Computer Modeling in
Engineering & Sciences

Tech Science Press

DOI: 10.32604/cmes.2023.029911

ARTICLE

# Terrorism Attack Classification Using Machine Learning: The Effectiveness of Using Textual Features Extracted from GTD Dataset

**Mohammed Abdalsalam**[1,*], **Chunlin Li**[1], **Abdelghani Dahou**[2] **and Natalia Kryvinska**[3]

[1]School of Computer Science and Technology, Wuhan University of Technology, Wuhan, 430070, China

[2]LDDI Laboratory, Faculty of Science and Technology, University of Ahmed DRAIA, Adrar, 01000, Algeria

[3]Information Systems Department, Faculty of Management, Comenius University, Bratislava, 82005, Slovakia

*Corresponding Author: Mohammed Abdalsalam. Email: 79834@whut.edu.cn

Received: 14 March 2023 Accepted: 07 July 2023 Published: 17 November 2023

## ABSTRACT

One of the biggest dangers to society today is terrorism, where attacks have become one of the most significant risks to international peace and national security. Big data, information analysis, and artificial intelligence (AI) have become the basis for making strategic decisions in many sensitive areas, such as fraud detection, risk management, medical diagnosis, and counter-terrorism. However, there is still a need to assess how terrorist attacks are related, initiated, and detected. For this purpose, we propose a novel framework for classifying and predicting terrorist attacks. The proposed framework posits that neglected text attributes included in the Global Terrorism Database (GTD) can influence the accuracy of the model's classification of terrorist attacks, where each part of the data can provide vital information to enrich the ability of classifier learning. Each data point in a multiclass taxonomy has one or more tags attached to it, referred as "related tags." We applied machine learning classifiers to classify terrorist attack incidents obtained from the GTD. A transformer-based technique called DistilBERT extracts and learns contextual features from text attributes to acquire more information from text data. The extracted contextual features are combined with the "key features" of the dataset and used to perform the final classification. The study explored different experimental setups with various classifiers to evaluate the model's performance. The experimental results show that the proposed framework outperforms the latest techniques for classifying terrorist attacks with an accuracy of 98.7% using a combined feature set and extreme gradient boosting classifier.

## KEYWORDS

Artificial intelligence; machine learning; natural language processing; data analytic; DistilBERT; feature extraction; terrorism classification; GTD dataset

## 1 Introduction

Terrorism is the most critical threat to human life at any time. It can affect the quality of life for individuals and society. Fear of terrorism restricts people from contributing to the country's development. In every country, dealing with terrorism is a top priority. Unfortunately, terrorism has evolved through time and space, taking many forms. Contemporary terrorism affects a larger geographic area and operates on a fundamentally new scale. There is an unprecedented threat to peace, security, and development. No country can claim to be immune from terrorism. The scale and scope of terrorist attacks have increased over the past decade, destroying entire societies and wreaking havoc in parts of the world. More than 200,000 terrorist attacks have been recorded from 1970 to 2020 [1]. The world has witnessed that most of them were successful over the past decade, as shown in Fig. 1 according to statistics from the GTD. Thus, the threat posed by terrorism is real and severe, sadly, it will remain so in the future [2]. The danger of terrorism also increases given the enormous numbers of terrorist organizations that practice terrorism involving unlimited violence and is not restricted by law or morality, and because of the complexity of the organization and personal activities of these terrorist organizations, in addition to the development of weapons and equipment used by these organizations. Terrorist crime has several effects, as it crime affects the building of society. Whether the loss of innocent victims or the suffering of families, which threatens the cohesion of society. The increase in the scope and spread of terrorist organizations' operations has led to the success of terrorist groups and terrorists in general in achieving their goals, in whole or in part. With the high rates of terrorist crimes, the number of victims and the number of deaths has increased (Fig. 2). In a world that is increasingly dependent on technology, it is necessary to study the issue of the role of AI and its extent. Some researchers suggested systems based on AI that the intelligence officer or strategic decision-maker might conduct a multi-dimensional analysis in an interactive way to benefit from in the fight against terrorism. Security services can make the most of the outputs of those systems that depend on them [3]. The application of methods based on it has recently been applied to develop AI-based counter-terrorism technologies. Some of the previous studies applied machine learning (ML) and deep learning (DL) techniques to make an AI-based model of terrorists. Recent research papers have offered several strategies to study the causes of terrorism and are based on analyzing the pattern of terrorism. In several areas of informatics, including recognizing and forecasting terrorist attacks, there has been extensive research on countering terrorism by comparing the data with the prior histories of suspected organizations or persons. They evaluated information about the operations under investigation, such as the type of operation, location, weapon, and target [4]. Several models were used to improve other particular filtering and classification criteria to attain great accuracy. Despite the efforts of the research community to develop models that help in the war on terrorism, research still makes a meagre and insufficient contribution to the fight against terrorism. The work done still needs to be more effective and needs more effort. There is a research gap in using ML to model and forecast future terrorist operations. This study emphasizes using Natural Language Processing (NLP) tools to extract helpful information from terrorism events to categorize different types of assaults. The proposed framework blends ML and NLP techniques [5]. Early classification techniques relied on the "bag of the word" feature and Term Frequency-Inverse Document Frequency (TF-IDF) [6]. These techniques depended on counting the words/frequency appearing in the text and treating them as text representations without considering the context [7]. Deep neural networks have made tremendous strides in extracting text information during the last ten years. Recent advancements in AI research have led to the widespread usage of numerous neural network-based models for text classification. Pre-trained word vectors such as Word2V provide better initial decoration for sentence tokens. In semantic vector extraction, bidirectional encoder representations from Transformers (BERT) have also been demonstrated. Recent years have seen a rise in interest in the BERT [8]. The BERT model obtains feature representations containing semantic and contextual feature information about the text, which significantly increases classification accuracy [9]. This study aims to improve the classification of terrorist attacks by applying the BERT model in the context of text feature extraction. As a result, a novel strategy incorporating text features and key features has been proposed, focusing on the textual feature attribute. We advocate using BERT-extracted elements as well as additional features in GTD. The former depicts the attacks' semantic content, while the latter is a representation that takes into account the text's structural elements. As far as we are aware, no other works have merged full-text features with the primary features (numeric and categorical features) in GTD to classify terrorist attack types as we have done. The suggested framework combines various GTD aspects with textual (summary) features. The most promising characteristic qualities, chosen based on several prior studies, are being found after a thorough pilot investigation of a wide range of traits in the terrorism-based GTD data set. In this regard, seven features used regularly were found based on the related studies to be combined with the textual features. Collecting a word list from text data and turning it into a feature set the classifier may use is known as "text feature extraction". The compiled summary text features encompass all conceivable attack-related detail that helps with terrorist attack prediction.

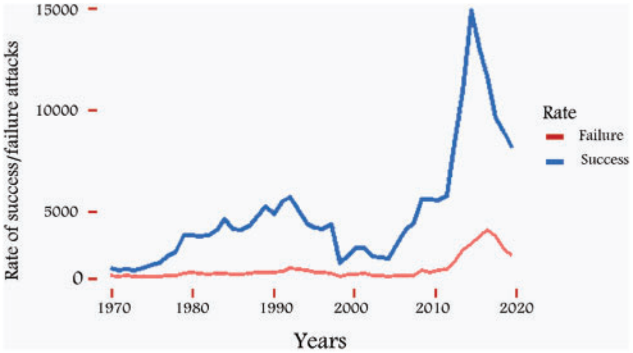


**Figure 1:** Rate of success/failure attacks by years based on GTD

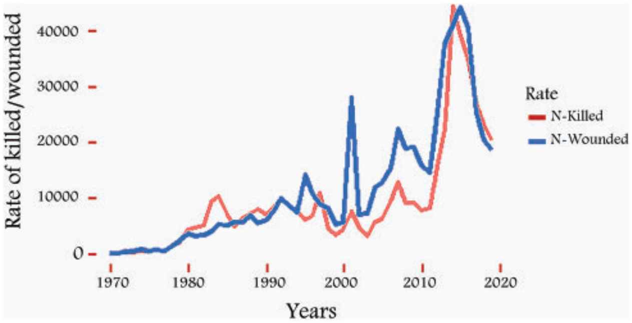


**Figure 2:** Rate of killed/wounded by years based on GTD

In contrast to earlier works, we also incorporated a joint interest module to address the issue of feature interaction. We conducted several experiments to combine features, which can improve the ability of the extracted features to be represented. Our contribution can be summarised in two main points. Firstly, to represent text, our model may extract features with increasing levels of detail from a trained language model. It accounts for semantic and structural information, enhancing the effect of the acquired text representation. Secondly, to avoid the separation of the two traits during the classification process, we created and tested various clustering approaches that can fully exploit the interactions between the two representations to improve classification skills and tests on shared interest units.

This paper explores these techniques to understand and classify the behavior of terrorist activities. These predictions are essential to comprehend counter-terrorism performance; with the help of these tools, terrorist activity can be stopped before it occurs. The research aims to classify and predict terrorist activities based on factors such as success, suicide, type of weapon, and region. In addition to studying the effect of features elicited from summary narration on the performance of ML classifiers, the proposed model is suitable for classifying and predicting future terrorist activities [10].

### 1.1 Problem Description

AI has provided opportunities to analyze big data and predict the future. Among the most promising tools based on AI are search engines, recommendation systems, and NLP. These provide the ability to manage content over the Internet, especially regarding the languages in which small groups of people communicate, thus contributing to the fight against terrorism and extremism. Recently, many academics have focused on multi-label classification, which is now widely utilized. In the real world, applications for data analysis and data instances are linked with levels of semantics [11].

#### 1.1.1 Problem Formulation and Notation

In recent years, there has been increasing interest in multi-label classification that can be applied to many applications. A typical ML routine is dividing the data into training and test sets; the first set is used to learn the attributes of the data, and then these acquired attributes are evaluated on the unseen test set [12]. We expect the correct class to be named for a given input value in classification. We can define the classification problem mathematically as follows: We formulate the task as a classification problem so that our goal is to learn a function $f : x \mapsto y$, where x is a set of input features and y output represents the type of attack. Let $Tn$ be a set of terrorist attack types: $T_1$, $T_2$, $T_3$,..., $T_n$, where n is the number of attack types. Let $C_n$ be a set of labels characterizing each class of attacks (T), $C_k = C_1$, $C_2$, $C_3$,...,$C_n$, where k is the total number of labels. The problem is finding the correct class ($C_k$) for each terrorist attack ($T_n$). To address this problem, we need an exact (f) classifier to map $Tn$ to one of $Ck$ based on some specific parameter or feature. This paper aims to construct a model to classify terrorist attacks using the GTD dataset. The proposed system focuses on extracting and integrating textual and critical features within the benchmark dataset to preserve the knowledge base without duplication or loss of information. The research aims to address the problem mentioned above. To effectively identify and classify terrorist attacks using a rule-based approach to reduce under-represented or uncharacteristic entities obtained from the database. We applied ML classifiers to classify incidents of attack types obtained from object termination and to measure the effects of scripted features on hyperparameters to improve the performance of these classifiers. In our experiment and validation process, we explored different experiments to compare the performance of classifiers. Our research concludes that platform features and other features improve the accuracy of defined collections.

### 1.2 Objective and Contributions

The following contributions and objectives are included in this research work: By using the Global Terrorism Database (GTD), the study has provided two significant contributions: BERT is used for exploratory analysis of terrorist attack trends, text-based feature generation, and textual feature extraction. BERT is a recently created language model that uses a transformer-encoder-based architecture to comprehend text semantics deeply and learn contextual representations from raw text. It is trained using a mixture of 800 million-word novels and 2,500 million-word English Wikipedia [13]. In our study, we fine-tune the pre-trained distilled version of the vanilla BERT named DistilBERT to learn and extract contextual representations from attack summaries. The extracted feature is concatenated with key features to improve the overall framework performance and enrich the attack representation, an ML model for classification exploits. By merging other characteristics and attack summaries, features can be made richer. Many methods have enhanced the models, including implementing more intricate categorization schemes and thorough model validation measures ($F_1$-score, Accuracy, and ROC). To achieve these objectives, we use a text-mining method to analyze the data set's primary variable, the terrorism incident summary. From the synopsis, we try to glean information regarding the kind of terrorist incident. We demonstrate through experiments that classification approaches can gather additional data from news feeds to identify the type of incident. Several classification methods employed in the studies demonstrated that we could extract this information from the free text summary in the database. We used eight fields from GTD for each incident, including Summary (a text field), which describes the incident, and common domains, which are used in state of the art and have as their target domain the kind of incident that derives value from a particular kind of terrorist incident. Text mining is used to process the summary field further because it contains the text. Steps in text mining, including encoding, stop word removal, feature weighting, etc., need this extra preprocessing. Finally, the framework employs a novel method for locating pertinent textual aspects and combining them with other features. Depending on how well our suggested model performs, it may enhance the speed and precision with which attacks are categorized and identified. This demonstrates the promise and efficacy of AI-based models when used in the fight against crime generally and terrorism specifically.

This article uses GTD as a data source to explore several terrorist attack types. The rest of the paper is organized as follows: Section 2 discusses related works, presents a detailed review, and reveals their limitations. Section 3 provides a detailed methodology of the proposed work, covering data exploration and feature preprocessing, including text preprocessing. Section 4 demonstrates the feature extraction model used to prepare the features. Section 5 discusses the creation of classifiers, an experimental setup that includes the data set and evaluation metrics, and a discussion of the outcome. Section 6 presents a comparative analysis with recent techniques. Finally, Section 7 presents the conclusion, work limitations, and future research directions.

## 2 Background and Related Work

This section aims to provide an overview of work in terrorism, integration, and knowledge representation based on AI. Recognizing patterns of terrorist attacks is one of the most critical steps in combating terrorism. Current studies include predicting and identifying conventional terrorist attacks, email tracking, phone signal information, and social network analytics. Although significant, prediction requires more reliable and intelligent techniques that can handle the complexities associated with each terrorist act while simultaneously allowing for extracting knowledge and predictions from various digital data sources [14]. Terrorism has been of interest to many researchers in many fields. A literature review showed that the field of terrorism studies has changed a lot. Communities from various fields have been involved in one or more ways to provide tools that facilitate counter-terrorism. Tables 1 and 2 outline the research progress in the terrorism domain.

**Table 1:** Summary of the related survey

| Studies | Focus of review | Main contribution |
|---|---|---|
| [14] | Overview of the open source intelligence (OSINT) cycle in the context of terrorism-related information from various data sources. | Examined how OSINT can be used to extract textual information on terrorism from sources that are openly accessible. |

(Continued)

**Table 1 (continued)**

| Studies | Focus of review | Main contribution |
|---|---|---|
| [15,16] | Introduced to databases and datasets on terrorism information. | Reviewed the layout and properties of terrorism datasets concerning their use for terrorism and counter-terrorism research. |
| [17] | Terrorism on the dark web. | Review methods for identifying and handling terrorist websites' content on the dark web, such as data mining and text analysis techniques, highlighting difficulties and gaps, including DL and ML. |
| [18] | Studied the techniques of practical analysis of terrorist activity data on Twitter. | They studied technology providing accurate predictions of terrorist activities using sentiment analysis for terrorist-related tweets based on 17 articles and presented their findings. |
| [19] | Reviewed computational techniques to counter-terrorism on social media. | An overview of computational strategies based on the phases of a terrorist attack. |
| [20] | Created new avenues for research into online extremism detection, categorization, validation techniques using large datasets, and tools for detecting extremism that is not restricted to a few ideologies. | From 2015 to 2020, only 64 studies were included for the survey. Studies retrieved from libraries, including SCOPUS, ACM, Web of Science, and IEEE, were manually screened. As a result, the survey is subject to a bias risk because of the search query utilized. The research focused on the alt-right, jihadism, and white nationalism were just a handful of the ideas or groups. |
| [21,22] | Studied social network analysis techniques to counter-terrorism on social media. | Analyzed data collection methods and tools used in counter-terrorism and social network analysis. |

### *2.1 Related Survey*

Since the U.S. tragedy of September 11, there has been an increase in public awareness of terrorism. It has brought attention to the absence of terrorism-related information. The war on terrorism has sparked a renewed search for more effective methods of obtaining and analyzing intelligence data. The capacity to evaluate large amounts of data using different AI tools to detect possible threats is often an important activity in counter-terrorism. The body of literature devoted to intelligence-surveillance data has grown significantly recently. But when the information is dispersed across various departments and comes in multiple formats, such as phone tap scripts, internet usage logs, emails, seized hard drives, CCTV video, images, etc. Many research articles summarized the technological issues in obtaining terrorist data and outlined many essential solutions. Researchers such as [14–22] reviewed current AI methods in terrorism domain. Table 1 summarizes some of this research work by examining different AI approaches in counter-terrorism.

### *2.2 A Machine Learning Approach for Enhancing Defense against Global Terrorism*

According to literature studies, systems that use data mining and ML techniques may instantly detect, track, and predict potential terrorist behavior in real-time. This section describes how terrorism has been countered using AI approaches like data mining, ML, and DL. Among these works are predicting the crime category, anticipating the perpetrator, geographic and socio-economic features, predicting the future trend and quantifying risks, and predicting and classifying attacks. ML techniques such as classification and clustering are at the core of the solutions that computer scientists and statisticians provide to detect patterns of terrorism attacks [23]. Sarda et al. [24] developed a classifier that divides diverse non-situational tweets into several groups. Different types of tweets have been categorized into specific categories by ML models. Only English-language tweets uploaded during the horrific events of the Gurudapur terrorist attack and the Nepal earthquake were deemed non-situational tweets based on Twitter's selected language. This method eliminates group tweets, which in the event of specific crises, might worsen the problem. By categorizing tweets into three groups those that address terrorism, those that do not, news, and tweets that are not tainted by other information. Fraiwan [25] attempted to discern between themes connected to terrorism. Twitter posts that address terrorism and those that have not been divided into categories using algorithms. The accuracy of their model ranged from 78% to 83%, which is typical when examining the hundreds of tweets that supported and endorsed the Islamic State (commonly known as ISIS). Using Wikipedia, Wikidata, and semi-supervised learning, Zajec et al. [26] created a system that tests earthquakes and terrorist attacks. The study automatically generated a small noisy, labeled dataset and a large unlabeled dataset using Wikipedia and Wikidata. The study applied a semi-supervised event argument extraction system. The dataset comprises event clusters with multilingual Wikipedia articles. Semi-supervised learning and probabilistic soft logic are used to label the unlabeled data iteratively, and each example's pseudo-label is inferred from the predictions of numerous base learners. In a multilingual situation, the suggested approach is applied to Wikipedia entries regarding earthquakes and terrorist incidents. Their tests indicate that using the proposed technique improves the outcomes. When trained using the method that combines probabilistic soft logic with semi-supervised learning, the system gets an $F_1$-score of 0.79% when just the automatically labeled dataset is utilized and an $F_1$-score of 0.84% when both datasets are used. Kant et al. [27] proposed a method for evaluating the informative quality of topic models. The study used ML classifiers to categorize geocoded tweets from the social networking site Twitter according to the location of the message by feeding them an LDA model and STM out-of-sample topic predictions. The performance of a state-of-the-art ANN is then compared to the prediction performance. The ANN employed one of the best pre-trained word embeddings, GloVe. GloVe was trained explicitly on Twitter data and all relevant tweet information. Bridgelall [28] used two AI approaches to determine perpetrators' motives from a massive database of terrorist acts globally. First, they categorize attackers' motivations into six categories using NLP and ML techniques: hate, protest, revenge, vulnerability, strength, and intimidation. Next, they developed an empirical course of action for classifying subjects using NLP methods to extract text features from narratives of the motivations of terrorist incidents. Then they used extracted features from a brief narrative of each event in the GTD dataset to train 11 different ML models. According to the results, the Extreme Gradient Boosting model achieved the best prediction performance. Pan [1] proposed a paradigm for predicting terrorist organizations with the highest attack frequency. The framework includes five classifier prediction models, data splitting, model evaluation, and data preprocessing. Based on a quantitative statistical examination of the activity of terrorist groups in GTD from 1970 to 2017. The SelectKBest feature selection method was used with five ML models. Experiments showed that the five models significantly improved at predicting the 32 terrorist groups responsible for the most attacks.

Olabanjo et al. [29] created an ensemble ML model incorporating an SVM and KNN. To predict continents susceptible to terrorism from GTD, two feature selection techniques, Chi-squared, Information Gain, and a hybrid of both, were applied to the dataset. According to their findings, hybrid-based selection characteristics generated the best outcomes among feature selection strategies for forecasting terrorist sites. Abdalsalam et al. [30] introduced a framework for predicting terrorist attacks on the GTD dataset. The research used textual features extracted through different text representation methods (Bow, TF-IDF, and W2vec) and combined them with other features. Nine different classifiers are used. The results show that the proposed framework improves the prediction accuracy significantly. To obtain early warning of terrorist acts and to increase model stability and classification accuracy. Feng et al. [31] and Feng et al. [32] proposed two frameworks to provide information for decision support and early warning of terrorist attacks. In their suggested methodology, features are chosen using a unique method that combines random forest (RF) and principal component analysis (PCA). The XGBoost hyperparameters are tuned using a genetic algorithm (GA) on the GTD dataset, where their suggested approach is assessed. Experiment results show that the suggested method works well for the dataset of terrorist attacks in China and can be widely used. In their other framework, a multilayer depth Neural network (NN) Graph convolutional networks (GCN) model (NNGCN). A multi-layered deep neural network is employed to categorize terrorist incidents. Five datasets, namely: Cornell, Texas, Washington, Wiki, and terror attacks, were the subjects of experiments. The NNGCN model introduces the idea of a correlation index between event nodes, and it is merged with important information. According to the findings, correlation prediction accuracy increased by roughly 33 percentage points, while classification accuracy increased by about three percentage points. The new model performs better when analyzing the actual circumstances of terrorist attacks than the previous model in node categorization accuracy and association prediction.

Huamaní et al. [33] used classification models, decision trees, and RFs to visualize and forecast future terrorist strikes. An organized database of terrorist attacks from 1970 to the most recent year of record, 2018, was used as the input. The probability results from 75.45% to 90.414% of the bundles are the same for the decision tree and RF models, respectively. These findings show that ML approaches are perfect for adding to research pertinent to current global events.

Hu et al. [34] and Zhenkai et al. [35] proposed two risk assessment models for terrorist attacks using the GTD. The results of the experiments show that the suggested method can be used to analyze and predict information about terrorist attacks thoroughly and accurately.

Saiya et al. [36] used a specific kind of algorithmic analysis known as C4.5 classification trees to examine social and political violence. Their proposed method has been applied based on four unique outline techniques for comparing classification models for the C4.5 classification method. Their state-level research shows that religious and secular terrorists have distinct target choices. This study shows how classification trees can help to comprehend terrorism and help countries and governments develop strategies and search for new ways to face this emerging challenge. Meng et al. [37] presented a novel framework for predicting terrorist attacks using a hybrid classifier and the GTD data set. Additionally, the genetic algorithm is used to optimize the weight of each classifier, thereby combining multiple classifiers. They found that the proposed model with a hybrid classifier outperformed the single classifier when predicting the kind of terrorist attack.

As an overview, AI can predict terrorism by analyzing metadata and inaccessible patterns. Communications and information about financial transactions, travel patterns, surfing activities, and publicly available information such as social media activity enable the identification of terrorists by distinguishing what characterizes the activity of a particular subgroup on these media [38]. These methods include analyzing relationships between entities or using more complex tools for an image or sound recognition [39,40]. Through advanced data analytics and ML techniques, AI can help identify potential terrorists and prevent or mitigate the risks associated with terrorist activities. As outlined in the related works, several types of research have been published in the counter-terrorism domain, as shown in Table 2. As a result, there are concerns over the limits of its predictive uses in the fight against terrorism and the associated dangers and potential. The literature survey demonstrates the use of single- and ensemble classifiers. However, the difficulties of small sample sizes and unbalanced data sets are some of the drawbacks of the current investigations.

According to the state-of-the-art survey that deals with terrorism attack classification, the study found that most of the previous studies dealt with the classification of terrorist attacks in the GTD dataset. Classification is one of the most common machine-learning problems. The best way to approach any classification problem is to start by analyzing and exploring the dataset in exploratory data analysis (EDA). The purpose is to generate as many insights and information about the data as possible. It is also used to find any problems in the dataset. One common problem in GTD datasets used for classification is the problem of imbalanced categories. Data imbalance usually reflects the unequal distribution of categories within a dataset. Many traditional ML techniques are less successful due to the skewed distribution, particularly when anticipating minority class cases; however, the study demonstrated that some researchers did not care about it when dealing with terrorism attack classification. There are other issues when a class disparity is not only prevalent but also anticipated. The question is, how can models that handle unbalanced data be evaluated? accuracy is not the best statistic to use in this situation. However, several studies that looked at the same issue employed only accuracy measurements [41,42]. Looking at performance metrics, the metric to use should not be limited to accuracy when working with an unbalanced dataset. There are more efficient metrics that can give insight into model accuracy than traditional classification accuracy when working with unbalanced categories. An example is a recall measure: Scale of Completeness of classifiers, F1 Grade (or F Grade): a weighted average of accuracy and recall. ROC curves: Like accuracy and retrieval, accuracy is divided into sensitivity and specificity, and models can be chosen based on the equilibrium thresholds for these values. A variety of algorithms are used. Moreover, researchers discussed using data mining technologies to increase the accuracy of global information to fight terrorism. However, previous research has not fully exploit additional elements, such as association information present in the events and the association between terrorist attacks based on the graphical structure.

**Table 2:** State-of-the-art on the research progress

| Ref. | Techniques | Aim of work | Dataset | Limitation |
|---|---|---|---|---|
| [27] | ML Classifiers such as Naive Bayes, KNN, SVM, and XGBoost classifiers and ANN | Classify Twitter posts that have been geocoded into categories based on their location | Twitter | The outcomes of an LDA model that was trained using the text of tweets could not be used by the ANN to outperform ML classifiers. In addition, topic model output and a neural network acting as the classifier could be combined. |

(Continued)

**Table 2 (continued)**

| Ref. | Techniques | Aim of work | Dataset | Limitation |
|---|---|---|---|---|
| [43] | Hybrid DL platform based on convolutional neural network (CNN) and long-term memory (LSTM) model | Determine the traits of future terrorist operations | GTD | Have little interest in incorporating the regional geographic characteristics of terrorist actions to obtain a deeper understanding and forecast terrorist activities in the future. The suggested hybrid method performs well with similar data sets but could be better with more varied data sets. Numerous issues with the GTD dataset exist, such as data imbalance not being discussed and mentioned. |
| [44] | Remote sensing, ML, and spatial technologies | Predicts the presence or absence of terrorism in Europe on a previously unexplored spatial scale | GTD ACLED ICEWS | The study utilized a hexagonal grid cell technique, covering an area of 25 square kilometers, to identify the most probable locations for terrorism occurrences. However, it did not take into account the adjacent pixel values in the raster used for sampling the features. Future studies should incorporate a more comprehensive analysis of the spatial-temporal aspects of terrorism by considering the neighboring pixel values within the raster datasets. |
| [28] | Descriptive Analysis: ML models used for topic modeling and text classification to identify the perpetrator's aim of the attacks | Aims of the attacks are identified in six categories: Protest, retaliate, intimidate, weaken, force, and despise | GTD | Manual effort required for empirical topic modeling. |
| [1] | Five ML classifier prediction models | Predicting terrorist organizations with the highest attack frequency | GTD | The method can be extended to predict a broader range of terrorist organizations. The model could handle only a small number of terrorist organizations; however, the study did not mention the effect of increasing the number of organizations. |
| [30] | ML and NLP with textual features | Classification and predication of attacks types | GTD | The difficulties related to training ML models due to high dimensional data. |
| [29] | Ensemble ML model which combines SVM and KNN | Prediction of continents susceptible to terrorism | GTD | The study predicted which continents would be vulnerable to terrorism. Even though it is believed that predicting countries will be better, this was avoided because it would result in too many categories. Models may perform better when there are fewer categories. |

(Continued)

**Table 2 (continued)**

| Ref. | Techniques | Aim of work | Dataset | Limitation |
|---|---|---|---|---|
| [5] | DNN and ML models (LR, SVM, and NB) | Prediction of terrorist activities, including the likelihood of suicide attacks succeeding, the type of weapon to be used, the location of the attack, and the type of attack | GTD | excludes predictions based on behavioral analysis of terrorist organizations. |
| [31] | ML-based model, XG Boost that is based on the RF is used in predictive analysis | Determine whether terrorist strikes will result in the deaths of innocent | Terrorist attacks in China and the GTD | The strategy ignores qualitative elements in favor of a quantitative analysis of the features. |
| [32] | Multi-layered deep neural network (NNGCN) model of GCN. A multi-layered deep neural network | To study classification and early warning of terrorist attacks | Cornell, Texas, Washington, Wiki, terror attacks | GCN's internal structure must be changed to improve the representation of node features. For instance, it is possible to set up various convolution layers with the required filtering functions in between. |
| [45] | Ontology-based knowledge graphs were established between entities, such as places, people, and organizations, using links and verbs based on violence, such as "attack," "killed," etc. | Constructing a framework to develop a knowledge base that includes crime entities and relationships between them | Online news sources | LDA cannot show correlations resulting in uncorrelated themes. LDA implies that words may be interchanged and do not represent sentence structure. |
| [34] | The K-means clustering algorithm and quantitative research | Risk assessment models for identifying hidden or developing terrorist organizations and networks of terrorist organization alliances | GTD | One major limitation caused by k-means are that it constrains all groups. This means k-groups do not work well when groups have irregular shapes. It is also challenging to integrate categorical variables. As it is familiar with many clustering algorithms, it is sensitive to outliers and has difficulty with high-dimensional data. |
| [36] | The C4.5 classification approach used to examine terrorism | The analysis demonstrates how classification trees might deepen knowledge of terrorism and even provide politicians with advice on preventing further attacks | GTD | Many real-world ML applications in the industrial, finance, and medical areas still favor tree-based ensemble techniques like RFs and gradient-boosting machines. The drawback of tree ensembles is that it is difficult to comprehend the internal workings of complicated models. |

Accordingly, there is a need to pay more attention to the imbalance of categories that express the type of attacks, which makes most of the findings of these studies questionable, especially since they use accuracy scale to measure the model's performance. This research sheds light on the role of AI-based technologies role in the counter-terrorism domain [46]. The suggested methodology for classifying terrorist attacks takes linguistic elements into account. If we train the classification model without fixing this problem, the model will be completely biased. This bias also impact the relationships between features. However, the overall performance of ML models built on unbalanced datasets will be limited by their ability to predict rare and minority points. Identifying and resolving imbalances at these points is critical to the quality and performance of the generated models. Balancing the data before splitting may lead to bias in the test set as some data points in the test set are synthetically generated and are well-known from the training set. Low classification accuracy is a problem that the model can address. The suggested approach enhances the performance of the classifiers by integrating textual features with the features employed intermittently in state-of-the-art studies. It is likely to improve the research that is already being done on how to use predictive AI to fight terrorism. Moreover, the research focus should be on analyzing the integrated performance of the new test data using various ML algorithms. DistilBERT, which more accurately reflect the semantic aspects of the text than other embedding methods like Word2vec and Glove, was used to extract the semantic features of the text [47]. Based on some of the existing pitfalls of the previous study, this research presents a unique hybrid method using automatic feature extraction from text features based on DistilBERT. It combines these features with existing features in state-of-the-art to classify terrorist attack events.

### *The Proposed Study Contributions vs. the State of Arts*

Compared with the state of arts mentioned in this study, the main contributions of this paper are as follows: This study represents a first step in testing the possibilities of using text features extracted from GTD summary attributes with the other features to test the effect of various feature combinations that can be used to help distinguish terrorism types. Textual features play an important role in distinguishing terrorism types, which motivated us to utilize summary textual representation. This kind of representation using the BERT model is robust in extracting high-level text-based features, including semantic, contextual, and syntactic features. Irrelevant or partially relevant features can negatively impact model performance, so to make a judgment of our model and for more validation, we have to take a random sample as an example from test data to present various cases and scenarios to understand which features are contributing positively and negatively to the model's performance. This can give us more confidence in the model's performance. The main contributions of this work are summarized as follows:

(i) To the best of our knowledge, this is the first terrorism classification framework based on BERT and ML and using NLP techniques.

(ii) The proposed framework combined various features extracted from the GTD dataset to classify terrorist events. Additionally, we employed 15 ML models to identify a suitable model for the GTD dataset. Our proposed model enhances classification performance by combining textual and related features with various metrics, providing further validation.

(iii) Experiments are carried out on publicly accessible (GTD datasets) datasets. The results show that our proposed model outperforms state-of-the-art models in classifying terrorist attack types.

## 3 Research Methodology

This section provides an overview of the proposed framework and explains how textual features are utilized (Fig. 3) to classify the types of terrorism attacks. The framework consists of the following steps: gathering the data, preparing it, representing the features, selecting features, building the model, and testing it. In early studies, statistical features of the text were employed to analyze and extract text data for specific objectives. These statistical features were then used as training data to classify sentences or documents. Common word-based representation techniques include "bags of words" (BOW) and "n-grams" [48], which express a sentence using a group of words or n-gram sequences that are included in it. When used with ML models like SVM [49], these features have produced positive results. However, because ML relies on domain expertise and substantial feature engineering, it is challenging to generalize the features of one task to other aspects. DL-based methods have addressed the shortcomings of manual feature engineering. The main reason why DL-based frameworks have gained so much popularity recently is that they do not rely on hand-crafted functionality [50]. The model automatically maps the text to a low-dimensional vector to extract text features. Pre-trained language models have significantly increased in a study recently. Models like BERT have garnered significant attention in NLP research [51]. Accordingly, in this study, the textual features, namely summary attributes, are represented using Distill-BERT. The BERT-extracted features from the text are integrated with the other key features. In this research, we demonstrate how the format and characteristics of the data are used to augment the data provided by the feature vectors of the training samples. The performance of a classification task is influenced by the training of the model. The pre-processed input text can be transformed into useful feature vectors using the hybrid feature combination proposed in this research. This enables effective, reliable, and robust analysis. The results show that the hybrid strategy significantly enhances classifier performance compared to other techniques.

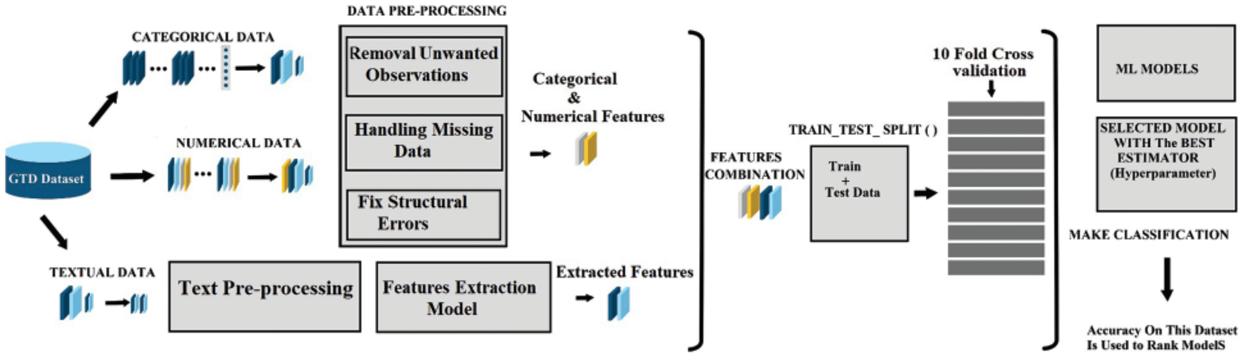


**Figure 3:** Proposed framework for terrorist attack classification based on a combination of features

## *3.1 Explore Data*

### *3.1.1 Terrorism Attack, and Alternative Types*

GTD has shown different sorts of attacks besides bombing and blasting: assassination, armed assault, kidnapping, two types of hostage-taking, facility and infrastructure attacks, and unarmed attacks. It is essential to distinguish these techniques from bomb attacks. The three categories of anti-human bodies, anti-material objects, and hybrids can be used to categorize all eight species. Armed assault attacks under GTD aim to kill or hurt people using weapons or other lethal equipment [52]. Assassination and hostage-taking (of two varieties) are also targeted human attacks. Contrary to the first category, hijacking is an attack on a physical object whose primary goal is to seize control of the target facility's infrastructure and vehicles. The "9/11" incident is an example of the first type, whereas the bus attack in Angola in August 2001 exemplifies the latter. Explosive attacks serve a variety of goals in contrast to other varieties. Targets can be both individuals and objects. For instance, coordinated assaults in Iraq in August 2007 selected and destroyed towns while making minorities the number one target [53]. Aside from the difference in aim, the main distinction between explosives and other methods is that explosives are frequently very destructive, small, cheap, and simple to make. Terrorists have also gained a "force multiplier thanks to improvements in the affordability, portability, and concealment of weapons. In contrast to other strategies, kidnapping is regarded as" one of the more difficult acts [54]. The needs of terrorists appear to be best met by explosives. Although explosives can launch attacks with various advantages, the prevalence of particular attacks is influenced by several coupling variables and not only the advantages of the weapon itself, such as convenience, affordability, and concealment. The aim is one of the most crucial factors, along with weapon choice, from the perspective of a situational crime prevention study. Specific terrorist attacks are brought on by target exposure, vitality, destructibility, and originality [53]. Then, suppose the terrorist group's expertise is limited to attacks on vulnerable or poorly fortified targets. In that case, it is unlikely that it will serve as a learning ground for terrorist groups that operate against vital, even military targets. At the same time, the institutional framework in which the target is located frequently relates the level of defense to several political, military, and economic elements. As a result, comparatively, unarmed offensive attacks aim to accomplish the same using different methods, such as biological, chemical, and radioactive. It is essential to consider larger contexts and where various terrorist attacks can be categorized in the same category base on the similarity of attacks. In this context, it should be noted that the unknown type of attack was excluded. According to the explanation above and a case study conducted using the GTD database, attack methods can be classified into three categories: assassination, armed attack; bombing/explosion; and hijacking (hands-free attack and others) [53]. The outcomes of various attacks vary dramatically, and armed attacks and hijackings cause the highest number of casualties. Although assassinations are the most common type of attack, they do not result in many deaths because they frequently have a precise target figure and do not result in many bystander casualties. Weapons fall into three categories: the first category (biochemical, nuclear, missile, and radiation weapons), the second category (light weapons, explosives/bombs/explosives, and burning weapons), and the third category (light weapons, light weapons, explosives/bombs/explosives, and burning weapons) (fake weapons and vehicles) [55]. Accordingly, we also decided on a class grouping that merges similar-looking classes into single merged classes, the merged class namely Hostage Taking (Kidnapping) and Hostage Taking, which have the same general meaning, were combined with maintaining the balance of data which reduced the number of classes and slightly reduced the class distribution imbalances. Finally, eight attributes and 40720 records, and seven types of terrorist attacks with complete data were preserved during the experiment.

### *3.2 Data Preparation and Preprocessing*

#### *3.2.1 Dataset*

Terrorism databases have gained significant importance by providing a security vision on a sound scientific and technical basis. Security plans to combat terrorist crimes cannot be developed without accurate and updated databases of terrorist operations. Terrorist databases play a critical role in the fight against terrorism, with an increasingly important role in identifying and preventing terrorist fighters from crossing borders [10]. The Global Terrorism Database (GTD), which provides details on terrorist attacks worldwide, is one of the most significant databases (with annual updates). Unlike many other event databases, the GTD presently contains more than 180,000 cases and regularly updates data on domestic and international terrorist acts that took place throughout this time. The University of Maryland developed GTD as an open-source project featuring data sets on international terrorism from the 1970s through 2020. It details the incident's date and place, the weapons used, the targets, the number of victims, and guilty groups. In-depth, trustworthy, and open-source data from GTD is available to researchers to help them identify and foretell terrorist acts [56]. Assassination, Assault with a weapon, bombing or explosion, Hostage taking (Barricade Incident), kidnapping, attack on a facility or infrastructure, Assault without a weapon, and unknown are the eight main categories of terrorist acts currently in existence, according to the GTD. The following information pertains to the attack:

1. **Assassination:** An action carried out to kill one or more public figures or celebrities.
2. **Armed Assault:** Is any attack that uses a weapon, such as a gun, an incendiary, or a sharp instrument like a knife, with the primary intent of harming or killing someone.
3. **Bombing/Exposure:** An attack in which the main effects come from a substance that breaks down quickly and sends out a pressure wave that hurts the surrounding area.
4. **Hijacking:** An assault on human freedom if the offender assaults the victim by moving him from one place to another against his will and without legal basis; Alternately, an act in which the offender pursues hidden ends and goals such as forcing the government or authority to release people or achieve another political goal.
5. **Hostage Taking (Barricade Incident):** A person commits the crime of Hostage taking if they arrest or detain another person and threaten to kill or hurt him or keep him detained to force a third party, such as a state, an intergovernmental organization, a natural or legal person, or a group of people, to do or not do a particular act as an explicit or implicit condition to release the Hostage.
6. **Hostage Taking (Kidnapping):** An act in which the main goal is to get control of hostages in exchange for concessions or to stop normal operations.
7. **Facility/Infrastructure Attack:** An action is done with the main goal of hurting something that is not human, like damaging civilian buildings and critical infrastructure facilities, like homes, schools, places of worship, etc.
8. **Unarmed Assault:** An attack whose primary goal is to cause physical harm or death to another person with a weapon other than an explosive, firearm, incendiary, or sharp tool (like a knife).

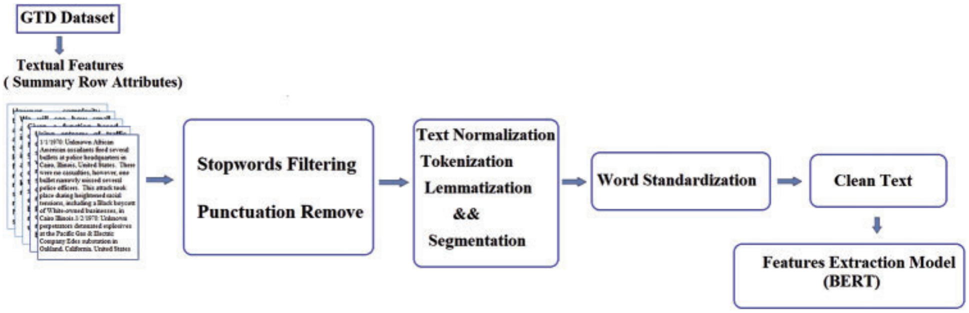
### *3.2.2 Correlation in Dataset Attributes*

There are up to 134 attributes in the GTD dataset. Accordingly, the features of the dataset are analyzed. Then we evaluated the correlation between the features base-on the literature research. Which studied the classification of terrorism attacks process on the GTD dataset. The interpreted result found that most studies such as [30,37,57–59] often employed the seven features in their work. These attributes are shown a demonstrated effect. Accordingly, the seven attributes are kept to make the prediction process more straightforward, and then they are encoded using a label encoder before being combined with textual data. Based on characteristics utilized in the state-of-the-art, seven attributes closely related to terrorist attacks are kept in the preprocessing stage to support classification. These attributes include the city (where the terrorist attack happened), the specific area (a category variable that represents the specific area where the terrorist attack happened), the type of attack (a category variable that includes assassinations, kidnappings, armed attacks, hijackings, roadblocks, infrastructure damage, unarmed attacks, explosions, and unknown), the name of the terrorist organization (the group that made the terrorist attack), the date of the terrorist attack, the amount of loss sustained, and whether the attack includes a ransom request.

### *3.2.3 Data Preparation*

Real-world data is often cluttered, contains missing values, and may be in an unusable format, making it incompatible with ML models [60]. In addition, it makes acquiring knowledge during the training phase more complex. Preprocessing data may affect how the final data processing results are interpreted. Data preprocessing is an essential step in the data mining process. This often leads to the presence of missing and out-of-range values in the collected data. Failing to carefully examine the data to address these issues before analysis can result in misleading and illogical results [61]. Therefore, ensuring data representation and quality should be prioritized before performing any analysis. Often, data preprocessing is the most critical stage in ML. The steps involved in preparing and filtering data can be time-consuming to process. Cleaning, deleting missing values, selecting features, and digitizing data are all examples of data preprocessing, and to improve prediction or classification accuracy, it is important to ensure that the data source is complete, continuous, and noise-free [62]; As a result, the processing of the data source involves various steps such as cleaning the raw data, selecting relevant features, optimizing, transforming, and extracting properties, among others. The output of the data processing stage is the final training set. This section describes the two-stage data preprocessing process. The first stage, consisting of general preprocessing, was applied to all features, while the second stage, consisting of text preprocessing, was applied to textual features. This section discusses the data preprocessing steps used by the proposed framework.Data cleaning involves the removal of invalid, insufficient, and inaccurate data from datasets, as well as replacing missing values [63]; Several data purification techniques are used to prepare the GTD dataset.

**Pre-Processing of Textual Data (Summary Attributes)** Big data is the fuel that powers AI. The diverse amount of big data enables ML applications to acquire and master skills. NLP is an AI application that enables computers to understand and process human languages. NLP is critical for analyzing text data. Text processing is an essential part of NLP. It has many sub-fields, such as text compilation, classification, machine translation, and human-computer interaction. The idea behind all of them is to turn text into data that a computer can easily understand and use [64]. Text pre-processing is a stage of textual processing limited to the summary's text attributes. ML models methodically search the data for patterns that illustrate how the characteristics of the input attributes of the model connect to the label of the output. The input attributes' data representation directly impacts the final model's quality. Handling structured numerical inputs is relatively simple, but the data needed to train a machine-learning model can come in many forms, such as category variables, text, pictures, audio, time series, etc. Prior to utilizing the summary feature, it is necessary to apply preprocessing techniques to clean the text and convert it into a vector format that ML classifiers can utilize effectively. As a result, this summary is an essential feature of the proposed framework. It is informative and briefly describes the terrorist attack, including when, where, who, what, how, and why. Before the summary feature can be used, some preprocessing techniques must be used to clean up the text, as shown in Fig. 4. NLP tools are employed to process text data, and consistent criteria are applied to all data fields to ensure accurate results. Completing these steps is crucial before integrating text data into a new model. In this process, words are identified and encoded based on their parts of speech, such as nouns, verbs, adjectives, adverbs, and pronouns. Additionally, word derivation is employed to reduce words to their root forms. During encoding, the text is broken down into smaller semantic units or individual sentences. Stop words, including phrases that do not provide new information (e.g., prepositions), are eliminated. Pre-processing is crucial in turning raw text data into a structure that can be used [65]. These preprocessing techniques remove redundant data and transform the textual input into a more comprehensible feature extractor format. Finally, the textual features are further processed for analysis and modeling.



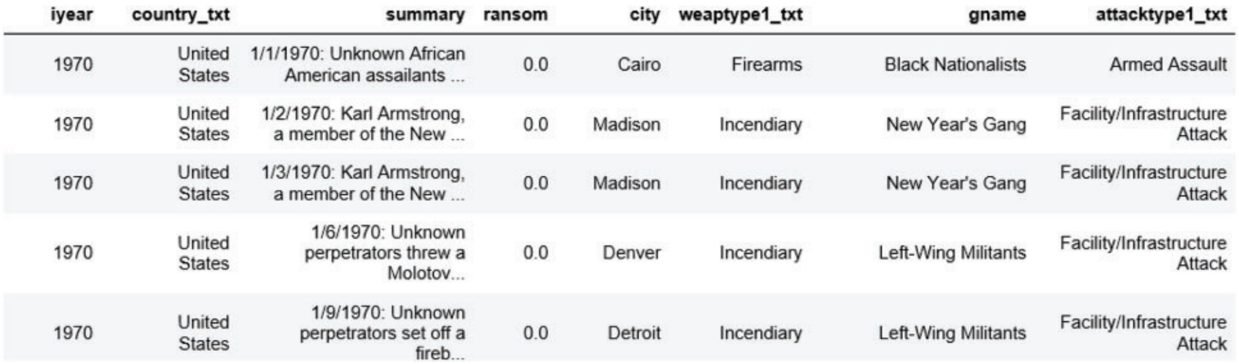
1/1/1970: Unknown African American assailants fired several bullets at police headquarters in Cairo, Illinois, United States. There were no casualties, however, one bullet narrowly missed several police officers. This attack took place during heightened racial tensions, including a Black boycott of White-owned businesses, in Cairo Illinois.1/2/1970: Unknown perpetrators detonated explosives at the Pacific Gas & Electric Company Edes substation in Oakland, California, United States

**Figure 4:** Text preprocessing techniques used for preparing summary text attributes

Removing outliers, replacing missing values, symbols, word filtering, and memes are all standard preprocessing techniques used by the proposed framework. The high-quality data leads to more accurate models and classification. Tokenization breaks down a group of texts into their words by removing punctuation and substituting the white space between them. In English text classification, stop-word filters are essentially always used. The term "spelling correction" refers to correcting misspelled words. For instance, "attackrr" rather than "attack." In addition, word lengthening is a spelling mistake in which the letters that make up the word are mistakenly repeated, as in "awwsome" rather than "awesome." The root words of the inflected words are then revealed via lemmatization by doing a morphological analysis of the inflected words using a dictionary. The word is changed back to its original form [66]. As shown in Fig. 3, data pre-processing is divided into two parts to make the procedure easier. A textural pre-processing stage is used in the first stage to condense the features described in the text. Since the abstract feature is a textual feature, various pre-processing methods must be employed to tidy up the text and turn it into a vector so that ML classifiers can use it (Fig. 4). These pre-processing techniques seek to eliminate extraneous data and transform the input into a more understandable structure for the feature extractor. Words and memes are filtered out, missing numbers and symbols are replaced, and outliers are eliminated, as shown in Fig. 4. It also breaks down a text block into its words by replacing the white space between words with punctuation. One fundamental process is converting the retrieved features into digital form after pre-processing and extracting the features from the GTD dataset. The proposed framework's categorical features are converted to numbers using nomenclature coding with scaling. As shown in Fig. 3, text representation techniques are used to turn the entered text into features to create feature vectors. The processed text features are routed via the BERT model and transformed into vectors.

These traits are kept for simple prediction before combined with the text data and encoded with a label encoder. Data in the chosen attributes are both categorized and numerical. ML models frequently incorporate many data sources, handling various data kinds. Vectors representing several qualities are gathered into a single composite vector in traditional clustering algorithms. ML must incorporate these properties into a single model to use this characteristic, including GTD text functionalities as optional extras. A hybrid data frame that combines features collected using essential features and representation techniques results from the data transformation. Since they function independently from the classification models in the experiments, each is evaluated separately for comparison.

**Pre-processing of Main Features (Key Features)** After the data is acquired, the raw data must be prepared, explored, visualized, and transformed. These steps are repeated until the data is ready to create the model. Data preparation includes the revision and processing of raw data before analysis. Understanding the available data before creating any ML model is necessary. Raw data may be cluttered, redundant, or inaccurate. Data should be cleaned by eliminating external noise and missing value information by identifying, replacing, or deleting corrupt, inaccurate, and incomplete data; data merging; data transformation; and data reduction without compromising the outcome. The primary objective of this stage is to eliminate terrorist attacks with missing values and to maintain the classification of terrorist attacks. GTD consists of 135 features. Many of them have many missing values, with 132,947 records missing. After this step, seven features are closely linked to terrorist attacks, and a text summary feature was chosen. As a result of that, a total of 47,053 records are considered. Literature reviews indicate that seven attributes are regularly applied throughout numerous studies. Specific attributes are selected to improve classification before encoding with a label encoder and textual data. The attack's city, region, and attack type classify attacks, including assassinations, kidnappings, armed attacks, hijackings, roadblocks, infrastructure damage, unarmed attacks, explosions, and unknown attacks. These are examples of attributes that contain both categorical and numerical data types, as shown in Fig. 5. Additional information includes the type of weapon used in the terrorist attack, the quantity of property destroyed, and whether ransom or not. The terrorist organization name feature also reveals the name of the terrorist organization that committed the attack. Table 3 shows the data set attributes considered when the proposed framework was made. It should be noted that the group of non-textual attributes selected at this phase will be referred to as "significant features." Scientific data sets often only include one sort of data, such as text, pictures, or numbers. ML models, however, usually incorporate data from several sources, dealing with various data. A composite vector is often created by combining several feature vectors. ML must merge these properties into a single form to use this property. Text features for GTD should be added as a supplement. Text representation techniques are used to transform text data into feature vectors. When data is modified, data frames are created by fusing the primary features with features generated using textual representation approaches. The performance of NLP is integrated with ML for accurately classifying terrorism attacks based on text features. DistilBERT aims to extract features from the text data in GTD.



| iyear | country_txt | summary | ransom | city | weaptype1_txt | gname | attacktype1_txt |
|---|---|---|---|---|---|---|---|
| 1970 | United States | 1/1/1970: Unknown African American assailants ... | 0.0 | Cairo | Firearms | Black Nationalists | Armed Assault |
| 1970 | United States | 1/2/1970: Karl Armstrong, a member of the New ... | 0.0 | Madison | Incendiary | New Year's Gang | Facility/Infrastructure Attack |
| 1970 | United States | 1/3/1970: Karl Armstrong, a member of the New ... | 0.0 | Madison | Incendiary | New Year's Gang | Facility/Infrastructure Attack |
| 1970 | United States | 1/6/1970: Unknown perpetrators threw a Molotov... | 0.0 | Denver | Incendiary | Left-Wing Militants | Facility/Infrastructure Attack |
| 1970 | United States | 1/9/1970: Unknown perpetrators set off a fireb... | 0.0 | Detroit | Incendiary | Left-Wing Militants | Facility/Infrastructure Attack |

**Figure 5:** Example of selected features in GTD

**Table 3:** An overview of the GTD dataset features utilized in the proposed framework

| Features name | Description | Data type |
|---|---|---|
| Iyear | Includes the incident's year of occurrence | Numeric |
| City | The location's name, whether it be a town, village, or city occurred | Categorical |
| Country | Indicates the nation or region where the incident happened | Categorical |
| Gname | The identity of the attackers' organization, or the attack perpetrators' group | Textual |
| Weaptype1 | A description of the terrorist attack's weaponry | Categorical |
| Ransom | A demand for a monetary ransom may or may not have been made during the attack | Categorical |
| Summary | An overview of the incident that includes a brief narrative | Textual |
| Attack type | The strategies and methods used in terrorism attacks | Categorical |

## 4 Transformer-Based Model for Feature Extraction

This section details the textual feature extraction used in this study to learn more meaningful textual representations based on a Transformer model. The BERT model offers a generalized numerical representation of text that can be used for various tasks in NLP [67]. For instance, we use the DistilBERT model to extract the embedding of the text after fine-tuning the pre-trained model for several epochs. It is worth mentioning that we only fine-tune the top layer (sixth) of DistilBERT for a fast training process. Later, the extracted features are concatenated and used as input to a classifier to predict the attack type. DistilBERT uses the BERT Transformer architecture to distill the canonical model using knowledge transfer and produce a new model with a smaller size and similar performance to the original. Fig. 6 shows the distillation process, which relies on the BERT base model (BERT-base-uncased). The main differences between the original BERT model and the distilled model are:

(i) DistilBERT has fewer parameters than BERT base (less by 40%)

(ii) DistilBERT possesses a 60% inference speedup

(iii) DistilBERT uses dynamic masking rather than static masking during the inference phase

(iv) DistilBERT omits next sentence prediction (NSP) and segment embedding learning during the training phase

(v) DistilBERT uses six transformer layers (encoders) instead of 12 such in the BERT base

(vi) DistilBERT can be trained in 3.5 GPU days (instead of 12 GPU days)

Concerning the training data, DistilBERT shares the same datasets for training as the BERT base, which are the Toronto books corpus and English Wikipedia. As shown in Fig. 6, we replaced the classification layer in the DistilBERT model with two layers for feature extraction and attack classification, respectively. The model receives an input sentence $X$ represented as a sequence of tokens $X = x_1, \ldots, x_s$ and outputs a single semantic vector [CLS] of each sentence. Before processing the raw sentence having tokens, DistilBERT applies a sequence of sub-word tokenization and word-piece tokenization [68] to produce a set of embedding vectors named input embedding ($S1$). The tokenization maps each token to three different embeddings: word, segment, and positional. The tokenizer places special tokens at each sequence's beginning and end, namely [SEP] and [CLS], respectively. The [CLS] token helps the model understand that the sequence is being used for classification and allows it to learn the relevant patterns. The second special token is [SEP], which stands for "separator". It is used to separate multiple sequences when they are concatenated together. Later, each token's word, segment, and positional embedding are summed up using a multi-layered RNN, which uses a self-attention mechanism named the lexicon encoder to produce a single contextual vector $S2$. The $S3$ semantic embeddings are generated by concatenating all the contextual vectors $S2$ learned by the [CLS] token. The [CLS] token embeddings are fed to a fully-connected layer for feature extraction. The output of the feature extraction layer is a single vector $F$ of size $d = 128$, where $d$ is the number of neurons. This layer serves as a dimensionality reduction phase which maps the [CLS] token embedding of size 768 to a lower dimension equal to 128. The extracted features with dimension $d = 128$ are later combined with the key features and fed to the ML model for classification. The feature extraction layer uses the GELU [69] as a non-linear activation function to improve the model's performance, defined as in Eq. (1).

$$\mathrm{GELU}(m) = x * \Phi(m) \tag{1}$$

$m$ is the output of the fully-connected layer, and $\Phi(m)$ represents the Cumulative Distribution Function for Gaussian Distribution. The classification layer is used to fine-tune the model and update the pre-trained model weight matrix $W$ on the terrorism attack detection task represented as a multi-class classification problem. The classification layer is a fully-connected layer with $r = 2$ neurons representing the number of classes. The Soft max activation function defined in Eq. (2) maps the output of the classification layer to the probability distribution of $X$ being classified as class $c$ (i.e., the attack type). The classification layer is trained using the Cross-Entropy loss function.

$$P_r(c|X) = Softmax(W^T \cdot X) \tag{2}$$

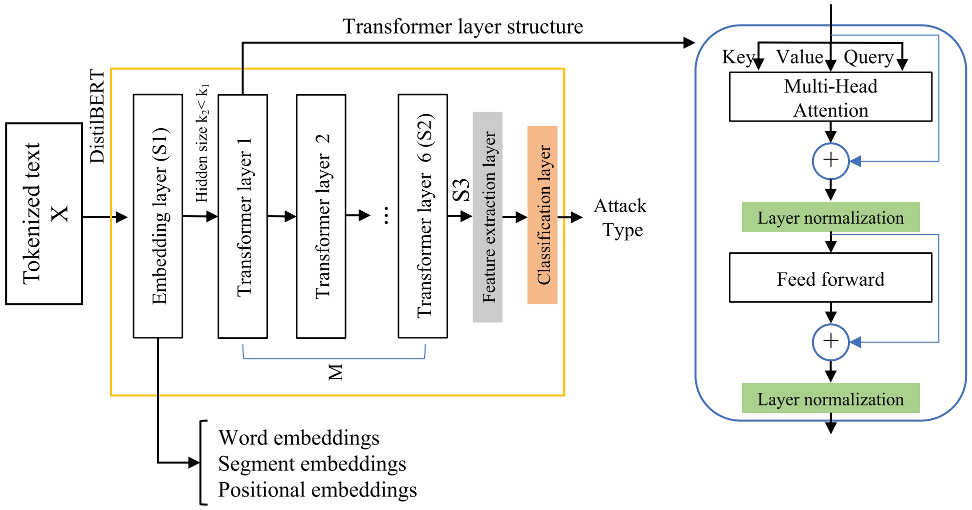


**Figure 6:** The proposed feature extraction model

## 5 Experiment

The experimental setup and results are summarized in this section. It starts by doing a series of tests on each set of features separately, and it ends by discussing the impact of combining two sets of features (textual features and key features). It also examines how merging text features from collected data affects the classifier's performance. The experiments are divided into three scenarios. The first experiment used the seven key features to determine the type of terrorist attack without considering the textual summary features in the proposed approach. The second experiment shows how the performance of the proposed framework changes when textual features extracted from summary features are used. Finally, experimental three show the enhancement of the performance of the classifiers based on the combination between the textual features and the other features. Due to the objective of determining the proposed framework's performance in various terrorism attack classes, the framework was evaluated using accuracy, $F_1$-score, precision, and recall measures. Additionally, we examined overall classification performance using average measures such as micro and macro averaging, which are widely accepted and frequently used in many multiclass classification studies [70].

## *5.1 Experiment Setup*

### *5.1.1 ML Models*

This study applied various ML models to determine the most appropriate classifier for a terrorism classification framework. It examined the probability of having an effect when using textual features with the most commonly used features in the state of the art, depending on the classifier used. To determine the most appropriate classifier for a given set of features, we must understand the properties of the available classifiers. This section describes the machine-learning classifiers used in this study.

(1) The K-Nearest Neighbors (KNN) fall under the category of supervised learning used for classification (the most common) and regression. It is a versatile algorithm to calculate missing quantities and reconfigure data sets [71,72].

(2) The Support Vector Machine (SVM) is a supervised learning algorithm that can be used in classification and regression problems. It is usually used in classification problems due to its effectiveness and excellent accuracy with most data [73].

(3) Decision Trees (DT) are a predictive modeling method used in statistics, data mining, and ML. This algorithm aims to create a model that predicts the value of the target variable by learning simple decision rules inferred from the data features.

(4) Logistic regression (LR) is a straightforward supervised learning algorithm. Compared to other algorithms, it does not require complex mathematical operations, making it suitable to classify data into separate classes [74].

(5) Linear discriminant analysis (LDA) is one of the linear classification algorithms. That creates a probabilistic model for each class based on the data distribution. Thus, new samples are classified by reference to the conditional probability of belonging to one of the classes [75].

(6) Additionally, various ensemble techniques are used. Ensemble methods in ML combine different learning algorithms so that each algorithm supports the other to strengthen the prediction process. A group of weak decision classifiers outperforms a single large decision classifier [76]. The RF uses the random method to generate samples for training and then generates a decision tree for each sample in the last step. The algorithm collects all the results from the decision tree to make a prediction based on the voting mechanism. RF can combine weak and strong variables and handle outliers. Besides, it is not affected by overfitting. A popular boosting algorithm, similar to a RF, Gradient Boosting (GB), is a collaboration method that combines several weak tree learners to create a significantly stronger learner. This algorithm works by adding prediction models sequentially. Each model corrects the previous model instead of adjusting the weights of examples at each step, as in AdaBoost.

The XGBoost Gradient Boosting Algorithm (XGB) is one of the most popular and widely used boosting algorithms because it is so powerful. It is similar to Gradient Boost but has some additional features that make it more powerful. Training is swift and can be balanced or distributed across groups [77]. Another gradient-boosting approach that employs a leaf-wise algorithm to generate trees vertically is the Light Gradient Boosting Machine (LGBM). LGBM uses an exclusive feature bundling approach to tackle dataset sparsity. It reduces the number of characteristics while maintaining the most informative ones by combining mutually incompatible features in a virtually lossless manner. LGBM also demonstrated proficiency in handling high-dimensional and imbalanced data [78]. A boosting algorithm is a descriptive group whose primary function is to reduce bias and variance in supervised learning. It is also considered a group of ML algorithms that transform weak learners into strong ones. The classification accuracy of the resulting robust classifier depends on all the weak classifiers. Adaboost can significantly improve learning accuracy, whether applied to synthetic or accurate data. The Bagging algorithm (Bag) is also known as Bootstrap clustering and can be used to solve classification and regression problems. Its idea is to assemble several predictions about our data and find the best results after collecting them. It is used if we want to reduce the contrast. In addition, filling algorithms improve the accuracy of the model. An ensemble ML model called Extra Trees (ETs) aggregates the forecasts from numerous decision trees. It has a connection to the standard RF algorithm [79].

(7) The GaussianNB (NB) model has a consistent classification efficiency derived from the Bayes theorem. It performs effectively with modest amounts of data, can handle several classification tasks, and can be trained significantly further when data exceeds the memory. The NB model should have the lowest mistake rate compared to other classification techniques [80].

### *5.1.2 Performance Evaluation*

Numerous models and applications in various spheres have used classification techniques. For comparing various learning algorithms, such metrics' analysis and significance must be interpreted correctly for evaluating different learning algorithms. Some of these measurements use graphical techniques, but most use scalar metrics. The dataset is observed to be unbalanced; therefore, the classification system's precision cannot only be used as a model performance metric. AUC ROC can be used as an additional metric. In addition, various metrics were used to evaluate the multiclass classification performance of the proposed model. The preliminary metrics are accuracy, recall, precision, and $F_1$-score. To extend our metrics to multiclass classification, micro-average, macro-average, and ROC AUC were also calculated [81]. The area under the receiver operating characteristic curve (AUC-ROC) has been utilized in the literature to evaluate the model's performance. The F1 measure is widely employed in most ML application domains for binary and multiclass scenarios. In multiclass cases, researchers can use the F1 micro/macro averaging methods [82].

Various benchmarking studies have also recommended this metric's application. Additionally, the literature suggests using the area under the precision-recall curve (AUC-PR) as a model evaluation metric. This study examines whether the AUC-PR curve provides specific information about model performance. This study ranks the existing models based on AUC-ROC, AUC-PR, and other evaluation metrics to accomplish this. Using AUC-PR to evaluate the models helps avoid the extra cost, time, and effort of testing functional modules. Finally, seven evaluation metrics are used to evaluate the performance of the proposed framework, including the $Accuracy_M$, $Precision_M$, $Recall_M$, $F - score_M$, $Macro - average_M$, $Micro - average_M$, and $ROC - AUC_M$, (3)–(6), respectively.

$$Accuracy = \frac{TP + TN}{TP + FN + TN + TP} \tag{3}$$

$$Precision = \frac{TP}{TP + FP} \tag{4}$$

$$Recall = \frac{TP}{TP + FN} \tag{5}$$

$$F1 - score = \frac{2 * Precision \times Recall}{Precision + Recall} \tag{6}$$

where TP is True Positives, TN is True Negatives, FP is False Positives, FN is False Negatives, TP is True Positives, TN is True Negatives, FP is False Positives, and FN is False Negatives. The receiver operating characteristic (ROC) curve is frequently used to evaluate the predictive ability of the current method across the entire range of algorithm decision value [83]. The ROC demonstrates the relationship between the true positive rate (TPR) and the false positive rate (FPR). The area under the ROC curve, or AUC, has been used to evaluate the model's performance.

## 6 Experimental Results

Many tests were executed to compare various ML techniques and analyze the impact of adding script features on performance. Several well-known classifiers, including Naive Bayes, SVM, RF (with 100 random trees), and other ML models, are tested as part of the ML technique to determine the impact of the chosen features. The same dataset is used for training and testing the classifiers.

### *6.1 Experiment Series 1. Main Features Used on The State of Art Result*

Table 4 summarizes the first experiment's outcome, which used only the essential features (standard features used in state-of-the-art). In the multi-classification of terrorism attacks, models showed approximately similar accuracy, precision, recall, and $F_1$-score. It was noted that LGBM achieved the highest rating report values across all scales. The report on the classification test shows that learning models like LGBM and RF perform better than other models. Table 4 shows that the performance of the XGB is closely similar to the SVC and ExtraTrees models, which gave an average of 80%. It is also clear from Table 4 that models such as Bag, KNN, LinearSVC, LDA, SGD, and LR produced very similar results. With a relative difference in the time taken to build the model due to the nature of each model and its characteristics in learning and training. The AdaBoost algorithm showed the lowest performance results. Overall, performance value is more consistent than other models across all metrics. Comparing all models, AdaBoost has the lowest value of 0.62%, closely followed by Extra Tree with a value of 0.67% LGBM outperformed all other models with a value of 0.81%, 0.81%, 0.79%, 0.78% in accuracy, precision, recall, and $F_1$-score, respectively. AdaBoost had the lowest accuracy, precision, recall, and $F_1$-score values of 0.63%, 0.61%, 0.63%, and 0.63%, respectively. Therefore, it is the lowest-performing model. The model performs well and ranks attacks based on critical features well, with an overall average accuracy of over 0.75%, very close to the results of studies that dealt with the same topic and used the same features for training. The performance of the classifiers can be improved if the features are enhanced by those taken from the text, taking into account the time and complexity of the proposed model.

**Table 4:** Model performance with main features

| Model name | ACC | Precision | Recall | $F_1$-score |
|---|---|---|---|---|
| LGBM | 0.81 | 0.81 | 0.79 | 0.78 |
| RandomForest | 0.81 | 0.80 | 0.79 | 0.78 |
| SVC | 0.80 | 0.80 | 0.79 | 0.75 |
| XGB | 0.80 | 0.79 | 0.78 | 0.78 |
| ExtraTrees | 0.80 | 0.79 | 0.77 | 0.77 |
| Bagging | 0.79 | 0.78 | 0.77 | 0.77 |
| KNeighbors | 0.78 | 0.78 | 0.77 | 0.75 |
| LinearSVC | 0.77 | 0.76 | 0.76 | 0.69 |
| LinearDiscriminantAnalysis | 0.77 | 0.76 | 0.73 | 0.72 |
| LogisticRegression | 0.76 | 0.74 | 0.74 | 0.70 |
| GaussianNB | 0.74 | 0.74 | 0.74 | 0.70 |
| QuadraticDiscriminantAnalysis | 0.74 | 0.73 | 0.72 | 0.70 |
| DecisionTree | 0.72 | 0.70 | 0.72 | 0.72 |
| ExtraTree | 0.67 | 0.67 | 0.67 | 0.66 |
| AdaBoost | 0.63 | 0.61 | 0.63 | 0.63 |

### *6.2 Experimental Series 2. Textual Features with Target*

The second experiment involved applying various classifiers to the features collected from the summary text and comparing their outcomes. Table 5 includes the other performance measurements. We wanted to demonstrate through this experiment that terrorist acts might be classified using the traits taken from the text. This experiment can also show that the features extracted from the text represent the context of the summaries, which can improve the performance of the classifier and help detect the semantic relationship between input and target classes. Thus, instead of relying on raw text or sparse text representations as the input of the classification model. The extracted text features using fine-tuned DistilBERT can rely on the transformer architecture, which employs an attention mechanism to encode semantic features and learn excellent text representation from the context. To monitor and track our ML experimental status, test, and performance of the models, algorithms with similar accuracy rates were again compared using other metrics like accuracy, recall, and $F_1$-score. Based on the accuracy, precision, recall, and $F_1$-score values, the XGB and LGBM perform best among other classifiers, followed by the Bagging, RF, and ExtraTree classifiers. Classifiers such as Bag, LR, DT, RF, GB, and LDA perform exceptionally well, correctly classifying more than 0.75% of attack types. Intriguingly, the best experimental results presented by Bag for various experiments are approximately in line with RandomForest and ExtraTrees. While KNN performed well and remained consistent across all experiments, the key aim was to demonstrate that ensemble modeling can result in more accurate predictions. This is due to the model's design, which makes it simpler to train than models of other types and yields superior outcomes with fewer data. Additionally, group models comprise several poor learners-models that are bad in and of themselves. However, when used together, they compensate for each other's shortcomings and outperform when used separately. As seen from Table 5, SVC and LinearSVC perform better than other classifiers such as LogisticRegression, KNN, QuadraticDiscriminantAnalysis, LinearDiscriminantAnalysis, GaussianNB, DecisionTree ExtraTree, and AdaBoost. In terms of recall and measure accuracy, we attribute this to the widespread use of SVC for text classification; using linear kernels for text classification is recommended, as linear kernels work well when there are many features. Hence, linear SVC is used in experiments as well. It is also intriguing that less than 0.60% of terrorist attack types are accurately classified by AdaBoost, which has the lowest accuracy. The algorithms can train well and attain high accuracy, precision, recall, and F1-scores using text features derived from the attributable summary, as evidenced by the overall accuracy performance reported in Table 5.

**Table 5:** Model performance with extracted textual features only

| Model name | ACC | Precision | Recall | $F_1$-score |
|---|---|---|---|---|
| XGB | 0.886 | 0.886 | 0.879 | 0.879 |
| LGBM | 0.881 | 0.881 | 0.880 | 0.880 |
| Bagging | 0.799 | 0.797 | 0.798 | 0.796 |
| RandomForest | 0.788 | 0.789 | 0.786 | 0.786 |
| ExtraTrees | 0.780 | 0.780 | 0.782 | 0.782 |
| SVC | 0.778 | 0.779 | 0.772 | 0.773 |
| LinearSVC | 0.752 | 0.752 | 0.749 | 0.749 |
| LogisticRegression | 0.750 | 0.751 | 0.752 | 0.753 |
| KNeighbors | 0.740 | 0.740 | 0.743 | 0.743 |
| QuadraticDiscriminantAnalysis | 0.726 | 0.729 | 0.728 | 0.725 |
| LinearDiscriminantAnalysis | 0.724 | 0.726 | 0.728 | 0.727 |
| GaussianNB | 0.722 | 0.725 | 0.723 | 0.722 |
| DecisionTree | 0.717 | 0.716 | 0.716 | 0.719 |
| ExtraTree | 0.705 | 0.707 | 0.704 | 0.705 |
| AdaBoost | 0.702 | 0.702 | 0.706 | 0.706 |

### *6.3 Experimental Series 3. The Impact of Text Features on the Framework's Performance*

Tables 6 and 7 show how well the proposed framework for hybrid features works by comparing different performance measurements. Table 6 shows the performance measures for the hybrid features, focusing on the essential measures: accuracy, recall, precision, and $F_1$-score. Furthermore, multi-class classification metrics can be calculated by taking the average performance of each class, called macro averaging, or the average performance of all the classes, called micro averaging [84] as shown in Table 7. Experimental Series 3 shows the performance of classifiers with additional measurements such as micro-average ROC, macro-average ROC, and micro-average precision. Micro-averaged precision and micro-averaged recall are both equal to the accuracy when each data point is precisely assigned to precisely one class. The micro-averaged metrics differ from the overall accuracy when the classifications are multi-labeled or when some classes are excluded in the multi-class case. In additional micro-averaging, each class counts the same for the average, as larger classes dominate the measure; in macro-averaging, the average for each class is determined. Only then does each class count the same for the final average. This difference is significant when the collection is skewed, which is the case with terrorist attacks since, in a dataset, some attacks are expected to happen much more often than others. Figs. 7–16 also show performance measures of classifiers with micro-average, macro-average, and micro-averaged precision metrics.

**Table 6:** Model performance with features combination

| Model name | ACC | Precision | Recall | $F_1$-score |
|---|---|---|---|---|
| XGB | 0.987 | 0.981 | 0.977 | 0.979 |
| LGBM | 0.965 | 0.969 | 0.962 | 0.964 |
| Bagging | 0.953 | 0.955 | 0.954 | 0.953 |
| RandomForest | 0.944 | 0.944 | 0.943 | 0.945 |
| ExtraTrees | 0.929 | 0.922 | 0.921 | 0.928 |
| SVC | 0.918 | 0.925 | 0.923 | 0.932 |
| LinearSVC | 0.914 | 0.901 | 0.902 | 0.905 |
| LogisticRegression | 0.911 | 0.905 | 0.908 | 0.909 |
| KNeighbors | 0.903 | 0.906 | 0.905 | 0.903 |
| QuadraticDiscriminantAnalysis | 0.896 | 0.899 | 0.898 | 0.895 |
| LinearDiscriminantAnalysis | 0.894 | 0.899 | 0.898 | 0.897 |
| GaussianNB | 0.892 | 0.895 | 0.893 | 0.892 |
| DecisionTree | 0.887 | 0.886 | 0.886 | 0.899 |
| ExtraTree | 0.865 | 0.869 | 0.864 | 0.865 |
| AdaBoost | 0.776 | 0.767 | 0.778 | 0.745 |

**Table 7:** Model performance with features combination

| Model name | Micro-average ROC | Macro-average ROC | Micro-average Pre-Reca |
|---|---|---|---|
| LGBM | 0.995 | 0.996 | 0.975 |
| XGB | 0.996 | 0.984 | 0.975 |
| ExtraTrees | 0.994 | 0.974 | 0.963 |
| Bagging | 0.994 | 0.982 | 0.971 |
| RandomForest | 0.993 | 0.982 | 0.964 |
| ExtraTree | 0.983 | 0.970 | 0.974 |
| SVC | 0.972 | 0.956 | 0.943 |
| LSVC | 0.953 | 0.946 | 0.933 |
| LogisticRegression | 0.992 | 0.986 | 0.964 |
| KNeighbors | 0.931 | 0.936 | 0.935 |
| QuadraticDiscriminantAnalysis | 0.887 | 0.890 | 0.886 |
| LinearDiscriminantAnalysis | 0.874 | 0.863 | 0.883 |
| GaussianNB | 0.843 | 0.845 | 0.836 |
| DecisionTree | 0.805 | 0.814 | 0.814 |
| AdaBoost | 0.782 | 0.791 | 0.791 |

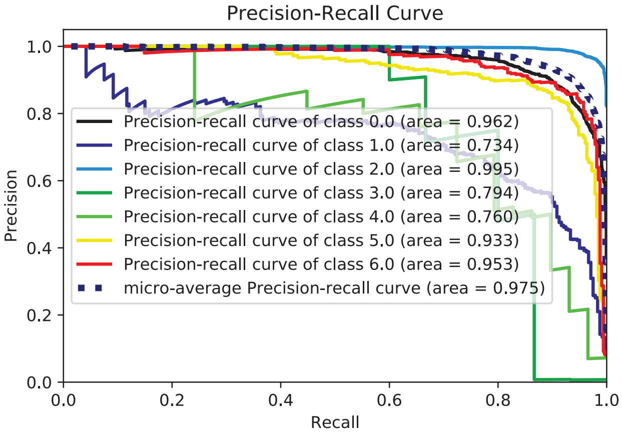


**Figure 7:** Precision-recall-curves for XGB

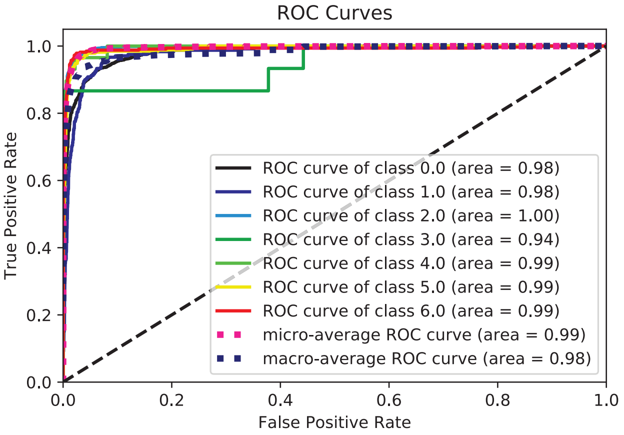


**Figure 8:** ROC curve for XGB

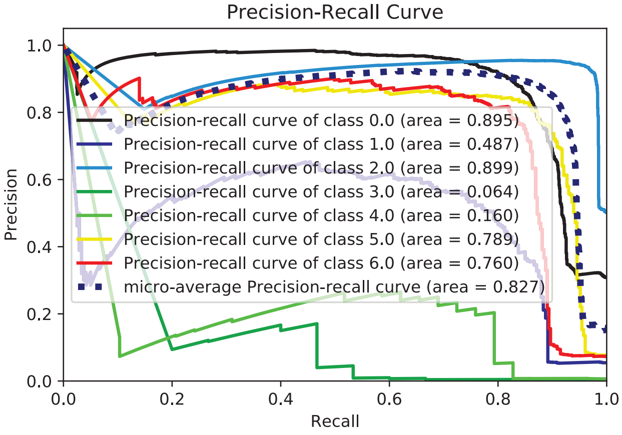


**Figure 9:** Precision-recall-curves for LGBM

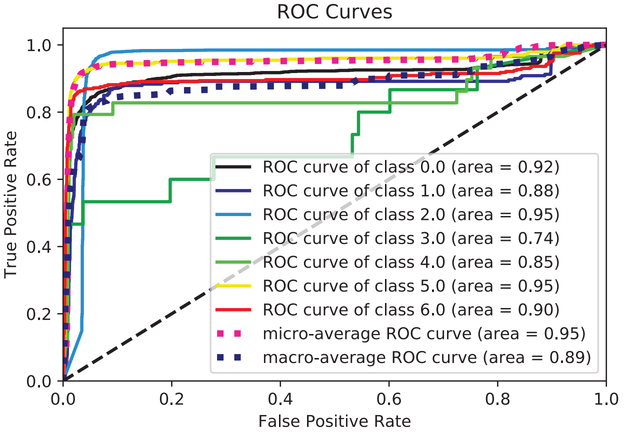


**Figure 10:** ROC curve for LGBM

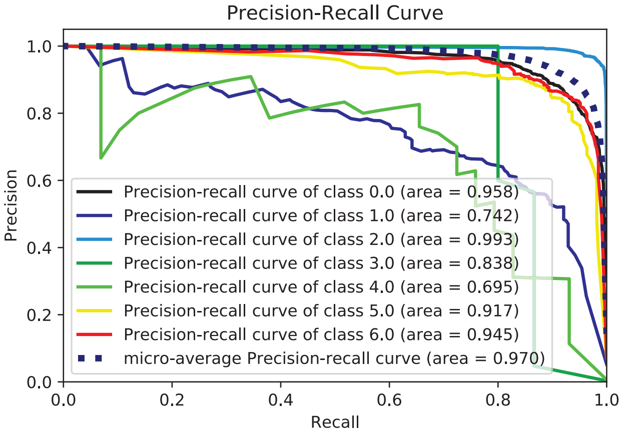


**Figure 11:** Precision-recall-curves for bagging

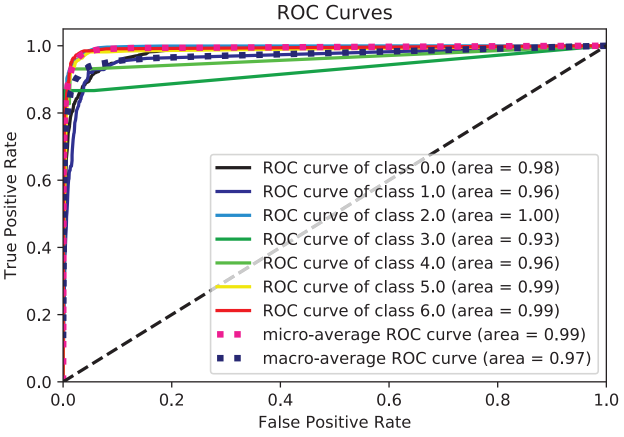


**Figure 12:** ROC curve for bagging

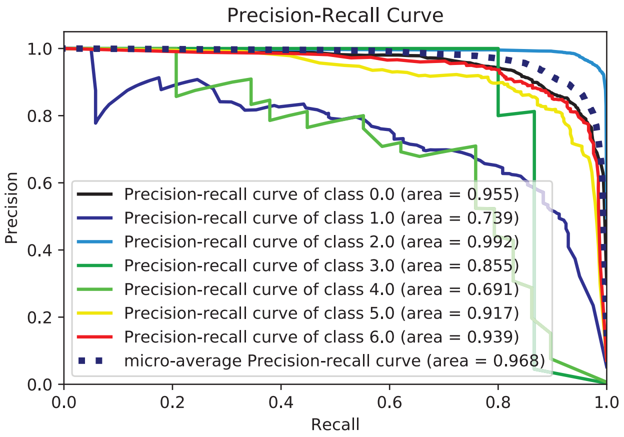


**Figure 13:** Precision-recall-curves for RF

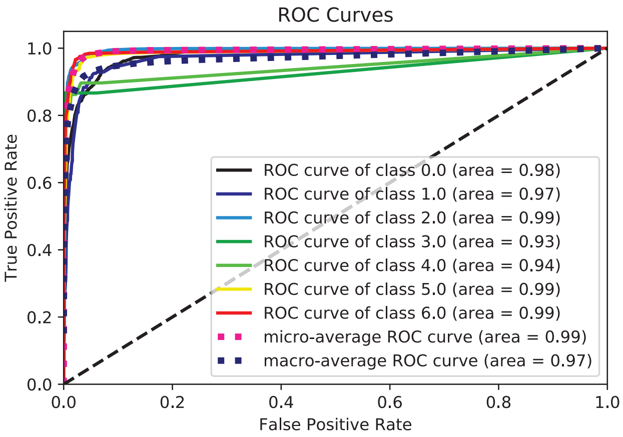


**Figure 14:** ROC curve for RF

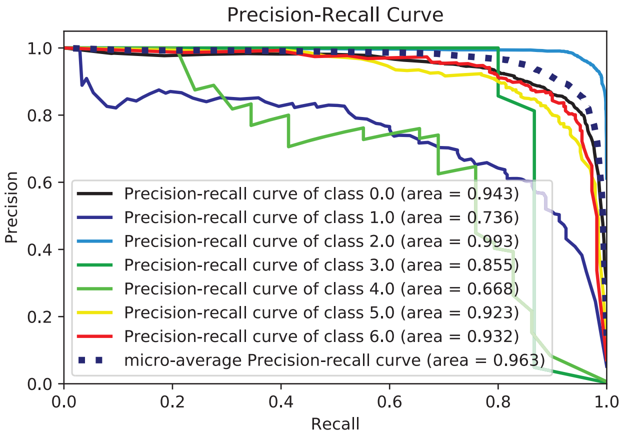


**Figure 15:** Precision-recall-curves for ETrees

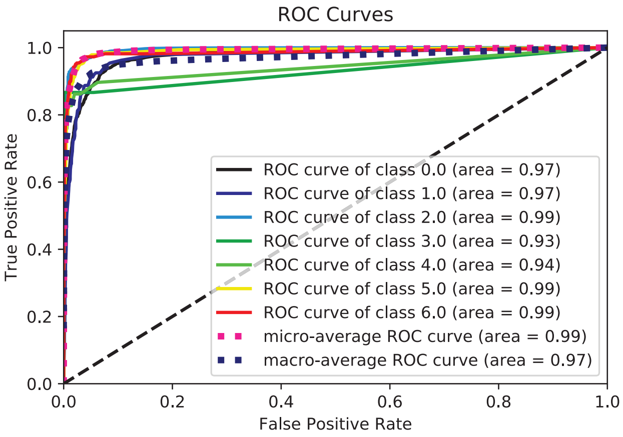


**Figure 16:** ROC curve for ETrees

From these tables and figures, the variance of performance between different classifiers can be seen. Table 6 shows the experimental results of the proposed model based on the generated data set by reviewing the results recorded by different algorithms with a focus on accuracy and recall measures. Experimental three agree that adding text description information (summary features from GTD) is a straightforward way to improve the ability of classifiers to classify terrorist attacks. Both accuracy and recall scores are over 0.98% for some metrics, and the models used have improved performance for many other metrics. Hence the effect of the features included in most models, as they provide outstanding performance in their entirety, which is superior to models that use particular features. This indicates the effect of the built-in features on the effectiveness of the models in this task. More specifically, of all the models, Adaboost gives the comparatively fewest results, with an accuracy of 0.776% based on mixed features. The rest of the measures scored 0.767%, 0.778%, and 0.745%, recall, precision, and $F_1$-score, respectively. However, let us compare the performance of the Adaboost model (in Table 6) with the combined features. We find an improvement compared to its performance with the different features in the previous two experiments in Tables 4 and 5.

We attribute this to the fact that AdaBoost performs excellently on classification problems. This is a common belief but cannot produce accurate probability estimates. It makes a correct classification but could solve problems with more than two classes better. The XGB scored best among the models, with an accuracy of 0.98% based on the features included in all models. The figures show the ROC curves for the top 5 classifiers in the test set. We plot each algorithm's partial mean ROC curve, the total mean ROC curve, and any two types of ROC curves. According to Figs. 7 and 8, it can also be seen that the AUCs of XGBoost defining classes of 0, 2, 5, and 6 types of a terrorist attack are 0.96%, 0.99%, 0.91%, and 0.93%, respectively. The XGBoost model's performance with the experiment's hybrid features can be compared with the previous experiments, and performance improvement can be observed. In Tables 4 and 5, the overall accuracy of the XGBoost model increased from 0.81% and 0.88% in Experimental Series 1 and 2 to more than 0.98% in experimental series 3 as shown in Table 6. The main reason is that XGB is a reinforced model with a clustering mechanism and many descriptive classifiers. This makes the model more stable and accurate. The LGBM classifier also fared closely with the XGB model. This is evident from the results recorded in both Tables 6 and 7, as shown in Figs. 7–10.

Two potential reasons for this behavior could be accounted for. Firstly, tree-based methods are deterministic. Thus, when encountered with structured data (as in this study), tree-based algorithms can explicitly fit the hyperparameters to the input feature, a natural extension of their workflow. Another potential reason could be that LGBM and XGB are ensembles of decision trees, and their predictions are a compilation of the predictions of many decision trees into a single one. XGBoost and LGBM based on Gradient Boosted Decision Trees (GBDTs) achieved great success in our experiment. Both algorithms work similarly in model performance, but training on LGBM happens within a fraction of the time required by XGBoost. LGBM's speed training makes it a preferred choice. XGBoost requires many resources to train on large amounts of data, making it an excellent choice, while LGBM is lightweight and can be used on modest hardware. LGBM allows passing feature names to be treated as classes and handles this issue easily by dividing by equals. The Bagging, RandomForest, and ExtraTrees were reasonably close, with the relative superiority of the Bagging classifier, with an accuracy of more than 0.96% compared to the rest of the classifiers, as shown in both tables. This superiority also seems clear if we look at Figs. 11–16. The Bagging classifier can distinguish between attack types, especially classes 2, 6, and 5, with an average of more than 0.92%, as shown in Figs. 11 and 12. Moreover, the SVC, LinearSVC, and LogisticRegression classifiers showed relatively average performance and somewhat good power when compared with their peers in the previous experiments (Series 1 and Series 2), with the SVC classifier showing better performance, scoring 0.91%, 0.92%, 0.92%, and 0.93% for accuracy, precision, recall, and $F_1$-score, respectively.

Based on the results, the QDA showed better performance when compared to the LDA, ExtraTree, and NB, as it had higher discrimination and classification abilities. The performance parameters of this model were 0.896%, 0.899%, 0.898%, and 0.895% for accuracy, recall, precision, and $F_1$-score, respectively. By combining features, most models achieve over 0.86% accuracy, much higher than separate models. This indicates that the models have a solid ability to learn the built-in traits thoroughly. It can also be observed from the literature review and our previous work that the text feature and the other separate feature achieve a slightly lower degree of textual features than the literature suggests. This indicates that the two attention units play a role in improving performance. The above analysis demonstrates the effectiveness of the proposed model. Based on the above analysis, all models can perform better by combining text and numeric features rather than text-only or numeric features. This is because different types of features can make their contributions. Meanwhile, the results for text-only features are better than those for numerical features. The main reason is that the way the text describes the attack gives vital clues about the type of attack. In conclusion, adding text-based features to other feature types would enhance the performance of the present prediction system. The results tables show that models using hybrid feature information performed better throughout the trial. Notably, a conservative model using only the textual features shows good performance approximately with comprehensive effectiveness with the many classifiers. The emphasis throughout this study was nearly entirely on the text. Figs. 11–16 provide a comprehensive overview of the out-performance classifiers of ML that show excellent performance via a side-by-side comparison. In actuality, the text (abstract) is regarded as the primary source of information. However, the “request” to abandon category- and numerically-based trait predictions instead of text-based predictions alone was ineffective. Contrarily, it is thought that combining text with various data sources might enhance prediction overall. According to the assessment’s findings, categorical and numerical data and textual features provide valuable information for forecasting different sorts of terrorist strikes. As a result, text-based, category-based, and digital classifiers each have unique advantages and disadvantages. The framework’s performance is firmly considered to be improved by combining text elements with important aspects in classifiers.

### *6.4 Discussion*

The purpose of this study is to explore the role of textual features in the classification of terrorist attacks in the harsh dataset. Our findings show that combining text and other information makes the model classification result more accurate. This indicates that the text in the standard GTD database contains data that enhances classifiers’ ability to learn and discover types of attacks. By combining powerful features extracted from the text, the performance of classifiers can be more accurate. The results showed that these features are related to classification accuracy. These features reflect other hidden aspects of attack intelligence related to defining attack categories. This indicates that the content of texts can be enriched with other external information about the event that will enrich the textual content, as reflected in the extracted features and results. From the accuracy change trend graph in Fig. 17, it can be seen that the model’s accuracy shows a gradual increase after textual features are added. The model combining text features (summary features) and other features in a classification scenario has a better classification effect than other scenarios. The model also outperforms other models in the data set by increasing textual sequences, indicating that integrating textual feature learning and multilevel feature sets with weighted features is more effective for classification. We have reached a maximum accuracy of 98.7%. Through careful data analysis and selecting appropriate algorithms to train our models, we test multiple classifiers to predict the type of attack correctly. We also thoroughly evaluated our models with multiple evaluation scales.

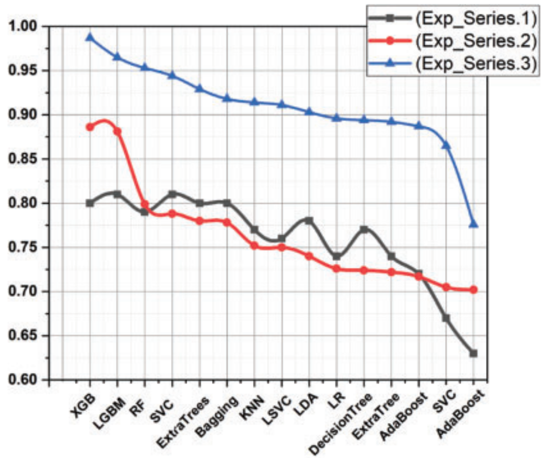


**Figure 17:** The performance of the classifiers with various experimental series

### *6.4.1 Analysis of Result with BERT*

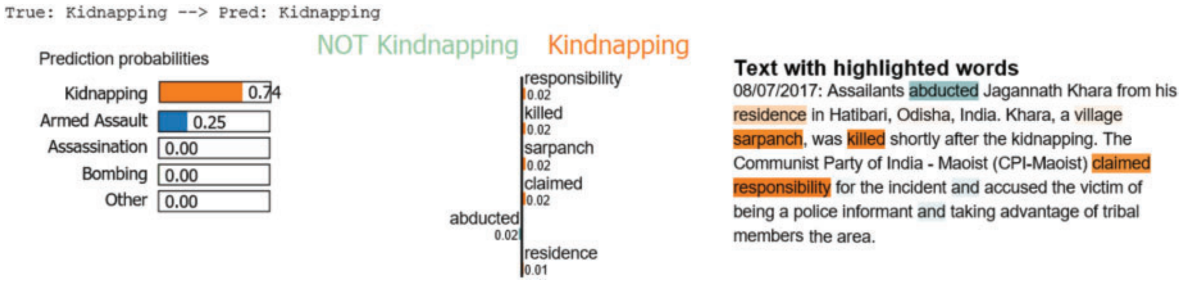
We first compare the two input methods of the attributes in the first trial. All the features included in Bert are compared with the target; in the third experiment, the features extracted from the text are combined with other features. It is clear from Tables 5 and 6 that both are much better than using single features set in Experiment 1, as shown in Table 4. It is also evident that entering text and features as separate sequences into classifiers affects the overall performance of the classifiers. As shown in Table 5, the model can learn to represent each sentence string through fine-tuning; however, it cannot effectively associate any feature information with the attack labels. In other words, because the text and other features are entered as separate parts, as shown in Tables 4 and 5, they may need more information, which is the main reason for poor performance. We need to enter the textual features and the other features combined. Meanwhile, it can be shown from Table 6 that the different features combined with the extracted text features contribute to detecting attack types. The built-in feature has the most improvement, and its accuracy is higher than 8% compared to other experiments. This shows that the built-in features can effectively improve attack detection accuracy. Tables 6 and 7 demonstrate how much better BERT performs than other models. This increases accuracy, demonstrating the value of BERT. Additionally, it demonstrates how the self-attention mechanism can better grasp sequence semantics. When the model analyzes a single sentence, self-attention enables each word to look at other words to understand which word contributes better. In other words, self-attention means the phrase will consider itself while deciding how to represent each symbol. This is how each word is effectively understood by BERT, depending on its context in the background (the sentence). One word can have various meanings depending on the context (how it is used in a sentence), and self-attention can encode (understand) each word based on the words used. The model can directly learn the association between the target text sequence and the related label based on confirming the model's task realization, which leads to a simplified training process. It is evident from the previous tests that the classification model, based on a pre-trained model with the integration of numerous features, is a solution to the issues of inadequate feature extraction and information integration in the classification domain. Together with the other tiers of characteristics represented, the features derived by the BERT model give the classifiers additional assistance. Table 6 clearly shows the significant improvement in outcomes. We credit this to a more comprehensive representation of the features' semantic data. Moreover, attention methods are used to create weighted local feature vectors, which lessen the impact of many features. It is possible to try additional attention mechanism optimization when optimizing the attention mechanism for other feature. Nevertheless, during the extraction of local features from the text, some information about the original characteristics is frequently lost in the process. Along with other multi-level vital features, the complete contextual semantic features acquired by the BERT model can enhance feature information while ensuring that other features do not negatively affect the overall performance of the feature. The experimental results demonstrate that the proposed model performs satisfactorily on a data set and performs much better in classification, effectiveness and accuracy when compared to existing models that rely on categorical and numerical touches. However, the primary and most significant challenge in this subject is integrating and acquiring information due to the complexity of text content and the difficulty of encoding textual information. In the future, we will keep integrating DL models into text tasks and explore methods for improving text feature representation and optimizing semantic information for tex tasks. Represent text feature information using pre-trained models with attention mechanisms for text semantic representation, combine external semantic information and other techniques to provide a more accurate representation of the features. In other words, the tiny text cannot adequately capture and convey the terrorist incident. Enhancing the events conveyed in brief texts using textual elements is feasible.

### *6.5 Model Interpretation*

Explaining or interpreting the predictions of ML models has become paramount nowadays because we work with complex models that deal with unstructured data types such as images, audio, text, etc. To validate our result and identify the model's behavior, we used the Local Interpretable Model-Agnostic Explanations (LIME). Below, we have created a function that inputs a single sample from raw test data and then returns a prediction of terrorism types which is one of the terrorist attack types target. To validate the effect of the textual features, we used the transformed text using the BERT and the other features as a part of the function, then returned probabilities for the XGB model. We used this function when creating an explanation for a random sample of test data. We need to perform this step because our model works on pre-processed data; hence we need to write a function that uses text data for the model. The top features (words in this case) that result in a prediction of class 1 with probability values and refer to other classes with other probability values are highlighted in different colors. We mention that one of the inputs for Lime-Text-Explainer is char level, a Boolean indicating whether or not we should treat each character as a separate occurrence in the string. As the default is False, we do not consider each character separately. Tokenization and word indexing in a text instance are done using the Indexed-String function; otherwise, the Indexed-Characters function is used. LIME begins by generating some altered samples near the desired data point. Some words are removed randomly from the instance to construct disturbed samples for text data. As a default measure, the distance between the original and perturbed samples is calculated using the cosine distance. Then, the local prediction made by the explanation model is represented by the prediction probabilities given in the leftmost. The features and values supplied in the middle represent the relevant features and their coefficients. As for this specific data instance, we are choosing the top crucial features; We validated two scenarios to find out the importance of textual features and their ability to help models in the learning process as follows:

(i) The validation case 1

We take a random sample of test data from a category mis-classified in the first experiment that does not consider textual data as input. We use this sample as input for the test in the two experiments, Experimental Series 2 and Experimental Series 3, to examine the effect of the textual features. Fig. 18 shows that the model that takes the textual features could correctly classify the random sample, which has been falsely classified. We can see from the figure that the features (words in this case) highlighted with orange are the top features that cause a prediction of kidnapping class with a probability near 0.75 and a probability of 0.25 for other classes.



08/07/2017: Assailants abducted Jagannath Khara from his residence in Hatibari, Odisha, India. Khara, a village sarpanch, was killed shortly after the kidnapping. The Communist Party of India - Maoist (CPI-Maoist) claimed responsibility for the incident and accused the victim of being a police informant and taking advantage of tribal members the area.

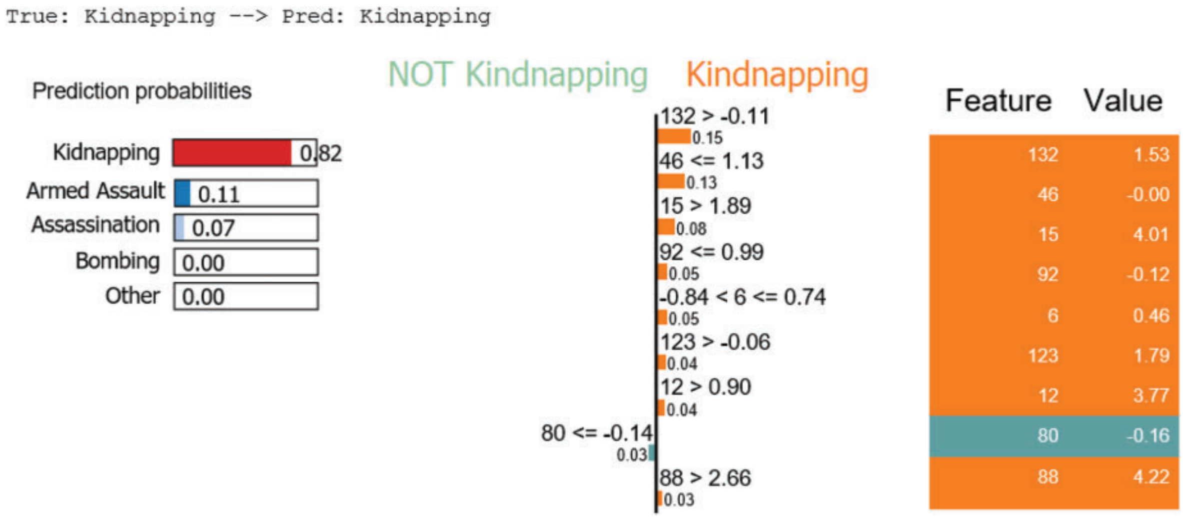
**Figure 18:** LIME plot for textual features

(ii) The validation case 2

The exact random sampling is used with the model that depends on the combined features; Fig. 19 shows that the model that takes the combined features had a more significant positive effect on performance. (Here, it should be noted that the textual features fall between numbers 8 to 133) when converted into representative vectors. We can see from Fig. 19 that the probability increased to 0.82 for the kidnapping class.

(iii) The model failure case 3

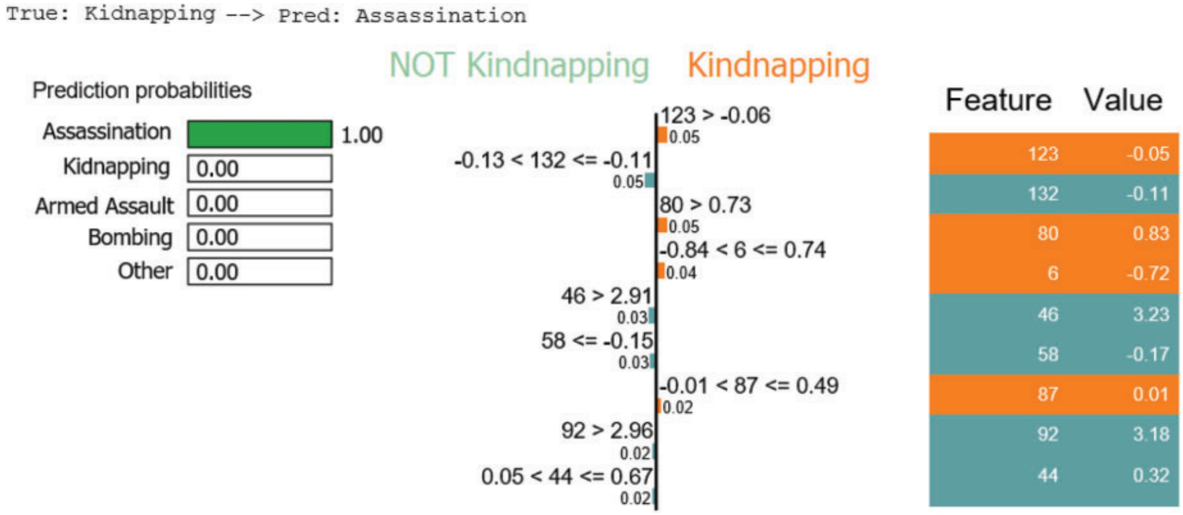
Identifying the related features from a data set and removing the irrelevant or less important ones is still a problem facing most ML and DL models. Irrelevant or partially relevant features can negatively impact model performance. Fig. 20 explains this case; we have selected the same sample example from test data, but this time with the wrong prediction. Fig. 20 highlights features contributing to the incorrect prediction, explaining the model failure case. However, this can give us more confidence in the model's performance.



```
True: Kidnapping --> Pred: Kidnapping
```

| Feature | Value |
|---|---|
| 132 | 1.53 |
| 46 | -0.00 |
| 15 | 4.01 |
| 92 | -0.12 |
| 6 | 0.46 |
| 123 | 1.79 |
| 12 | 3.77 |
| 80 | -0.16 |
| 88 | 4.22 |

**Figure 19:** LIME plot for a feature combination



```
True: Kidnapping --> Pred: Assassination
```

| Feature | Value |
|---|---|
| 123 | -0.05 |
| 132 | -0.11 |
| 80 | 0.83 |
| 6 | -0.72 |
| 46 | 3.23 |
| 58 | -0.17 |
| 87 | 0.01 |
| 92 | 3.18 |
| 44 | 0.32 |

**Figure 20:** LIME plot for model failure case

### *6.6 Comparative Analysis with Recent Techniques*

To make a more comprehensive judgment, the relevant background studies should be examined during the development of the proposed framework. The performance of the proposed framework was compared with that of similar approaches such as [5,30,32,37,85–88] to confirm its effectiveness. These studies were selected for comparison with this study under examination because they provided high-level performance results on GTD data, as described in the Related Works section. For fair comparisons, the performance of the proposed framework was compared with the latest technology using the same reference data set and standard features as those used in this research before the textual features were introduced. The accuracy performance results for the relevant investigations and the proposed framework are presented in Table 8, which attests to the latter's effectiveness. The column named best experimental result in Table 8; refers to the studies shown to have high-level performance outcomes on GTD data. Here, we seek to compare the results recorded in our study with other results. However, in different situations, the overlap factors differ. Therefore, testing our method in this study in other contexts would be interesting. We encourage other researchers to use textual traits as a framework and to incorporate different features to improve classification accuracy.

**Table 8:** State of art model performance

| Ref. | Technical methods | Classification target | Data size | Model evaluation |
|---|---|---|---|---|
| [88] | TF-IDF Feature Extraction, grid-search method and ML | Terrorist attacks | 102,669 records | Accuracy 0.88<br>Precision 0.87<br>Recall 0.88<br>F-1 0.87 |
| [37] | Using GA genetic algorithms to improve the hybrid classifiers, which are then trained to predict future attacks | Terrorism attack | 45221 records | Accuracy 0.94<br>Precision Non<br>Recall Non<br>F-1 Non |
| [5] | Using ML and DL for prediction of future terrorist activities | Terrorism attacks | 315,104 instances | Accuracy 0.94<br>Precision 0.94<br>Recall 0.94<br>F-1 0.94 |
| [32] | Using GCN multilayer deep neural network (NNGCN) model. topological graphs' spatial features are extracted using GCN. Its fundamental concept is based on the Laplacian matrix spectral decomposition | Early warning | No. nodes 3210<br>No. edge 22087 | Accuracy 0.96<br>Precision Non<br>Recall Non<br>F-1 Non |

(Continued)

**Table 8 (continued)**

| Ref. | Technical methods | Classification target | Data size | Model evaluation |
|---|---|---|---|---|
| [85] | Using four ML algorithms | Terrorist organization<br>Terrorist attack target | 641 records | Accuracy 0.73<br>Precision Non<br>Recall Non<br>F-1 Non |
| [86] | Features selection techniques and ML | Success attack<br>Terrorist organization | Not mentioned | Accuracy 0.92<br>Precision 0.90<br>Recall 0.91<br>F-1 0.90 |
| [87] | Dimension Reduction, Neo4j Sandbox and ML | Not mentioned | Terror activities | Accuracy 0.90<br>Precision 0.93<br>Recall 0.94<br>F-1 0.93 |
| [30] | Several methods for text feature extraction, representation, and classification have been used | Terrorism attack | 47053 records | Accuracy 0.95<br>Precision 0.94<br>Recall 0.94<br>F-1 0.95 |
| **Proposed Model** | Feature extraction, representation, with classification techniques | Terrorism attack | 40720 records | Accuracy 0.98<br>Precision 0.98<br>Recall 0.97<br>F-1 0.97 |

## 7 Conclusion and Future Work

Technological advances are changing how conflicts evolve. Advances in AI and ML will play a vital role in this transformation, as they will change the nature of the threat to state and non-state actors. AI enhances the fight against electronic, physical, and biological attacks, making it more accurate in identifying the target and more challenging to discover the perpetrator simultaneously. Terrorist attacks are among the causes of instability in countries around the world.

A clear understanding of how this event occurred will help us conduct more in-depth investigations. This study investigated the effect of incorporating text features extracted from the summary trait with significant features on a GTD dataset. To determine whether adding features to the extracted script can improve the classification of the proposed terrorist attack classification framework. Over a dozen ML models were applied to the same standard data set. In addition, performance was evaluated using four primary performance measures: accuracy, precision, recall, and the $F_1$-score, where the resulting scores are 0.98, 0.98, 0.97, and 0.97, respectively. The ROC curves and exact recall are two diagnostic tools to help interpret probabilistic predictions. The performance of the ML models was enumerated and compared before and after the script features were added. The results showed that the predictive performance of the framework for terrorist attack types was significantly improved by integrating text features with other features. In addition, the results indicate that the abstract text features of GTD complement other GTD performance optimization features. The research results will help obtain a clearer understanding of terrorism, improve government vigilance and emergency management capabilities in countering terrorist attacks, and provide a solid and reliable foundation and reference for joint counter-terrorism for different countries and regions. However, there are significant limitations to this study. The dimension curse occurs because, as dimensions increase, the size of space increases so rapidly that the available data becomes sparse. ML models need help with this disparity and perform poorly. Larger training datasets are needed to counter this. The possibility of optimization increases with feature count.

Additionally, as the number of features increases, the model becomes more complex. The temporal complexity of the proposed framework has increased due to the increased dimensionality of the text-generated features. In future work, we will endeavor to identify only the essential features and drop the undesirable elements, which is expected to improve the accuracy of the classification model. This can be achieved using feature-reduction techniques to reduce complexity and expand the proposed framework, which can predict additional terrorist attacks based on the names of target groups. Future research will also strive to improve the model's performance further by adding new features using feature engineering. Future work will review the available optimization methods, such as meta-heuristic optimization algorithms for feature selection. Only the most relevant features that boost the performance are selected using a Swarm-based algorithm. Also, we will focus on laying out the role of DL as a proposed algorithmic solution.

**Acknowledgement:** The authors would like to acknowledge the comments and suggestions given by anonymous reviewers. Also, the authors would like to thank the Chinese Government Scholarship Council, Wuhan University of Technology, Wuhan, China and Comenius University Bratislava, Slovakia for supporting this work.

**Funding Statement:** The authors received no specific funding for this study.

**Author Contributions:** The authors confirm contribution to the paper as follows: study conception and design: Mohammed Abdalsalam, Chunlin Li, Abdelghani Dahou; data collection: Mohammed Abdalsalam; analysis and interpretation of results: Mohammed Abdalsalam, Abdelghani Dahou and Natalia Kryvinska; draft manuscript preparation: Mohammed Abdalsalam, Abdelghani Dahou, Chunlin Li and Natalia Kryvinska. All authors reviewed the results and approved the final version of the manuscript.

**Availability of Data and Materials:** The authors declare that all data supporting the findings of this study are available within the article.

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

## References

1. Pan, X. (2021). Quantitative analysis and prediction of global terrorist attacks based on machine learning. *Scientific Programming, 2021,* 1–15.
2. Basu, N. (2021). Learning lessons from countering terrorism: The UK experience 2017–2020. *Cambridge Journal of Evidence-Based Policing, 5(3),* 134–145.
3. Sufi, F. K., Alsulami, M., Gutub, A. (2023). Automating global threat-maps generation via advancements of news sensors and AI. *Arabian Journal for Science and Engineering, 48(2),* 2455–2472.

4. Haghani, M., Kuligowski, E., Rajabifard, A., Lentini, P. (2022). Fifty years of scholarly research on terrorism: Intellectual progression, structural composition, trends and knowledge gaps of the field. *International Journal of Disaster Risk Reduction, 68,* 102714.
5. Uddin, M. I., Zada, N., Aziz, F., Saeed, Y., Zeb, A. et al. (2020). Prediction of future terrorist activities using deep neural networks. *Complexity, 2020(17),* 1–16.
6. Yang, Y., Cui, X. (2021). Bert-enhanced text graph neural network for classification. *Entropy, 23(11),* 1536.
7. Naseem, U., Razzak, I., Khan, S. K., Prasad, M. (2021). A comprehensive survey on word representation models: From classical to state-of-the-art word representation language models. *Transactions on Asian and Low-Resource Language Information Processing, 20(5),* 1–35.
8. Naseem, U., Razzak, I., Khushi, M., Eklund, P. W., Kim, J. (2021). COVIDSenti: A large-scale benchmark twitter data set for COVID-19 sentiment analysis. *IEEE Transactions on Computational Social Systems, 8(4),* 1003–1015.
9. Obied, Z., Solyman, A., Ullah, A., FathAlalim, A., Alsayed, A. (2021). BERT multilingual and capsule network for arabic sentiment analysis. *2020 International Conference On Computer, Control, Electrical, and Electronics Engineering (ICCCEEE)*, Khartoum, Sudan, IEEE.
10. Ganor, B. (2021). Artificial or human: A new era of counterterrorism intelligence? *Studies in Conflict & Terrorism, 44(7),* 605–624.
11. Qurtuby, S. A., Aldamer, S. (2021). Terrorism and counterterrorism in Saudi Arabia. *Contemporary Review of the Middle East, 8(1),* 56–76.
12. Liu, J., Lin, Y., Du, J., Zhang, H., Chen, Z. et al. (2023). ASFS: A novel streaming feature selection for multi-label data based on neighborhood rough set. *Applied Intelligence, 53(2),* 1707–1724.
13. Annamoradnejad, I., Zoghi, G. (2020). ColBERT: Using BERT sentence embedding for humor detection. arXiv preprint arXiv: 2004.12765.
14. Chaudhary, M., Bansal, D. (2022). Open source intelligence extraction for terrorism-related information: A review. *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery, 12(5),* e1473.
15. Bowie, N. G. (2017). Terrorism events data: An inventory of databases and data sets, 1968–2017. *Perspectives on Terrorism, 11(4),* 50–72.
16. Jongman, B. (2022). Recent online resources for the analysis of terrorism and related subjects. *Perspectives on Terrorism, 16(2),* 91–131.
17. Sönmez, E., Codal, K. S. (2022). Terrorism in cyberspace: A critical review of dark web studies under the terrorism landscape. *Sakarya University Journal of Computer and Information Sciences, 5(1)*, 1–21.
18. Najjar, E., Al-augby, S. (2021). Sentiment analysis combination in terrorist detection on twitter: A brief survey of approaches and techniques. *Research in Intelligent and Computing in Engineering, Select Proceedings of RICE 2020, 1254,* 231–240.
19. Almoqbel, M., Xu, S. (2019). Computational mining of social media to curb terrorism. *ACM Computing Surveys (CSUR), 52(5),* 1–25.
20. Gaikwad, M., Ahirrao, S., Phansalkar, S., Kotecha, K. (2021). Online extremism detection: A systematic literature review with emphasis on datasets, classification techniques, validation methods, and tools. *IEEE Access, 9,* 48364–48404.
21. Choudhary, P., Singh, U. (2015). A survey on social network analysis for counter-terrorism. *International Journal of Computer Applications, 112(9),* 24–29.
22. Ball, L. (2016). Automating social network analysis: A power tool for counter-terrorism. *Security Journal, 29(2),* 147–168.
23. Paolanti, M., Frontoni, E. (2020). Multidisciplinary pattern recognition applications: A review. *Computer Science Review, 37,* 100276.
24. Sarda, P., Chouhan, R. L. (2017). Extracting non-situational information from twitter during disaster events. *Journal of Cases on Information Technology, 19(1),* 15–23.

25. Fraiwan, M. (2022). Identification of markers and artificial intelligence-based classification of radical twitter data. *Applied Computing and Informatics, 5(4),* 10892.
26. Zajec, P., Mladenić, D. (2022). Using semi-supervised learning and wikipedia to train an event argument extraction system. *Informatica, 46(1).*
27. Kant, G., Weisser, C., Kneib, T., Säfken, B. (2023). Topic model—Machine learning classifier integrations on geocoded twitter data. In: *Biomedical and other applications of soft computing*, pp. 105–120. Springer.
28. Bridgelall, R. (2022). An application of natural language processing to classify what terrorists say they want. *Social Sciences, 11(1),* 23.
29. Olabanjo, O. A., Aribisala, B. S., Mazzara, M., Wusu, A. S. (2021). An ensemble machine learning model for the prediction of danger zones: Towards a global counter-terrorism. *Soft Computing Letters, 3,* 100020.
30. Abdalsalam, M., Li, C., Dahou, A., Noor, S. (2021). A study of the effects of textual features on prediction of terrorism attacks in GTD dataset. *Engineering Letters, 29(2).*
31. Feng, Y., Wang, D., Yin, Y., Li, Z., Hu, Z. (2020). An XGBoost-based casualty prediction method for terrorist attacks. *Complex & Intelligent Systems, 6(3),* 721–740.
32. Feng, Y., Gai, M., Wang, F., Wang, R., Xu, X. (2020). Classification and early warning model of terrorist attacks based on optimal GCN. *Chinese Journal of Electronics, 29(6),* 1193–1200.
33. Huamaní, E. L., Alicia, A. M., Roman-Gonzalez, A. (2020). Machine learning techniques to visualize and predict terrorist attacks worldwide using the global terrorism database. *Machine Learning, 11(5)*, 1487.
34. Hu, X., Lai, F., Chen, G., Zou, R., Feng, Q. (2019). Quantitative research on global terrorist attacks and terrorist attack classification. *Sustainability, 11(5),* 1487.
35. Zhenkai, L., Yimin, D., Jinping, L. (2020). Analysis model of terrorist attacks based on big data. *2020 Chinese Control and Decision Conference (CCDC)*, Hefei, China, IEEE.
36. Saiya, N., Scime, A. (2019). Comparing classification trees to discern patterns of terrorism. *Social Science Quarterly, 100(4),* 1420–1444.
37. Meng, X., Nie, L., Song, J. (2019). Big data-based prediction of terrorist attacks. *Computers & Electrical Engineering, 77,* 120–127.
38. Masood, A., Scazzoli, D., Sharma, N., Le Moullec, Y., Ahmad, R. et al. (2020). Surveying pervasive public safety communication technologies in the context of terrorist attacks. *Physical Communication, 41,* 101109.
39. Hassani, H., Beneki, C., Unger, S., Mazinani, M. T., Yeganegi, M. R. (2020). Text mining in big data analytics. *Big Data and Cognitive Computing, 4(1),* 1.
40. Murty, C. A., Rana, H., Verma, R., Pathak, R., Rughani, P. H. (2022). Building an AI/ML based classification framework for dark web text data. *Proceedings of International Conference on Computing and Communication Networks: ICCCN 2021*, Manchester, Metropolitan University, UK, Springer.
41. Cil, A. E., Yildiz, K., Buldu, A. (2021). Detection of ddos attacks with feed forward based deep neural network model. *Expert Systems with Applications, 169,* 114520.
42. Panigrahi, R., Borah, S., Bhoi, A. K., Ijaz, M. F., Pramanik, M. et al. (2021). A consolidated decision tree-based intrusion detection system for binary and multiclass imbalanced datasets. *Mathematics, 9(7),* 751.
43. Saidi, F., Trabelsi, Z. (2022). A hybrid deep learning-based framework for future terrorist activities modeling and prediction. *Egyptian Informatics Journal, 23(3),* 437–446.
44. Buffa, C., Sagan, V., Brunner, G., Phillips, Z. (2022). Predicting terrorism in europe with remote sensing, spatial statistics, and machine learning. *ISPRS International Journal of Geo-Information, 11(4),* 211.
45. Srinivasa, K., Thilagam, P. S. (2019). Crime base: Towards building a knowledge base for crime entities and their relationships from online news papers. *Information Processing & Management, 56(6),* 102059.
46. Schuurman, B. (2019). Topics in terrorism research: Reviewing trends and gaps, 2007–2016. *Critical Studies on Terrorism, 12(3),* 463–480.

47. Zhang, R., El-Gohary, N. (2021). A deep neural network-based method for deep information extraction using transfer learning strategies to support automated compliance checking. *Automation in Construction, 132,* 103834.
48. Alawadh, H. M., Alabrah, A., Meraj, T., Rauf, H. T. (2023). Discourse analysis based credibility checks to online reviews using deep learning based discourse markers. *Computer Speech & Language, 78,* 101450.
49. Koklu, M., Unlersen, M. F., Ozkan, I. A., Aslan, M. F., Sabanci, K. (2022). A CNN-SVM study based on selected deep features for grapevine leaves classification. *Measurement, 188,* 110425.
50. Roy, P. K., Saumya, S., Singh, J. P., Banerjee, S., Gutub, A. (2023). Analysis of community question-answering issues via machine learning and deep learning: State-of-the-art review. *CAAI Transactions on Intelligence Technology, 8(1),* 95–117.
51. Jamil, M. L., Pais, S., Cordeiro, J., Dias, G. (2022). Detection of extreme sentiments on social networks with bert. *Social Network Analysis and Mining, 12(1),* 55.
52. Bell, L. N. (2021). *Targets of terror: Contemporary assassination*. Rowman & Littlefield Publishers.
53. Luo, L., Qi, C. (2022). The tendency of terrorist organizations to explosive attacks: An institutional theory perspective. *Frontiers in Psychology, 13,* 747967.
54. Ridley, A. R., Nelson-Flower, M. J., Wiley, E. M., Humphries, D. J., Kokko, H. (2022). Kidnapping intergroup young: An alternative strategy to maintain group size in the group-living pied babbler (turdoides bicolor). *Philosophical Transactions of the Royal Society B, 377(1851),* 20210153.
55. Li, Z., Li, X., Dong, C., Guo, F., Zhang, F. et al. (2021). Quantitative analysis of global terrorist attacks based on the global terrorism database. *Sustainability, 13(14),* 7598.
56. Husain, S. S., Sharma, K., Kukreti, V., Chakraborti, A. (2020). Identifying the global terror hubs and vulnerable motifs using complex network dynamics. *Physica A: Statistical Mechanics and Its Applications, 540,* 123113.
57. Nayak, N., Rayachoti, M., Gupta, A. M., Prerna, G., Sreenath, M. et al. (2023). Learning future terrorist targets using attention based hybrid cnn and bilstm model. *2023 IEEE International Conference on Integrated Circuits and Communication Systems (ICICACS)*, Raichur, India, IEEE.
58. Tolan, G. M., Soliman, O. S. (2015). An experimental study of classification algorithms for terrorism prediction. *International Journal of Knowledge Engineering, 1(2),* 107–112.
59. Muhammad, H., Kazi, H. (2016). Use of predictive modeling for prediction of future terrorist attacks in pakistan. *International Journal of Computer Applications, 179,* 8–16.
60. Munappy, A. R., Bosch, J., Olsson, H. H., Arpteg, A., Brinne, B. (2022). Data management for production quality deep learning models: Challenges and solutions. *Journal of Systems and Software, 191,* 108180.
61. Solyman, A., Wang, Z., Tao, Q., Elhag, A. A. M., Zhang, R. et al. (2022). Automatic arabic grammatical error correction based on expectation-maximization routing and target-bidirectional agreement. *Knowledge-Based Systems, 241,* 108180.
62. Chai, C., Wang, J., Luo, Y., Niu, Z., Li, G. (2022). Data management for machine learning: A survey. *IEEE Transactions on Knowledge and Data Engineering, 35(5),* 4646–4667.
63. Maharana, K., Mondal, S., Nemade, B. (2022). A review: Data pre-processing and data augmentation techniques. *Global Transitions Proceedings, 3,* 91–99.
64. Chowdhary, K. (2020). Natural language processing. In: *Fundamentals of artificial intelligence*, pp. 603–649. Springer.
65. Jhaveri, R. H., Revathi, A., Ramana, K., Raut, R., Dhanaraj, R. K. (2022). A review on machine learning strategies for real-world engineering applications. *Mobile Information Systems, 2022,* 1–26.
66. Kumari, S. (2022). Text mining and pre-processing methods for social media data extraction and processing. In: *Handbook of research on opinion mining and text analytics on literary works and social media*, pp. 22–53. Hershey, Pennsylvania, USA: IGI Global.

67. Sung, Y. W., Park, D. S., Kim, C. G. (2023). A study of bert-based classification performance of text-based health counseling data. *Computer Modeling in Engineering & Sciences, 135(1),* 795–808. https://doi.org/10.32604/cmes.2022.022465

68. Kowsher, M., Sami, A. A., Prottasha, N. J., Arefin, M. S., Dhar, P. K. et al. (2022). Bangla-BERT: Transformer-based efficient model for transfer learning and language understanding. *IEEE Access, 10,* 91855–91870.

69. Xiong, J., Yu, L., Niu, X., Leng, Y. (2023). XRR: Extreme multi-label text classification with candidate retrieving and deep ranking. *Information Sciences, 622,* 115–132.

70. Hou, S., Liu, Y., Yang, Q. (2022). Real-time prediction of rock mass classification based on tbm operation big data and stacking technique of ensemble learning. *Journal of Rock Mechanics and Geotechnical Engineering, 14(1),* 123–143.

71. Yıldız, K., Buldu, A., Demetgul, M. (2016). A thermal-based defect classification method in textile fabrics with k-nearest neighbor algorithm. *Journal of Industrial Textiles, 45(5),* 780–795.

72. Zhang, C., Yao, J., Hu, G., Schøtt, T. (2020). Applying feature-weighted gradient decent k-nearest neighbor to select promising projects for scientific funding. *Computers, Materials & Continua, 64(3),* 1741–1753. https://doi.org/10.32604/cmc.2020.010306

73. Lu, D. N., Le, H. Q., Vu, T. H. (2020). The factors affecting acceptance of e-learning: A machine learning algorithm approach. *Education Sciences, 10(10),* 270.

74. Sun, D., Xu, J., Wen, H., Wang, D. (2021). Assessment of landslide susceptibility mapping based on bayesian hyperparameter optimization: A comparison between logistic regression and random forest. *Engineering Geology, 281,* 105972.

75. Amiri, V., Nakagawa, K. (2021). Using a linear discriminant analysis (LDA)-based nomenclature system and self-organizing maps (SOM) for spatiotemporal assessment of groundwater quality in a coastal aquifer. *Journal of Hydrology, 603,* 127082.

76. Louk, M. H. L., Tama, B. A. (2023). Dual-IDS: A bagging-based gradient boosting decision tree model for network anomaly intrusion detection system. *Expert Systems with Applications, 213,* 119030.

77. Bentéjac, C., Csörgő, A., Martínez-Muñoz, G. (2021). A comparative analysis of gradient boosting algorithms. *Artificial Intelligence Review, 54(3),* 1937–1967.

78. Alzamzami, F., Hoda, M., El Saddik, A. (2020). Light gradient boosting machine for general sentiment classification on short texts: A comparative evaluation. *IEEE Access, 8,* 101840–101858.

79. Yerima, S. Y., Bashar, A. (2022). A novel android botnet detection system using image-based and manifest file features. *Electronics, 11(3),* 486.

80. Kowsari, K., Jafari Meimandi, K., Heidarysafa, M., Mendu, S., Barnes, L. et al. (2019). Text classification algorithms: A survey. *Information, 10(4),* 150.

81. Zou, Z., Ergan, S. (2023). Towards emotionally intelligent buildings: A convolutional neural network based approach to classify human emotional experience in virtual built environments. *Advanced Engineering Informatics, 55,* 101868.

82. Chicco, D., Jurman, G. (2020). The advantages of the matthews correlation coefficient (MCC) over f1 score and accuracy in binary classification evaluation. *BMC Genomics, 21(1),* 1–13.

83. Khosravi, K., Shahabi, H., Pham, B. T., Adamowski, J., Shirzadi, A. et al. (2019). A comparative assessment of flood susceptibility modeling using multi-criteria decision-making analysis and machine learning methods. *Journal of Hydrology, 573,* 311–323.

84. Al-Tamimi, Y., Shkoukani, M. (2023). Employing cluster-based class decomposition approach to detect phishing websites using machine learning classifiers. *International Journal of Data and Network Science, 7(1),* 313–328.

85. Ghalleb, A. E. K., Amara, N. E. B. (2020). Terrorist act prediction based on machine learning: Case study of tunisia. *2020 17th International Multi-Conference on Systems, Signals & Devices (SSD)*, Sfax, Tunisia, IEEE.
86. Alsaedi, A. S., Almobarak, A. S., Alharbi, S. T. (2019). Mining the global terrorism dataset using machine learning algorithms. *2019 IEEE/ACS 16th International Conference on Computer Systems and Applications (AICCSA)*, Abu Dhabi, UAE, IEEE.
87. Raj, A., Somani, S. B. (2022). Predicting terror attacks using Neo4j sandbox and machine learning algorithms. *2022 6th International Conference on Computing, Communication, Control and Automation (ICCUBEA)*, Pune, India, IEEE.
88. Diab, S. (2019). Optimizing stochastic gradient descent in text classification based on fine-tuning hyper-parameters approach. A case study on automatic classification of global terrorist attacks. arXiv preprint arXiv:1902.06542.