PROJ 201 Project Final Report

Project Title: Morphological Classification of Magnetar Bursts detected by Fermi GBM

Name, Surname & ID of group members:
Şefik Efe Altınoluk
Muhammed Abtaha Farooq

Supervised by:Ersin Göğüş

5/12/2024

Abstract

Magnetars, highly magnetized neutron stars, are known for their intense bursts of X-rays and gamma rays, typically detected as light curves by the Fermi Gamma-ray Burst Monitor (GBM). This project aims to classify magnetar burst events based on their morphological features using unsupervised machine learning techniques, specifically K-Means clustering. The analysis involved preprocessing Fermi GBM data, extracting nine key features from light curves, and applying Principal Component Analysis (PCA) to reduce dimensionality while retaining critical information. Three clustering techniques—K-Means, DBSCAN, and Hierarchical Clustering—were used to identify patterns and groupings in the data. The results demonstrated that K-Means clustering effectively categorized the bursts into three distinct groups, with features such as Decay Time, Rise Time, and Duration emerging as primary differentiators. The study offers valuable insights into the temporal and energetic properties of magnetar bursts, contributing to a better understanding of their underlying physical mechanisms. However, limitations related to detector dependency and algorithmic assumptions were noted, highlighting areas for further investigation in future work.

Introduction

Magnetars, a rare and highly magnetized type of neutron star, are known for their intense bursts of hard X-rays and gamma rays, often triggered by fractures in their magnetically stressed crusts. These bursts are typically detected in the form of light curves, which graph the intensity of the emissions over time, providing a visual representation of their energy release. The light curve is an essential tool for studying these events and reveals the rapid and intense nature of magnetar bursts.

Our awareness of magnetar bursts has been significantly enhanced by the Fermi Gamma-ray Space Telescope, equipped with the Gamma-ray Burst Monitor (GBM), which detects and monitors these high-energy events. Fermi, a space-based observatory launched in 2008, has expanded our capacity to observe these phenomena, offering critical data on the temporal and spectral characteristics of bursts. The GBM is specifically designed to detect and monitor high-energy events from 8 keV to 40 MeV, making it well-suited for studying transient cosmic phenomena such as gamma-ray bursts (GRBs), solar flares, and magnetar outbursts. There are 12 Sodium Iodide (NaI) and 2 Bismuth Germanate scintillators available on it. These are basically materials that scintillates when excited by ionized radiation by absorbing and re-emitting the energy in the form of light. Each detector is placed in a way that overall they cover the space in all angles.

A burst is detected by multiple detectors that are close enough to each other to detect the same event. Thus, the background noise is reduced and burst resolution is increased when the data collected by both detectors are combined.

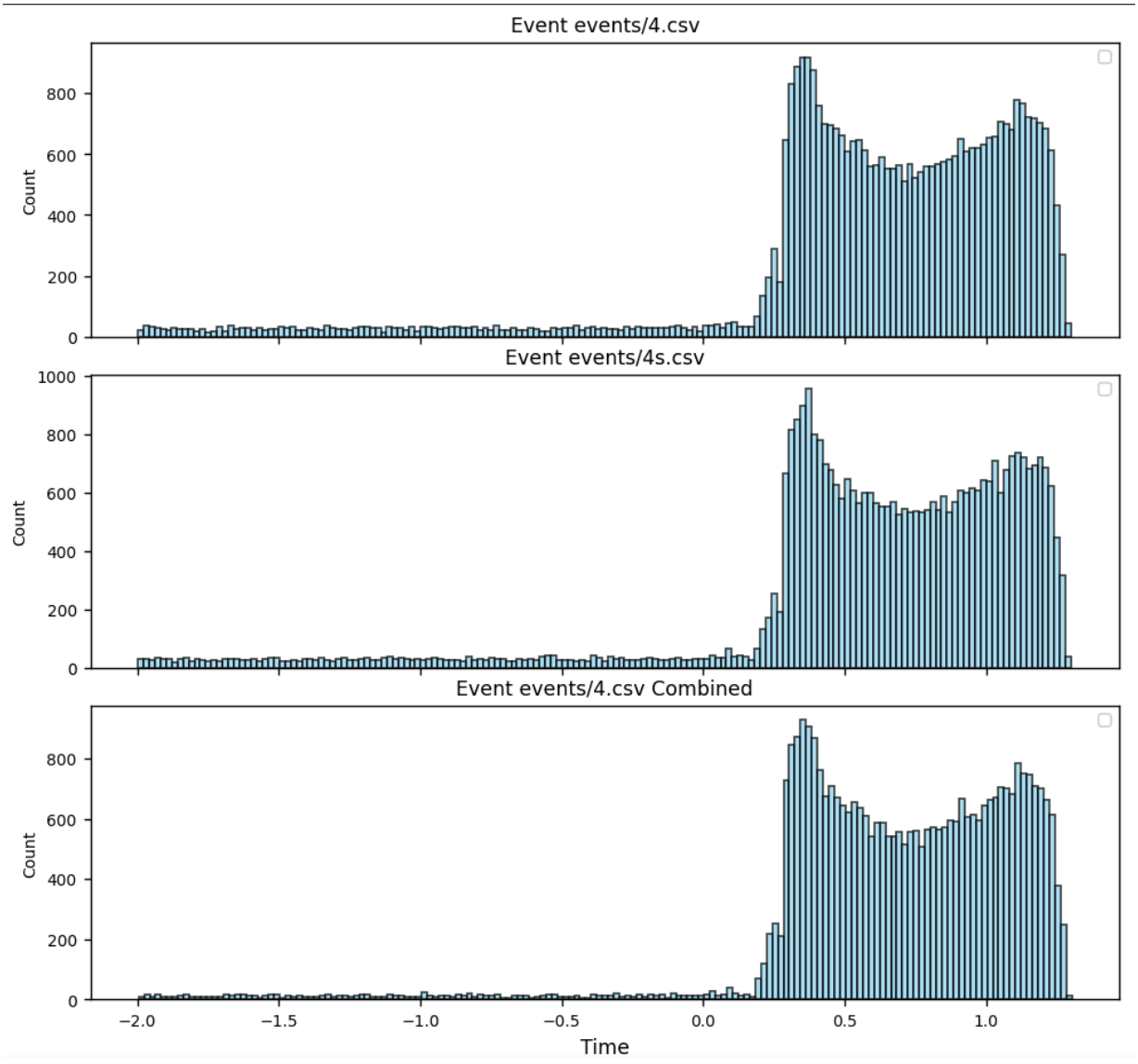
Magnetar bursts generally form a light curve formation, these light curves can be visualized by a histogram of equally distanced bins based on the boundaries of event arrival times and number of photons detected in the particular bins. The magnetar bursts occur as one pulse at a time.If there are more than one pulse in a time series, the pulses should be separated for a correct processing and interpretation as they represent different bursts.

In this project our purpose was categorizing a set of various magnetar bursts based on features that we had extracted from each one of them. Leveraging data from Fermi's GBM, we utilized K-Means Clustering algorithm, an unsupervised machine learning process, to classify these bursts by the patterns recognized between distinct features of the bursts. Then we drew morphological conclusions based on the results we achieved, offering new insights into the extreme environments that drive such powerful emissions.

Methods and Materials

This project focuses on processing and interpreting morphological data collected by Fermi GBM. To achieve this, we used several methods and materials. First of all, for each event, we used two data sets. One of them is collected by the detector directly facing the burst, where the second one is collected by the angularly closest detector to the main detector. We employed Matplotlib, a library packet which has statistical and geometrical plotting functions available, in the Python programming language to visualize the data that we had in the form of time - energy pairs (time series). We wrote functions that plot time series as a histogram enabling us to see the light curve formation in event records for both datasets. Then we combined two datasets into one single dataset to have a more clean and precise event data. The histograms in *Fig 1* (see below) represents main detector data, secondary detector data and combined data from top to bottom respectively.

Fig 1



Then we computed a unique threshold value for each event to filter out background noise and detect the burst itself. Here's how the threshold is calculated:

σ = standard deviation of pre-burst photon counts

μ = mean of pre-burst photon counts

Threshold = σ x 5 + μ

Here are the figures of an event with its threshold calculated and shown on its histogram.

Fig 2

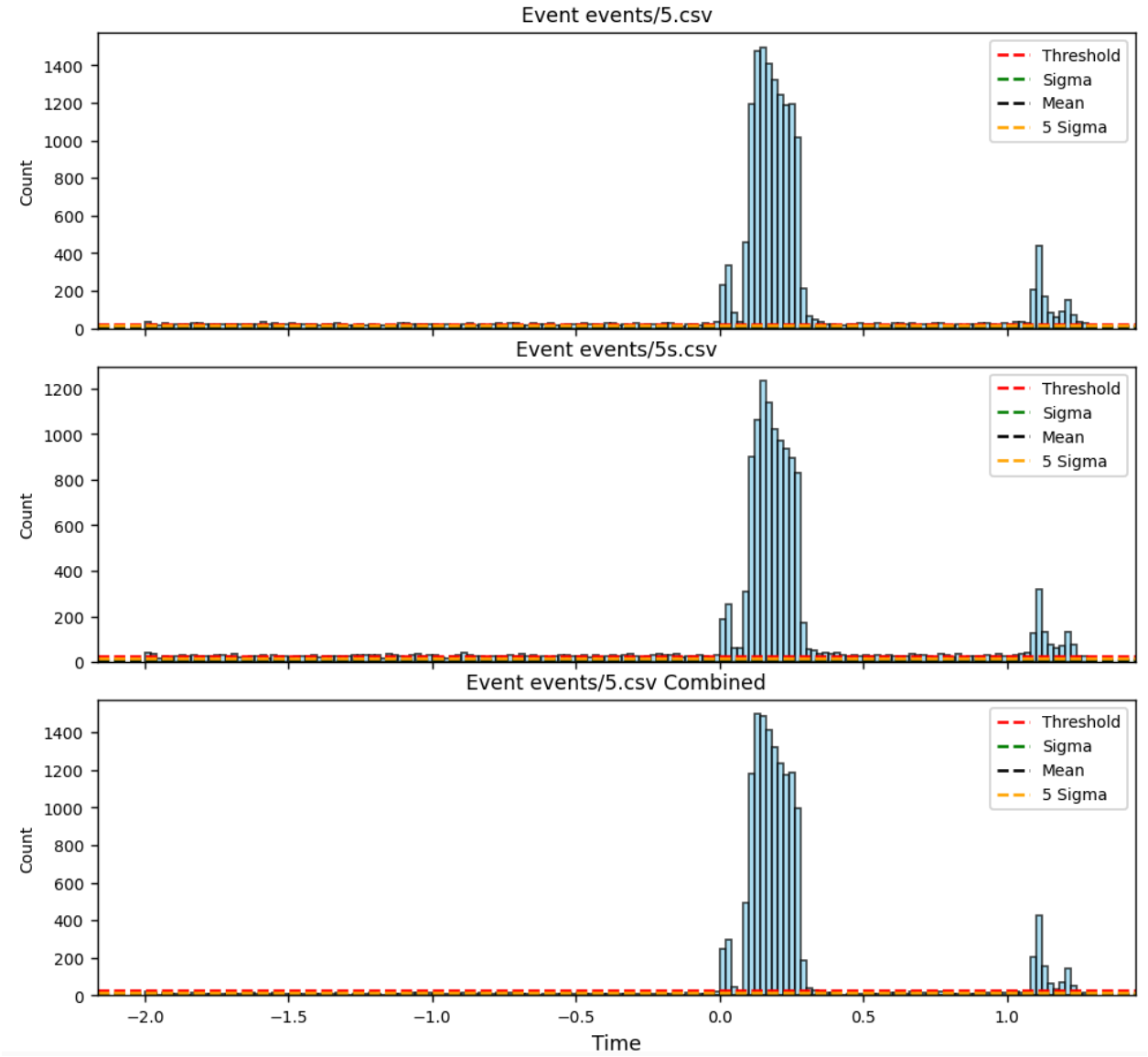


Fig 3

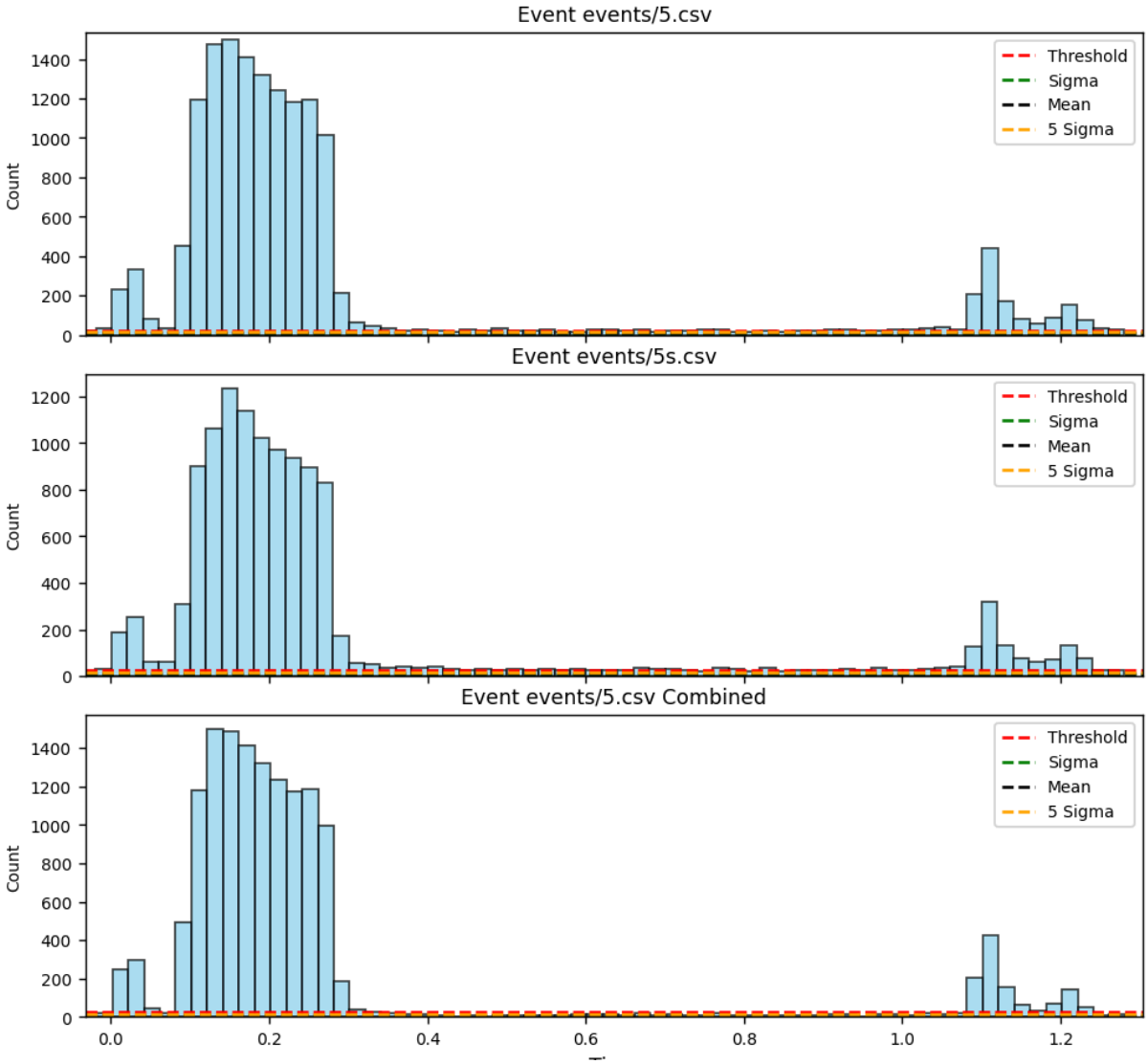
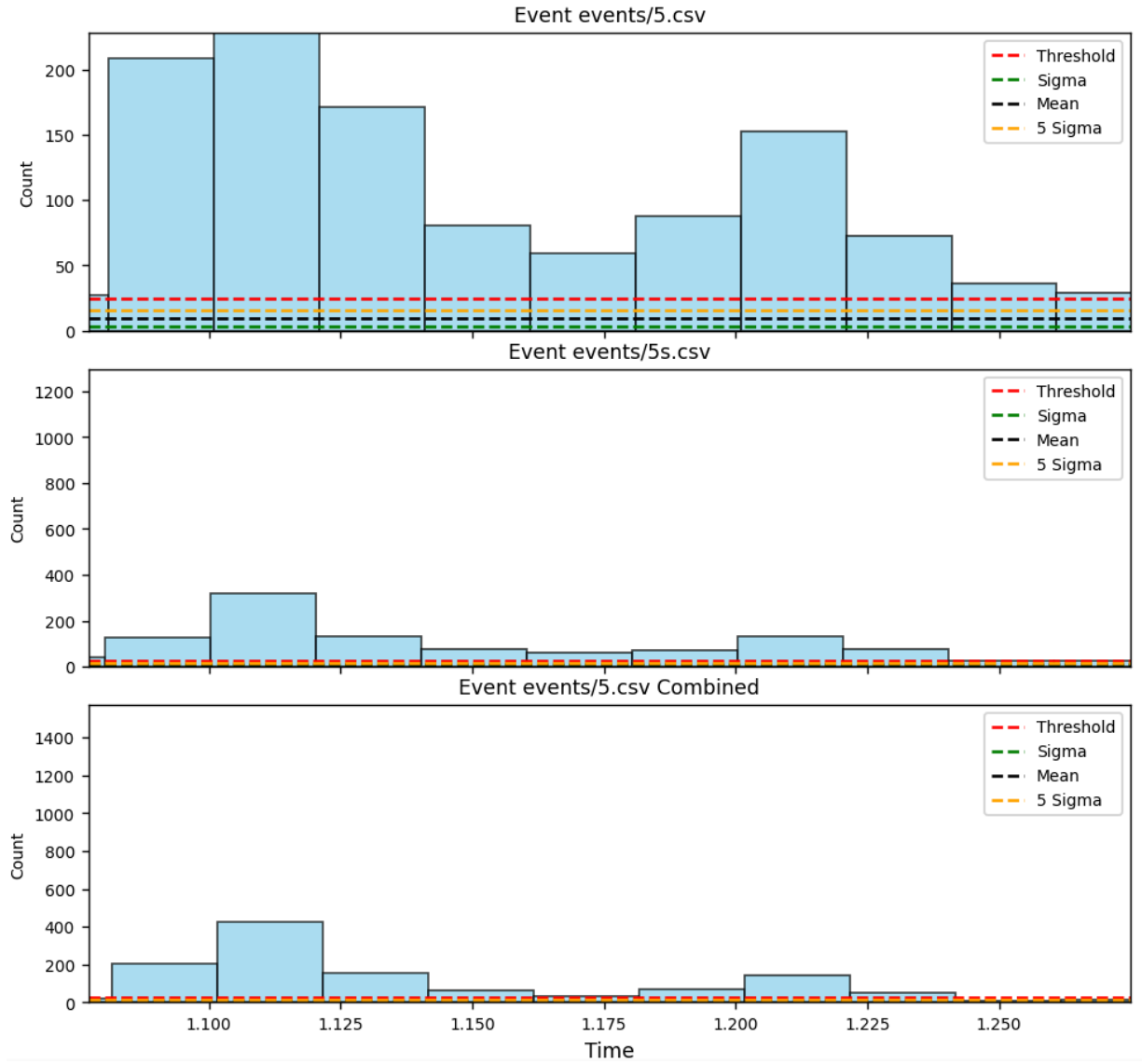
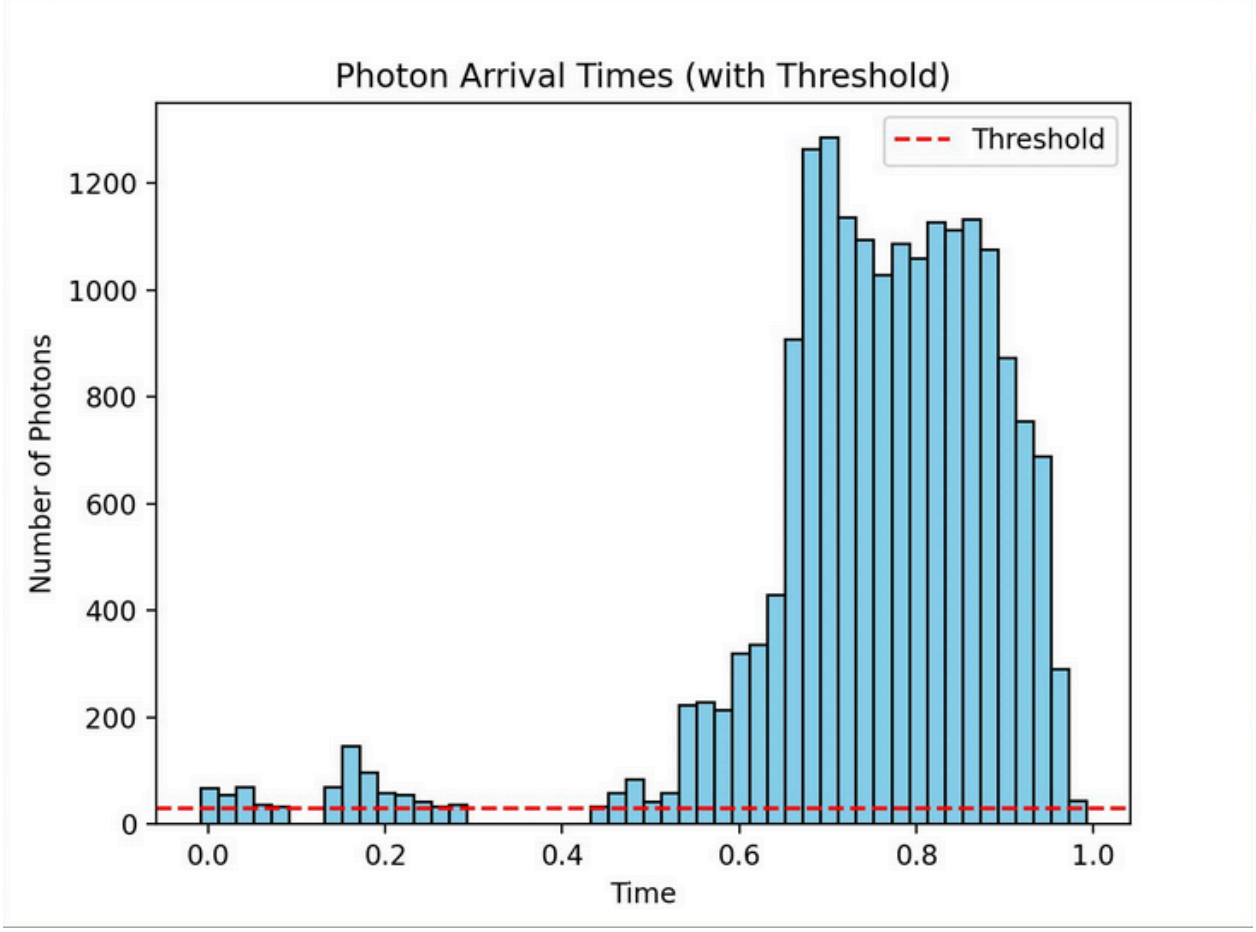

Event events/5.csv
Threshold
Sigma
Mean
5 Sigma
Count
Event events/5s.csv
Threshold
Sigma
Mean
5 Sigma
Count
Event events/5.csv Combined
Threshold
Sigma
Mean
5 Sigma
Count
0.0
0.2
0.4
0.6
0.8
1.0
1.2

Fig 4



According to the threshold, we determined the number of pulses in each event record. We wrote another function to separate them and export their data separately to further process each pulse individually. The total number of pulses revealed to be 221 according to our calculations. See another thresholded event example in fig 5.

Fig 5: Thresholded Histogram of an event recording



Processing event data:

Feature extraction: We calculated the following 9 morphological/mathematical features for each pulse to use in classifying the magnetar bursts.

- "Peak Energy Bin": bin which has the most energy emission (time interval).
- "Peak Energy In Bin": Total energy level in the bin which has the most energy emission.
- "Rise Time": Rising time of the event, determined by calculating the time between the bin with the highest photon count and beginning of the event.
- "Decay Time": Decaying time of the event, determined by calculating the time between the bin with the highest photon count and the end of the event.
- "Duration": Timespan of the event interval.
- "Centroid": center of the event based on photon count
- "Skewness": The skewness of the light curve histogram.
- "Kurtosis": The kurtosis of the light curve histogram.

- "Total Energy Released": The total energy released by the event, calculated by summing the photon energies.

Feature decision: We utilized the Principal Component Analysis to determine the features that are the most uncorrelated in our dataset. The idea is finding out the features that will have the most impact on the classification.

- Principal Component Analysis (PCA) is a widely used method for both deciding the variables that affect the variance of a dataset and visualizing a dataset with more than three dimensions.
- Principal components are derived variables formed by combining the original variables through linear combinations. These combinations are designed to ensure that the principal components are uncorrelated, with the majority of the information from the original variables concentrated in the first few components.
- In a 10-dimensional dataset, Principal Component Analysis (PCA) generates 10 principal components. The primary objective of PCA is to maximize the variance captured by the first principal component, followed by the next highest variance in the second component, and so forth, ensuring that each subsequent component captures the maximum remaining information from the dataset. See fig 6.
- Principal components are difficult to interpret and lack intrinsic meaning because they are derived as linear combinations of the original variables

Fig 6: Sum of Variance (Information) for by Principal Component (out of 1)

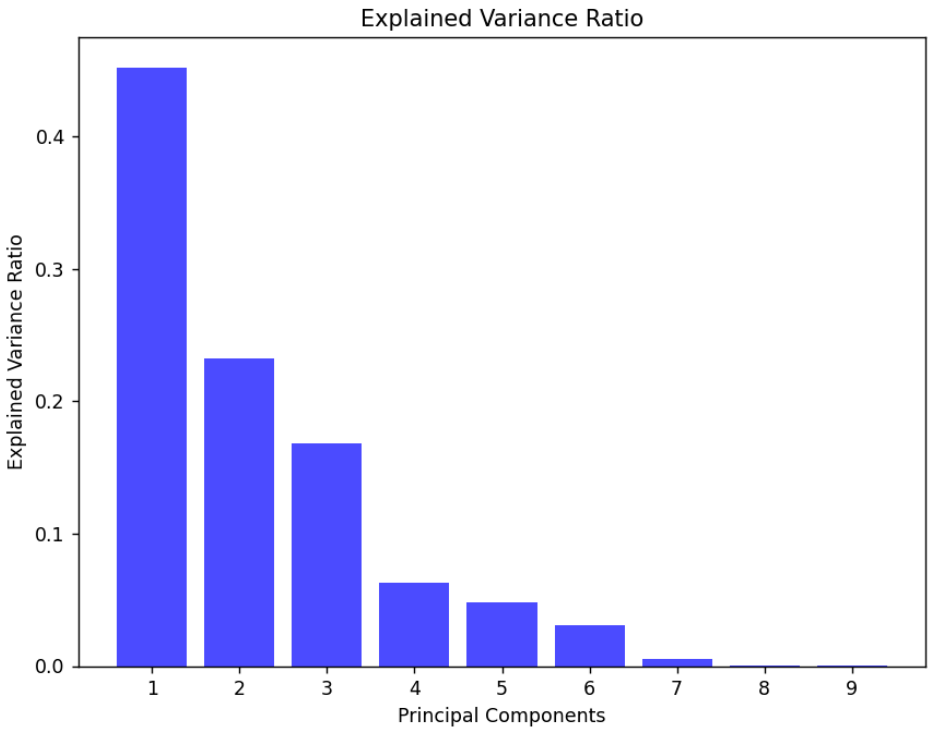


Fig 7: Feature Ranking based on PCA contributions

1	Decay Time	2.6305
2	Rise Time	2.3541
3	Duration	2.2902
4	Kurtosis	2.2542
5	Centroid	2.1430
6	Peak Energy Bin	2.1152
10	Skewness	2.0219
9	Peak Energy In Bin	1.9882
8	Total Energy Released	1.9404

With the features ranked according to their correlation between pulses, a set of first *n* features can be selected to use in clustering to perform a statistically reasonable classification. See *Further classifications using K-Means Algorithm*.

The visualization is accomplished via drawing different kinds of plots showing the clusters in two dimensional space by utilizing matplotlib package. Principal Component Analysis (PCA) was applied to reduce the dimensions of a classification problem involving more than three features, enabling visualization in two dimensions. By leveraging PCA's ability to minimize data loss during dimension reduction, the accuracy of the visualization was optimized.

Classification: To explore potential classes amongst pulses with respect to their features, three clustering algorithms were applied: K-means, DBSCAN, and Hierarchical Clustering. Each of these methods offers different perspectives on the data's structure.

1. **K-means Clustering** K-means clustering with three clusters revealed distinct groupings in the data. The scatter plot of the first two Principal Components colored by K-means labels showed clear separations between the clusters, suggesting that the data may naturally divide into three categories.

Fig 8:K-Means Clustering with respect to PC 1 and PC 2

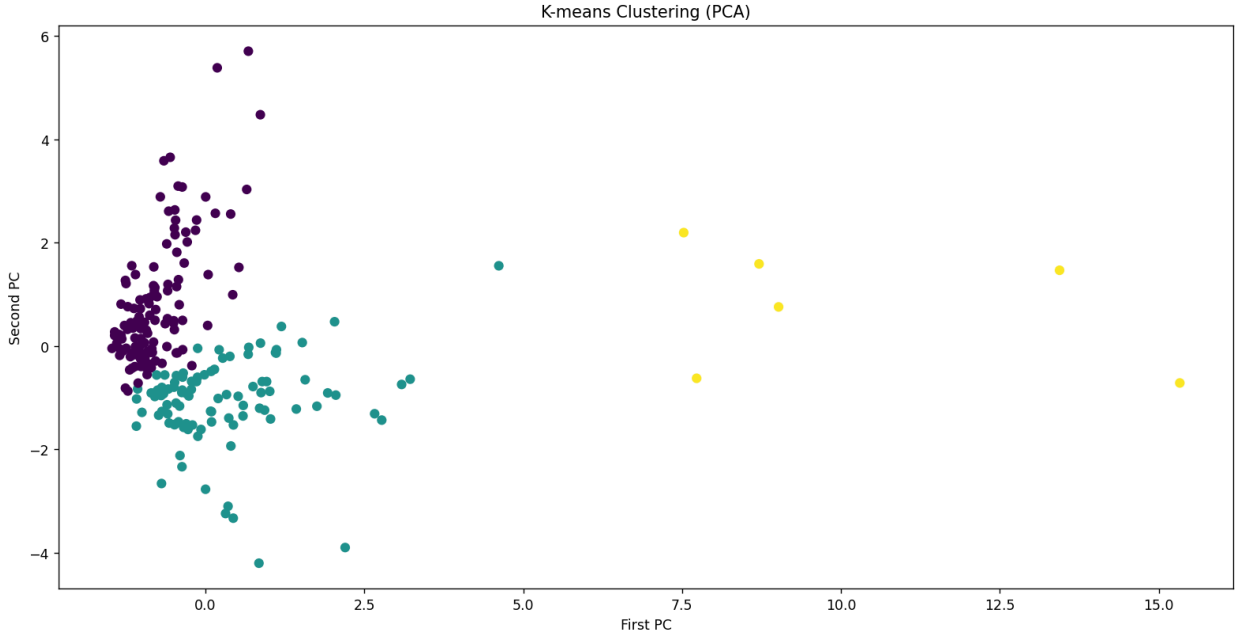
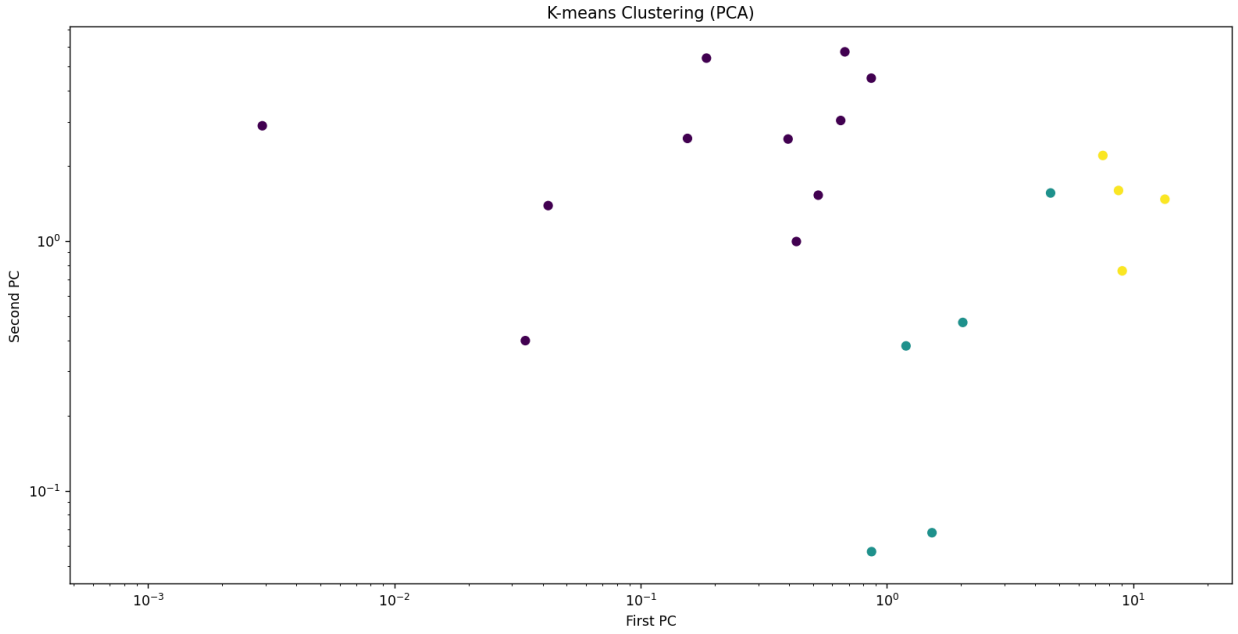


Fig 9:K-Means Clustering with respect to PC 1 and PC 2, values shown as power of ten



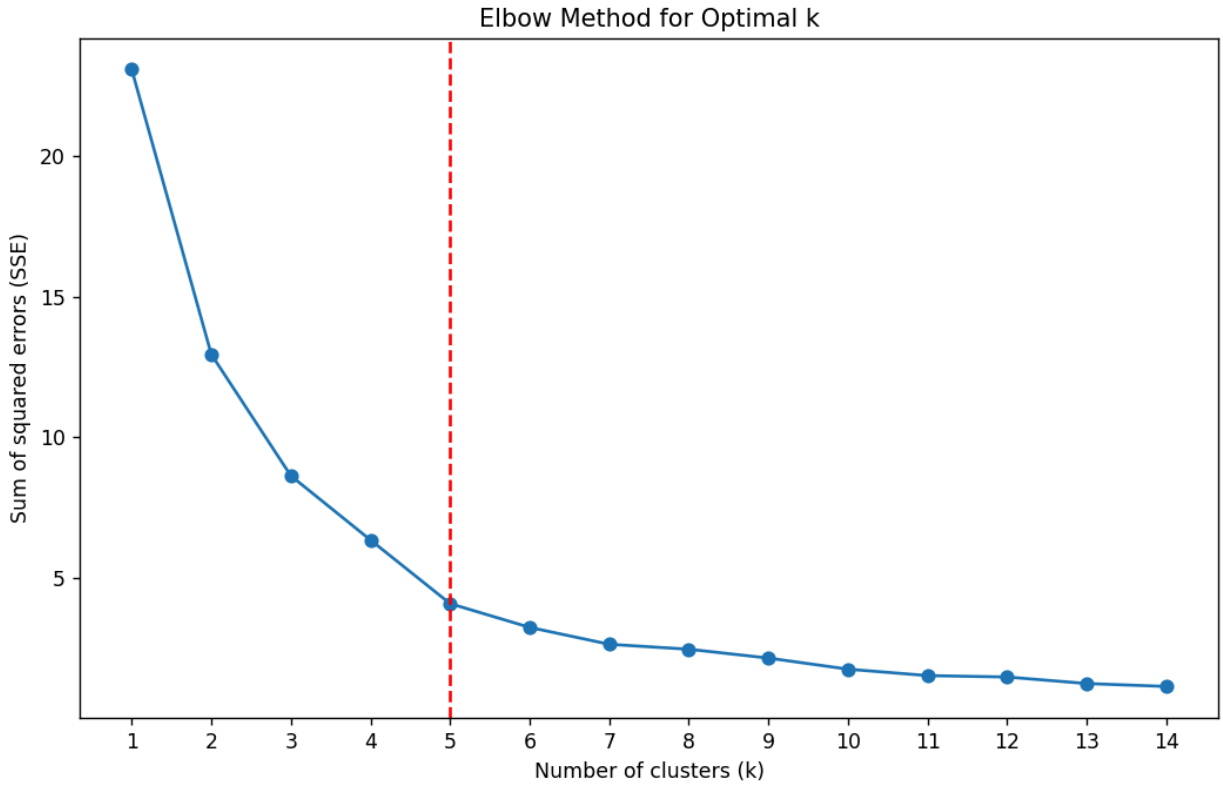
- We use the *elbow method* to decide the optimal K value in the K Means algorithm.
- The elbow method is a visual technique used to determine the optimal number of clusters (*K*) in a *k*-means clustering algorithm. In an elbow plot, the x-axis represents different *K* values, while the y-axis displays the within-cluster sum of squares (WCSS). The optimal *K* is identified at the point where the graph exhibits

a noticeable "elbow," indicating a balance between cluster compactness and model complexity.

- For example, Fig 7 is an elbow plot by clusters of all 221 events with respect to “Duration”, “Peak Energy Bin”, “Skewness”, “Rise Time” and “Total Energy Released” features. By looking at the graph, we can see that *k=5* is an efficient value for cluster numbers. We utilized the *kneed* package in Python to decide the elbow behaviour properly.

Fig 10

Optimal k demonstration of a classification using “Duration”, “Peak Energy Bin”, “Skewness”, “Rise Time” and “Total Energy Released” features.



2. **DBSCAN Clustering** DBSCAN, a density-based clustering algorithm, was used to identify regions of high data density. Unlike K-means, DBSCAN can detect noise and find clusters of arbitrary shapes. The results showed a mixture of dense clusters and noise, indicating that some data points may not belong to any particular cluster.

Fig 11:DBSCAN Clustering with respect to PC 1 and PC 2

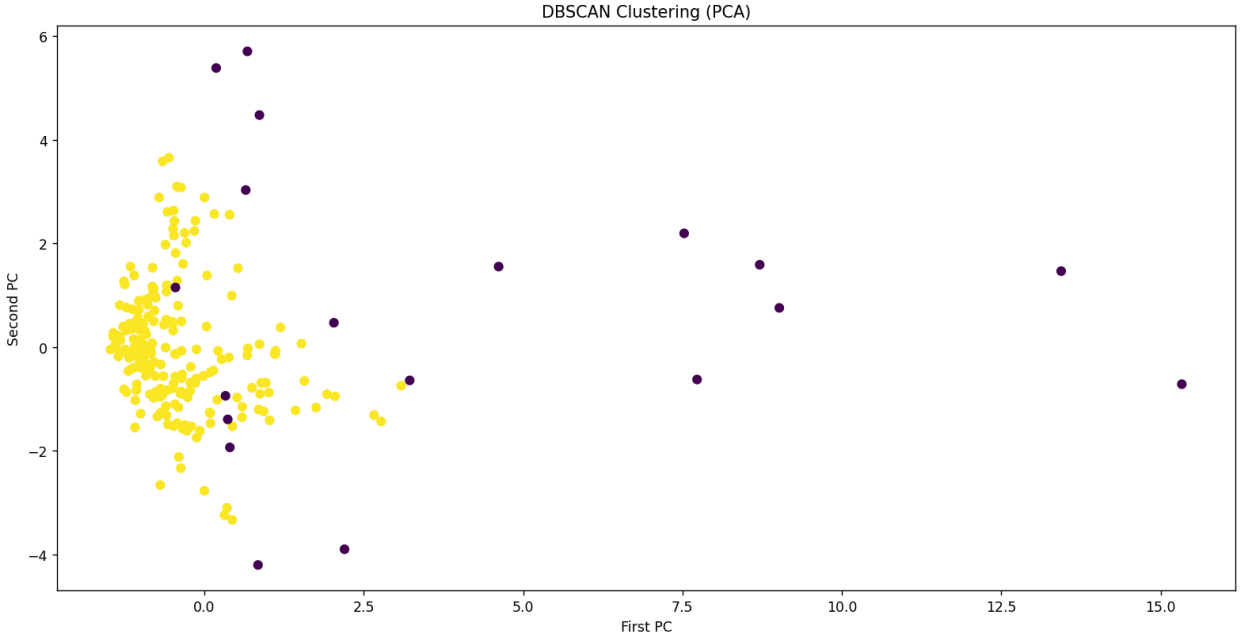
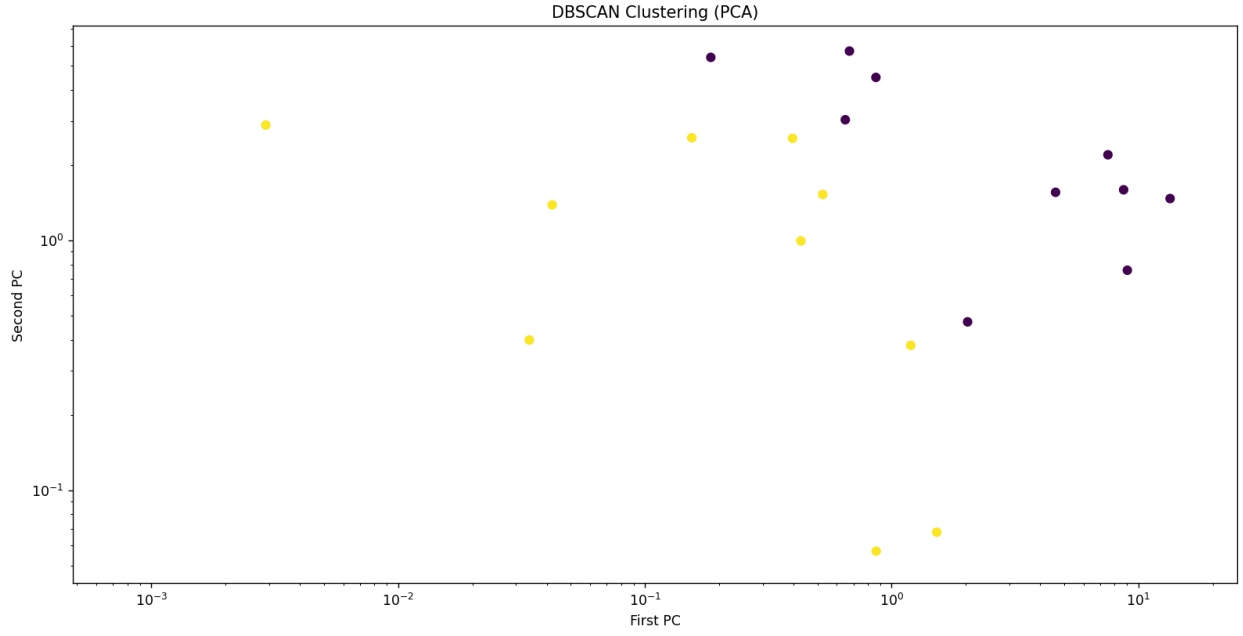
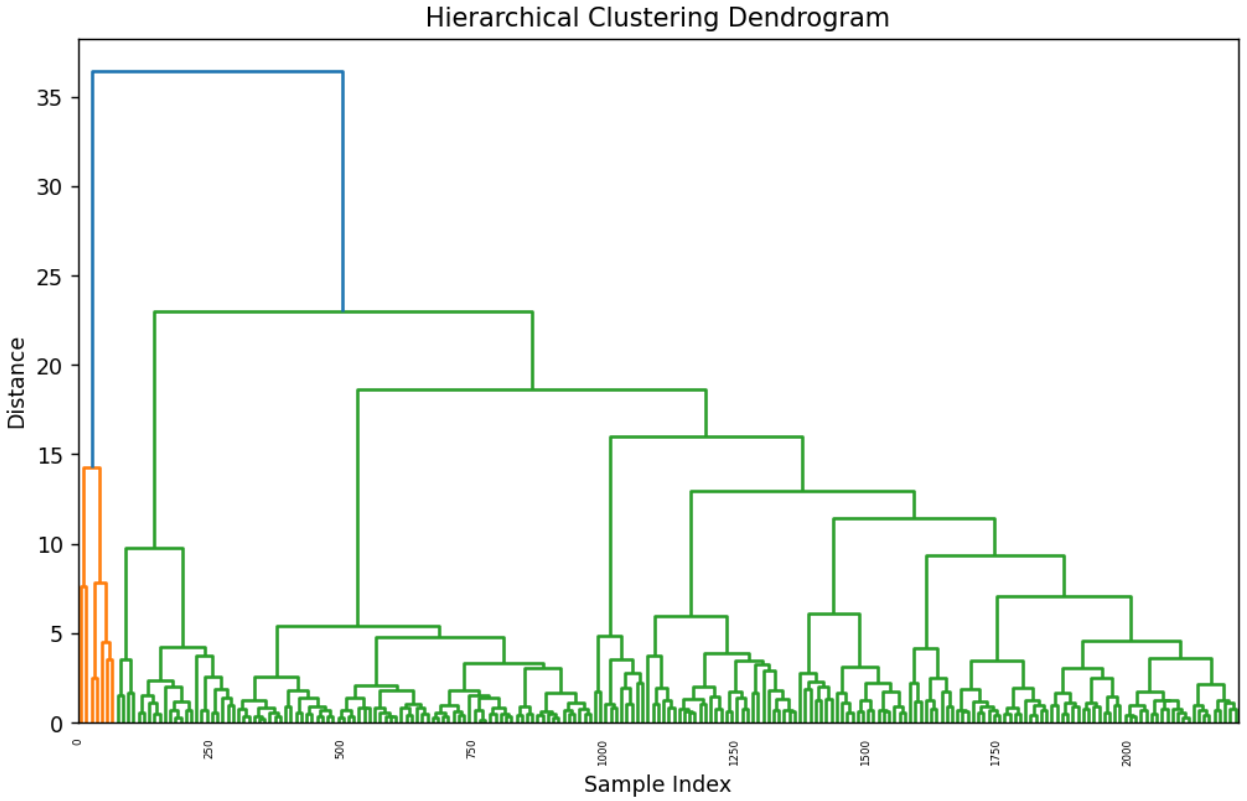
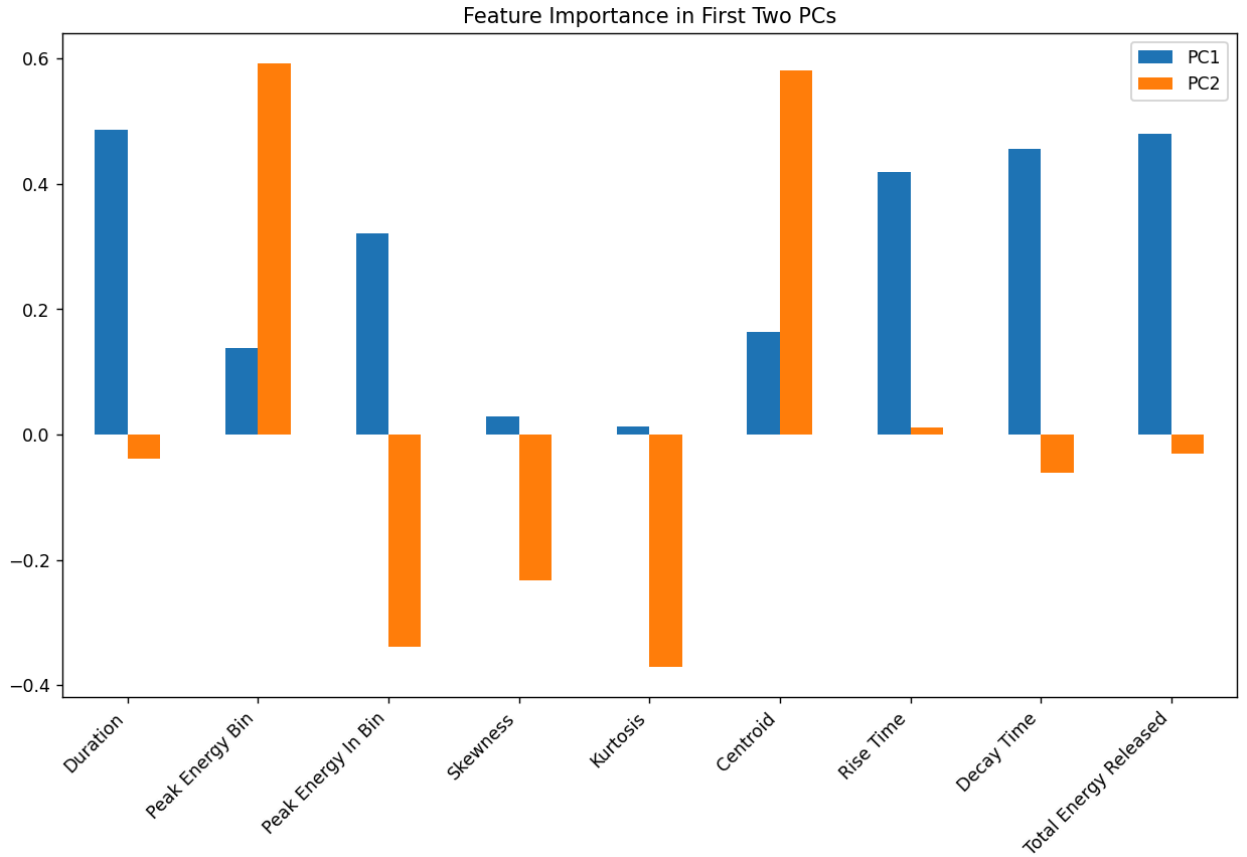


Fig 12: DBSCAN Clustering with respect to PC 1 and PC 2, values shown as power of ten



3. **Hierarchical Clustering** The dendrogram from hierarchical clustering provided a hierarchical view of the data, revealing how the data points are grouped at various levels. This method offers a different perspective compared to K-means and DBSCAN, where the tree structure helps understand the relationships between clusters.

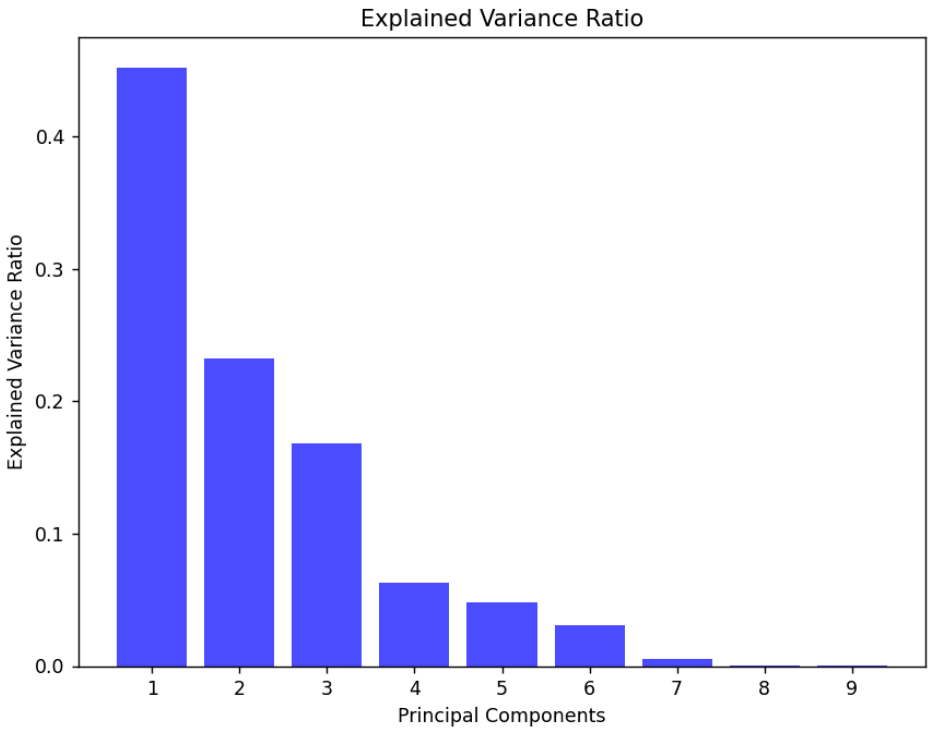

Hierarchical Clustering Dendrogram
Distance
Sample Index
0
5
10
15
20
25
30
35
0
250
500
750
1000
1250
1500
1750
2000


Feature Importance in First Two PCs
PC1
PC2
0.6
0.4
0.2
0.0
−0.2
−0.4
Duration
Peak Energy Bin
Peak Energy In Bin
Skewness
Kurtosis
Centroid
Rise Time
Decay Time
Total Energy Released

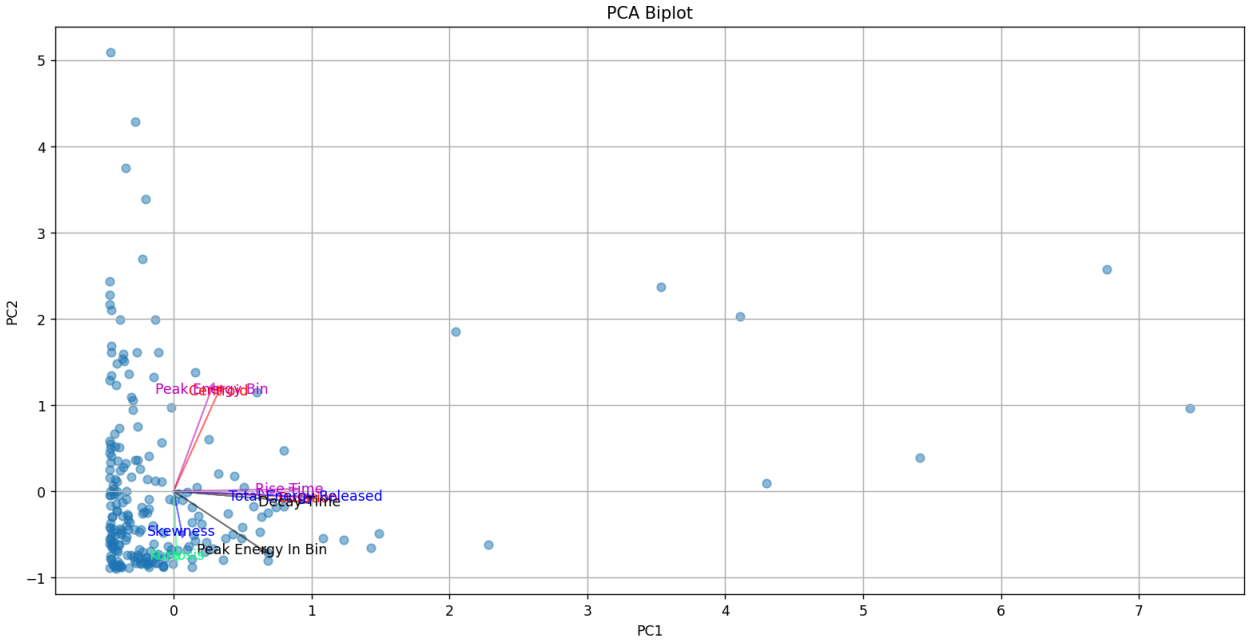
Feature Analysis

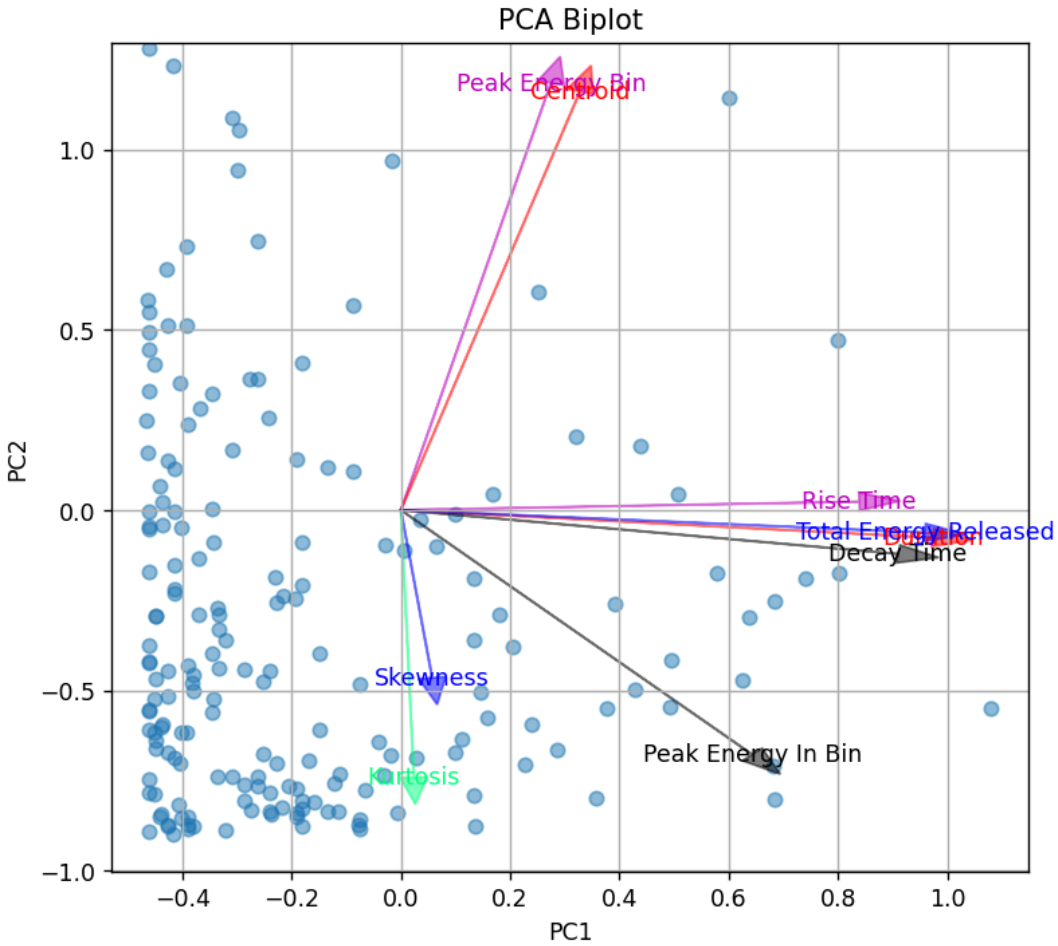
To gain a deeper understanding of the data, several analyses were performed:

1. **Effect of Varying Each Feature on PCA Variance** We investigated the impact of varying individual features on the explained variance in PCA. By systematically altering the value of each feature, we observed how the principal components' variance changes. This analysis helps understand the relative importance of each feature in determining the data's variance.

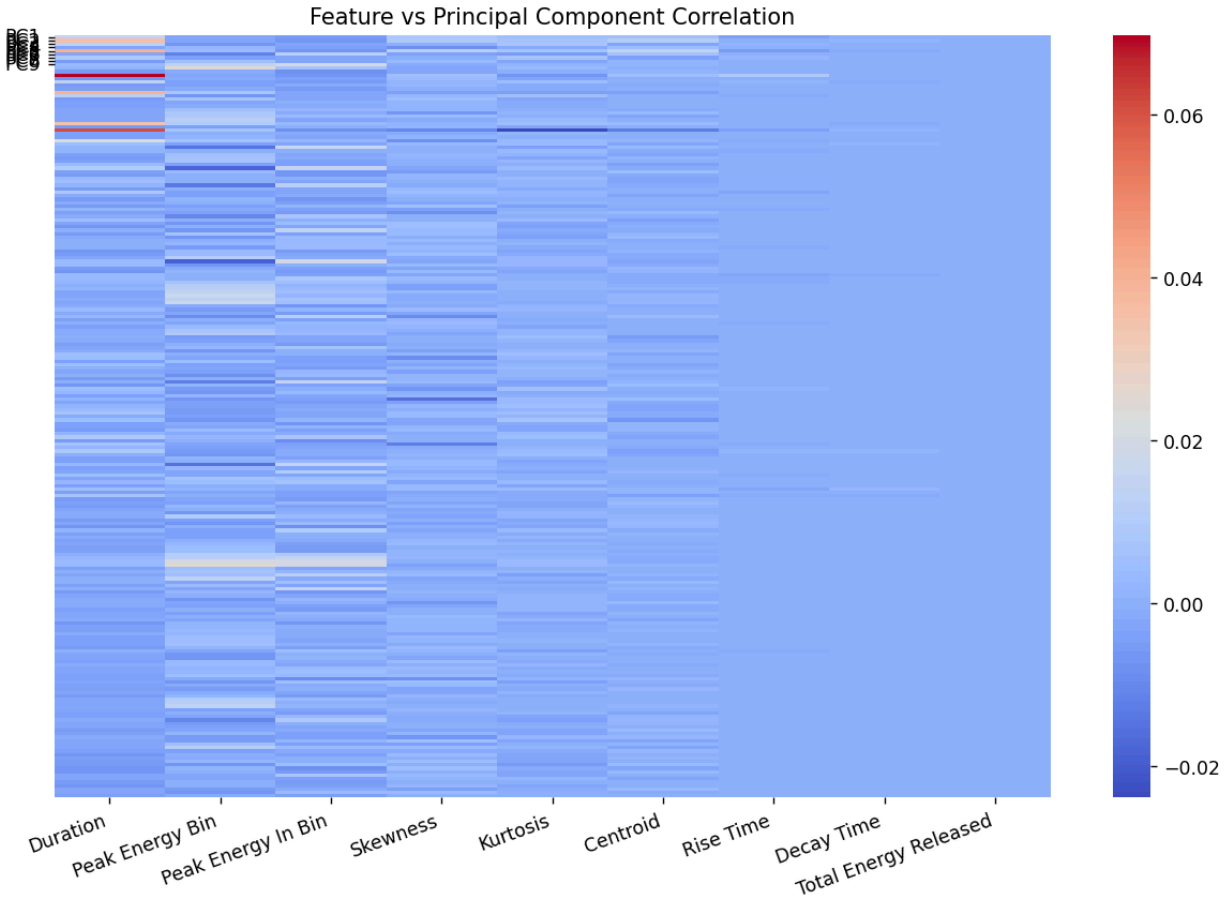


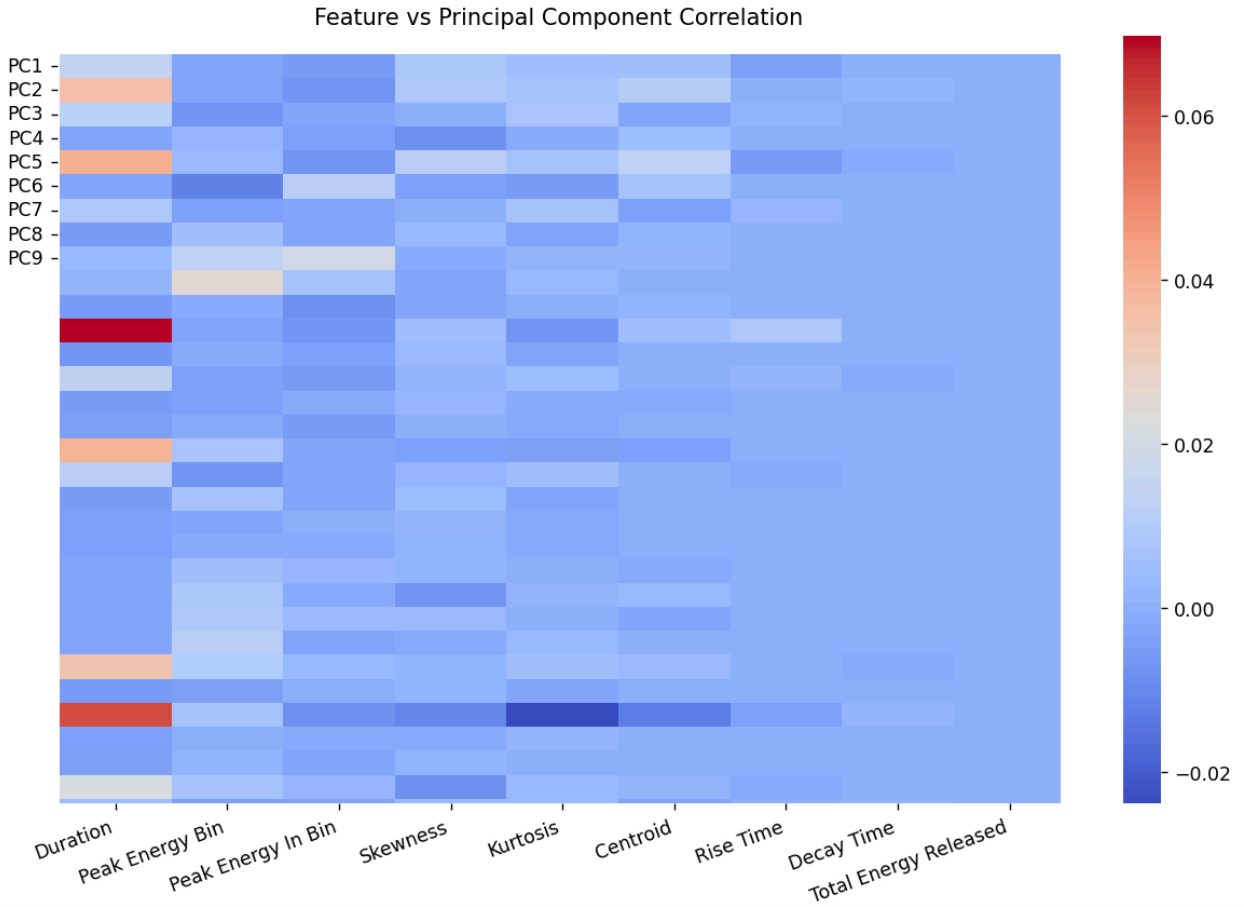
2. **PCA Biplot** A biplot was created to visualize the loadings of the features in the first two principal components. This plot provided insight into the relationships between features and how they contribute to the principal components.


PCA Biplot
PC1
PC2
Peak Energy Bin
Centroid
Rise Time
Total Energy Released
Duration
Decay Time
Skewness
Kurtosis
Peak Energy In Bin


PCA Biplot
PC1
PC2
Peak Energy Bin
Centroid
Rise Time
Total Energy Released
Duration
Decay Time
Skewness
Kurtosis
Peak Energy In Bin

3. **Feature vs. Principal Component Correlation Heatmap** A heatmap was generated to show the correlations between features and principal components. This visualization helped identify which features are most strongly associated with each principal component, providing valuable insight into the underlying data structure.




Feature vs Principal Component Correlation
PC1
PC2
PC3
PC4
PC5
PC6
PC7
PC8
PC9
Duration
Peak Energy Bin
Peak Energy In Bin
Skewness
Kurtosis
Centroid
Rise Time
Decay Time
Total Energy Released
0.06
0.04
0.02
0.00
−0.02

Clustering and Evaluation

The clustering methods were evaluated using **Silhouette Scores**, which measure how similar each point is to its own cluster compared to other clusters. The silhouette score for K-means clustering was found to be higher than that of DBSCAN, suggesting that the K-means clusters are more compact and well-separated. However, DBSCAN was able to identify noise points, which K-means could not.

- **K-means Silhouette Score**: The higher score indicates that K-means has successfully separated the data into meaningful clusters.
- **DBSCAN Silhouette Score**: The lower score suggests that while DBSCAN can identify dense regions, its results are more sensitive to noise.

Results

- **Cluster Sizes**: K-means identified three clusters, while DBSCAN found fewer, with some data points categorized as noise.
- **PCA**: The explained variance plot confirmed that most of the information in the data is captured by the first few principal components.
- **Feature Importance**: The biplot and correlation heatmap revealed which features contribute most significantly to the principal components, helping to interpret the clusters.
- **Silhouette Scores**: K-means performed better in terms of cluster cohesion, but DBSCAN provided additional insights by detecting outliers.

K-means Cluster Sizes	
Cluster Name	Cluster Size
Cluster 0	118
Cluster 1	97
Cluster 2	6

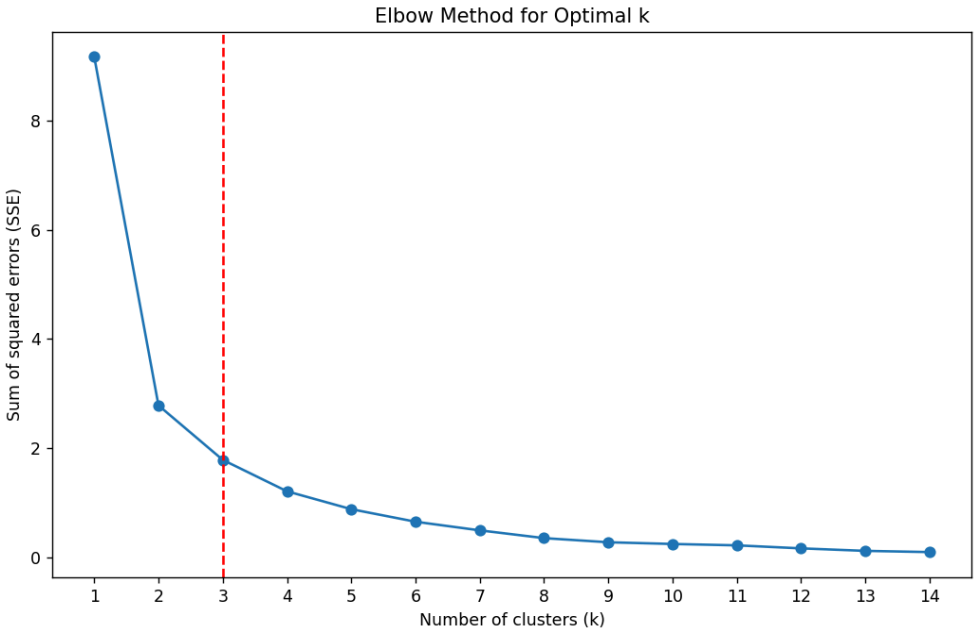
DBSCAN Cluster Sizes	
Cluster Name	Cluster Size
Cluster 0	202
Cluster -1	19

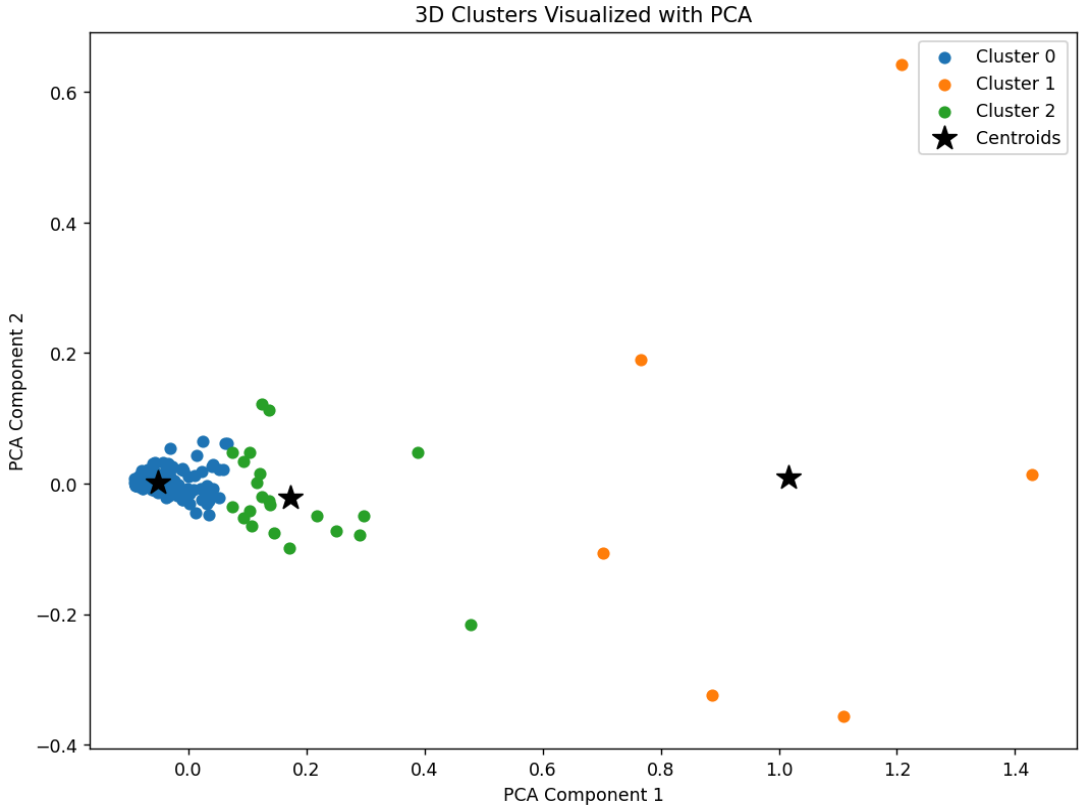
Silhouette Scores	
Name	Silhouette Score
K-means	0.24
DBSCAN	0.57

Further classifications using K-Means Algorithm:

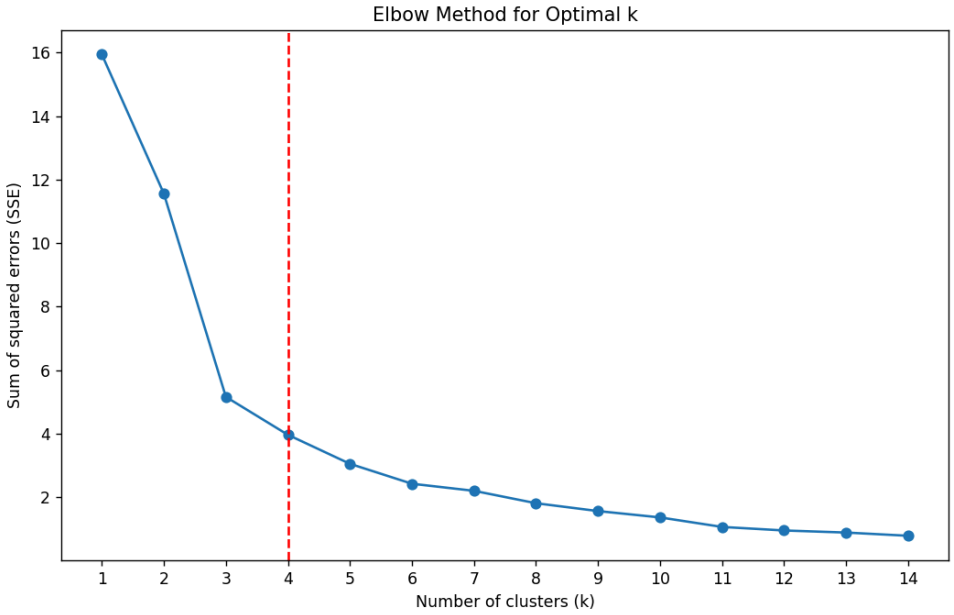
In this section, the results of three classifications has been shown and discussed, each one featuring the top n most influencing set of features. See *Fig 7* for feature ranking.

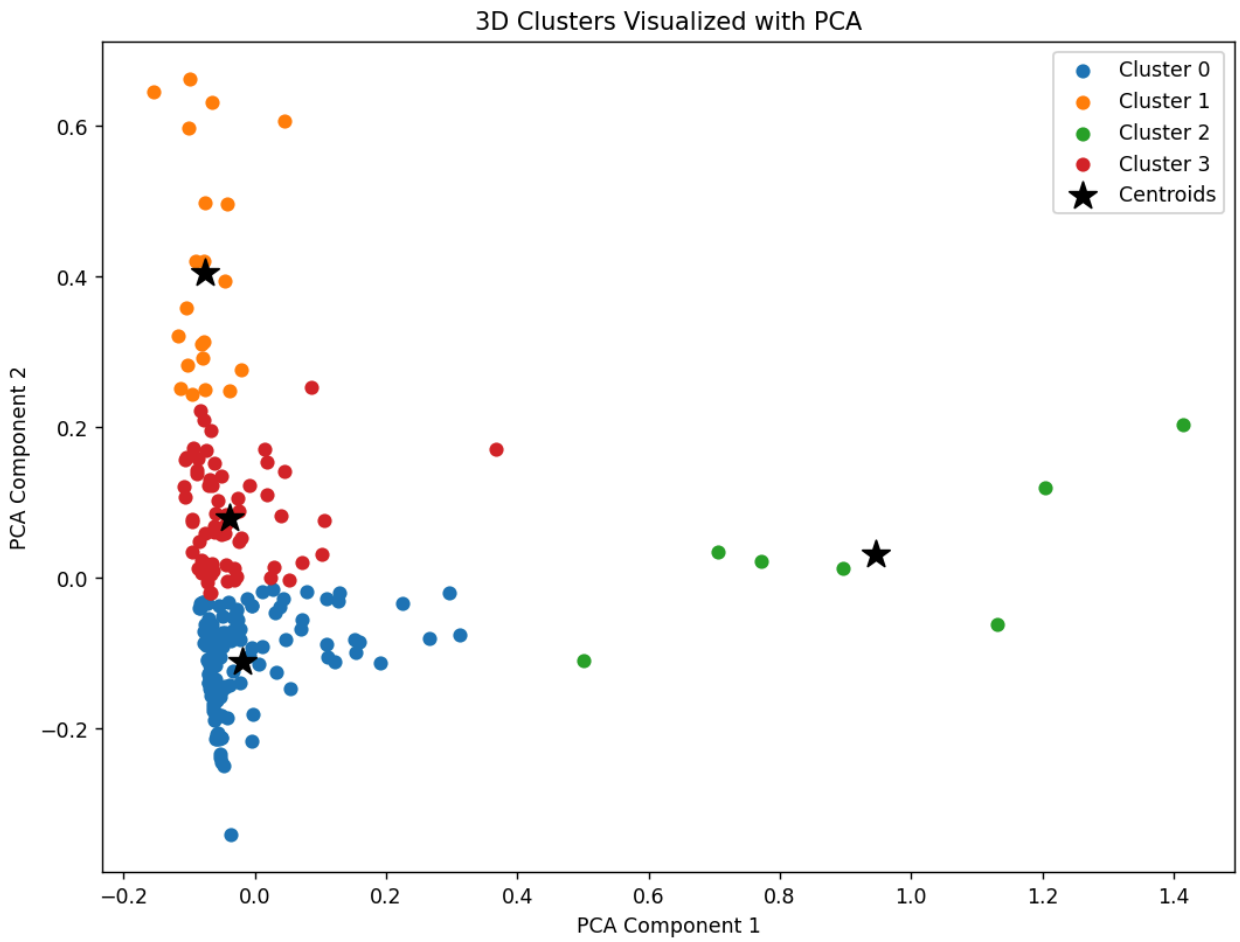
Classification 1: Clustering according to top 3 features: Decay Time, Rise Time, Duration



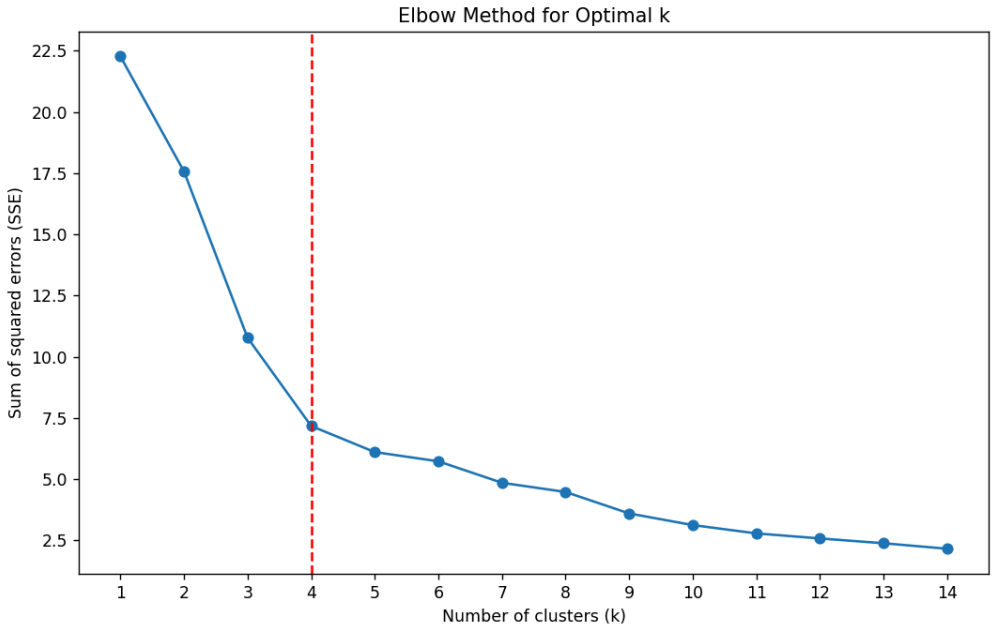


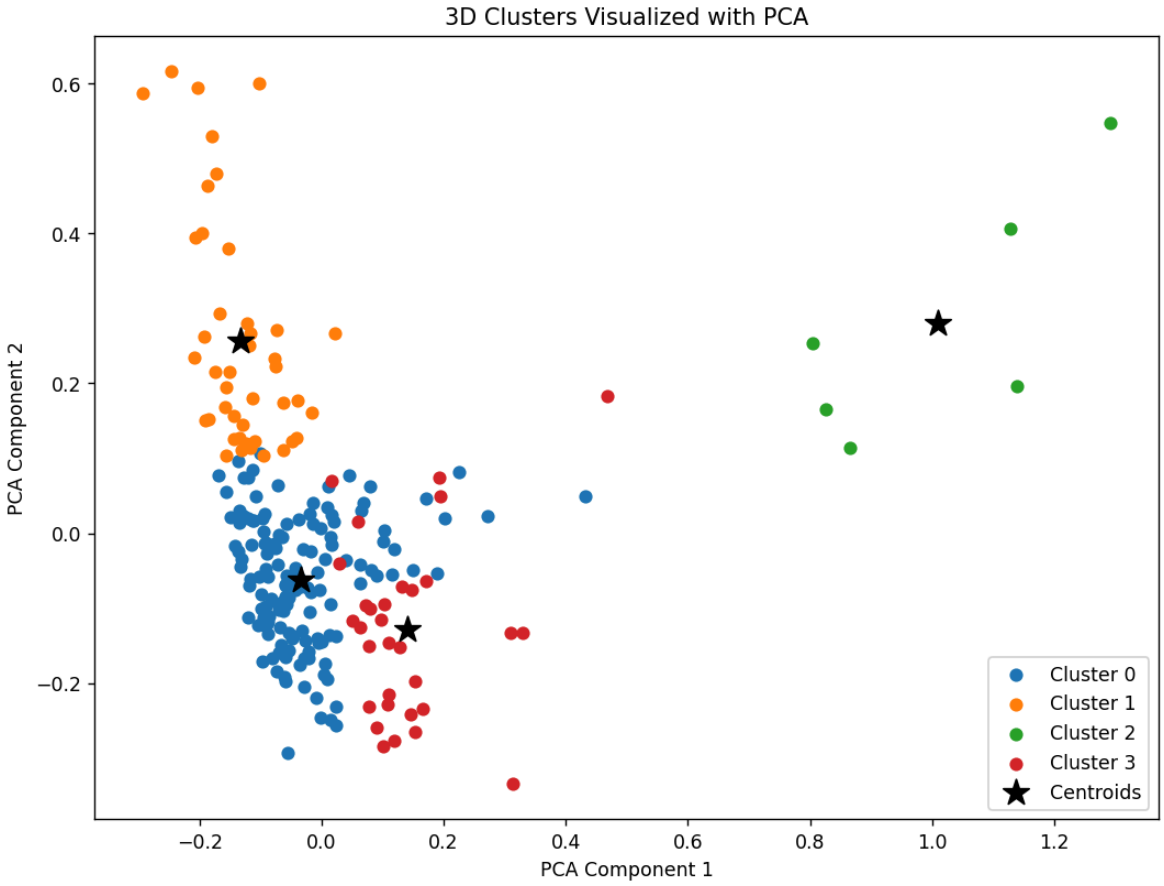
Classification 2: Clustering according to top 4 features: Decay Time, Rise Time, Duration, Kurtosis





Classification 3: Clustering according to top 5 features: Decay Time, Rise Time, Duration, Kurtosis, Centroid





In all Classifications, PC 1 yields a similar distribution whereas PC 2 yields a narrower distribution in Classification 1 and wider distributions in Classification 2 and 3. This reveals the

fact that the Classification 1, which is done by only taking Decay Time, Rise Time and Duration features into account, is providing a sufficient distribution by correctly utilizing PC 1 when doing a general classification amongst magnetar bursts.

Conclusion:

This analysis provided a comprehensive exploration of magnetar burst data using Principal Component Analysis (PCA), clustering techniques, and feature analysis. By employing K-Means, DBSCAN, and Hierarchical Clustering, we identified distinct patterns and groupings within the burst dataset, offering valuable insights into the temporal and energetic characteristics of these high-energy events. PCA played a crucial role in simplifying the dataset by reducing dimensionality while preserving essential information, enabling clearer visualization and interpretation of clustering results.

Among the ten extracted morphological features, **time-based characteristics—specifically Rise Time, Decay Time, and Duration—emerged as the most influential factors** driving the classification and differentiation of magnetar bursts. These features consistently ranked highest in PCA contributions and played a pivotal role in defining the structure of the clusters identified through K-Means analysis.

Contribution to the Field:

This project significantly enhances our understanding of magnetar burst morphology by leveraging high-resolution observational data from the Fermi Gamma-ray Burst Monitor (GBM) and applying advanced data processing techniques. The use of Principal Component Analysis (PCA) allowed for efficient feature selection and dimensionality reduction, highlighting critical features such as Decay Time, Rise Time, and Total Energy Released as key differentiators among burst events. K-Means clustering successfully categorized the bursts into distinct groups, revealing patterns that contribute to a deeper understanding of the physical mechanisms driving magnetar emissions.

The methodologies developed in this project are adaptable and scalable, providing a framework for analyzing other transient astrophysical phenomena, including gamma-ray bursts (GRBs) and fast radio bursts (FRBs). Techniques such as background noise thresholding, feature extraction, and clustering can be directly applied to similar datasets. Additionally, the visualization of high-dimensional data through PCA enables clearer interpretation and identification of hidden patterns, bridging the gap between raw observational data and theoretical astrophysical models.

This study also demonstrates the value of machine learning in astrophysical research, offering a systematic approach to identifying and classifying transient high-energy events with greater precision and insight.

Limitations:

Despite its contributions, the project has certain limitations. Firstly, the reliance on data from only two detectors (main and angularly closest) may limit the completeness of the burst representation, as additional detectors might offer complementary perspectives. Secondly, while PCA effectively reduces dimensionality, the principal components lack direct interpretability, which can obscure the physical meaning of some results. Thirdly, the classification results depend heavily on the choice of features included in the clustering process, and slight variations in feature selection may impact the outcomes. Lastly, the K-Means Clustering algorithm assumes spherical clusters of equal size, which might not perfectly align with the natural clustering structure of the data.

Addressing these limitations in future studies—such as incorporating data from more detectors, exploring alternative clustering algorithms like DBSCAN, or refining the feature selection process—could further enhance the robustness and accuracy of the analysis.

References:

1. NASA. (n.d.). *Fermi Gamma-ray Space Telescope overview*. Retrieved from https://www.nasa.gov/mission_pages/GLAST/overview/index.html
   Fermi GBM Team. (n.d.). *Fermi Gamma-ray Burst Monitor (GBM) homepage*. Retrieved from https://heasarc.gsfc.nasa.gov/W3Browse/fermi/fermigbrst.html
2. Wikipedia contributors. (2023). *Magnetar*. In *Wikipedia, The Free Encyclopedia*. Retrieved from https://en.wikipedia.org/wiki/Magnetar
3. Wikipedia contributors. (2023). *Gamma-ray burst*. In *Wikipedia, The Free Encyclopedia*. Retrieved from https://en.wikipedia.org/wiki/Gamma-ray_burst
4. Rea, N., & Esposito, P. (2011). Magnetar outbursts. In *The Astronomy and Astrophysics Review*, 19(1), 9-25. https://doi.org/10.1007/s00159-010-0021-5
5. Cavanagh, B., et al. (2021). An introduction to machine learning in astronomy. *Publications of the Astronomical Society of the Pacific*, 133(1021), 075001. https://doi.org/10.1088/1538-3873/abf5a9
6. Goodfellow, I., Bengio, Y., & Courville, A. (2016). *Deep learning*. MIT Press. Retrieved from https://www.deeplearningbook.org
7. Cortes, C., & Vapnik, V. (1995). Support-vector networks. *Machine Learning*, 20(3), 273-297. https://doi.org/10.1007/BF00994018
   Schölkopf, B., & Smola, A. J. (2002). *Learning with kernels: Support vector machines, regularization, optimization, and beyond*. MIT Press.
8. Breiman, L. (2001). Random forests. *Machine Learning*, 45(1), 5-32. https://doi.org/10.1023/A:1010933404324
   Quinlan, J. R. (1986). Induction of decision trees. *Machine Learning*, 1(1), 81-106. https://doi.org/10.1007/BF00116251
9. Nair, V., & Hinton, G. E. (2010). Rectified linear units improve restricted Boltzmann machines. In *Proceedings of the 27th International Conference on Machine Learning* (ICML-10) (pp. 807-814). Retrieved from http://www.icml-2010.org/papers/432.pdf
   Dhingra, B., et al. (2017). A framework for improving the convergence of deep learning models. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing* (pp. 1065-1075). https://doi.org/10.18653/v1/D17-1124
   Zhang, Y., et al. (2019). Softmax activation function in neural networks: A practical guide. *IEEE Access*, 7, 98827-98839. https://doi.org/10.1109/ACCESS.2019.2921382
10. NASA Goddard Space Flight Center. Retrieved from https://www.nasa.gov/goddard
    Fermi Science Support Center. Retrieved from https://fermi.gsfc.nasa.gov/ssc/
11. SciPy. (n.d.). SciPy: Scientific Library for Python. Retrieved from https://www.scipy.org/
12. PyTorch. (n.d.). PyTorch: An open source machine learning framework. Retrieved from https://pytorch.org/
13. Matplotlib. (n.d.). Matplotlib: Visualization with Python. Retrieved from https://matplotlib.org/
14. Pandas. (n.d.). Pandas: Powerful data structures for data analysis in Python. Retrieved from https://pandas.pydata.org/
15. Scikit Learn. (n.d.). Scikit Learn: https://scikit-learn.org/0.21/documentation.html
16. https://www.youtube.com/watch?v=_YPScrckx28
17. https://www.youtube.com/watch?v=FD4DeN81ODY