Industrial Engineering and Operations Research
COLUMBIA | ENGINEERING

European
Investment Bank

THE CENTER FOR ARTIFICIAL INTELLIGENCE
IN BUSINESS ANALYTICS & FINANCIAL TECHNOLOGY

IEOR 4721 AI Applications in Quant Finance

Generating Financial Losses From Extreme Events via GANs

Author
Wenhan Yang[†]
Yueru He[†]
Noah Dawang[†]

Supervisor
Dr. Ali Hirsa[‡]
Miao Wang[‡]
Giuseppe Bonavolontà[*]
Oleg Reichmann[*]

Summer 2023

Contents

1 Abstract

Analyzing financial losses stemming from extreme events plays a pivotal role in effective risk management. However, this task is fraught with challenges, including limited data availability, non-stationarity, and the distinctive distributions characterizing loss data, rendering accurate predictions a formidable endeavor. Building upon our ongoing investigation into the generation of high-quality synthetic data using Generative Adversarial Network (GAN) models, this research embarks on a quest to expand and diversify the dataset. To confront the inherent data limitations, we delve into the utilization of three distinct GAN types, each tailored to address specific nuances in the loss data. These GAN models are thoughtfully combined with a repertoire of diverse clustering techniques. Our goal is twofold: to augment the dataset size for GAN training and, more importantly, to imbue the generated data with an enhanced representation of extreme events and the associated long-tail distributions.

By extending the boundaries of data generation and employing cutting-edge techniques, this research seeks to provide a robust foundation for improved risk assessment, aiding decision-makers in navigating the complexities of extreme event forecasting and risk management.

2 Background

In recent years, the advancement of machine learning techniques has revolutionized various domains, including data synthesis and augmentation. One prominent technique that has garnered substantial attention is Generative Adversarial Networks (GANs). GANs are a class of deep learning models that excel in generating realistic data samples by training a generator network to produce data that is indistinguishable from authentic data, as assessed by a discriminator network. This adversarial interplay between the generator and discriminator results in the refinement of the generator's output over time, leading to the creation of high-quality synthetic data.

The application of GANs to generate extreme event data, particularly in the context of losses, presents an innovative approach to addressing data scarcity and the challenges associated with extreme events. Extreme events, such as catastrophic natural disasters, are by their nature infrequent and often accompanied by significant financial and societal ramifications. Consequently, the available real-world data of such events is inherently limited, which poses substantial limitations on the development and validation of models aimed at analyzing, predicting, and mitigating the impact of these events.

GANs offer a potent solution to this quandary. By employing GANs, researchers and practitioners can effectively expand the available dataset of extreme event occurrences. The generator network within the GAN architecture learns the underlying distribution of the authentic data, enabling it to produce synthetic data instances that closely resemble actual extreme events. The discriminator, trained alongside the generator, provides continuous feedback to enhance the authenticity of the generated data. Through this iterative process, GANs progressively improve their ability to generate data that mirrors the statistical characteristics and patterns of real extreme event occurrences.

[†] The Fu Foundation School of Engineering and Applied Science, Columbia University
[‡] Industrial Engineering & Operations Research Department, Columbia University
[*] European Investment Bank

3 Introduction

This project is a continuation of the previous two teams' work. Our purpose is to develop a GAN model that can output stable synthetic loss data that retains the unique feature of extreme events. Extreme events' losses, unlike normal events, rarely occur, and therefore it is hard to capture all the characteristics of patterns of such events. It has very complex dependencies on multiple factors, capturing these intricate relationships and accurately replicating them in a synthetic dataset requires a deep understanding of the underlying mechanisms driving these events.

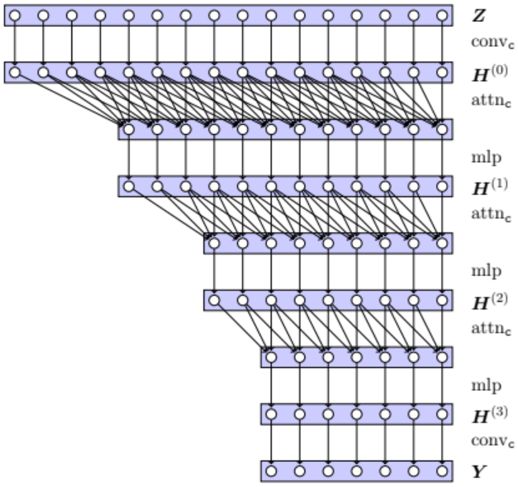
From a modeling perspective, the data for extreme events is very sparse, which adds challenges to modeling, especially if we want to model with time series, which we will explain in detail in the next sections. Besides, extreme event distributions are characterized by long tails, which means that the data points in the tails of the distribution deviate significantly from the majority of data points. Capturing this tail behavior accurately in a synthetic dataset demands a sophisticated approach that goes beyond traditional data generation techniques. Lastly, extreme events might be non-stationary, meaning that the statistical properties of the data change over time. Successfully capturing these variations and incorporating them into synthetic data requires a model that can adapt to changing conditions and trends.

Our work is based on the first team's efforts in cleaning and combining the loss dataset for natural disasters from the EM-DAT Public database, which is devoted to disaster data recording, and the second team's effort on GDP normalization, and clustering efforts of countries and regions based on climate change. Our progress this summer focuses on testing new GAN structures (TTGAN, TAGAN, and FlowGAN). We encountered sparsity, fat-tail, and non-stationary issues and tried to tackle them individually. We also devote a large portion of time to clustering countries and regions with similar characteristics in order to increase training data size.

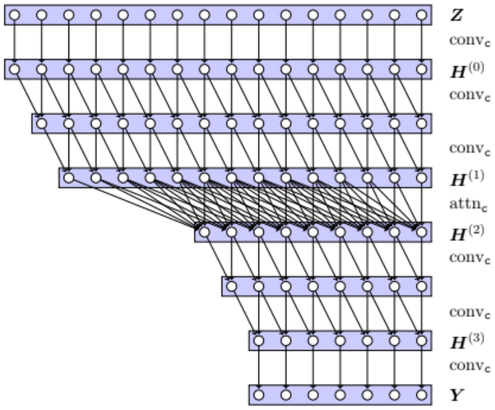
4 Key Definitions

- **Generative Adversarial Networks**: Generative Adversarial Networks, otherwise known as GANs, are machine learning models that are trained to replicate the data they are trained on. Involved are two different neural networks: a generator and a discriminator. The generator is trained to create samples that emulate the training data, while the discriminator is trained to correctly classify incoming data as fake (made by the generator), or real. These two neural networks are then trained together, in a competitive/adversarial manner, with the generator making samples in an attempt to fool the discriminator, while the discriminator works to correctly label generator samples as fake. As such, the hope is that the generator comes to make samples with quality replications of the training data. The applications of GANs include using them to generate synthetic music, art, images, and text, and they have seen much success in generating synthetic data that is difficult to distinguish from real data.
- **Extreme/Realistic Samples**: Extreme samples are samples that stray far away from the bulk of the rest of their distribution. In the context of this project, an extreme climate sample would be a significantly extreme weather event, such as a hurricane, tsunami, or earthquake. The issue with understanding such data is that there is not much of it, making it hard to train or classify. Realistic samples are samples that well reflect what is seen in the real world. This is the kind of data that a GAN, for example, would hope to emulate.

- **TTGAN**: TTGAN, short for Temporal Transformer GAN, is a GAN designed for time series data, and employs transformer architecture for the generator and discriminator. The Architecture is as follows:



- **TAGAN**: TAGAN, short for Temporal Attention GAN, is a GAN designed for time series data, and employs the attention mechanism and temporal convolution methods in its neural network architectures. The Architecture is as follows:



- **DCGAN**: DCGAN, short for Deep Convolutional GAN) is a GAN architecture that was created for generating images. It uses deep

- **FlowGAN**: A special type of GAN where the activation functions in the generator are restricted to (piecewise) differentiable invertible functions, with (piecewise) differentiable inverse. For example, an allowed activation function is LeakyReLU given by $max(x, 0.01x)$ while the regular ReLU given by $max(x, 0)$ is not allowed. With these types of activation functions, a maximum likelihood loss term can be added to the standard GAN loss.

- **EM-DAT**: The Emergency Events Database is maintained by the Centre for Research on the Epidemiology of Disasters at the University of Louvain in Belgium, and holds information on disasters, both natural and human-induced, that have occurred since 1900. EM-DAT is widely used by government and research institutions to understand disaster trends, as well as to find ways to reduce risk. The database includes a myriad of information on location, type of disaster, impact/loss, etc.

- **WEO**: The World Economic Outlook is a publication by the International Monetary Fund that presents developments and findings on economic development twice a year. The WEO provided a global outlook on a wide range of topics, including economic growth, inflation, exchange rates, and GDP.
- **GPD**: The Generalized Pareto Distribution (GDP) is a family of continuous probability distributions that is often used to model the tails of distributions, and is based in extreme value theory. More information on this distribution is presented later in this report.
- **K-Previous Means**: A K-means clustering method that utilizes the centroids from previous results and uses them as initial centroids, so that the centroids won't be randomly initialized.
- **K-Means++**: An enhancement of the K-Means clustering algorithm that improves the initialization of cluster centroids. It selects initial centroids probabilistically, favoring data points farther from existing centroids. This approach leads to more consistent and accurate clustering results by reducing sensitivity to initialization, making K-Means++ a more robust and reliable clustering algorithm.
- **Dynamic Time Warping**: Dynamic Time Warping (DTW) is a distance measure used to compare sequences of data that may have variations in their timing or speed. It is particularly useful when dealing with time series data or sequences that might be similar but have different lengths. DTW takes into account both the similarities in shape and the temporal distortions that might exist between two sequences. In our project, DTW is used in DBSCAN, K-previous means, and hierarchical clustering for the time series data of countries.
- **DBSCAN**: DBSCAN is a sort of clustering method called Density-Based Spatial Clustering of Applications with Noise. It is used to discover clusters in data that might have irregular shapes and varying densities. Unlike traditional clustering algorithms like k-means, DBSCAN doesn't require the user to specify the number of clusters beforehand. It's particularly effective when dealing with noisy data and clusters of different sizes and shapes.

5 Previous Work

5.1 Work From First Team

The first team focused on preparing and enhancing a dataset containing critical information about financial losses incurred due to extreme climate events and the GDP figures of the countries impacted. The initial dataset was compiled through data collection from two primary sources: EM-DAT and WEO. However, to ensure the comprehensiveness and accuracy of the GDP data, they supplemented the WEO figures with data from the World Bank's Economic Indicators dataset. Despite these efforts, gaps in GDP information remained, prompting the team to resort to interpolation methods when feasible and omitting data entries for which GDP information could not be adequately filled. A crucial outcome of this data preprocessing was the introduction of a "Normalized Losses" column, a calculated metric that normalized disaster losses by the GDP of the respective countries.

As the next step, the first team recognized that varying data entry counts across countries, combined with the voracious data requirements of Generative Adversarial Networks (GANs), called for a more strategic approach to data processing. To this end, the researchers adopted a clustering methodology centered around geographical locations. Rather than focusing on individual country-level data, the team categorized countries into distinct climate zones. This classification strategy drew inspiration from a 2006 paper authored by Tsonis et al.[7], which highlighted the interconnectedness of countries with similar climates. Notably, tropical regions exhibited a particularly strong degree of correlation. As part of this approach, the dataset was segregated into four primary climate zones: Tropical, Subtropical, Temperate, and Polar. These zones were characterized by specific latitude ranges, ensuring a more regionally relevant clustering process.

Another effort involved the application of normalization techniques tailored to extreme event data. They used Extreme Value Theory (EVT) to achieve this. Specifically, they leveraged EVT to define a threshold beyond which losses could be considered exceptional. This threshold was denoted as miu_i and was integral to the normalization process. Utilizing a Generalized Pareto Distribution, the first team modeled extreme value distributions for countries within the same climate band. The parameter u_i was introduced to allow manual configuration of the threshold, indicating the point beyond which losses would be deemed significant and retained for analysis. For losses below this threshold, the data would be considered non-exceptional and excluded from further consideration. The default value assigned to u_i was 0.3.

Lastly, they implement the pipeline using Deep Convolutional Generative Adversarial Networks (DCGANs) for data generation and manipulation. The pipeline consists of multiple steps designed to enhance the data, shift its distribution, and employ a Conditional GAN for refined data generation, as outlined in the resources.

To begin with, the dataset undergoes normalization using an empirical quantile technique, aligning it with a generalized Pareto distribution. This normalized data then serves as the foundation for training a DCGAN. Following the DCGAN training phase, the first team applies parameter shifting. This involves iteratively selecting data points that exceed a specified quantile threshold (denoted as c) and retaining them. The removed data points are then replaced with data generated by the trained GAN. This iterative process is repeated for a set number of iterations (k). The end goal of this distribution shifting is to condition the results to exceed a particular quantile, namely the 1 - c^k th quantile.

The next step involves the integration of a Conditional GAN into the pipeline. The generated data, which has undergone distribution shifting, is fitted to a generalized Pareto distribution. Samples derived from this distribution are utilized as supplementary input during the training of the Conditional GAN. The user defines a desired level of extremeness (τ), which is then adjusted to $\tau' = \frac{\tau}{\tau c^k}$ to account for the previous distribution shifting. Within the Conditional GAN framework, values are generated and retained if they meet a specific extremeness probability criterion, symbolized as $e' \sim \text{GPD}(1 - \tau')$.

5.2 Work From Second Team

The team initiated their work with an exploratory data analysis, where they computed summary statistics and density curves for both regular and log losses within each cluster. They observed that clusters exhibited distinct characteristics in terms of skewness and kurtosis. For regular losses, each cluster displayed positive skewness and substantial excess kurtosis, indicating the presence of extreme tails. In contrast, log losses exhibited negative skewness, with kurtosis varying between clusters. The team attempted curve fitting for log losses and found interesting models, particularly for certain clusters. However, fitting absolute losses proved challenging due to kurtosis issues.

Building on the previous team's strategy, the second team used data normalization to amplify extremeness for GANs. This process involved quantile-based filtering and scaling to align with the generalized Pareto distribution, aiming to improve data suitability for GAN-based generation. As progressed, the team scrutinized changes in the pipeline results. They found that the new clustering approach didn't lead to significant alterations in the final pipeline outcomes. This indicates that the clustering technique wasn't the root cause of the challenges.

In the GAN validation part, the team used Generalized Pareto Distribution (GPD) and justified this choice with the Pickands-Balkema-De Haan theorem, finding the convergence of excess random variables to a GPD under certain conditions. However, due to data sparsity and rarity in climate loss data, they acknowledged the limitations of directly emulating tail distributions using GPD. Consequently, the GANs were utilized for initial distribution shifting to enhance extremeness, while a conditional GAN was employed for fine-tuning.

To evaluate the GANs' performance, the team established three criteria: internal consistency, inter-GAN consistency, and GPD approximation. They employed the Kolmogorov-Smirnov statistic as an error measure across various comparisons, aiming to ensure the quality and reliability of GAN outputs. Next, the team delved into hyperparameter tuning to address the narrow range of losses generated by the conditional GAN. They meticulously explored a five-dimensional hyperparameter space, employing line search techniques to optimize learning rates and noise parameters. Despite exhaustive efforts, the team encountered challenges in obtaining satisfactory hyperparameters, prompting them to consider alternative approaches.

To improve the performance of the conditional GAN and achieve a desired Generalized Pareto Distribution (GPD) shape and range, the team explored internal data transformations. They turned to the Box-Cox family of transformations, aiming to stabilize variance and align the data more closely with the desired GPD characteristics. The Box-Cox transformation can potentially guide the GAN towards an appropriate local minimum, representing extreme climate loss data with the desired distribution.

The team conducted consistency testing to evaluate the efficacy of the Box-Cox transformation. Utilizing parameters such as tau (0.00365) and λ within a defined range [0, 1], they observed that certain results exhibited qualities resembling the GPD shape and adequately covering a more reasonable range for extreme samples. This alignment with the GPD shape was confirmed through Kolmogorov-Smirnov (K-S) tests comparing fitted GPDs to GAN-generated samples. However, consistency was not uniformly achieved, and while some improvements were notable, replicating these results consistently is challenging.

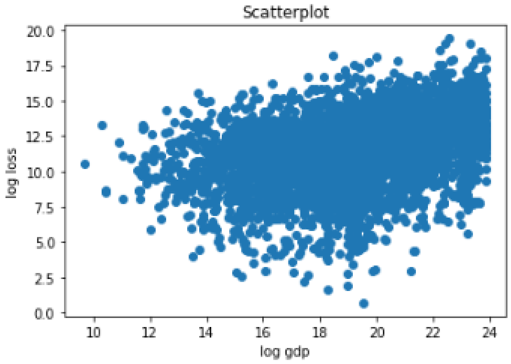
In pursuit of improved clustering and data handling, the team introduced a novel clustering method. The new clustering approach aimed to group regions based on historical loss severity within the Intergovernmental Panel on Climate Change (IPCC) regions. By aggregating data based on climate regions and clustering regions by their impact/losses, the team sought to create a more refined and localized dataset for GAN training.

6 Data

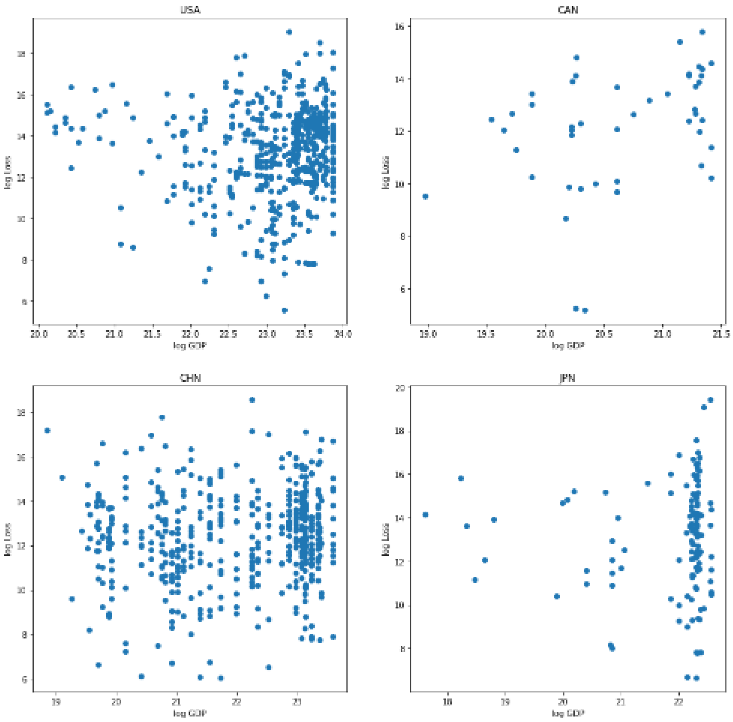
The dataset used was the Emergency Events Database (EM-DAT) from Center for Research on the Epidemiology of Disasters[2], a database containing records of natural disaster events and monetary losses since 1900. This is an amalgamated dataset sourced from a combination of public data such as the UN, NGOs, and the press plus private data from insurance companies. This is the origin point of all other natural disaster related datasets.

The work of the previous team involved preprocessing ISO code data, including merging defunct countries with their successors, creating dictionaries to map countries to regions, and handling provinces in countries spanning multiple regions. A regex search assigned provinces based on textual details, with manual consideration for roughly 100 losses. Region imputation methods included cross-referencing city information, non-English references, and online research. The ISO code data was then made aligned with region area and GDP dataset for future usage.

Most of the exploratory data analysis was done by the previous two groups, however we decided to take a closer look at the effect of GDP on losses. Interestingly, on an accumulated scale, there is a significant correlation between losses and GDP:



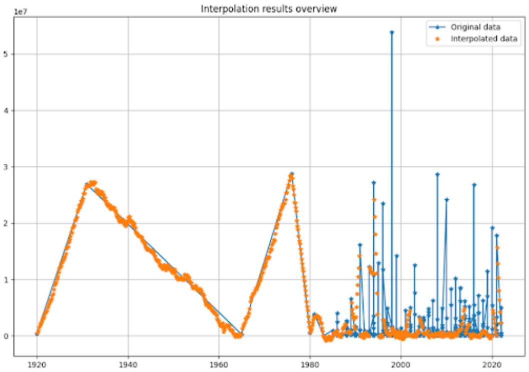
However, within a country this relationship seems to go away:



There are a few interpretations: either the correlation on the accumulated basis is spurious and can be ignored, or, there is only a dependence between "levels" of GDP and losses. These are good questions for the next team's exploratory data analysis.

7 Brownian Bridge Imputation For Losses

Following the last team's suggestion, we started exploring the possibility of generating data with TTGAN and TAGAN. Both models require input to be time series data. The data for most countries, however, does not have a record for every year. This is the sparsity issue mentioned in the introduction part. As a first try, we used Brownian Bridge to impute the data from China. Below is the imputation plot, We immediately spot the sparsity of data before 1980. There is too much missing data before 1980 so we decided to focus on the imputation of data after 1980.



The Brownian motion refers to a conditional distribution that describes the behavior of a Brownian motion path between two fixed points, given its values at those points. Mathematically, if B(t) is a Brownian motion process with B(0) = a and B(T) = b, then the Brownian Bridge process $B_{a,b}(t)$ is defined as $B_{a,b}(t) = B(t) - \frac{t}{T}B(T) - \frac{T-t}{T}B(0)$. T represents the total time interval from 1920 to 2022, and t is a specific point within that interval. B(t) denotes the Brownian motion process at time t. The Brownian Bridge process $B_{a,b}(t)$ ensures that the path starts at a and ends at a band and maintains the same statistical properties as a Brownian motion path. We cannot ask the Brownian Bridge interpolation package to impute the losses per year, because the gap between the two records is different. A workaround here is to only save data for integer years and predictions within one month of the year start/end, and drop the imputation otherwise.

As the next step, we use the imputed loss as a time series to feed to TTGAN and TAGAN.

8 TTGAN,TAGAN On Single Country Time Series

The concepts and initial code stem from paper titled *Simulating financial time series using attention* by Fu et al.[4] Given the constraints of merging loss time series from different countries, including a country's previous year's losses with the subsequent country's following year's losses, our approach began by focusing on utilizing single-country time series data. To illustrate this approach, we chose China as a sample due to its abundance of records. The process involved aggregating the loss data, computing the sum of losses per year, and experimenting with different imputation techniques. Initially, we employed a basic Brownian Bridge imputation, followed by exploring the use of FlowGAN imputation for loss data.

The generator of the Temporal Attention GAN (TAGAN) consists of convolutional layers, supplemented by an attention layer strategically positioned amidst these layers to enhance the receptive field. This design empowers the network to capture local features effectively.

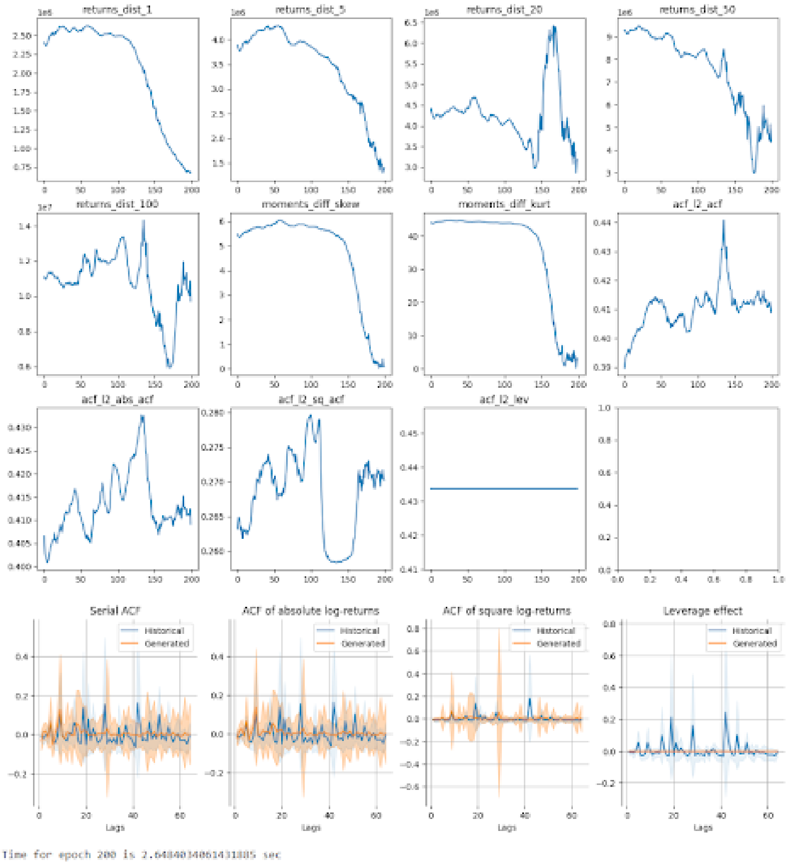
On the other hand, the discriminator adheres to a conventional convolutional neural network structure. TAGAN's suitability for time series data is rooted in its generator's utilization of causal layers, which enables the integration of information from the past up to the present. Meanwhile, the discriminator operates with standard layers, incorporating all available information.

In a similar vein, the Temporal Transformer GAN (TTGAN) employs a generator built exclusively with causal layers and a discriminator designed with regular layers. The distinction between TTGAN and TAGAN lies in the utilization of transformers within both the generator and discriminator of TTGAN. The transformer architecture comprises a series of convolutional layers followed by blocks housing two-layer MLPs, each featuring an input layer and an output layer, in addition to an attention layer. This arrangement facilitates the generation and assessment of temporal sequences, making TTGAN a suitable choice for time series data analysis and synthesis.

TAGAN With Brownian Bridge

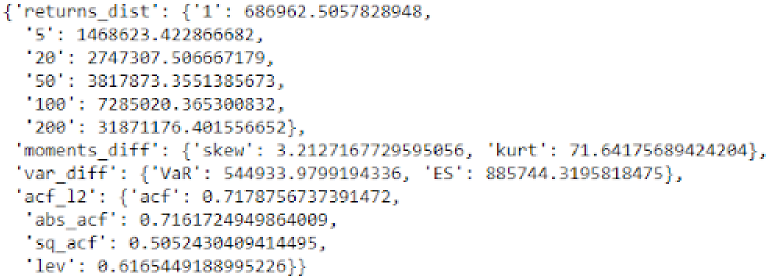
The outcomes of the experiment do not yield promising results. GANs typically require a substantial amount of data to perform well, and our time series, spanning 43 years (China data from 1980 to 2022), is inadequate to achieve the desired outcomes. Here, we present the results obtained after training and testing the model for 200 epochs.

Training Results



```
Time for epoch 200 is 2.6484034061431885 sec
```

Testing Results



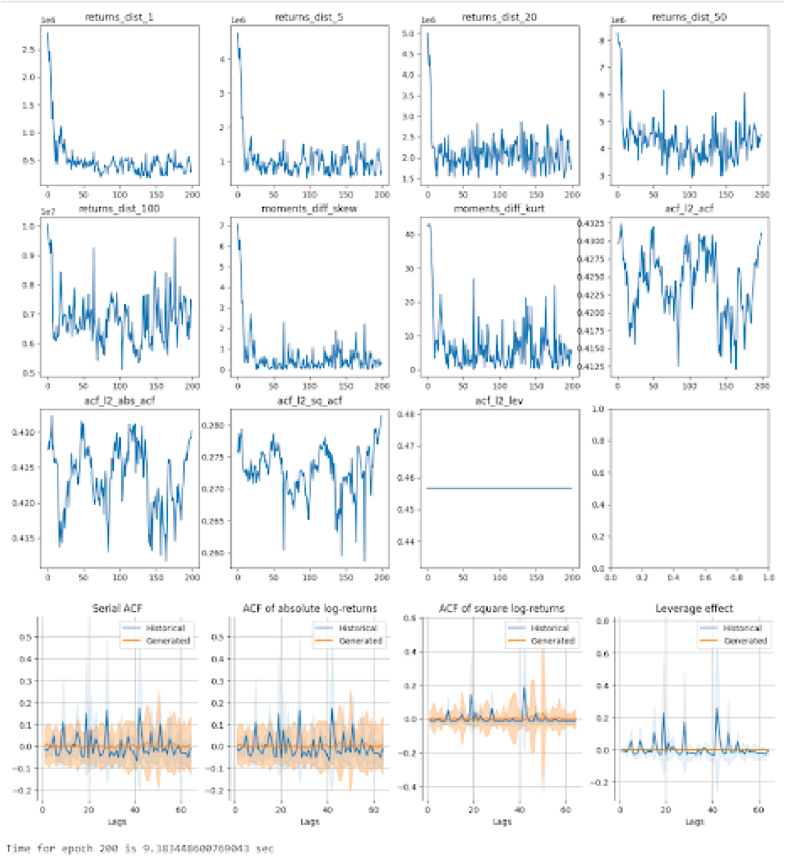
```
{'returns_dist': {'1': 686962.5057828948,
  '5': 1468623.422866682,
  '20': 2747307.506667179,
  '50': 3817873.3551385673,
  '100': 7285020.365300832,
  '200': 31871176.401556652},
 'moments_diff': {'skew': 3.2127167729595056, 'kurt': 71.64175689424204},
 'var_diff': {'VaR': 544933.9799194336, 'ES': 885744.3195818475},
 'acf_l2': {'acf': 0.7178756737391472,
  'abs_acf': 0.7161724949864009,
  'sq_acf': 0.5052430409414495,
  'lev': 0.6165449188995226}}
```

The Wasserstein distance metric, which quantifies the dissimilarity between the generated and real data distributions, has values at the hundred-thousand level. Additionally, we have observed that the range of generated data is limited, and only in a narrow spectrum of losses. Remarkably large losses appear to be treated as anomalies by the model. This phenomenon could potentially be attributed to the application of Brownian Bridge Imputation. This imputation technique aims to smoothly fill the gaps in the data series, leading the model to emphasize the gradual transition between years rather than focusing on extreme events.

TTGAN With Brownian Bridge

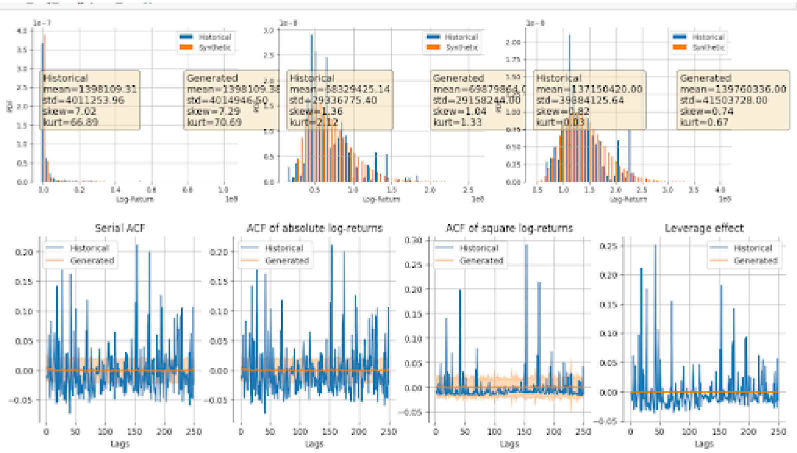
We fed the same data to TTGAN, and the results were actually better than TAGAN. Below are the 200 epoch training and testing results.

Training Results



```
Time for epoch 200 is 9.383448600769043 sec
```

Testing Results



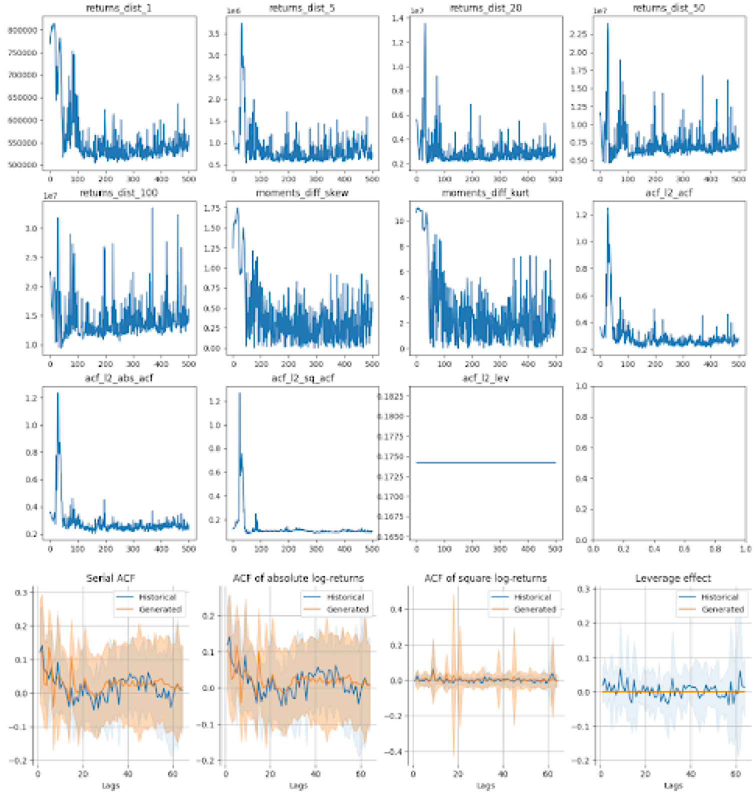
We observe a lower Wasserstein distance and in the test results we observe that the distributions of generated data and historical data are pretty aligned with each other, but the extreme values are not captured at all. The takeaway is similar to that of TAGAN but we have an initial expectation that TTGAN performs better than TAGAN. Due to the limitation of the Brownian Bridge, we switched our focus to another imputation method called FlowGAN.

TAGAN With FlowGAN Imputation

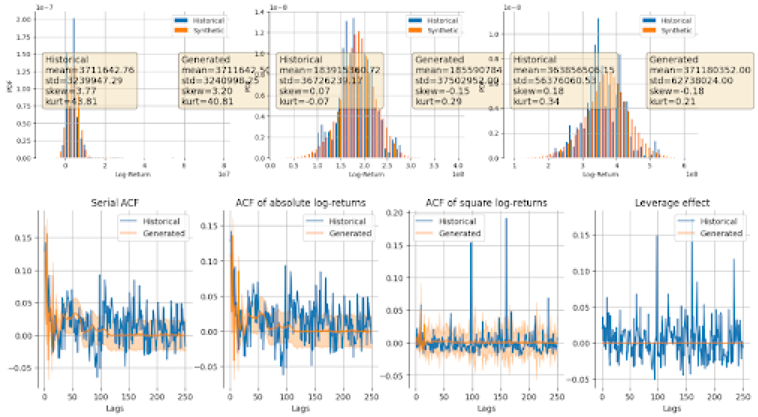
FlowGAN is another GAN model. We explain its structures in another section of this report. But its primary use here is to impute the loss values. We split the data into two parts based on whether the loss is greater than a certain quantile. The reason for doing this is that FlowGAN performs well with data below certain quantiles, but poorly for values higher than that quantile. Therefore we apply FlowGAN for part of the data, and for another part of the data we use parametric methods, and the combination of two methods works better than applying FlowGAN to all data.

Here we used the imputed data to run TAGAN. We trained the model with more epochs and did some hyperparameter tuning. Below are the 500 epochs of training and testing results.

Training Results



Testing Results

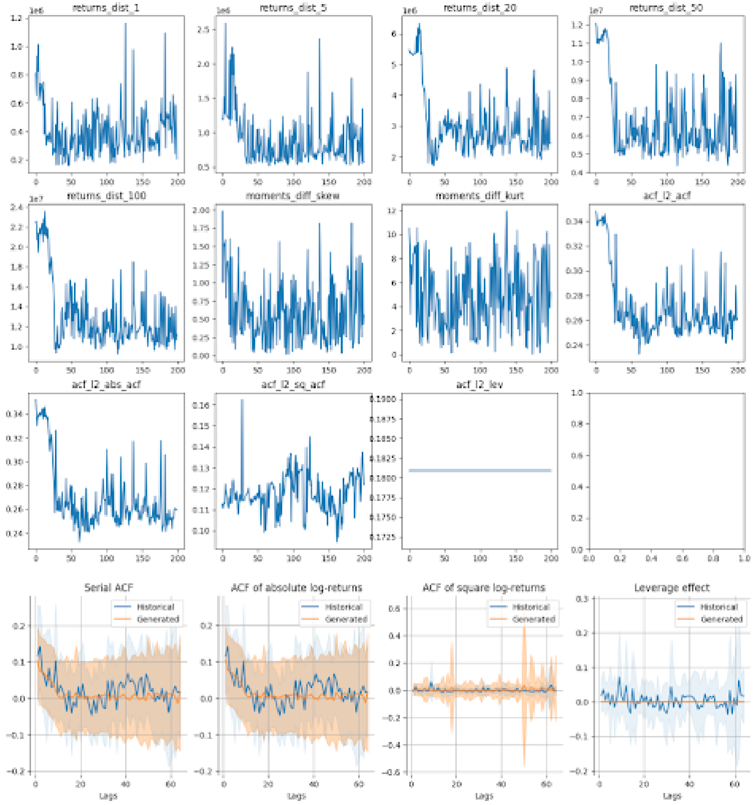


Even with FlowGAN imputation, the narrow range output issue persists. The distribution of generated data aligns better with the historical distribution, but the Wasserstein distance does not improve a lot.

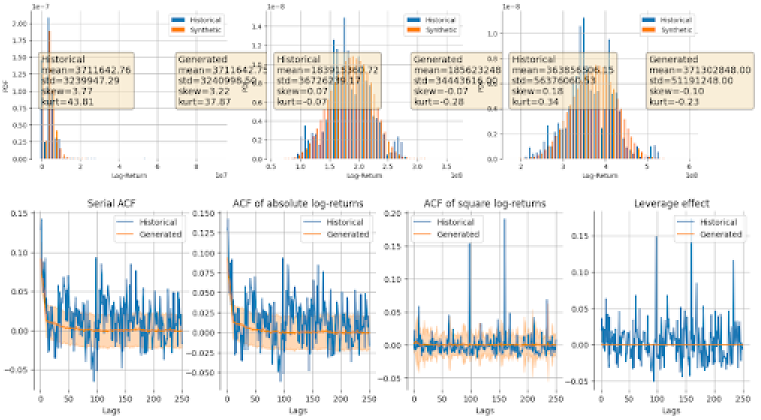
TTGAN With FlowGAN Imputation

With TTGAN, the results already improve with 200 epoch training. More epochs of training result in overfitting.

Training Results



Testing Results



The resulting Wasserstein distance oscillates around 20 thousand and still sufferers from narrow range issues. From the loss plots of testing results, we can see the fat tails are not captured by TTGAN.

Performance Difference Between TTGAN and TAGAN

The performance of FlowGAN imputed losses data in general performs better than Brownian Bridge imputed data. The TTGAN-imputed data is better than the TAGAN-imputed data. This could be because transformers excel at capturing sequential dependencies in data, making them particularly effective for time series data like extreme event losses. The combination of convolutional layers, two-layer MLP, and attention layers in the TTGAN's generator provides a hierarchy of feature representations. The convolutional layers capture local patterns, the MLP processes them further, and the attention layers capture global relationships. This hierarchical approach enhances the model's capacity to learn intricate features and patterns in extreme event data.

Besides, the transformer-based discriminator in the TTGAN is likely more proficient at understanding complex patterns and relationships within the generated data than regular Convolutional NNs. Transformers can process and analyze the generated sequences effectively, enabling the discriminator to distinguish between real and generated extreme event loss data with greater accuracy.

9 Severity Modelling

9.1 FlowGAN

Parallel to our work on the time series approach, we decided to take another look at the modelling of unordered losses that were done in during the previous two team's projects. More specifically, we decided to test out a FlowGAN[5] recommended by the EIB/ECB representatives. In terms of architecture, a FlowGAN simply restricts the activation functions to ones that are piecewise differentiable and invertible (i.e LeakyReLU instead of ReLU). With these activation functions, we can easily compute the log likelihood function of the learnable parameters by the following formula: Let Y be the output of the GAN and X be the random noise input vector with $X \sim p_X(x)$. Let g be the composition of the activation functions with the linear layers and g^{-1} be the inverse, and let $\frac{dg^{-1}}{dy}$ be the inverse Jacobian. Then, the log likelihood function is given by:

$$\log(p_Y(y)) = \log(p_X(g^{-1}(y))) + \log\left(\left|\frac{dg^{-1}}{dy}\right|\right)$$

For a shallow Neural network, the inverse Jacobian should be composed of alternating multiplications of terms of Diagonal matrices representing the inverse derivatives of the activation functions, with the inverse learnable weight matrices.

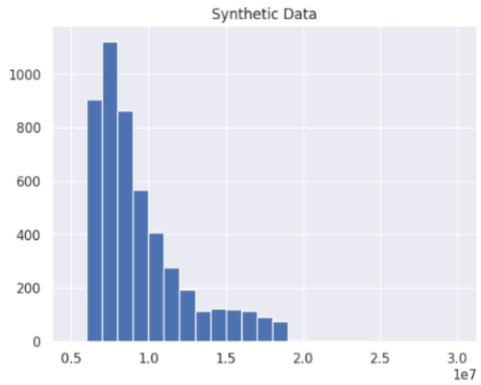
The advantage of the FlowGAN architecture is that with the log likelihood, we can add a maximum likelihood term to the usual GAN loss, hopefully giving better results than we saw in the previous terms.

$$Loss = GAN_{Loss} + w * MLE_{Loss}$$

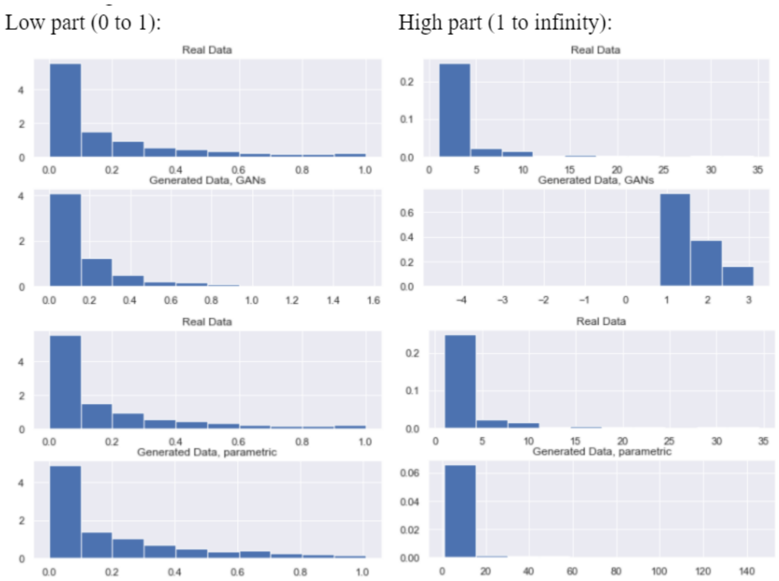
Where w is a hyperparameter representing how much weight we give to the MLE Loss.

9.2 Problems with FlowGAN

We decided to test the FlowGAN architecture using the normalizing flows library from the python library *pyro*. The *pyro* library has a special spline architecture which is especially suited for a FlowGAN architecture. Unfortunately, naively training a FlowGAN to the data for a cluster did not yield promising results:



We were not seeing the long tail or similar descriptive statistics to the real data, which implied that a pure FlowGAN was not good at generating synthetic data. However, we were seeing interesting results when we separated the (normalized) data into a low part, in the interval $(0, 1]$, and a high part, in the interval $[1, \infty)$.



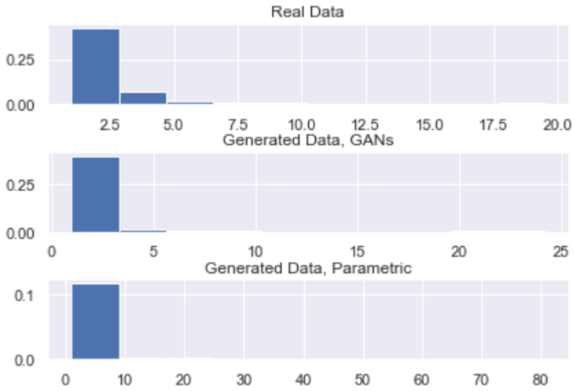
In particular, the FlowGAN was generating promising results for the lower part. We decided to stick with the FlowGAN for the low part and try something else for the higher part of the data, then combine them with a stratified sampler.

9.3 Shallow GAN with a Modified Loss

Previous research teams had conducted experiments employing the DCGAN architecture, while our investigations delved into the TTGAN and TAGAN architectures. Nevertheless, it became apparent that the fundamental, shallow GAN model had not been explored by any prior studies. A shallow GAN's architecture is the simply composition of a few layers of activation functions. Additionally, we needed some way to incorporate the heavy tail that was present in the data. The generator loss we settled on was:

$$\begin{aligned} Loss &= w_1 * ClassicLoss + w_2 * |IQR(x_{fake}) - IQR(x_{real})| \\ &+ w_3 * |tail(x_{fake}) - tail(x_{real})| + w_4 * |min(x_{fake}) - min(x_{real})| \end{aligned}$$

where x_{fake} is the generated data, x_{real} is the real data, IQR is the interquartile range, *tail* is derived from formula 10 of the tail index estimators used in the paper *Fast Tail Index Estimation for Power Law Distributions in R*[6] by Munasinghe et al. and the w_i terms are hyperparameters determining the weight given to each part of the loss. Surprisingly, this actually generated very good results for the upper part of the data:



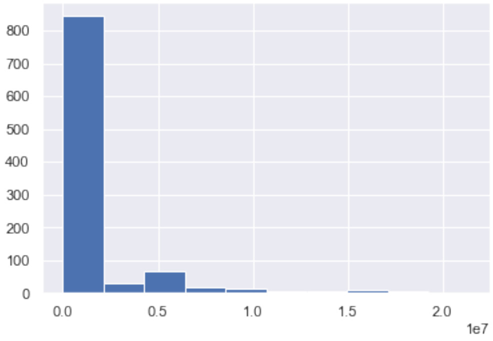
9.4 Stratified Sampling

We decided to combine the positive results from FlowGAN for the lower part of the data, and the positive results from shallow GAN for the upper part of the data into a stratified sampler. Usually, a stratified sampler takes a fixed amount of samples from the lower part and a fixed amount from the upper part, but since we don't care about variance reduction, we decided that treating the sampler like a mixture distribution was simpler. More specifically, we normalize the data by a quantile q and model the segment in $(0, 1)$ by FlowGAN and the segment in $[1, \infty)$ by ShallowGAN. When sampling, we sample by FlowGAN with probability q and from the ShallowGAN with probability $1 - q$. In pseudocode:

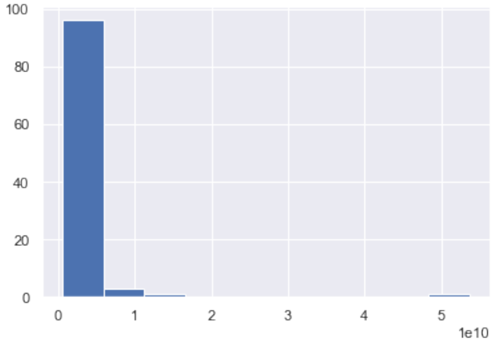
Algorithm: Stratified Sampler

```
Set $q$, the quantile normalization threshold;
Normalize the data by dividing by the $q$th quantile for each country in the cluster;
Train FlowGAN using the segment of the data in $(0, 1)$, and train ShallowGAN using the segment of the data in $[1, \infty)$.
for $i = 1, .., N$:
- generate $U \sim Unif(0, 1)$
- If $U < q$, generate $X_i \sim FlowGAN$
- Else generate $X_i \sim ShallowGAN$
```

We got very good severity modelling results on both the cluster and unnormalizing to the country level as well. For example, here is the synthetic data generated for China:



We can also use the stratified sampler to sample from the tail. For China this is,



10 Clustering Efforts

The previous research teams had already grouped the data based on climate regions, but when we used the test called Kolmogorov-Smirnov (K-S) test, it indicated that the clustering in terms of losses wasn't showing promising results. So, we decided to invest a significant amount of effort into finding a better way to cluster countries. The goal was to group countries together in a way that they share similar patterns and features.

Since methods like FlowGAN, TTGAN, and TAGAN all require a substantial amount of data to work effectively, our approach was to cluster countries based on their losses data and other relevant characteristics. The idea was that by grouping countries with similar attributes, we could increase the size of our dataset. This larger dataset could potentially reduce the need for data imputation methods. In other words, if we have more real data from different countries grouped together, we might rely less on filling in missing yearly data artificially. This approach aimed to enhance the reliability and performance of the data generation models.

10.1 Distance Metrics

Distance matrices play a crucial role in clustering, a technique used to group similar data points together. In clustering, the goal is to find patterns or relationships within the data by measuring how close or far apart different data points are from each other, and the measurement of how close depends heavily on which distance matrix we choose. A distance matrix is a table that stores the distances between every pair of data points. These distances act as a measure of dissimilarity or similarity between points, forming the basis for grouping them into clusters.

Euclidean Distance

In this project, we mainly used Euclidean Distance and Dynamic Time Warping distance matrices. Euclidean Distance is one of the most common ways to measure the distance between two data points in a multi-dimensional space. We used it as the first step in most of our clustering efforts.

$$Euclidean\ Distance(p,q) = \sqrt{\sum_{i=1}^{n}(x_i - y_i)^2}$$

Dynamic Time Warping

DTW distance is employed when dealing with sequences of data, such as time series. It differs from Euclidean Distance because it accommodates shifts and deformations in sequences. For our project the length of time series data for each country are different: some country has more than 40 years of data, and other countries only have 2-3 years of data. DTW accounts for these length differences and identifies the optimal alignment that minimizes differences between the sequences. It's suitable to find the best way to match two sequences even when they're slightly out of sync. Mathematically, for sequences $X = [x_1, x_2, ..., x_n]$ and $Y = [y_1, y_2, ..., y_n]$, the DTW distance between them is computed by finding the optimal alignment that minimizes the cumulative distance between corresponding elements using a dynamic programming approach.

Algorithm: Calculate DTW Distance

```
1. Create a matrix D of size (n+1)×(m+1), where n is the length of sequence X and m is the length of sequence Y.
2. Initialize the first row and column of D to represent the cumulative distances when aligning with an empty subsequence. This step essentially handles the start of the alignment.
3. For each element xi in sequence X and each element $y_j$ in sequence Y, calculate the local distance d($x_i$,$y_j$) between them
4. Update the value in matrix D at position (i+1, j+1) by taking the minimum of the following three values:
    • D(i, j+1)
    • D(i+1, j)
    • D(i,j)
Then, add the local distance d($x_i$,$y_j$) to the minimum value

The DTW distance between X and Y is the value D(n+1, m+1)
DTW = D(n+1,m+1)
```

K-Medoids Customized Distance Metrics

Specifically, K-Medoids focuses on a central point within each cluster, called a medoid. The distance between a data point and a medoid is assessed using customized metrics, depending on the data's nature. K-Medoids aims to identify data points that are closest to the medoid of each cluster, shaping more resilient clusters that are less affected by outliers.

10.2 K-previous-Means clustering

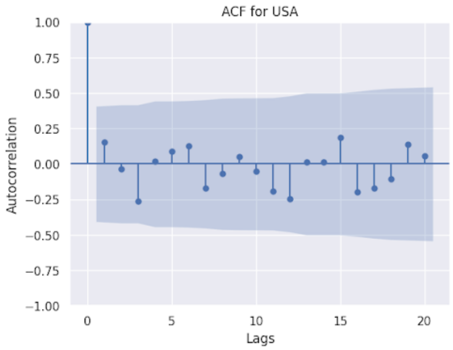
In our pursuit of refining the K-previous means clustering, we engaged in a series of iterations. Our first attempt was working with imputed one-dimensional loss data. To address the uncertainty introduced by random initial states in K-means, we introduced hierarchical clustering before employing K-previous means. We derived centroids from the hierarchical clustering results and used them as the starting points for K-previous means. This helped mitigate the impact of unpredictable initializations. However, this initial iteration didn't incorporate time-related features, such as years. Instead, we compiled individual country records into a collective data frame. As countries had varying numbers of records, we utilized bootstrapping to upsample the record count, which sometimes resulted in repeated records, especially for countries with sparse data.

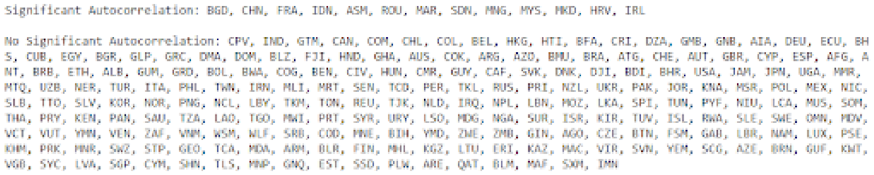
The clustering outcome from this initial approach revealed an interesting pattern. Among the five clusters, four contained a single country each (Chile, Puerto Rico, USA, and Japan), while the remaining 200+ countries were grouped together in one cluster. The challenges posed by bootstrapping and the absence of temporal features were evident.

In a subsequent attempt, we used feature engineering and combined latitude, longitude, and loss data into a single cell for each record. The newly engineered features were then employed in conjunction with K-previous means. The clustering outcomes were similar to our prior experiment: four distinct clusters each dominated by a specific country, with the majority of countries clustered together. Interestingly, even though the resulting clusters seemed unconventional, K-previous means demonstrated better performance compared to clustering based on climate regions.

For the next iteration, we delved into dynamic time warping (DTW) and attempted to cluster time series losses without resorting to upsampling. However, due to potential gaps in the EM-DAT dataset, which served as the source for natural disaster records, imputation was still required. To handle this, we turned to statistical distributions like Poisson, Binomial, and Negative Binomial to estimate disaster counts per year. Based on the relationship between mean and variance, we selected the appropriate distribution. We focused on data from 1980 onwards. We assumed that more recent data would be more complete. Following this, we utilized FlowGAN to predict losses for these newly imputed counts.

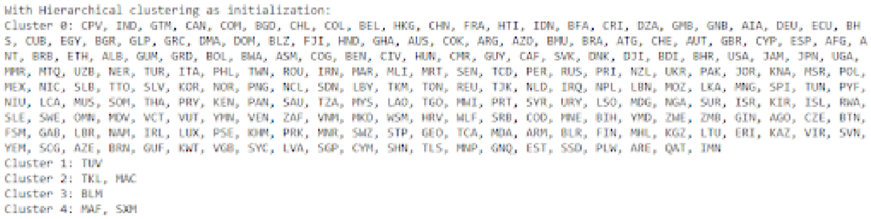
In an effort to refine our approach further, we tried to look into the autocorrelation of the counts for each country and found that most of the countries do not have significant autocorrelation. In this case, we can either look into other types of dependencies between counts or assume the count values are independent of each other. Since "significant" means p-value $< .05$, and there are more than 200 ISO codes, we can conclude that there is pretty much no significant autocorrelation for any country. An example of such lack of correlation is as below plot.





```
Significant Autocorrelation: BGD, CHN, FRA, IDN, ASM, ROU, MAR, SDN, MNG, MYS, MKD, HRV, IRL

No Significant Autocorrelation: CPV, IND, GTM, CAN, COM, CHL, COL, BEL, HKG, HTI, BFA, CRI, DZA, GMB, GNB, AIA, DEU, ECU, BH
S, CUB, EGY, BGR, GLP, GRC, DMA, DOM, BLZ, FJI, HND, GHA, AUS, COK, ARG, AZO, BMU, BRA, ATG, CHE, AUT, GBR, CYP, ESP, AFG, A
NT, BRB, ETH, ALB, GUM, GRD, BOL, BWA, COG, BEN, CIV, HUN, CMR, GUY, CAF, SVK, DNK, DJI, BDI, BHR, USA, JAM, JPN, UGA, MMR,
MTQ, UZB, NER, TUR, ITA, PHL, TWN, IRN, MLI, MRT, SEN, TCD, PER, TKL, RUS, PRI, NZL, UKR, PAK, JOR, KNA, MSR, POL, MEX, NIC,
SLB, TTO, SLV, KOR, NOR, PNG, NCL, LBY, TKM, TON, REU, TJK, NLD, IRQ, NPL, LBN, MOZ, LKA, SPI, TUN, PYF, NIU, LCA, MUS, SOM,
THA, PRY, KEN, PAN, SAU, TZA, LAO, TGO, MWI, PRT, SYR, URY, LSO, MDG, NGA, SUR, ISR, KIR, TUV, ISL, RWA, SLE, SWE, OMN, MDV,
VCT, VUT, YMN, VEN, ZAF, VNM, WSM, WLF, SRB, COD, MNE, BIH, YMD, ZWE, ZMB, GIN, AGO, CZE, BTN, FSM, GAB, LBR, NAM, LUX, PSE,
KHM, PRK, MNR, SWZ, STP, GEO, TCA, MDA, ARM, BLR, FIN, MHL, KGZ, LTU, ERI, KAZ, MAC, VIR, SVN, YEM, SCG, AZE, BRN, GUF, KWT,
VGB, SYC, LVA, SGP, CYM, SHN, TLS, MNP, GNQ, EST, SSD, PLW, ARE, QAT, BLM, MAF, SXM, IMN
```

Then we tried to cluster countries by the counts of extreme events with DTW distance, but the results are still with the same structure.



```
With Hierarchical clustering as initialization:
Cluster 0: CPV, IND, GTM, CAN, COM, BGD, CHL, COL, BEL, HKG, CHN, FRA, HTI, IDN, BFA, CRI, DZA, GMB, GNB, AIA, DEU, ECU, BH
S, CUB, EGY, BGR, GLP, GRC, DMA, DOM, BLZ, FJI, HND, GHA, AUS, COK, ARG, AZO, BMU, BRA, ATG, CHE, AUT, GBR, CYP, ESP, AFG, A
NT, BRB, ETH, ALB, GUM, GRD, BOL, BWA, ASM, COG, BEN, CIV, HUN, CMR, GUY, CAF, SVK, DNK, DJI, BDI, BHR, USA, JAM, JPN, UGA,
MMR, MTQ, UZB, NER, TUR, ITA, PHL, TWN, ROU, IRN, MAR, MLI, MRT, SEN, TCD, PER, RUS, PRI, NZL, UKR, PAK, JOR, KNA, MSR, POL,
MEX, NIC, SLB, TTO, SLV, KOR, NOR, PNG, NCL, SDN, LBY, TKM, TON, REU, TJK, NLD, IRQ, NPL, LBN, MOZ, LKA, MNG, SPI, TUN, PYF,
NIU, LCA, MUS, SOM, THA, PRY, KEN, PAN, SAU, TZA, MYS, LAO, TGO, MWI, PRT, SYR, URY, LSO, MDG, NGA, SUR, ISR, KIR, ISL, RWA,
SLE, SWE, OMN, MDV, VCT, VUT, YMN, VEN, ZAF, VNM, MKD, WSM, HRV, WLF, SRB, COD, MNE, BIH, YMD, ZWE, ZMB, GIN, AGO, CZE, BTN,
FSM, GAB, LBR, NAM, IRL, LUX, PSE, KHM, PRK, MNR, SWZ, STP, GEO, TCA, MDA, ARM, BLR, FIN, MHL, KGZ, LTU, ERI, KAZ, VIR, SVN,
YEM, SCG, AZE, BRN, GUF, KWT, VGB, SYC, LVA, SGP, CYM, SHN, TLS, MNP, GNQ, EST, SSD, PLW, ARE, QAT, IMN
Cluster 1: TUV
Cluster 2: TKL, MAC
Cluster 3: BLM
Cluster 4: MAF, SXM
```

In our third attempt, we took a different approach by using imputed loss data generated by FlowGAN. This data was then fed into the K-previous means clustering algorithm, where we tested both Euclidean Distance and DTW distance metrics. However, even with these modifications, we continued to face the challenge of a phenomenon known as the "small number of countries dominating one cluster."

To address this persistent issue, we excluded the countries that significantly dominated a single cluster. The rationale behind this was that these dominant countries might be too dissimilar from the rest, causing them to stand out as outliers. By removing them, we aimed to create a more balanced representation of countries that shared similarities in their loss patterns. Unfortunately, this attempt also yielded unsatisfactory results. Instead of resolving the dominance problem, we observed a different outcome. After eliminating the initially dominant countries, new dominating countries emerged from the original cluster that contained the majority of countries. This unexpected outcome indicated that the underlying complexities of the data and its clustering behavior were more intricate than initially anticipated.

10.3 K-Medoids Using Weighted Averages Of Geographical Distance

Parallel to the other clustering methods, we also experimented with K-Medoids, a clustering algorithm that only required a distance matrix. The advantage of this clustering algorithm is that we are able to incorporate more general metric spaces rather than just euclidean space. For example, the shape of the earth is well approximated by a sphere and the distance between 2 points is not given by the Euclidean Distance, but the Great Circle Distance. More specifically, for two points on earth's surface, parametrized by their latitudes and longitudes (ϕ_1, λ_1), (ϕ_2, λ_2), the distance between them (arbitrarily setting the radius of the earth to 1 unit) is well approximated by:

$$
\begin{aligned}
\text{atan2}(&\sqrt{(\cos\phi_2 \sin(\lambda_2 - \lambda_1))^2 + (\cos\phi_1 \sin\phi_2 - \sin\phi_1 \cos\phi_2 \cos(\lambda_2 - \lambda_1))^2}, \\
&\sin\phi_1 \sin\phi_2 + \cos\phi_1 \cos\phi_2 \cos(\lambda_2 - \lambda_1))
\end{aligned}
$$

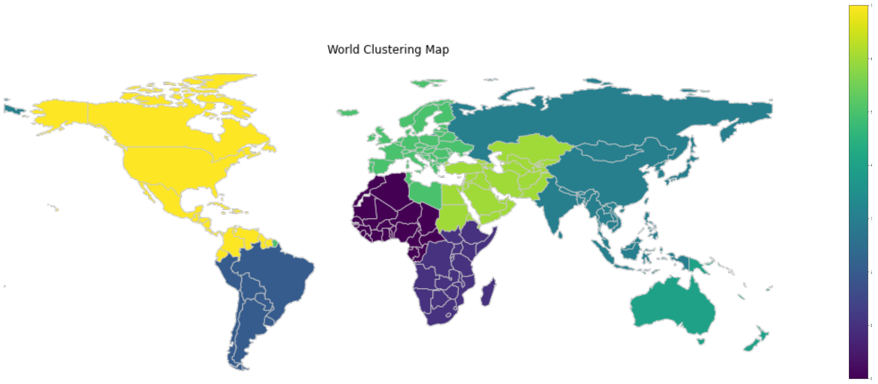
We decided to test K-Medoids clustering using:

1. Constructing the distance matrix from only the Great Circle Distance between geograph-

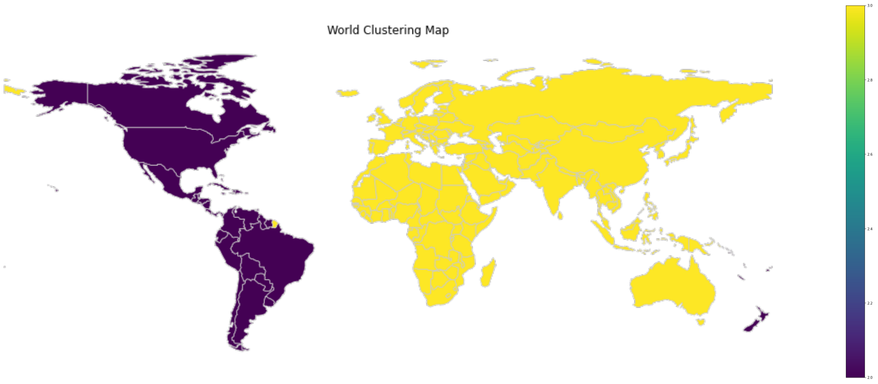
ical position of losses (if exact values were not given, we used the average latitude and longitude of the country)

2. Constructing the distance matrix from a weighted average of the Great Circle Distance and other metrics of loss (for example dynamic time warping of counts).

When we were only using the Great Circle Distance, we got decent results:



However, when we added other measures of distance to the distance matrix, for example, DTW between counts, the results just got worse:

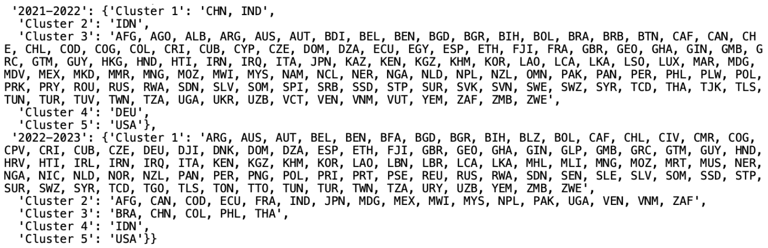


We decided to try other clustering methods instead.

10.4 Multidimensional Hierarchical Clustering

With the belief that multidimensional data often contains complex relationships and interactions among various features of records, for example, there might be underlying causal relations extreme event loss and land area, we considered utilizing multiple dimensions simultaneously for clustering countries, you can capture these intricate patterns and dependencies, in order to capture any missed signals when clustering in a single dimension. We were also able to include any joint effects of various variables that might explains the rarity and outlier cases, resulting in more meaningful and robust clustering results to combat with the single-dominating-country-cluster issue, as mentioned earlier in 11.2.

To begin with our attempt, we need careful feature engineering to remove biases through normalization. We incorporated both the EM-DAT and the land area database to recreate 3 features: frequency per year, total loss per land area per year, and CPI per year for each country. The records were grouped by countries, into a dataframe of feature tuples each year(col) for each country(row). We performed Hierarchical Clustering for stabilizing the initial centroids for each year's data where each country should represent one normalized point, and countries without full records of features of certain year will not participate in clustering on that round. The result showed the same challenge of "small number of countries dominating one cluster".



```
 '2021–2022': {'Cluster 1': 'CHN, IND',
  'Cluster 2': 'IDN',
  'Cluster 3': 'AFG, AGO, ALB, ARG, AUS, AUT, BDI, BEL, BEN, BGD, BGR, BIH, BOL, BRA, BRB, BTN, CAF, CAN, CH
E, CHL, COD, COG, COL, CRI, CUB, CYP, CZE, DOM, DZA, ECU, EGY, ESP, ETH, FJI, FRA, GBR, GEO, GHA, GIN, GMB, G
RC, GTM, GUY, HKG, HND, HTI, IRN, IRQ, ITA, JPN, KAZ, KEN, KGZ, KHM, KOR, LAO, LCA, LKA, LSO, LUX, MAR, MDG,
MDV, MEX, MKD, MMR, MNG, MOZ, MWI, MYS, NAM, NCL, NER, NGA, NLD, NPL, NZL, OMN, PAK, PAN, PER, PHL, PLW, POL,
PRK, PRY, ROU, RUS, RWA, SDN, SLV, SOM, SPI, SRB, SSD, STP, SUR, SVK, SVN, SWE, SWZ, SYR, TCD, THA, TJK, TLS,
TUN, TUR, TUV, TWN, TZA, UGA, UKR, UZB, VCT, VEN, VNM, VUT, YEM, ZAF, ZMB, ZWE',
  'Cluster 4': 'DEU',
  'Cluster 5': 'USA'},
 '2022–2023': {'Cluster 1': 'ARG, AUS, AUT, BEL, BEN, BFA, BGD, BGR, BIH, BLZ, BOL, CAF, CHL, CIV, CMR, COG,
CPV, CRI, CUB, CZE, DEU, DJI, DNK, DOM, DZA, ESP, ETH, FJI, GBR, GEO, GHA, GIN, GLP, GMB, GRC, GTM, GUY, HND,
HRV, HTI, IRL, IRN, IRQ, ITA, KEN, KGZ, KHM, KOR, LAO, LBN, LBR, LCA, LKA, MHL, MLI, MNG, MOZ, MRT, MUS, NER,
NGA, NIC, NLD, NOR, NZL, PAN, PER, PNG, POL, PRI, PRT, PSE, REU, RUS, RWA, SDN, SEN, SLE, SLV, SOM, SSD, STP,
SUR, SWZ, SYR, TCD, TGO, TLS, TON, TTO, TUN, TUR, TWN, TZA, URY, UZB, YEM, ZMB, ZWE',
  'Cluster 2': 'AFG, CAN, COD, ECU, FRA, IND, JPN, MDG, MEX, MWI, MYS, NPL, PAK, UGA, VEN, VNM, ZAF',
  'Cluster 3': 'BRA, CHN, COL, PHL, THA',
  'Cluster 4': 'IDN',
  'Cluster 5': 'USA'}}
```

Using the same practice of excluding dominating single clusters, unfortunately the results were still unpromising to still outputting extreme separation of countries. The continuation of issue indicated the intricacy of centroid selection methods that needed to be further investigated.

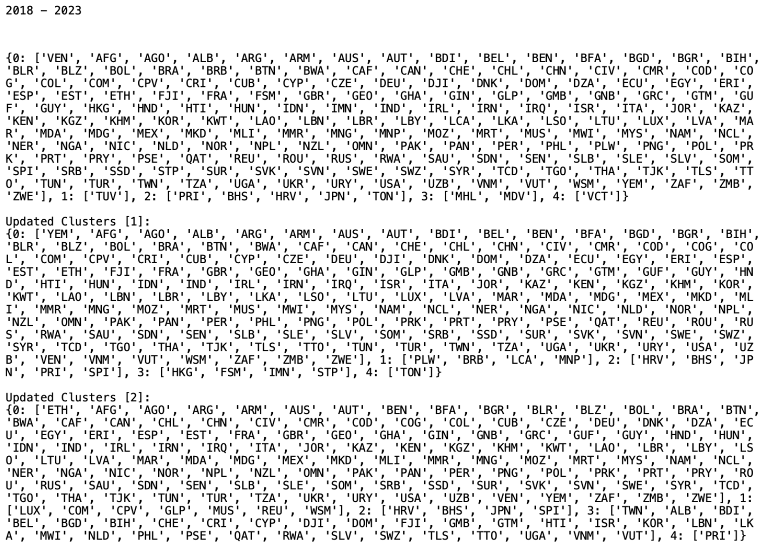
10.5 Sliding Window Clustering With K-means++

Noting that Hierarchical Clustering for centroid initialization is more likely to give versatile clustering shapes and sizes, we attempt to use KMeans++ for centroid initialization, which allowed selection of data points further from pre-selected extremes. As a results, points closer to the "major crowd" is more likely to be selected if the previously selected points were the "edged outliers". Technically, it increases the possibility of revealing more even clustering shapes and sizes, as oppose to giving single-country-clusters that maximizes intra-cluster variance.

We also extended our clustering approach, originally designed for single-year data, to encompass records from adjacent years in a given timeframe—a technique known as Sliding Window Clustering[1] by Borassi et al. Essentially, it allows us to update clusters with each arrival of recent years' data instead of recomputing them entirely from scratch for the whole data sets. This strategy significantly enhances computational efficiency. Furthermore, the sliding-window method provides the benefit of isolating the effects of data from the distant past. For instance, the financial situation in 2009 closely resembled that of 2008 but differed significantly from the situation in 2023. By doing so, it allows us to place greater emphasis on recent extreme influences, which is particularly valuable when analyzing evolving trends and patterns. This approach not only optimizes our clustering process but also ensures that our models are responsive to the most recent and impactful data, contributing to more accurate and up-to-date results. The window size is a hyperparameter that can be selected based on the recovery cycle length of various types of disasters, which may also be a subject for future research.

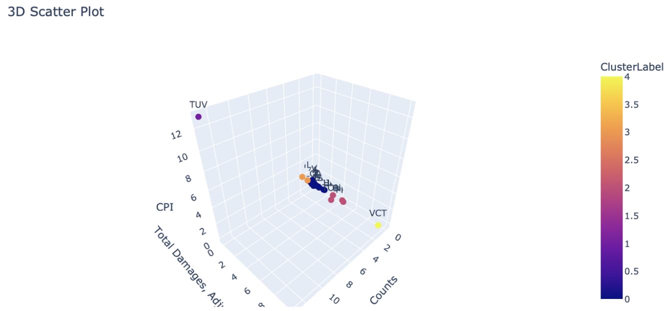
Without loss of generality, we conducted experiments using window sizes ranging from 5 to 10 years. We continued to utilize the same three features as in the previous section, with slight adjustments: counts per land area, total losses per land area, and CPI per land area. For each window frame, records that matched that specified time frame were selected, and normalized features were computed. We then employed K-Means++ to determine centroids for each window, to which countries were assigned based on their Euclidean Distances. While the first iteration still posed the "small number of countries dominating one cluster" issue, we re-clustered the data for several iterations, during which clusters containing fewer than a certain number of countries were discarded; additionally, we recomputed the features for that frame for re-clustering purpose.

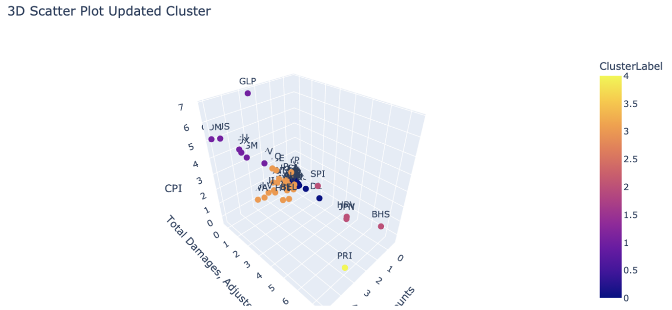
3-dimensional KMeans++ with re-clustering threshold of 4, 5-years window after 3 iterations:



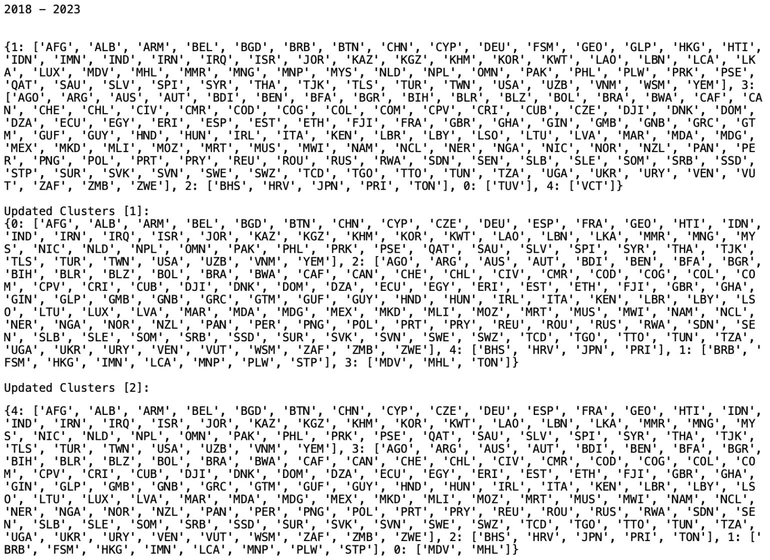
```
2018 - 2023


{0: ['VEN', 'AFG', 'AGO', 'ALB', 'ARG', 'ARM', 'AUS', 'AUT', 'BDI', 'BEL', 'BEN', 'BFA', 'BGD', 'BGR', 'BIH',
'BLR', 'BLZ', 'BOL', 'BRA', 'BRB', 'BTN', 'BWA', 'CAF', 'CAN', 'CHE', 'CHL', 'CHN', 'CIV', 'CMR', 'COD', 'CO
G', 'COL', 'COM', 'CPV', 'CRI', 'CUB', 'CYP', 'CZE', 'DEU', 'DJI', 'DNK', 'DOM', 'DZA', 'ECU', 'EGY', 'ERI',
'ESP', 'EST', 'ETH', 'FJI', 'FRA', 'FSM', 'GBR', 'GEO', 'GHA', 'GIN', 'GLP', 'GMB', 'GNB', 'GRC', 'GTM', 'GU
F', 'GUY', 'HKG', 'HND', 'HTI', 'HUN', 'IDN', 'IMN', 'IND', 'IRL', 'IRN', 'IRQ', 'ISR', 'ITA', 'JOR', 'KAZ',
'KEN', 'KGZ', 'KHM', 'KOR', 'KWT', 'LAO', 'LBN', 'LBR', 'LBY', 'LCA', 'LKA', 'LSO', 'LTU', 'LUX', 'LVA', 'MA
R', 'MDA', 'MDG', 'MEX', 'MKD', 'MLI', 'MMR', 'MNG', 'MNP', 'MOZ', 'MRT', 'MUS', 'MWI', 'MYS', 'NAM', 'NCL',
'NER', 'NGA', 'NIC', 'NLD', 'NOR', 'NPL', 'NZL', 'OMN', 'PAK', 'PAN', 'PER', 'PHL', 'PLW', 'PNG', 'POL', 'PR
K', 'PRT', 'PRY', 'PSE', 'QAT', 'REU', 'ROU', 'RUS', 'RWA', 'SAU', 'SDN', 'SEN', 'SLB', 'SLE', 'SLV', 'SOM',
'SPI', 'SRB', 'SSD', 'STP', 'SUR', 'SVK', 'SVN', 'SWE', 'SWZ', 'SYR', 'TCD', 'TGO', 'THA', 'TJK', 'TLS', 'TT
O', 'TUN', 'TUR', 'TWN', 'TZA', 'UGA', 'UKR', 'URY', 'USA', 'UZB', 'VNM', 'VUT', 'WSM', 'YEM', 'ZAF', 'ZMB',
'ZWE'], 1: ['TUV'], 2: ['PRI', 'BHS', 'HRV', 'JPN', 'TON'], 3: ['MHL', 'MDV'], 4: ['VCT']}

Updated Clusters [1]:
{0: ['YEM', 'AFG', 'AGO', 'ALB', 'ARG', 'ARM', 'AUS', 'AUT', 'BDI', 'BEL', 'BEN', 'BFA', 'BGD', 'BGR', 'BIH',
'BLR', 'BLZ', 'BOL', 'BRA', 'BTN', 'BWA', 'CAF', 'CAN', 'CHE', 'CHL', 'CHN', 'CIV', 'CMR', 'COD', 'COG', 'CO
L', 'COM', 'CPV', 'CRI', 'CUB', 'CYP', 'CZE', 'DEU', 'DJI', 'DNK', 'DOM', 'DZA', 'ECU', 'EGY', 'ERI', 'ESP',
'EST', 'ETH', 'FJI', 'FRA', 'GBR', 'GEO', 'GHA', 'GIN', 'GLP', 'GMB', 'GNB', 'GRC', 'GTM', 'GUF', 'GUY', 'HN
D', 'HTI', 'HUN', 'IDN', 'IND', 'IRL', 'IRN', 'IRQ', 'ISR', 'ITA', 'JOR', 'KAZ', 'KEN', 'KGZ', 'KHM', 'KOR',
'KWT', 'LAO', 'LBN', 'LBR', 'LBY', 'LKA', 'LSO', 'LTU', 'LUX', 'LVA', 'MAR', 'MDA', 'MDG', 'MEX', 'MKD', 'ML
I', 'MMR', 'MNG', 'MOZ', 'MRT', 'MUS', 'MWI', 'MYS', 'NAM', 'NCL', 'NER', 'NGA', 'NIC', 'NLD', 'NOR', 'NPL',
'NZL', 'OMN', 'PAK', 'PAN', 'PER', 'PHL', 'PNG', 'POL', 'PRK', 'PRT', 'PRY', 'PSE', 'QAT', 'REU', 'ROU', 'RU
S', 'RWA', 'SAU', 'SDN', 'SEN', 'SLB', 'SLE', 'SLV', 'SOM', 'SRB', 'SSD', 'SUR', 'SVK', 'SVN', 'SWE', 'SWZ',
'SYR', 'TCD', 'TGO', 'THA', 'TJK', 'TLS', 'TTO', 'TUN', 'TUR', 'TWN', 'TZA', 'UGA', 'UKR', 'URY', 'USA', 'UZ
B', 'VEN', 'VNM', 'VUT', 'WSM', 'ZAF', 'ZMB', 'ZWE'], 1: ['PLW', 'BRB', 'LCA', 'MNP'], 2: ['HRV', 'BHS', 'JP
N', 'PRI', 'SPI'], 3: ['HKG', 'FSM', 'IMN', 'STP'], 4: ['TON']}

Updated Clusters [2]:
{0: ['ETH', 'AFG', 'AGO', 'ARG', 'ARM', 'AUS', 'AUT', 'BEN', 'BFA', 'BGR', 'BLR', 'BLZ', 'BOL', 'BRA', 'BTN',
'BWA', 'CAF', 'CAN', 'CHL', 'CHN', 'CIV', 'CMR', 'COD', 'COG', 'COL', 'CUB', 'CZE', 'DEU', 'DNK', 'DZA', 'EC
U', 'EGY', 'ERI', 'ESP', 'EST', 'FRA', 'GBR', 'GEO', 'GHA', 'GIN', 'GNB', 'GRC', 'GUF', 'GUY', 'HND', 'HUN',
'IDN', 'IND', 'IRL', 'IRN', 'IRQ', 'ITA', 'JOR', 'KAZ', 'KEN', 'KGZ', 'KHM', 'KWT', 'LAO', 'LBR', 'LBY', 'LS
O', 'LTU', 'LVA', 'MAR', 'MDA', 'MDG', 'MEX', 'MKD', 'MLI', 'MMR', 'MNG', 'MOZ', 'MRT', 'MYS', 'NAM', 'NCL',
'NER', 'NGA', 'NIC', 'NOR', 'NPL', 'NZL', 'OMN', 'PAK', 'PAN', 'PER', 'PNG', 'POL', 'PRK', 'PRT', 'PRY', 'RO
U', 'RUS', 'SAU', 'SDN', 'SEN', 'SLB', 'SLE', 'SOM', 'SRB', 'SSD', 'SUR', 'SVK', 'SVN', 'SWE', 'SYR', 'TCD',
'TGO', 'THA', 'TJK', 'TUN', 'TUR', 'TZA', 'UKR', 'URY', 'USA', 'UZB', 'VEN', 'YEM', 'ZAF', 'ZMB', 'ZWE'], 1:
['LUX', 'COM', 'CPV', 'GLP', 'MUS', 'REU', 'WSM'], 2: ['HRV', 'BHS', 'JPN', 'SPI'], 3: ['TWN', 'ALB', 'BDI',
'BEL', 'BGD', 'BIH', 'CHE', 'CRI', 'CYP', 'DJI', 'DOM', 'FJI', 'GMB', 'GTM', 'HTI', 'ISR', 'KOR', 'LBN', 'LK
A', 'MWI', 'NLD', 'PHL', 'PSE', 'QAT', 'RWA', 'SLV', 'SWZ', 'TLS', 'TTO', 'UGA', 'VNM', 'VUT'], 4: ['PRI']}
```

In this specific sample window frame, we observed the emergence of clusters with over 10 countries, indicating a trend toward forming a 'major crowd' and subsequently separating the central core around the origin (after normalization). Visualizing in 3D:



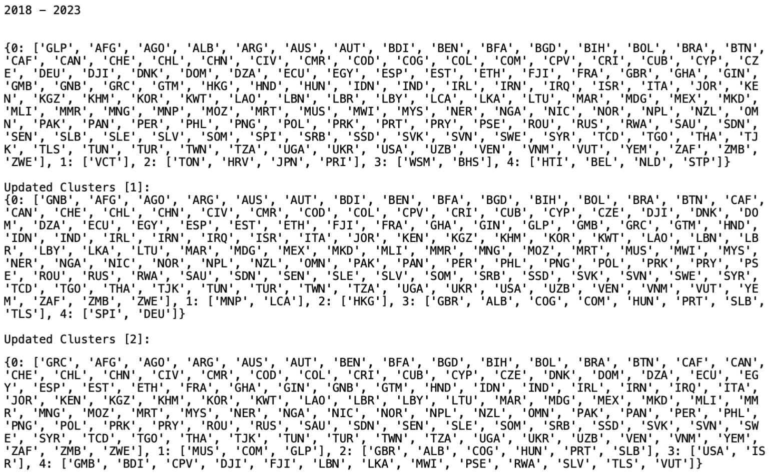


For the next step we delve into higher dimensions of 5D. We tried 2 combinations of adding extra features: categorical features(continent, region) using K-Prototypes or numerical features(total death per land area, total number of injured per land area). The recent clustering result of 5-D KMeans++ with categorical features:



```
2018 - 2023


{1: ['AFG', 'ALB', 'ARM', 'BEL', 'BGD', 'BRB', 'BTN', 'CHN', 'CYP', 'DEU', 'FSM', 'GEO', 'GLP', 'HKG', 'HTI',
'IDN', 'IMN', 'IND', 'IRN', 'IRQ', 'ISR', 'JOR', 'KAZ', 'KGZ', 'KHM', 'KOR', 'KWT', 'LAO', 'LBN', 'LCA', 'LK
A', 'LUX', 'MDV', 'MHL', 'MMR', 'MNG', 'MNP', 'MYS', 'NLD', 'NPL', 'OMN', 'PAK', 'PHL', 'PLW', 'PRK', 'PSE',
'QAT', 'SAU', 'SLV', 'SPI', 'SYR', 'THA', 'TJK', 'TLS', 'TUR', 'TWN', 'USA', 'UZB', 'VNM', 'WSM', 'YEM'], 3:
['AGO', 'ARG', 'AUS', 'AUT', 'BDI', 'BEN', 'BFA', 'BGR', 'BIH', 'BLR', 'BLZ', 'BOL', 'BRA', 'BWA', 'CAF', 'CA
N', 'CHE', 'CHL', 'CIV', 'CMR', 'COD', 'COG', 'COL', 'COM', 'CPV', 'CRI', 'CUB', 'CZE', 'DJI', 'DNK', 'DOM',
'DZA', 'ECU', 'EGY', 'ERI', 'ESP', 'EST', 'ETH', 'FJI', 'FRA', 'GBR', 'GHA', 'GIN', 'GMB', 'GNB', 'GRC', 'GT
M', 'GUF', 'GUY', 'HND', 'HUN', 'IRL', 'ITA', 'KEN', 'LBR', 'LBY', 'LSO', 'LTU', 'LVA', 'MAR', 'MDA', 'MDG',
'MEX', 'MKD', 'MLI', 'MOZ', 'MRT', 'MUS', 'MWI', 'NAM', 'NCL', 'NER', 'NGA', 'NIC', 'NOR', 'NZL', 'PAN', 'PE
R', 'PNG', 'POL', 'PRT', 'PRY', 'REU', 'ROU', 'RUS', 'RWA', 'SDN', 'SEN', 'SLB', 'SLE', 'SOM', 'SRB', 'SSD',
'STP', 'SUR', 'SVK', 'SVN', 'SWE', 'SWZ', 'TCD', 'TGO', 'TTO', 'TUN', 'TZA', 'UGA', 'UKR', 'URY', 'VEN', 'VU
T', 'ZAF', 'ZMB', 'ZWE'], 2: ['BHS', 'HRV', 'JPN', 'PRI', 'TON'], 0: ['TUV'], 4: ['VCT']}

Updated Clusters [1]:
{0: ['AFG', 'ALB', 'ARM', 'BEL', 'BGD', 'BTN', 'CHN', 'CYP', 'CZE', 'DEU', 'ESP', 'FRA', 'GEO', 'HTI', 'IDN',
'IND', 'IRN', 'IRQ', 'ISR', 'JOR', 'KAZ', 'KGZ', 'KHM', 'KOR', 'KWT', 'LAO', 'LBN', 'LKA', 'MMR', 'MNG', 'MY
S', 'NIC', 'NLD', 'NPL', 'OMN', 'PAK', 'PHL', 'PRK', 'PSE', 'QAT', 'SAU', 'SLV', 'SPI', 'SYR', 'THA', 'TJK',
'TLS', 'TUR', 'TWN', 'USA', 'UZB', 'VNM', 'YEM'], 2: ['AGO', 'ARG', 'AUS', 'AUT', 'BDI', 'BEN', 'BFA', 'BGR',
'BIH', 'BLR', 'BLZ', 'BOL', 'BRA', 'BWA', 'CAF', 'CAN', 'CHE', 'CHL', 'CIV', 'CMR', 'COD', 'COG', 'COL', 'CO
M', 'CPV', 'CRI', 'CUB', 'DJI', 'DNK', 'DOM', 'DZA', 'ECU', 'EGY', 'ERI', 'EST', 'ETH', 'FJI', 'GBR', 'GHA',
'GIN', 'GLP', 'GMB', 'GNB', 'GRC', 'GTM', 'GUF', 'GUY', 'HND', 'HUN', 'IRL', 'ITA', 'KEN', 'LBR', 'LBY', 'LS
O', 'LTU', 'LUX', 'LVA', 'MAR', 'MDA', 'MDG', 'MEX', 'MKD', 'MLI', 'MOZ', 'MRT', 'MUS', 'MWI', 'NAM', 'NCL',
'NER', 'NGA', 'NOR', 'NZL', 'PAN', 'PER', 'PNG', 'POL', 'PRT', 'PRY', 'REU', 'ROU', 'RUS', 'RWA', 'SDN', 'SE
N', 'SLB', 'SLE', 'SOM', 'SRB', 'SSD', 'SUR', 'SVK', 'SVN', 'SWE', 'SWZ', 'TCD', 'TGO', 'TTO', 'TUN', 'TZA',
'UGA', 'UKR', 'URY', 'VEN', 'VUT', 'WSM', 'ZAF', 'ZMB', 'ZWE'], 4: ['BHS', 'HRV', 'JPN', 'PRI'], 1: ['BRB', '
FSM', 'HKG', 'IMN', 'LCA', 'MNP', 'PLW', 'STP'], 3: ['MDV', 'MHL', 'TON']}

Updated Clusters [2]:

{4: ['AFG', 'ALB', 'ARM', 'BEL', 'BGD', 'BTN', 'CHN', 'CYP', 'CZE', 'DEU', 'ESP', 'FRA', 'GEO', 'HTI', 'IDN',
'IND', 'IRN', 'IRQ', 'ISR', 'JOR', 'KAZ', 'KGZ', 'KHM', 'KOR', 'KWT', 'LAO', 'LBN', 'LKA', 'MMR', 'MNG', 'MY
S', 'NIC', 'NLD', 'NPL', 'OMN', 'PAK', 'PHL', 'PRK', 'PSE', 'QAT', 'SAU', 'SLV', 'SPI', 'SYR', 'THA', 'TJK',
'TLS', 'TUR', 'TWN', 'USA', 'UZB', 'VNM', 'YEM'], 3: ['AGO', 'ARG', 'AUS', 'AUT', 'BDI', 'BEN', 'BFA', 'BGR',
'BIH', 'BLR', 'BLZ', 'BOL', 'BRA', 'BWA', 'CAF', 'CAN', 'CHE', 'CHL', 'CIV', 'CMR', 'COD', 'COG', 'COL', 'CO
M', 'CPV', 'CRI', 'CUB', 'DJI', 'DNK', 'DOM', 'DZA', 'ECU', 'EGY', 'ERI', 'EST', 'ETH', 'FJI', 'GBR', 'GHA',
'GIN', 'GLP', 'GMB', 'GNB', 'GRC', 'GTM', 'GUF', 'GUY', 'HND', 'HUN', 'IRL', 'ITA', 'KEN', 'LBR', 'LBY', 'LS
O', 'LTU', 'LUX', 'LVA', 'MAR', 'MDA', 'MDG', 'MEX', 'MKD', 'MLI', 'MOZ', 'MRT', 'MUS', 'MWI', 'NAM', 'NCL',
'NER', 'NGA', 'NOR', 'NZL', 'PAN', 'PER', 'PNG', 'POL', 'PRT', 'PRY', 'REU', 'ROU', 'RUS', 'RWA', 'SDN', 'SE
N', 'SLB', 'SLE', 'SOM', 'SRB', 'SSD', 'SUR', 'SVK', 'SVN', 'SWE', 'SWZ', 'TCD', 'TGO', 'TTO', 'TUN', 'TZA',
'UGA', 'UKR', 'URY', 'VEN', 'VUT', 'WSM', 'ZAF', 'ZMB', 'ZWE'], 2: ['BHS', 'HRV', 'JPN', 'PRI', 'TON'], 1: ['
BRB', 'FSM', 'HKG', 'IMN', 'LCA', 'MNP', 'PLW', 'STP'], 0: ['MDV', 'MHL']}
```

And with purely numerical features:



```
2018 - 2023


{0: ['GLP', 'AFG', 'AGO', 'ALB', 'ARG', 'AUS', 'AUT', 'BDI', 'BEN', 'BFA', 'BGD', 'BIH', 'BOL', 'BRA', 'BTN',
'CAF', 'CAN', 'CHE', 'CHL', 'CHN', 'CIV', 'CMR', 'COD', 'COG', 'COL', 'COM', 'CPV', 'CRI', 'CUB', 'CYP', 'CZ
E', 'DEU', 'DJI', 'DNK', 'DOM', 'DZA', 'ECU', 'EGY', 'ESP', 'EST', 'ETH', 'FJI', 'FRA', 'GBR', 'GHA', 'GIN',
'GMB', 'GNB', 'GRC', 'GTM', 'HKG', 'HND', 'HUN', 'IDN', 'IND', 'IRL', 'IRN', 'IRQ', 'ISR', 'ITA', 'JOR', 'KE
N', 'KGZ', 'KHM', 'KOR', 'KWT', 'LAO', 'LBN', 'LBR', 'LBY', 'LCA', 'LKA', 'LTU', 'MAR', 'MDG', 'MEX', 'MKD',
'MLI', 'MMR', 'MNG', 'MNP', 'MOZ', 'MRT', 'MUS', 'MWI', 'MYS', 'NER', 'NGA', 'NIC', 'NOR', 'NPL', 'NZL', 'OM
N', 'PAK', 'PAN', 'PER', 'PHL', 'PNG', 'POL', 'PRK', 'PRT', 'PRY', 'PSE', 'ROU', 'RUS', 'RWA', 'SAU', 'SDN',
'SEN', 'SLB', 'SLE', 'SLV', 'SOM', 'SPI', 'SRB', 'SSD', 'SVK', 'SVN', 'SWE', 'SYR', 'TCD', 'TGO', 'THA', 'TJ
K', 'TLS', 'TUN', 'TUR', 'TWN', 'TZA', 'UGA', 'UKR', 'USA', 'UZB', 'VEN', 'VNM', 'VUT', 'YEM', 'ZAF', 'ZMB',
'ZWE'], 1: ['VCT'], 2: ['TON', 'HRV', 'JPN', 'PRI'], 3: ['WSM', 'BHS'], 4: ['HTI', 'BEL', 'NLD', 'STP']}

Updated Clusters [1]:
{0: ['GNB', 'AFG', 'AGO', 'ARG', 'AUS', 'AUT', 'BDI', 'BEN', 'BFA', 'BGD', 'BIH', 'BOL', 'BRA', 'BTN', 'CAF',
'CAN', 'CHE', 'CHL', 'CHN', 'CIV', 'CMR', 'COD', 'COL', 'CPV', 'CRI', 'CUB', 'CYP', 'CZE', 'DJI', 'DNK', 'DO
M', 'DZA', 'ECU', 'EGY', 'ESP', 'EST', 'ETH', 'FJI', 'FRA', 'GHA', 'GIN', 'GLP', 'GMB', 'GRC', 'GTM', 'HND',
'IDN', 'IND', 'IRL', 'IRN', 'IRQ', 'ISR', 'ITA', 'JOR', 'KEN', 'KGZ', 'KHM', 'KOR', 'KWT', 'LAO', 'LBN', 'LB
R', 'LBY', 'LKA', 'LTU', 'MAR', 'MDG', 'MEX', 'MKD', 'MLI', 'MMR', 'MNG', 'MOZ', 'MRT', 'MUS', 'MWI', 'MYS',
'NER', 'NGA', 'NIC', 'NOR', 'NPL', 'NZL', 'OMN', 'PAK', 'PAN', 'PER', 'PHL', 'PNG', 'POL', 'PRK', 'PRY', 'PS
E', 'ROU', 'RUS', 'RWA', 'SAU', 'SDN', 'SEN', 'SLE', 'SLV', 'SOM', 'SRB', 'SSD', 'SVK', 'SVN', 'SWE', 'SYR',
'TCD', 'TGO', 'THA', 'TJK', 'TUN', 'TUR', 'TWN', 'TZA', 'UGA', 'UKR', 'USA', 'UZB', 'VEN', 'VNM', 'VUT', 'YE
M', 'ZAF', 'ZMB', 'ZWE'], 1: ['MNP', 'LCA'], 2: ['HKG'], 3: ['GBR', 'ALB', 'COG', 'COM', 'HUN', 'PRT', 'SLB',
'TLS'], 4: ['SPI', 'DEU']}

Updated Clusters [2]:

{0: ['GRC', 'AFG', 'AGO', 'ARG', 'AUS', 'AUT', 'BEN', 'BFA', 'BGD', 'BIH', 'BOL', 'BRA', 'BTN', 'CAF', 'CAN',
'CHE', 'CHL', 'CHN', 'CIV', 'CMR', 'COD', 'COL', 'CRI', 'CUB', 'CYP', 'CZE', 'DNK', 'DOM', 'DZA', 'ECU', 'EG
Y', 'ESP', 'EST', 'ETH', 'FRA', 'GHA', 'GIN', 'GNB', 'GTM', 'HND', 'IDN', 'IND', 'IRL', 'IRN', 'IRQ', 'ITA',
'JOR', 'KEN', 'KGZ', 'KHM', 'KOR', 'KWT', 'LAO', 'LBR', 'LBY', 'LTU', 'MAR', 'MDG', 'MEX', 'MKD', 'MLI', 'MM
R', 'MNG', 'MOZ', 'MRT', 'MYS', 'NER', 'NGA', 'NIC', 'NOR', 'NPL', 'NZL', 'OMN', 'PAK', 'PAN', 'PER', 'PHL',
'PNG', 'POL', 'PRK', 'PRY', 'ROU', 'RUS', 'SAU', 'SDN', 'SEN', 'SLE', 'SOM', 'SRB', 'SSD', 'SVK', 'SVN', 'SW
E', 'SYR', 'TCD', 'TGO', 'THA', 'TJK', 'TUN', 'TUR', 'TWN', 'TZA', 'UGA', 'UKR', 'UZB', 'VEN', 'VNM', 'YEM',
'ZAF', 'ZMB', 'ZWE'], 1: ['MUS', 'COM', 'GLP'], 2: ['GBR', 'ALB', 'COG', 'HUN', 'PRT', 'SLB'], 3: ['USA', 'IS
R'], 4: ['GMB', 'BDI', 'CPV', 'DJI', 'FJI', 'LBN', 'LKA', 'MWI', 'PSE', 'RWA', 'SLV', 'TLS', 'VUT']}
```

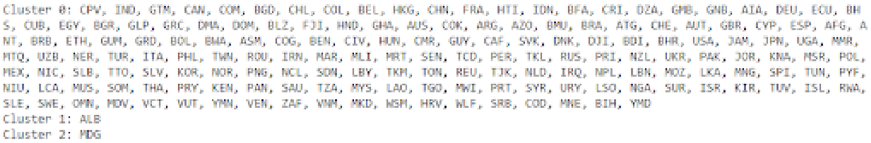
We noticed improved performance with the categorical version of the 5D model, resulting in more balanced cluster sizes. This suggests a connection between incident occurrences and the geographical locations of countries. Possible future work at this stage includes subdividing regions based on factors such as seismic zones and climate activities, which may involve the incorporation of external databases. Another consideration is the development of a "vulnerability index," as discussed in the book *Towards a European-wide Vulnerability Framework: A Flexible Approach for Vulnerability Assessment Using Composite Indicators*[3], which aims to quantify different types of disasters.

10.6 DBSCAN

As a parallel practice with K-previous means, we considered using DBSCAN to test whether the structure of clustering results is due to the outliers. The reason for using DBSCAN was that it doesn't require us to decide in advance how many clusters we expect in the data, and it's also capable of identifying and excluding outliers. To put this to the test, we applied DBSCAN to the imputed loss data with FlowGAN. We used both Euclidean Distance and DTW distance metrics for this experiment, maintaining the same framework as the other clustering methods we employed.

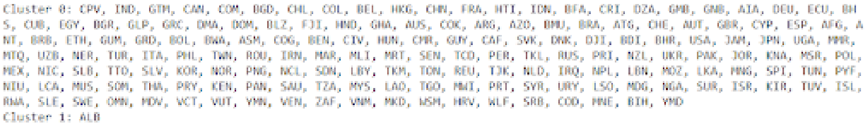
When we used DBSCAN with the Euclidean Distance metric, it outputs 3 distinct clusters within the data. This approach helps us group similar data points together based on their proximity in a one-dimensional space. On the other hand, when we applied DBSCAN with the DTW distance metric, it detected 2 clusters in the dataset. Both results still give most of the countries to be in one cluster. This eliminates the assumption that this clustering structure is caused by outliers.

DBSCAN with Euclidean Distance metrics:



```
Cluster 0: CPV, IND, GTM, CAN, COM, BGD, CHL, COL, BEL, HKG, CHN, FRA, HTI, IDN, BFA, CRI, DZA, GMB, GNB, AIA, DEU, ECU, BH
S, CUB, EGY, BGR, GLP, GRC, DMA, DOM, BLZ, FJI, HND, GHA, AUS, COK, ARG, AZO, BMU, BRA, ATG, CHE, AUT, GBR, CYP, ESP, AFG, A
NT, BRB, ETH, GUM, GRD, BOL, BWA, ASM, COG, BEN, CIV, HUN, CMR, GUY, CAF, SVK, DNK, DJI, BDI, BHR, USA, JAM, JPN, UGA, MMR,
MTQ, UZB, NER, TUR, ITA, PHL, TWN, ROU, IRN, MAR, MLI, MRT, SEN, TCD, PER, TKL, RUS, PRI, NZL, UKR, PAK, JOR, KNA, MSR, POL,
MEX, NIC, SLB, TTO, SLV, KOR, NOR, PNG, NCL, SDN, LBY, TKM, TON, REU, TJK, NLD, IRQ, NPL, LBN, MOZ, LKA, MNG, SPI, TUN, PYF,
NIU, LCA, MUS, SOM, THA, PRY, KEN, PAN, SAU, TZA, MYS, LAO, TGO, MWI, PRT, SYR, URY, LSO, NGA, SUR, ISR, KIR, TUV, ISL, RWA,
SLE, SWE, OMN, MDV, VCT, VUT, YMN, VEN, ZAF, VNM, MKD, WSM, HRV, WLF, SRB, COD, MNE, BIH, YMD
Cluster 1: ALB
Cluster 2: MDG
```

DBSCAN with DTW distance metrics:



```
Cluster 0: CPV, IND, GTM, CAN, COM, BGD, CHL, COL, BEL, HKG, CHN, FRA, HTI, IDN, BFA, CRI, DZA, GMB, GNB, AIA, DEU, ECU, BH
S, CUB, EGY, BGR, GLP, GRC, DMA, DOM, BLZ, FJI, HND, GHA, AUS, COK, ARG, AZO, BMU, BRA, ATG, CHE, AUT, GBR, CYP, ESP, AFG, A
NT, BRB, ETH, GUM, GRD, BOL, BWA, ASM, COG, BEN, CIV, HUN, CMR, GUY, CAF, SVK, DNK, DJI, BDI, BHR, USA, JAM, JPN, UGA, MMR,
MTQ, UZB, NER, TUR, ITA, PHL, TWN, ROU, IRN, MAR, MLI, MRT, SEN, TCD, PER, TKL, RUS, PRI, NZL, UKR, PAK, JOR, KNA, MSR, POL,
MEX, NIC, SLB, TTO, SLV, KOR, NOR, PNG, NCL, SDN, LBY, TKM, TON, REU, TJK, NLD, IRQ, NPL, LBN, MOZ, LKA, MNG, SPI, TUN, PYF,
NIU, LCA, MUS, SOM, THA, PRY, KEN, PAN, SAU, TZA, MYS, LAO, TGO, MWI, PRT, SYR, URY, LSO, MDG, NGA, SUR, ISR, KIR, TUV, ISL,
RWA, SLE, SWE, OMN, MDV, VCT, VUT, YMN, VEN, ZAF, VNM, MKD, WSM, HRV, WLF, SRB, COD, MNE, BIH, YMD
Cluster 1: ALB
```

11 Conclusion

In the context of this study, we have undertaken the following tasks:

1. Accomplished the original goal of the fall 2022 and spring 2023 teams: We built a GAN architecture that can generate both typical and extreme losses from an unordered and stationary perspective. To do this we used a FlowGAN for the "typical" losses, and a ShallowGAN with a custom tail loss for the "extreme" losses.
2. Started looking at the more general problem of dealing with the losses from a time series perspective, which incorporates the timing of losses into the data.
3. Explored many different methods of clustering, concluding that more work needs to be done using high dimensional methods.

There were a lot of experiments with different methods in this project that led to failure, so if we could restart we would focus more on the high dimensional clustering, since that has lead to the most promising results for time series methods.

12 Future Work

In the pursuit of future developments, a focus on further fine-tuning the parameters of TAGAN and TTGAN is envisioned. This meticulous parameter optimization process aims to enhance the performance of these GAN models and their capacity to generate synthetic data that emulates real-world extreme event distributions.

Recognizing the significance of multidimensional clustering, we aim to shift from single-dimensional approaches to mitigate bias from dominating countries and achieve a more nuanced understanding of extreme event distribution. In this context, a more meticulous feature engineering process involving line search may be integrated to identify the optimal combination for clustering.

In addition to these primary areas of focus, exploration of advanced clustering techniques and the incorporation of external datasets is contemplated, such as subdividing the region category to align with geographical activities. These efforts align with the overarching goal of enhancing the robustness and accuracy of risk assessment models, ultimately enabling more informed decision-making in the realm of extreme event forecasting and risk management.

Finally, we discussed a change of perspective of the problem to using renewal-reward processes: i.e modelling the interarrival times *between* natural disasters as opposed to the accumulation of natural disasters in a fixed timeframe. We decided not to go this route because a lot of the data only showed the year of disaster, as opposed to anything finer scaled. However, there may be some interesting results from only modelling the countries that have this fine scaled data to use.

References

[1] Michele Borassi, Alessandro Epasto, Silvio Lattanzi, Sergei Vassilvitskii, and Morteza Zadimoghaddam. Sliding window algorithms for k-clustering problems. *CoRR*, abs/2006.05850, 2020. URL https://arxiv.org/abs/2006.05850.

[2] Center for Research on the Epidemiology of Disasters. Em-dat: The international disaster database, 2022. URL https://www.emdat.be.

[3] European Commission, Joint Research Centre, G Eklund, A Salvi, T Antofie, A Sibila, D Rodomonti, S Salari, K Poljansek, S Marzi, Z Gyenes, and C Corbane. *Towards a European wide vulnerability framework – A flexible approach for vulnerability assessment using composite indicators*. Publications Office of the European Union, 2023. doi: doi/10.2760/353889.

[4] Weilong Fu, Ali Hirsa, and Jörg Osterrieder. Simulating financial time series using attention. 2022. URL https://arxiv.org/abs/2207.00493.

[5] Aditya Grover, Manik Dhar, and Stefano Ermon. Flow-gan: Combining maximum likelihood and adversarial learning in generative models. 2018. URL https://arxiv.org/abs/1705.08868.

[6] Ranjiva Munasinghe, Pathum Kossinna, Dovini Jayasinghe, and Dilanka Wijeratne. Fast tail index estimation for power law distributions in r. 2020. URL https://arxiv.org/abs/2006.10308.

[7] Anastasios A. Tsonis, Kyle L. Swanson, and Paul J. Roebber. What do networks have to do with climate? *Bulletin of the American Meteorological Society*, 89(5):585–596, 2006. URL https://doi.org/10.1175/BAMS-87-5-585.