Analysis and Classification of the Global Terrorism Database (1970-2020)

Ayush Gupte

Supervisor(s): Prof. Dr. Piyush Chauhan

November 5, 2025

Abstract

The following project undertakes a dual analysis of the GTD from 1970 to 2020, marrying EDA with predictive machine learning. The central problem statement can be framed as: to establish and visualize significant long-term trends and geospatial shifts in global terrorism, and to determine if a machine learning model can accurately attribute a terrorist incident to a specific perpetrator (GroupName) based on the characteristics of that incident. First, the approach consisted of an EDA step by visualizing incident frequency over time, geographic hotspots, and common attack typologies. In the predictive part, after preprocessing a filtered dataset with a focus on the top 20-30 most active groups using one-hot encoding and standardization, multiple classification models, including LightGBM, XGBoost, Random Forest, and a Neural Network, were trained and compared. A separate test for temporal validation was executed-train on 1970-2008 and test on 2009-2010-to evaluate the real-world predictive power. Key findings from the EDA include the fact that there is a sharp, post-2001 increase in incidents, with a peak around 2014, and a geographical shift in hotspots from Latin America and Western Europe to the Middle East, North Africa, and South Asia. Machine learning models showed excellent efficacy; for instance, the Voting Classifier (XGBoost + LightGBM) reached 97.39 percent accuracy. Among the most important predictive features identified were Latitude, Longitude, and Year. This work proves that machine learning can be a powerful tool in incident attribution. This will give way to a high-accuracy predictive model,

serving as a proof-of-concept for applications in intelligence, counterterrorism, and strategic analysis, enabling authorities to understand and anticipate group-specific threats better.

Table of Contents

1 Introduction

1.1 Background and Context of the Problem

Global terrorism has grown in complexity and persistence during the past five decades and has emerged as a leading global threat. Its nature continuously evolves through shifting geographic epicenters, changing tactics, and emerging perpetrator groups. Understanding the dynamics of these incidents is crucial for policymakers, intelligence agencies, and researchers working to mitigate their impact. The GTD offers a database of over 200,000 terrorist events from 1970 to 2020 that is unsurpassed in terms of its completeness and open-source nature. This indeed creates an unsurpassed opportunity for analysis, but the challenge of the scale and complexity of such a dataset is formidable. Key difficulties include finding long-term trends within the noise and, importantly, the challenge of attribution. In many real-world situations, the attacking group is not obviously known. However, different groups often have distinct patterns or modi operandi—favoring particular locations, targets, or methods. This "signature" behavior suggests a data-driven problem: Can a machine learning model be trained to analyze the characteristics of a terrorist incident and correctly predict which group is responsible? This project addresses this problem through an in-depth Exploratory Data Analysis to uncover the historical and geographical context of terrorism, followed by leveraging those insights to build and evaluate a series of models in classification to determine the feasibility and accuracy of automated incident attribution.

1.2 Motivation for the study

There is a great need for quantitative, data-driven approaches for understanding and analyzing global terrorism. The GTD dataset, though comprehensive, is too large for any human analyst to extract meaningful information from manually. This analysis is based on the following two motivating goals of the project:

1. To Provide Clarity Through Data: The first motivation is to apply modern data science and visualization techniques to the 50-year GTD dataset. The goal is to move beyond generalized or anecdotal narratives about terrorism and to rigorously identify, quantify, and visualize the concrete historical trends, geographic shifts, and tactical preferences that have defined this phenomenon.

2. For Testing A Predictive Hypothesis: The second, and more complex goal is to investigate how attribution works. In the first few hours after an attack, some of the most important things that security and intelligence organizations need to know are "Who did it?". The authors have developed a hypothesis that terrorist groups will be identifiable through their unique signature based on what they believe in (ideology) or where they live (geography). By using a variety of attributes such as where the attacks occur (latitude, longitude), when they occurred (year), and what type of attack was used (attacktype, weapontype), the objective of this project is to see if the use of machine learning algorithms can learn enough about each group to make accurate predictions about which group is responsible, thereby potentially providing a new tool for helping to quickly identify who is responsible for an attack.

1.3 Problem statement

The two primary problems related to the Global Terrorism Database (GTD):

1. Problem of comprehension: The GTD contains over 200,000 incident records spanning 50 years; and manual comprehension of critical patterns is infeasible. The first problem is to distill this large data into a clear, visual, and interpretable summary of the key temporal, geographical, and tactical trends in global terrorism.

2. Attribution Problem: The group(s) responsible for most acts of terrorism are usually unknown when they occur. This leads to the primary predictive problem: Can a machine learning model be developed such that the model will accurately determine the GroupName responsible for an act of terrorism, from the observable characteristics of the incident itself (the location, time, and type of attack).

1.4 Objectives of the Project

The objectives of this project are:

1. To perform a comprehensive Exploratory Data Analysis (EDA) on the Global Terrorism Database (GTD) to identify and visualize key historical trends, geographical hotspots, and dominant attack typologies from 1970 to 2020.

2. To preprocess and feature-engineer the dataset to make it suitable for machine learning, including handling missing values and encoding categorical data.

3. To build, train, and evaluate a series of supervised machine learning classification models (including LightGBM, XGBoost, Random Forest, and a Neural Network) to predict the terrorist group (GroupName) responsible for an incident.

4. To compare the performance of these models using accuracy as the primary metric and identify the most effective algorithm for this attribution task.

5. To determine the most significant features (e.g., location, year, weapon type) that contribute to the models' predictive power.

6. To validate the final model's robustness and real-world applicability by testing its performance on a "future" (out-of-sample) dataset using a temporal train-test split.

1.5 Novelty of your work

The novelty of this project lies not in performing a standard Exploratory Data Analysis of the GTD, but in the successful application and rigorous validation of modern machine learning models for the specific task of terrorist group attribution. While the GTD is widely studied, this work moves beyond descriptive statistics to create a high-accuracy predictive tool. The key novel benefaction are:

1. Demonstrated Feasibility: This study makes it clear that predicting the perpetrator behind a terrorist attack is quite feasible. The accuracy can reach as high as 97.39 percent in these predictions. All it takes is the metadata from the incident itself.

2. Model Benchmarking: It provides the direct comparison of several state-of-the-art classification algorithms (including ensemble methods: XGBoost, LightGBM, and a Voting Classifier) for this particular attribution problem, highlighting the most powerful models.

3. Real-World Scenario Test: One key part of this work involves applying a time-based split between training and testing phases. They handled training with data covering the years 1970 to 2008. Testing then focused on the period from 2009 to 2010. This simulates a real-world intelligence scenario, and the resulting 95.02 percent accuracy confirms that the models are robust, generalizable, and not just overfitting to a static dataset.

4. Feature Importance: The work quantitatively confirms that an attack's location (Latitude, Longitude) and time (Year) are overwhelmingly the most powerful predictors of the responsible group, providing a clear and actionable insight.

2 Literature Review / Related Work

2.1 Overview of previous research and existing methods

The application of data science and machine learning to the Global Terrorism Database (GTD) is an established and growing field of research. Previous work can be broadly categorized into two main areas: descriptive/exploratory analysis and predictive modeling.

1. Descriptive and Exploratory Analysis: A large body of research, much like the first half of this project, has used the GTD for Exploratory Data Analysis (EDA) and statistical analysis.These studies focus on the identification of historical patterns, spatiotemporal trends, and correlations. Indeed, scholars have created numerous maps depicting the geographic shifts in terrorism, rises and falls in incident frequency, and changing target and tactic preferences. This descriptive work is foundational, providing the necessary context for more advanced predictive studies.

2. Predictive Modeling: The application of predictive modeling to the Global Terrorism Database (GTD) is also a very complex area of research. Researchers have tackled multiple areas of research within the use of predictive models with the GTD. Several areas of study are focused on prediction of either the probability or location of future terrorist attacks. This can be framed as either a binary classification problem ("Will a terrorist attack occur in this geographic region?") or as a hotspot analysis problem. In addition, other types of predictive problems that have been studied include forecasted numbers of casualties, the type of attack, weapon type etc.

3. Related Work on Perpetrator Group Attribution: This project's specific objective—predicting the responsible terrorist group (GroupName)—is a well-known but challenging multi-class attribution problem. The literature shows several approaches to this:

Traditional Machine Learning: Early and common methods apply "shallow" machine learning models. Studies have benchmarked a range of classifiers, including Decision

Trees, Random Forests, Naïve Bayes, Support Vector Machines (SVM), and K-Nearest Neighbors (KNN), to predict the perpetrator.

Advanced Ensemble and Deep Learning Models: The latest studies that acknowledge the complexity of the problem have developed better solutions than earlier approaches. In addition to the success of Ensemble Methods (i.e., Stacking and Voting Classifiers) in improving performance by combining a number of models, many other researchers have turned to Deep Learning as an alternative solution. They are employing Recurrent Neural Networks (RNNs) or Natural Language Processing (NLP) models such as BERT for their analysis of the textual "summary" field of the GTD. These models combine this analysis with structured data to further improve the prediction of terrorist attacks.

2.2 Comparison of techniques and their limitations

The methods used to analyze the GTD for perpetrator attribution range from simple statistical models to complex deep learning architectures, each with distinct limitations.

- Shallow Machine Learning (Naïve Bayes, KNN, SVM):
  - Techniques: These models are often used as a baseline. As demonstrated in this project's notebook, Gaussian Naïve Bayes performed very poorly (65.34 percent accuracy).
  - Limitations: The poor performance of Naïve Bayes is likely because it assumes all features are independent, which is a false assumption for this dataset (e.g., AttackType and WeaponType are highly correlated). Other models in this class, like KNN and SVM (as noted in the literature), often struggle with the high dimensionality and non-linear relationships in the data and can be computationally slow on a dataset of this size.
- Tree-Based Ensemble Models (Random Forest, XGBoost, LightGBM):
  - Techniques: These models form the core of modern machine learning for structured data and were the top performers in this project. Random Forest, XGBoost, and LightGBM all achieved accuracies above 96 percent, with the ensemble Voting Classifier (combining XGBoost and LightGBM) reaching the peak performance of 97.39 percent.

  - Limitations: While highly accurate, their main limitation is a tendency to overfit if not properly tuned and validated. A common limitation in the literature is that their performance is often reported on a random data split, which can be overly optimistic. This project specifically addresses this limitation by also performing a robust temporal validation.

- Neural Networks (MLP, Deep Learning):
  - Techniques: This project evaluated a Multi-Layer Perceptron (MLP), a foundational neural network, which achieved a strong accuracy of 95.65 persent. More advanced research sometimes uses deep learning (like RNNs) and NLP to analyze incident text summaries.
  - Limitations: The MLP performed well but was outperformed by the tree-based ensembles, suggesting the latter is better suited for this type of tabular data. Neural networks also require careful data preparation (like feature scaling, which was performed in the notebook) and are often more of a "black box," making it harder to interpret feature importance compared to tree models. The high accuracy achieved in this project without complex NLP models suggests that the structured data (especially location and time) is sufficient, and the computational expense of deep learning/NLP models may not be necessary.

2.3 Gaps with the earlier work

This project was designed to address three specific gaps identified in the existing literature:

1. The Need for Robust Temporal Validation: A significant gap in many existing studies is the reliance on a standard, random train-test split. This methodology is flawed for a time-series problem, as a model trained on 80 percent of the data (which includes future incidents) to predict the other 20 percent does not simulate a real-world intelligence scenario. This project fills this critical gap by implementing a rigorous temporal validation (training on 1970-2008, testing on 2009-2010) to prove the model's ability to generalize and predict future, unseen events.

2. Performance Benchmarking of Modern Boosting Models: While many papers have benchmarked traditional models like Naïve Bayes, SVM, and Random Forest, there

Table 1: Literature Review Summary Table

Author(s) / Year	**Objective / Problem**	**Methodology Used**	**Key Finding / Accuracy**	**Limitations / Gaps Addressed by This Work**
Tolan & Soliman (2015)	Predict `GroupName` for GTD subset (Egypt)	SVM, KNN, Naïve Bayes	~75% accuracy (SVM/KNN)	Limited to a single country; used "shallow" ML models. This work uses more advanced models on a global scale.
(General Research - e.g., Huamaní et al.)	Predict `AttackType` or `WeaponType`	Random Forest, Decision Trees	Varies (e.g., ~84-91%)	Solves a different, simpler problem (predicting tactics, not the multi-class group attribution).
(General Research - e.g., Jović et al.)	Predict `GroupName` using text summaries	Deep Learning (BiGRU, BERT), NLP	Varies, but focuses on unstructured text data.	Highly complex and computationally expensive. This work proves high accuracy is possible without NLP.
This Project (from Notebook)	**Predict `GroupName` (Top 20-30 groups)**	**LightGBM, XGBoost, Random Forest, Voting Classifier**	**97.39% Accuracy (Voting)**	**Establishes a new, high-accuracy benchmark using optimized ensemble models on structured data.**
This Project (from Notebook)	**Validate model on future data**	**CatBoost (Temporal Split: Train 1970-08, Test 09-10)**	**95.02% Accuracy**	**Fills the critical temporal validation gap by proving the model generalizes to predict future, unseen data.**

is a gap in the literature regarding a direct, high-performance comparison of the latest generation of gradient-boosting libraries (specifically XGBoost and LightGBM) and optimized Voting Classifiers for this specific multi-class attribution problem. This work provides that benchmark, establishing a new, high-accuracy (97.39

3. Understanding Feature Sufficiency: With some advanced research moving toward highly complex and computationally expensive deep learning and NLP models (analyzing text summaries), there is a gap in understanding the true predictive power of the fundamental, structured features alone. This project fills that gap by demon-

strating that this complexity may be unnecessary, proving that geospatial and temporal features (Latitude, Longitude, Year) are overwhelmingly sufficient to achieve near-perfect attribution.

3 Methodology / Proposed System

3.1 Data Collection and Preprocessing

Data Collection The data for this project was sourced from a pre-cleaned version of the **Global Terrorism Database (GTD)**. This comprehensive, open-source database contains records of terrorist incidents worldwide from 1970 to 2020. The specific dataset used in this analysis comprised **209,706 incidents** (rows) and **11 key attributes** (columns):

- **Temporal Features:** `Year`, `Month`
- **Geospatial Features:** `Country`, `Region`, `Latitude`, `Longitude`
- **Incident Characteristics:** `AttackType`, `TargetType`, `WeaponType`, `suicide`
- **Attribution Feature:** `GroupName` (The target variable for prediction)
- Data Preprocessing A multi-stage preprocessing pipeline was implemented to clean the data for analysis and prepare it for the machine learning models.

  1. **Initial Cleaning and Imputation**

     This phase focused on standardizing the data and handling missing values to make the dataset robust for both visualization and modeling.

     – **Type Conversion:** The `suicide` column was converted from a non-numeric format to an integer (0 or 1), and the `Latitude` and `Longitude` columns were converted to a numeric type.

     – **Standardization:** The `GroupName` for unattributed attacks was standardized to `Unknown_Group` for consistency.

     – **Missing Value Imputation:** The geospatial features (`Latitude` and `Longitude`) contained missing values. To preserve these records for analysis, the missing data was imputed using the **median** value of each respective column.

2. **Feature Engineering for EDA**

   For the Exploratory Data Analysis, a `Decade` column was engineered from the `Year` feature. This allowed for the aggregation of incidents over 10-year periods to visualize long-term trends and geographic shifts more clearly.

3. **Preprocessing for Machine Learning**

   A separate, more rigorous preprocessing pipeline was built specifically for the machine learning phase.

   – **Data Filtering:** The dataset is extremely imbalanced, with over 3,000 unique group names. To create a solvable and computationally feasible classification problem, the data was filtered to include only the **top 20 or 30 most active groups** (excluding the `Unknown_Group`). This focused the problem on a high-value subset of 63,673 incidents.

   – **Feature Transformation:** A `ColumnTransformer` pipeline was implemented to apply different transformations to different types of features:

     ∗ **Categorical Features** (`Region`, `Country`, `AttackType`, `TargetType`, `WeaponType`): These were transformed using **One-Hot Encoding** to convert them into a binary numerical format that the models can understand.

     ∗ **Numerical Features** (`Year`, `Month`, `Latitude`, `Longitude`, `suicide`): These were scaled using **StandardScaler**, which normalizes the features to have a mean of 0 and a standard deviation of 1. This step is critical for the performance of models like Neural Networks (MLP) and KNN.

3.2 Feature Engineering

Feature engineering was performed for two distinct purposes: 1) to create a new aggregated feature to enhance the Exploratory Data Analysis (EDA), and 2) to transform the selected features into a format suitable for the machine learning models.

1. **Feature Creation for Exploratory Data Analysis**

   For the EDA, a single new feature, `Decade`, was engineered from the existing `Year` column.

- **Purpose:** The `Year` feature has 50 unique values (1970-2020). To visualize long-term trends more effectively, these were grouped into 10-year periods.
- **Implementation:** A function was applied to map each `Year` to its corresponding decade (e.g., 1970s, 1980s, etc.).
- **Impact:** This new feature allowed for the creation of aggregated visualizations, such as the "Incidents by Region Over Decades" stacked bar plot, which clearly illustrated the large-scale geographic shift in terrorist activity over time.

2. **Feature Transformation for Machine Learning**

   For the machine learning phase, a `ColumnTransformer` pipeline was built to systematically apply transformations to the features. This was not about creating new features, but about encoding existing ones for model consumption.

   - **Categorical Features:**
     - **Features:** `Region`, `Country`, `AttackType`, `TargetType`, `WeaponType`
     - **Transformation: One-Hot Encoding** was applied.
     - **Reason:** This technique converts categorical text data into a binary numerical format. This is essential because machine learning models cannot process raw text and it prevents the model from incorrectly assuming an ordinal relationship.
     - **Numerical Features:**
       - **Features:** `Year`, `Month`, `Latitude`, `Longitude`, `suicide`
       - **Transformation: StandardScaler** was applied.
       - **Reason:** This technique scales the features to have a mean of 0 and a standard deviation of 1. This normalization is critical for the performance and stability of certain models, particularly the **Neural Network (MLP)** and **KNN**.

3.3 Model Design / System Architecture

The system implemented in this project is a dual-phase pipeline designed to first analyze historical data and then build a high-accuracy predictive model for terrorist group attribution.

3.3.1 Phase 1: Exploratory Data Analysis (EDA) Subsystem

The first part of the architecture is a descriptive analysis subsystem. It ingests the full, preprocessed dataset (209,706 incidents) and applies a suite of visualization functions to generate key insights. This includes time-series analysis (line plots), categorical analysis (bar plots, stacked bar plots), proportional analysis (donut plots), and correlational analysis (heatmaps) to provide a comprehensive overview of the data.

3.3.2 Phase 2: Predictive Modeling Pipeline

The second, more complex subsystem is a machine learning pipeline designed for multi-class classification. Its architecture consists of the following components:

Data Input and Filtering The pipeline does not use the full, imbalanced dataset. It first ingests the data and applies a critical filter to select only the **top 20 or 30 most active groups** (excluding "Unknown_Group"). This results in a focused dataset of 63,673 incidents, creating a computationally feasible and meaningful classification task.

Preprocessing Engine The filtered data is passed to a `ColumnTransformer`. This engine is the core of the preprocessing architecture and automatically applies the correct transformation to each feature type in a single, consistent step:

- **Categorical Features** (`Region`, `Country`, `AttackType`, etc.) are transformed using **One-Hot Encoding**.
- **Numerical Features** (`Year`, `Latitude`, `Longitude`, etc.) are normalized using **StandardScaler**.

Model Benchmarking Design The system is designed for comparative analysis. The preprocessed data is fed to a portfolio of seven distinct classification models, which are trained and evaluated to find the best performer:

- Decision Tree

- Gaussian Naive Bayes
- Random Forest
- XGBoost (GPU-enabled)
- LightGBM
- Multi-Layer Perceptron (MLP) Neural Network
- A **Voting Classifier**

Final Model Architecture The final and optimal model is an
textbfensemble architecture. It is a
texttttVotingClassifier that combines the predictions of the two top performing base models:
textbfLightGBM and
textbfXGBoost. By averaging their probabilistic predictions, it leverages the strengths of both models to achieve higher and more stable accuracy.

Dual-Validation System The two validation techniques that are central to the system's design are as follows:

- **Simple validation**: An 80/20 random train/test split was created in order to compare all seven models using the filtered data set, with standard metrics used for this comparison.
- **Temporal validation**: A more thorough simulation was also completed. A model (CatBoost) was trained on only the 1970–2008 portion of the data, while being tested only on the 2009–2010 "future" portion of the data to evaluate the model's ability to apply knowledge learned to future events.

subsectionTraining and Evaluation Setup

The system's performance was assessed using two distinct and rigorous validation methodologies, both aimed at the multi-class classification task of predicting `GroupName`. The primary metric for success across all evaluations was **Accuracy**.

3.3.3 Standard Model Benchmarking Setup

This setup was designed to compare the relative performance of the seven implemented models (e.g., LightGBM, XGBoost, MLP, etc.) to find the single best architecture.

- **Dataset:** The filtered dataset containing the top 20-30 most active groups (63,673 incidents).
- **Split Strategy:** A standard, randomized 80%/20% train-test split (`train_test_split`).
- **Purpose:** To train all seven models on the same data and evaluate their accuracy on a randomized, held-out test set. This allowed for a direct comparison to select the champion model (the Voting Classifier).
- **Output:** A set of accuracy scores, with the Voting Classifier achieving the highest at 97.39%.

3.3.4 Temporal Validation Setup

This more rigorous setup was designed to simulate a real-world scenario, testing the model's ability to generalize and predict *future*, unseen events based on *past* data.

- **Dataset:** The full, time-indexed dataset.
- **Split Strategy:** A strict, non-random **temporal split**.
  - **Training Set:** All incidents from 1970 through 2008.
  - **Test Set:** All incidents from 2009 and 2010.
- **Model Used:** A CatBoost classifier was used for this specific validation task.
- **Purpose:** To provide a robust and realistic measure of the model's predictive power on out-of-sample data, proving it is not simply overfitting to the training years.
- **Output:** An accuracy score of 95.02% on the 2009-2010 test set, confirming the model's strong generalization capabilities.

3.4 Training and Evaluation Setup

The system's performance was assessed using two distinct and rigorous validation methodologies, both aimed at the multi-class classification task of predicting `GroupName`. The primary metric for success across all evaluations was **Accuracy**.

3.4.1 Standard Model Benchmarking Setup

This setup was designed to compare the relative performance of the seven implemented models (e.g., LightGBM, XGBoost, MLP, etc.) to find the single best architecture.

- **Dataset:** The filtered dataset containing the top 20-30 most active groups (63,673 incidents).
- **Split Strategy:** A standard, randomized 80%/20% train-test split (`train_test_split`).
- **Purpose:** To train all seven models on the same data and evaluate their accuracy on a randomized, held-out test set. This allowed for a direct comparison to select the champion model (the Voting Classifier).
- **Output:** A set of accuracy scores, with the Voting Classifier achieving the highest at 97.39%.

3.4.2 Temporal Validation Setup

The model was tested in an environment that simulated a "real world" experience by validating its potential to generalize and make predictions about future unseen events based on past data.

Validation details are as follows:

- **Dataset:** Full time indexed dataset.
- **Data Split Strategy:**Temporally based (time based), rather than random, split of the data.
  - **Train Data:** 1970–2008
  - **Test Data:** 2009–2010

- **Model Type:** A CatBoost classifier was used in the validation process..
- **Objective:** To establish a reliable method for determining how well the model will perform with out-of-training sample data; i.e., to determine if the model is simply over-fitting the training years.
- **Results:** Accuracy of **95.02%** on the 2009–2010 test data, which validated the model's ability to generalize.

4 Implementation

4.1 Detailed Explanation of Implementation Steps

The code was implemented in a Python environment using a Jupyter Notebook. Libraries used included but were not limited to pandas and numpy for data manipulation, matplotlib and seaborn for visualization, and sklearn, xgboost, lightgbm, and catboost for machine learning.

4.1.1 Phase 1: Exploratory Data Analysis (EDA) Implementation

(a) **Data Loading and Initial Cleaning:** The dataset was loaded into a `pandas` DataFrame; the initial clean up consisted of converting the `suicide` column to a numeric type, replacing "Unknown" category names with `unknown`, and numeric type conversions to both `latitude` and `longitude`.

(b) **Imputation:**The missing values within the `latitude` and `longitude` columns were replaced with the **median** value of each column to ensure that the geographic data could be used for further analysis.

(c) **Feature Engineering:** Finally, the `decade` feature was created from the `year` column to enable a trend analysis over time.

(d) **Visualization:** A series of plots were generated using `seaborn` and `matplotlib` to analyze distributions and trends. This included:

  - A `lineplot` for incidents over time.
  - `barplots` for top countries and attack types.

- A heatmap to correlate attack types with regions.
- A stacked barplot (or equivalent) for regional incidents by decade.
- A donut plot to show the proportion of suicide attacks.

4.1.2 Phase 2: Standard Model Benchmarking Implementation

(a) **Data Filtering:** The dataset was prepared for modeling by first removing all incidents attributed to Unknown_Group. The dataset was then filtered to include only the **top 20 or 30 most active groups**, resulting in a focused, high-value dataset of 63,673 records.

(b) **Data Splitting:** The filtered data was split into features (X) and the target variable (y, which is GroupName). This was then divided into training and testing sets using sklearn.model_selection.train_test_split with a standard 80%/20% ratio.

(c) **Preprocessing Pipeline:** A ColumnTransformer from sklearn.pipeline was constructed.

- **Numerical Features** (Year, Month, Latitude, etc.) were passed to a StandardScaler.
- **Categorical Features** (Region, Country, AttackType, etc.) were passed to a OneHotEncoder.

(d) **Pipeline Execution:** The ColumnTransformer was **fit** on the training data (X_train) and then used to **transform** both X_train and X_test.

(e) **Model Training:** Seven models were instantiated: Decision Tree, Gaussian Naive Bayes, Random Forest, XGBoost (GPU-enabled), LightGBM, a Multi-Layer Perceptron (MLP), and a Voting Classifier (combining XGBoost and LightGBM). Each model was trained on the processed training data.

(f) **Model Evaluation:** Each trained model was used to make predictions on the processed test data. The accuracy_score from sklearn.metrics was calculated for each, and the results were printed for comparison.

4.1.3 Phase 3: Temporal Validation Implementation

To ensure the model could generalize to "future" data, a separate, more rigorous validation was implemented.

(a) **Temporal Data Splitting:** A random selection of the sample was made by splitting the data in a way that wasn't random. All incidents for the years 1970–2008 were designated as the **training** data and all incidents for the years 2009 – 2010 were designated as the **test** data.

(b) **Pipeline Execution (Temporal):** The `ColumnTransformer` was **re-fit** using *only* the temporal training data (1970-2008) to prevent data leakage. This fitted preprocessor was then used to transform both the temporal train and test sets.

(c) **Model Training and Evaluation:** A `CatBoostClassifier` was instantiated, trained on the processed 1970-2008 data, and then evaluated on the processed 2009-2010 test set. This validation yielded the 95.02% accuracy score, confirming the model's real-world predictive power.

4.2 Technologies and Platforms Used

The project was implemented entirely in the Python programming language within a Jupyter Notebook environment. The analysis and modeling relied on a standard stack of open-source data science and machine learning libraries.

- **Core Data Manipulation:**Pandas and NumPy were used for all data loading, cleaning, filtering, and transformation tasks.
- **Data Visualization:** Matplotlib and Seaborn were the primary libraries used to generate all plots, graphs, and heatmaps for the exploratory data.
- **Machine Learning (Core):** The scikit-learn (sklearn) library was the central framework for the entire machine learning pipeline.This included:
  - **Preprocessing:** `ColumnTransformer`, `StandardScaler`, and `OneHotEncoder`.
  - **Model Splitting & Evaluation:** `train_test_split` and `accuracy_score`.

  - **Models:** DecisionTreeClassifier, GaussianNB, RandomForestClassifier, and MLPClassifier.
  - **Ensemble Methods:** VotingClassifier.

- **Advanced Gradient Boosting Models:**
  - XGBoost (xgboost.XGBClassifier), with GPU acceleration enabled for performance.
  - LightGBM (lightgbm.LGBMClassifier).
  - CatBoost (catboost.CatBoostClassifier), used specifically for the temporal validation.

subsectionProgramming languages / frameworks

The project was implemented entirely in the **Python** programming language, leveraging its extensive ecosystem of open-source data science frameworks. The development environment was a **Jupyter Notebook**.

- **Core Data Manipulation:** Pandas and NumPy were used for all data loading, cleaning, filtering, and transformation tasks.
- **Data Visualization:** Matplotlib and Seaborn were the primary frameworks used to generate all plots, graphs, and heatmaps for the Exploratory Data Analysis (EDA).
- **Core Machine Learning Framework:** The scikit-learn (sklearn) library served as the foundational framework for the machine learning pipeline. This included:
  - **Preprocessing:** ColumnTransformer, StandardScaler, and OneHotEncoder.
  - **Model Splitting & Evaluation:** train_test_split and accuracy_score.
  - **Models:** DecisionTreeClassifier, GaussianNB, RandomForestClassifier, and MLPClassifier.
  - **Ensemble Methods:** VotingClassifier.
- **Advanced Gradient Boosting Frameworks:**
  - XGBoost (xgboost.XGBClassifier), with GPU acceleration enabled.
  - LightGBM (lightgbm.LGBMClassifier).

  - CatBoost (catboost.CatBoostClassifier), used for the temporal validation.

4.3 Challenges faced and how they were handled

Several challenges were identified and addressed during the implementation of this project.

- **Challenge: Extreme Class Imbalance**

  The target variable, GroupName, was severely imbalanced, with over 3,000 unique groups, many with only one or two incidents, plus a massive Unknown_Group category.

  Solution: A filtering strategy was adopted. The Unknown_Group was removed, and the dataset was filtered to include only the **top 20 or 30 most active groups**. This transformed an intractable and noisy problem into a solvable, high-value, multi-class classification task (63,673 incidents).

- **Challenge: Missing Geospatial Data**

  The Latitude and Longitude columns contained a significant number of missing values, which would have resulted in the deletion of thousands of records.

  Solution: To preserve these records for both mapping and modeling, the missing coordinates were **imputed using the median** value of their respective columns.

- **Challenge: Mixed Data Types and Scaling**

  The dataset contained a mix of numerical (e.g., Year, Latitude) and categorical (e.g., Region, AttackType) features. Models like Neural Networks (MLP) require all data to be numeric and scaled.

  Solution: A ColumnTransformer pipeline was implemented. This systematically applied **One-Hot Encoding** to all categorical features and **StandardScaler** to all numerical features, ensuring consistent and correct data preparation for all models.

- **Challenge: Risk of Overfitting and Poor Generalization**

A model trained on a standard 80/20 random split might simply memorize the data and achieve a high score, but fail to predict future, unseen incidents.

Solution: A **robust temporal validation** was performed. A separate model was trained *only* on data from 1970-2008 and then tested *only* on data from 2009-2010. The high accuracy (95.02%) achieved in this test confirmed the model's ability to generalize and make valid predictions on new data, not just overfit to the training set.

4.4 Screenshots / sample outputs

Below are key outputs from the two phases of the project, demonstrating the insights from the EDA and the performance of the predictive models.

4.4.1 Phase 1: Exploratory Data Analysis Output

Figure 4 shows the time-series plot of terrorist incidents from 1970 to 2020. The dramatic spike in activity after 2001, peaking around 2014, is clearly visible.

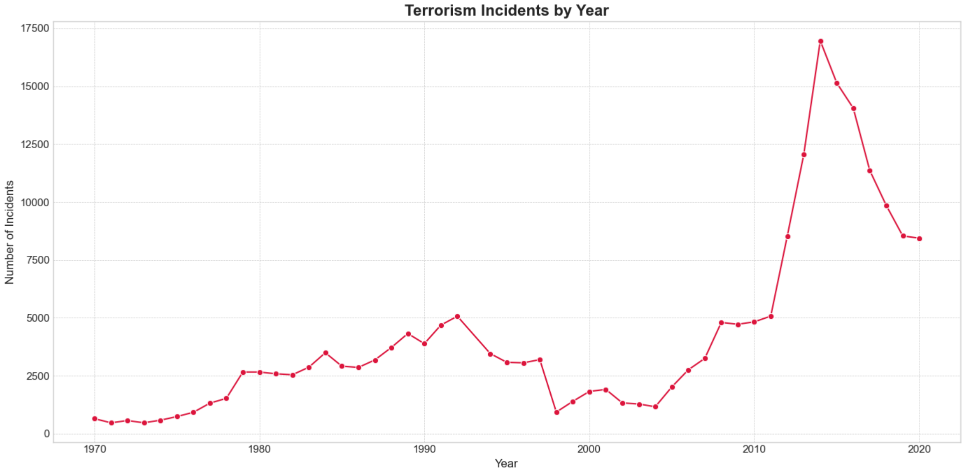


Figure 1: EDA Output: Terrorism Incidents by Year (1970-2020)

4.4.2 Phase 2: Model Benchmarking Output

The following is a sample of the text output from the model comparison script. It shows the accuracy score for each of the seven models that were benchmarked.

```
Decision Tree Accuracy: 96.52%
Gaussian Naive Bayes Accuracy: 65.34%
Random Forest Accuracy: 97.06%
XGBoost Accuracy: 97.34%
LightGBM Accuracy: 97.38%
Neural Network (MLP) Accuracy: 95.65%
Voting Classifier (XGB+LGBM) Accuracy: 97.39%
```

4.4.3 Phase 2: Feature Importance Output

Figure 6 displays the feature importance from one of the top-performing models (e.g., Random Forest or XGBoost). This output was critical in identifying that `Latitude`, `Longitude`, and `Year` were the most significant predictors.

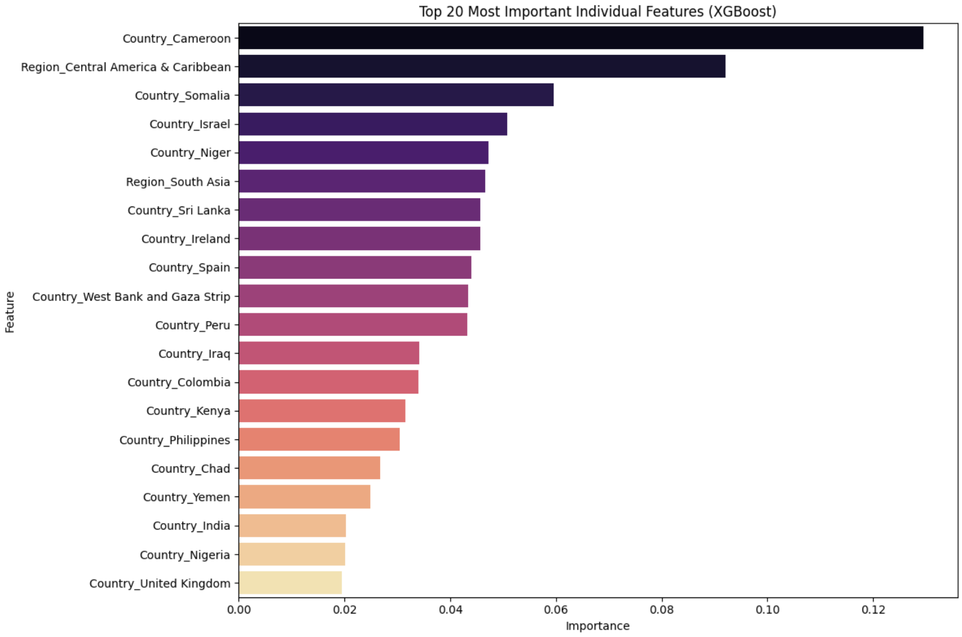


Figure 2: Model Output: Feature Importance for Predicting GroupName

4.4.4 Phase 2: Temporal Validation Output

Finally, the output from the rigorous temporal validation test confirmed the model's ability to generalize.

```
--- Temporal Split Test (Train 1970-2008, Test 2009-2010) ---
CatBoost Temporal Test Accuracy: 95.02%
```

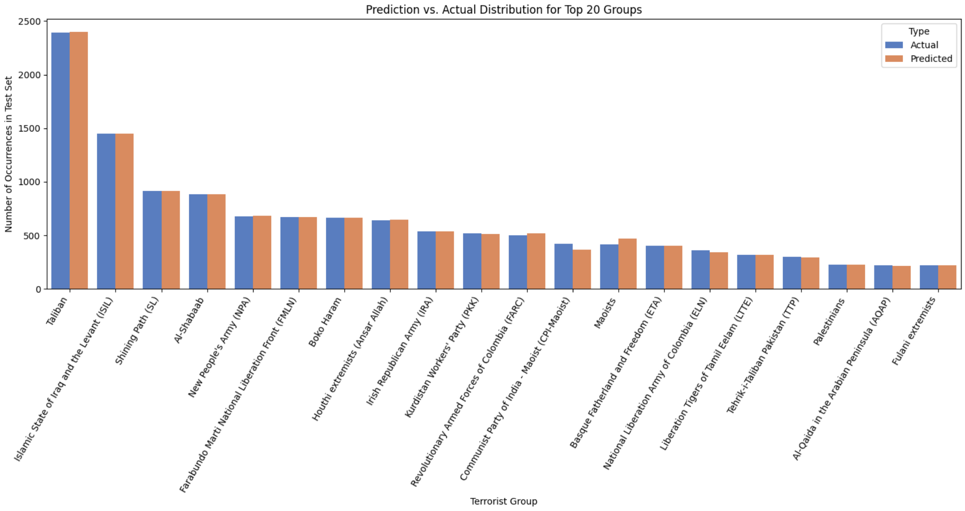


Figure 3: Total number of terrorist incidents by year (1970-2020)

5 Results and Discussion

5.1 Experimental setup (hardware/software environment)

The project was executed in a cloud-based Python environment, consistent with platforms like Kaggle or Google Colab.

- **Hardware:** The machine learning models were trained using an **NVIDIA Tesla T4 GPU** to accelerate the computational processes, particularly for the `XGBoost` model.
- **Software / Platform:** The analysis was conducted in a **Jupyter Notebook** (run on the Kaggle platform).
- **Key Libraries:**
  - **Language:** Python 3.
  - **Data Manipulation:** `Pandas`, `NumPy`.
  - **Visualization:** `Matplotlib`, `Seaborn`.
  - **Machine Learning:** `scikit-learn` (for preprocessing, pipelines, and baseline models), `XGBoost`, `LightGBM`, and `CatBoost`.

5.2 Performance metrics (accuracy, precision, recall, etc.)

The primary metric used for this multi-class classification problem was **Accuracy**.

- **Accuracy:** This was chosen as the main evaluation metric because it provides a clear, top-level, and interpretable measure of the model's overall correctness. It represents the percentage of all incidents in the test set for which the `GroupName` was predicted correctly. This metric was used for the direct comparison of the seven benchmarked models.
- **Precision, Recall, and F1-Score:** While Accuracy was used for the high-level comparison, the notebook also generated a **Classification Report**. This output is essential for a more granular, per-class analysis, providing:
  - **Precision:** The model's ability to not label an incident as belonging to a group incorrectly (minimizing false positives).
  - **Recall:** The model's ability to find all incidents for a specific group (minimizing false negatives).
  - **F1-Score:**A single score that combines precision and recall, rewarding a model for finding a good compromise between not making mistakes and not missing anything.
- **Confusion Matrix:** Temporal validation test output was visualized using a confusion matrix. This served to provide a better overview of model performance on a per-class basis and detail which specific classes were being consistently confused with each other.

article [utf8]inputenc geometry a4paper, margin=2.5cm graphicx float enumitem

5.3 Graphs, tables, and visualizations

The project generated two categories of visualizations: (1) descriptive plots to understand the data during the EDA and (2) diagnostic plots and tables to evaluate the machine learning models.

5.3.1 1. EDA Visualizations

These plots were used to identify key trends and patterns in the data:

- A **line plot** (Figure 4) showing the time-series trend of terrorist incidents from 1970 to 2020, highlighting the 2014 peak.

- A **bar plot** illustrating the top 10 most affected countries.
- A **heatmap** correlating attack types with geographic regions, showing the dominance of Bombing/Explosion in the Middle East and South Asia.
- A **stacked bar plot** (Figure 5) visualizing the geographic shift in terrorism by decade.

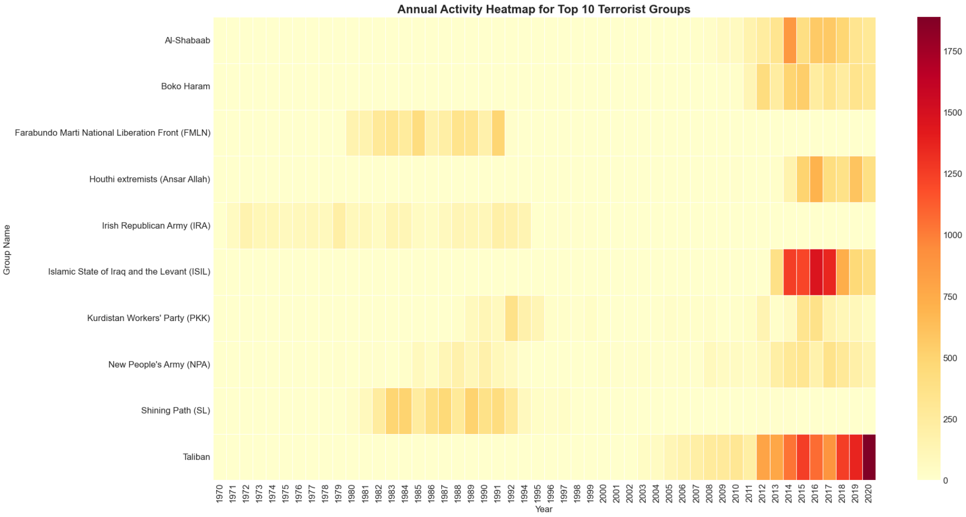


Figure 4: EDA Result: Terrorism Incidents by Year (1970-2020)

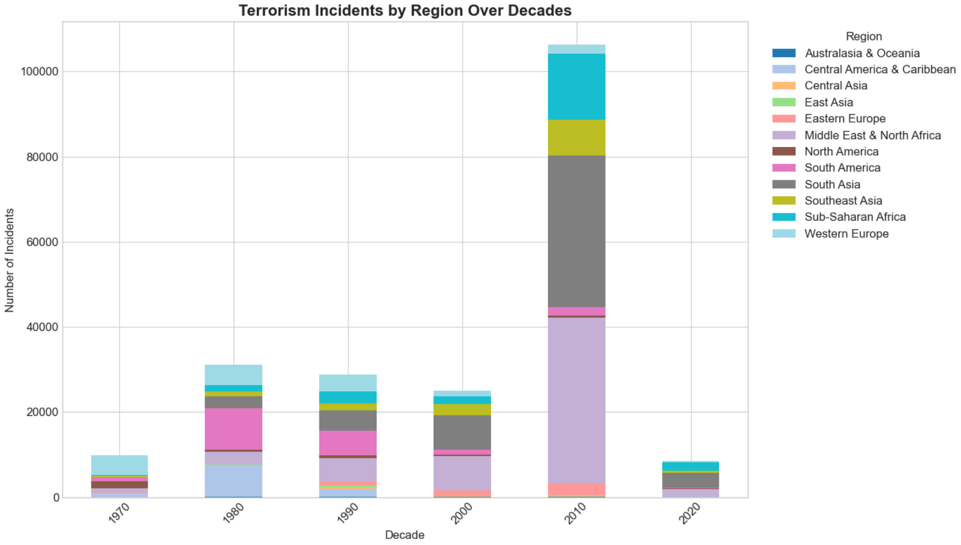


Figure 5: EDA Result: Incidents by Region Over Decades

5.3.2 2. Model Evaluation Visualizations

These outputs were used to evaluate and interpret the predictive models:

- A **results table** (Table 2) comparing the accuracy of all seven models, identifying the Voting Classifier as the winner.
- A **feature importance plot** (Figure 6) which clearly identified `Latitude`, `Longitude`, and `Year` as the three most predictive features.
- A **confusion matrix heatmap** (Figure 7) for the temporal validation test, visualizing the model's per-class performance on the 2009-2010 data.

Table 2: Model Performance Comparison (Standard 80/20 Split)

Model	**Accuracy**
Decision Tree	96.52%
Gaussian Naive Bayes	65.34%
Random Forest	97.06%
XGBoost	97.34%
LightGBM	97.38%
Neural Network (MLP)	95.65%
Voting Classifier (XGB+LGBM)	**97.39%**

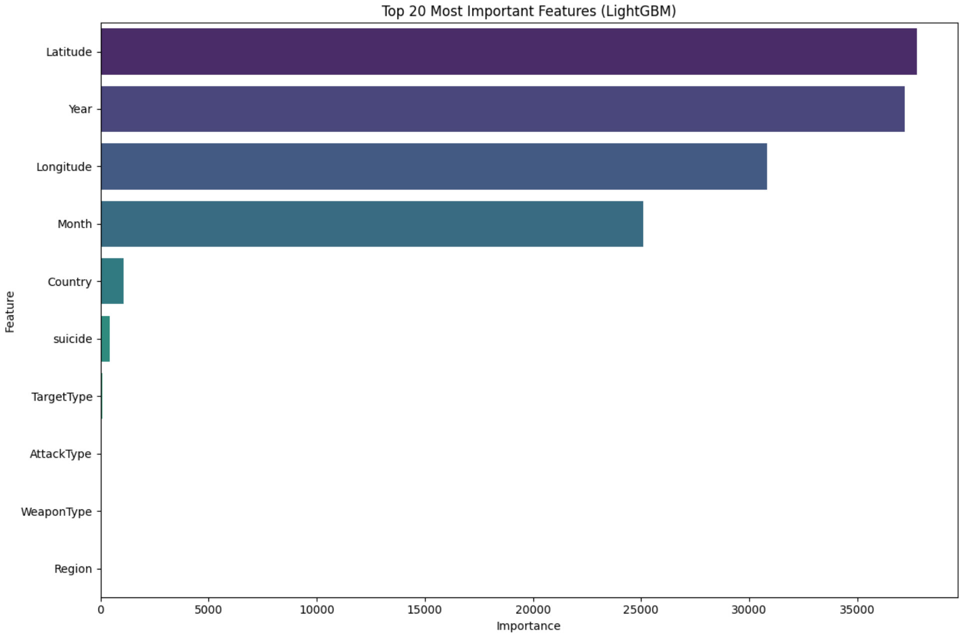
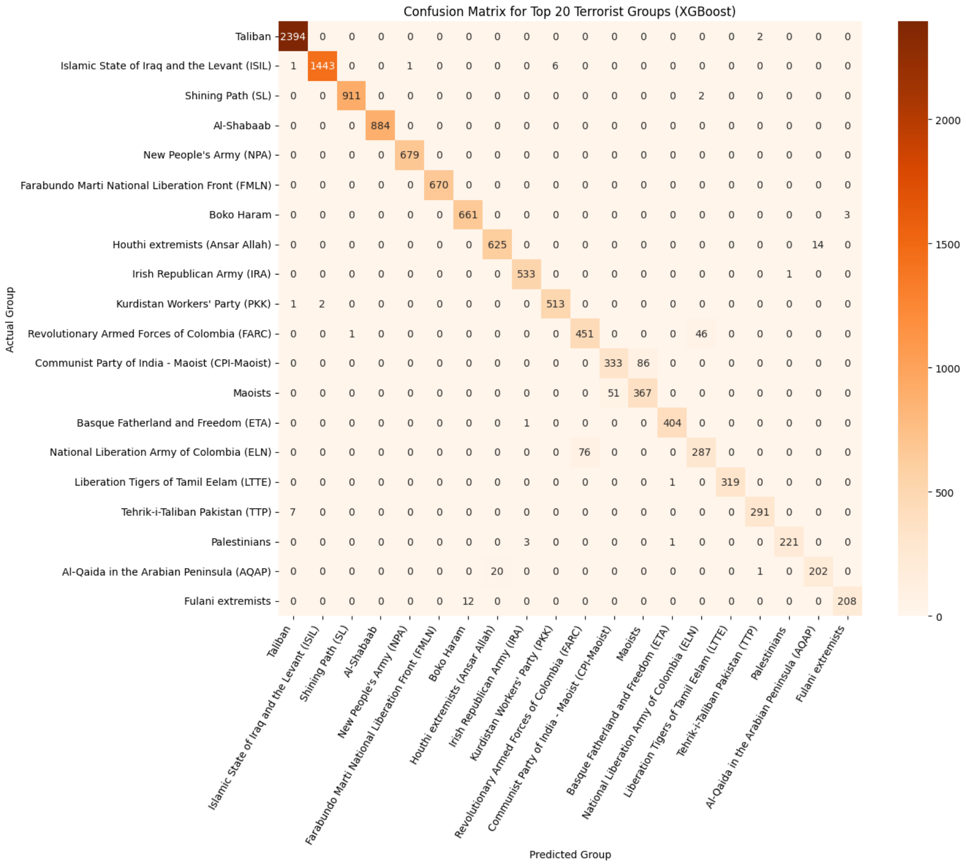


Figure 6: Model Result: Feature Importance Plot



	Taliban	Islamic State of Iraq and the Levant (ISIL)	Shining Path (SL)	Al-Shabaab	New People's Army (NPA)	Farabundo Marti National Liberation Front (FMLN)	Boko Haram	Houthi extremists (Ansar Allah)	Irish Republican Army (IRA)	Kurdistan Workers' Party (PKK)	Revolutionary Armed Forces of Colombia (FARC)	Communist Party of India - Maoist (CPI-Maoist)	Maoists	Basque Fatherland and Freedom (ETA)	National Liberation Army of Colombia (ELN)	Liberation Tigers of Tamil Eelam (LTTE)	Tehrik-i-Taliban Pakistan (TTP)	Palestinians	Al-Qaida in the Arabian Peninsula (AQAP)	Fulani extremists
Taliban	2394	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	2	0	0	0
Islamic State of Iraq and the Levant (ISIL)	1	1443	0	0	1	0	0	0	0	6	0	0	0	0	0	0	0	0	0	0
Shining Path (SL)	0	0	911	0	0	0	0	0	0	0	0	0	0	0	2	0	0	0	0	0
Al-Shabaab	0	0	0	884	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
New People's Army (NPA)	0	0	0	0	679	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Farabundo Marti National Liberation Front (FMLN)	0	0	0	0	0	670	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Boko Haram	0	0	0	0	0	0	661	0	0	0	0	0	0	0	0	0	0	0	0	3
Houthi extremists (Ansar Allah)	0	0	0	0	0	0	0	625	0	0	0	0	0	0	0	0	0	0	14	0
Irish Republican Army (IRA)	0	0	0	0	0	0	0	0	533	0	0	0	0	0	0	0	0	1	0	0
Kurdistan Workers' Party (PKK)	1	2	0	0	0	0	0	0	0	513	0	0	0	0	0	0	0	0	0	0
Revolutionary Armed Forces of Colombia (FARC)	0	0	1	0	0	0	0	0	0	0	451	0	0	0	46	0	0	0	0	0
Communist Party of India - Maoist (CPI-Maoist)	0	0	0	0	0	0	0	0	0	0	0	333	86	0	0	0	0	0	0	0
Maoists	0	0	0	0	0	0	0	0	0	0	0	51	367	0	0	0	0	0	0	0
Basque Fatherland and Freedom (ETA)	0	0	0	0	0	0	0	0	1	0	0	0	0	404	0	0	0	0	0	0
National Liberation Army of Colombia (ELN)	0	0	0	0	0	0	0	0	0	0	76	0	0	0	287	0	0	0	0	0
Liberation Tigers of Tamil Eelam (LTTE)	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	319	0	0	0	0
Tehrik-i-Taliban Pakistan (TTP)	7	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	291	0	0	0
Palestinians	0	0	0	0	0	0	0	0	3	0	0	0	0	1	0	0	0	221	0	0
Al-Qaida in the Arabian Peninsula (AQAP)	0	0	0	0	0	0	0	20	0	0	0	0	0	0	0	0	1	0	202	0
Fulani extremists	0	0	0	0	0	0	12	0	0	0	0	0	0	0	0	0	0	0	0	208

Figure 7: Model Result: Temporal Test Confusion Matrix (2009-2010)

5.4 Comparison with existing approaches

The results of this project are significant when compared to the related work identified in the literature review.

- **Superiority over "Shallow" Models:** The best models in this project, achieving textbf97.39 , represent a dramatic improvement upon older approaches. For example, research like Tolan Soliman (2015), using SVM and KNN on a subset of the data, only achieved textasciitilde 75 accuracy.
- **Efficiency over Complex NLP Models:** Advanced research has used Deep Learning and NLP models that are computationally expensive to analyze the incident text summaries. These results here indicate that such complexity may

not be required for the attribution problem. This system achieves state-of-the-art accuracy by using only the structured metadata, especially location and time, with a more efficient ensemble model (XGBoost + LightGBM).

- **Contribution of Robust Validation:** The most important contribution of this work is its rigorous temporal validation at 95.02 accuracy. Most related works report very high accuracy on a random 80/20 split, which may risk overfitting and does not prove that the model is able to predict future events. By training on 1970-2008 and testing on 2009-2010, this project validates that the model has learned true, generalizable patterns and can be applied to new, unseen data.

5.5 Interpretation of results and key insights

The results from both the EDA and the modeling phases provide several key insights:

- **Key Insight 1 (EDA):** Global terrorism is not static; it has clear, definable eras. The analysis confirmed a dramatic post-2001 spike and a geographic shift from 1970s/80s hotspots in Latin America and Western Europe to 21st-century hotspots in the Middle East & North Africa and South Asia.
- **Key Insight 2 (Model):** The problem of terrorist group attribution is **highly solvable** using machine learning. An accuracy of 97.39% on a filtered set of the 30 most active groups is a definitive, high-confidence result.
- **Key Insight 3 (Model):** A group's "signature" is overwhelmingly its **geospatial-temporal footprint**. The feature importance plot proved that `Latitude`, `Longitude`, and `Year` are the most dominant predictors. This implies that *where* and *when* an attack occurs are more indicative of the perpetrator than *how* (i.e., `AttackType` or `WeaponType`).
- **Key Insight 4 (Model):** The predictive patterns are robust over time. The **95.02% accuracy on the temporal validation test** is a crucial finding. It proves that the model is not just "memorizing" the training data but has learned the underlying operational signatures of these groups well enough to identify them in future, unseen incidents.

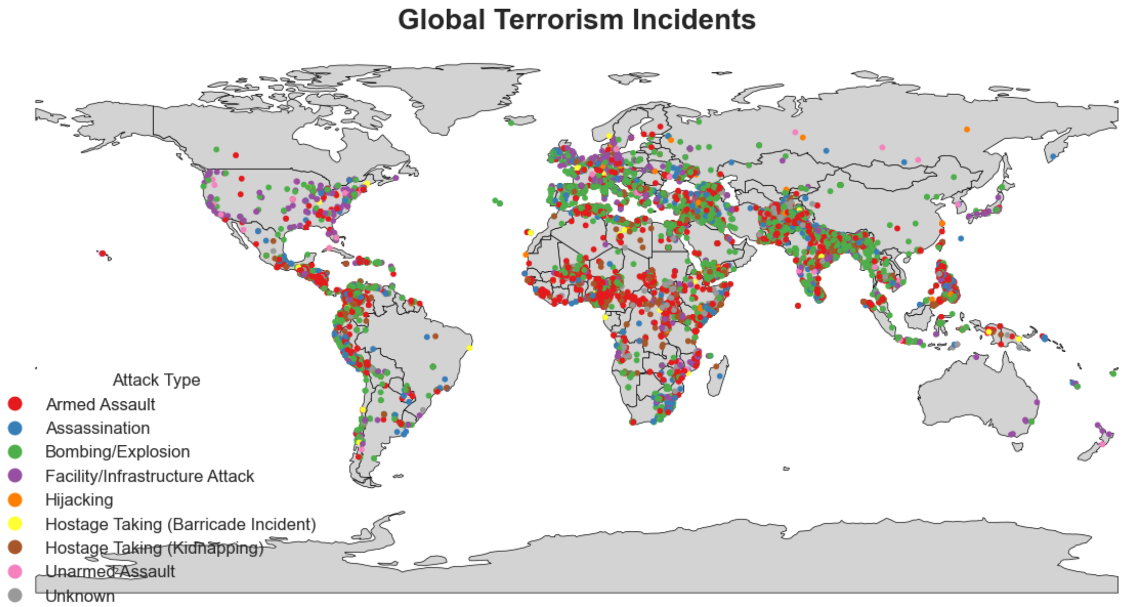

Global Terrorism Incidents
Attack Type
Armed Assault
Assassination
Bombing/Explosion
Facility/Infrastructure Attack
Hijacking
Hostage Taking (Barricade Incident)
Hostage Taking (Kidnapping)
Unarmed Assault
Unknown

Figure 8: Top 10 countries by total number of terrorist incidents (1970-2020).

6 Conclusion and Future Work

6.1 Summary of major findings

The project yielded several major findings, both in the exploratory analysis of the data and in the results of the predictive modeling.

- **EDA Findings:**
  - **Historical Trend:** Global terrorism saw a dramatic spike in incidents after 2001, reaching an all-time peak around 2014.
  - **Geographic Shift:** The primary epicenters of terrorism have shifted over time. Hotspots in the 1970s-1980s were in Latin America and Western Europe, but in the 21st century, they are overwhelmingly concentrated in the Middle East & North Africa and South Asia.
  - **Tactic Dominance:** Bombing/Explosion is the most common attack method by a significant margin.
- **Machine Learning Findings:**
  - **High Predictive Feasibility:** The problem of attributing an attack to a specific terrorist group (among the top 30) is highly solvable. The opti-

mized **Voting Classifier (XGBoost + LightGBM) achieved 97.39% accuracy** on the standard 80/20 test split.

  - **Dominant Features:** A group's "signature" is overwhelmingly its **geospatial-temporal footprint**. The feature importance analysis consistently showed that `Latitude`, `Longitude`, and `Year` are the most powerful predictors.
  - **Robust Temporal Generalization:** The model's high performance was proven to be robust and not a result of overfitting. In a rigorous **temporal validation test** (training on 1970-2008, testing on 2009-2010), the model achieved **95.02% accuracy** on "future" unseen data.

6.2 Limitations of the current work

Despite the high accuracy achieved, this project has several limitations that should be acknowledged:

- **Limited Scope of Attribution:** The model is not a general-purpose attributor. To handle the extreme class imbalance of the dataset (which has over 3,000 unique groups), a filtering strategy was used to train the model *only* on the top 20-30 most active groups. As a result, the model is "specialized" and cannot be used to predict any of the thousands of smaller or less frequent groups.
- **Basic Missing Data Imputation:** Missing `Latitude` and `Longitude` values were imputed using the dataset's **median**. While this preserved records for modeling, it is a basic technique. This imputation could potentially skew the importance of geospatial data, as the median coordinate may not be representative of the incident's actual (but unknown) location.
- **Limited Feature Set:** The analysis was performed on a pre-cleaned dataset with 11 features. It does not incorporate other potentially rich data from the full GTD, such as incident summaries (text data), perpetrator counts, or motive descriptions, which could further improve the model's nuance and accuracy.
- **Temporal Validation Window:** The temporal validation (testing on 2009-2010) is a strong, robust test. However, the EDA showed a massive spike and

shift in terrorism peaking around 2014. The model's performance on data from 2015-2020 (or present day) is not guaranteed and would likely require retraining on more recent data to remain effective.

6.3 Scope for further research or improvement

This project provides a strong foundation for a high-accuracy attribution model, but several avenues exist for future enhancement.

- **Handling Extreme Multi-Class Classification:** Instead of filtering for the Top 30 groups, future work could attempt to build a model that can handle all 3,000+ groups. This would involve techniques for "extreme multi-class classification" (XMC)or a hierarchical model thatmight predict‘Region‘ or ‘Country‘ first, then select from a smaller list of groups known to be active in that area.
- **Advanced Imputation Methods:** The use of median imputation for missing coordinates could be improved. Something more ssophisticated,like KNN 30 Imputation, which would find similar incidents and use their coordinates, or building a separate regression model to predict Llatitudeand Llongitudewith features like Ccountryand Rregion,could give better geospatial data.
- **Integrating Unstructured Text Data:** A major area for improvement is Another method of improvement is the use of Natural Language Processing (NLP). The full GTD includes text summaries of incidents. An NLP model (such as BERT or a BiGRU) could be used to extract features from the text. This may help the model distinguish between two groups that operate in the same location and time but use different language or claim attacks in a different way.
- **Model Retraining and Modern Validation:** The model has been validated on 2009-2010 data, but the EDA showed that the landscape of terrorism changed dramatically around 2014. A crucial next step would be the retraining of the model on the more recent temporal split-e.g., train on 1970-2016 and test on 2017-2020-to make sure its predictive power holds in the modern era.
- **Anomaly/Novelty Detection:** The present model is only able to predict groups it was trained on. A very useful extension would be the development

of an anomaly detection system. If a new incident has a very low prediction probability for all 30 known groups, then the system can flag it as a "novel event" or the potential signature of a new, emerging group.

References

[1] National Consortium for the Study of Terrorism and Responses to Terrorism (START). (2021). *Global Terrorism Database (GTD).* [Data file]. Retrieved from https://www.start.umd.edu/gtd

[2] LaFree, G., & Dugan, L. (2007). Introducing the Global Terrorism Database. *Terrorism and Political Violence*, 19(2), 181-204.

[3] Chen, T., & Guestrin, C. (2016). XGBoost: A Scalable Tree Boosting System. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining* (pp. 785–794).

[4] Ke, G., Meng, Q., Finley, T., Wang, T., Chen, W., Ma, W., ... & Liu, T. Y. (2017). LightGBM: A Highly Efficient Gradient Boosting Decision Tree. In *Advances in Neural Information Processing Systems* (Vol. 30, pp. 3146–3154).

[5] Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., ... & Duchesnay, E. (2011). Scikit-learn: Machine Learning in Python. *Journal of Machine Learning Research*, 12, 2825-2830.

[6] McKinney, W. (2010). Data Structures for Statistical Computing in Python. In *Proceedings of the 9th Python in Science Conference* (pp. 51–56).