Data Science Minor Project

(Area Code: 01)

Classification of X-ray transient events using Machine Learning techniques

A Dissertation Submitted

in Partial Fulfilment of the Requirements

for the Degree of

Minor Degree

in

Data Science

by

Shreya Umesh Prabhu

(Roll No. IMS19211)

to

SCHOOL OF DATA SCIENCE

INDIAN INSTITUTE OF SCIENCE EDUCATION AND RESEARCH

THIRUVANANTHAPURAM - 695 551, INDIA

April 2023

DECLARATION

I, **Shreya Umesh Prabhu (Roll No: IMS19211)**, hereby declare that, this report entitled **Classification of X-ray transient events using Machine Learning techniques** submitted to Indian Institute of Science Education and Research Thiruvananthapuram towards the partial requirement of **Minor degree** in **Data Science**, is an original work carried out by me under the supervision of **Shabnam Iyyani** and has not formed the basis for the award of any degree or diploma, in this or any other institution or university. I have sincerely tried to uphold academic ethics and honesty. Whenever a piece of external information or statement or result is used then, that has been duly acknowledged and cited.

Thiruvananthapuram - 695 551 **Shreya Umesh Prabhu**

April 2023

CERTIFICATE

This is to certify that the work contained in this project report entitled **"Classification of X-ray transient events using Machine Learning techniques"** submitted by **Shreya Umesh Prabhu** (**Roll No: IMS19211**) to Indian Institute of Science Education and Research, Thiruvananthapuram towards the partial requirement of **Minor** in **Data Science** has been carried out by her under my supervision and that it has not been submitted elsewhere for the award of any degree.

Thiruvananthapuram - 695 551 — Shabnam Iyyani

April 2023 — Project Supervisor

ACKNOWLEDGEMENT

I would like to express my deepest gratitude to all those who have contributed to the successful completion of this data science project. I would like to first express my profound gratitude and deep regard to my project guide Dr. Shabnam Iyyani, IISER Thiruvananthapuram for providing invaluable guidance, encouragement, and support throughout the project. Her expertise, constructive feedback, and insightful suggestions have been instrumental in shaping the direction of the project and bringing it to fruition.

I would also like to acknowledge the support of my colleagues and friends, who provided valuable feedback and moral support throughout the project. Finally, I am grateful to the Indian Institute of Science Education and Research Thiruvananthapuram for providing the necessary resources and facilities to complete this project to the best of my ability.

Thiruvananthapuram - 695 551 **Shreya Umesh Prabhu**

April 2023

ABSTRACT

Name of the student: **Shreya Umesh Prabhu** Roll No: **IMS19211**

Degree for which submitted: **Minor** Department: **School of Data Science**

Thesis title: **Classification of X-ray transient events using Machine Learning techniques**

Thesis supervisor: **Dr. Shabnam Iyyani**

Date of thesis submission: **April 2023**

The Fermi Gamma-ray Burst Monitor (GBM) is an all-sky gamma-ray monitor that detects different types of hard X-ray transients such as gamma ray burst (GRB), terrestrial gamma ray flashes (TGF), solar flares (SF), short gamma ray repeaters (SGR) etc. The light curves of each class of triggers show different features which can be used to identify the trigger type. This project aims to develop an algorithm based on supervised machine learning technique to classify different types of transient triggers using their light curves which are time series data. In this direction, the current algorithm developed in the project is trained for identifying GRBs and TGFs using the convolutional neural network for the supervised clustering.

Keywords:

Convolutional Neural Network, supervised classification, X-ray transients

Contents

List of Figures

List of Tables

Introduction

Transients refer to astronomical phenomena with durations of fractions of a second to weeks or years. Typically they are extreme, short-lived events associated with the total or partial destruction of an astrophysical object. X-ray transients detected can be classified into different types. Gamma-ray bursts (GRBs) are the most powerful cosmic explosions for electromagnetic output. They are likely produced by the collapse of massive stars into black holes or by the coalescence of two neutron stars. Soft gamma-ray repeaters (SGRs) are X-ray sources believed to be powered by magnetars, that is, by neutron stars with the strongest magnetic fields in the Universe. Terrestrial gamma-ray flashes (TGFs) are kinds of high-energy emissions produced during thunderstorms and are almost exclusively detected with spacecraft-based gamma-ray detectors. Solar flares result from magnetic reconnections occurring in the solar corona.

The Fermi gamma ray space telescope launched in 2008 comprises of two instruments: the Gamma-Ray Burst Monitor and the Large Area Telescope. The prime objective of GBM is spectral and timing analysis of GRBs. It can observe the whole unocculted sky at any given time with energy coverage from 8 keV to 40 MeV. Therefore, GBM offers excellent capabilities to observe all kinds of high-energy as-

trophysical phenomena mentioned above. These transient event types are listed in Table 1.

GBM is composed of 12 thallium sodium iodide (NaI(Tl)) scintillation detectors which are sensitive from 8 keV to 1 MeV, and two bismuth germanate (BGO) detectors, sensitive from 200 keV to 40 MeV. GBM continuously observes the whole unocculted sky, with its flight software (FSW) constantly monitoring the count rates recorded in the various detectors. For GBM to trigger on a GRB or any other high-energy transient, two or more detectors must have a statistically significant increase in the count rate above the background rate. The measured data is all-sky light curves with sources superimposed upon each other as a detector scans the sky. Thus, in any analysis of temporal transients such as GRBs, GBM relies on the ability to separate source and background during any scan(s).

Once the trigger algorithm identifies the trigger in the data, it is classified into different trigger types using the on-board trigger classification algorithm. The GBM flight software includes an algorithm to classify triggers. The probability that the trigger event is a GRB, as opposed to a solar flare, SGR, particle precipitation event, or known transient source, is calculated using a Bayesian approach that considers the event localization, spectral hardness, and the spacecraft geomagnetic latitude. These triggers are entered in the Fermi GBM trigger Catalog.

The main objective of this project is to use the light curves of different triggers to classify them into different classes. Light curves give the information regarding how the number of photon counts varies with time during the event, thereby making it a type of time series data. Here, the Fermi GBM trigger catalog data is taken for plotting light curves. A simple supervised learning algorithm is implemented

to classify this time series data. The algorithm tries to identify the features of light curves of different triggers in different binning. A simple model using a convolutional neural network is built to classify two classes of triggers - GRBs and TGFs.

1 Data for light curves of transients

In this project, we use only the data from the brightest of the 12 NaI detectors. The data files were obtained from the Fermi GBM Trigger Catalog. The data files used in this project include a trigger data file (.fit extension) used to find the brightest detector and a Time-Tagged Event (TTE) data file used to plot the light curves. The TTE data is a time series of counts, where each count is mapped to an energy channel. Here TTE consists of individual events, each tagged with arrival time (2 µs resolution), energy (128 channels), and detector number. The TTE data of the brightest detector is used to plot the light curve of the respective trigger.

Light curves are graphs that show the brightness of an object in terms of number of photons as a function of time. Light curves for different triggers are obtained by plotting histograms of the time data present in the TTE file. The features visible in these light curves vary depending on the trigger type. A few light curves for random triggers were plotted to visualize these features (Fig. 1).

Fig.2 and 3 show the light curves of GRB and TGF at different bin sizes. At 1 sec binning in GRB, the peak in the range of a few seconds can be seen. As the number of bins increases, the peak becomes less significant. For TGF, a significant peak at trigger time can be seen at a bin size of 0.1 sec and less. Hence the signal-to-noise ratio at each bin size varies for a single trigger. The different classes of triggers will

DISTPAR	Distance particle event
GALBIN	Galactic binary
GRB	Gamma-ray burst
LOCLPAR	Local particles
SFLARE	Solar flare
SGR	Soft gamma repeater
TGF	Terrestrial Gamma-Ray Flash
TRANSNT	Generic transient
UNCERT	Uncertain classification

Table 1: Classes of triggers identified in the Fermi GBM Catalog

have different features in light curves depending on the bin size. These differences in the light curve at different bin sizes can be used to classify the trigger. In this project, this fact is used to build up a data set and thereby classify only two classes of triggers (GRBs and TGFs).

2 Data Preparation

The current algorithm aimed at classifying only two types of triggers - GRBs and TGFs. A uniform data set of these two triggers was generated by choosing 50 TGFs at random from the trigger catalog. For GRBs, another 50 random triggers were chosen. The fluence (flux integrated over the burst duration) for GRBs was obtained from the Fermi GBM Burst Catalog. The fluence of all GRBs follows a log-normal distribution (Fig.4). It was ensured that the random sample of 50 from all GRB populations also had log-normal distribution by plotting the histogram.

The TTE data is then processed to give histogram counts for different bin sizes. In accordance with the light curves seen in Fig.2 and 3, the chosen bin sizes for

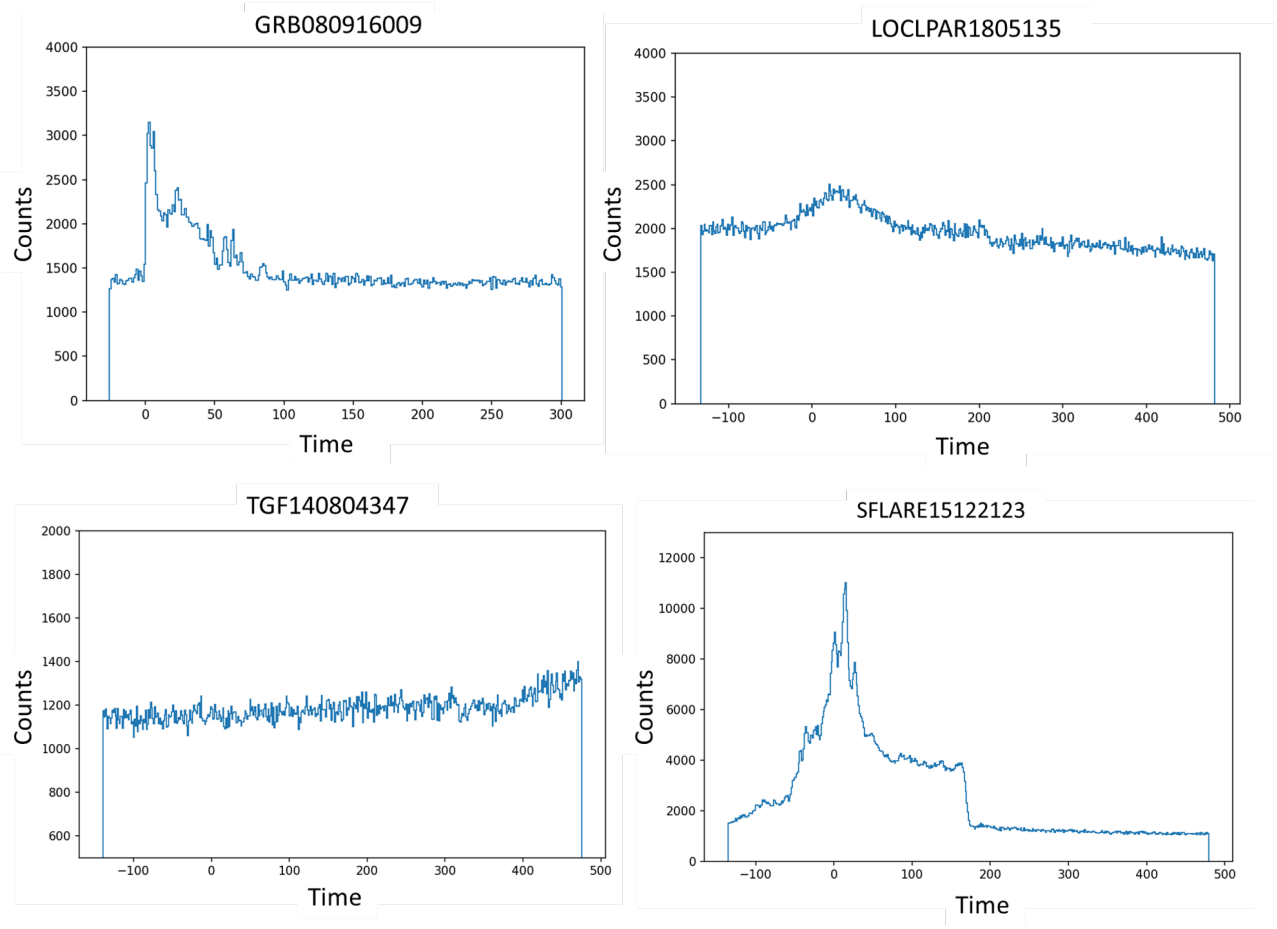

Figure 1: Light curves for different triggers at 1 second binning. (These are plotted for the brightest detector for particular trigger)

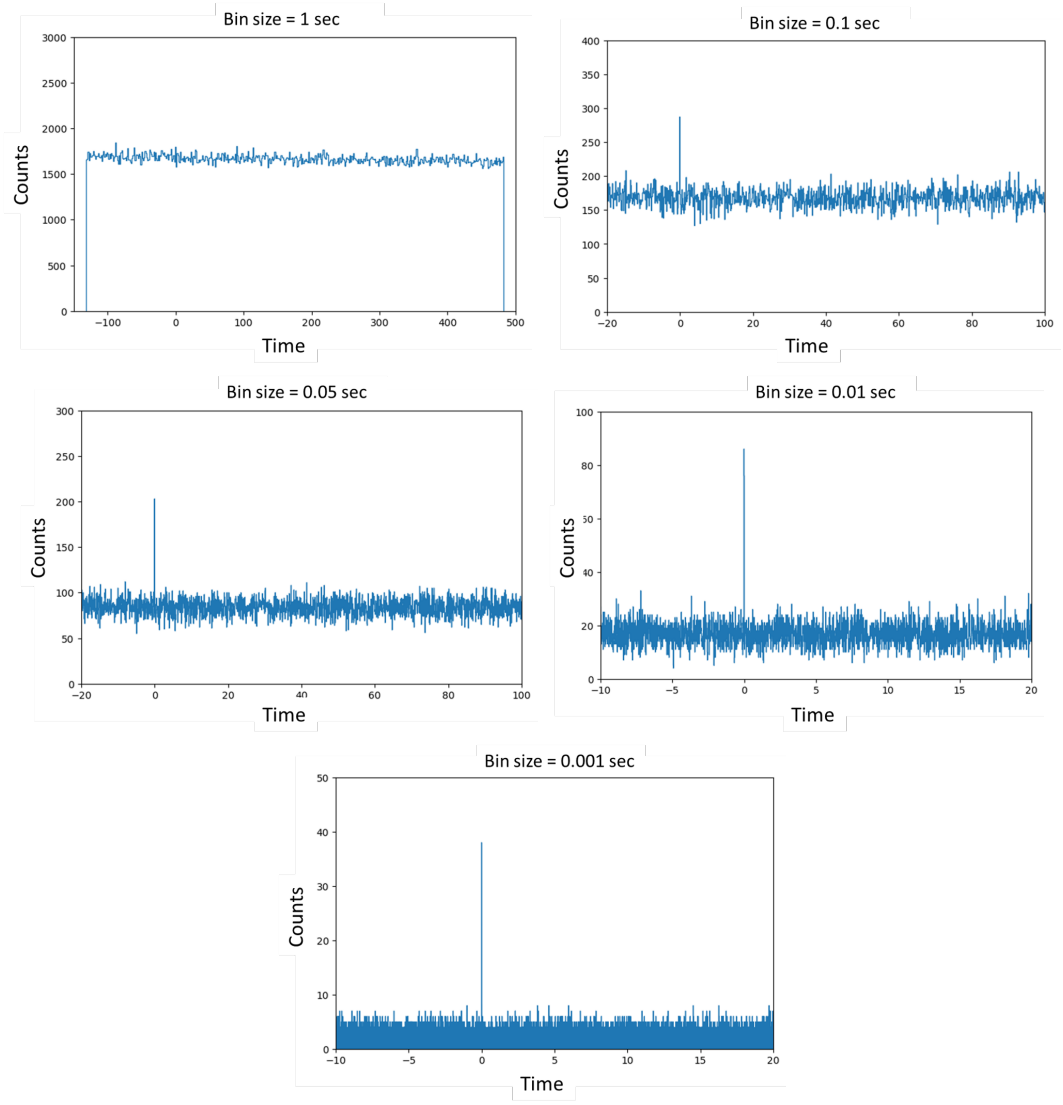

Figure 2: Light curves for the trigger TGF201025784 at bin sizes of 1 sec, 0.1 sec, 0.05 sec, 0.01 sec, and 0.001 sec. (These are plotted for the na detector which is the brightest detector for this trigger)

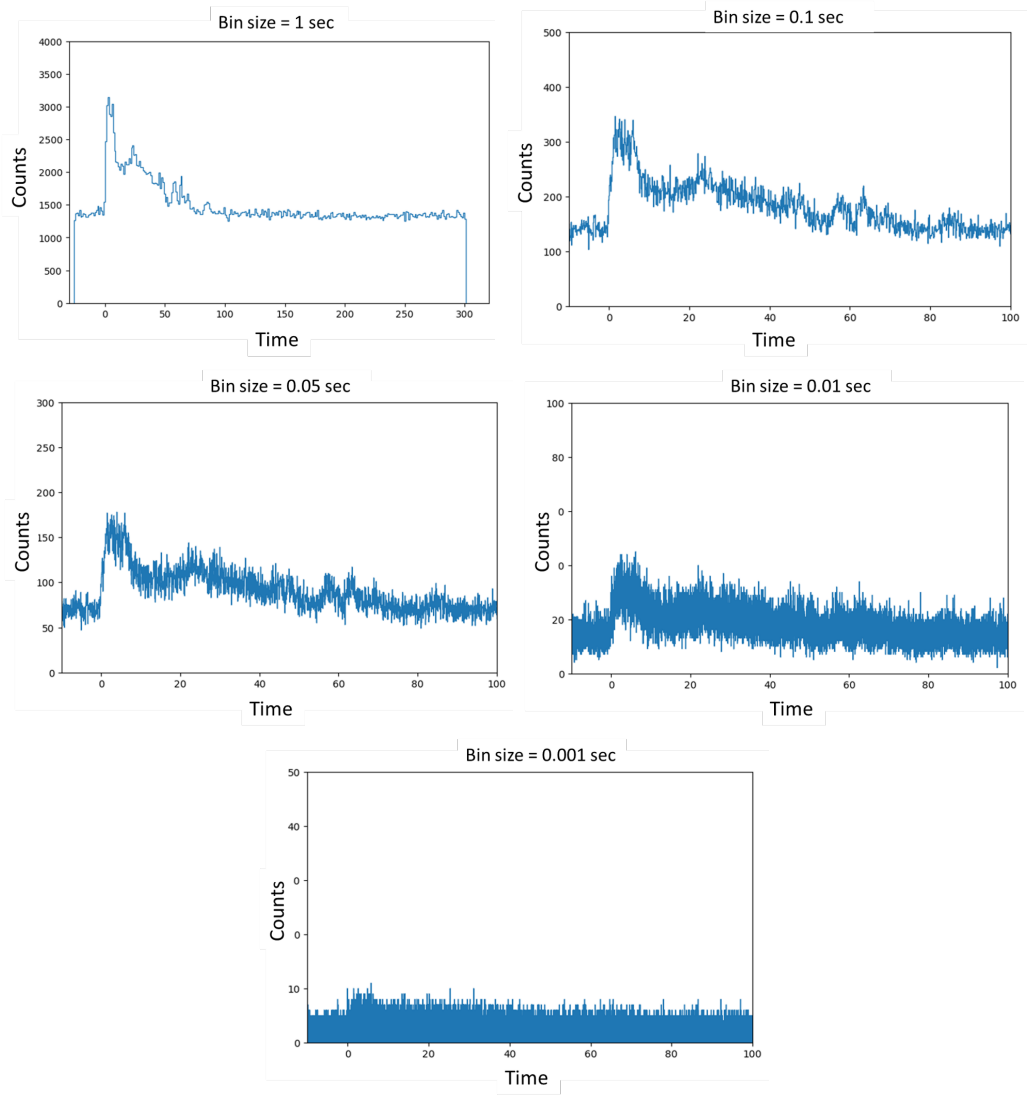

Figure 3: Light curves for the trigger GRB080916009 at bin sizes of 1 sec, 0.1 sec, 0.05 sec, 0.01 sec, and 0.001 sec. (These are plotted for the n3 detector which is the brightest detector for this trigger)

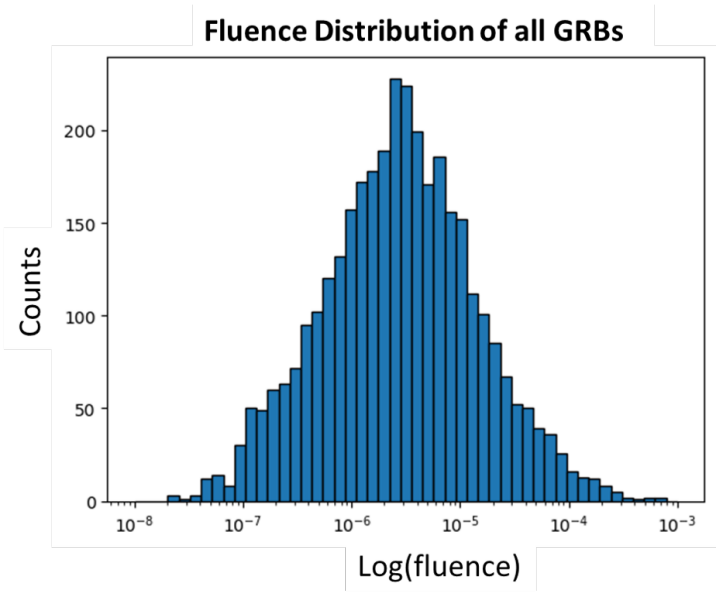

Figure 4: Distribution of Fluence of all GRBs in Fermi GBM Catalog

comparing light curves are 1 sec, 0.1 sec, 0.05 sec, 0.01 sec, and 0.001 sec. These bin sizes are chosen such that the light curves of both classes vary significantly over them.

The histogram counts from -10 to 100 sec of different bins are put in a single matrix for a trigger. The data set input for the algorithm has 2-D matrices for each trigger, with a label of either 0 for GRB or 1 for TFG. These matrices are used to train the model and predict the class for testing data. The algorithms used for classification are discussed in the next section.

3 Algorithms for classification

Time series is an important class of temporal data objects that are obtained by recording a series of observations chronologically. Classification of time series data is a challenging problem due to the nature of time series data: high dimensionality, large data size, and more. There are many approaches to time series classification, two of which were implemented here to classify the light curves of the different triggers.

The classification algorithm takes the input of 5 times series data (for each of 5 decided bin sizes) for a single trigger. The model is fit using the training data set consisting of 70 triggers (35 GRBs and 35 TGFs) with the labels. The testing dataset contains 30 triggers with labels. The predicted classes are compared with the labels of the testing dataset to find the accuracy.

3.1 Distance-Based Classification (Dynamic Time warping)

The distance-Based Classification algorithm defines a distance function to measure the similarity between 2-time series. Some known algorithms, like K nearest neighbor (KNN), and support vector machines (SVM), can be used for the classification of time series data. The major drawback of this method is that the series must be of equal size. This is overcome by adapting such algorithms for time series by replacing the Euclidean distance metric with the Dynamic Time Warping (DTW) metric. However this can computationally expensive.

The DTW algorithm was implemented for the classification of light curves at five different bin sizes. The algorithm takes up a long computation time at smaller bin sizes due to large array sizes.

3.2 Convolution Neural Network

Convolution Neural Network (CNN) is widely applied for sequential data such as speech or audio. CNNs take advantage of the basic idea behind the convolution sum to perform locality modeling over high-dimensional data. CNNs can focus more on the local features in high-dimensional data. CNN can automatically discover and extract the internal structure of the input time series to generate deep features for classification.

Here a simple convolution neural network was defined consisting of a single convolutional layer, a max pooling layer, a flatten layer, and a single dense layer with a sigmoid activation function. This model is compiled with the Adam optimizer and binary cross-entropy loss and trained on the training data using a batch size of 4 and 35 epochs. This a supervised classification; the training and testing data has labels either 0 (GRB) or 1(TGF). The model is then tested on the testing data set, and accuracy is determined.

3.3 Layers in the CNN model

Input layer

The input layer has 5 time series (histogram counts at each of 5 chosen bin sizes) in the form of a matrix for a single trigger.

single convolutional layer

This layer Perform convolution operations on the time series of the preceding layer with convolution filters. The number of filters can be chosen such that the computation time and accuracy is optimized.

Max pooling layer

This is used to reduce the spatial dimensions of the feature maps. It downsamples the input feature map The main purpose of max pooling is to extract the most prominent features from the feature maps, while also reducing the dimensionality of the data.

Flatten layer

This converts multidimensional input data into a one-dimensional array. The flattened output is then passed on to the subsequent layers in the neural network, that is the Dense layer for further calculation

Single dense layer

The output of each neuron in this layer is calculated as a weighted sum of the inputs from the previous layer, which is then passed through a sigmoid activation function to produce the final output.

The predicted labels will be floating point values between 0 and 1. These floating point values are then converted to binary values using a threshold value of 0.5. All values above the threshold are classified as class 1, and those below the threshold are classified as class 0.

4 Results and Discussion

The final model is trained on a training set of 60 triggers and a validation data set of 20 triggers using a batch size of 4 and 35 epochs. The other parameters were tuned to optimize the accuracy and computation time. The accuracy on the training dataset is almost equal to 1 in the final epoch, with the loss converging to 0 in every epoch (Table 3). The model is tested on 20 triggers for which an accuracy of 0.7 was obtained.

4.1 Tuning the parameters

It is essential to tune the parameters so that the model is not overfitting or underfitting. The best combination of the following parameters was found to optimize the model accuracy and computation time.

The number of filters in the convolution layer

As the number of filters increased from 64 to 256, the accuracy of the model increased. However, the time required for computation is also increased.

Learning rate and batch size

These are the hyperparameters that affect the speed and stability of the training process. A higher learning rate can speed up training but can also make the optimization process unstable. A smaller batch size makes the training process more stable but also slows down the training. Here the chosen batch size is 4 for all the cases.

Number of epochs

This hyperparameter indicates how many times the model will be trained on the entire training dataset during the training process. More number of epochs will allow more opportunities for the model to adjust its weights and improve its performance. However training for too many epochs will lead to overfitting that is the model becomes too specialized to the training dataset and loses its ability to generalize to unseen data. Whereas too few epochs lead to underfitting, where the model cannot capture the underlying patterns in the data, thereby performing poor on both training and unseen testing data. The effect of the varying number of epochs is easily reflected in the accuracy of the model, as seen in Table 2.

A good combination of parameters was identified by monitering the loss and accuracy on the training data set. Converging loss obtained for training data set can be seen for one of such training in table 3. Here the additional data set is used as validation dataset. The early stopping technique was used to prevent the model from overfitting.

No. of files in training set	No. of filters	no.of epochs	Accuracy
40	128	10	0.5
40	256	20	0.8
70	64	20	0.48
70	128	20	0.52
70	256	20	0.48
70	256	35	0.68
70	256	33 (With early stopping)	0.76

Table 2: Accuracy on testing dataset (25 triggers) obtained with different parameters. The batch size is 4 in all the cases and rest other parameters are constant.

4.2 Case of overfitting

Initially, the model was trained on a smaller data set with a training data set consisting of 20 of each GRBs and TGFs and testing data of 15 triggers. The model performed very well on the training data set (accuracy of 0.9), which can be considered as a case of overfitting. However, the accuracy on the unseen testing data set of above 0.7 is also to be considered. This case usually arises when there is less variance in the dataset. To check the similarity of the testing and training data set, the matrices used as input for the model were flattened to vectors, and cosine similarity between all the pairs was computed. The average cosine similarity was close to 1, indicating that the matrices are very similar. Hence, the model is expected to work well on the testing data set despite overfitting.

The immediate solution is to increase the size of the training data set and thereby increase the variance within the data. Implementing early stopping techniques is also another solution to prevent overfitting of the model. The training data set was split into two parts, 70% is used as the training data, and 30% is used as validation data.

Epoch	Loss	Accuracy	Val_loss	Val_accuracy
1	198.0213	0.4762	31.1639	0.5625
2	27.7698	0.4286	5.4507	0.5625
3	48.6603	0.5079	33.6812	0.5
4	40.9856	0.4444	37.3411	0.5
5	14.4252	0.4921	1.1163	0.625
6	7.7231	0.6032	6.075	0.5
7	14.6343	0.5873	10.9024	0.5
8	6.0987	0.4603	1.6587	0.5625
9	1.7347	0.7778	2.0201	0.5625
10	17.2176	0.5556	28.4349	0.5625
11	38.8073	0.5079	0.0492	1
12	26.2673	0.5714	27.8052	0.5
13	32.7349	0.5079	39.9016	0.5
14	16.7185	0.6032	12.7969	0.5625
15	3.2193	0.6984	0.009	1
16	0.3141	0.873	0.5902	0.6875
17	0.1115	0.9365	0.001	1
18	0.002	1	0.0014	1
19	9.89E-04	1	4.38E-04	1
20	6.28E-04	1	2.92E-04	1

Table 3: The loss and accuracy while training during each epochs. The columns *loss* and *accuracy* are for training data set, *val_loss* and *val_accuracy* is for the validation dataset.

Using early stopping, the validation loss gets monitored, and if the loss does not improve for a certain number of epochs, the training process is stopped. This helps to reduce the generalization error of the model and improve its performance on new, unseen data.

5 Conclusion and Future Outlook

A simple neural network model consisting of a single convolution layer gives the desirable accuracy on the training data set and can make predictions for unseen data with considerably high accuracy. However, considering that the training data set is small, the possibility of overfitting the model must be considered. Expanding the training data set can help diversify the data and reduce the risk of overfitting.

This project focussed on only two classes of triggers - GRBs and TGFs. Other triggers like solar flares, local particles, soft gamma repeaters, and more (Table 1) can be included. Suitable modifications like covering up more bin sizes will be required to include more classes. The computational complexity of the model will also increase when other classes are included.

Furthermore, the algorithm only identifies if the given trigger belongs to either class 1 or class 2 (GRB or TGF); it would be substantial to include the case where the algorithm also identifies that a given trigger does not belong to any of the classes identified in the training data set, which will help in identifying potential unknown classes of triggers. This will require building an unsupervised learning model for the classification of data.

Further extension of this project is to assemble a complete package for determining if there is any trigger, thereby classifying it into known classes or the unknown.

Bibliography

Gruber, David. *High-energy transients with Fermi/GBM.*

Jenke, P. A. et al. "THE FERMI–GBM THREE-YEAR X-RAY BURST CATALOG". In: *The Astrophysical Journal* 826:228 (2016), p. 15.

Kunzweiler, F. et al. "Automatic detection of long-duration transients in Fermi-GBM data". In: *Astronomy Astrophysics* 665 (2022).

Luca, A. De et al. "The EXTraS project: Exploring the X-ray transient and variable sky". In: *Astronomy Astrophysics* 650 (2021).

Meegan, Charles et al. *The Fermi Gamma-Ray Burst Monitor.*

Zhao, Bendong et al. "Convolutional neural networks for time series classification". In: *Journal of Systems Engineering and Electronics* 28.1 (2017), pp. 162–169.