML) models to sort the data into groups. These models included K- Nearest Neighbour (KNN), Support Vector Machine (SVM), Decision Tree (DT), Random Forest (RF) and adaboost. The model achieved 90% accuracy. The study did not include the model's generalizability and scalability. An et al. [9] conducted a study to predict and examine the impact of microblogging on terrorist incidents. Emotion analysis was conducted utilizing user, time and feature content and topics were created using a combination of Word2Vec and K means clustering. The proposed Logistic Regression (LR) based approach outperformed six existing classification models with an accuracy of 85% and was able to identify high influence attributes. Jyothilinga [10] used the GTD's historical terrorist attack data to predict future occurrences and causalities. The study made use of ML techniques such as RF, Multilayer Perceptron (MLP) regressors for predicting deaths, DT and MLP classifiers for classifying offenders, and KNN classifiers for classifying weapons. Using time series analysis to predict terrorists' attacks, KNN achieved 92.25% accuracy, while DT and MLP both produced 90 % accuracy for classifying offenders. The study recognized certain limitations, though including the intricacy of terrorist incidents and a lack of thorough data necessary for precise model validation and training.

Gaikwad et al. [11] created the Merged Islamic State of Iraq and Syria/ jihadist White Supremacist (MIWS) dataset with the goal of examining the impact of extremist groups on social media. The effective use of NLP with pretrained models such as BERT, RoBERTa and DistilBERT demonstrated the effective application of DL for identifying extremist content across many approaches with BERT achieved an F1- score of 0.72. Nevertheless, it was pointed out that the dataset relies on terms that can result in false positives from non- extremist sources such as journalists and critics. Yi Feng et al. [12] designed RP-GA-XGBoost, a causality prediction system based on XGBoost to determine if terrorist acts will cause the deaths of innocent people. The proposed approach integrated Principal Component Analysis (PCA) and RF for feature selection and utilized a Genetic Algorithm (GA) to optimize the hyperparameters of XGBoost. The findings showed that, when compared to some- well known ML techniques, the suggested approach performed the best. Ruifang Zhao et al. [13] combined Inverse Distance Weighting (IDW) with a Multilabel K- Nearest (I-MLKNN) based evaluation framework in order to assess terrorist activities. The dimensions of the features of terrorist attack categories were reduced using the Locally Linear Embedding (LLE) dimension reduction technique. Before and after the dimension reduction, the correlation between the attributes was assessed using the Maximal Information Coefficient (MIC). The inverse distance weighting method was subsequently integrated with the MLKNN multi-label classification algorithm on a grid framework. The outcomes of the simulation showed that the I-MLKNN multi-label classification approach is a perfect tool for determining the geographic distribution of terrorist acts.

Lanjun Luo and Chao Qi [14] focused on determining important markers that influence the likelihood of terrorist attacks from a predictive approach. Factors at the event level and the root cause are considered, and they are qualified using

28 indicators. A recursive feature elimination technique using RF kernels was suggested to find the most important ones among the 28 initial signs. The simulation findings demonstrated that the indicators severely degrade when the bare minimum of input indicators was used before the prediction performance. A method for identifying statistically significant change points in time series connected to terrorism that could indicate the onset of events that require attention was developed by Theodosiadou et al. [15]. Before and after the dimension reduction, the correlation between the attributes was assessed using the Maximal Information Coefficient (MIC). The potential of the suggested methodology in successfully identifying such changes was demonstrated by applying it to a real- world dataset. Change point detection techniques can be used to estimate statistically significant changes in the structural behavior of the aforementioned indicators at particular time points by analyzing the resulting time series. Shynar Mussiraliyeva et al. [16] suggested text categorization methods for identifying extremist activity using MLP technologies. The produced corpus was divided into two sections: 3000 words of postings with extremist intent and 15,000 words of non- extremist posts, including news portals and religious materials. According to the simulation findings, the suggested approach classified extremist texts from the collected corpus with a high degree of accuracy.

Ghada M. A. Soliman and Tarek H. M. Abou- El- Enien [17] designed a hybrid computationally intelligent system to assist in making decisions on the subject of terrorism. The suggested hybrid prediction algorithm combined Data Mining (DM) approaches with various Operations Research (OR) and decision support tools. The development, implementation and evaluation of the suggested system have utilized a variety of assessment metrics. The experimental findings showed that the suggested method identify the terrorist group that is responsible for terror strikes in different places. Georgios Koutidis et al. [18] developed a framework based on ML and DL models for predicting the probability of terrorist occurrences in a target region over a certain time frame. Based on RF models, a feature selection procedure was suggested in which each feature's predictive power is determined and only features that are thought to have adequate predictive power are then taken into consideration. The simulation findings showed that the suggested framework improved the predictive ability of the ML and DL models with the suggested feature selection in predicting the probability of terrorist attacks. Ben Chaabene et al. [19] developed a computational model that uses a variety of ML and recommender system techniques to predict how terrorists' actions will affect social networks based on the text and image data. The suggested method comprises two models: the text classification framework and the image classification system. The text classification model comprises LR, NB and SVM. The model for classifying images was based on CNN. The simulation results showed that the suggested model successfully identified and predicted the behaviour of militant terrorists on twitter.

The field of ML and DL techniques for terrorist attack prediction has advanced significantly in recent years, yet there is still a notable gap in the existing literature. Nations with insufficient or missing data find it difficult to understand terrorist behaviour and highlighting particular groups or ideas results in mispredictions. The socio- political, economic and psychological elements that influence terrorism are not adequately represented by current models. Despite their higher accuracy, many models generate false positives, misclassify harmless behaviours associated with terrorism, and create ethical concerns like discriminating against the community. Moreover, it is challenging to comprehend ML models decision- making processes because of their complexity. Models that can adjust and learn from new data in real time are necessary due to the dynamic nature of terrorist actions, but many previous studies rely on static dataset, which reduces their applicability in situations that change rapidly. Overcoming these gaps is crucial to improving the effectiveness and moral applications of DL and ML in predicting terrorist activities, which will ultimately lead to safer communities across the world. So, in order to rectify the above-mentioned limitations, a novel NLP model based on DistilBERT and DNN for the detection of terrorism threats from tweets has been proposed in this paper.

III. Materials and Methods

The proposed approach combines DistilBERT with Deep Neural Network (DNN) in a structured pipeline to identify terrorist threats in tweets. To standardize and clean the input text, raw twitter data is initially collected and processed through preprocessing approaches that include removing URLs, mentions, hashtags, numbers, punctuation and lowercase texts, as well as stop words. The DistilBERT tokenizer tokenizes the processed text, transforming it into attention masks and input IDs. A DistilBERT model with numerous transformer layer, pre- trained using these representations, captures contextual embeddings of the input text. The DistilBERT output embeddings are sent to a DNN classification head, which is made up of Fully Connected (FC) layers that are intended to train discriminative features for classification. Finally, the model provides a reliable and effective method for detecting terrorist threats by predicting whether a tweet is a threat or not. The detailed block diagram of the suggested terrorism threat detection model is visualized in Fig. 1.

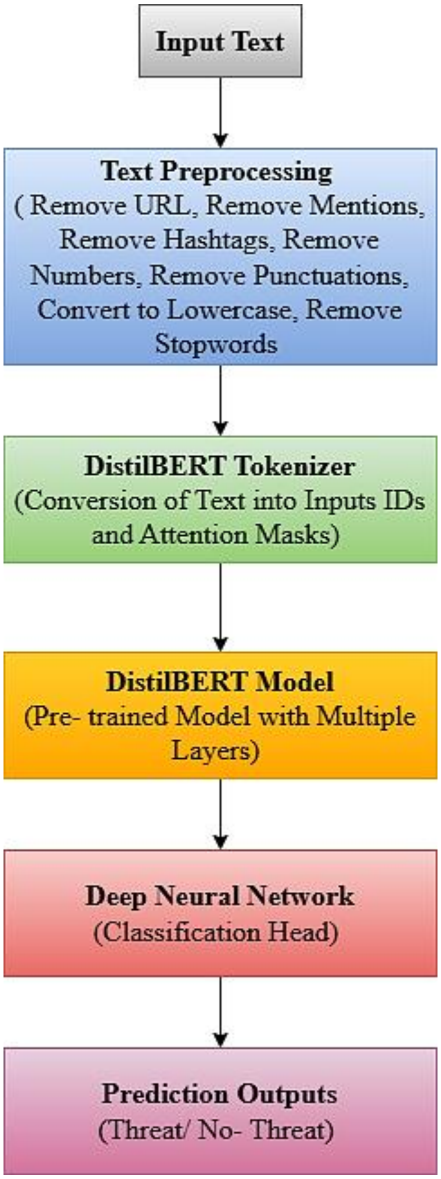
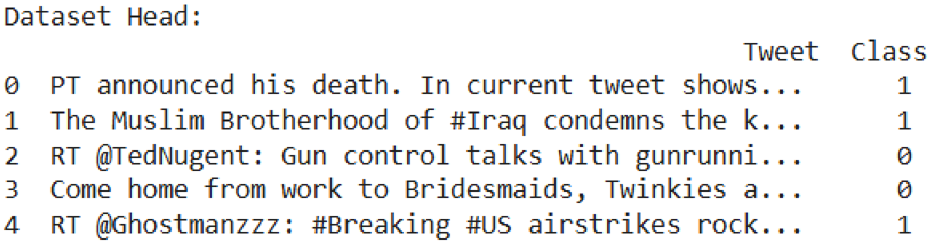

Input Text
Text Preprocessing
(Remove URL, Remove Mentions, Remove Hashtags, Remove Numbers, Remove Punctuations, Convert to Lowercase, Remove Stopwords
DistilBERT Tokenizer
(Conversion of Text into Inputs IDs and Attention Masks)
DistilBERT Model
(Pre- trained Model with Multiple Layers)
Deep Neural Network
(Classification Head)
Prediction Outputs
(Threat/ No- Threat)

Fig. 1. Block diagram of suggested terrorism threat detection system.

A. Dataset Description and Exploratory Data Analysis

The textual data in the dataset was gathered from social media, particularly tweets and was classified as either indicating non- threating or threatening material. The almost equal distribution of these classes is crucial for maintaining the balance of classification tasks. Preprocessing eliminates features like mentions (@username), hashtags (#hashtag) and URLs for a cleaner analysis. The tweets range significantly in length from mere lines to extensive paragraphs. The dataset also includes linguistic variables that aid in distinguishing between threatening and non- threatening tweets, such as average word length and frequency. The dataset is downloaded from GitHub [20]. Fig. 2 shows the dataset sample.



```
Dataset Head:
                                                Tweet  Class
0  PT announced his death. In current tweet shows...      1
1  The Muslim Brotherhood of #Iraq condemns the k...      1
2  RT @TedNugent: Gun control talks with gunrunni...      0
3  Come home from work to Bridesmaids, Twinkies a...      0
4  RT @Ghostmanzzz: #Breaking #US airstrikes rock...      1
```

Fig. 2. Sample dataset.

Each class in the text dataset for terrorist threat detection contains the same number of samples, indicating a balanced class distribution. The number of tweets classified as threatening (Class 1) and non- threatening (Class 0) is 5265 respectively. This balanced distribution promotes accurate predictions for both threatening and non- threatening tweets and allows for equitable learning during training by preventing the model from being biased toward any one class. The class distribution is plotted in Fig. 3.

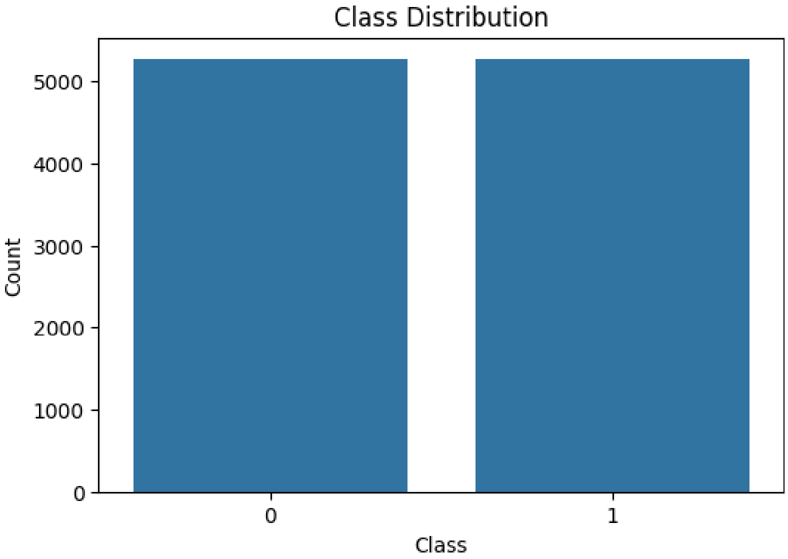
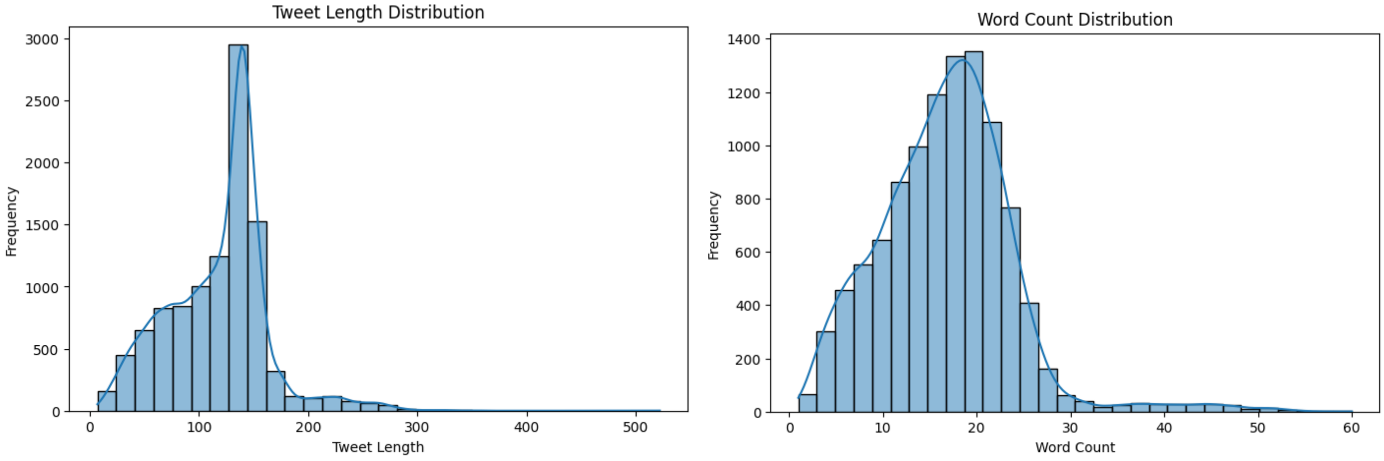


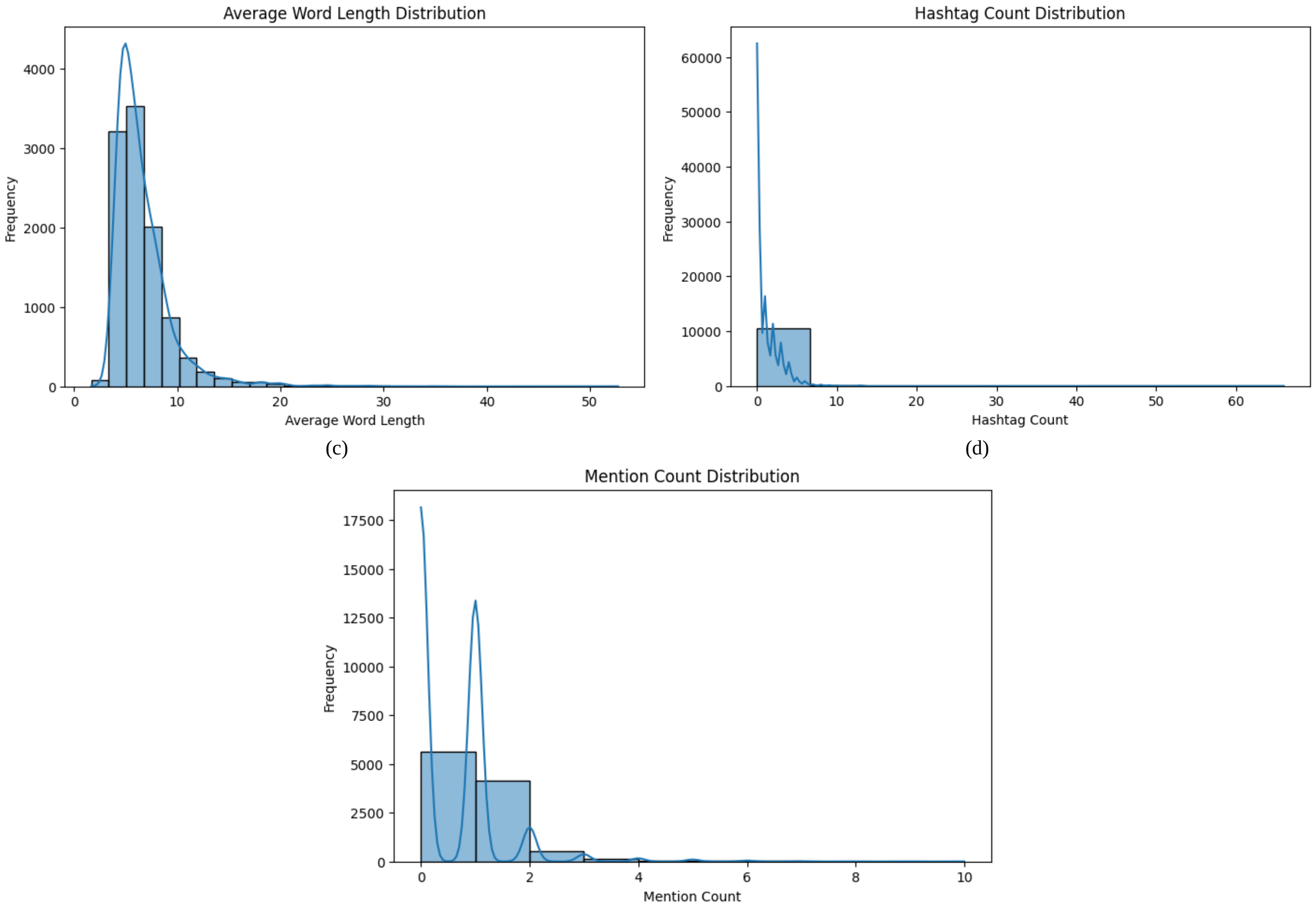
Fig. 3. Class distribution.

The statistical distribution of the character count in each tweet in the dataset is known as the “Tweet Length Distribution” and is displayed in Fig. 4 (a). It provides information about the range of tweet sizes by displaying how lengthy or short the tweets are. The statistical distribution of the word count in each tweet within the dataset is referred to as the “Word Count Distribution” and is displayed in Fig. 4 (b). It provides insights into the typical word count in a tweet, ranging from brief tweets with few words to longer ones with numerous words. The "Average Word Length Distribution" is the distribution of the average number of characters per word in each tweet in the dataset, as shown in Fig. 4(c). It gives information about the complexity of the language used in the tweets, including whether individuals typically use longer, more complicated phrases or shorter, simpler ones. As seen in Fig. 4 (d), the “Hashtag Count Distribution” represents the distribution of hashtags used in each tweet across the datasets. It shows the frequency with which people add hashtags to their tweets, ranging from one to multiple per tweet. The “Mention Count Distribution” represents the distribution of mentions (i.e., tagging another user with “@username”) in each tweet in the dataset, as illustrated in Fig. 4 (e). This distribution shows how frequently users mention or engage with other people in their tweets.



(a)

(b)



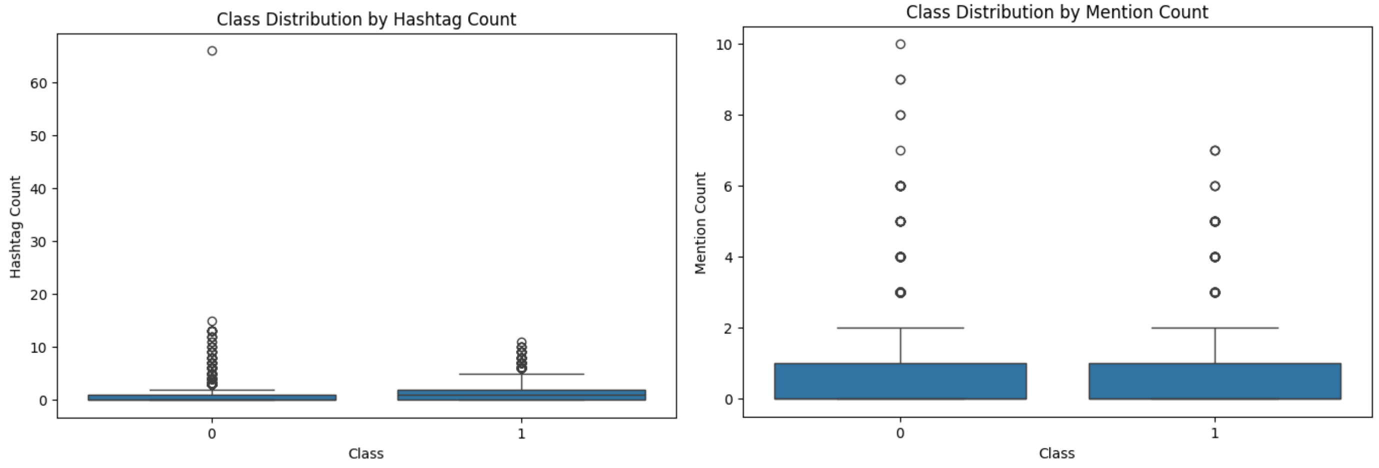
(c)

(d)

(e)

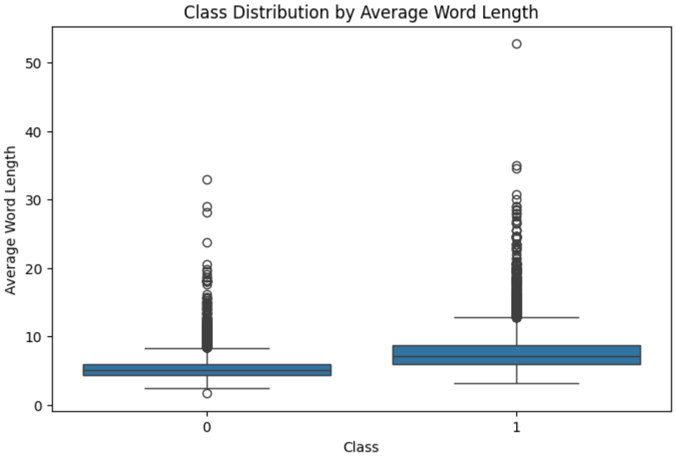
Fig. 4. (a) Tweet length distribution (b) Word count distribution (c) Average word length distribution (d) Hashtag count distribution (e) Mention count distribution.

As illustrated in Fig. 5 (a), "Class Distribution by Hashtag Count" is an analysis that looks at the relationship between a tweet's hashtag count and its classification as "threat" or "non-threat". Box plots commonly compare the quantity of hashtags for each class labels when displaying this kind of distribution.



(a)

(b)



(c)

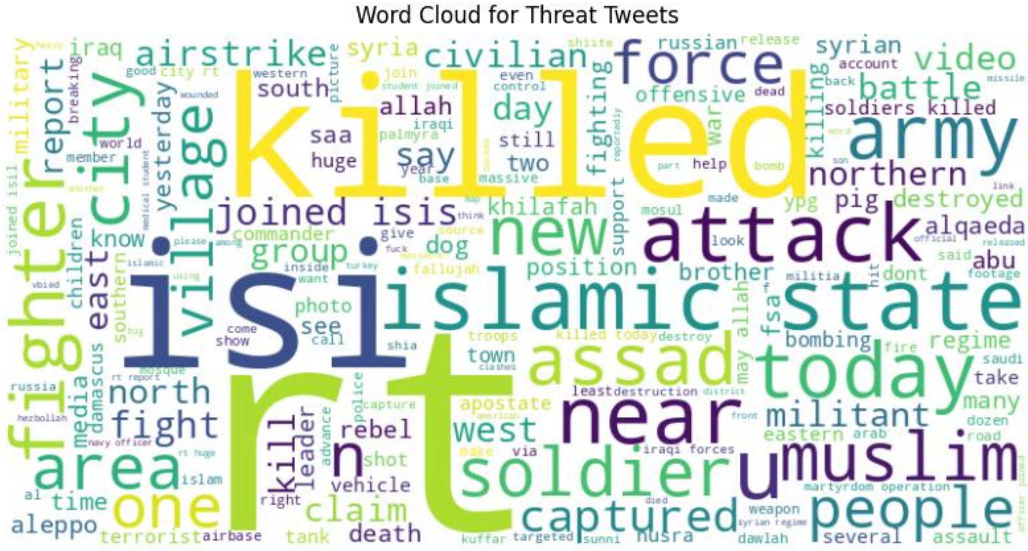
Fig. 5. (a) Class distribution by hashtag count (b) Class distribution by mention count (c) Class distribution by average word length.

It provides information about whether tweets with a threat label typically contain more or fewer hashtags compared to tweets without a threat label. The analysis known as "Class Distribution by Mention Count" examines the frequency of mentions of a user, such as "@username", in tweets classified as "threat" or "non- threat". Box plots, as seen in Fig. 5 (b) are frequently used to illustrate this kind of distribution. They display the median, quartiles, and overall range of mention counts for each class. An association between the quantity of mentions in a tweet and its probability of being categorized as threatening can be ascertained by comparing these distributions. "Class Distribution by Average Word Length" examines the differences in average word length between the two classes of tweets. Fig. 5 (c) display this distribution using box plots. They show the median, quartiles and range of average word lengths for each class. It is feasible to determine whether the complexity of language used in threatening versus non- threatening tweets differs by comparing these distributions.

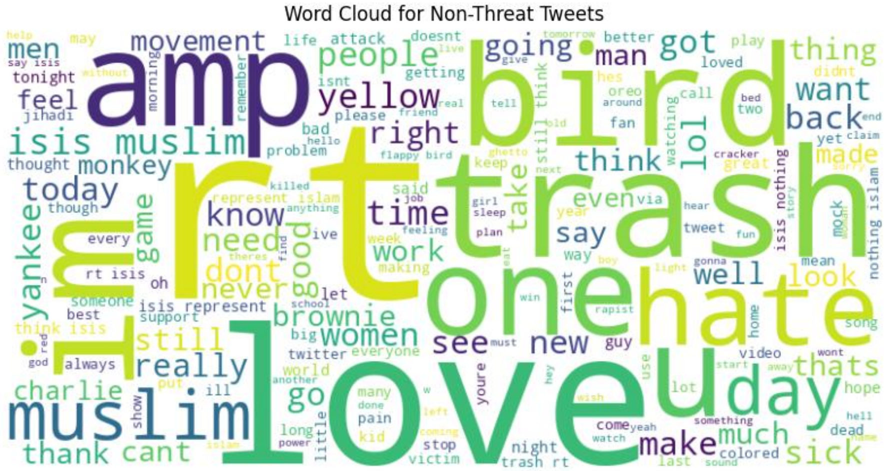
The most common word in a corpus of text is represented visually in a word cloud, where the size of each word indicates its significance or frequency. It is possible to create distinct word clouds for threat and non- threat tweets in order to emphasize the unique terminology utilized in each category. The word cloud for threat and non- threat tweets is displayed in Fig. 6 (a) and Fig. 6 (b).

B. Text Preprocessing

In NLP, text preprocessing is an essential step that cleans and gets raw data ready for analysis. A variety of methods standardize and simplify the text. URLs, hashtags, and mentions are eliminated in order to remove unnecessary components that are not beneficial to the semantic meaning. Eliminating punctuations and numbers ensures that the words stay the main emphasis, which lowers noise. Text conversion to lowercase standardizes data and prevents case- sensitivity induced duplication. Stopwords like "is", "the" and "and" which are often used words that don't significantly advance the analysis, can be removed. When combined, these strategies improve the text's quality and make it more appropriate for NLP tasks.



(a) Threat tweet.



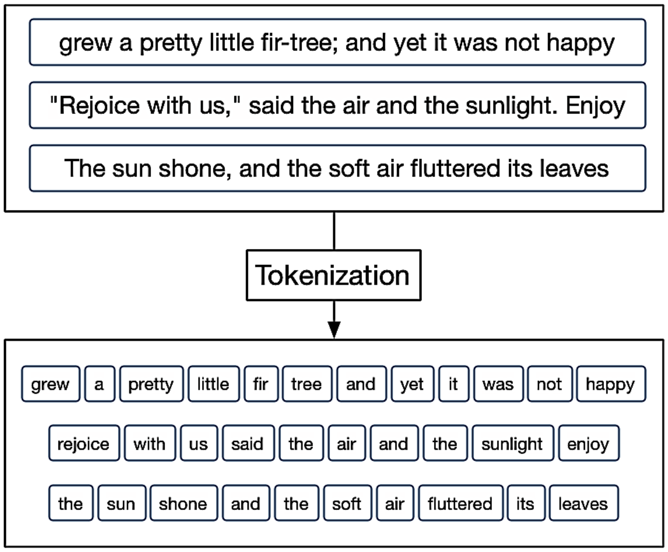
Word Cloud for Non-Threat Tweets

(b) Non-threat tweet.

Fig. 6. Word cloud.

C. Tokenization using DistilBERT Tokenizer

The process of tokenization involves converting unprocessed text data, such as tweets, into a format that the DistilBERT model understands [21]. Since DistilBERT is a transformer- based model, incoming text must be divided into tokens, which are effectively smaller subwords or units. The DistilBERT tokenizer manages this process by dividing words into tokens, assigning a token ID to each token and generating input sequences that sum up the tweet. In addition, the tokenizer uses truncation to eliminate extra tokens that exceed the allowable length and padding to ensure that every sequence has a constant length. This phase is crucial for preparing the data for feeding into the model, which will enable the transformer to retrieve the text effectively and comprehend the context of each tweet. An example of tokenization operation is visualized in Fig. 7.



grew a pretty little fir-tree; and yet it was not happy
"Rejoice with us," said the air and the sunlight. Enjoy
The sun shone, and the soft air fluttered its leaves
Tokenization
grew a pretty little fir tree and yet it was not happy
rejoice with us said the air and the sunlight enjoy
the sun shone and the soft air fluttered its leaves

Fig. 7. An illustration of tokenization.

D. Proposed Terrorism Threat Detection Model Based on DistilBERT and DNN

DistilBERT is a faster and more compact version of Bidirectional Encoder Representations from Transformer (BERT) designed to be more efficient on devices with limited processing power. Using a method called knowledge distillation, the BERT model is shrunk by teaching a smaller model to behave similarly to a bigger one [22]. DistilBERT is more efficient due to its reduced size and simplified training procedure. It uses a simplified transformer architecture, notably removing the token type embeddings from the input and the pooling layer from the output. Fig. 8 visualizes the DistilBERT architecture.

The primary goal of DistilBERT is to preserve the majority of BERT's performance while lowering its memory and computational costs. DistilBERT accomplishes this through:

- DistilBERT employs a smaller architecture than BERT with fewer layers and hidden components. In particular, it features 6 layers rather than 12 and 2048 hidden units rather than 7680 for the base model.
- Training through Knowledge Distillation: DistilBERT is trained to replicate the behaviors of the larger BERT model via a strategy called knowledge distillation. To train DistilBERT, the outputs of BERT are employed as soft targets.

The general structure of DistilBERT is similar to that of BERT, despite having fewer layers and hidden components. A feed-forward neural network with a self-attention mechanism is included within each layer of DistilBERT's multi-layer bidirectional transformer encoder. The self-attention mechanism enables the model to focus on different components of the input sequence and identify connections among words. DistilBERT first embeds a sequence of tokens in a high dimensional vector space as its input [23]. The transformer encoder creates contextualized representations for every token by further processing these embeddings. A classification or regression layer processes the contextualized representations to predict the outcome.

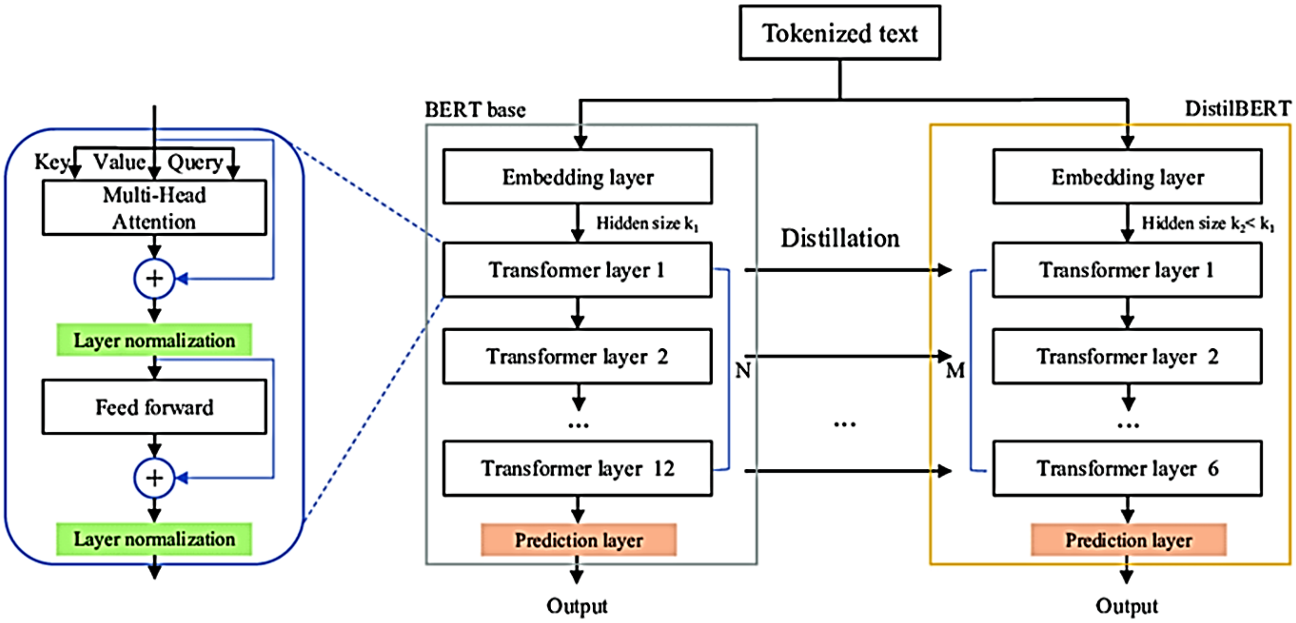


Fig. 8. DistilBERT architecture.

The tokenized input text is fed into both DistilBERT and BERT. The tokenization process divides the input text into smaller tokens, transforms them into IDs and ensures that each input sequence has a constant length. This uniform input is necessary for both BERT and DistilBERT to process the data effectively. The transformer layer and the embedding layer are two of the layers that compose the BERT base architecture. The embedding layer provides numerical representations of words or subwords by transforming input tokens into dense vectors with specified dimensions. BERT comprises 12 transformer layers, including feed- forward, multi- head attention and layer normalization [24]. Each transformer layer can focus on different parts of the sequence at the same time due to a multi- head attention mechanism that accepts three inputs: key, query and value. Layer normalization normalizes the output of the multi- head attention to maintain stability throughout the training process. The feed- forward layer extracts higher level characteristics from the output of the attention mechanism by applying a FC layer. The feed-forward procedure is followed by another normalization step. The structure of DistilBERT is similar to that of BERT, although there are a few significant differences. Similar to BERT, an embedding layer transforms tokens into dense vectors. DistilBERT employs six transformer layers instead of twelve, reducing the number of levels. Its distillation training approach allows it to preserve most of the BERTs context-handling capabilities despite having fewer layers. A major factor in DistilBERT’s lightweight design is that its hidden size (k_2) is usually smaller than BERT’ s (k_1).

Eq. (1) calculates the scaled dot-product attention, the fundamental component of the attention process.

$$Attention\ (Q, K, V) = softmax\left(\frac{QK^T}{\sqrt{d_k}}\right) V \qquad (1)$$

The input generates three separate representations: Q, K, and V. The three variables possess dimensions of $n \times d$, with d representing the dimensionality of each token and n indicating the sequence length (number of tokens). The model produces attention scores by evaluating each query against every key. The attention scores are evaluated employing the dot product of the Q and K metrics and the square root of the dimensionality d_k. These scores highlight the level of attention that each component of the input should receive when producing the output. The attention mechanism then uses the attention scores to determine the weighted sum of the values. The system may retrieve all types of relationships in the data because multiple heads produce queries, keys, and values concurrently. The multi- head attention mechanism then creates the final output by combining the individual outputs of each attention head. A prediction layer is applied to the output of the transformer layer. This layer is utilized for a variety of NLP tasks. The output layer of DistilBERT is essentially the same as that of BERT, but because of its simplified architecture and fewer transformer layer, it utilizes fewer parameters.

Specifically for binary classification tasks, the output layer of the DistilBERT is the DNN with the sigmoid layer, which generates a final prediction [25]. Fig. 9 illustrates that a dense layer, also referred to as a FC layer, connects every input node to every output node in a neural network. This dense layer in the DistilBERT context receives the output from the global average pooling operation or earlier transformation layers. The transformer layers extract high- level characteristics and send them to the dense layer, which applies an activation function after a linear transformation.

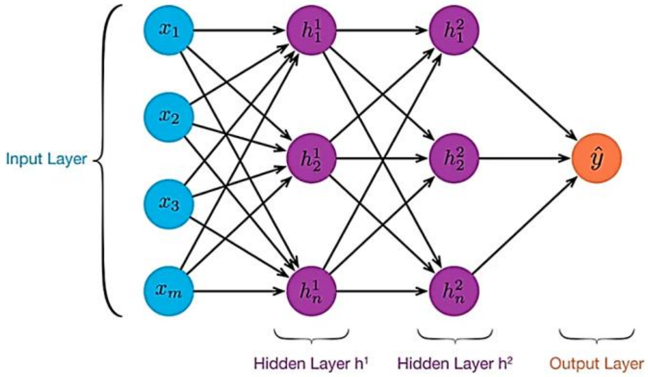


Fig. 9. Deep neural network.

The output layer uses the sigmoid activation function to classify data into binary categories. The sigmoid function is appropriate for situations where the objective is to categorize the input into one of two groups since it converts the result to a number between 0 and 1. In the classification of threats versus non- threats, a sigmoid output around 1 would suggest a high likelihood that the input tweet is a threat, whereas a value near 0 would suggest a non- threat. The sigmoid function is mathematically expressed as in Eq. (2).

$$Sigmoid\ (x) = \frac{1}{1+e^{-x}} \tag{2}$$

The embedding layer and transformer levels receive tokenized text data (viz multi- head attention and feed- forward sublayers). Next, a global average pooling layer usually shrinks the sequence of vectors into a single vector of fixed length that represents the whole sequence. Next, a dense layer employing a sigmoid activation function receives this vector. The dense layer then produces a single probability score that indicates the probability that the input fits to a specific category. The model learns to modify its parameters during training so that it generates output values that are nearly equal to 0 for non-threat class and 1 for threat class. The sigmoid output is subjected to a decision threshold (usually 0.5) during inference in order to classify the input. The model classifies the output as a threat if it exceeds 0.5 and as a non- threat otherwise.

In order to ensure uniformity in the input sequence, the suggested model starts with a DistilBERT tokenizer, which tokenizes and pads input tweets to a constant length of 100 tokens. A custom DistilBERT layer receives the tokenized outputs, which comprise input_ids and attention_mask. This layer captures the semantic complexities of the tweets and uses the pre- trained DistilBERT model to extract contextual embeddings. To reflect the deep contextual relationships in the text, the DistilBERT model creates a series of embeddings for every token in the input. The resultant embeddings are then subjected to a global average pooling layer. This layer aggregates the token- level embeddings into a fixed- size vector, simplifying the representation while keeping the essential elements. A dropout layer with 20% rate is applied to the pooled output to further reduce overfitting by randomly turning off some neurons during training. A deep layer with 128 neurons and ReLU activation enables the network to subsequently discover complex patterns in the data. Regularization is boosted by adding a second dropout layer at the same rate. The output layer, comprising one neuron with a sigmoid activation function, finishes the proposed system by providing a binary classification for deciding whether the tweet contains information relevant to terrorism. Fig. 10 shows the proposed model architecture.

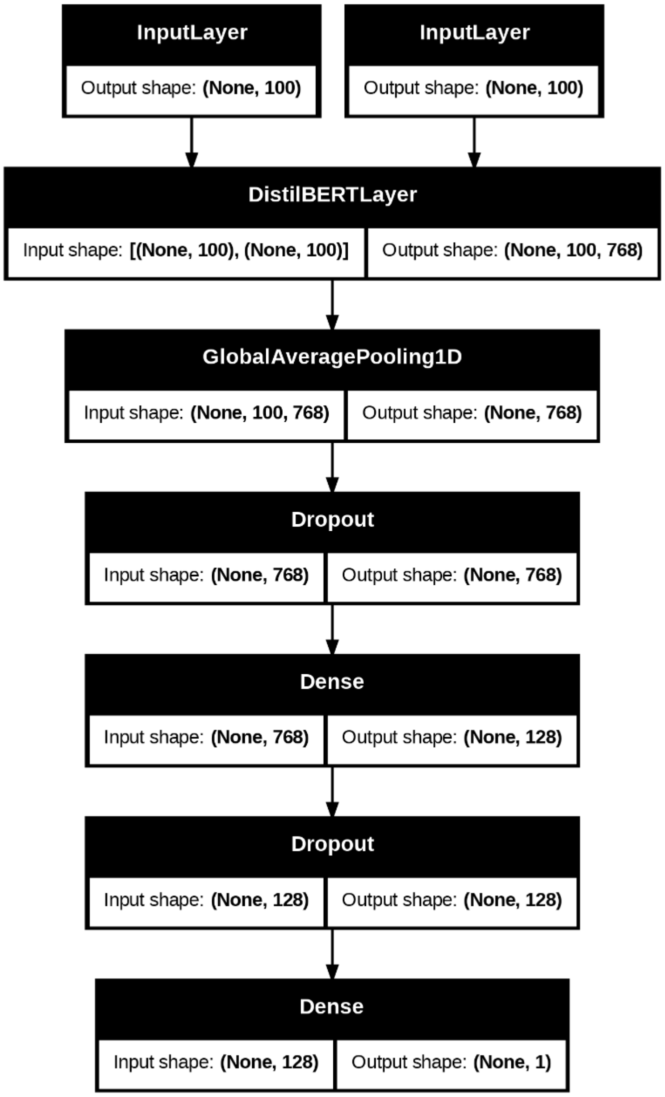

InputLayer
Output shape: (None, 100)
InputLayer
Output shape: (None, 100)
DistilBERTLayer
Input shape: [(None, 100), (None, 100)]
Output shape: (None, 100, 768)
GlobalAveragePooling1D
Input shape: (None, 100, 768)
Output shape: (None, 768)
Dropout
Input shape: (None, 768)
Output shape: (None, 768)
Dense
Input shape: (None, 768)
Output shape: (None, 128)
Dropout
Input shape: (None, 128)
Output shape: (None, 128)
Dense
Input shape: (None, 128)
Output shape: (None, 1)

Fig. 10. Proposed terrorism attack detection model architecture.

The detailed algorithm of the proposed terrorism attack detection model is explained below.

Algorithm: A Novel NLP Model Based on DistilBERT and DNN for the Detection of Terrorism Threat from Tweets

Input: *Text data related to terrorism attacks, Pre- trained DistilBERT Model*
Output: *Efficient Terrorism Attack Detection Model*

```
Begin:
    ❖ Collect Dataset: $\mathcal{D}$= {(X, y), where X is a tweet and y$\in \{0,1\}$, (0 for non- threat tweet and 1 for threat tweet)
    ❖ Text Preprocessing:
          ➢ URLs Removal
          ➢ Mentions Removal
          ➢ Hashtags Removal
          ➢ Numbers Removal
          ➢ Punctuations Removal
          ➢ Convert to lowercase
          ➢ Stopwords Removal
    ❖ Dataset Splitting:
          ➢ X = df['Cleaned_Tweet']
          ➢ y = df['Class']
          ➢ X_train, X_test, y_train, y_test = train_test_split (X, y, test_size=0.2, random_state=42)
    ❖ Tokenization using DistilBERT Tokenizer:
          ➢ Tokenized_Input=tokenizer (X, padding='max_length', truncation=True, max_length=100)
          ➢ Tokenizer returns:
                o input_ids $\in Z^{n\times 100}$: Tokenized IDs for each word.
                o attention_mask $\in \{0,1\}^{n\times 100}$: Masks indicating relevant tokens.
    ❖ Proposed DistilBERT- DNN Model Creation:
          ➢ distilbert_model = TFDistilBertModel.from_pretrained ('distilbert-base uncased')
          ➢ input_ids = Input (shape= (100,), dtype=tf.int32, name='input_ids')
          ➢ attention_mask = Input (shape= (100,), dtype=tf.int32, name='attention_mask')
          ➢ bert_output = DistilBERTLayer () ([input_ids, attention_mask])
          ➢ x = GlobalAveragePooling1D () (bert_output)
          ➢ x = Dropout (0.2) (x)
          ➢ x = Dense (128, activation='relu') (x)
          ➢ x = Dropout (0.2) (x)
          ➢ output = Dense (1, activation='sigmoid') (x)
          ➢ M = Model (inputs= [input_ids, attention_mask], outputs=output)
    ❖ Model Compilation and Training:
          ➢ M.compile(optimizer=Adam(learning_rate=0.0001), loss='binary_crossentropy', metrics=['accuracy'])
          ➢ M.fit([X_train_tokens['input_ids'], X_train_tokens['attention_mask']], y_train, epochs=100, batch_size=32,
            validation_split=0.2)
    ❖ Model Evaluation:
          ➢ y_pred_prob=M.predict([X_test_tokens['input_ids'], X_test_tokens['attention_mask']])
          ➢ y_pred = (y_pred_prob > 0.5). astype(int). flatten ()
          ➢ accuracy = accuracy_score (y_test, y_pred)
          ➢ report = classification_report (y_test, y_pred)
          ➢ conf_matrix = confusion_matrix (y_test, y_pred)
          ➢ roc_auc = roc_auc_score (y_test, y_pred_prob)
    ❖ Save the Model:
    End
```

E. Simulation Setup

The proposed terrorism attack detection model has been designed and implemented on Google Collaboratory platform with Python language. Gradient estimation stability and computing performance were balanced by using a batch size of 32. The Adam optimizer was selected to speed up convergence because of its popularity for adaptable learning. The model underwent training for 100 epochs, providing enough iterations to discover intricate patterns in the data. The learning rate was chosen to be 0.001 to allow a fine-grained change to model weights during training, limiting overshooting and ensuring stable convergence. The loss function, binary cross entropy was appropriate for binary classification tasks. Table I offers various hyperparameters utilized by the model.

TABLE I. HYPERPARAMETERS

Hyperparameters	Values
Batch Size	32
Optimizer	Adam
Number of Epochs	100
Loss Function	Binary Crossentropy
Learning Rate	0.0001

A batch size of 32 was selected in order to balance model generalization with computational efficiency. This implies that the model processes 32 datasets at a time before updating the weights. The model successfully improved its parameters repeatedly throughout several dataset iterations during the 100 training epochs. This binary classification problem is well-fitted by the binary cross-entropy loss function, which counts the discrepancy between expected probability and actual binary class labels. The Adam optimizer was utilized to optimize the weights, combining the advantages of adaptive learning rates and momentum for effective training. The step size during weight updates was controlled by setting the learning rate to 0.0001, which allowed for consistent convergence.

IV. Experimental Results

The performance of a model during training and validation can be assessed using the accuracy plot and loss plot. The accuracy plot provides insight into the effectiveness of the model in generalizing to unknown data by illustrating changes in its capacity to accurately categorize samples over different epochs. A steadily increasing accuracy curve demonstrates effective learning. The loss plot on the other hand, displays the model’s error over epochs and a decreasing loss indicates a reduction in the difference between predicted and assigned labels. When combined, these plots offer a thorough understanding of how the model learns and provide optimization techniques for enhanced efficiency. Fig. 11(a) and Fig. 11(b) visualizes the accuracy plot and loss plot of the model. The model performs much better in the first epochs with training accuracy increasing from 77% in epoch 1 to over 93% by epoch 50 and training loss decreasing in parallel. After epoch 30, validation accuracy steadily increases between 91 and 92%, while validation loss levels off at 0.22, suggesting efficient learning and little overfitting. The model may have achieved an ideal balance if both training and validation measures exhibit convergence.

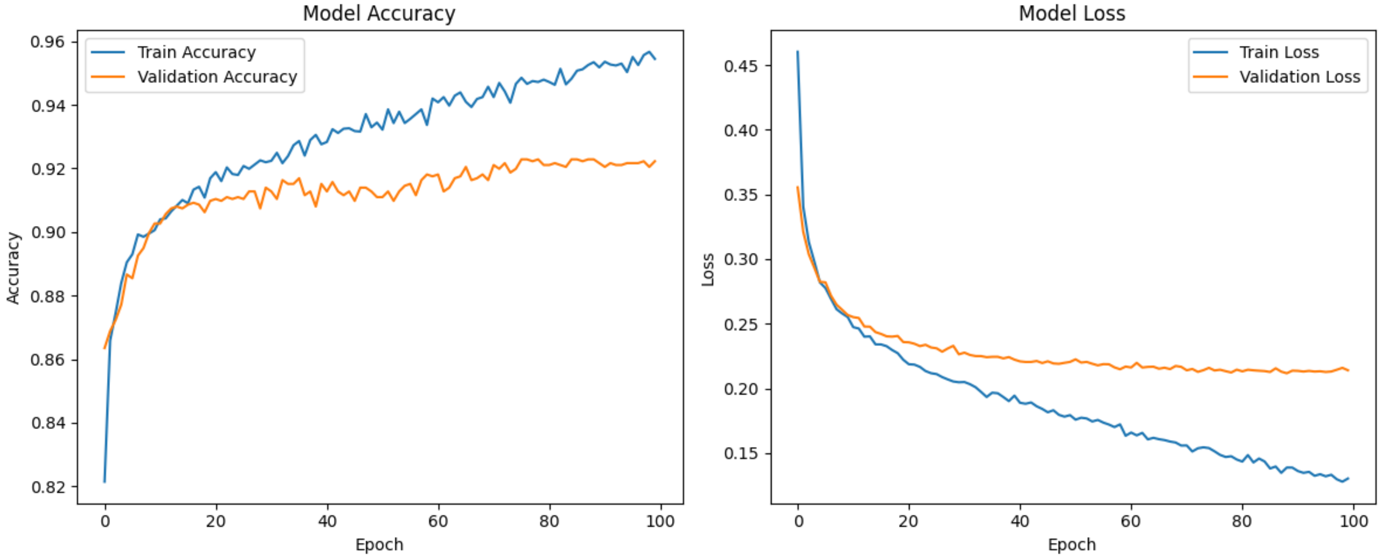


Fig. 11. (a) Accuracy plot (b) Loss plot of proposed terrorism attack detection model.

Accuracy is a crucial performance static in the context of terrorism attack detection from twitter data. It quantifies the percentage of correctly recognized occurrences among all occurrences. It is computed using Eq. (3).

$$Accuracy = \frac{TP+TN}{TP+TN+FP+FN} \quad (3)$$

The term true positive (TP) refers to the quantity of threat-related tweets correctly identified. True negativity (TN) refers to the quantity of accurately identified, non-threatening tweets. False positives (FP) refer to the number of non-threat-related tweets that receive incorrect classification as threats. The number of tweets about threat that are inaccurately classified as non- threat is known as false negative (FN). The proposed model for terrorism attack detection accomplished a remarkable performance with an accuracy of 93%.

A classification report is a thorough performance assessment tool that offers important indicators for evaluating the effectiveness of a detection model. It comprises support for every class, F1- score, precision and recall. The F1 score offers a balanced measure of precision and recall, with recall demonstrating the model's ability to identify all pertinent events and precision quantifying the accuracy of positive predictions. The number of instances of each class in the dataset is known as support. Table II presents the classification report for the suggested model.

TABLE II. Classification Report of Suggested Model

Class	Precision	Recall	F1- Score	Support
0 (Non- Threat)	0.93	0.93	0.93	1074
1 (Threat)	0.92	0.92	0.92	1032
Accuracy			0.93	2106
Macro Average	0.93	0.93	0.93	2106
Weighted Average	0.93	0.93	0.93	2106

The classification report shows that the terrorism attack detection model performs well in both the threat and non-threat classes. For non- threat class, the model’s precision, recall and F1- score of 0.93 show that it is quite accurate in detecting non- threats without producing a lot of false positives. The precision, recall and F1- score for the threat class are 0.92, but they still show a high level of threat detection accuracy. The classification between the two classes is well balanced with an overall model accuracy of 93%. The macro average, which calculates the measures without taking into account class imbalance shows consistent performance across classes, getting a strong 0.93 for precision, recall and F1- score. Additionally, the weighted average supports the efficacy of the model by confirming constant performance while accounting for the number of cases in each class.

An essential tool for assessing the effectiveness of detection models is the confusion matrix, which offers a thorough analysis of the prediction model. The confusion matrix aids in locating class imbalances and directs model enhancements, such as threshold adjustments, regularization or class distribution balancing. Fig. 12 visualizes the confusion matrix of the suggested model.

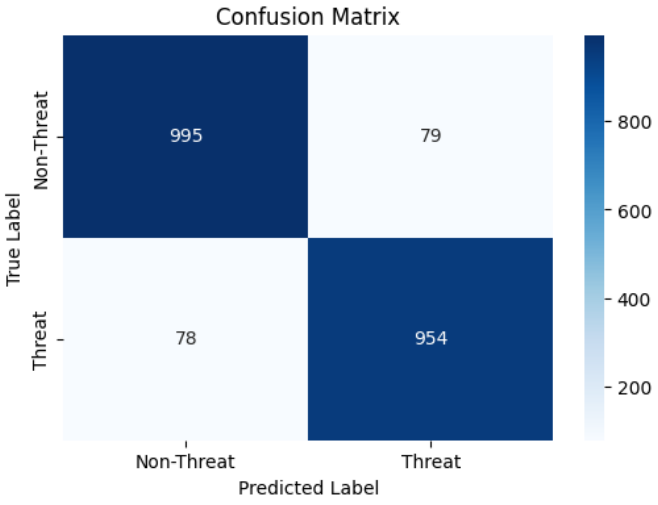


Fig. 12. Confusion matrix of suggested system.

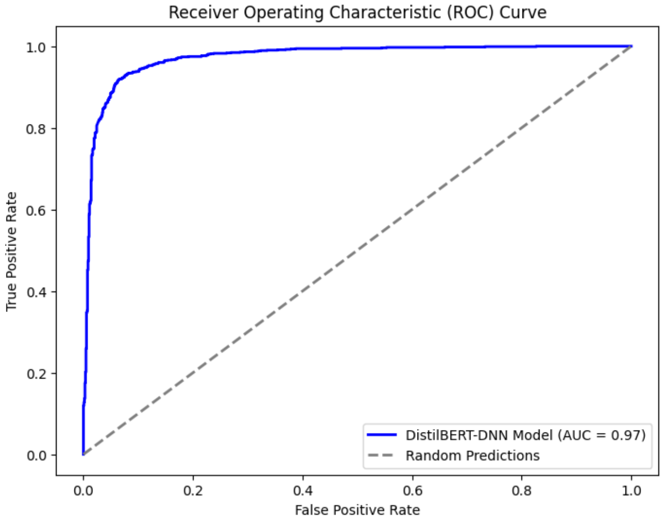
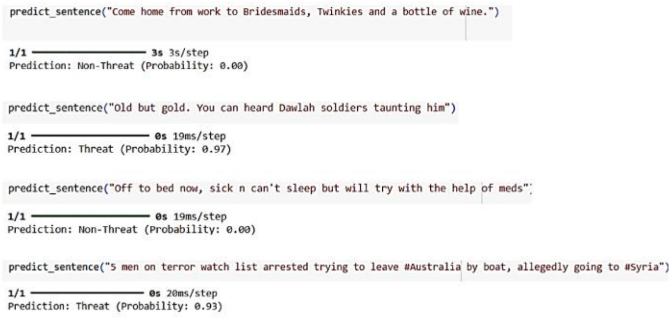


Fig. 13. ROC Curve of proposed model.

The confusion matrix illustrates that the model has a high degree of accuracy in both classifications, properly classifying 995 non- threat cases as non- threats and 954 threat cases as threats. However, there are 78 false negatives where real threats were incorrectly classified as non- threats and 79 false positives where non- threats were incorrectly classified as threats. The effectiveness of detection models is evaluated using a performance evaluation tool called the Receiver Operating Characteristic (ROC) curve, which is shown in Fig. 13. The rates of false positives and true positives are shown at different thresholds.

The ROC curve illustrates the balance between minimizing false positives and identifying true positives. The efficacy of a model is frequently evaluated by the area under the ROC curve, known as AUC. A greater AUC denotes better model performance. Some of the prediction outputs are displayed in Fig. 14.



```
predict_sentence("Come home from work to Bridesmaids, Twinkies and a bottle of wine.")

1/1 ━━━━━━━━━━━━━━━━━━━━ 3s 3s/step
Prediction: Non-Threat (Probability: 0.00)

predict_sentence("Old but gold. You can heard Dawlah soldiers taunting him")

1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 19ms/step
Prediction: Threat (Probability: 0.97)

predict_sentence("Off to bed now, sick n can't sleep but will try with the help of meds"

1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 19ms/step
Prediction: Non-Threat (Probability: 0.00)

predict_sentence("5 men on terror watch list arrested trying to leave #Australia by boat, allegedly going to #Syria")

1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 20ms/step
Prediction: Threat (Probability: 0.93)
```

Fig. 14. Prediction outputs.

V. Discussion

Table III provides the performance comparison of the suggested terrorist threat detection system with existing approaches. The suggested DistilBERT- DNN model substantially outperformed a number of current approaches in the field of terrorist threat identification obtaining an incredible 93% accuracy rate. The suggested model shows the benefits of advanced NLP techniques with a significant improvement over the NB classifier, maximum entropy classifier and SVM, which all reached an accuracy of 73.33%. This highlights the effectiveness of merging lightweight transformer-based embeddings with a DNN.

The DistilBERT- DNN model also outperformed the SVM based technique, which obtained 84% and ensemble methods, including RF, KNN, DT, SVM and Adaboost, which achieved 90%. Additionally, the model outperformed Word2Vec in gradient boosting with Word2Vec (87.9%) and K- means clustering (85.8%). Additionally, it outperforms conventional and neural based methods like ANN at 75 % and BERT and its variants at 72%. This demonstrates how well the DistilBERT-DNN model captures the semantic context of tweets for accurate threat identification, making it an innovative approach in this field. Fig. 15 shows the performance comparison of the suggested model with state-of-art methodologies.

TABLE III. PERFORMANCE COMPARISON

Authors	Proposed Methodology	Detection Accuracy
Mayur Gaikwad et al. [11]	BERT and its Variants	72%
Kuljeet Kaur [26]	NB Classifier, Maximum Entropy Classifier and SVM	73.33%
Ghada M.A. Soliman and Tarek H.M. Abou-El-Enien [17]	Artificial Neural Network	75%
Waqas Sharif et al. [27]	SVM	84 %
Lu An et al. [9]	Word2Vec with the K-means clustering	85.8%
Shynar Mussiraliyeva et al. [16]	Gradient boosting with word2vec	89%
Sagar Shinde et al. [8]	RF, KNN, DT, SVM, Adaboost	90%
Proposed DistilBERT- DNN Model		**93%**

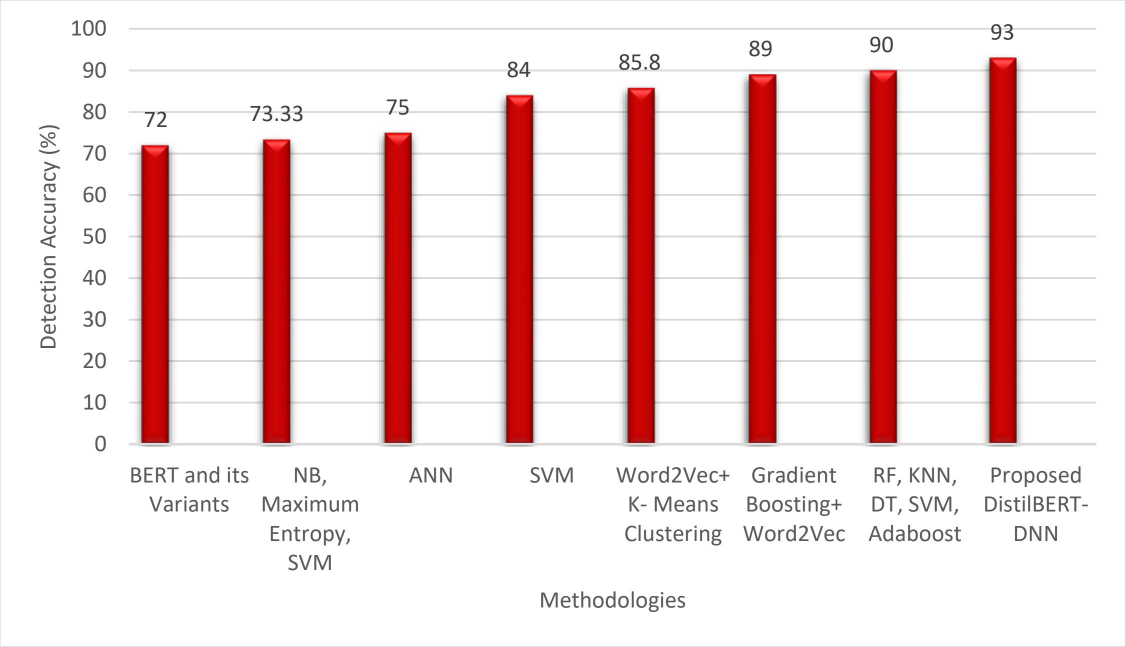


Fig. 15. Graphical illustration of performance comparison with existing methodologies.

VI. CONCLUSION

Global terrorist activity has increased significantly in recent years, causing major risks to national security and public safety. Conventional methods of detecting and preventing terrorist attacks often rely on human contact and existing surveillance systems, which limits their monitoring capabilities and response measures. The intricacy, speed and extent of modern terrorism renders these conventional techniques insufficient, as it often involves decentralized operations, digital traces and secret conversations. The development of increasingly advanced automated systems that can identify possible terrorist actions with greater accuracy and speed is required due to the technological advancements, particularly the growth of digital media and communication platforms. This paper proposed an effective terrorism threat detection model using DistilBERT with Deep Neural Network from twitter data. The model ensures accurate and efficient representation of text data by utilizing DistilBERT's transformer-based architecture, which is lightweight and powerful to extract rich semantic information from tweets. A Dense Neural Network, which is excellent at spotting intricate patterns and relationship in the data, processes these embeddings further to accurately classify tweets as either threats or non- threat. The DistilBERT-DNN model has demonstrated remarkable efficacy in detecting terrorist threats from twitter data with a 93% accuracy. The methodology surpasses existing neural network-based approaches and conventional machine learning techniques by effectively combining the advantages of deep learning and natural language processing. This demonstrates how advanced NLP techniques can improve the effectiveness and reliability of automated systems for detecting terrorist threats. Future research can focus on integrating multimodal data sources, such as image and video analysis, to improve threat detection beyond textual data. Additionally, incorporating real-time streaming capabilities can enable proactive identification and mitigation of threats as they emerge. Moreover, leveraging graph-based neural networks can help analyze network structures and relationships between suspicious accounts to identify coordinated terrorist activities. These future enhancements will contribute to the development of a more comprehensive, accurate, and real-time terrorism threat detection system, strengthening national security and public safety.

REFERENCES

[1] Schmid, A. P. (2011). The definition of terrorism. In The Routledge handbook of terrorism research (pp. 39-157). Routledge.

[2] Nasution, A. R. (2017, December). Acts of terrorism as a crime against humanity in the aspect of law and human rights. In 2nd International Conference on Social and Political Development (ICOSOP 2017) (pp. 346-353). Atlantis Press.

[3] Wilkinson, P. (2006). Terrorism versus democracy: The liberal state response. Routledge.

[4] Huang, R. (2008). Counterterrorism and the Rule of Law. Civil War and the Rule of Law: Security, Development, Human Rights, edited by A. Hurwitz and R. Huang, 261-284.

[5] Turk, A. T. (2004). Sociology of terrorism. Annu. Rev. Sociol., 30(1), 271-286.

[6] Kelleher, J. D. (2019). Deep learning. MIT press.

[7] Chowdhary, K., & Chowdhary, K. R. (2020). Natural language processing. Fundamentals of artificial intelligence, 603-649.

[8] Shinde, S., Khoje, S., Raj, A., Wadhwa, L., & Shaikha, A. S. (2024). Artificial intelligence approach for terror attacks prediction through machine learning. Multidisciplinary Science Journal, 6(1), 2024011-2024011.

[9] An, L., Han, Y., Yi, X., Li, G., & Yu, C. (2023). Prediction and evolution of the influence of microblog entries in the context of terrorist events. Social Science Computer Review, 41(1), 64-82.

[10] Jyothilinga, S. (2023). Analysis and Prediction of Terrorist Attacks using Supervised Machine Learning and Deep Learning Techniques (Doctoral dissertation, Dublin, National College of Ireland).

[11] Gaikwad, M., Ahirrao, S., Kotecha, K., & Abraham, A. (2022). Multi-ideology multi-class extremism classification using deep learning techniques. IEEE Access, 10, 104829-104843.

[12] Feng, Y., Wang, D., Yin, Y., Li, Z., & Hu, Z. (2020). An XGBoost-based casualty prediction method for terrorist attacks. Complex & Intelligent Systems, 6, 721-740.

[13] Zhao, R., Xie, X., Zhang, X., Jin, M., & Hao, M. (2021). Spatial distribution assessment of terrorist attack types based on I-MLKNN model. ISPRS International Journal of Geo-Information, 10(8), 547.

[14] Luo, L., & Qi, C. (2021). An analysis of the crucial indicators impacting the risk of terrorist attacks: A predictive perspective. Safety science, 144, 105442.

[15] Theodosiadou, O., Pantelidou, K., Bastas, N., Chatzakou, D., Tsikrika, T., Vrochidis, S., & Kompatsiaris, I. (2021). Change point detection in terrorism-related online content using deep learning derived indicators. Information, 12(7), 274.

[16] Mussiraliyeva, S., Bolatbek, M., Omarov, B., & Bagitova, K. (2020). Detection of extremist ideation on social media using machine learning techniques. In Computational Collective Intelligence: 12th International Conference, ICCCI 2020, Da Nang, Vietnam, November 30–December 3, 2020, Proceedings 12 (pp. 743-752). Springer International Publishing.

[17] Soliman, G. M., & Abou-El-Enien, T. H. (2019). Terrorism Prediction Using Artificial Neural Network. Rev. d'Intelligence Artif., 33(2), 81-87.

[18] Koutidis, G., Loumponias, K., Tsikrika, T., Vrochidis, S., & Kompatsiaris, I. (2024). Towards AI-driven Prediction of Terrorism Risk based on the Analysis of Localised Web News. IEEE Access.

[19] Chaabene, N. E. H. B., Bouzeghoub, A., Guetari, R., & Ghezala, H. H. B. (2021, October). Applying machine learning models for detecting and predicting militant terrorists behaviour in Twitter. In 2021 IEEE international conference on systems, man, and cybernetics (SMC) (pp. 309-314). IEEE.

[20] https://github.com/juanbetancur96/TerrorismDetectionTweeter/blob/main/Dataset.xlsx

[21] A. Mullen, L., Benoit, K., Keyes, O., Selivanov, D., & Arnold, J. (2018). Fast, consistent tokenization of natural language text. Journal of Open-Source Software, 3(23), 655.

[22] Geoffrey Hinton, Oriol Vinyals, and Jeff Dean. 2015. Distilling the knowledge in a neural network. ArXiv preprint arXiv:1503.02531.

[23] Sanh, V. (2019). DistilBERT, a distilled version of BERT: smaller, faster, cheaper and lighter. arXiv preprint arXiv:1910.01108.

[24] Ghojogh, B., & Ghodsi, A. (2020). Attention mechanism, transformers, BERT, and GPT: tutorial and survey.

[25] Kriegeskorte, N., & Golan, T. (2019). Neural network models and deep learning. Current Biology, 29(7), R231-R236.

[26] Kaur, K. (2016, March). Development of a framework for analyzing terrorism actions via Twitter lists. In 2016 International conference on computational techniques in information and communication technologies (ICCTICT) (pp. 19-24). IEEE.

[27] Sharif, W., Mumtaz, S., Shafiq, Z., Riaz, O., Ali, T., Husnain, M., & Choi, G. S. (2019). An empirical approach for extreme behavior identification through tweets using machine learning. Applied Sciences, 9(18), 3723.