Natural Disaster Clustering Using K-Means, DBSCAN, SOM, GMM, and Mean Shift: An Analysis of Fema Disaster Statistics

Ting Tin Tin[1]*, Yap Jia Hao[2], Yong Chang Yeou[3], Lim Siew Mooi[4],
Goh Ting Yew[5], Temitope Olumide Olugbade[6] and Ali Aitizaz[7]
Faculty of Data Science and Information Technology, INTI International University, Negeri Sembilan, Malaysia[1]
Tunku Abdul Rahman University of Management and Technology, Kuala Lumpur, Malaysia[2, 3, 4, 5]
University of Dundee, Dundee, United Kingdom[6]
School of Technology, Asia Pacific University, Malaysia[7]

***Abstract*—Natural disasters tend to ruin people's lives and infrastructure, which requires comprehensive analysis and understanding to inform effective disaster management and response planning. This research addresses the lack of in-depth analysis of federally declared disasters in the United States using a dataset sourced from FEMA. Through the application of unsupervised learning techniques, including K-means clustering, DBSCAN, self-organizing maps (SOM), and the Gaussian mixture model (GMM), similar types of disasters are clustered based on their frequency. The relationship between disaster type and disaster frequency is analyzed to gain insight into patterns and correlations, facilitating targeted mitigation and adaptation strategies. By using the techniques of clustering, we can accurately group similar disaster types, duration time, occurring time and location of disaster. By implementing these approaches, our study aims to improve the understanding of disaster occurrences and inform decision-making processes in disaster mitigation strategies and adaptation strategies.**

***Keywords*—Natural disasters; disaster management; unsupervised learning; clustering; disaster frequency; disaster types; mitigation strategies; adaptation strategies**

I. Introduction

The United States (USA) faces a wide range of natural disasters annually, including hurricanes, tornadoes, wildfires, floods, heat waves, thunderstorms, and flash floods, all of which pose significant threats to lives and cause extensive damage. For example, $ 182.5 billion was lost in Hurricane Katrina 2005 [1]. In 2022, there are a total of 119 natural disasters occurred in the United States, 52% (62 cases) of severe thunderstorms, 21.8% (26 cases) of wildfires, heat waves, and drought, and 12.6% (15 cases) of floods and flash floods [2]. In the same year 2022, 1143 tornadoes were reported with an inconsistent pattern of occurrence throughout 1995-2022, as shown in Fig. 1 [3]. A total of 466 deaths were reported in 2022 due to natural disasters, among which 33.7% (157 fatalities) were due to a tropical cyclone [4]. With these occurrences of disasters and their impacts on humans and property, disaster management comes into the picture to alleviate the suffering caused by disasters. The four main components in disaster management include mitigation, preparedness, response, and recovery. If a country could reduce the risk of loss by predicting the occurrence of disasters, it could significantly avoid unnecessary and severe consequences. Though there is research done in predicting rainfall, streamflow, etc. less research uses the real world big data set to predict the disaster using machine learning analytics algorithms [5], [6], [7], [8], [9]. Therefore, much research is necessary to predict disaster occurrence, especially using big data analytics.

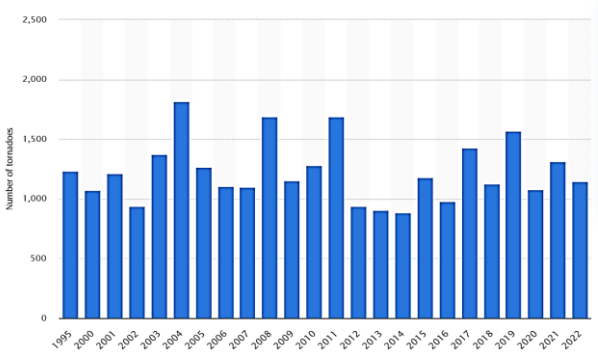


Fig. 1. Number of tornadoes reported in the USA from 1995-2022 [3].

This study uses a dataset, sourced from the Federal Emergency Management Agency (FEMA), and is regularly updated, offering a comprehensive overview of federally declared disasters since 1953 [10]. It includes data on biological disasters, notably declarations related to the ongoing Covid-19 pandemic. The data set has undergone basic cleaning and formatting measures. Additionally, a subset tailored to parameters relevant to the M5 forecast competition is provided, allowing for specific analysis and forecasting tasks related to disaster occurrences. We need to analyze these disasters more deeply, including how often they happen, what types occur, and how they affect people and places. This will help us plan better responses and ways to prevent disasters.

The remaining paper is constructed with the first overview of existing research done using different algorithms (K-Means clustering, density-based spatial clustering of Noise Applications, Self-Organising Maps, and Gaussian mixture model). This is followed by research methodologies that describe the steps to process the data set and construct forecasting models. The results and discussion are presented

*Corresponding Author.

with the content of preprocessing techniques, exploratory data analysis, robust scaler and descriptive analysis, clustering modeling, K-Means clustering, Gaussian mixture clustering, self-organizing maps, density-based spatial clustering of noise applications, and mean shift clustering. Lastly, the conclusion is presented based on the research result and discussion.

II. Literature Review

A. K-Means Clustering

Clustering techniques, particularly K-Means Clustering, play a crucial role in various domains such as customer segmentation, fraud detection, and targeted marketing. In customer segmentation, K-Means clustering enables businesses to group customers according to preferences, demographics, and purchasing behavior, facilitating the development of customized marketing strategies to meet diverse customer needs [11]. In fraud detection, clustering algorithms identify patterns in consumption habits, helping to detect potentially fraudulent activities by detecting anomalies in customer behavior and transactions [12]. Furthermore, clustering techniques are also valuable in targeting client incentives, as they allow businesses to segment customers with similar behaviors and preferences, enabling the offering of targeted incentives to encourage specific actions and increase sales or customer engagement [13]. In general, clustering techniques offer versatile solutions for understanding customer behavior, detecting fraud, and optimizing marketing strategies to improve business performance.

Chakraborty & Nagwani (2014) conducted a project that employs K-means clustering for weather forecasting, leveraging incremental K-means to enhance the model's adaptability to new data. According to the PDF, this methodology uses historical air pollution data from West Bengal, collected in 2009 and 2010, to predict weather patterns. The process involves initially applying K-means clustering to group data based on air pollutant levels such as CO2, RPM, SO2, and NOx. Each cluster represents a specific weather category defined by the maximum mean values of the pollutants within that cluster. Once the initial clusters are established, the incremental K-means algorithm is used to integrate new data into the existing clusters without re-running the entire algorithm. This approach allows for real-time updating and forecasting. For example, new pollution data for a given day is assigned to the existing cluster that it most closely matches, based on the previously computed means. This assignment helps predict the weather category for the coming days, enhancing the model's responsiveness to changing environmental conditions [14].

Wang et al. (2018) discusses a similar application of clustering techniques, specifically for wind power prediction. Here, K-means clustering is used to categorize wind power data to improve the accuracy of forecasting. The process involves grouping historical wind power data into groups that represent different wind power levels or conditions of wind power. This clustering helps to understand the distribution and variability of wind power, helping to provide more accurate and reliable forecasting [15].

In conclusion, for both research, by grouping similar data points, these methods improve the accuracy of predictions and allow real-time updates. In the context of the weather disaster project, robust scaling ensures that the data are normalized, mitigating the influence of outliers and enhancing the performance of the K-means algorithm. This approach is crucial for accurate weather forecasting and effective disaster management, as it allows for accurate and timely predictions based on continuously updated data.

B. Density-Based Spatial Clustering of Noise Applications (DBSCAN)

Density-Based Spatial Clustering of Noise Applications (DBSCAN), a density-based clustering algorithm, has demonstrated notable effectiveness in diverse fields, with relevance in geographic data analysis and customer segmentation. DBSCAN's proficiency in analyzing geographical data, showcasing its ability to estimate population density within specific metropolitan statistical areas (MSAs) based on location data [16]. This capability has significant implications for urban planning, resource allocation, and demographic studies. Moreover, in the realm of e-Commerce and marketing, Hshan (2022) highlights DBSCAN's utility in customer segmentation, where it can group customers based on their purchasing behaviors or preferences. By leveraging DBSCAN, businesses can devise targeted marketing strategies and offer personalized recommendations, ultimately improving customer engagement and satisfaction [17].

Dey & Chakraborty (2015) conducted a project of the weather forecasting using DBSCAN that utilizes the admissions of this algorithm in finding dense clusters and the detection of outliers in spatial data, thus efficient over the normally complex data sets of weather. In weather forecasting, DBSCAN clusters data points based on their density. For example, grouping together closely packed points and marking isolated points as noise. This approach has paramount suitability for weather data, mainly consisting of dense clusters, such as high rainfall areas, and sparse outliers, such as extreme weather events. These clusters could be used by meteorologists to identify key weather patterns with the aim of establishing future weather forecasts. For instance, clusters of high humidity, combined with low pressure, could indicate the approaching of a storm, hence issuing early warnings to be better prepared in case of disasters. In connection with this, the spatial capabilities of DBSCAN allow it to deal with irregularly shaped clusters; therefore, it is very essential for weather forecasting [18].

C. Self-Organising Maps (SOM)

Self-Organizing Maps (SOM) have emerged as a valuable tool in both image clustering and customer segmentation applications. GeoSense (2023) highlights the ability of SOM to effectively group similar regions or objects within images, enabling the creation of clusters that represent distinct visual elements based on their similarities. This capability has wide-ranging applications in image analysis, from object recognition to scene understanding [19]. Similarly, in the realm of e-Commerce and marketing, Kaushik (2020) underscores the utility of SOM in customer segmentation tasks. By grouping customers according to their purchasing behaviors or preferences, SOM enables businesses to develop targeted marketing strategies and deliver personalized recommendations, thus improving customer satisfaction and engagement. The

versatility of SOM in the image and customer-centric domains makes it an asset to uncover patterns and insights from complex datasets [20].

Mohan & Patil (2018) presented the deep learning-based weighted SOM to enhance the accuracy of weather and crop prediction. The SOM algorithm has been performed by the dimension of the present study so that complicated weather data can be transformed into interpretable clusters. The algorithm maps high-dimensional input data on a lower-dimensional grid while preserving the topological relationships of the data points. This provides the means to identify patterns and similarities within weather data, in order to facilitate more accurate forecasting. In the methodology, latent Dirichlet Allocation is combined with the deep neural network classifier examining, raising the modification prediction precision by up to 23% compared to existing methods [21].

For example, SOM is applied to organize weather into meaningful clusters that represent various conditions of the weather. This clustering may allow better visualization and interpretation of data for meteorologists to detect and predict weather patterns more effectively. Integration with LDA refines the data, hence improving efficiency and accuracy of the DNN classifier in predicting weather. Advanced approaches to weather prediction, such as the one mentioned earlier, which is supported by deep neural networks, enhance decision making in agriculture by allowing farmers to plan activities based on accurate weather forecasts [21].

D. Gaussian Mixture Model (GMM)

The Gaussian mixture model (GMM), as highlighted by Amy (2022), offers an effective approach to anomaly detection by identifying outliers within low-density regions of the data distribution. This capability makes GMM particularly suitable for detecting anomalies in datasets with complex or multimodal distributions [22]. On the other hand, identifying restaurant hotspots involves uncovering subgroups within the data set that can improve predictive models or improve understanding. O'Sullivan (2020) emphasizes the importance of this task in the context of restaurant analytics, where identifying hotspots can provide insight into customer preferences, demand patterns, and potential areas for business expansion or optimization [23].

Jouan et al. (2023) applied GMM to the calibration of weather forecasts contributes much to showing how the technique is utilized in clustering ensemble weather forecasts. GMM represents the distribution of the weather variables, which includes weather regimes as different kinds of distribution errors occurring in ensemble forecasts. GMM identifies clusters that reproduce weather patterns and error types in the ensemble data by fitting a mixture model. These clusters help correct for biases and increasing the accuracy of weather forecasts. There are a few steps using GMM. This GMM algorithm models the ensemble data distribution, which is variable and uncertain by nature, just as it is with any weather-related prediction by its very nature. It then identifies clusters within these data, which can be considered to be different weather regimes or error types. A separate calibration model, such as nonhomogeneous Gaussian Regression, is applied to each of the identified clusters for correcting distribution errors. In that respect, cluster-specific calibrations will ensure that accurate adjustments have been made to the forecast distribution for different weather. In this project, GMM to medium-range forecasts of temperature and wind in several locations within France. It significantly improves the interpretability and flexibility of forecasts by identifying and calibrating different kinds of errors related to each cluster [24].

Recent disaster prediction studies point to the use of cutting-edge machine learning models for surge forecasting, aiming to enhance accuracy around natural disaster prediction centers. The application of clustering techniques such as K-means, DBSCAN, Self-Organizing Maps (SOM), and Gaussian Mixture Models (GMM) on diversified disaster datasets has been emphasized in recent research. K-means effectively groups different types of disaster, making it easier to identify and analyze trends or distributions, which aids in emergency preparedness [5]. DBSCAN is a powerful tool for identifying dense regions and outliers within geographical data, enhancing the analysis of spatial disaster distributions and source identification [6]. SOMs provide accurate topological mapping and clustering of disaster-related data, optimizing visualization and interpretation despite some limitations [7]. GMM is efficient for modelling complex multimodal distributions in disaster datasets, particularly useful for anomaly detection. These advanced clustering techniques improve the understanding of disaster events, support better disaster management strategies, and ensure a faster response. The established literature on clustering and predictive modelling further underscores their effectiveness in various domains [7].

III. RESEARCH METHODOLOGY

Fig. 2 summarizes the steps used in this study to clean and transform the FEMA dataset (64092 cases, 24 variables): data preprocessing, exploratory data analysis, scaling using robust scaler, descriptive analysis, clustering modelling, visualization, analysis and conclusion. First, the data set is cleaned and transformed by dealing with missing values, empty cases, data type conversion, and parentheses removal. Once cleansed, the data set is explored using bar chart analysis, time series plot, network visualization, heatmap, frequency visualization, boxplots, violin plot, horizontal bar plot, kernel density estimation (KDE) plot and 2x2 grid of subplots. Data exploration is important to gain insight into the quality and information available. Scaling using a robust scaler is used to improve the performance of clustering models. Meanwhile, descriptive analysis is used to examine the mean, standard deviation, minimum and maximum values, 25%, 50%, 75% quartile of data, to ensure the quality of dataset before continuing cluster modelling. Five machine learning algorithms are used in clustering modelling, which are K-means, DBSCAN, self-organizing maps, Gaussian mixture model, and mean shift model. These models are visualized in 2D and 3D graphics with means as a performance indicator. Finally, silhouette scores are generated to compare five clustering models.

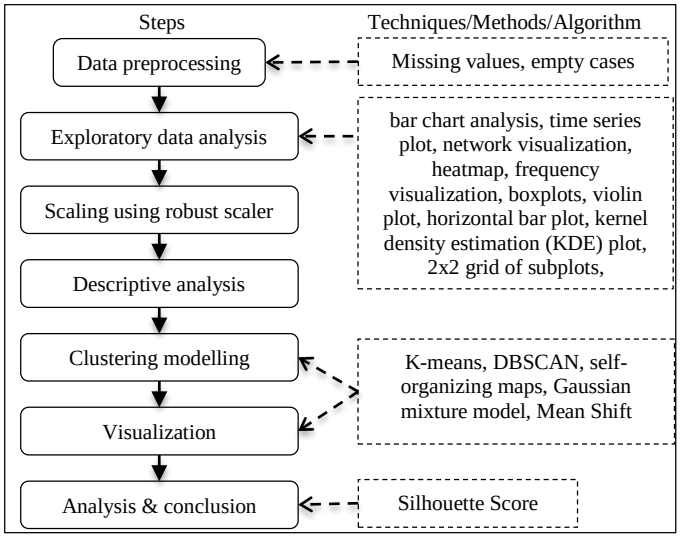


Fig. 2. Steps to process the data set and construct forecasting models.

IV. Results and Discussion

A. Preprocessing Techniques

In this section a detailed description of the step-by-step data preprocessing with the techniques used is presented. First, as shown in Fig. 3, the unique key, the missing values of the variables, and the summary statistics are displayed to facilitate understanding of the quality of the data set. This is to prepare the dataset for the next cleaning process to accurately target the preprocessing techniques. It was found that there are 77.14% blanks in the "last_ia_filing_date" column, which makes it less reliable for research. For accuracy's sake, we suggest getting rid of rows where this column is missing values. With this method, the integrity of the data is maintained and assumptions about missing numbers are avoided. This lets us make decisions based on accurate data. We remove the columns 'last_refresh', 'hash', and 'id' from the dataset as they contain redundant or irrelevant information that does not contribute to our analysis. Then, unique keys of the data set are displayed. Examining the unique values of the 'declaration_title' column in the dataset serves to understand the variety and specific types of disaster declarations recorded. This step is to gain insight into the composition of the data, ensure consistency, and identify any anomalies or duplicates. Fig. 4 illustrates the distribution of the data points and highlights any outliers present in the data set. Outliers are data points that significantly deviate from the rest of the data and may indicate errors, anomalies, or rare events. These outliers will be removed from the data set. The violin plot illustrates the distribution of disaster numbers across different fiscal years (FY declared), the number of flips, and place codes. Provides information on the density and variability of these attributes, highlighting where most disaster occurrences are concentrated and how they vary between different categories.

```
Count of unique values for each column:
fema_declaration_string        4738
disaster_number                4738
state                            59
declaration_type                  3
declaration_date               3324
fy_declared                      71
incident_type                    25
declaration_title              2239
ih_program_declared               2
ia_program_declared               2
pa_program_declared               2
hm_program_declared               2
incident_begin_date            3042
incident_end_date              3082
disaster_closeout_date         1825
fips                           3324
place_code                     1303
designated_area                2953
declaration_request_number     4738
last_ia_filing_date             486
last_refresh                      9
hash                          64092
id                            64092
dtype: int64
```

(a) Unique value for variable.

```
fema_declaration_string            0
disaster_number                    0
state                              0
declaration_type                   0
declaration_date                   0
fy_declared                        0
incident_type                      0
declaration_title                  0
ih_program_declared                0
ia_program_declared                0
pa_program_declared                0
hm_program_declared                0
incident_begin_date                0
incident_end_date               8410
disaster_closeout_date         15180
fips                               0
place_code                         0
designated_area                    0
declaration_request_number         0
last_ia_filing_date            46339
last_refresh                       0
hash                               0
id                                 0
dtype: int64
```

(b) Missing value of the variable.

```
Summary Statistics or Distributions Before Cleaning:
        state declaration_type incident_type  declaration_title
count   64092            64092         64092              64092
unique     59                3            25               2239
top        TX               DR  Severe Storm  Covid-19 Pandemic
freq     5173            43265         17490               4165

Summary Statistics or Distributions After Cleaning:
        state declaration_type incident_type           declaration_title
count   59927            59927         59927                       59927
unique     59                3            25                        2238
top        TX               DR  Severe Storm  Severe Storms And Flooding
freq     4916            39100         17490                        3826
```

(c) Summary statistics.

Fig. 3. Unique key, missing data, and summary statistics of FEMA data set.

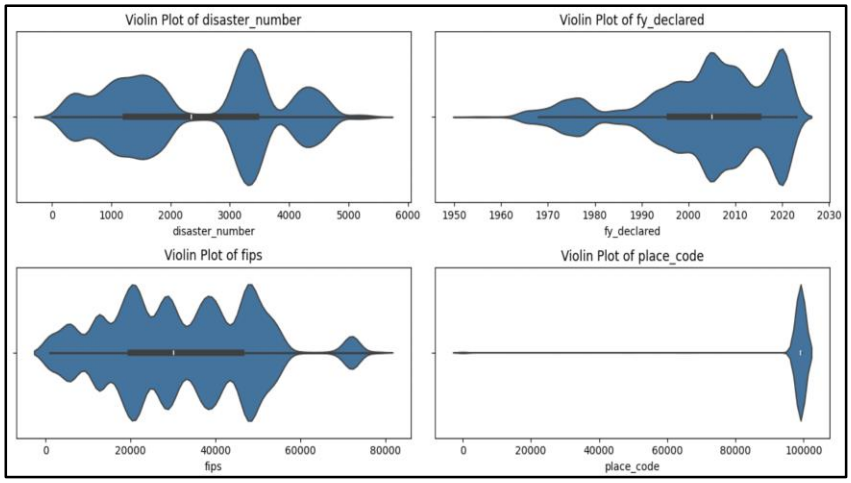


Fig. 4. Distribution of data points to analyse outliers of the data set.

Fig. 5 shows a graph of the Z scores for certain groups of numbers in a dataset. This will find out how far away a data point is from the dataset's mean by its Z-score. Using standard deviations from the mean to figure out Z scores for numerical fields lets us find outliers and see how the data are spread out. Next, rows (cases) where the incident's start date is after the end date are filtered out. This ensures logical consistency, since an incident cannot start after it ends. Filtering ensures valid data for accurate analysis and interpretation. After filtering, 59016 cases remain in the dataset. Several data transformation steps are carried out on the variables which include: 1) Convert the date columns in the data set to datetime format. 2) Convert specific columns in the data set to lowercase to mitigate potential inconsistencies due to variations in capitalization. 3) Remove the paratheses (Fig. 6) from the columns 'declaration_title' and 'designated_area'. After removal, the unique values in these columns are retrieved to observe the changes. Finally, it assigns the modified data set back to itself, although this step is optional. This process helps to clean and standardise the data, removing unnecessary information contained within parentheses. 4) Remove duplicates, which after checking the dataset, no duplicates are found.

The last step in data preprocessing is to convert the categorical values in the 'declaration_type' column to numerical representations for better analysis and modelling (Fig. 7). This is achieved by mapping specific declaration types ('dr', 'em', 'fm') to corresponding numerical values (1, 2, 3) using a predefined dictionary (declaration_type_map). After conversion, the modified data set is displayed to show the changes.

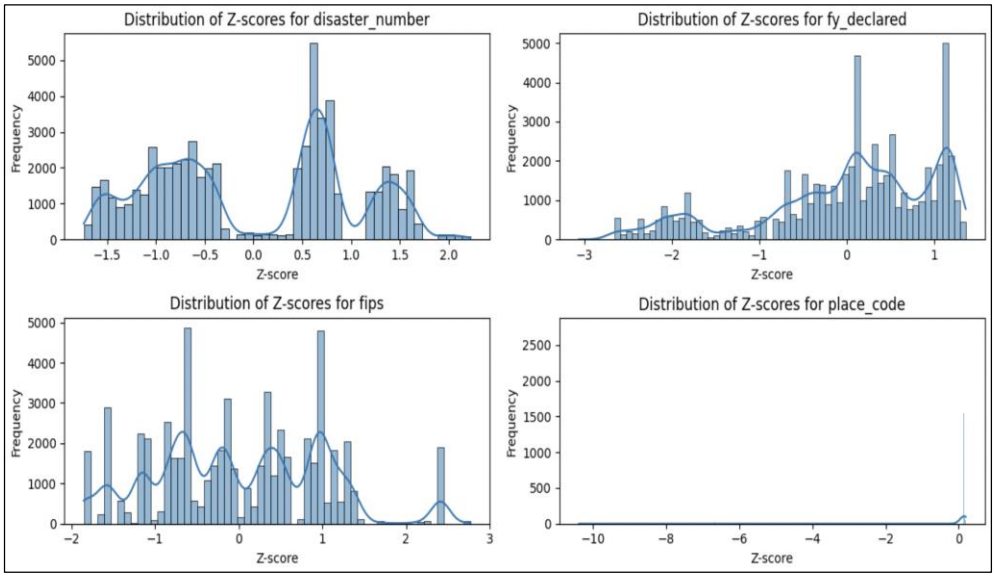


Fig. 5. Z scores of the FEMA dataset.

```
Column 'declaration_title' has 13 unique values containing parentheses.
Unique values:
['fire (city of chelsea)' 'floodng (nys barge canal)'
 'tornadoes, flooding torrential rain(trop storm alberto)'
 'severe storms and flooding (tropical storm alberto)'
 'the el nino (the salmon industry)' 'blizzard of 96 (severe snow storm)'
 'debruce grain elevator explosion  (sedgwick cty)'
 'nv wildfire (pioche) 06-08-2002'
 'ca - wildfire (pacific fire) - 01-06-2003'
 'az-wildfire (edge complex) 07-22-2005' 'parks highway (tamarack) fire'
 'blanco (cr 4901) fire' 'lehigh acres (anna ave. n.) fire']
Column 'designated_area' has 2210 unique values containing parentheses.
Unique values:
['clay (county)' 'baker (county)' 'benton (county)' ...
 'trescott (township of)' 'juneau (borough)'
 'rohnerville rancheria (indian reservation)']
```

```
Unique declaration titles after removal:
['flood' 'heavy rains & flooding' 'severe storms, heavy rains & flooding'
 ... 'tropical storm nicole' 'hurricane nicole'
 'severe winter storm, snowstorm, and straight-line winds']
Unique designated areas after removal:
['clay' 'baker' 'benton' ... 'miccosukee indian reservation'
 'seminole indian trust lands' 'rohnerville rancheria']
```

Fig. 6. Pre-processing of data in parentheses.

	fema_declaration_string	disaster_number	state	declaration_type	declaration_date	fy_declared	incident_type	declaration_title	ih_program_declared	ia_program_declared	pa_program_declared	hm_program_declared	incident_begin_d
89	dr-91-in	91	in	1	1959-01-29	1959	flood	flood	0	1	1	0	1959-0
181	dr-184-or	184	or	1	1964-12-24	1965	flood	heavy rains & flooding	0	1	1	0	1964-12
182	dr-184-or	184	or	1	1964-12-24	1965	flood	heavy rains & flooding	0	1	1	0	1964-12
183	dr-184-or	184	or	1	1964-12-24	1965	flood	heavy rains & flooding	0	1	1	0	1964-12
184	dr-184-or	184	or	1	1964-12-24	1965	flood	heavy rains & flooding	0	1	1	0	1964-12

Fig. 7. Data conversion from categorical values to numerical values.

B. Exploratory Data Analysis (EDA)

Several data visualization techniques are deployed to better understand the FEMA data set, including bar chart, time series plot, network visualization, heatmap, frequency visualization, boxplots, violin plot, horizontal bar plot, kernel density estimation (KDE) plot, and 2x2 grid of subplots. First, a bar chart is created using Seaborn and matplotlib to display the count of different incident types per state, as shown in Fig. 8. Set the figure size, rotate the x-axis labels for better readability, and then show the plot. In addition, the high-volume variable states with high-specific incident types are removed. This will make the graph more visible on it. This is because high volume values will make the EDA less accurate and visible, as the bar shown will not be clear and the bars will be covered by high values. Fig. 9 shows a time series graph generated using matplotlib to visualize the count of natural disaster declarations over time, categorized by incident type. It first converts the "declaration_date" column to a date-time format. Then, it groups the data by month and incident type, calculates the count of each type, and creates a time series plot. The plot is customized with a title, labels for the x- and y-axes, a legend showing the incident types, and an adjusted layout for better visualization. Fig. 10 shows a network visualization graph where each node represents a type of disaster (for example, flood, and tornado) and adds edges between pairs of disaster types. The lines indicate relationships or connections between different types of disasters. Finally, it draws the network graph, customizing node and edge properties such as color, size, and labels, and displays the plot with a title. The network diagram is split into four parts: this will enhance the visibility and understanding of the graph with its relations. Each of the figures into 5, 5, 6, 6 types of disasters for better visibility on the relation lines. To conclude, this shows that all types of disaster will relate to each other, in other words, one disaster might trigger another incident to happen.

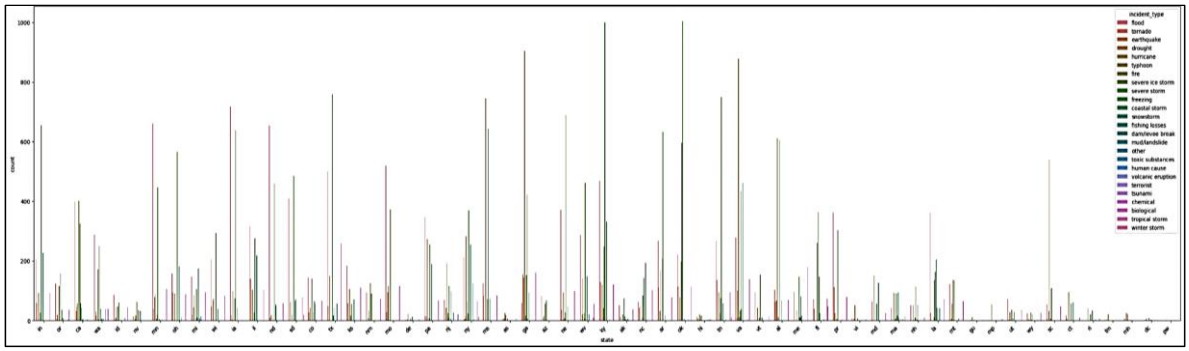
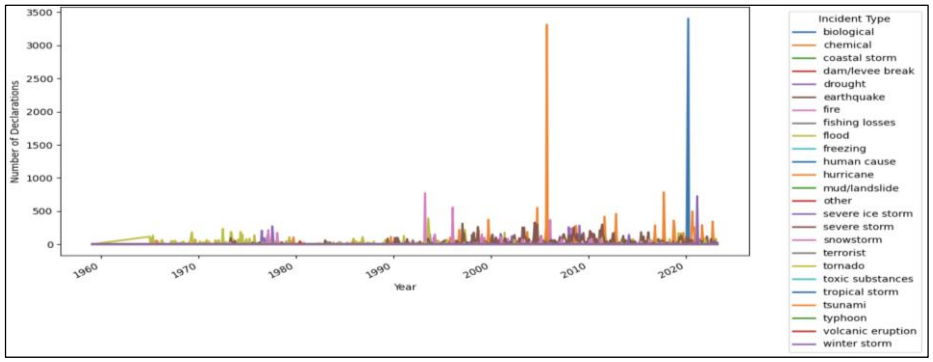
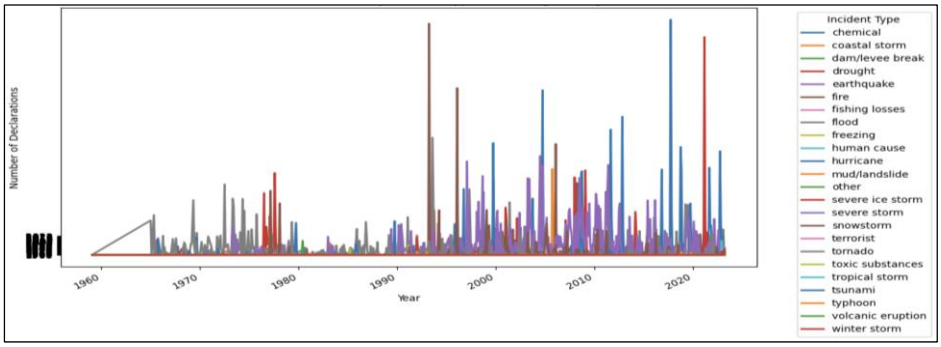


Fig. 8. Bar chart analysis using seaborn and matplotlib.



(a) Natural disaster plot over time by incident type.



(b) Natural disaster plot over time by incident type excluding "biological" and "hurricane".

Fig. 9. Time series plot using Matplotlib to visualise natural disaster over time, categorised by type.

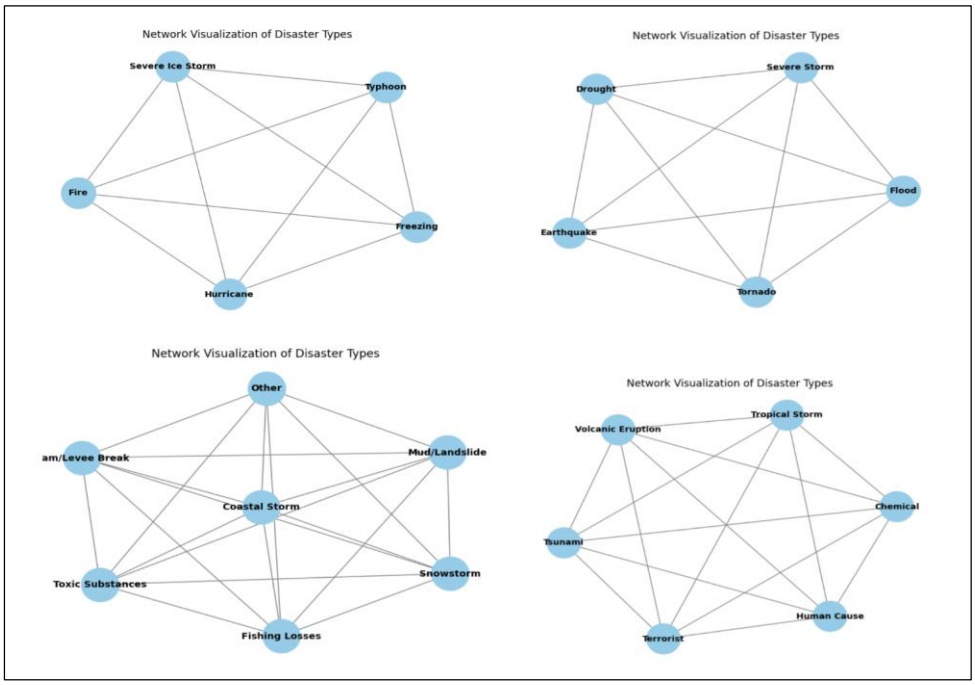


Fig. 10. Visualisation of network of different types using the network library in python.

A random relationship matrix where each cell represents the strength of the relationship between two types of disaster. The heatmap displays these values, with annotations showing the exact values, and uses a color scale (YlGnBu) to represent the strength of the relationships. The x and y axis labels represent the different types of natural disasters. In Fig. 11, it displays the heat map with a title and adjusts the layout for better visualization; each cell shows the actual values of the relationships between the corresponding pair of disasters. These values range from 0 to 1, where 0 indicates that there is no relationship and 1 indicates a perfect relationship. In Fig. 12, it first filters the data set to include only instances where the IH program was declared. Then, it creates two separate counts plots: one displaying the frequency of IH program declarations by incident type and the other showing the frequency by state. Each count plot is custom-made with appropriate titles, labels, and rotation of x-axis labels for better readability. In Fig. 13, two separate boxplots are created by calculating the duration of each incident by subtracting the incident end date from the incident start date. In Fig. 14, the variables "biological" and "fire" are removed from the outlier diagram. This is because fires and biological disasters are the outliers that will contribute the most to the outliers' diagram. The outliers in these appear the most.

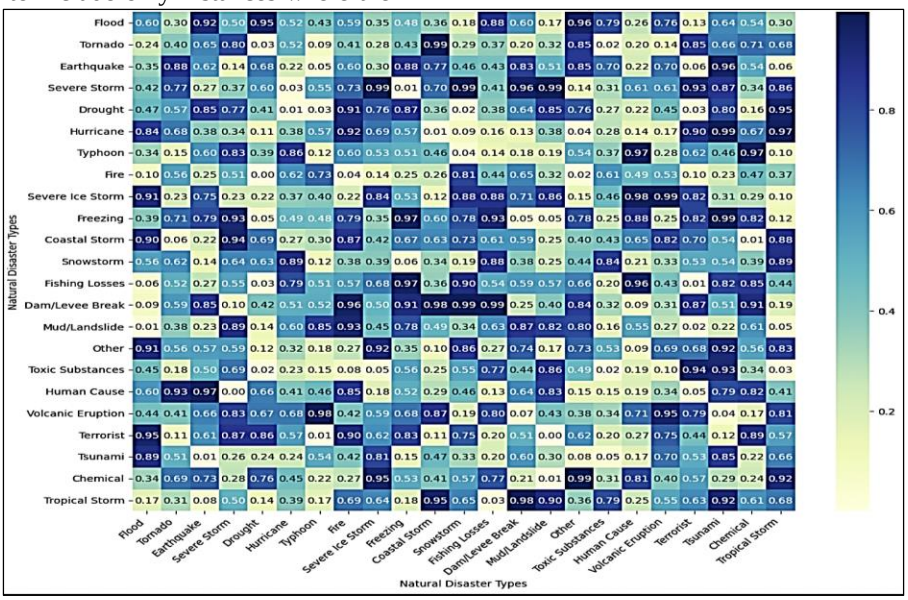


Fig. 11. Heat map revealing the relationships between types of natural disasters.

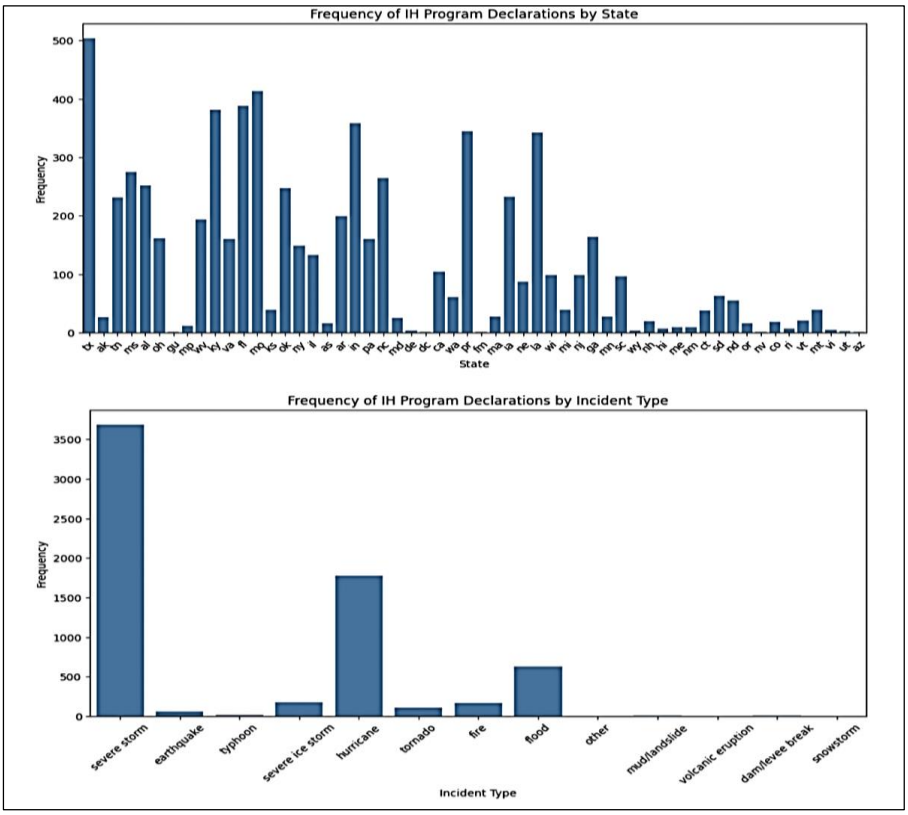


Fig. 12. Visualise the frequency of IH (Individuals and Households) programme declarations.

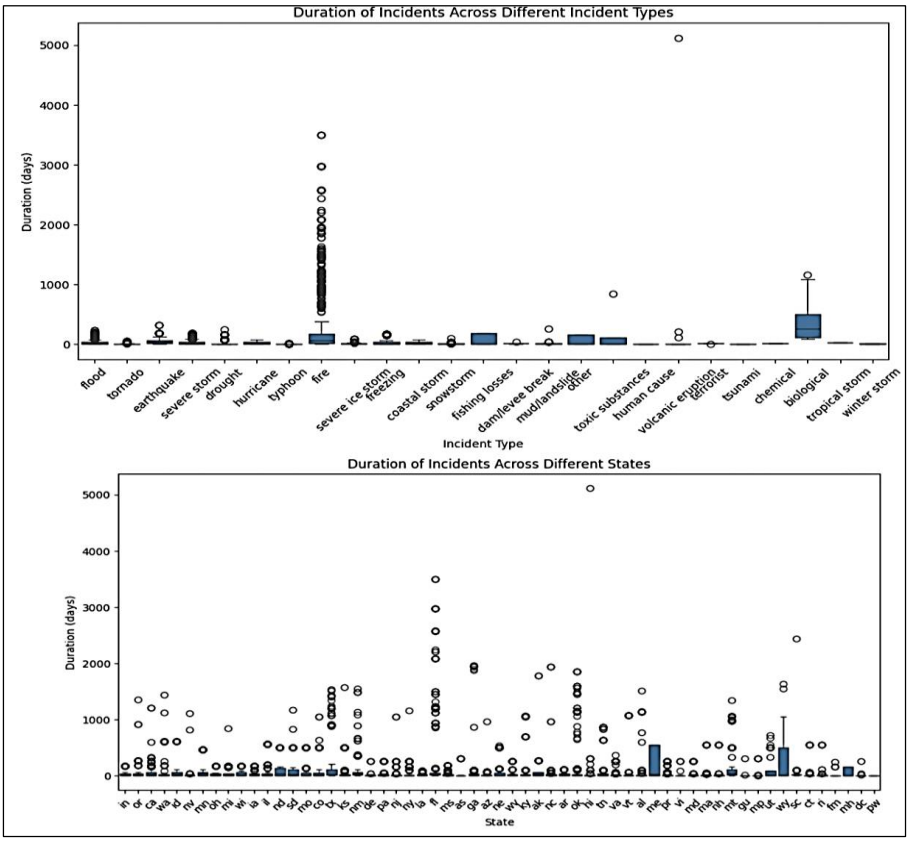


Fig. 13. Visualisation of the duration of incidents.

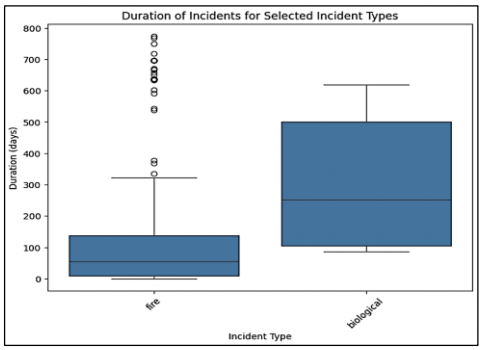
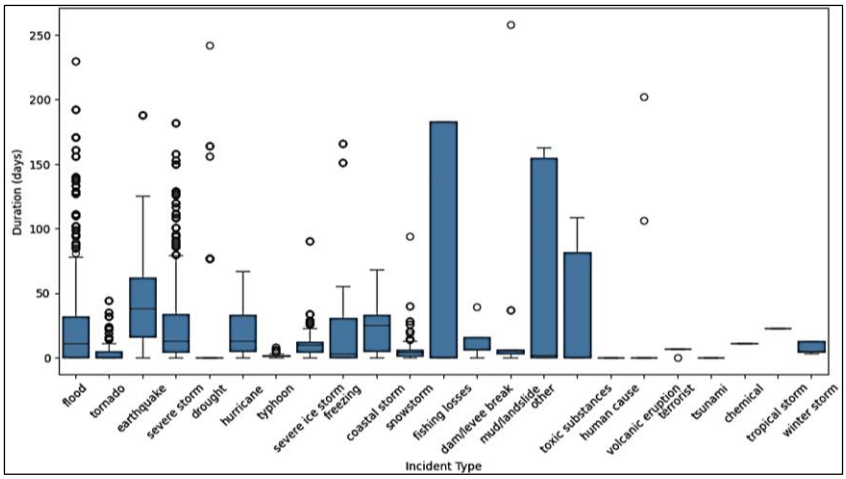


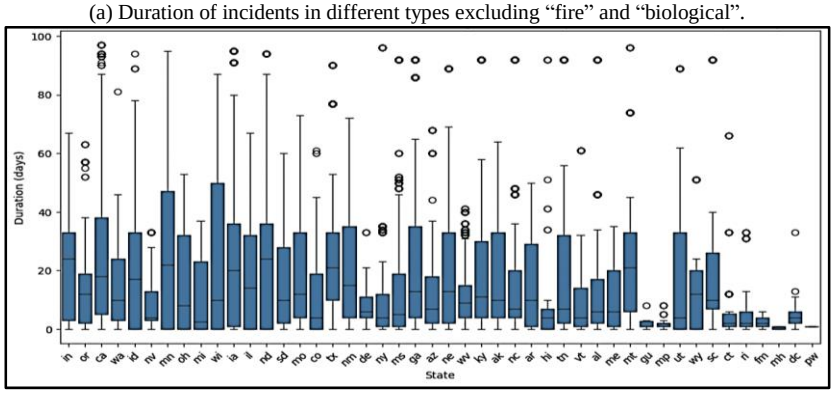
Fig. 14. The variables "biological" and "fire" are removed from the outlier diagram.

The next visualization technique as shown in Fig. 15 reveals the duration of the incident in different types and states, excluding several variables accordingly. The duration in days of each disaster reveals that the average duration is between the range of 50 and 100 days. Different states will have a duration for the incident, and it indicates that the average days for the state are around 30 to 40 days in each state.

Next, we group the data by declaration_date and count the number of disasters declared for each date. A line graph is generated to visualize the trend of the number of disasters declared over time (Fig. 16). The x-axis represents the declaration date, while the y-axis shows the corresponding count of disasters. The plot is customized with a title, axis labels, and grid lines for better interpretation. The second plot is where several years 2020 and 2008 are removed from the graph in order to make it more visible. This will make the number of disasters more clearly indicated. This is continued in Fig. 17 which is a generated violin plot where each violin represents the distribution of incident durations for a specific type of incident. Incident of 'fire', 'human cause', 'tsunami', 'chemical', 'biological', 'tropical storm' up to 3000 days, to ensure that the violet plot is more accurate in its presentation. A horizontal bar plot first calculates the frequency of each incident type using the value_counts() function. Then it creates a horizontal bar graph (kind = barh) where each type of incident is represented on the y-axis and its frequency on the x axis (Fig. 18). The time taken to close the disaster after declaration is calculated by subtracting the declaration dates from the closing dates. The resulting time differences are then converted into days. The kernel density estimation (KDE) plot is generated using seaborn to visualize the distribution of time taken for closeout. The KDE plot represents the estimated probability density function of the data. It provides information on the density of observations in different time intervals (Fig. 19). Lastly, a snippet of the 2x2 grid of subplots generated using Seaborn and Matplotlib is represented in Fig. 20. Each subplot represents the count of occurrences of Boolean values (True/False) in specific columns of a dataset.



(a) Duration of incidents in different types excluding "fire" and "biological".



(b) Duration of incidents in different states (excluding fl, ok, va, la, pa, ks, vi, md, nh, ma, pr, as, nj).

Fig. 15. Duration of incidents in different types and states.

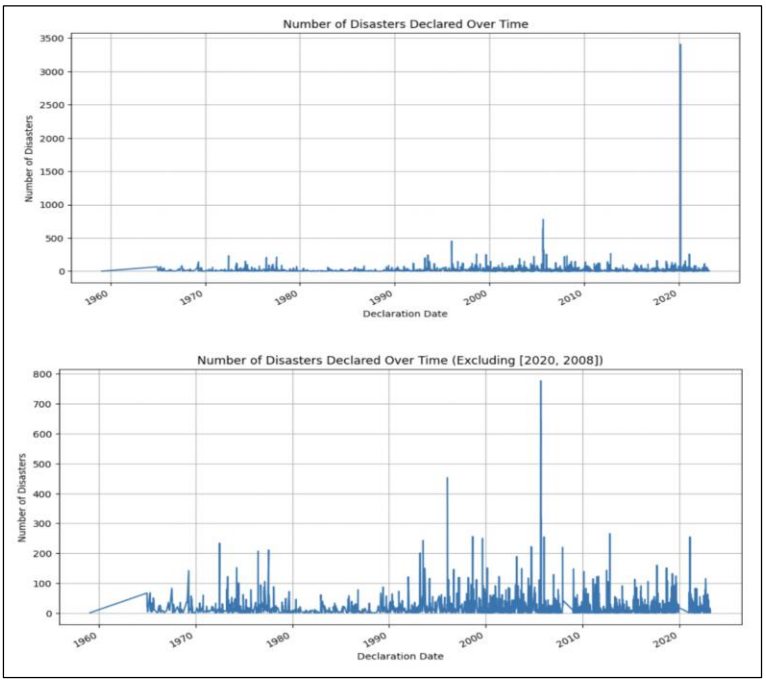


Fig. 16. Trend in the number of disasters declared over time.

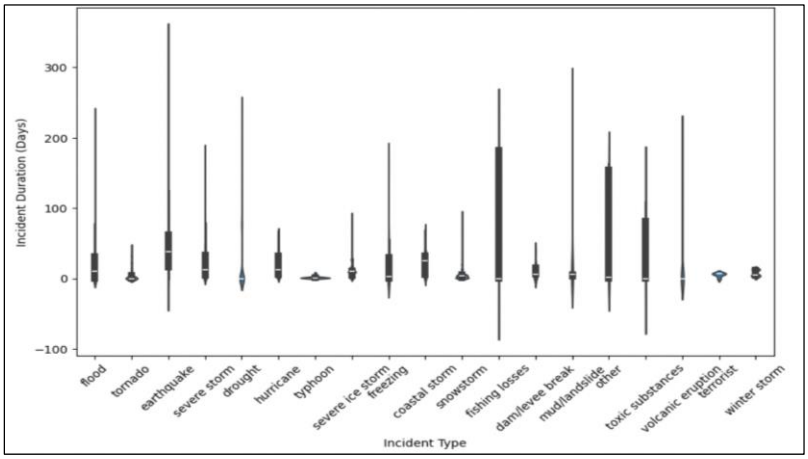


Fig. 17. Violin plot – distribution of incident types based on incident duration.

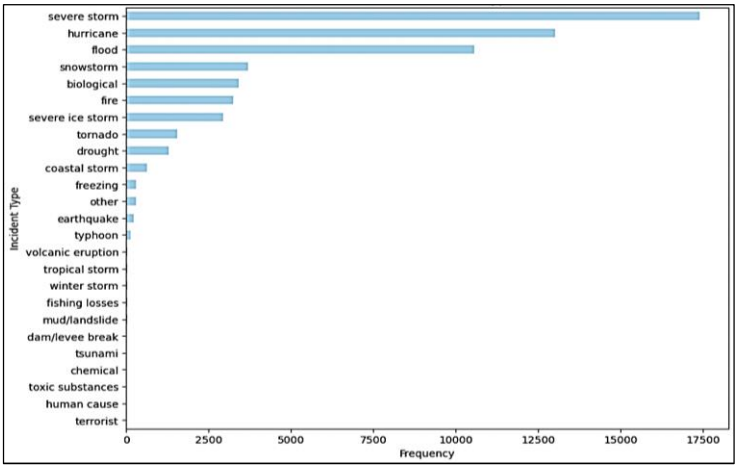


Fig. 18. Horizontal bar plot to visualise the distribution of incident types within a data set.

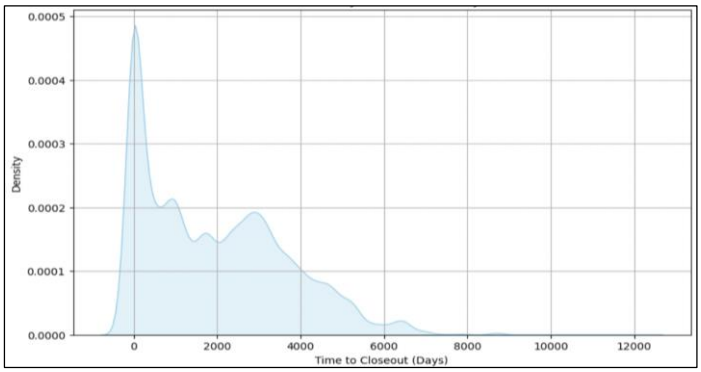


Fig. 19. Time-to-closeout analysis (Kernel density estimation).

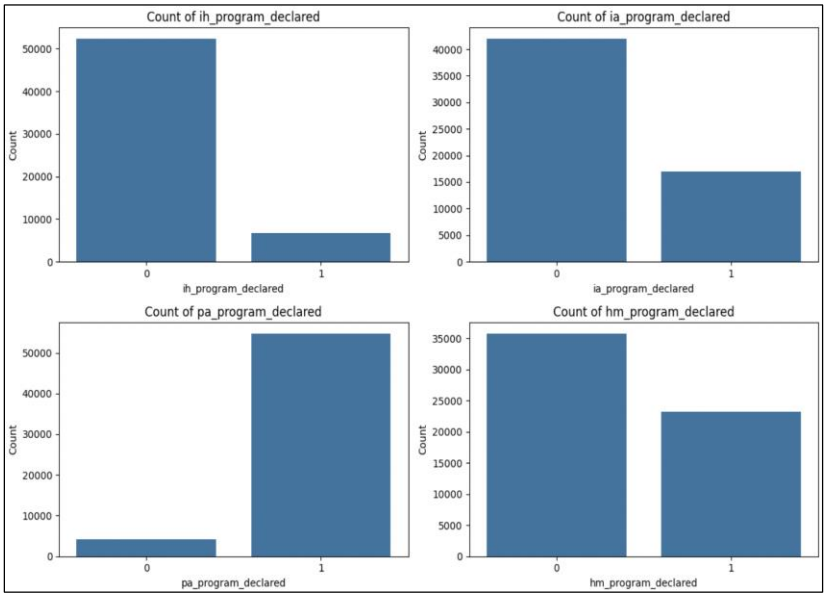


Fig. 20. 2x2 Grid of subplots for several variables.

C. Robust Scaler and Descriptive Analysis

The weather disaster data set is preprocessed using robust scaling to improve the performance of these clustering algorithms: K-means, DBSCAN, SOM, and GMM. This step is important because weather-related datasets often contain unique values, a nature that is heavily influenced by disasters. Robust scaling uses the median and interquartile range, mitigating the influence of these outliers on the central tendency of data. Moreover, normalization ensures that distances between points indicate the actual similarities of the data entries more closely and, therefore, enhance accuracy in clustering. Furthermore, it is easier to visualize and interpret normalized data for effective dissemination of results to interested parties, such as agencies concerned with disaster management.

Descriptive analysis is used to gain an initial understanding of the weather disaster dataset as shown in Table I: count, mean, standard deviation, min and max. This also involves summarizing the main characteristics of the data through statistical measures and visualizations. By performing descriptive analysis, it can identify key patterns, trends, and anomalies within the data, such as the frequency of different types of disasters, the distribution of disaster occurrences over time, and the geographical locations most affected.

TABLE I. DESCRIPTIVE ANALYSIS OF THE WEATHER DATA SET

Descriptive Statistics	Variables		
	declaration_ type	incident_ duration	state_label_ encoded
Count	58944	58944	58944
Mean	1.3666	45.2700	30.6819
SD	0.5321	132.3896	16.1234
Min	1.0000	0.0000	0.0000
25%	1.0000	3.0000	18.0000
50%	1.0000	13.0000	31.0000
75%	2.0000	33.0000	44.0000
Max	3.0000	5117.0000	58.0000

It helps in preprocessing the weather disaster dataset by providing a clear picture of the data structure and quality in this project. For example, it highlights issues such as missing values, outliers, and inconsistencies, which can then be addressed through appropriate preprocessing techniques. This foundational step ensures that the subsequent machine learning models are built on a clean dataset, enhancing their accuracy and reliability.

D. Clustering Modelling

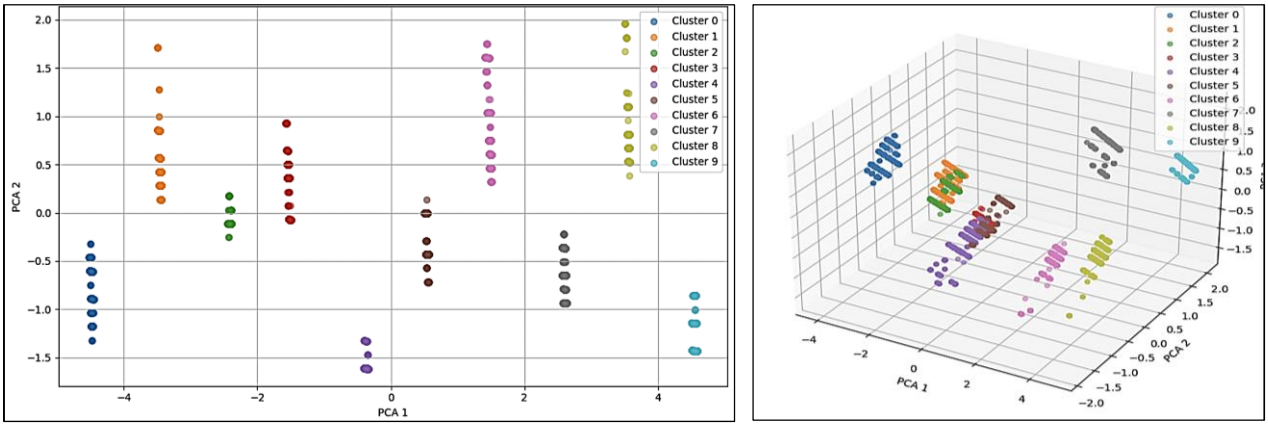
The first step involves clustering similar types of disasters based on their frequency. Each clustering method, including K-Means Clustering, DBSCAN, Self-Organizing Maps (SOM), and Gaussian Mixture Model (GMM), employs distinct algorithms and criteria for grouping data points. K-Means partitions data into K clusters by minimizing the within-cluster sum of squares within the cluster. DBSCAN identifies dense regions of points separated by sparser areas, while SOM organizes data onto a low-dimensional grid based on similarity. GMM models data distribution using a mixture of Gaussian distributions.

After clustering, it is crucial to analyze the relationship between the type of disaster and its frequency. Unsupervised learning techniques provide a means to explore this relationship without labelled data. By examining the distribution of disaster types within each cluster and the corresponding frequencies, we can gain insight into patterns and correlations. For example, certain clusters may predominantly contain hurricanes or floods with high frequencies, while others may include less frequent events such as earthquakes or biological disasters. Understanding these relationships can inform disaster preparedness and response strategies tailored to specific risk profiles.

To assess the effectiveness of each clustering method in accurately grouping similar disaster types, various evaluation metrics can be used. For example, the silhouette score, Davies-Bouldin index, or Calinski-Harabasz index can gauge the compactness and separation of clusters generated by K-Means and GMM. DBSCAN's performance can be evaluated on the number and coherence of resulting clusters. Quality can be assessed by quantization error and topographic error. By comparing these metrics across different clustering methods, we can identify the most suitable approach for our dataset and analysis objectives. By employing these unsupervised learning techniques and evaluation methods, we can gain valuable insights into the relationships between disaster types and their frequency, facilitating more informed decision making in disaster management and response planning.

E. K-Means Clustering

In K-means clustering, the silhouette_score method was used to calculate the suitable number of clusters used for the modelling, as shown in Fig. 21 (Principle Component Analysis (PCA)). PCA1 represents incident_type while PCA2 represents area. Each cluster represents the frequency of occurrence of each state and each type of incident. A total number of 10 clusters are displayed in two and three dimensional visuals. Furthermore, the means for all groups are calculated for evaluation to decide which model is better (Fig. 22).



(a) 2D K-means clustering (PCA visualization). (b) 3D K-means clustering (PCA visualization)

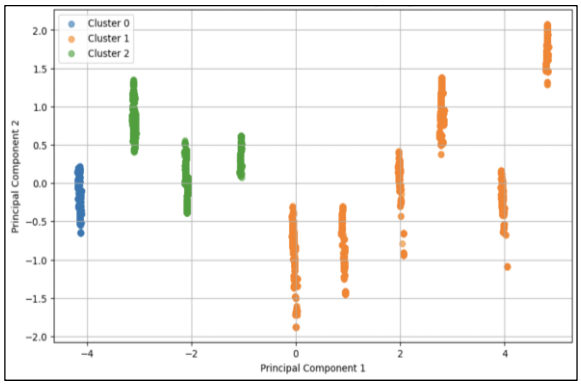
Fig. 21. K-means clustering with 10 clusters displayed in 2D and 3D.

Cluster	Mean state_label_encoded	Mean incident_type_label_encoded
Cluster 0	-0.8305234410559853	-0.545832628909552
Cluster 1	0.656128536563319	-0.5749958032566728
Cluster 2	-0.1465296393064368	-0.2076690402671876
Cluster 3	0.6420863858363859	0.004166666666666665
Cluster 4	-0.3867889267170983	0.6300834477659237
Cluster 5	-0.9346339314693742	0.2963489193145612
Cluster 6	0.7832611910932602	0.6056245747549327
Cluster 7	-0.4202422868055141	-1.4516790424470798
Cluster 8	0.24339020905853237	0.596222114270767
Cluster 9	0.5864160742411988	-1.5360898094711046

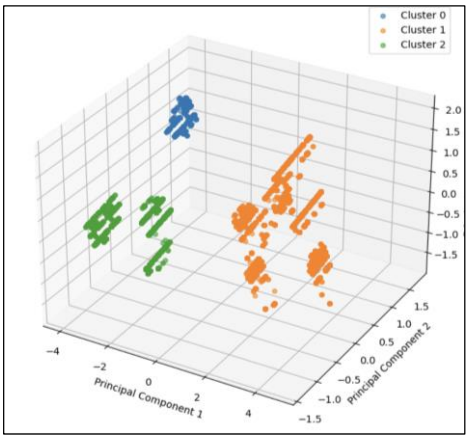
Fig. 22. Mean of clusters.

F. Gaussian Mixture Clustering

For Gaussian mixture clusters, 10 clusters are displayed in 2D and 3D images as the silhouette_score method is used. Fig. 23 displayed the mean; as a result, the mean is relatively lower than the K-means for the state and similar for incident type (Fig. 24).



a. Clustering of the Gaussian mixture model using PCA.



b. Clustering of Gaussian mixture models using PCA (3D visualisation).

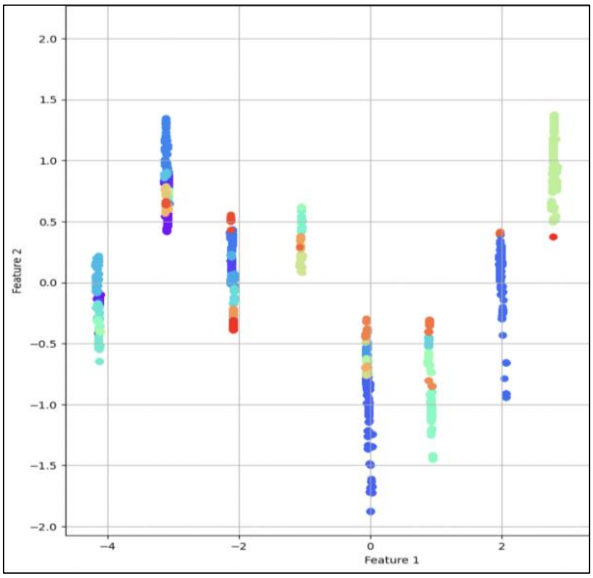
Fig. 23. Clustering of Gaussian mixture models.

Cluster	Mean state_label_encoded	Mean incident_type_label_encoded
Cluster 0	-0.8305234410559853	-0.545832628909552
Cluster 1	0.7832611910932602	0.6056245747549327
Cluster 2	-0.1465296393064368	-0.2076690402671876
Cluster 3	-0.3867889267170983	0.6300834477659237
Cluster 4	0.24339020905853237	0.596222114270767
Cluster 5	0.656128536563319	-0.5749958032566728
Cluster 6	-0.9346339314693742	0.2963489193145612
Cluster 7	0.5864160742411988	-1.5360898094711046
Cluster 8	0.6420863858363859	0.004166666666666665
Cluster 9	-0.4202422868055141	-1.4516790424470798

Fig. 24. Mean value of the clustering of the Gaussian mixture model.

G. Self-Organizing Maps (SOM)

In self-organizing map (SOM) clustering, the topology-preserving characteristics of SOM are used to organize the data into a grid of nodes, where each node represents a cluster. Unlike K-means, where cluster centers are calculated iteratively, SOM assigns data points to the nearest node in the grid, creating a topological map of the data space. The SOM clusters are visualized in 2D and mean value for each cluster is calculated to gain insight into the data distribution and cluster formations (Fig. 25).



a. SOM visualisation clusters.

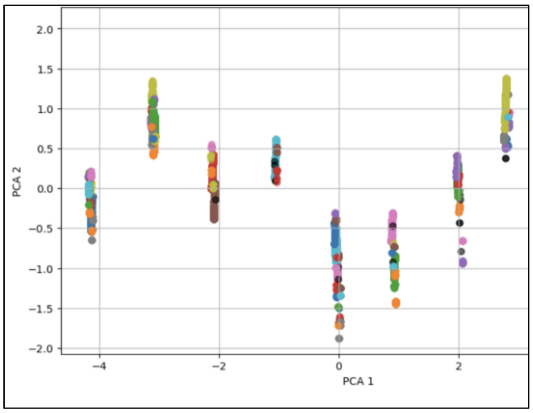
Cluster	Mean state	Mean incident type
Cluster (9, 9)	-4.11887	-0.115643
Cluster (9, 2)	-3.09063	0.512774
Cluster (5, 8)	-4.13412	-0.415435
Cluster (7, 4)	-3.07683	0.846698
Cluster (0, 2)	-4.1213	-0.17087
Cluster (3, 9)	-2.11355	0.170233
Cluster (3, 2)	3.97324	-0.446005
Cluster (2, 4)	-0.0422648	-0.948891
Cluster (0, 8)	1.98033	0.11528
Cluster (1, 6)	-2.10474	0.338972
Cluster (5, 2)	-3.105	1.01336
Cluster (4, 3)	-3.0814	0.659245
Cluster (3, 5)	-0.0497852	-0.522195
Cluster (4, 1)	-2.09281	0.00488725
Cluster (8, 2)	-2.1117	0.212243
Cluster (0, 6)	-4.15823	0.0289123
Cluster (8, 8)	-3.11657	0.894484
Cluster (4, 0)	-2.08924	-0.290384
Cluster (1, 8)	0.887093	-0.456394
Cluster (1, 7)	-2.07871	-0.0897168
Cluster (3, 7)	-4.14953	-0.206124
Cluster (4, 9)	-4.1362	-0.49732
Cluster (1, 5)	-3.08064	0.700448
...		
Cluster (3, 6)	3.93616	0.136698
Cluster (9, 0)	4.81073	1.77199
Cluster (2, 7)	-2.09231	-0.348949
Cluster (6, 2)	2.77579	0.376224

b. Mean value of SOM clusters

Fig. 25. Self-organising map visualisation and means value.

H. Density-Based Spatial Clustering of Noise Applications (DBSCAN)

For DBSCAN, 104 clusters are implemented as shown in Fig. 26 with their performance. Since there are too many clusters generated, the 3D visualization is too complicated to analyze due to overlapping clusters with unidentified clusters for each state and each incident_type.



a. DBSCAN clustering with PCA visualisation.

Cluster	Mean state_label_encoded	Mean incident_type_label_encoded
Cluster 0	-0.8305234410559853	-0.545832628909552
Cluster 1	0.7832611910932602	0.6056245747549327
Cluster 2	-0.1465296393064368	-0.2076690402671876
Cluster 3	-0.3867889267170983	0.6300834477659237
Cluster 4	0.24339020905853237	0.596222114270767
Cluster 5	0.656128536563319	-0.5749958032566728
Cluster 6	-0.9346339314693742	0.2963489193145612
Cluster 7	0.5864160742411988	-1.5360898094711046
Cluster 8	0.6420863858363859	0.004166666666666665
Cluster 9	-0.4202422868055141	-1.4516790424470798

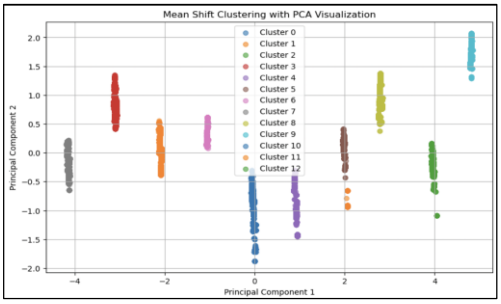
b. Means of DBSCAN clustering

Fig. 26. DBSCAN clustering and its performance (means).

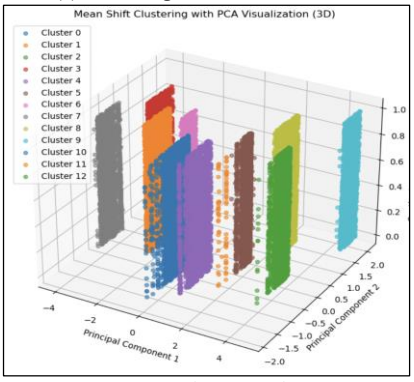
I. Mean Shift Clustering

As shown in Fig. 27, a total of 10 clusters are displayed and these clusters are separated compared to DBSCAN which

produces a clearer vision. The means of this model are on average negative values (Fig. 27). A cluster with a negative mean value indicates that the data points within that cluster have on average, negative values for the particular variable or feature analyzed (Fig. 28).



(a) Clustering of mean shifts in 2D.



(b) Clustering of mean shifts in 3D.

Fig. 27. Clustering of mean shifts with PCA visualisation in 2D and 3D.

	Mean state	Mean incident type
Cluster 0	-0.0513854	-0.0866459
Cluster 1	-2.09635	0.0206626	0.100683
Cluster 2	3.95537	-0.144531	-0.278133
Cluster 3	-3.09321	0.775598	-0.363575
Cluster 4	0.904547	-0.760637	0.609919
Cluster 5	1.97926	0.126576	-0.837415
Cluster 6	-1.05824	0.387688	-0.573405
Cluster 7	-4.13698	-0.217754	0.701689
Cluster 8	2.78928	0.99087	1.4467
Cluster 9	4.8101	1.76041	0.859684
Cluster 10	0.00743375	-1.48187	-0.521425
Cluster 11	2.06528	-0.775608	-1.59734
Cluster 12	4.04344	-1.00891	-1.08566

Fig. 28. Mean value of mean shift clustering.

Based on Table II, DBSCAN has the highest silhouette score. It is the best clustering algorithm, followed by the Gaussian mixture. This is because it not only displays the firm clusters in the diagram, but also has a relatively higher silhouette score compared to other clustering algorithms. However, they are imperfect when applied in the FEMA dataset. There are some inaccuracies in the result with the clustering algorithm due to outliers in the mean calculation for state and incident. To conclude, having a definite and firm clean dataset is crucial in the clustering process, as it will provide the most accurate and identical result for analysis.

TABLE II. COMPARISON OF FIVE CLUSTERING MODELS USING THE SILHOUETTE SCORE

Model	Silhouette Score
K-Means	0.4546
Gaussian Mixture Model	0.8161
Self-Organizing Maps	0.6003
DBSCAN	0.9883
Mean Shift Model	0.4715

V. CONCLUSION

In view of the information provided on natural disaster data and after running different clustering techniques, this paper concludes that modern machine learning algorithms such as K-means, DBSCAN, SOM, and GMM have been very efficient in classifying and understanding patterns in disaster incidences. K-means identifies trends/distributions of disaster incidents and thus improves preparedness and response strategies. Spatial distribution studies, such as those related to the location of disaster sources and eventual mitigation of their impacts, are based on attributes central to DBSCAN. SOMs are a robust method for topological mapping and clustering, which can be used effectively for the visualization and better interpretation of data. GMMs are efficient ways of modelling even very complex multimodal distributions and are therefore suitable for anomaly detection in disaster datasets.

However, several limitations with this study. The accuracy of clustering results greatly depends on the dataset quality and completeness of the data set. The existence of missing values and inconsistencies within the data will greatly decrease the reliability of the result. Moreover, computational complexity and time for some clustering algorithms are highly needed mainly with the large datasets, which becomes a problem in the application.

Future work has to be directed towards enhancing data preprocessing techniques so that it can answer missing and inconsistent data more efficiently. In addition, other data sources, such as real-time satellite imagery, sensor data, etc., will add comprehensiveness and accuracy to the analysis. Computational efficiency in clustering algorithms can be achieved by parallel processing or any other advanced computing technique, which requires further research. Moreover, the development of hybrid models involving multiple clustering techniques would tie the different strengths of each technique and therefore further enhance disaster predictions. There will be a need to continue at this higher level of collaboration with disaster management agencies to ensure that the findings of such studies translate to actionable strategies that reduce the impacts of natural disasters.

ACKNOWLEDGMENT

This research is sponsored by: Tunku Abdul Rahman University of Technology and Management, INTI International University.

REFERENCES

[1] V. Korhonen, "Natural disasters in the U.S. - Statistics & Facts," Statista. Accessed: Jul. 08, 2024. [Online]. Available: https://www.statista.com/topics/1714/natural-disasters/.

[2] V. Korhonen, "Number of natural disasters in the United States in 2022, by type," Statista. Accessed: Jul. 08, 2024. [Online]. Available: https://www.statista.com/statistics/216819/natural-disasters-in-the-united-states/.

[3] E. B. Salas, "Number of tornadoes in the United States from 1995 to 2022," Statista. Accessed: Jul. 08, 2024. [Online]. Available: https://www.statista.com/statistics/203682/number-of-tornadoes-in-the-us-since-1995/.

[4] V. Korhonen, "Number of fatalities due to natural disasters in the United States in 2022, by type," Statista. Accessed: Jul. 08, 2024. [Online]. Available: https://www.statista.com/statistics/216831/fatalities-due-to-natural-disasters-in-the-united-states/.

[5] P. Duraisamy and Y. Natarajan, "Twitter Disaster Prediction Using Different Deep Learning Models," SN Comput Sci, vol. 5, no. 1, p. 179, Jan. 2024, doi: 10.1007/s42979-023-02520-7.

[6] M. T. Majemite, A. Obaigbena, M. A. Dada, J. S. Oliha, and P. W. Biu, "Evaluating the role of big data in U.S. disaster mitigation and response: a geological and business perspective," Engineering Science & Technology Journal, vol. 5, no. 2, pp. 338–357, Feb. 2024, doi: 10.51594/estj.v5i2.764.

[7] T. Venkat Narayana Rao, P. Jakkam, and S. Medipally, "Future Trends and Innovations in Natural Disaster Detection Using AI and ML," 2024, pp. 110–134. doi: 10.4018/979-8-3693-2280-2.ch005.

[8] Y. Wei et al., "Comparative Analysis of Artificial Intelligence Methods for Streamflow Forecasting," IEEE Access, vol. 12, pp. 10865–10885, 2024, doi: 10.1109/ACCESS.2024.3351754.

[9] T. T. Tin, E. H. C. Sheng, L. S. Xian, L. P. Yee, and Y. S. Kit, "Machine learning classification of rainfall forecasts using Austin weather data," International Journal of Innovative Research and Scientific Studies, vol. 7, no. 2, pp. 727–741, Mar. 2024, doi: 10.53894/ijirss.v7i2.2881.

[10] Heads or Tails, "US Natural Disaster Declarations: County-level data from the Federal Emergency Management Agency: 1953 - today," U.S. Government Works. Accessed: Jul. 08, 2024. [Online]. Available: https://www.kaggle.com/datasets/headsortails/us-natural-disaster-declarations.

[11] S. C. Ipiankama, "Customer Segmentation Using K-Means Clustering." Accessed: Jul. 08, 2024. [Online]. Available: https://sampsonipiankama.medium.com/customer-segmentation-using-k-means-clustering-ae73e3d82934.

[12] S. Hu, Z. Xiao, Q. Rao, and R. Liao, "An anomaly detection model of user behavior based on similarity clustering," in 2018 IEEE 4th Information Technology and Mechatronics Engineering Conference (ITOEC), IEEE, Dec. 2018, pp. 835–838. doi: 10.1109/ITOEC.2018.8740748.

[13] B. Tan, "Customer Segmentation with k-Means Clustering," LinkedIn. Accessed: Jul. 08, 2024. [Online]. Available: https://www.linkedin.com/pulse/customer-segmentation-k-means-clustering-bryan-tan/.

[14] S. Chakraborty and N. K. Nagwani, "Weather Forecasting using Incremental K-means Clustering," Computers and Society, 2014.

[15] K. Wang, X. Qi, H. Liu, and J. Song, "Deep belief network based k-means cluster approach for short-term wind power forecasting," Energy, vol. 165, pp. 840–852, Dec. 2018, doi: 10.1016/j.energy.2018.09.118.

[16] R. Panchotia, "Clustering Geo-location : DBSCAN," Analytics Vidhya. Accessed: Jul. 08, 2024. [Online]. Available: https://medium.com/analytics-vidhya/clustering-geo-location-dbscan-cadb33b0442e.

[17] T. Hshan, "Demonstrating Customers Segmentation with DBSCAN Clustering Using Python," Medium. Accessed: Jul. 08, 2024. [Online]. Available: https://hshan0103.medium.com/demonstrating-customers-segmentation-with-dbscan-clustering-using-python-8a2ba0db2a2e.

[18] R. Dey and S. Chakraborty, "Convex-hull & DBSCAN clustering to predict future weather," in 2015 International Conference and Workshop on Computing and Communication (IEMCON), IEEE, Oct. 2015, pp. 1–8. doi: 10.1109/IEMCON.2015.7344438.

[19] GeoSense, "Self-Organizing Maps for Sentinel 2 Image Segmentation using Python," Medium. Accessed: Jul. 08, 2024. [Online]. Available: https://geosen.medium.com/self-organizing-maps-for-sentinel-2-image-segmentation-using-python-b42cefcb32e9.

[20] Kaushik, "Self-Organizing Map (Customer Segmentation in Banking)," Analytics Vidhya. Accessed: Jul. 08, 2024. [Online]. Available: https://medium.com/analytics-vidhya/self-organizing-map-customer-segmentation-in-banking-9d7ce96bd3ec.

[21] P. Mohan and K. Patil, "Deep Learning Based Weighted SOM to Forecast Weather and Crop Prediction for Agriculture Application," International Journal of Intelligent Engineering and Systems, vol. 11, no. 4, pp. 167–176, Aug. 2018, doi: 10.22266/ijies2018.0831.17.

[22] Amy, "Gaussian Mixture Model (GMM) for Anomaly Detection," GrabNGoInfo. Accessed: Jul. 08, 2024. [Online]. Available: https://medium.com/grabngoinfo/gaussian-mixture-model-gmm-for-anomaly-detection-e8360e6f4009.

[23] C. O'Sullivan, "Identifying Restaurant Hotspots with a Gaussian Mixture Model," Towards Data Science. Accessed: Jul. 08, 2024. [Online]. Available: https://towardsdatascience.com/identifying-restaurant-hotspots-with-a-gaussian-mixture-model-2a840ab0c782.

[24] G. Jouan, A. Cuzol, V. Monbet, and G. Monnier, "Gaussian mixture models for clustering and calibration of ensemble weather forecasts," Discrete and Continuous Dynamical Systems - S, vol. 16, no. 2, pp. 309–328, 2023, doi: 10.3934/dcdss.2022037.