

**PROJECT DOCUMENTATION**

**Terrorism prediction: A Challenge for the  
21st Century**

**Authored by  
Abu Bakar**

**Supervisor Sir Sami Ullah**

**Year**

**2024**

**Project Title**

# **Terrorism prediction: A Challenge for the 21st Century**

Project submitted to  
Department of Computer Science/ IT  
Gujrat Institute of Management Sciences

**By**

AbuBakar

In partial fulfillment of the requirements for the degree of  
(BSCS/IT (Hons.))

---

**Gujrat Institute of Management Sciences**  
**PMAS-Arid Agriculture University, Rawalpindi**

## Members' Detail

Project ID	(GIMS-BSCS-F20202)
------------	--------------------

Terrorism prediction: A Challenge for the 21st Century			
<b>Group Leader: Abubakar</b>			
<b>Group Members:3</b>			
Abu-Bakar	20-Arid-1660	baker4508@gmail.com	BSCS
Ameer Hamza	20-Arid-1668	<a href="mailto:askameerhamza.cs@gmail.com">askameerhamza.cs@gmail.com</a>	BSCS
Muhammad Ahmed Raza	20-Arid-1702	muhammadahmedraza20 arid1702@gmail.com	BSCS



**Sir Awais Ilyas Baig**

PMO, GIMS

## **Dedication**

We dedicate our dissertation work to our parents who taught us to trust Allah and believe in hard work. Their unconditional support makes us able to do our project.

## Project Summary

<b>Project Title</b>	Terrorism prediction: A Challenge for the 21st Century
<b>Project ID</b>	GIMS-BSCS-F20202
<b>Organization</b>	Gujrat Institute of Management Sciences
<b>Objective</b>	<ul style="list-style-type: none"><li>• Predict terrorist events</li><li>• Identify and assess the risk of terrorist attacks</li><li>• Comprehensive Reporting of Casualties</li></ul>
<b>Undertaken By</b>	Abu Bakar (20-Arid-1660), Ameer Hamza (20-Arid-1668), and Muhammad Ahmed Raza (20-Arid-1702)
<b>Supervised By</b>	Sir Sami Ullah
<b>Date started</b>	September 07, 2023
<b>Date Completed</b>	May 14, 2024
<b>Technologies Used</b>	PYTHON, R, STREAMLIT, VSCODE, RSTUDIO
<b>System Used</b>	Window 10 pro

## Proofreading Certificate

This is to acknowledge that the project entitled

Terrorism prediction: A Challenge for the 21st Century  
GIMS-BSCS- F20202

has been proofread

By



---


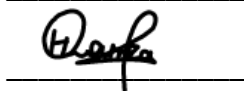

Mr. Samiullah

Coordinator of BSCS

Dated: \_\_\_\_\_

## Declaration

We hereby declare that we developed this project and this report entirely on the basis of our personal efforts made under the sincere guidance of our project supervisor. We further declare that, the titled project and all associated documents, reports are submitted as partial requirements for the degree of “BS (Hons.) in Computer Sciences/ IT – Masters in Computer Sciences/ IT”.

Members' Name	Registration #	Signature
Abu Bakar	20-Arid-1660	
Ameer Hamza	20-Arid-1668	
Muhammad Ahmed Raza	20-Arid-1702	

Dated: \_\_\_\_\_



## **Acknowledgement**

We thank Allah Almighty whose unlimited blessings made us capable to do hard work and achieve our life goals. We could not have done it without their support and supervision.

We would like to thank our parents and siblings to support us in hard times. Their unconditional love and support make us courageous to do our work. We are very thankful to Sir Shoaib who was always there to help us in a real sense during our project. we are very thankful to our friend Daniyal Arif who gives us support in every hard time. We are also grateful to our other friends Ahmad Faraz and Abdullah. They always stand with us in hard times and support us where it is needed.

We cannot forget to mention the name of our respected teachers Sir Samiullah, Sir Awais Illyas Baig and Sir Ashar Javed who remained with us all the time to make our minds motivated for the project.

## Certificate

This is to certify that **Abu Bakar, 20-Arid-1660, Ameer Hamza, 20-Arid-1668, and Muhammad Ahmed Raza, 20-Arid-1702** have successfully completed the final project titled: **“Terrorism prediction: A Challenge for the 21st Century”**, accepted by the Department of Computer Science/IT and find satisfactory for the requirement of:

**Gujrat Institute of Management Sciences**  
**PMAS-Arid Agriculture University Rawalpindi**

**For Award of the Degree**

**BSCS**



Supervisor

Mr. Sami Ullah

Lecturer, GIMS



Examiner 1

Lecturer, GIMS



Examiner 2

Lecturer, GIMS

  
HOD/CS-IT  
GIMS

Director General  
GIMS

Dated: \_\_\_\_\_

**Approval**

I Mr. / Ms. \_\_\_\_\_ Sir Sami Ullah \_\_\_\_\_ am willing to guide these students in all phases of project titled “. \_\_\_\_\_ Terrorism prediction: A Challenge for the 21st Century \_\_\_\_\_ ” as supervisor. I have carefully seen the Title and description of the proposal and believe that it is of an appropriate difficulty level for the number of students named above.

Supervisor



---

**Sir Sami Ullah**

Lecturer, GIMS

**Dated:** \_\_\_\_\_

**Abstract**

Terrorist attacks a serious threat to global security, and their prediction and prevention are imperative. The terrorist prediction system will be based on a machine learning approach called supervised learning. Supervised learning algorithms are trained on a set of labeled data, where each data point has an input and an output. Once the algorithm is trained, it can be used to predict the output for new data points that it has never seen before. This project aims to develop a novel terrorist prediction system based on model it can identify potential terrorist threats. The system will be trained on a large dataset of historical terrorist attacks and the system will then be able to identify individuals or groups who are at high risk of carrying out a terrorist attack, even if they have not yet engaged in any overt terrorist activity. as it will be trained on a larger and more comprehensive dataset of data. The system will be able to identify potential terrorist threats earlier, as it will be able to identify groups who are at high risk of carrying out a terrorist attack. The system will be more efficient and cost-effective, as it will be able to automate many of the tasks that are currently performed manually by human analysts. The system would help to prevent terrorist attacks by identifying potential terrorist threats and allowing law enforcement agencies to intervene early. The system would also help to reduce the fear and anxiety that is associated with terrorism, and would allow people to live their lives more safely and securely. The system will be trained on a subset of the data, and then evaluated on the remaining subset of the data. This process will be repeated several times to ensure that the system is generalizing well to unseen data. Once the system has been evaluated and shown to be effective, it will be deployed to a production environment so that it can be used by law enforcement agencies to prevent terrorist attacks. The successful development and implementation of the proposed terrorist prediction system would have a significant impact on global security

<b>Abstract.....</b>	<b>3</b>
<b>Table of Contents .....</b>	<b>4</b>
<b>List of Tables .....</b>	<b>6</b>
<b>List of Figures.....</b>	<b>7</b>
<b>Chapter 1 Introduction.....</b>	<b>10</b>
1.1 Problem Statement .....	10
1.2 Significance.....	11
1.3 Objectives .....	11
1.4 Limitations and Restrictions .....	12
1.5 Overview :.....	12
1.6 System Architecture .....	14
1.7 Software/Hardware Requirements .....	15
1.8 Implementation Tools and Technology .....	15
1.9 Implementation Plan .....	15
1.9.1 Deliverable Items.....	16
1.9.2 Milestone Chart.....	16
<b>Chapter 2 Requirement Analysis.....</b>	<b>17</b>
2.1 Functional Requirements .....	18
2.2 Non-Functional Requirement.....	18
2.3 Use Cases .....	19
2.3.1 Use-Case for User's Signup.....	19
2.3.2 Use-Case for User's Login.....	20
2.3.3 Use-Case for Admin's Login.....	21
2.3.4 Use-Case for Logout.....	22
<b>Chapter 3 DESIGN .....</b>	<b>23</b>
3.1 ERD.....	24
3.1.1 ERD Diagram.....	24
3.2 Use-Case Diagram .....	25
3.2.1 Use-Case Diagram for Sign in.....	25

3.2.2 Use-Case Diagram for SignUp.....	26
3.2.3 Use-Case Diagram for System.....	27
3.2.4 Use-Case Diagram for Register.....	28
3.2.5 Use-Case Diagram for LogOut.....	29
3.4 Sequence Diagram .....	30
3.4.1 Sequence Diagram for Register.....	30
3.4.2 Sequence Diagram for User's Login.....	31
<b>Chapter 4 Material and Method .....</b>	<b>32</b>
4.1 Data Description .....	33
4.2 Data Preprocessing.....	33
4.3 List of Variables and Short Description .....	34
4.4 EDR (Exploratory Data Analysis) .....	36
4.4.1 Analysis of Global Attack .....	36
4.4.2 The top 10 most active and violent groups .....	40
<b>Chapter 5 Statistical Testing .....</b>	<b>41</b>
5.1 Correlation Test.....	42
5.2 Hypothesis Testing.....	43
5.2.1 ANOVA Test.....	44
5.2.2 PostHoc Test.....	44
5.3 Results interpretation .....	46
<b>References:.....</b>	<b>100</b>

## **List of Table**

Table 2.1: Use Case to User Signup.....	19
Table 2.2: Use Case to User Login.....	20
Table 2.3: Use Case for Admin's Login.....	21
Table 2.4: Use-Case to User Logout.....	22
Table 4.1: List of variables and short description.....	34
Table 5.1: Posthoc test (lsd, scheffe, bonf).....	45
Table 5.2: PostHoc test with Tukey HSD for pair of groups.....	45

## List of Figures

Figure 1.1	Model-View-Controller.....	14
Figure 1.2	Machine Learning Implementation Method.....	15
Figure 1.3	Gantt chart for milestones.....	16
Figure 3.1	Erd diagram .....	24
Figure 3.2	Use case diagram sign-in .....	25
Figure 3.3	Use case diagram sign-up.....	26
Figure 3.4	Use case diagram machine learning model.....	27
Figure 3.5	Use case diagram Register.....	28
Figure 3.6	Use case diagram for Logout.....	29
Figure 3.7:	Sequence Diagram for Register.....	30
Figure 3.8:	Sequence Diagram for Login.....	31
Figure 4.1	Attack Frequency by Year & Region.....	36
Figure 4.2	Yearly casualties due to terrorism.....	37
Figure 4.3	Yearly suicide events .....	37
Figure 4.4	region wise distribution of terrorist event.....	38
Figure 4.5	By Attack Type.....	38
Figure 4.6	By Weapon Type.....	39
Figure 4.7	By Target Type.....	39



Figure 4.8	Top 10 Most Active & Violent Groups.....	40
Figure 4.9	Threat Level in Africa Middle-East.....	40
Figure 5.1	Correlation Web Plot.....	42
Figure 5.2	Boxplot of Groups vs Fatalities.....	43
Figure 5.3	Anova Test Output.....	44

# Chapter 1

## Introduction

## Chapter 1

### Introduction

#### 1: Introduction

Terrorism is a complicated and multidimensional phenomenon, making accurate forecasting difficult. Machine learning advancements have made it feasible to create systems that can recognize patterns and trends in data that can be related to terrorism. A terrorism prediction system is a type of software that studies data from many sources, such as historical terrorism data, using the model a machine learning algorithm that is used to select key features and estimate relationships among events in a network machine learning and AI to determine the likelihood of terrorist strikes. These technologies can help law enforcement and intelligence organizations with their investigations, resource allocation, and decision-making regarding the best ways to stop terrorist acts. The aim of this project is to develop a terrorism prediction system that can accurately identify and assess the risk of terrorist attacks. The system will be used to help law enforcement and intelligence agencies prevent terrorist attacks and protect the public. The intended audience for the terrorism prediction system is law enforcement and intelligence agencies. The system will be used by these agencies to inform their investigations, allocate resources, and make decisions about how to best prevent terrorist attacks. The scope of the project is to develop a terrorism prediction system that can be used to predict terrorist attacks anywhere in the world. The system will be developed using a variety of data sources, including historical terrorism data, open-source intelligence, and social media data.

#### 1.1. Problem Statement

The universe is a vast and complex place, filled with both beauty and danger. To maintain balance, it is sometimes necessary to make difficult choices. One such choice is the development of a terrorism prediction system. This system would use machine learning to analyse data from a source, including historical terrorism data to identify and assess the risk of terrorist attacks. If we don't have a terrorism prediction system, then we are fighting a blindfolded war. We are fighting an enemy that we cannot see, and we cannot predict their next move. This is a dangerous situation, and it puts us all at risk. A terrorism prediction system would give us the ability to see the future, and to

identify and prevent terrorist attacks before they happen. Imagine that a group of terrorists is planning an attack on a major city. If we had a terrorism prediction system, it would be able to identify this activity and alert the authorities. The authorities could then take steps to prevent the attack, such as arresting the terrorists or stopping them from accessing their weapons. without a terrorism prediction system, we are vulnerable to attack.

## **1.2. Significance**

Terrorism prediction systems are significant because they have the potential to save lives and prevent catastrophic attacks. By identifying and assessing the risk of terrorist attacks, these systems can help law enforcement and intelligence agencies to take preventive action. We are using model it is a powerful tool that can be used to understand complex phenomena and predict future events. It is particularly well-suited for applications in the areas of security and counterterrorism Imagine a group of terrorists is planning to bomb a major city. They are communicating with each other online and using social media to spread their propaganda. A terrorism prediction system would be able to identify this activity and alert the authorities. The authorities could then take steps to prevent the attack, such as arresting the terrorists or stopping them from accessing their weapons. In addition to saving lives, terrorism prediction systems can also have a significant economic impact. The cost of terrorism attacks is high, both in terms of human life and financial resources. Terrorism prediction systems can help to reduce these costs by preventing attacks from happening in the first place.

## **1.3. Objectives**

- ✓ Predict terrorist events
- ✓ Identify and assess the risk of terrorist attacks
- ✓ Comprehensive Reporting of Casualties

## 1.4 Limitations and Restrictions:

Accuracy: Terrorism prediction systems are not perfect, and they can generate false positives and false negatives. This means that they may identify terrorist attacks that are not actually going to happen, or they may fail to identify terrorist attacks that are actually going to happen.

- Bias: Terrorism prediction systems are trained on data that is collected from the real world. If this data is biased, then the prediction system will also be biased. This means that the system may be more likely to identify terrorist attacks in certain groups or communities than in others.
- Transparency: Terrorism prediction systems are often complex and opaque, making it difficult to understand how they work and why they make the predictions that they do. This lack of transparency can make it difficult to trust the system and to use its predictions effectively.
- Ethics: Terrorism prediction systems raise a number of ethical concerns. For example, there is the concern that these systems could be used to oppress the innocent or to justify violence against certain groups of people. There is also the concern that these systems could be hacked or misused.

## 1.5 Overview

---

### Project Goal:

The end goal is to create a world where everyone feels safe and respected. A world where there is no terrorism with the help of machine-based learning. A world where everyone has the opportunity to reach their full potential. If we can use these systems to identify and prevent terrorist attacks, we can save lives and make the world a safer place. But it's important to remember that these systems are only as good as the data we put into them. And we need to make sure that we're using them in a responsible and ethical way.

---

Type of project: ☒ R&D

---

---

**Project Success criteria:**

The system must be able to identify potential terrorist suspects and targets with an accuracy of at least 70%.

The system must be able to generate reports on the risk of terrorist attacks that are accurate, timely, and useful to decision-makers.

The system must be able to be integrated with other law enforcement and intelligence systems with minimal disruption.

The system must be easy to use and maintain, and it must be available to users 24/7.

---

**Risks of the Project:**

(Please mark <input checked="" type="checkbox"/> where applicable)	Low	Medium	High
Technical risk		<input checked="" type="checkbox"/>	
Timing risk			<input checked="" type="checkbox"/>
Budget risk	<input checked="" type="checkbox"/>		

---

**Target End users:**

Law enforcement agencies

Intelligence agencies

Military organizations

---

**Development Technology/ Languages:**

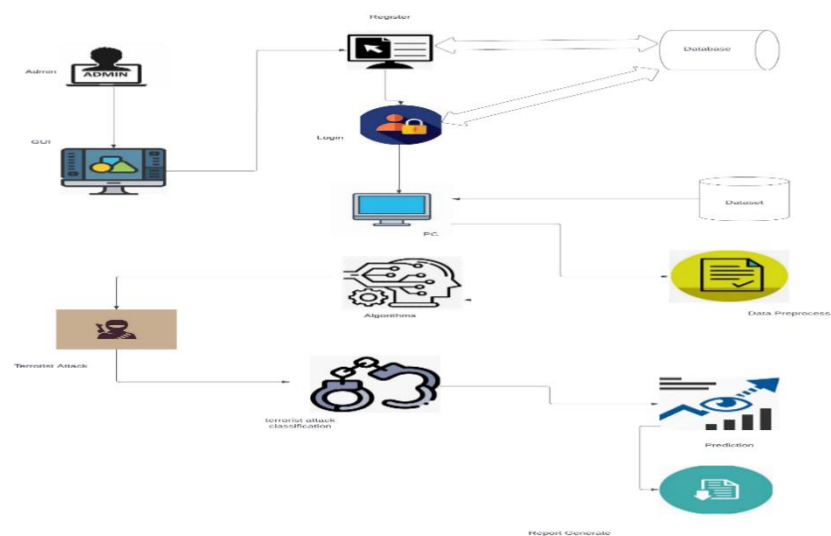
- ✓ PYTHON
  - ✓ R
  - ✓ STREAMLIT
  - ✓ VSCODE
  - ✓ RSTUDIO
-

---

**Platform:**

- ☒ Web based      ☐ Distributed      ☐ Setup Configurations  
☐ Desktop based      ☐ Android      ☐ iOS  
☐ Other \_\_\_\_\_
- 

## 1.6 System Architecture



*Figure 0.1: Model-View-Controller*

## 1.7 Software/Hardware Requirements

- **RAM:** - Minimum 4gb.
  - **Processor:** - intel core i3 or later.
- Software:** -
- **Operating System:** - Windows 10 or later
  - **Browser:** - Google
  - **Jupyter notebook**
  - **Vscode**
  - **R Studio**

## 1.8 Implementation Tools and Technology

1. PYTHON STREAMLIT/FLASK for Front-End.
2. Jupyter notebook python for model training

## 1.9 Implementation Plan

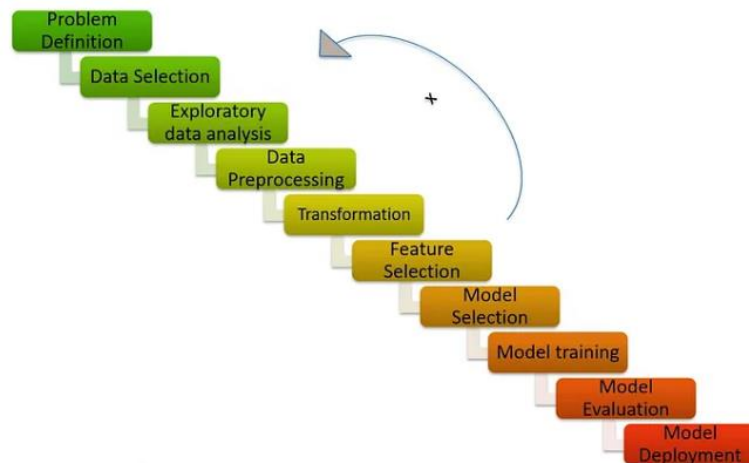


Figure 0.2: Machine Learning Implementation Method

### 1.9.1 Deliverable Items

- Project Proposal
- Project Presentation and Final Documentation
- Executable File

### 1.9.2 Milestone Chart

#### Gantt chart

Sr. No.	TASK NAME	START DATE	END DATE	DURATION	JUL	AUG	SEP	OCT	NOV	DEC	JAN	FEB	MARCH	MAY	JUNE
1	Project Proposal	24/07/2023	09/08/2023	2 weeks	■	■									
2	Documentation	10/08/2023	28/09/2023	7 weeks		■	■	■							
3	Dataset Collection	29/09/2023	07/12/2023	10 weeks			■	■	■	■					
4	Design & Coding Web Application	08/12/2023	22/02/2024	12 weeks						■	■	■			
5	Integration and Testing	23/02/2024	04/05/2024	10 weeks							■	■	■	■	
6	Deployment and Real-world Testing	05/05/2024	29/06/2024	8 weeks										■	■

Figure 0.3: Gantt chart for milestones



# Chapter 2

## Requirement Analysis

## **2.1 Functional requirements**

### **1. Admin Can Sign Up:**

- Enables new administrators to register and create accounts in the prediction system.

### **2. Admin Can Login:**

- Allows registered administrators to access the prediction system through secure login authentication.

### **3. Analysis:**

- Involves detail analytical procedures for casualties and threat detection.

### **4. Prediction:**

- Utilizes collected data and analysis to predict potential security threats.

### **5. Report Generation:**

- Generates comprehensive reports summarizing future security incidents, and system performance for analysis and review.

### **6. Ability to Filter Results by Location,Date:**

- Provides users with the capability to refine and organize data analysis results based on location, time, and for the efficient investigation and review.

## **2.2 Non-Functional Requirement**

All non-functional requirements of proposed system are as followed:

- Application availability will be 24/7 in a year. It means user can use this platform at any time anywhere.
- Application security is good. User data is secure and can be seen by only that user.
- Interactive Design/Usability is good. It is easy to use this application.
- Performance/Efficiency of this application is good. It does not take long time to open and do not hang on any browser.
- This application secures data of each user and do not involve third party to use his all-user data. It means privacy of this application is good.

## 2.3 Use Cases

### 2.3.1 Use-Case for User's Signup

Table 0.1: Use-Case to User Signup

ID and Name	UC 1: Sign Up
<b>Primary Actor:</b>	User
<b>Description:</b>	User want to access system features.
<b>Trigger:</b>	User want to register into the system to access features.
<b>Pre-Condition:</b>	The user must be on the registration screen and have a valid email and internet connection. The user must have a mobile device.
<b>Post-Condition:</b>	User registered successfully.
<b>Normal Flow:</b>	<ol style="list-style-type: none"><li>1. Applicant chooses that he is a user or lobby admin.</li><li>2. Applicant enters his personal information first name, last name, email, password, confirms the password and clicks on register account.</li><li>3. System redirects the user to the login screen.</li><li>4. Applicant can now log in to his/her account.</li></ol>
<b>Extensions (Error Scenarios):</b>	<ol style="list-style-type: none"><li>1. Applicant may leave some fields empty.</li><li>2. The system shows an error message that all fields are required.</li><li>3. Password and confirm password do not match.</li><li>4.Account with this email already created.</li></ol>

### 2.3.2 Use-Case for User's Login

Table 0.2: Use-Case to User Login

ID and Name	UC 2: Login
<b>Primary Actor:</b>	User
<b>Description:</b>	Users can log in to the system to search for predicting the attack
<b>Trigger:</b>	The user wants to login into the system.
<b>Pre-Condition:</b>	The user is already registered. So, the user enters a valid email and the correct password.
<b>Post-Condition:</b>	User login successfully.
<b>Normal Flow:</b>	<ol style="list-style-type: none"><li>1. User enters email and password.</li><li>2. System verifies data entered by the user.</li><li>3. System redirects the applicant to the home screen.</li></ol>
<b>Extensions (Error Scenarios):</b>	<ol style="list-style-type: none"><li>1. 1a. Applicant may leave any field empty. The system shows the error that an email or password is required.</li><li>2. 2a. Applicant enters the wrong password or email. The system shows the error that an incorrect password or email was entered.</li></ol>

### 2.3.3 Use-Case for Admin's Login

Table 0.3: Use-Case for Admin's Login

<b>ID and Name</b>	<b>UC 3: Admin Login</b>
<b>Primary Actor:</b>	Admin
<b>Description:</b>	Admin can login into the system by using his credentials.
<b>Trigger:</b>	Admin wants to login into the system to use system features.
<b>Pre-Condition:</b>	The credentials (email & password) of the admin must be created, so he/ she can login into the system.

<b>Post-Condition:</b>	The admin will be successfully logged in into the system.
<b>Normal Flow:</b>	<ol style="list-style-type: none"><li>1. Admin enters his/her email and password.</li><li>2. System verifies the data and the admin will successfully login into the system.</li></ol>
<b>Extensions (Error Scenarios):</b>	<ol style="list-style-type: none"><li>1. 2a. Password or email can be incorrect.</li><li>2. 2b. System will redirect the admin to the login screen.</li></ol>

#### 2.3.4 Use-Case for Logout

*Table 0.4: Use-Case to User Logout*

<b>ID and Name</b>	<b>UC 4: Logout</b>
<b>Primary Actor:</b>	<b>User/Admin/Expert</b>
<b>Description:</b>	<b>User/Admin</b> can logout of the system.
<b>Trigger:</b>	<b>User/Admin</b> wants to logout of the system.
<b>Pre-Condition:</b>	<b>User/Admin</b> must be logged in.
<b>Post-Condition:</b>	<b>Exit from application</b>
<b>Normal Flow:</b>	Admin have clicks on the logout button.

### 2.3.5 Use-Case for Forget Password

*Table 0.5: Use-Case for Forget Password*

<b>ID and Name</b>	<b>UC 5: Forget Password</b>
<b>Primary Actor:</b>	<b>User/Admin</b>
<b>Description:</b>	<b>User/Admin</b> can Forget password of the system.
<b>Trigger:</b>	<b>User/Admin</b> wants to Reset password of the system.
<b>Pre-Condition:</b>	<b>Admin &amp; User is already registered &amp; on login screen where forget password is available.</b>
<b>Post-Condition:</b>	Reset password successfully.
<b>Normal Flow:</b>	Admin have clicks on the Forget Password button.

### 2.3.6 Use-Case for Prediction

*Table 0.6: Use-Case for Prediction*

<b>ID and Name</b>	<b>UC 6: Prediction</b>
<b>Primary Actor:</b>	<b>User/Admin</b>
<b>Description:</b>	User/Admin can predict the attack using web app.
<b>Trigger:</b>	<b>User/Admin wants to predict the attack.</b>
<b>Pre-Condition:</b>	<b>Admin and user are already logged in and on dashboards where prediction parameters are given.</b>
<b>Post-Condition:</b>	Attack predicts successfully.
<b>Normal Flow:</b>	System have successfully predicted the attack.



### 2.3.7 Use-Case for Report Generator

*Table 0.7: Use-Case for Report Generator*

<b>ID and Name</b>	<b>UC 7: Report Generator</b>
<b>Primary Actor:</b>	<b>User/Admin</b>
<b>Description:</b>	After successfully predicting the attack using the web application, the User/Admin can generate a report summarizing the prediction results.
<b>Trigger:</b>	<b>User/Admin wants to generate a report summarizing the predicted attack.</b>
<b>Pre-Condition:</b>	<b>Admin and user are already logged in and on dashboards where prediction parameters are given. Attack prediction has been successfully completed.</b>
<b>Post-Condition:</b>	Report is generated summarizing the prediction results.
<b>Normal Flow:</b>	Report is displayed to the User/Admin for review and download.

# Chapter 3

## Design

### 3.1 ERD

#### 3.1.1 ERD Diagram

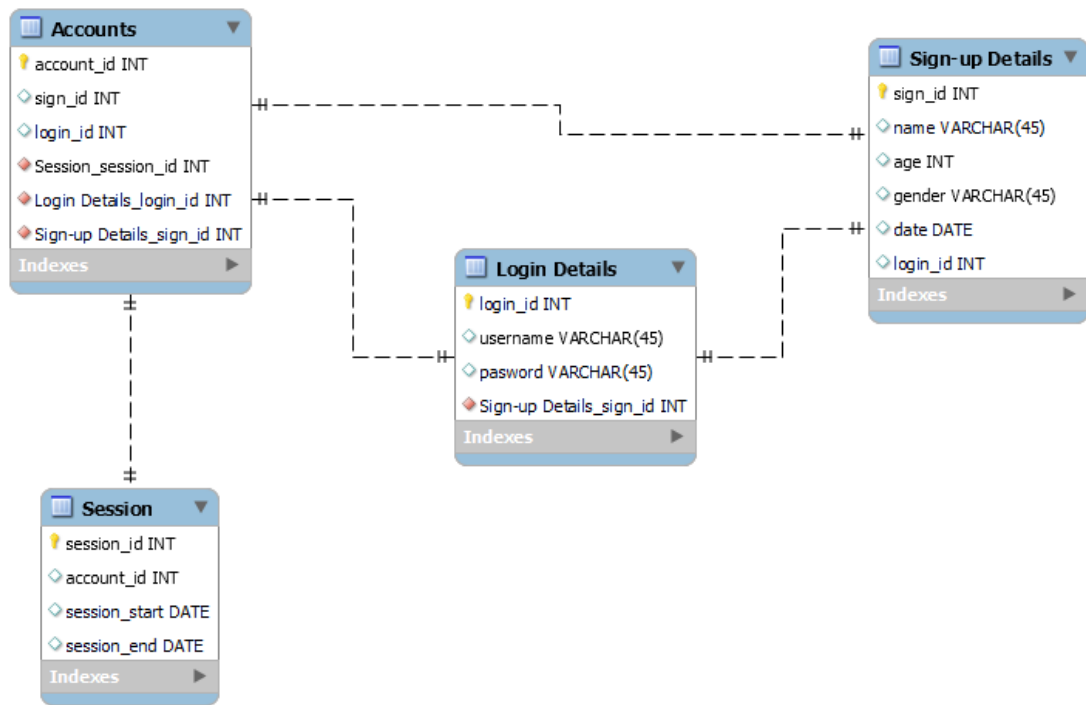
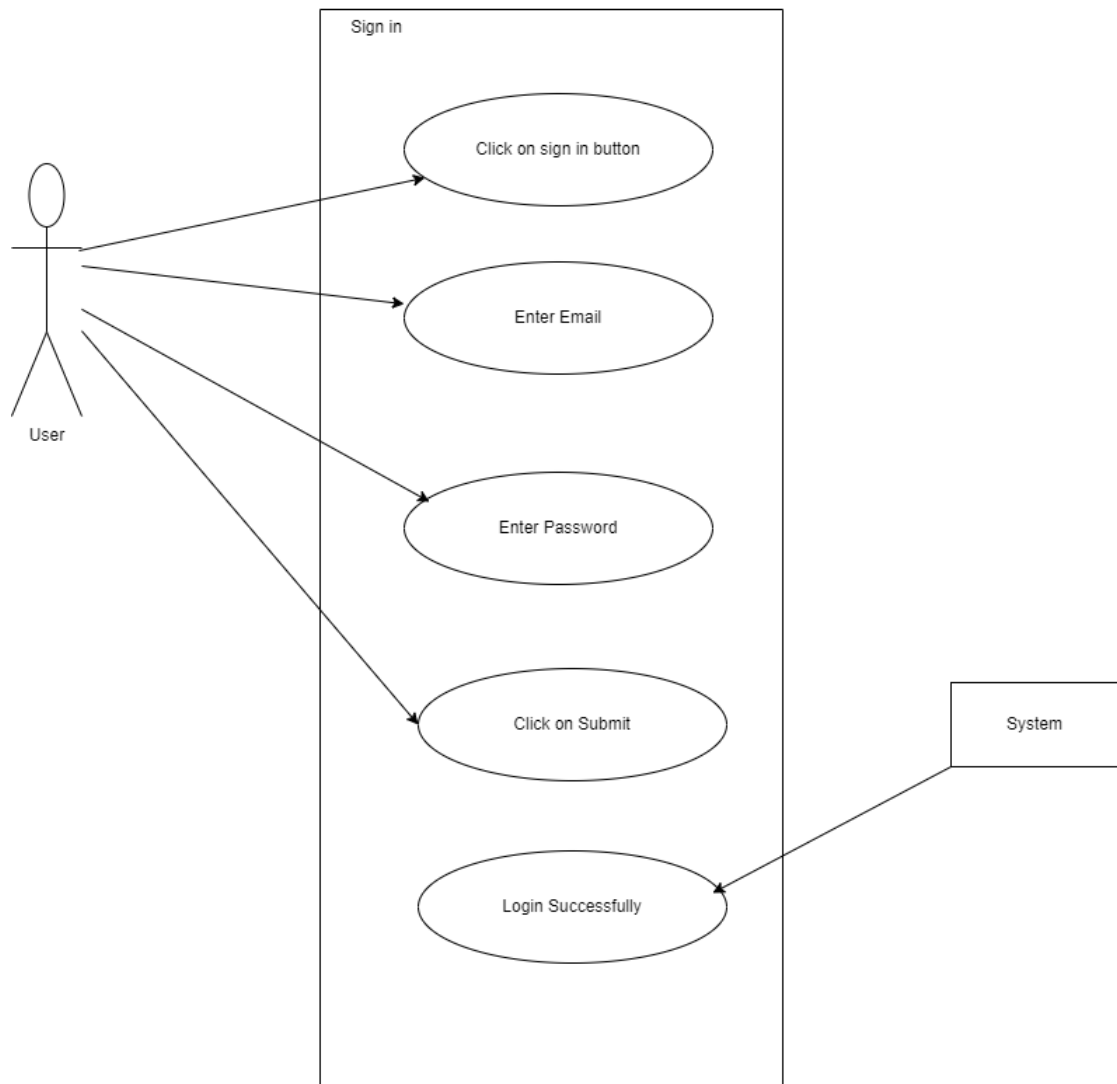


Figure 3.1 Erd diagram

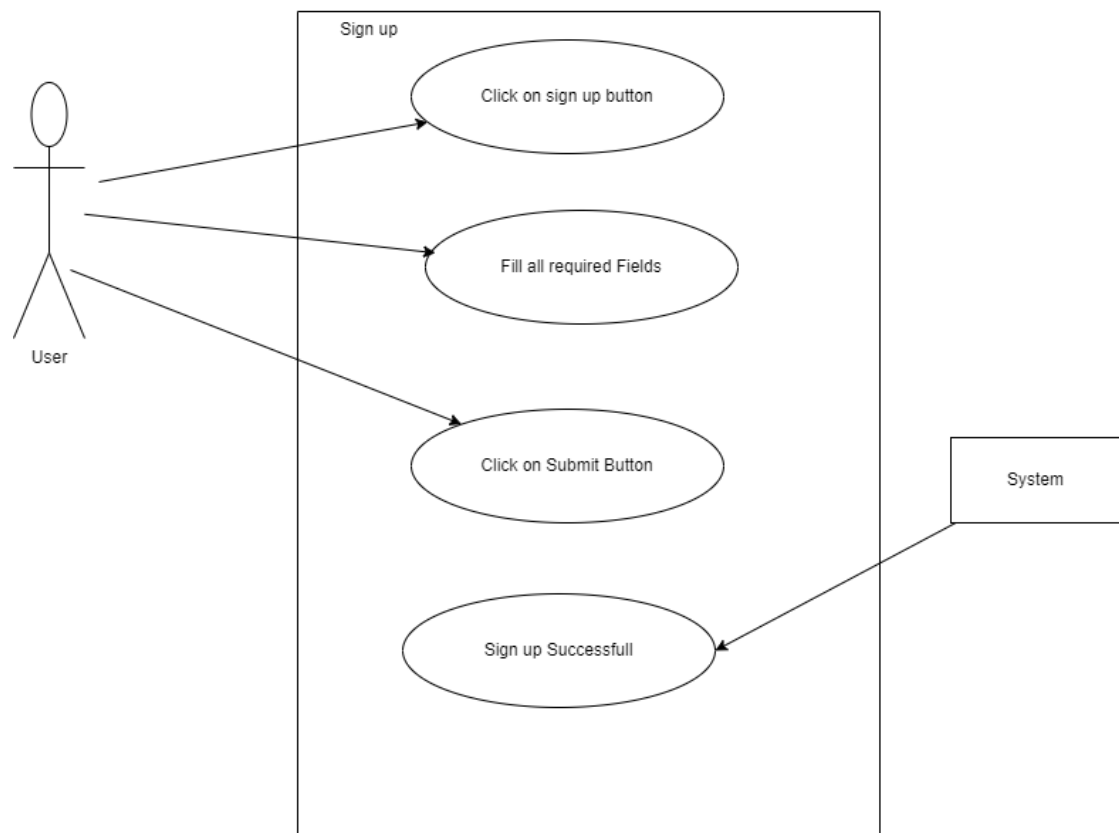
## 3.2: Use-Case Diagram

### 3.2.1: Use-Case Diagram for Sign in



*Figure 3.2 sign-in use case*

### 3.2.2 Use-Case Diagram for Sign Up



*Figure 3.3 sign-up*

### 3.2.3: Use-Case Diagram for System

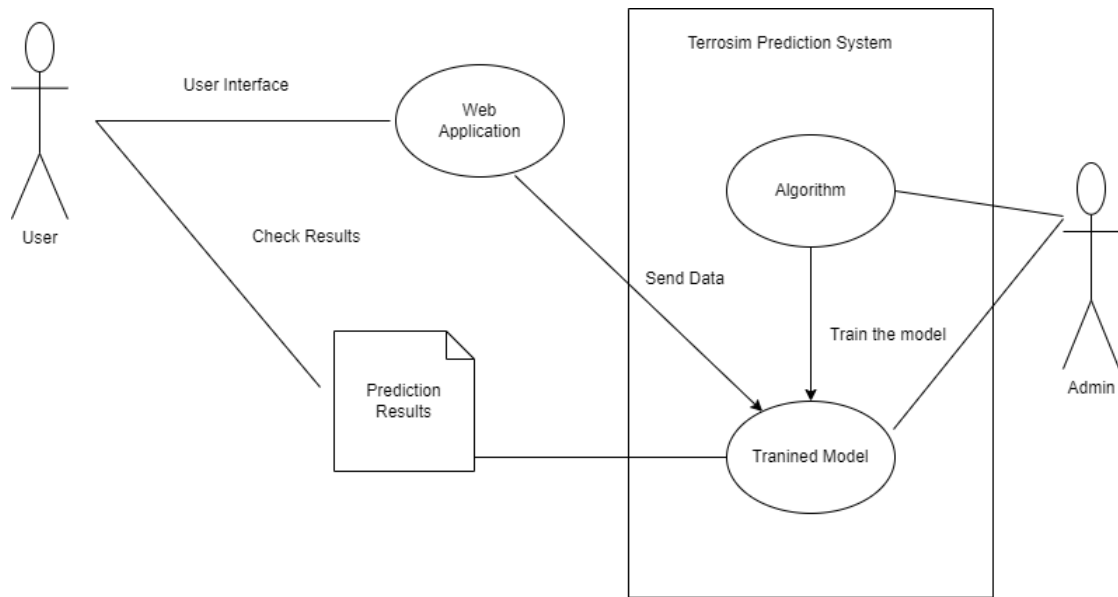
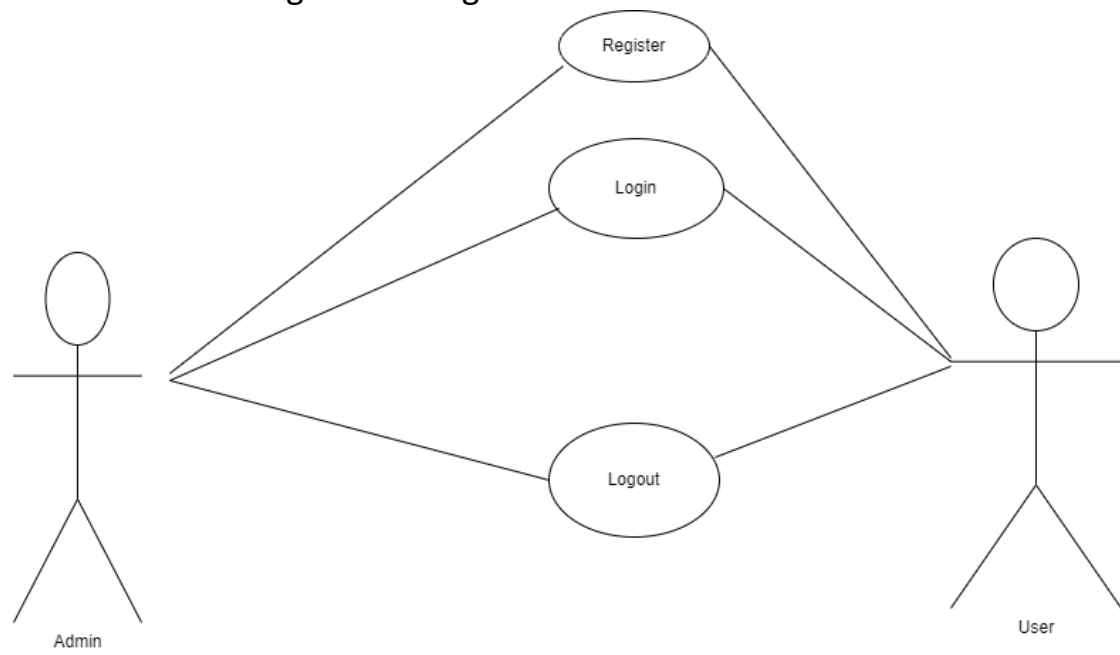


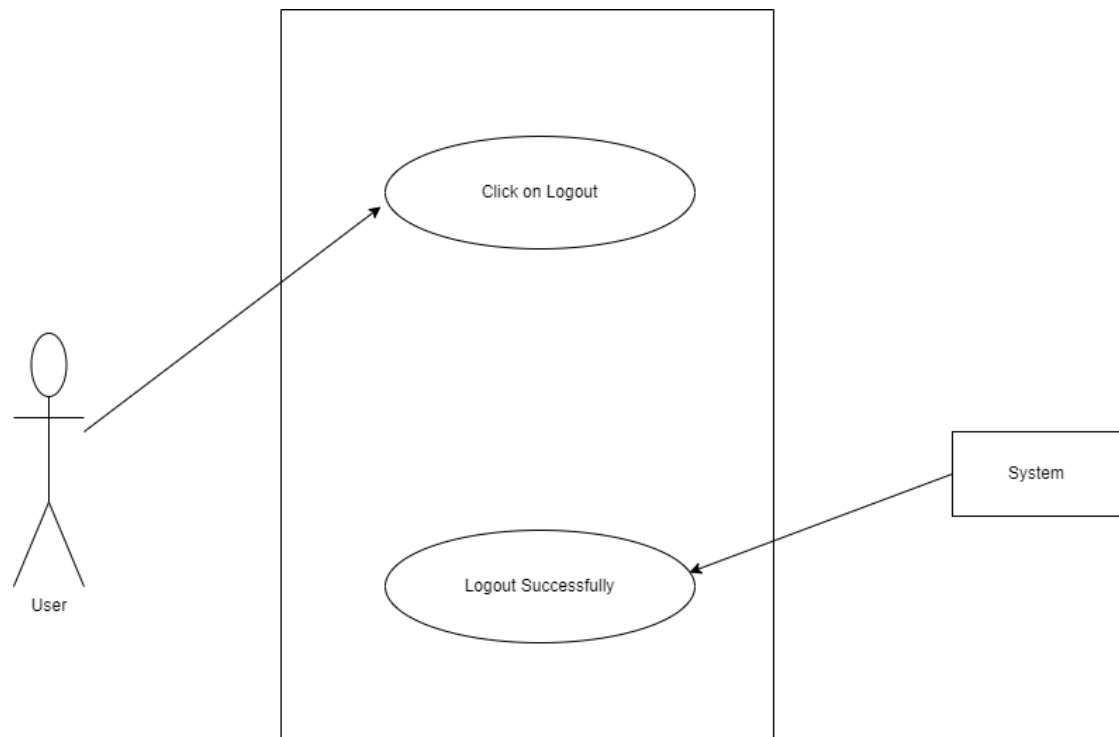
Figure 3.4 Use case diagram machine learning model

### 3.2.4 Use-Case Diagram for Register



*Figure 3.5 Use case diagram*

### 3.2.5 Use-Case Diagram for Logout



*Figure 3.6 Use case diagram for Logout*



### 3.3 Sequence diagram

#### 3.3.1: Sequence Diagram for Register

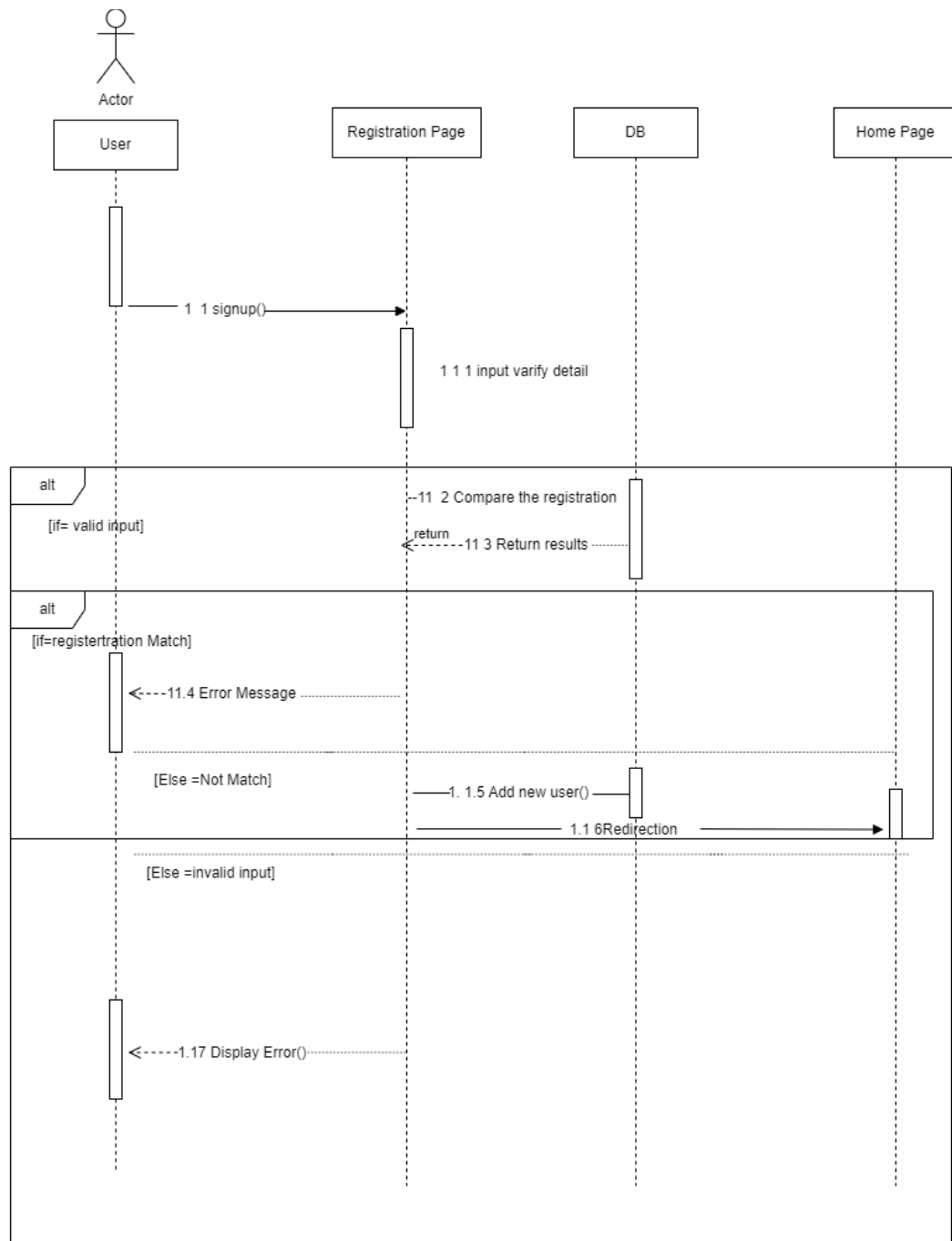


Figure 3.7: Sequence Diagram for Register

### 3.3.2:Sequence Diagram for Login

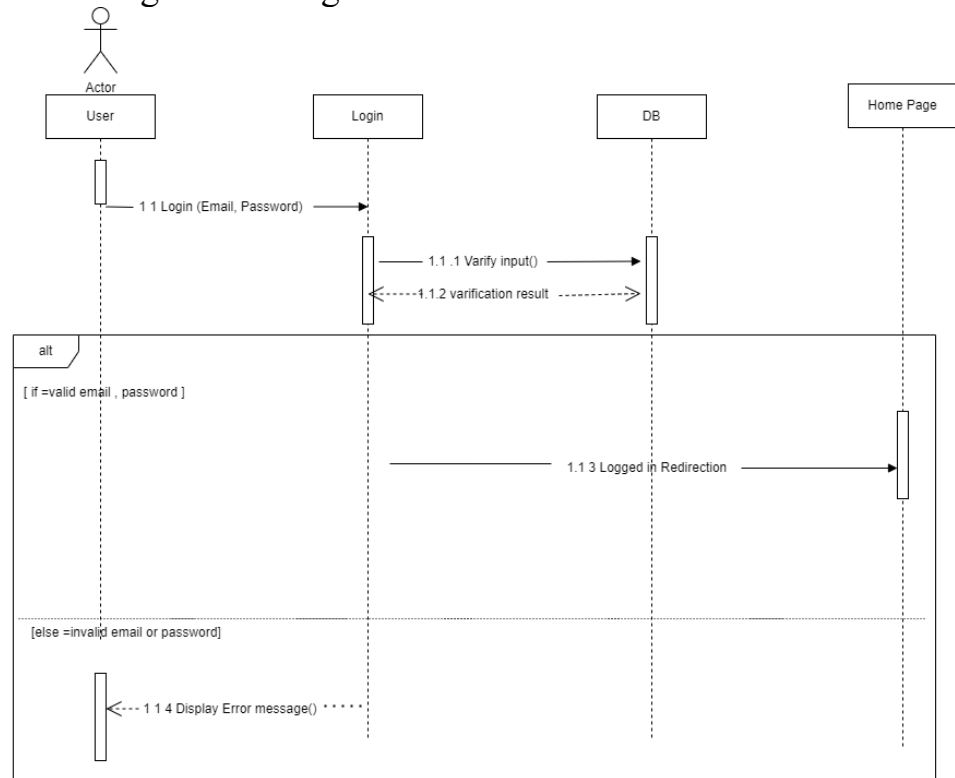


Figure 3.8: Sequence Diagram for Login

# **Chapter4**

## **Material and method**

## 4.1 Data Description:

The main dataset globalterrorismdb\_0617dist.xlsx we are using in this project have a total of 170000 terror attack information starting from 1970 to 2016 we have downloaded this file from a gtd website the dataset contains 130 variables also known as columns of a data set variables are labels as date and event id.

The type of attack the kind of weapon used in attack target and victim info there are many additional info are present in dataset which we will discuss later

We have found 38 columns which is valuable for our project we have also made some changes in data set for our analysis which we will explain in next section.

## 4.2 Data Preprocessing:

The names of variables are hard to understand and it get confusing so we rename some of the variables for example we rename

- attack\_success = success,
- suicide\_attack = suicide,
- individual\_attack = individual,
- intl\_logistical\_attack = INT\_LOG,
- intl\_ideological\_attack = INT\_IDEO
- A column named casualties was created by combining killed and wounded columns
- Replacing null values of latitude and longitude which is main variable of our project to geocodes of those country they represent these variables are from those countries which no longer exist Czechoslovakia.
- There are zero duplicated rows in dataset.
- There are 13853997 missing values in the dataset.
- Some of the null values represent in nkill means total number of people killed and null values of nwound total number of people wounded in attack these null values are added because of they are too vague to remains blank.

### 4.3 List of Variables and Short Description

Name of the Variable	Description
Eventid	a 12-digit Event ID
Year	year in which the incident occurred
Month	Month
Day	Day
Country	Country
Region	world region
Provstate	an administrative division or unit of a country
City	City
Latitude	Latitude
Longitude	Longitude
attack_type	method of attack (reflects the broad class of tactics used)
weapon_type	type of weapon used in the incident
target_type	type of target/victim
target_nalty	nationality of the target that was attacked
group_name	name of the group that carried out the attack
Nkill	number of total confirmed fatalities for the incident
Nwound	number of confirmed non-fatal injuries
Extended	whether or not an incident extended more than 24 hours
crit1_pol_eco_rel_soc	political, economic, religious, or social goal
crit2_publicize	intention to coerce, or publicize to larger audience

crit3_os_intl_hmn_law	action from the incident is outside intl humanitarian law
part_of_multiple_attacks	whether an incident being part of multiple attacks
attack_success	suicide attack
suicide_attack	whether an incident was successful
individual_attack	whether an attack carried out by unaffiliated Individual(s)
intl_logistical_attack	cross border incident
intl_ideological_attack	attack on target of a different nationality
ISO	ISO code for country
Date	Approx. date of incident
arms_export	Arms exports (SIPRI trend indicator values)
arms_import	Arms imports (SIPRI trend indicator values)
Population	Population, total
gdp_per_capita	GDP per capita (constant 2010 US\$)
refugee_origin	Refugee population by country or territory of origin
refugee_asylum	Refugee population by country or territory of asylum
net_migration	Net migration
n_peace_keepers	Presence of peace keepers
conflict_index	Extent of conflict-of-interest regulation index (0-10)

*Table 4.1 List of variables and short description*

## 4.4 EDR (exploratory data analysis)

### 4.4.1 Analysis of Global Attack

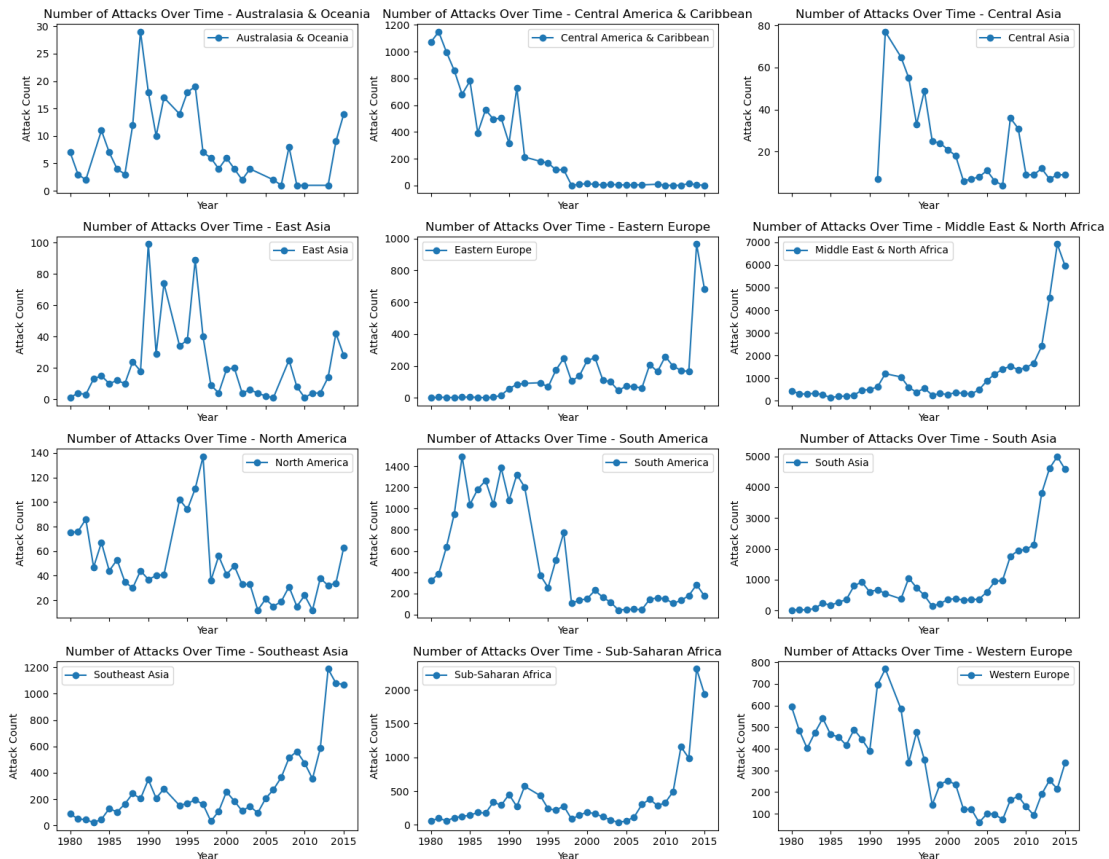
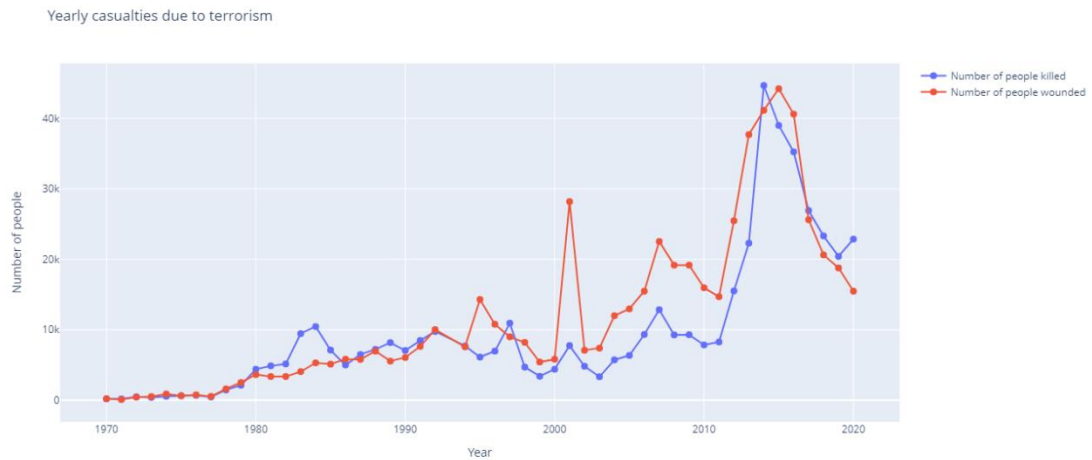


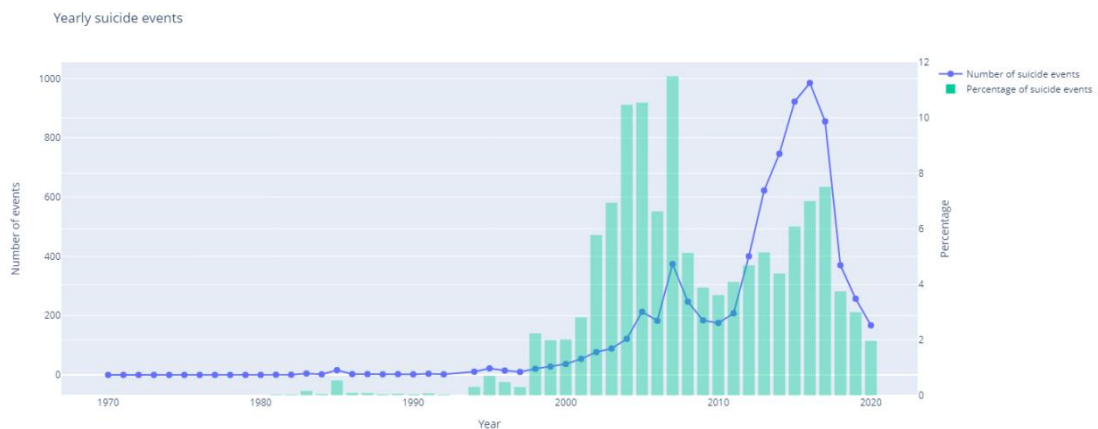
Figure 4.1 Attack Frequency by Year & Region

The number of attacks in the Eastern Asian region has fluctuated over time, but the overall trend has been upwards. The sharpest increase in the number of attacks occurred in the early 2000s. This is likely due to the rise of terrorism in countries such as Afghanistan and Pakistan. The number of attacks in the Eastern Asian region has declined slightly in recent years. We made an observation that Eastern Europe region there is a quick rise in a number of attacks can be seen in year 2014-2015 and then a quick low in 2016. In the most contacted regions, the nearly similar shift of gradual rise in a number of attacks after 2010 and peak during 2014-2015 is visible. It's worth mentioning that in June 2014, Islamic State announced the establishment of “Caliphate” while declaring Abu Bakr al-Baghdadi as “leader of Muslims everywhere” and urging other groups to pledge allegiance (Al Jazeera, 2014)



*Figure 4.2 Yearly casualties due to terrorism*

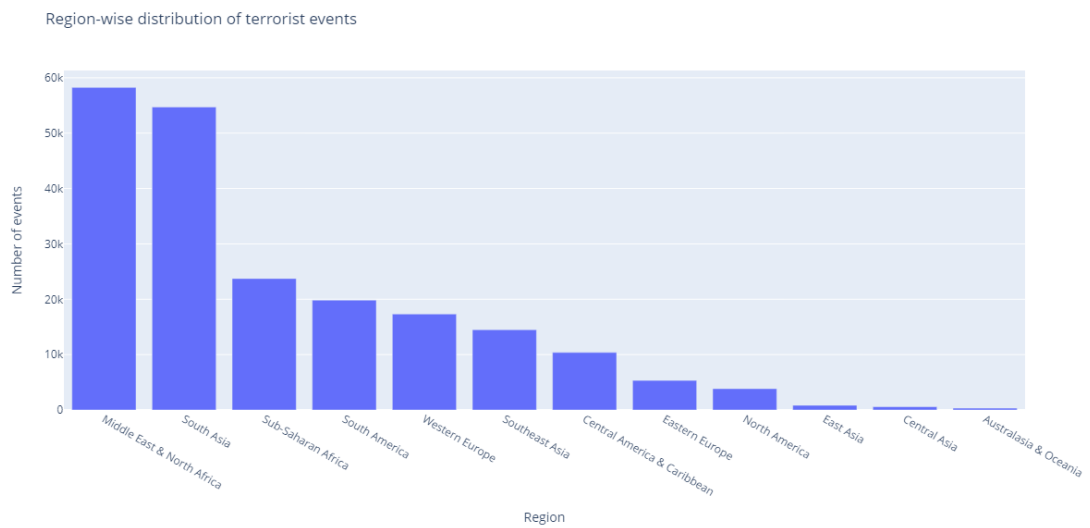
It is observed that the number of people killed and wounded due to terrorism has increased over time. In 1970, there were over 4000 people killed and wounded due to terrorism. However, by 2020, the number of people killed and wounded due to terrorism had increased to less than 4,000. we also observed that after 2010 the quick rise of lines



*Figure 4.3 yearly suicide events*

Events which involve **suicide attacks**. The number of successful attacks has also reduced significantly after 2006, number of suicide attacks has increased after 2000 which was almost rare before 2000.

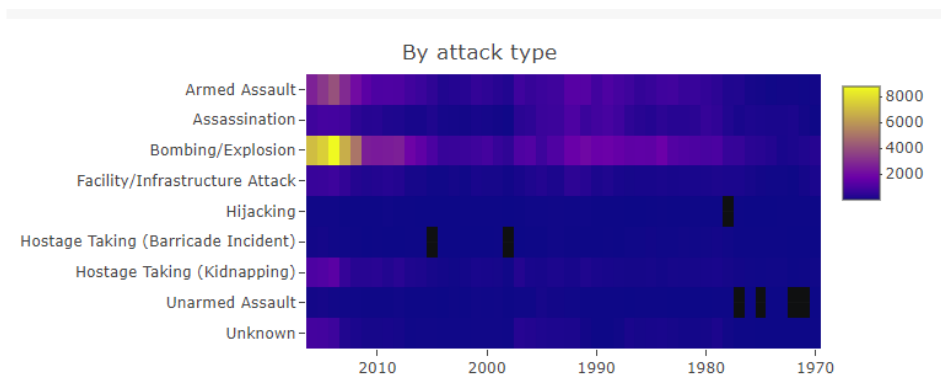




*Figure 4.4 region wise distribution of terrorist event*

To understand the attack characteristics, let's take a look at Frequency of attack type and type of weapon used by terrorist groups.

To visualize the attack feature, we built heat map at Frequency of attack type and type of weapon used by terror organization.



*Figure 4.5: Attack Type*

The heat signatures show Bombing and Explosive as one of the frequently used techniques by terrorist groups. Although the pattern in this tactic is visible throughout all the year, while rising during the late 80s and early 90s however it has now increased to nearly 7 times since 2006.

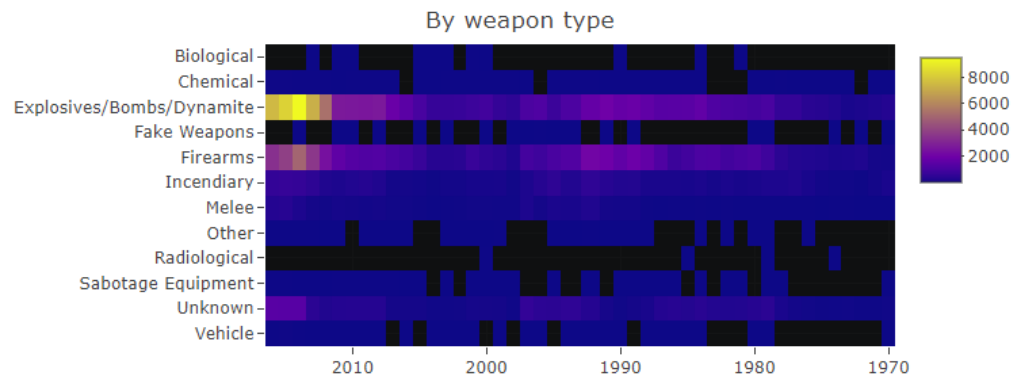


Figure 4.6 By Weapon Type

It is observed that use of Explosives/Bomb/Dynamites and Firearms is extremely high since 2011 and compared to other weapon types. Use of vehicles as weapon type was relatively low until 2013, however, it was on peak in 2015 with total 34 number of attacks.

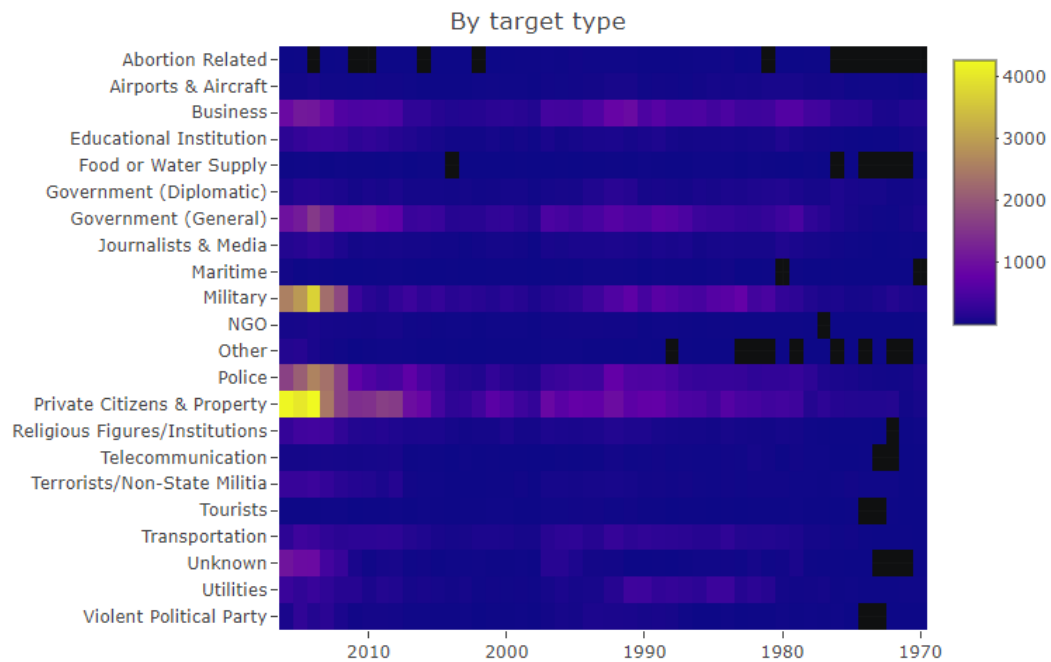


Figure 4.7 By Target Type

The most common target of terrorist attacks is private citizens and property, followed by government (general), business, and transportation. The number of attacks on each type of target has fluctuated over time, but there has been a general increase in the number of attacks on all types of targets since the 1970s. Military and private citizens and property attacks increased after 2010 up to 3000.

#### 4.4.2 The top 10 most active and violent groups

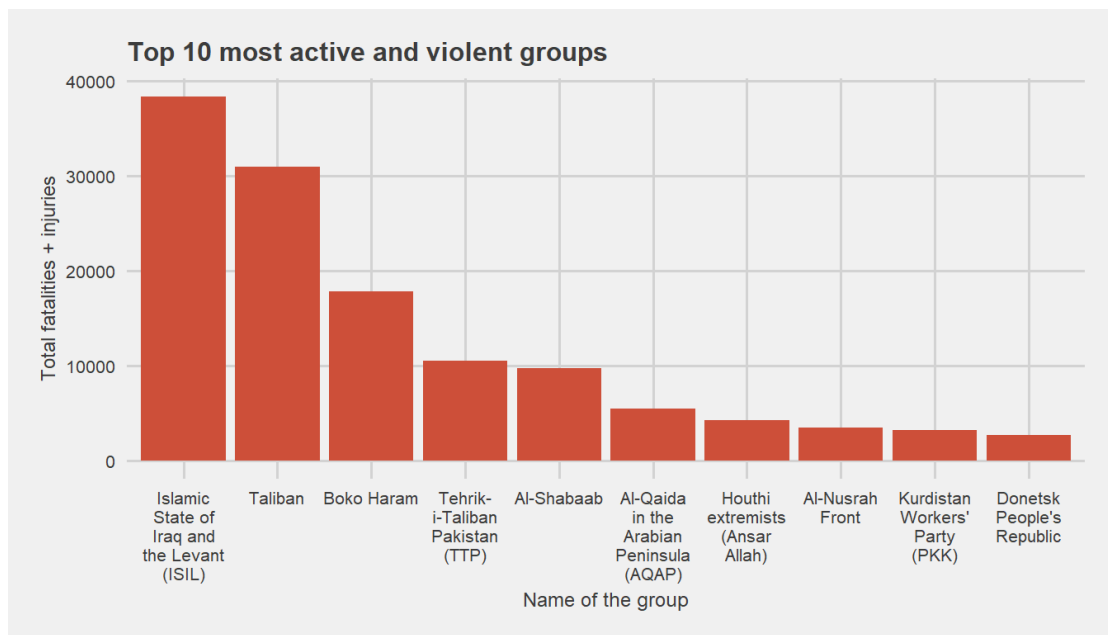


Figure 4.8: Top 10 Most Active & Violent Groups

we can see that ISIL and Taliban followed by Boko Haram are the most violent groups that are currently active. To better understand their activity over the period of time, we take a look at attack frequency from each group.

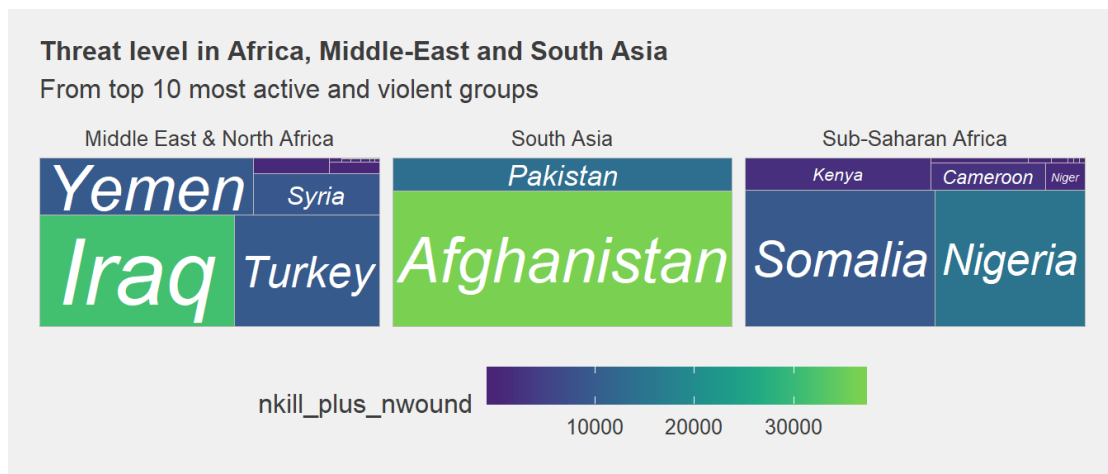


Figure 4.9 Threat Level in Africa, Middle-East & South Asia

From the plot and table above, we can see that all three regions are heavily impacted. While Afghanistan facing the largest impact in terms of fatalities and number of people injured followed by Iraq, we can also see that the spread in Southeast Asia is limited to Pakistan and Afghanistan only. In the case of Sub-Saharan Africa and the Middle East & North Africa region, we can also see many countries with a low number of attacks.

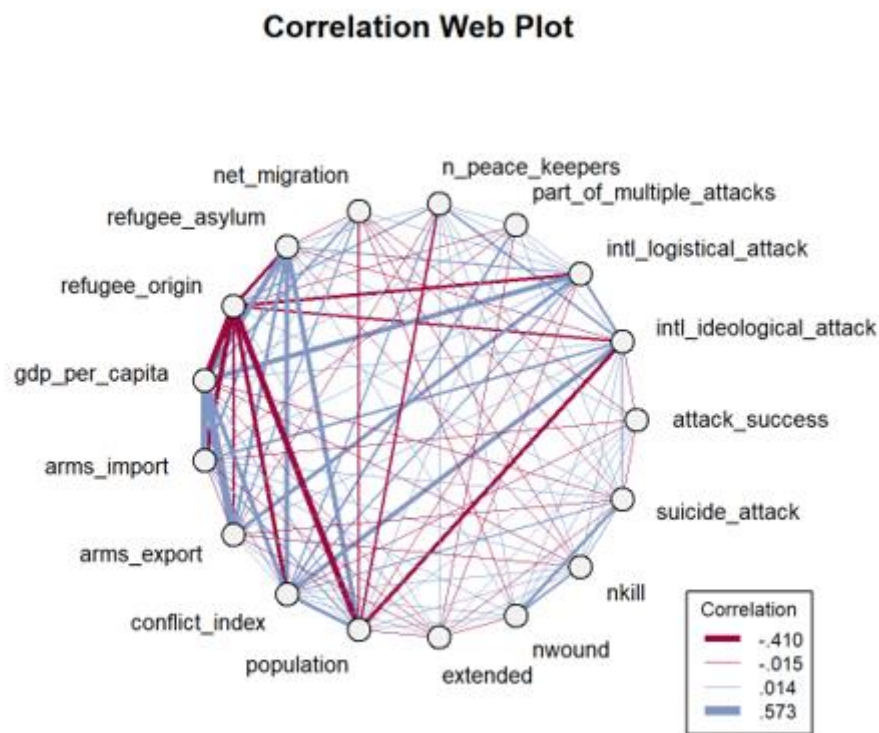
but the relatively large number of fatalities and injuries such as in Yemen, Niger, Nigeria, and Chad. In a comparison to other regions, the cumulative sum of a number of fatalities and injuries in Africa, Middle-East, and South Asia is more than 9,000 in each of the top five highly impacted countries.

# **Chapter 5**

## **Statistical Testing**

## 5.1 Correlation Test

We use pairwise complete observations method to compute correlation coefficients for each pair of numerical variables.



*Figure 5.1 Correlation Web Plot*

In the plot above, line width between the nodes is used in proportion to the correlation of two variables. To focus only on significant correlations, we have replaced observations with p-value more than 0.05 with NA. Legend on the bottom right represents correlation coefficient by line width and color depending on positive or negative linear relationship. The variables on the left-hand side of the plot are extracted from World Bank data (development indicators) and variables on the right-hand side are from GTD.

Specifically, we are more interested in the relationship to the variables on the right-hand side which will be used in time-series forecasting and classification modeling as the target variable. For example, a number of people wounded (wound) variable has a positive linear relationship with a suicide attack. The conflict index variable shows a

strong positive relationship with international ideological attacks and minor positive relationship with a part of multiple attacks. Overall, we can see that the majority of numerical variables shows a relationship with each other.

## 5.2 Hypothesis test

The objective behind this hypothesis test is to determine whether or not means of the top 10 groups with respect to average fatalities are same. If at least one sample mean is different to others, then we determine which pair of groups are different.

**H0: The means of the different groups are the same**

**(ISIL)=(Taliban)=(AQAP)=(PKK)=(Al-Shabaab)=(TTP)=(BokoHaram)=(Al-Nusrah)=(DonetskPR)=(HouthiExtrm)**

**Ha: At least one sample mean is not equal to the others**

we use a box plot to examine distribution by quartiles for each group.

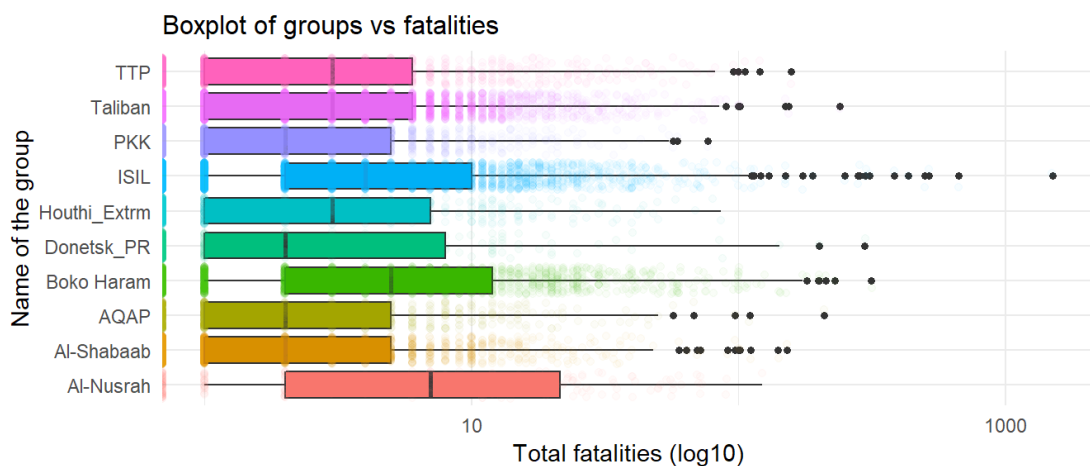


Figure 5.2 Boxplot of Groups vs Fatalities

In statistical terms, we have some extreme outliers for example  $n_{kill} \sim 1500$  in ISIL group so X axis is log transformed for visualization purpose.

### 5.2.1 ANOVA Test

The ANOVA model computes the residual variance and the variance between sample means in order to calculate the F-statistic. This is the first step to determine whether or not means are different in a pair of groups.

$$F\text{-statistic} = (S^2_{\text{between}} / S^2_{\text{within}})$$

```

          Df Sum Sq Mean Sq F value          Pr(>F)
group_name    9  111070    12341    40.7 <0.0000000000000002 ***
Residuals  21770 6597154      303
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

```

Figure 5.3 ANOVA Test Output

The model summary provides us F value and Pr(>F) corresponding to the p-value of the test. As we can see that the p-value is  $< 0.05$ , which means there are significant differences between the groups. In other words, we reject the null hypothesis. From this test, we identified that some of the group means are different however we don't s

## 5.2.2 PostHoc Test

PostHoc test is useful to determine where the differences occurred between groups. For this test, we use several different methods for the comparison purpose. This method can be classified as either conservative or liberal approach. Conservative methods are considered to be robust against committing Type I error as they use more stringent criterion for statistical significance. First, we run the PostHoc test by comparing results (p-value) from The Fisher LSD (Least Significant Different), Scheffe and Dunn's (Bonferroni) test.

Pair of groups	Lsd	Scheffe	bonf
Donetsk_PR-Al-Shabaab	0.9191	1.0000	1.0000
Houthi_Extrm-Al-Shabaab	0.7934	1.0000	1.0000
Houthi_Extrm-Donetsk_PR	0.7797	1.0000	1.0000
Taliban-AQAP	0.6811	1.0000	1.0000
PKK-Donetsk_PR	0.5800	1.0000	1.0000
Houthi_Extrm-AQAP	0.4850	1.0000	1.0000
Donetsk_PR-AQAP	0.3615	0.9997	1.0000
PKK-Houthi_Extrm	0.3152	0.9994	1.0000
PKK-Al-Shabaab	0.3021	0.9993	1.0000
AQAP-Al-Shabaab	0.2561	0.9984	1.0000
Taliban-Houthi_Extrm	0.1928	0.9954	1.0000
TTP-AQAP	0.1508	0.9904	1.0000
Taliban-Donetsk_PR	0.1476	0.9898	1.0000
TTP-Taliban	0.1253	0.9846	1.0000
Boko Haram-Al-Nusrah	0.0851	0.9656	1.0000
PKK-AQAP	0.0610	0.9406	1.0000
TTP-Houthi_Extrm	0.0324	0.8694	1.0000
TTP-Donetsk_PR	0.0278	0.8481	1.0000



Taliban-Al-Shabaab	0.0135	0.7301	0.6094
TTP-Al-Shabaab	0.0024	0.4187	0.1088
ISIL-Al-Nusrah	0.0008	0.2574	0.0354
Taliban-PKK	0.0005	0.2071	0.0226
ISIL-Boko Haram	0.0002	0.1338	0.0097

*Table 5.1: Posthoc test (lsd, scheffe, bonf)*

The Fisher LSD (Least Significant Different) test is the most liberal in all the PostHoc tests whereas the Scheffe test is the most conservative and protects against Type I error. On the other hand, Dunn's (Bonferroni) test is extremely conservative (Andri Signorell et mult. al., 2018). Out of all the possible combination of pairs (45), 16 pair of groups indicate p adj value  $> 0.9$  based on the Scheffe test. In statistical terms, it means 16 pairs of groups as shown in the table above have non-significantly different means in a number of fatalities.

Next, we use Tukey HSD (Honestly Significant Difference) method which is the most common and preferred method.

	Al-Nusrah	Al-Shabaab	AQAP	Boko Haram	Donetsk_PR	Houthi_Extrm	ISIL	PKK	Taliban
Al-Shabaab	0.000	NA	NA	NA	NA	NA	NA	NA	NA
AQAP	0.000	0.981	NA	NA	NA	NA	NA	NA	NA
Boko Haram	0.783	0.000	0.000	NA	NA	NA	NA	NA	NA
Donetsk_PR	0.000	1.000	0.996	0.000	NA	NA	NA	NA	NA
Houthi_Extrm	0.000	1.000	1.000	0.000	1.000	NA	NA	NA	NA
ISIL	0.027	0.000	0.000	0.008	0.000	0.000	NA	NA	NA
PKK	0.000	0.990	0.687	0.000	1.000	0.992	0	NA	NA
Taliban	0.000	0.285	1.000	0.000	0.912	0.953	0	0.018	NA
TTP	0.000	0.073	0.916	0.000	0.457	0.499	0	0.007	0.879

*Table 5.2: PostHoc test with Tukey HSD for pair of groups*

### 5.3 Results interpretation

The pairs of groups with adj p-value near or equals to 1 represents non-significantly different means in a number of fatalities such as Boko Haram - Al-Nusrah, Al-Qaida in Arabian Peninsula (AQAP)- Al-Shabaab, Houthi Extremist- PKK, Taliban- Tehrik-i-Taliban. Similarly, a pair of groups with adjusted p-value near zero indicates significantly different means in a number of fatalities such as pairs of ISIL with all the remaining groups, Taliban - Al-Nusrah, PKK - Boko Haram, Donetsk\_PR - Al-Nusrah

# **CHAPTER 06**

## **Pattern Discovery**

## CHAPTER 06

### Pattern Discovery

#### 6.1 Data preparation:

For this analysis, I have chosen specific variables that are not highly correlated with chosen groups i.e. target type, weapon type, attack type, suicide attack and a number of fatalities while excluding the observations where the value is “Unknown”.

#### 6.2 Explanation of key terms:

The Apriori algorithm has three main measures namely support, confidence and lift. These three measures are used to decide the relative strength of the rules. In the model parameters, we set RHS to the chosen group and LHS refers to a frequent pattern that is observed.

**Support** - Proportional frequency of an item in the database.  $\text{Support}(A) = \text{Frequency}(A) / \# \text{ Total records}$

**Confidence** - how confident we are of an event, given another event.  $\text{Confidence}(A \rightarrow B) = \text{Probability}(A \& B) / \text{Support}(A)$

**Lift** - a measure that tells us whether the probability of an event B increases or decreases given event A.  $\text{Lift}(A \rightarrow B) = \text{Confidence}(A \rightarrow B) / \text{Support}(B)$

In general, high confidence and good lift are the standard measures to evaluate the importance of a particular rule/ association however not all the rules are useful. This rules normally fall into three categories i.e. actionable, trivial and inexplicable.

Example of the useless rule can be an association that is obvious and thus not worth mentioning.

#### 6.3 Islamic State (ISIL):

##### 6.3.1 Apriori model summary:

In the model summary, we can see that the Absolute minimum support count is 18 which means the pattern needs to appear at least 18 times in order to be included. We have set this threshold with support value as explained previously. Out of all the patterns, the model is able to find 51 association rules for the ISIL group. We further remove the rules that may be redundant before starting our analysis.

confidence <dbl>	minval <dbl>	smax <dbl>	arem <chr>	aval <lg>	originalSupport <lg>	maxtime <dbl>	support <dbl>	minlen <int>	maxlen <int>	target <chr>	ext <lg>
0.5	0.1	1	none	FALSE	TRUE	5	0.001	2	10	rules	TRUE
row											

*Figure 6.1: Parameter specification*

## Algorithmic control:

filter <dbl>	tree <lgl>	heap <lgl>	memopt <lgl>	load <lgl>	sort <int>	verbose <lgl>
0.1	TRUE	TRUE	FALSE	TRUE	2	TRUE

Figure 6.2: Algorithmic control

## Absolute minimum support count: 18

```
set item appearances ... [1 item(s)] done [0.00s].
set transactions ... [52 item(s), 18006 transaction(s)] done [0.04s].
sorting and recoding items ... [48 item(s)] done [0.00s].
creating transaction tree ... done [0.02s].
checking subsets of size 1 2 3 4 5 6 done [0.03s].
writing ... [51 rule(s)] done [0.00s].
creating S4 object ... done [0.01s].
```

Figure 6.3: Absolute minimum support count

In the model summary, we can see that the Absolute minimum support count is 18 which means the pattern needs to appear at least 18 times in order to be included. We have set this threshold with support value as explained previously. Out of all the patterns, the model is able to find 51 association rules for the ISIL group. We further remove the rules that may be redundant before starting our analysis.

### 6.3.2 Top 5 patterns (ISIL)

lhs	rhs	support	confidence	coverage	lift	count
[1] {weapon_type=Chemical, attack_type=Bombing/Explosion}	=> {group_name=ISIL}	0.001055204	0.9047619	0.001166278	4.868841	19
[2] {target_type=Non-State Militia, attack_type=Bombing/Explosion, nkill=6 to 10}	=> {group_name=ISIL}	0.001055204	0.7307692	0.001443963	3.932526	19
[3] {target_type=Non-State Militia, attack_type=Bombing/Explosion, suicide_attack=1}	=> {group_name=ISIL}	0.003443297	0.6526316	0.005276019	3.512040	62
[4] {target_type=Military, suicide_attack=1, nkill=11 to 50}	=> {group_name=ISIL}	0.007997334	0.6457399	0.012384761	3.474953	144
[5] {target_type=Non-State Militia, suicide_attack=1}	=> {group_name=ISIL}	0.003498834	0.6237624	0.005609241	3.356684	63

Figure 6.4: Top 5 Patterns (ISIL)

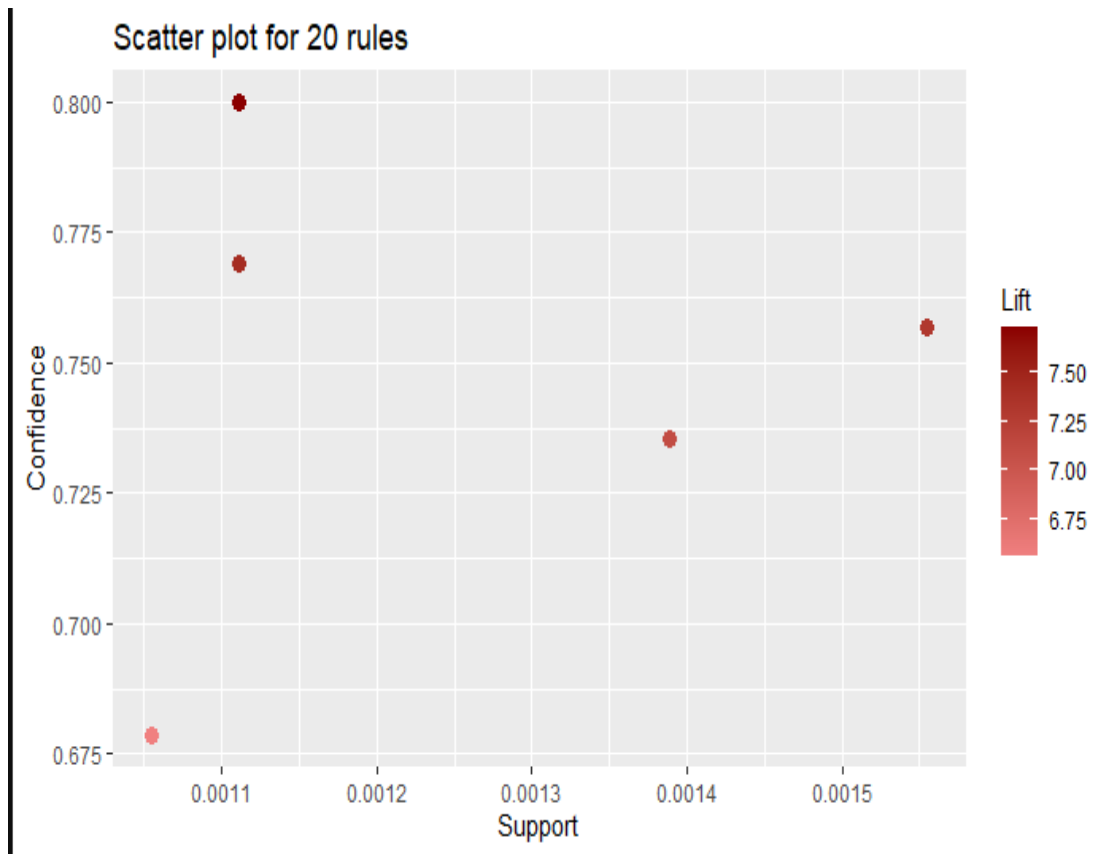


Figure 6.5: Association rules in ISIL group

The plot shown above represents all the discovered patterns (after removing redundant rules). We can see that majority of discovered rules are between 0.0011 to 0.0012 confidence while two rules with high support and both indicating an attack on the military with a suicide attack.

## 6.4 Taliban:

### 6.4.1 Apriori model summary:

#### Apriori

#### Parameter specification:

confidence	minval	smax	arem	aval	originalSupport	maxtime	support	minlen	maxlen	target	ext
<dbl>	<dbl>	<dbl>	<chr>	<dbl>	<dbl>	<dbl>	<dbl>	<int>	<int>	<chr>	<dbl>
0.5	0.1	1	none	FALSE	TRUE	5	0.001	2	10	rules	TRUE

Figure 6.6: Parameter specification

## Algorithmic control:

filter <dbl>	tree <lgl>	heap <lgl>	memopt <lgl>	load <lgl>	sort <int>	verbose <lgl>
0.1	TRUE	TRUE	FALSE	TRUE	2	TRUE

Figure 6.7: Algorithmic Control

```
Absolute minimum support count: 18
set item appearances ... [1 item(s)] done [0.00s].
set transactions ... [52 item(s), 18006 transaction(s)] done [0.03s].
sorting and recoding items ... [48 item(s)] done [0.00s].
creating transaction tree ... done [0.01s].
checking subsets of size 1 2 3 4 5 6 done [0.03s].
writing ... [139 rule(s)] done [0.00s].
creating S4 object ... done [0.01s].
```

Figure 6.8: Model Summary

From the model summary, we can see that the algorithm is able to identify 139 rules within the set threshold as defined in model parameters. However, it is possible that many rules may be redundant so we eliminate those rules.

### 6.4.2 Top 5 patterns (Taliban)

	lhs	rhs	support	confidence	coverage	lift	count
[1]	{weapon_type=Chemical, attack_type=Unarmed Assault}	=> {group_name=Taliban}	0.001221815	0.8800000	0.001388426	2.945219	22
[2]	{target_type=Police, weapon_type=Firearms, attack_type=Armed Assault, nkill=11 to 50}	=> {group_name=Taliban}	0.004998334	0.8256881	0.006053538	2.763446	90
[3]	{target_type=Police, weapon_type=Firearms, nkill=6 to 10}	=> {group_name=Taliban}	0.010163279	0.8243243	0.012329224	2.758882	183
[4]	{target_type=Police, weapon_type=Incendiary, attack_type=Facility/Infra., nkill=0}	=> {group_name=Taliban}	0.001999334	0.8000000	0.002499167	2.677472	36
[5]	{target_type=Police, weapon_type=Firearms, nkill=11 to 50}	=> {group_name=Taliban}	0.005664778	0.7968750	0.007108742	2.667013	102

Figure 6.9: Top 5 patterns (Taliban)

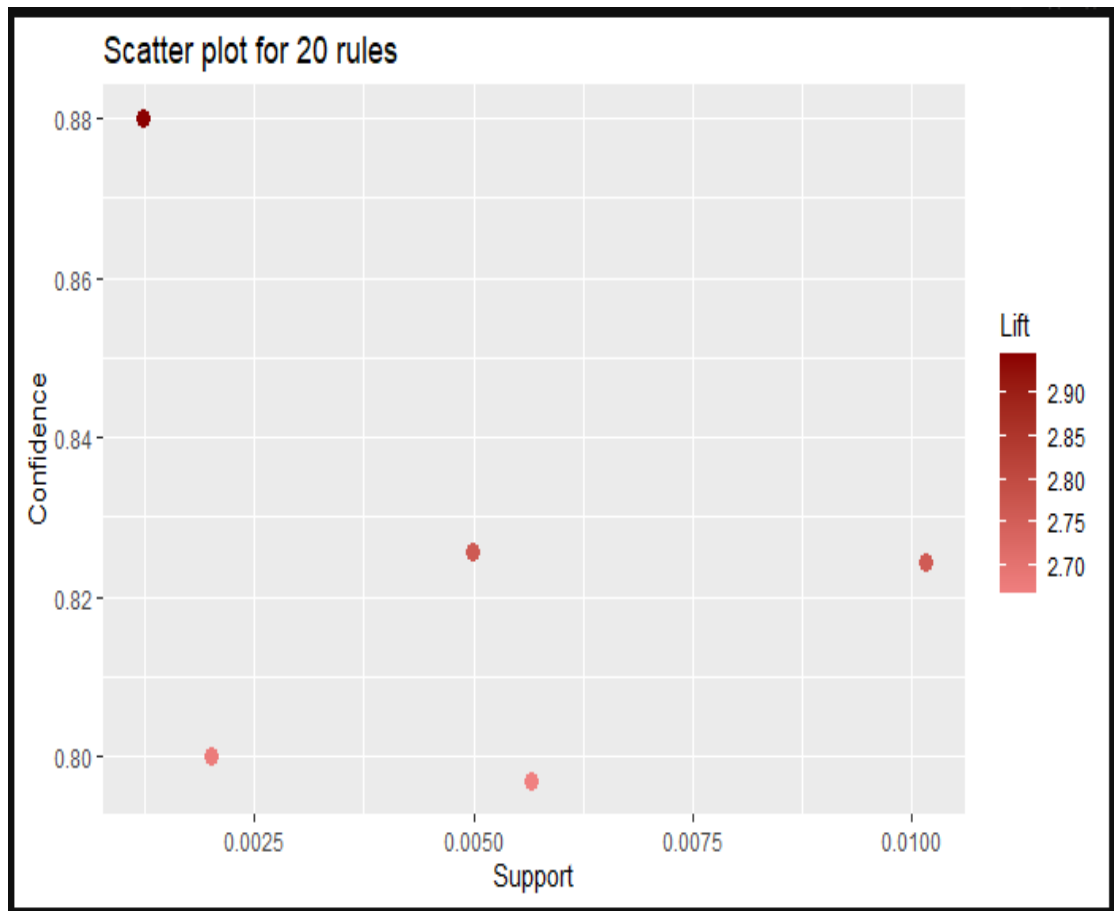


Figure 6.10: Association Rules in Taliban group

## 6.5 Boko Haram:

### 6.5.1 Apriori model summary:

#### Apriori

#### Parameter specification:

confidence	minval	smax	arem	aval	originalSupport	maxtime	support	minlen	maxlen	target	ext
<dbl>	<dbl>	<dbl>	<chr>	<lgl>	<lgl>	<dbl>	<dbl>	<int>	<int>	<chr>	<lgl>
0.5	0.1	1	none	FALSE	TRUE	5	0.001	2	10	rules	TRUE

Figure 6.11: Parameter specification

#### Algorithmic control:

filter	tree	heap	memopt	load	sort	verbose
<dbl>	<lgl>	<lgl>	<lgl>	<lgl>	<int>	<lgl>
0.1	TRUE	TRUE	FALSE	TRUE	2	TRUE

Figure 6.12: Algorithmic control:



## Absolute minimum support count: 18

```
set item appearances ... [1 item(s)] done [0.00s].
set transactions ... [52 item(s), 18006 transaction(s)] done [0.04s].
sorting and recoding items ... [48 item(s)] done [0.00s].
creating transaction tree ... done [0.01s].
checking subsets of size 1 2 3 4 5 6 done [0.02s].
writing ... [63 rule(s)] done [0.01s].
creating S4 object ... done [0.01s].
```

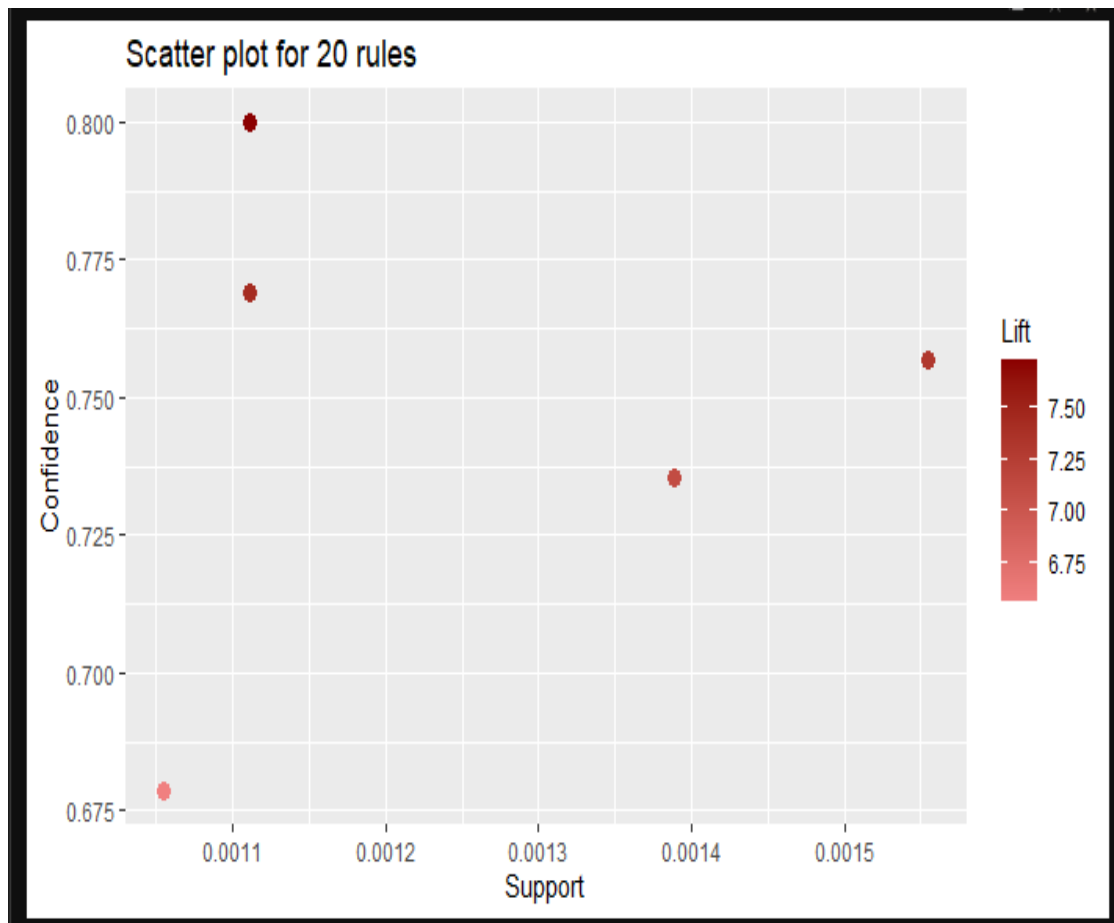
Figure 6.13: Absolute minimum support count

### 6.5.2 Top 5 patterns (Boko Haram)

	lhs	rhs	support	confidence	coverage	lift	count
[1]	{target_type=Civilians, weapon_type=Explosives, suicide_attack=0, nkill=more than 50}	=> {group_name=Boko Haram}	0.001110741	0.8000000	0.001388426	7.727897	20
[2]	{target_type=Civilians, weapon_type=Explosives, attack_type=Armed Assault, nkill=11 to 50}	=> {group_name=Boko Haram}	0.001110741	0.7692308	0.001443963	7.430670	20
[3]	{target_type=Civilians, attack_type=Armed Assault, nkill=more than 50}	=> {group_name=Boko Haram}	0.001555037	0.7567568	0.002054871	7.310173	28
[4]	{target_type=Civilians, weapon_type=Explosives, attack_type=Armed Assault, nkill=6 to 10}	=> {group_name=Boko Haram}	0.001388426	0.7352941	0.001888259	7.102847	25
[5]	{target_type=Civilians, weapon_type=Incendiary, attack_type=Armed Assault}	=> {group_name=Boko Haram}	0.001055204	0.6785714	0.001555037	6.554913	19

Figure 6.14: Top 5 patterns (Boko Haram)

In the case of Boko Haram, we can see quite different patterns in comparison to ISIL and Taliban group. All of the top five patterns, as shown above, indicates attacks on civilians. Specifically, incidents involving armed assault and use of explosives with resulting fatalities more than 50 are significant patterns. This also illustrates the differences in ideology between groups.



*Figure 6.15: Association Rules in Boko Haram group*

From the plot above, we can see patterns with high confidence and lift value with support between 0.0011 and 0.0012. Two patterns with high support value corresponds

to attack on civilians using firearms as a weapon type, armed assault as an attack type resulting fatalities between 6 to 10 and 11 to 50.

## **Chapter 7**

### **Predicting Class Probabilities**

## Chapter 7

### Predicting Class Probabilities

In our dataset, we have several categorical variables such as suicide attack, attack success, extended attack, part of multiple attacks etc. with qualitative value i.e. Yes/ No 1 or 0. In this chapter, we choose data from all the countries that are impacted by top 10 most active and violent groups and make use of a cutting-edge Light algorithm to predict the category of target variable which will be helpful to identify and understand the causal variables behind such attacks. This is a supervised machine learning approach, which means our dataset has labeled observations and the objective is to find a function that can be used to assign a class to unseen observations.

#### 7.1 LightGBM:

LightGBM is a fairly recent implementation of parallel GBDT process which uses histogrambased approach and offers significant improvement in training time and memory usage. The winning solutions from recent machine learning challenges on Kaggle and benchmarking of various GBM from the researcher indicate that LightGBM outperforms XGBoost and other traditional algorithms in terms of accuracy as well. LightGBM was developed by Microsoft researchers in October 2016 and it is an open-source library available in R and Python both.

##### 7.1.1 The mechanism behind the improvised accuracy:

The key difference between traditional algorithms and LightGBM algorithm is how trees are grown. Most decision tree learning algorithms controls the model complexity by depth and grow trees by level (depth-wise)

In contrast, the LightGBM algorithm uses a best-first approach and grows tree leaf-wise. As a result, the tree will choose the leaf with max delta loss to grow. According to (Microsoft Corporation, 2018), holding the leaf fixed, leaf-wise algorithms are able to achieve better accuracy i.e. lower loss compared to level-wise algorithms.

Researcher Shi further explains the phenomena behind tree growth in best-first and depth-first approach and suggests that most decision tree learners expand nodes in depth-first order whereas best-first tree learners expand the best node whose split achieves maximum reduction of impurity among all the nodes available for splitting. Although the resulting tree will be the same as a depth-wise tree, the difference is in the order in which it grown. One of the key advantages of using LightGBM algorithm is that it offers good accuracy with label encoded categorical features instead of one hot encoded feature. This eventually leads to faster training time. According to LightGBM documentation (Microsoft Corporation, 2018), the tree built on one-hot encoded features tends to be unbalanced and needs higher depth in order to achieve good accuracy in the case of categorical features with high cardinality. LightGBM implements Exclusive Feature Bundling (EFB) technique, which is based on research

by D. Fisher to find the optimal split over categories and often performs better than one-hot encoding. One disadvantage of the leaf-wise approach is that it may cause over fitting when data is small. To overcome this issue, LightGBM includes the `max_depth` parameter to control model complexity, however, trees still grow leaf-wise even when `max_depth` is specified (Microsoft Corporation, 2018).

## 7.2 Overview of the target variable:

For this analysis, I have selected suicide attack as a target variable. According to GTD codebook, this variable is coded “Yes” in those cases where there is evidence that the perpetrator did not intend to escape from the attack alive.

	level	freq	perc	cumfreq	cumperc
suicide_attack					
0	0	164580	96.61	164580	96.61
1	1	5770	3.39	170350	100.00

*Figure 7.1: Frequency Table*

## 7.3 Feature engineering:

Feature engineering is a process of creating representations of data that increase the effectiveness of a model (M. K. and K. Johnson, 2018). This is one of the most important aspects in machine learning that requires careful transformations and widening the feature space in order to improve the performance of the model. During the data cleaning process, we also have numeric variables with extreme values such as `arms_import`, `arms_export`, `nkill`, `nwound` etc. For the modeling purpose, we use log transformation for such features. Last but not the least, we add frequency count features to widen the feature space. Frequency count features is a known technique in machine learning competitions to improve the accuracy of the model. An example of the feature with frequency is a number of attacks by the group, year and region. Use of frequency count features adds more context to data and will be helpful to improve the performance of the model. At this point, all of our variables are numeric and there are no missing values or NAs in this prepared data.

	suicide_attack	year	month	day	region	country
58969	0	1995	4	20	South Asia	Afghanistan
59943	0	1995	8	3	South Asia	Afghanistan
60391	0	1995	9	21	South Asia	Afghanistan
60715	0	1995	10	25	South Asia	Afghanistan
71609	0	2001	1	7	South Asia	Afghanistan
...	...	...	...	...	...	...
170340	0	2016	12	31	South Asia	Afghanistan
170341	0	2016	12	31	South Asia	Afghanistan
170342	0	2016	12	31	South Asia	Afghanistan
170343	0	2016	12	30	South Asia	Afghanistan
170345	0	2016	12	31	Sub-Saharan Africa	Niger

	provstate	city	attack_type \
58969	Unknown	Unknown	Hostage Taking (Kidnapping)
59943	Kandahar	Kandahar	Hijacking
60391	Kandahar	Kandahar	Hijacking
60715	Kabul	Kabul	Bombing/Explosion
71609	Bamyan	Yakawlang	Armed Assault
...	...	...	...
170340	Jawzjan	Jakdalk	Unknown
170341	Jawzjan	Shirkhel	Unknown
170342	Jawzjan	Tofan	Unknown
170343	Jawzjan	Mangajek district	Bombing/Explosion
170345	Diffa	Garoua	Unknown

Figure 7.2: Log Transform 1

	target_type	...	nwound	arms_export	arms_import
58969	Police	...	0.009950	0.00995	0.009950
59943	Airports & Aircraft	...	0.009950	0.00995	0.009950
60391	Airports & Aircraft	...	0.009950	0.00995	0.009950
60715	Airports & Aircraft	...	0.009950	0.00995	0.009950
71609	Private Citizens & Property	...	0.009950	0.00995	0.009950
...	...	...	...	...	...
170340	Military	...	0.698135	0.00995	18.985995
170341	Military	...	1.101940	0.00995	18.985995
170342	Military	...	0.698135	0.00995	18.985995
170343	Terrorists/Non-State Militia	...	1.793425	0.00995	18.985995
170345	Military	...	2.080691	0.00995	0.009950

	population	gdp_per_capita	refugee_asylum	refugee_origin \
58969	16.654562	0.000000	19605.0	2679133.0
59943	16.654562	0.000000	19605.0	2679133.0
60391	16.654562	0.000000	19605.0	2679133.0
60715	16.654562	0.000000	19605.0	2679133.0
71609	16.858435	0.000000	6.0	3809767.0
...	...	...	...	...
170340	17.360982	617.889972	59770.0	2501410.0
170341	17.360982	617.889972	59770.0	2501410.0
170342	17.360982	617.889972	59770.0	2501410.0
170343	17.360982	617.889972	59770.0	2501410.0
170345	16.844338	391.132585	166084.0	1210.0

	net_migration	n_peace_keepers	conflict_index
58969	0.0	0.0	-1.0
59943	0.0	0.0	-1.0
60391	0.0	0.0	-1.0
60715	0.0	0.0	-1.0
71609	0.0	0.0	-1.0
...	...	...	...
170340	0.0	14.0	1.7
170341	0.0	14.0	1.7
170342	0.0	14.0	1.7
170343	0.0	14.0	1.7
170345	0.0	0.0	4.3

[19715 rows x 33 columns]

Figure 7.3: Log Transform 2

	suicide_attack	year	month	day	region	country
0	0	1995	4	20	South Asia	Afghanistan
1	0	1995	8	3	South Asia	Afghanistan
2	0	1995	9	21	South Asia	Afghanistan
3	0	1995	10	25	South Asia	Afghanistan
4	0	2001	1	7	South Asia	Afghanistan
...	...	...	...	...	...	...
19710	0	2016	12	31	South Asia	Afghanistan
19711	0	2016	12	31	South Asia	Afghanistan
19712	0	2016	12	31	South Asia	Afghanistan
19713	0	2016	12	30	South Asia	Afghanistan
19714	0	2016	12	31	Sub-Saharan Africa	Niger

	provstate	city	attack_type \
0	Unknown	Unknown	Hostage Taking (Kidnapping)
1	Kandahar	Kandahar	Hijacking
2	Kandahar	Kandahar	Hijacking
3	Kabul	Kabul	Bombing/Explosion
4	Bamyan	Yakawlang	Armed Assault
...	...	...	...
19710	Jawzjan	Jakdalk	Unknown
19711	Jawzjan	Shirkhel	Unknown
19712	Jawzjan	Tofan	Unknown
19713	Jawzjan	Mangajek district	Bombing/Explosion
19714	Diffa	Garoua	Unknown

	target_type	...	n_group_year	n_region_year \
0	Police	...	4	4
1	Airports & Aircraft	...	4	4
2	Airports & Aircraft	...	4	4
3	Airports & Aircraft	...	4	4
4	Private Citizens & Property	...	4	4
...	...	...	...	...
19710	Military	...	1064	1160
19711	Military	...	1064	1160
19712	Military	...	1064	1160
19713	Terrorists/Non-State Militia	...	1064	1160
19714	Military	...	237	814

Figure 7.4: Frequency Count Feature 1

	n_city_year	n_attack_year	n_target_year	n_weapon_year \
0	1	1	1	1
1	2	2	3	3
2	2	2	3	3
3	1	1	3	3
4	1	1	1	1
...	...	...	...	...
19710	1	622	1136	823
19711	2	622	1136	823
19712	1	622	1136	823
19713	2	1964	129	2146
19714	1	622	1136	823

	n_group_region_year	n_group	n_provstate	n_city
0	4	6575	118	1003
1	4	6575	571	185
2	4	6575	571	185
3	4	6575	297	259
4	4	6575	13	1
...	...	...	...	...
19710	1064	6575	172	1
19711	1064	6575	172	2
19712	1064	6575	172	1
19713	1064	6575	172	5
19714	237	2077	54	1

[19715 rows x 43 columns]

Figure 7.5: Frequency Count Feature 2

we have already taken care of missing values and NAs. With regard to LightGBM model, the primary requirement is to have all the variables in numeric. As discussed earlier, LightGBM offers good accuracy with label encoded categorical features compared to the one-hot encoding method used in most algorithms. In this regard, we label encode all the categorical variables and specify them as a vector in model parameters.

```

suicide_attack year month day region country provstate city
0 0 1995 4 20 4 0 215 4292
1 0 1995 8 3 4 0 111 2158
2 0 1995 9 21 4 0 111 2158
3 0 1995 10 25 4 0 108 2075
4 0 2001 1 7 4 0 34 4422
...
19710 0 2016 12 31 4 0 101 1953
19711 0 2016 12 31 4 0 101 3935
19712 0 2016 12 31 4 0 101 4232
19713 0 2016 12 30 4 0 101 2786
19714 0 2016 12 31 6 20 57 1506

attack_type target_type ... n_group_year n_region_year \
0 6 11 ... 4 4
1 4 0 ... 4 4
2 4 0 ... 4 4
3 2 0 ... 4 4
4 0 12 ... 4 4
...
19710 8 8 ... 1064 1160
19711 8 8 ... 1064 1160
19712 8 8 ... 1064 1160
19713 2 15 ... 1064 1160
19714 8 8 ... 237 814

n_city_year n_attack_year n_target_year n_weapon_year \
0 1 1 1 1
1 2 2 3 3
2 2 2 3 3
3 1 1 3 3
4 1 1 1 1
...
19710 1 622 1136 823
19711 2 622 1136 823
19712 1 622 1136 823
19713 2 1964 129 2146
19714 1 622 1136 823

n_group_region_year n_group n_provstate n_city
0 4 6575 118 1003
1 4 6575 571 185
2 4 6575 571 185
3 4 6575 297 259
4 4 6575 13 1
...
19710 1064 6575 172 1
19711 1064 6575 172 2
19712 1064 6575 172 1
19713 1064 6575 172 5
19714 237 2077 54 1

[19715 rows x 43 columns]
```

Figure 7.6: label encode categorical data (lightgbm requirement)



## 7.4 Validation strategy:

In general, cross-validation is the widely used approach to estimate performance of the model. In this approach, training data is split into equal sized (k) folds. The model is then trained on k-1 folds and performance is measured on the remaining fold (M. K. and K. Johnson, 2018). However, this approach is not suitable for our data. To further explain this, the observations in our dataset are time-based so training the model on recent years (for example 2000- 2010) and evaluating the performance on previous years (for example 1980- 1990) would not be meaningful. To overcome this issue, we use a time-based split to evaluate the performance of our model. In other words, we use the observations in the year 2016 as the test set and the remaining observations as our training set. This way we can be ensured that the model we have trained is capable of classifying target variable in current context. Following is the code used to implement validation strategy.

The next stage of the process is to convert our data into `lgb.Dataset` format. During this process, we create a vector containing names of all our categorical variables and specify it while constructing `lgb.Dataset`

Notice that we have assigned labels separately to training and test data. To summarize the process, we will train the model on training data (`dtrain`), evaluate performance on test data (`dtest`).

```
Training set shape: (15883, 43)
Testing set shape: (3912, 43)
```

*Figure 7.7: Validation Split*

## 7.5 Hyperparameter optimization:

Hyperparameter tuning is a process of finding the optimal value for the chosen model parameter. According to M. K. and K. Johnson parameter tuning is an important aspect of modeling because they control the model complexity. And so that, it also affects any variance-bias trade-off that can be made. There are several approaches for hyperparameter tuning such as Bayesian optimization, grid-search, and randomized search. For this analysis, we used random grid-search approach for hyperparameter optimization. In simple words, Randomized grid-search means we concentrate on the hyperparameter space that looks promising. This judgment often comes with the prior experience of working with similar data. Several researchers have also supported the randomized grid-search approach and have claimed that random search is much more efficient than any other approaches for optimizing the parameters. For this analysis, we choose number of leaves, max depth, bagging fraction, feature fraction and scale positive weight which are the most important parameters to control the complexity of the model. As shown in the code chunk below, first we define a grid by specifying parameter and iterate over a number of models in grids to find the optimal parameter values. From the hyperparameter tuning, we have extracted the optimized values based on AUC. Next, we use these parameters in the model building process.

```
[0.958668970599446, 0.958885195306815, 0.957234402829402, 0.9578246315979144, 0.9576077138626372, 0.9581528958171146, 0.958668970599446, 0.958885195306815]
1
(5, 4, 0.7, 0.7, 7)
(5, 4, 0.7, 0.7, 7)
```

Figure 7.8: best hyper parameter for debugging

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 19715 entries, 0 to 19714
Data columns (total 43 columns):
#   Column                                Non-Null Count  Dtype
---  -
0   suicide_attack                        19715 non-null  int64
1   year                                 19715 non-null  int64
2   month                               19715 non-null  int64
3   day                                  19715 non-null  int64
4   region                               19715 non-null  int32
5   country                             19715 non-null  int32
6   provstate                           19715 non-null  int32
7   city                                 19715 non-null  int32
8   attack_type                         19715 non-null  int32
9   target_type                         19715 non-null  int32
10  weapon_type                         19715 non-null  int32
11  target_nalty                        19715 non-null  int32
12  group_name                         19715 non-null  int32
13  crit1_pol_eco_rel_soc              19715 non-null  int64
14  crit2_publicize                    19715 non-null  int64
15  crit3_os_intl_hmn_low              19715 non-null  int64
16  part_of_multiple_attacks           19715 non-null  int64
17  individual_attack                  19715 non-null  int64
18  attack_success                     19715 non-null  int64
19  extended                           19715 non-null  int64
20  intl_logistical_attack             19715 non-null  int64
21  intl_ideological_attack             19715 non-null  int64
22  nkill                              19715 non-null  float64
23  nwound                             19715 non-null  float64
24  arms_export                        19715 non-null  float64
25  arms_import                        19715 non-null  float64
26  population                         19715 non-null  float64
27  gdp_per_capita                     19715 non-null  float64
28  refugee_asylum                   19715 non-null  float64
29  refugee_origin                     19715 non-null  float64
30  net_migration                      19715 non-null  float64
31  n_peace_keepers                    19715 non-null  float64
32  conflict_index                     19715 non-null  float64
33  n_group_year                       19715 non-null  int64
34  n_region_year                      19715 non-null  int64
35  n_city_year                        19715 non-null  int64
36  n_attack_year                      19715 non-null  int64
37  n_target_year                      19715 non-null  int64
38  n_weapon_year                      19715 non-null  int64
39  n_group_region_year                19715 non-null  int64
40  n_group                            19715 non-null  int64
41  n_provstate                        19715 non-null  int64
42  n_city                             19715 non-null  int64
dtypes: float64(11), int32(9), int64(23)
memory usage: 5.8 MB
```

Figure 7.9: Covert all the variable to numeric

```
['year', 'month', 'day', 'region', 'country', 'provstate', 'city', 'attack_type', 'target_type', 'weapon_type', 'target_nalty', 'group_name', 'crit1_pol_eco_rel_soc', 'crit2_publicize', 'crit3_os_intl_hmn_low', 'part_of_multiple_attacks', 'individual_attack', 'attack_success', 'extended', 'intl_logistical_attack', 'intl_ideological_attack', 'conflict_index']
```

Figure 7.10: define all categorical features

```
[(5, 4, 0.7, 0.7, 4), (5, 4, 0.7, 0.7, 7), (5, 4, 0.7, 0.8, 4), (5, 4, 0.7, 0.8, 7), (5, 4, 0.7, 0.9, 4), (5, 4, 0.7, 0.9, 7), (5, 4, 0.8, 0.7, 4), (5, 4, 0.8, 0.7, 7), (5, 4, 0.8, 0.8, 4), (5, 4, 0.8, 0.8, 7), (5, 4, 0.8, 0.9, 4), (5, 4, 0.8, 0.9, 7), (5, 4, 0.9, 0.7, 4), (5, 4, 0.9, 0.7, 7), (5, 4, 0.9, 0.8, 4), (5, 4, 0.9, 0.8, 7), (5, 4, 0.9, 0.9, 4), (5, 4, 0.9, 0.9, 7), (5, 6, 0.7, 0.7, 4), (5, 6, 0.7, 0.7, 7), (5, 6, 0.7, 0.8, 4), (5, 6, 0.7, 0.8, 7), (5, 6, 0.7, 0.9, 4), (5, 6, 0.7, 0.9, 7), (5, 6, 0.8, 0.7, 4), (5, 6, 0.8, 0.7, 7), (5, 6, 0.8, 0.8, 4), (5, 6, 0.8, 0.8, 7), (5, 6, 0.8, 0.9, 4), (5, 6, 0.8, 0.9, 7), (5, 6, 0.9, 0.7, 4), (5, 6, 0.9, 0.7, 7), (5, 6, 0.9, 0.8, 4), (5, 6, 0.9, 0.8, 7), (5, 6, 0.9, 0.9, 4), (5, 6, 0.9, 0.9, 7), (7, 4, 0.7, 0.7, 4), (7, 4, 0.7, 0.7, 7), (7, 4, 0.7, 0.8, 4), (7, 4, 0.7, 0.8, 7), (7, 4, 0.7, 0.9, 4), (7, 4, 0.7, 0.9, 7), (7, 4, 0.8, 0.7, 4), (7, 4, 0.8, 0.7, 7), (7, 4, 0.8, 0.8, 4), (7, 4, 0.8, 0.8, 7), (7, 4, 0.8, 0.9, 4), (7, 4, 0.8, 0.9, 7), (7, 4, 0.9, 0.7, 4), (7, 4, 0.9, 0.7, 7), (7, 4, 0.9, 0.8, 4), (7, 4, 0.9, 0.8, 7), (7, 4, 0.9, 0.9, 4), (7, 4, 0.9, 0.9, 7), (7, 6, 0.7, 0.7, 4), (7, 6, 0.7, 0.7, 7), (7, 6, 0.7, 0.8, 4), (7, 6, 0.7, 0.8, 7), (7, 6, 0.7, 0.9, 4), (7, 6, 0.7, 0.9, 7), (7, 6, 0.8, 0.7, 4), (7, 6, 0.8, 0.7, 7), (7, 6, 0.8, 0.8, 4), (7, 6, 0.8, 0.8, 7), (7, 6, 0.8, 0.9, 4), (7, 6, 0.8, 0.9, 7), (7, 6, 0.9, 0.7, 4), (7, 6, 0.9, 0.7, 7), (7, 6, 0.9, 0.8, 4), (7, 6, 0.9, 0.8, 7), (7, 6, 0.9, 0.9, 4), (7, 6, 0.9, 0.9, 7), (9, 4, 0.7, 0.7, 4), (9, 4, 0.7, 0.7, 7), (9, 4, 0.7, 0.8, 4), (9, 4, 0.7, 0.8, 7), (9, 4, 0.7, 0.9, 4), (9, 4, 0.7, 0.9, 7), (9, 4, 0.8, 0.7, 4), (9, 4, 0.8, 0.7, 7), (9, 4, 0.8, 0.8, 4), (9, 4, 0.8, 0.8, 7), (9, 4, 0.8, 0.9, 4), (9, 4, 0.8, 0.9, 7), (9, 4, 0.9, 0.7, 4), (9, 4, 0.9, 0.7, 7), (9, 4, 0.9, 0.8, 4), (9, 4, 0.9, 0.8, 7), (9, 4, 0.9, 0.9, 4), (9, 4, 0.9, 0.9, 7), (9, 6, 0.7, 0.7, 4), (9, 6, 0.7, 0.7, 7), (9, 6, 0.7, 0.8, 4), (9, 6, 0.7, 0.8, 7), (9, 6, 0.7, 0.9, 4), (9, 6, 0.7, 0.9, 7), (9, 6, 0.8, 0.7, 4), (9, 6, 0.8, 0.7, 7), (9, 6, 0.8, 0.8, 4), (9, 6, 0.8, 0.8, 7), (9, 6, 0.8, 0.9, 4), (9, 6, 0.8, 0.9, 7), (9, 6, 0.9, 0.7, 4), (9, 6, 0.9, 0.7, 7), (9, 6, 0.9, 0.8, 4), (9, 6, 0.9, 0.8, 7), (9, 6, 0.9, 0.9, 4), (9, 6, 0.9, 0.9, 7)]
```

Figure 7.11: define the gird for hyper parameter

## 7.6 Modelling:

### 7.6.1 Model evaluation:

In order to evaluate the performance of our model on test data, we have used AUC metric which is commonly used in binary classification problem. From the trained model, we extract AUC score on test data from the best iteration with the code as shown below:

```
Best iteration: 0
AUC score on test data: 0.958885195306815
```

*Figure 7.12: auc score on test data*

```
Current AUC: 0.9581478136124542 | Hyperparameters: (5, 4, 0.7, 0.9, 7)
[LightGBM] [Info] Number of positive: 1898, number of negative: 13905
[LightGBM] [Info] Auto-choosing col-wise multi-threading, the overhead of testing was 0.023595 seconds.
You can set `force_col_wise=true` to remove the overhead.
[LightGBM] [Info] Total Bins 2555
[LightGBM] [Info] Number of data points in the train set: 15803, number of used features: 40
[LightGBM] [Info] [binary:BoostFromScore]: pavg=0.120104 -> initscore=-1.991448
[LightGBM] [Info] Start training from score -1.991448
Current AUC: 0.9586671225250241 | Hyperparameters: (5, 4, 0.8, 0.7, 4)
[LightGBM] [Info] Number of positive: 1898, number of negative: 13905
[LightGBM] [Info] Auto-choosing col-wise multi-threading, the overhead of testing was 0.012712 seconds.
You can set `force_col_wise=true` to remove the overhead.
[LightGBM] [Info] Total Bins 2555
[LightGBM] [Info] Number of data points in the train set: 15803, number of used features: 40
[LightGBM] [Info] [binary:BoostFromScore]: pavg=0.120104 -> initscore=-1.991448
[LightGBM] [Info] Start training from score -1.991448
Current AUC: 0.9588648664881735 | Hyperparameters: (5, 4, 0.8, 0.7, 7)
[LightGBM] [Info] Number of positive: 1898, number of negative: 13905
[LightGBM] [Info] Auto-choosing col-wise multi-threading, the overhead of testing was 0.010916 seconds.
You can set `force_col_wise=true` to remove the overhead.
[LightGBM] [Info] Total Bins 2555
[LightGBM] [Info] Number of data points in the train set: 15803, number of used features: 40
[LightGBM] [Info] [binary:BoostFromScore]: pavg=0.120104 -> initscore=-1.991448
[LightGBM] [Info] Start training from score -1.991448
Current AUC: 0.9572205422712373 | Hyperparameters: (5, 4, 0.8, 0.8, 4)
[LightGBM] [Info] Number of positive: 1898, number of negative: 13905
[LightGBM] [Info] Auto-choosing col-wise multi-threading, the overhead of testing was 0.023729 seconds.
You can set `force_col_wise=true` to remove the overhead.
[LightGBM] [Info] Total Bins 2555
[LightGBM] [Info] Number of data points in the train set: 15803, number of used features: 40
[LightGBM] [Info] [binary:BoostFromScore]: pavg=0.120104 -> initscore=-1.991448
[LightGBM] [Info] Start training from score -1.991448
Current AUC: 0.9578181633374376 | Hyperparameters: (5, 4, 0.8, 0.8, 7)
[LightGBM] [Info] Number of positive: 1898, number of negative: 13905
[LightGBM] [Info] Auto-choosing col-wise multi-threading, the overhead of testing was 0.007629 seconds.
You can set `force_col_wise=true` to remove the overhead.
[LightGBM] [Info] Total Bins 2555
[LightGBM] [Info] Number of data points in the train set: 15803, number of used features: 40
[LightGBM] [Info] [binary:BoostFromScore]: pavg=0.120104 -> initscore=-1.991448
[LightGBM] [Info] Start training from score -1.991448
Current AUC: 0.9576026316579769 | Hyperparameters: (5, 4, 0.8, 0.9, 4)
[LightGBM] [Info] Number of positive: 1898, number of negative: 13905
[LightGBM] [Info] Auto-choosing col-wise multi-threading, the overhead of testing was 0.008739 seconds.
You can set `force_col_wise=true` to remove the overhead.
[LightGBM] [Info] Total Bins 2555
[LightGBM] [Info] Number of data points in the train set: 15803, number of used features: 40
[LightGBM] [Info] [binary:BoostFromScore]: pavg=0.120104 -> initscore=-1.991448
[LightGBM] [Info] Start training from score -1.991448
Current AUC: 0.9581478136124542 | Hyperparameters: (5, 4, 0.8, 0.9, 7)
No improvement for 10 rounds. Early stopping.
```

*Figure 7.13: model training*

To deal with overfitting, we have specified early stopping criteria which stops the model training if no improvement is observed within specified rounds. At the best iteration, our model achieves 96.36% accuracy on validation data. To further investigate the error rate, we use the confusion matrix.

### 7.6.2 Confusion Matrix:

A confusion matrix is another way to evaluate performance of binary classification model.

The accuracy of 0.83 indicates that our model is 83% accurate. Out of all the metrics, the one we are most interested in is specificity. We want our classifier to predict the “Yes”/“1” instances of suicide attack with higher accuracy. From the contingency table, we can see that our model has correctly predicted 667 out of instances of “1”/ “Yes” in suicide attacks and achieves an accuracy of 82.95%.

Classification Report:				
	precision	recall	f1-score	support
0	0.83	1.00	0.91	3245
1	0.00	0.00	0.00	667
accuracy			0.83	3912
macro avg	0.41	0.50	0.45	3912
weighted avg	0.69	0.83	0.75	3912

Figure 7.14: Confusion Matrix 1

```
Accuracy: 82.95%
Sensitivity: 0.0000
Specificity: 1.0000
```

Figure 7.15: Accuracy

### 7.6.3 Feature importance:

Gain is the most important measure for predictions and represents feature contribution to the model. This is calculated by comparing the contribution of each feature for each tree in the model. The Cover metric indicates a number of observations related to the particular feature. The Frequency measure is the percentage representing the relative number of times a particular feature occurs in the trees of the model. In simple words, it tells us how often the feature is used in the model. From the feature importance matrix, we can see that type of weapon contributes the most in terms of gain followed by number of people killed, province state, type of attack and type of target. In order to allow the model to decide whether an attack will be a suicide attack or not, these features are the most important compared to others.

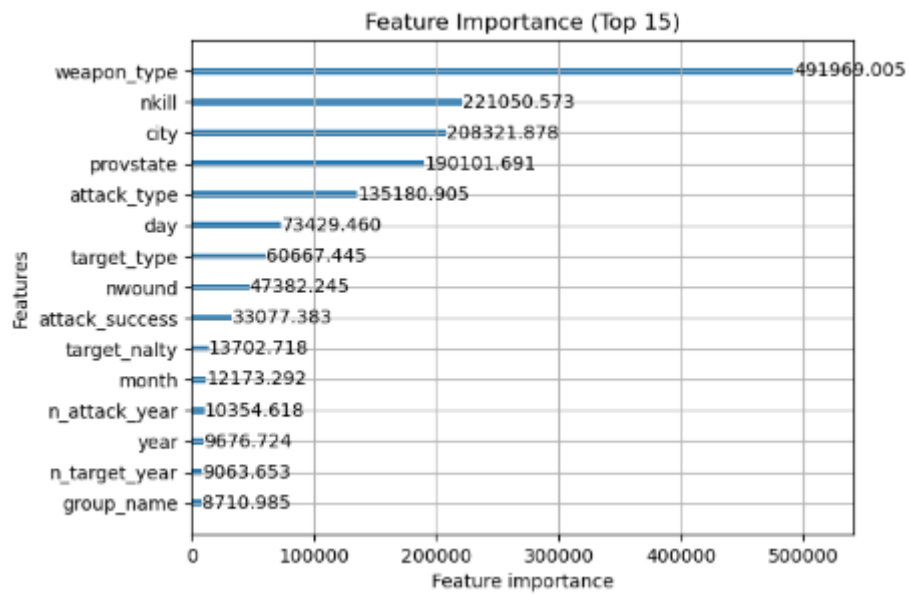


Figure 7.16: Feature Importance

## 7. Model interpretation:

We have checked highest AUC

---

Model 2 has the highest AUC: 0.958885195306815

Figure 7.17: Highest AUC

# **Chapter 8**

## **Time-series Analysis**

Time-series is a supervised machine learning approach that uses historical data to predict future occurrences. This is particularly helpful in terrorism context for long-term strategic planning. For this analysis, first, we select the appropriate data, examine seasonal components and then split the data in training and test set to evaluate the performance of Auto Arima, Neural Network, TBATS and ETS models with seven different metrics.

## 8.1 Afghanistan

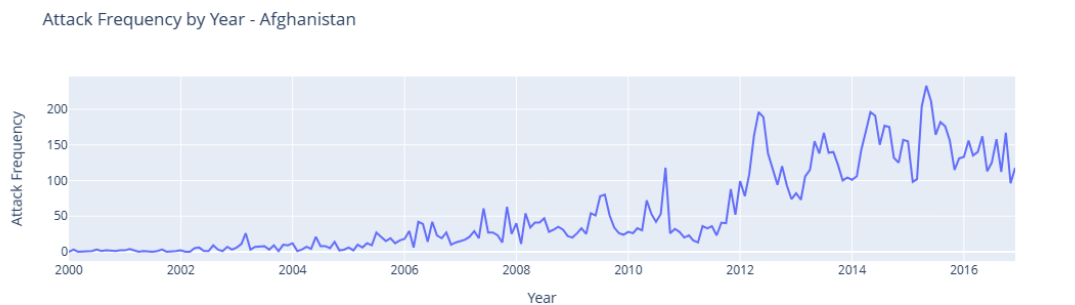
We have only taken Afghanistan for our time series analysis.

### 8.1.1 Data preparation

Based on exploratory data analysis, it is observed that the number of attacks with visible pattern began from the year 2000 so the data is selected between the year 2000 to 2016. To get the time-series frequency by months for all the years we add missing months and assign zero

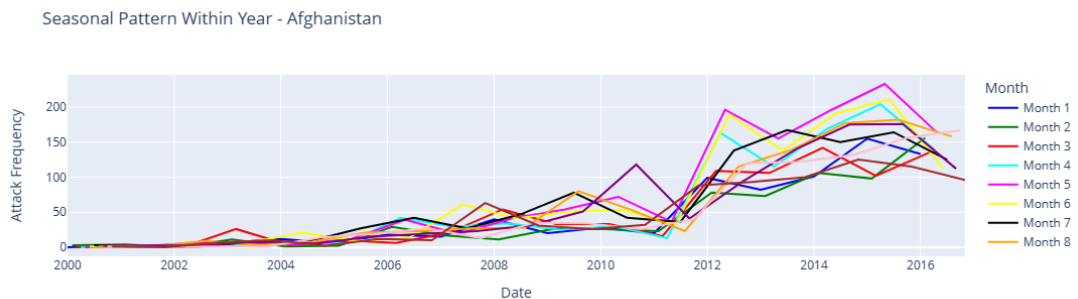
### 8.1.2 Seasonality analysis

First, we take a look at time plot to get an idea about how a number of attacks have changed over the period of time. In the plot below, observations (number of attacks) are plotted against the time of observation.



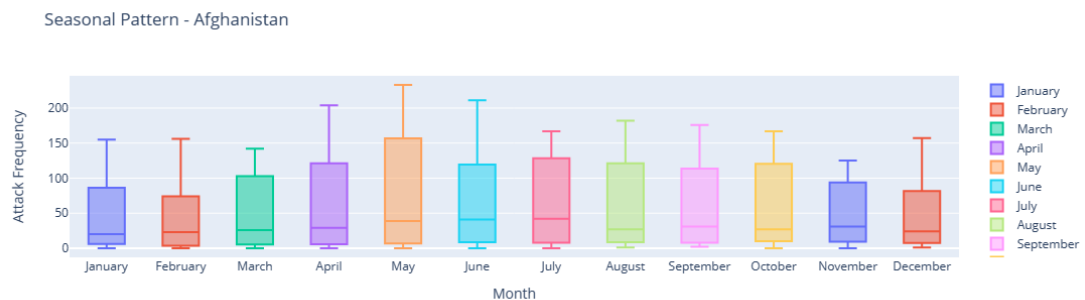
*Figure 8.1: Attack Frequency by Year - Afghanistan*

The seasonal plot is similar to time plot above with seasonality component (i.e. months) in which the number of attacks were observed



*Figure 8.2: Seasonal Pattern within Year - Afghanistan*

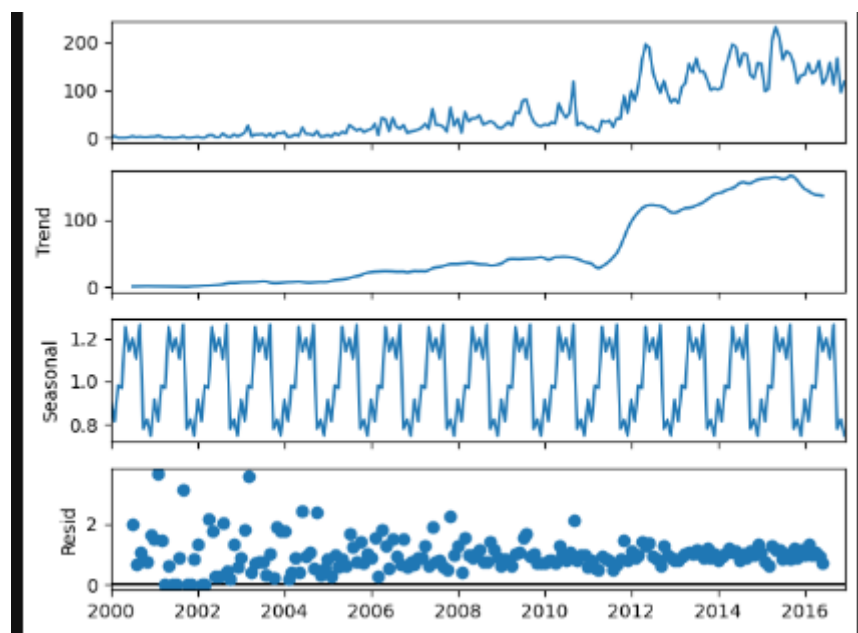
From the seasonal patterns within a year, as shown in the plot above, we can see that year 2015 (followed by 2012) was the deadliest year in terms of number of terror attacks. In both years, the spike is visible in May month.



*Figure 8.3: Seasonal Pattern - Afghanistan*

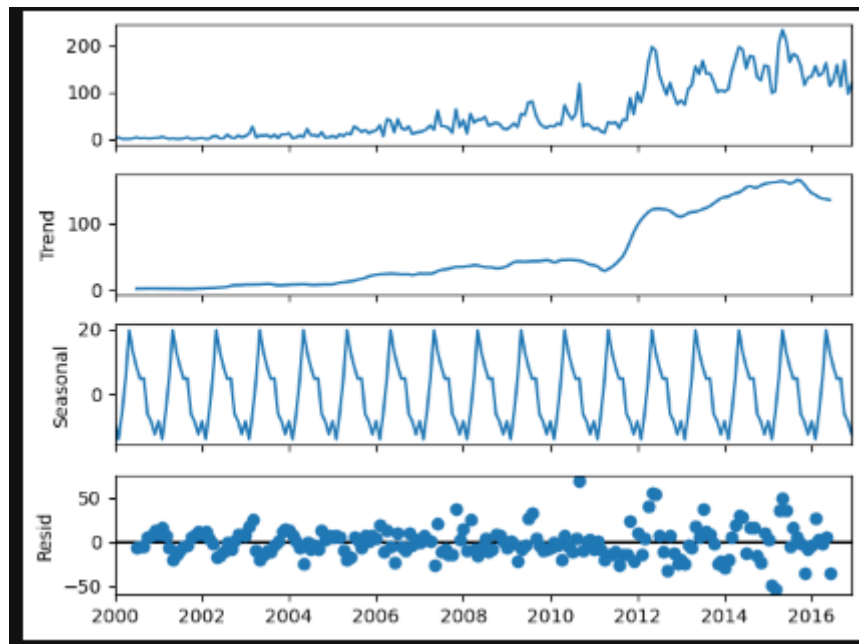
From the boxplot, we can confirm that the May month contributes the most in terms of terrorist incidents throughout all the years (2000-2016) in Afghanistan. We can see the upward trend in a number of attacks starting from February and reaching a peak in May month.

Decomposition by additive and multiplicative time-series is helpful to describe the trend and seasonal component within data. This also helps understand anomalies in data as shown in the plot below.



*Figure 8.4: Time-series decomposition Additive- Afghanistan 1*





*Figure 8.5: Time-series decomposition Multiplicative- Afghanistan 2*

Time-series decomposition comprises three components depending on observed patterns:

- a seasonal component,
- a trend-cycle component
- a residual (noise) component

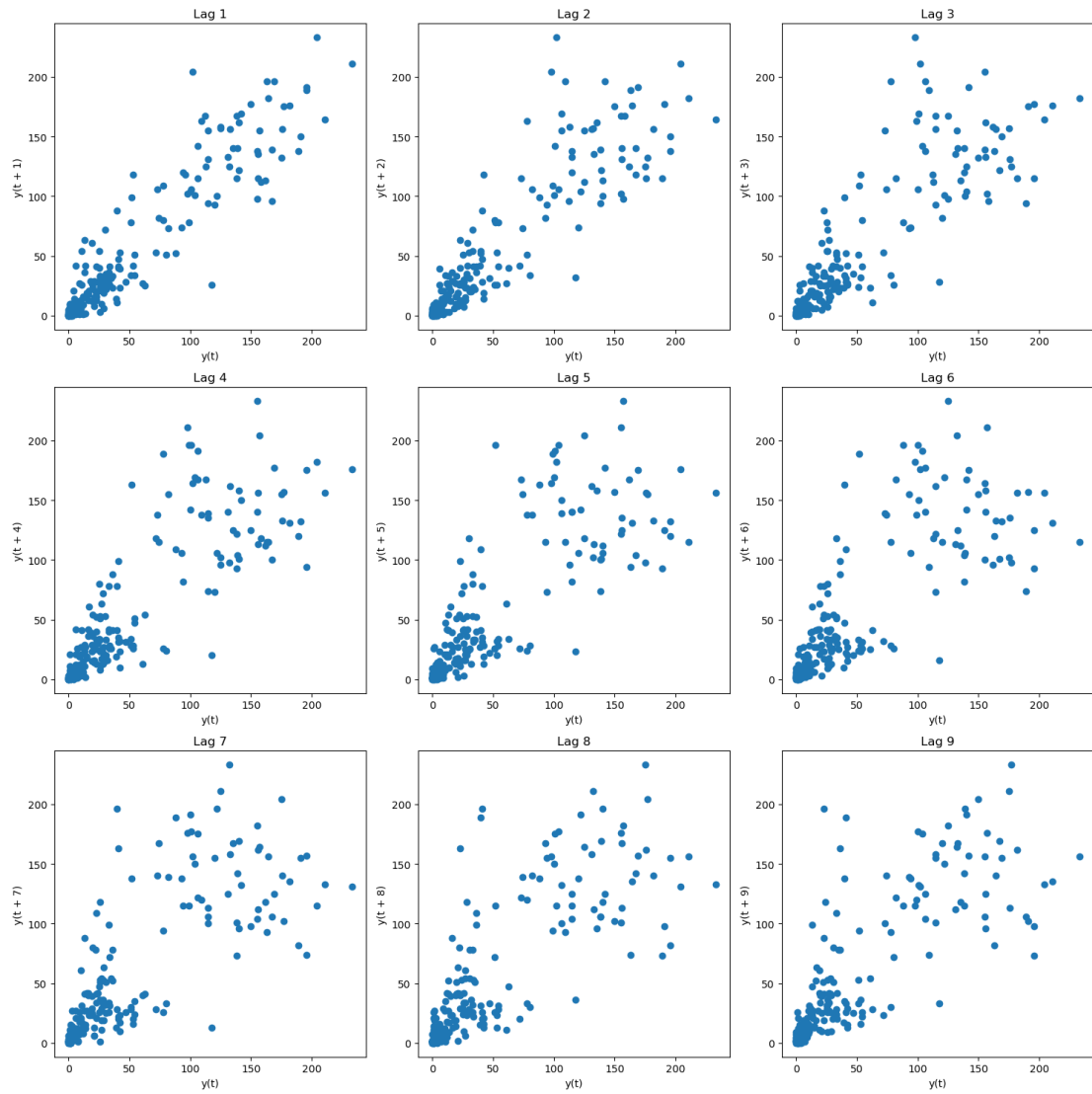
The seasonal component as shown in the plot above represents a pattern that occurs frequently within a fixed period of time. Trend-cycle contains both trend and cycle and a remainder component contains everything else in the time-series. The remainder component is also called random component/ noise and it represents residuals of the original time-series after removing seasonal and trend component (Anomaly.io, 2015; Hyndman & Athanasopoulos, 2018)

#### 6.1.3 Correlation test

There are several methods to identify a correlation between series and lags such as ACF, PACF and lag plots. In a lag plot, two variables are lagged and presented in scatterplot manner. In simple words, lag means a fixed amount of time from time-series data. We use lag plots method for this analysis which allows us to quickly visualize three things:

- outliers
- randomness and
- auto-correlation

The plot as shown below represents nine different lags. Although we can see a few outliers but there is no randomness in data. To further explain this, we can see the positive linear trend going upward from left to right in all nine plots. The positive linear trend is an indication that positive auto-correlation is present in our data



*Figure 8.6: Correlation test*

Specifically, lags 1, 2, 3 and 9 show strong positive auto-correlation. Presence of autocorrelation can be problematic for some models.

Figure 8.7: Autocorrelation Function (ACF) Plot

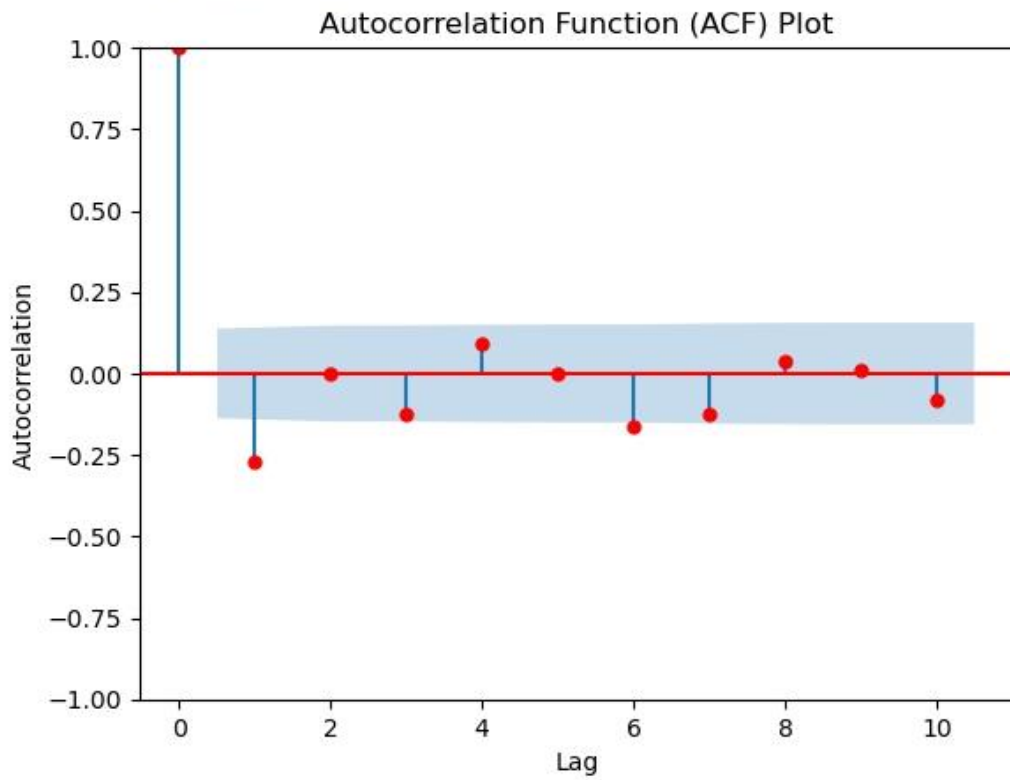


Figure 8.7: ACF Plot

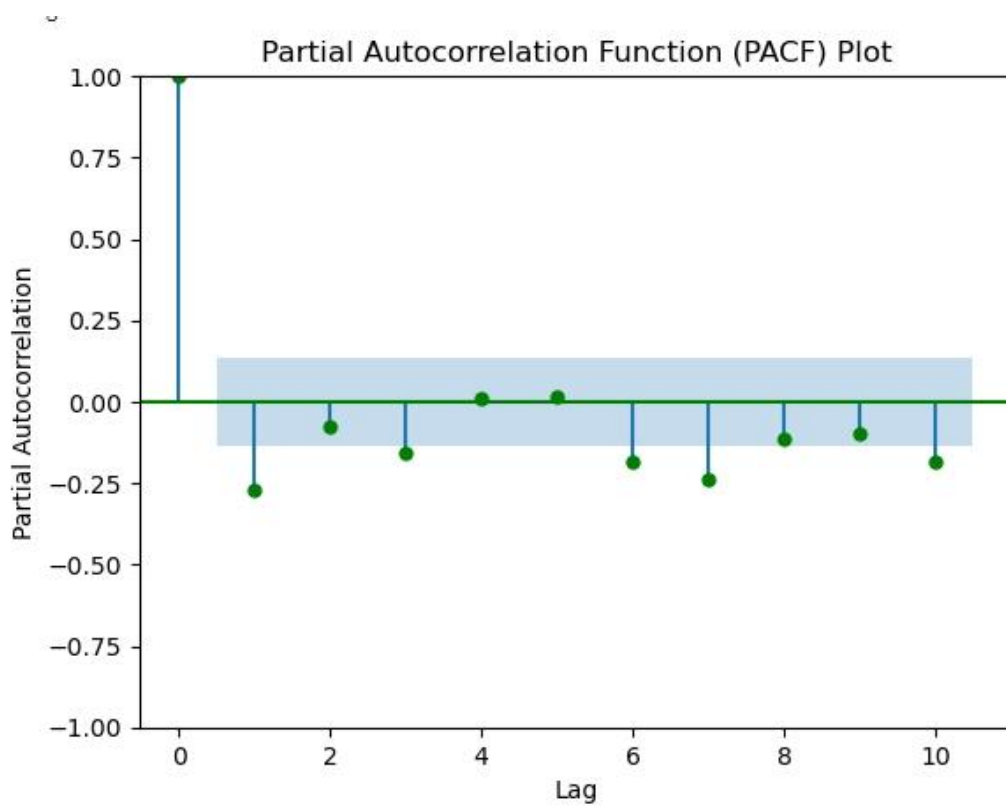


Figure 8.8: PACF Plot

#### 8.1.4 Modelling

Before moving further in our analysis we first need to check if our time series data is stationary or not for this we have taken the dicky fuller test. according to our result the series is not stationary

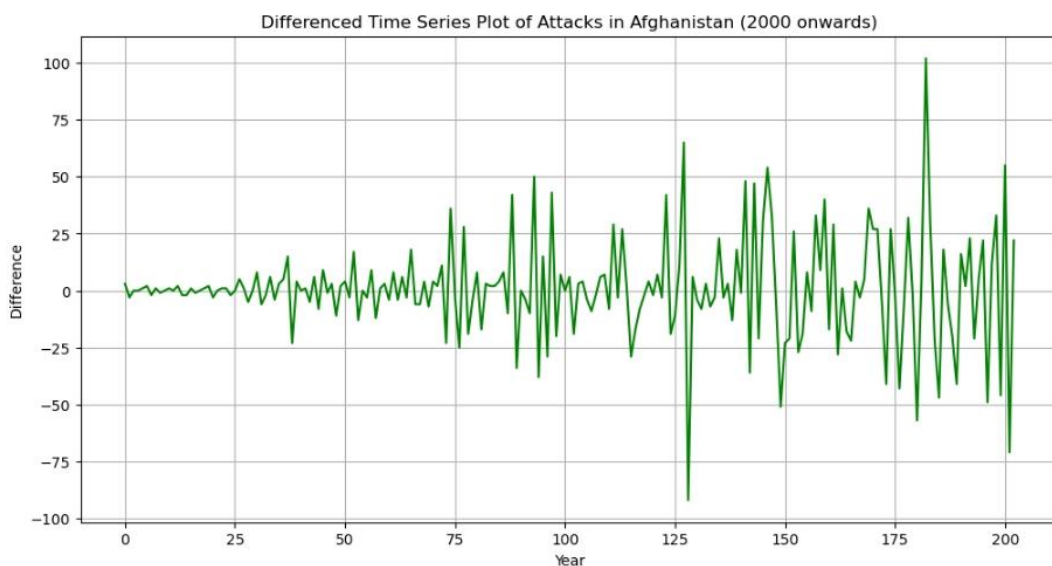
```
ADF Statistics : -0.582645
P-value : 0.874856
Critical Values:
  1%: -3.466
  5%: -2.877
 10%: -2.575
```

results shows dicky fuller and p-value greater than 0.05 means non stationary series

*Figure 8.9: dicky fuller test*

Moving ahead we have divided our series into two types a stationary and non-stationary series

After doing the series stationary this is how our series now look like in the plot given below



*Figure 8.10: Differenced Time Series Plot of Attack in Afghanistan*

In this part of the analysis, we split the data in training and test set in order to evaluate the performance of four different models

We have chosen 12 months look ahead period (horizon) so the test set contains the last 12 months from our data i.e. all the months in the year 2016 on which we will be evaluating the performance of the model.

A quick look at residuals from Auto Arima suggests that the mean of residuals is very close to zero however from the histogram, we can see that residuals don't follow the normal distribution. What this means is, forecasts from this method will probably be quite good but

prediction intervals computed assuming a normal distribution may be inaccurate (Hyndman & Athanasopoulos, 2018).

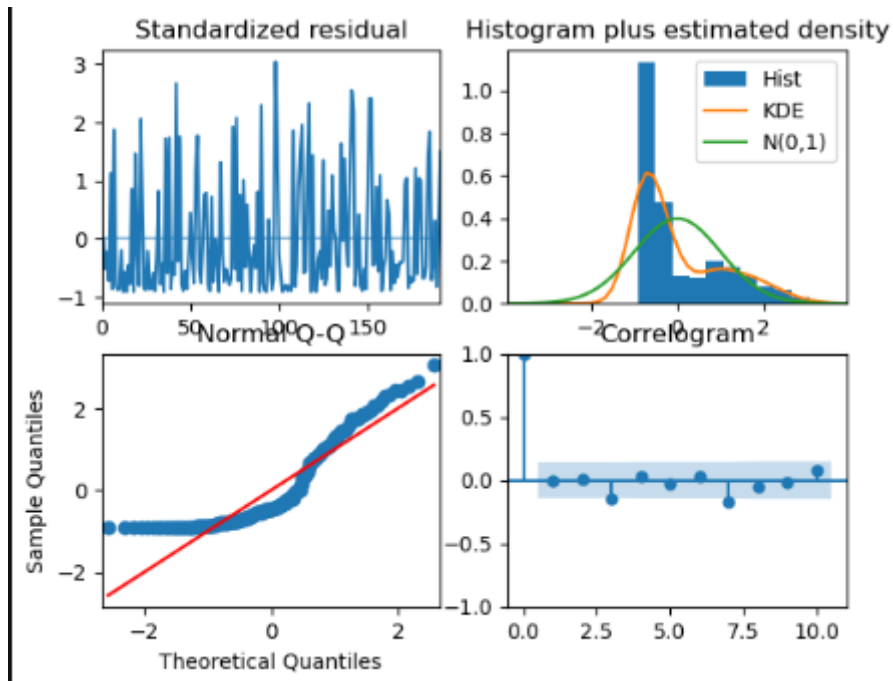


Figure 8.11: Theoretical Quantiles

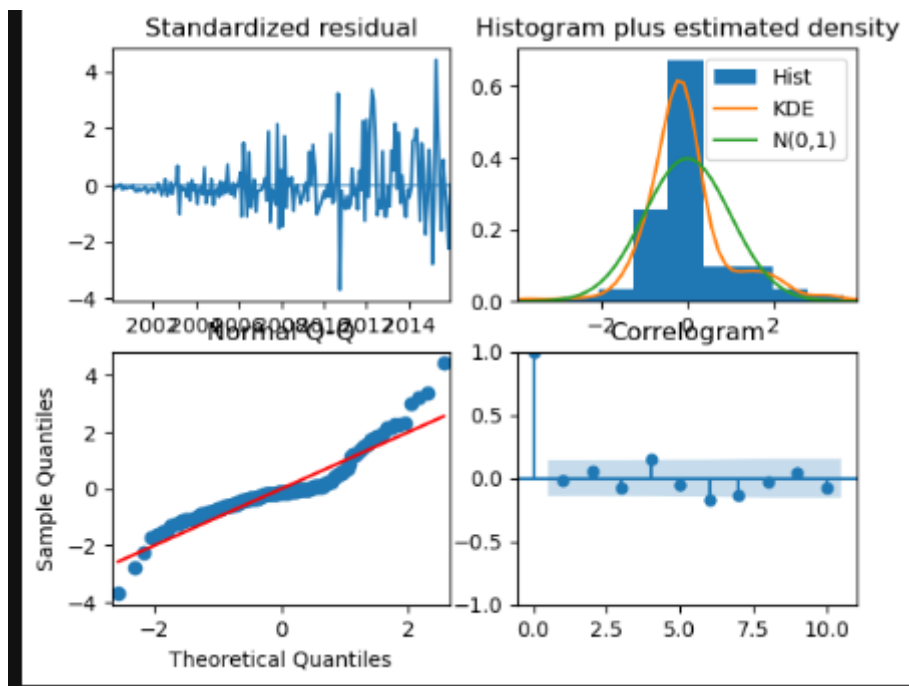
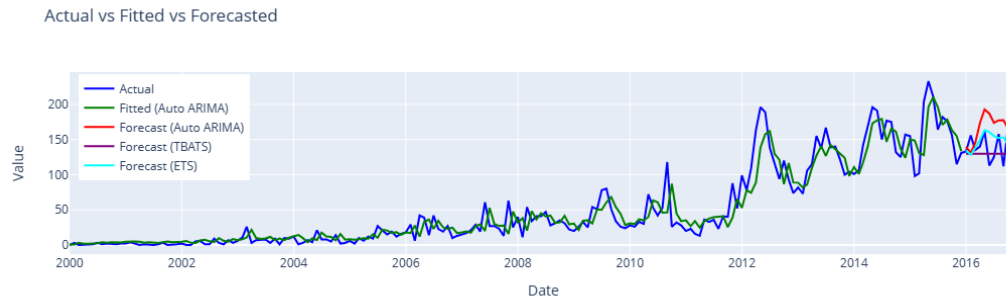


Figure 8.12: Theoretical Quantiles 2

From the plot below, it is observed that the Auto Arima model nearly captures fitted values based on training data but forecasted values are a little bit apart from actual values (test

data- year 2016). Next, we examine the pattern in actual vs fitted and forecasted values for the remaining three models.



*Figure 8.13: Actual vs Fitted vs Forecasted*

#### 8.1.5 Evaluating models' Performance

To compare the performance of all four models on test data, I have extracted mean accuracy from each model and have arranged the models by MAPE metric which is most commonly used. We will also look at six other metrics to get a better idea of the model's performance. Out of all the seven metrics, as shown in the table below, ME (Mean Error), RMSE (Root Mean Squared Error) and MAE (Mean Absolute Error) are a scale-dependent error. Whereas MPE (Mean Percentage Error) and MAPE (Mean Absolute Percent Error) are percentage errors. As suggested by researcher that percentage errors have the advantage of being unit-free, and so are frequently used to compare forecast performances between data sets.

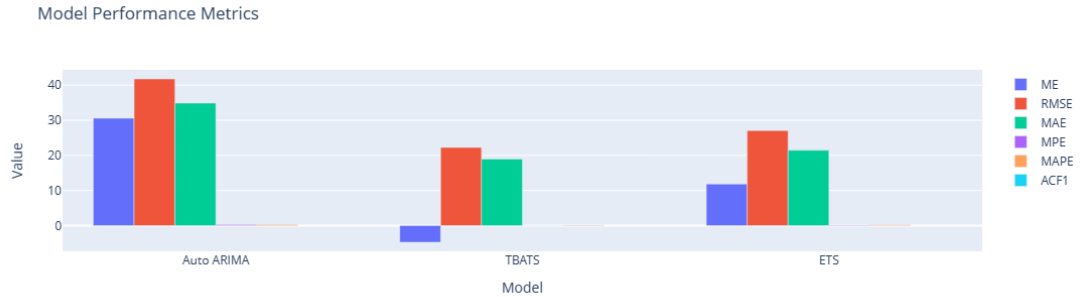
	Model	ME	RMSE	MAE	MPE	MAPE
0	Auto ARIMA	30.576442	41.771698	34.887270	0.261760	0.289322
1	TBATS	-4.740686	22.253475	18.942892	-0.008449	0.141781
2	ETS	11.836650	27.074616	21.439198	0.118555	0.178335

*Figure 8.14: Evaluating models' Performance*

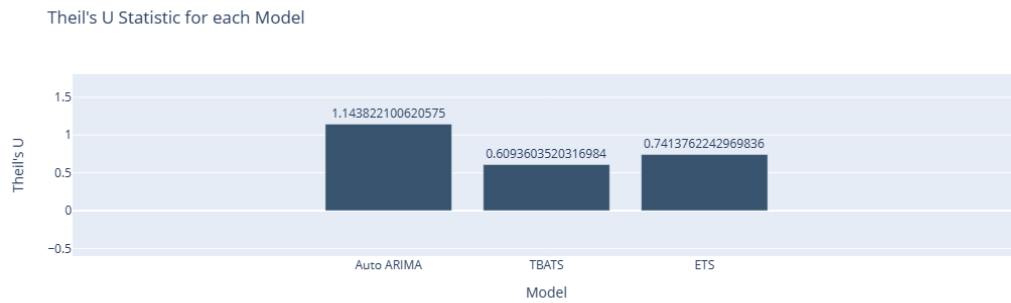
Based on MAPE metrics, we can see that TBATS and ETS models achieve the higher accuracy (~ 15) and out performs Auto Arima and Neural Network models. TBATS (Exponential Smoothing State Space Model With Box-Cox Transformation) and ETS (Exponential Smoothing State Space Model) both use exponential smoothing method. Specifically, TBATS modeling approach offers several key advantages such as handling of typical nonlinear features and allowing any auto-correlation in the residuals to be taken into account .In addition to MAPE metric which is chosen to identify the best model, we also look at Theil's U statistic to estimate how good or bad the model is. In simple words, Theil's U-statistic compares the performance of the model with naïve/ random walk model( $U=1$ ). If Theil's U statistic value equals one, it means that the model forecasting method is as good as naïve model (guessing). A value greater than one means the forecasting method is even worst than guessing. Similarly, the value less than 1 indicates that the forecasting method is better than naïve model and worth considering .From the comparison, we can see that all three models have Theil's U score less than one while TBATS and ETS models having a comparatively good score of 0.6

Theil's U for Auto ARIMA: 1.143822100620575  
Theil's U for TBATS: 0.6093603520316984  
Theil's U for ETS: 0.7413762242969836

*Figure 8.15: Theil's U Score*



*Figure 8.16: Model Performance Metrics*



*Figure 8.17: Theil's U Statistic for each Model*

# **Chapter 9**

## **Testing**



## Chapter 9

### Testing

This chapter explain testing phase of our project. We have carried out numbers of test cases during this process. We have designed all the possible test cases which covers the whole functionality of our project. We have successfully conducted all test cases.

#### 9.1: Test Case for Sign Up

*Table 9.2 Sign Up Test Case*

Sign Up Test Case	
Test Engineer	Abu Bakar
Test Case ID	TC-1
Date	September 07, 2023
Purpose	Signup into system
Pre-Request	Signup function
Test Data	<ul style="list-style-type: none"><li>Name, Valid Email and Password , Invalid Email or Password</li></ul>
Steps	<p>Following steps will take place in test.</p> <p>Case-1</p> <ul style="list-style-type: none"><li>Filled all the required fields like name: Abu Bakar, Email: baker4508@gmail.com and Password: XXX</li><li>Click on Register Button</li><li>System redirect to login page that indicate signup account successfully</li></ul> <p>Case-2</p> <ul style="list-style-type: none"><li>Name , Email or Password are not filled</li><li>Click on Register Button</li><li>Response shows name, email password fields are required</li></ul> <p>Case-3</p> <ul style="list-style-type: none"><li>Enter invalid Name   Email    Password</li><li>Click on Register Button</li></ul>

- 
- System show error message invalid data is input please enter valid Name|| Email || Password

Case-4

- Enter an email which is already used for another account
- Response show Email already exists.

---

<b>Status</b>
---------------

Pass
------

---

## 9.2: Test Case for Login

Table 9.3 Login Test Case

Login Test Case	
Test Engineer	Abu Bakar
Test Case ID	TC-2
Date	September 07, 2023
Purpose	Logged into system.
Pre-Request	Not logged in , must have an account
Test Data	<ul style="list-style-type: none"><li>Email, Password and User type.</li></ul>
Steps	<p>Following steps will take place in test.</p> <p>Case-1</p> <ul style="list-style-type: none"><li>Filled the valid email address, password and user type</li><li>Click on the Sign in button</li><li>Successfully login to the system</li></ul> <p>Case-2</p> <ul style="list-style-type: none"><li>Email, Password or User type are not filled</li><li>Click on Sign in Button</li><li>Response show Please fill email, password and user type are required fields</li></ul> <p>Case-3</p> <ul style="list-style-type: none"><li>Enter an Invalid Email or Password</li><li>Click on Sign in Button</li><li>Response show Please enter correct email or password</li></ul>
Status	Pass

### 9.3: Test Case for Admin Login

Table 9.3: *Admin Login Test Case*

Login Test Case	
Test Engineer	M Ahmed Raza
Test Case ID	TC-3
Date	September 07, 2023
Purpose	Logged into system.
Pre-Request	Not logged in , must have an account
Test Data	<ul style="list-style-type: none"><li>Email, Password.</li></ul>
Steps	<p>Following steps will take place in test.</p> <p>Case-1</p> <ul style="list-style-type: none"><li>Filled the valid email address, password and user type</li><li>Click on the Sign in button</li><li>Successfully login to the system</li></ul> <p>Case-2</p> <ul style="list-style-type: none"><li>Email, Password or User type are not filled</li><li>Click on Sign in Button</li><li>Response show Please fill email, password and user type are required fields</li></ul> <p>Case-3</p> <ul style="list-style-type: none"><li>Enter an Invalid Email or Password</li><li>Click on Sign in Button</li><li>Response show Please enter correct email or password</li></ul>
Status	Pass

#### 9.4: Test Case for Logout

Table 9.4: *Logout Test Case*

Login Test Case	
Test Engineer	Ameer Hamza
Test Case ID	TC-4
Date	September 07, 2023
Purpose	Logged into system.
Pre-Request	Not logged in , must have an account
Test Data	<ul style="list-style-type: none"><li>Email, Password.</li></ul>
Steps	<p>Following steps will take place in test.</p> <p>Case-1</p> <ul style="list-style-type: none"><li>Filled the valid email address, password and user type</li><li>Click on the Sign in button</li><li>Successfully login to the system</li></ul> <p>Case-2</p> <ul style="list-style-type: none"><li>Email, Password or User type are not filled</li><li>Click on Sign in Button</li><li>Response show Please fill email, password and user type are required fields</li></ul> <p>Case-3</p> <ul style="list-style-type: none"><li>Enter an Invalid Email or Password</li><li>Click on Sign in Button</li><li>Response show Please enter correct email or password</li></ul>
Status	Pass

## 9.5: Test Case for Forget Password

Table 9.5: Forget Password Test Case

Login Test Case	
Test Engineer	Ameer Hamza
Test Case ID	TC-5
Date	September 07, 2023
Purpose	Logged into system.
Pre-Request	Not logged in , must have an account
Test Data	<ul style="list-style-type: none"><li>Email, Password and User type.</li></ul>
Steps	<p>Following steps will take place in test.</p> <p>Case-1</p> <ul style="list-style-type: none"><li>Filled the valid email address, password and user type</li><li>Click on the Sign in button</li><li>Successfully login to the system</li></ul> <p>Case-2</p> <ul style="list-style-type: none"><li>Email, Password or User type are not filled</li><li>Click on Sign in Button</li><li>Response show Please fill email, password and user type are required fields</li></ul> <p>Case-3</p> <ul style="list-style-type: none"><li>Enter an Invalid Email or Password</li><li>Click on Sign in Button</li><li>Response show Please enter correct email or password</li></ul>
Status	Pass

## 9.6: Test Case for Prediction System

Table 9.6: Prediction System Test Case

Prediction System Test Case	
Test Engineer	Abu Bakar
Test Case ID	TC-5
Date	September 07, 2023
Purpose	Predict terrorism events and generate reports of deaths and injuries.
Pre-Request	Not logged in , must have an account
Test Data	• Predicted Event Data • Historical Data • Location Data • Time Data
Steps	<p>Following steps will take place in test.</p> <p>Case 1: Predict Event with Valid Data</p> <p>Steps:</p> <ul style="list-style-type: none"><li>• Input all required fields for a predicted terrorism event:</li><li>• Event Description: Bombing at central market</li><li>• Location: Central Market, City XYZ</li><li>• Date and Time: June 1, 2024, 14:00</li><li>• Predicted Deaths: 50</li><li>• Predicted Injured: 150</li><li>• Click on the "Predict" button.</li></ul> <p>Expected Outcome:</p> <ul style="list-style-type: none"><li>• The system processes the input data and predicts the event.</li><li>• A report is generated displaying:</li><li>• Event Description</li><li>• Location</li><li>• Date and Time</li><li>• Predicted Deaths</li></ul>

- 
- Predicted Injured

#### Case 2: Missing Required Fields

##### Steps:

- Leave one or more of the required fields (Event Description, Location, Date and Time, Predicted Deaths, Predicted Injured) empty.
- Click on the "Predict" button.

##### Expected Outcome:

- The system displays an error message indicating that all fields are required to make a prediction.

#### Case 3: System Error Handling

##### Steps:

- Simulate a scenario where the system experiences a technical issue during prediction (e.g., database connection failure).
- Click on the "Predict" button.

##### Expected Outcome:

- The system displays a user-friendly error message indicating that a technical issue occurred and prompts the user to try again later or contact support.

---

<b>Status</b>	Pass
---------------	------

---



## 9.6: Test Case for Prediction

Table 9.6: Prediction Test Case

Login Test Case	
Test Engineer	M Ahmed Raza
Test Case ID	TC-6
Date	September 07, 2023
Purpose	Logged into system.
Pre-Request	Not logged in , must have an account
Test Data	<ul style="list-style-type: none"><li>Email, Password and User type.</li></ul>
Steps	<p>Following steps will take place in test.</p> <p>Case-1</p> <ul style="list-style-type: none"><li>Filled the valid email address, password and user type</li><li>Click on the Sign in button</li><li>Successfully login to the system</li></ul> <p>Case-2</p> <ul style="list-style-type: none"><li>Email, Password or User type are not filled</li><li>Click on Sign in Button</li><li>Response show Please fill email, password and user type are required fields</li></ul> <p>Case-3</p> <ul style="list-style-type: none"><li>Enter an Invalid Email or Password</li><li>Click on Sign in Button</li><li>Response show Please enter correct email or password</li></ul>
Status	Pass

9.7: Test Case for Report Generation

Table 9.6: Prediction Test Case

9.7: Test Case for Report Generation

Table 9.7: Report Generation Test Case

Login Test Case	
Test Engineer	Abu Bakar
Test Case ID	TC-7
Date	September 07, 2023
Purpose	Generate reports for the number of deaths and injured individuals in a predicted terrorism even
Pre-Request	System is functional and has access to required datasets.
Test Data	• Predicted Event Data• Historical Data• Location Data• Time Data.
Steps	Case 1: Generate Report for Predicted Event Steps: Input all required fields for a predicted terrorism event: Event Description: Bombing at central market Location: Central Market, City XYZ Date and Time: June 1, 2024, 14:00 Predicted Deaths: 50 Predicted Injured: 150 Click on the "Generate Report" button. Expected Outcome:

<hr/>	
The system generates a report displaying:	
Event Description	
Location	
Date and Time	
Predicted Deaths	
Predicted Injured	
Case 2: System Error Handling	
Simulate a scenario where the system experiences a technical issue during report generation (e.g., database connection failure).	
Click on the "Generate Report" button.	
Expected Outcome:	
The system displays a user-friendly error message indicating that a technical issue occurred and prompts the user to try again later or contact support.	
<hr/>	
Status	Pass
<hr/>	

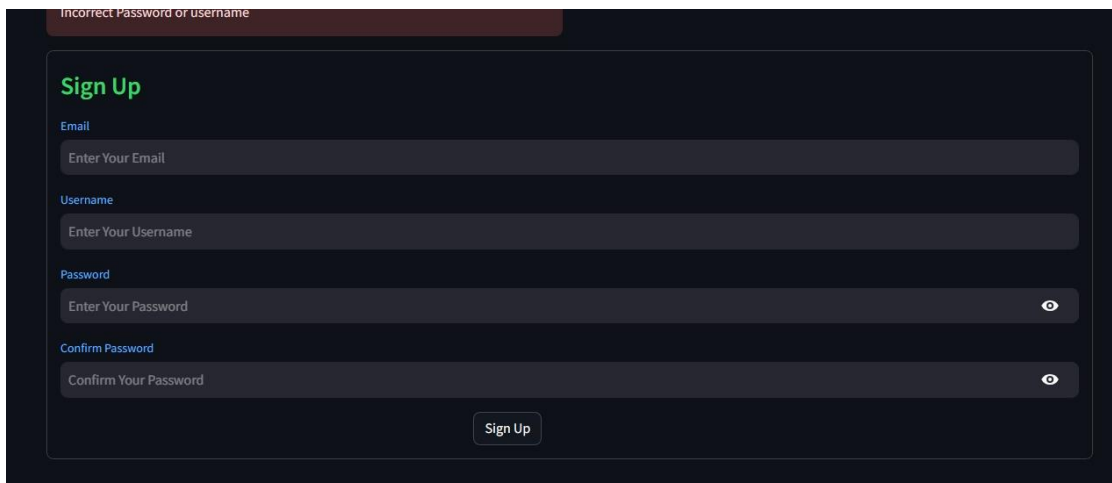
# **Chapter 10**

## **User Manual**

## Chapter 5 User Manual

### 5.1: Signup

Every General people approach to our system they must be sign up first. That information is use in further processing.

A screenshot of a web application's sign-up form. At the top, a dark red error banner displays the text "Incorrect Password or username". Below this, the form is titled "Sign Up" in green. It contains four input fields: "Email" with the placeholder "Enter Your Email", "Username" with "Enter Your Username", "Password" with "Enter Your Password" and an eye icon for toggling visibility, and "Confirm Password" with "Confirm Your Password" and an eye icon. A "Sign Up" button is positioned at the bottom center of the form area.

*Figure 5.1: Signup*

### 5.2: Login

This feature provides secure access to the administrative dashboard of the Terrorism prediction: A Challenge for the 21st Century platform. Authorized users can log in with their credentials to, operations, and access detailed analytics essential for the effective running of the.

Figure 5.2: Login

### 5.3: Dashboard

The Dashboard serves as the central command center for Terrorism prediction: A Challenge for the 21st Century. This intuitive interface allows you to monitor terrorism data and insights. Key features include comprehensive analytics panels for terrorism prediction.

Figure 5.3: Dashboard

## 5.4: Dead Analysis

This section of dashboard contain the graph of death based on death can occur in predicted event

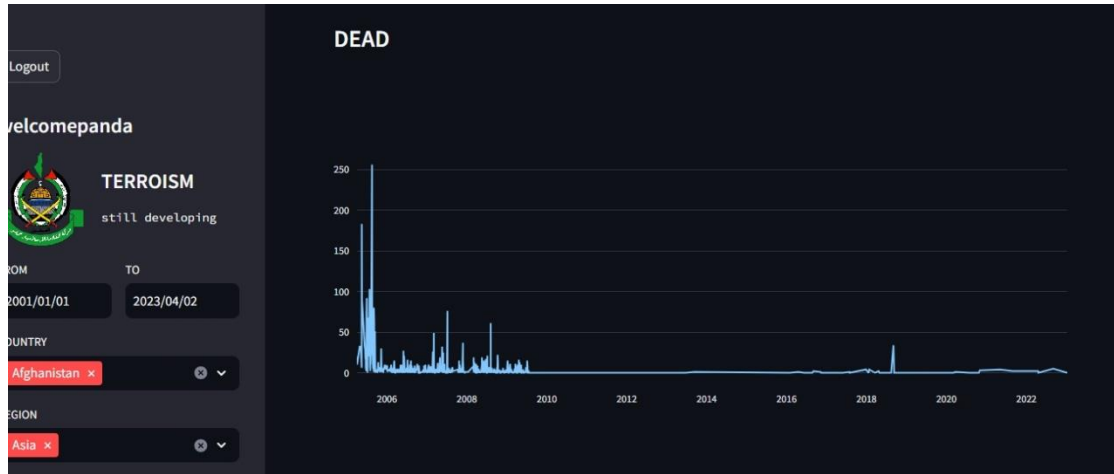


Figure 5.4: Dead Analysis

## 5.5: Injured Analysis

This section of dashboard contains the graph of injured person based on injuries can occur in predicted event

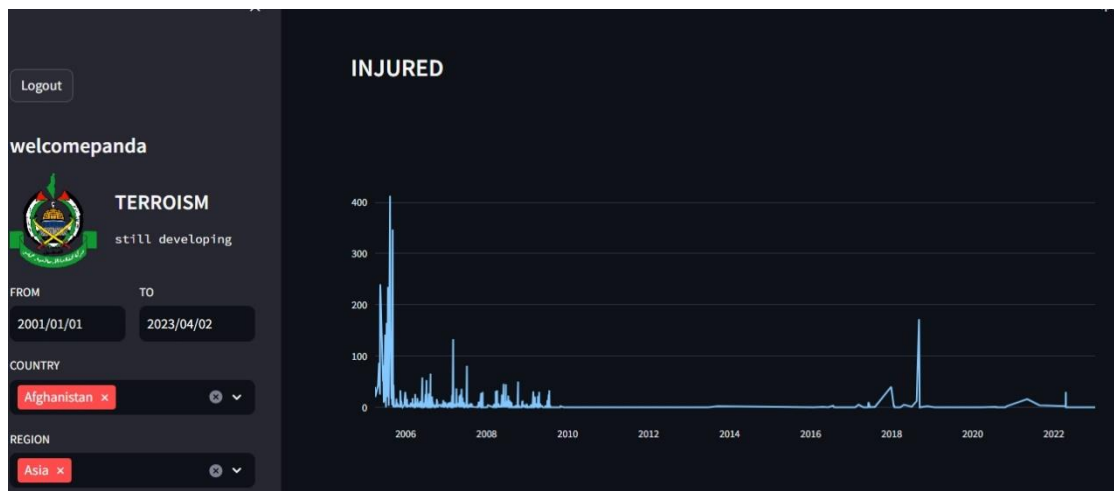


Figure 5.5: Injured Analysis

## 5.6: Report generated

The last section of dashboard where we can find the detail report of out predicted event given by the user the user can further download the report for his personal use also

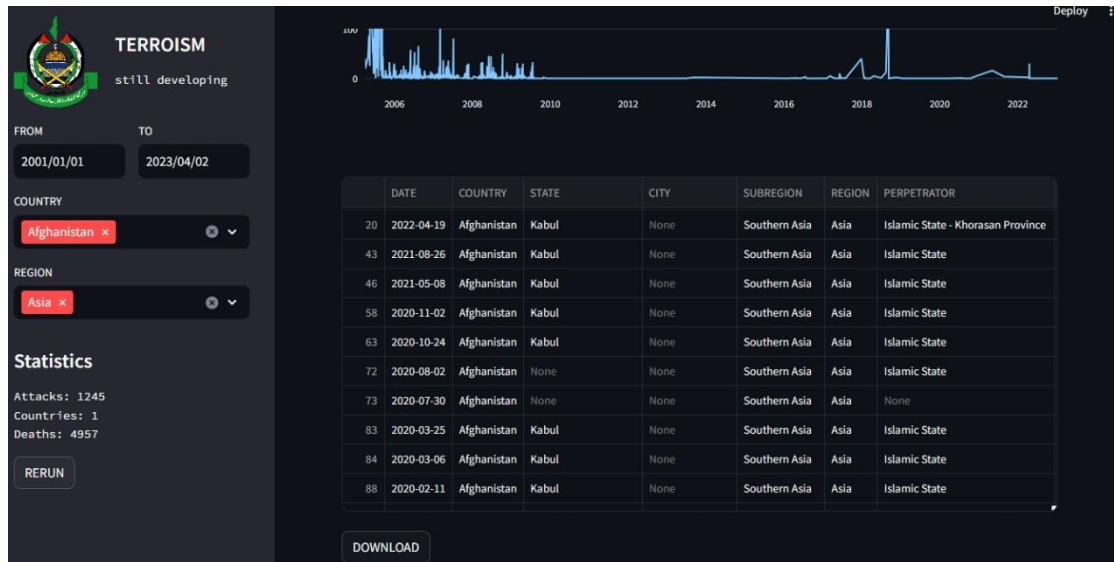


Figure 5.6: Report generated

## References:

For URL:



<https://www.aljazeera.com/news/middleeast/2014/06/isil-declares-new-islamic-caliphate-201462917326669749.html>

**For a paper in a journal/ conference:**

Terrorist attacks and oil prices: Hypothesis and empirical evidence [2021]

Comparison of Machine Learning Approaches in the Prediction of Terrorist Attack  
(survey paper)

**For a book:**

(Springer Monographs in Mathematics) Erich Lehmann, Joseph P. Romano - Testing statistical hypotheses-Springer (2008)

(Springer Texts in Statistics) Gareth James\_ Daniela Witten\_ Trevor Hastie\_ Robert Tibshirani - An Introduction to Statistical Learning with Applications in R-Springer (2023)