INTERNSHIP REPORT : MASTER 2

ETIENNE RUSSEIL

Identifying Pair Instability SuperNovae in the largest astronomical dataset ever created

Université Clermont Auvergne
Laboratoire de physique de Clermont

Supervisor : Emille E. O. Ishida
Team manager : Emmanuel Gangler

February 1st, 2021 — July 2nd, 2021

Acknowledgment

I want to thank Emille Ishida for her quality supervision and her wise advises throughout my internship. She has been very trustful and was open to my ideas, making my work interesting and pleasant. I deeply enjoyed learning by her side as her kindness and enthusiasm kept me motivated.

I want to thank Emmanuel Gangler for the kind welcome in the laboratory. His precious remarks on scientific points improved the overall quality of this work.

I also want to thank the whole LSST team from the "Laboratoire de Physique de Clermont" for their hospitality and their goodwill. Working in this environment has been a chance for me.

Finally I want to thank the Cosmostatistics Initiative (COIN) for providing me storage and computational power.

Abstract

The Vera Rubin Observatory Legacy Survey of Space and Time (LSST) is about to become operational and will produce 15 TB of data per night. In this data will certainly be hidden never observed but theoretically predicted objects and we need to be ready if we don't want to miss them. To deal with such amount of information we need to create machine learning models able to process the data. In this work we have implemented a pipeline to detect pair instability supernovae (PISNe), that are theoretical objects for now. We have used the PLAsTiCC simulated data base to train and test machine learning algorithms in both supervised and unsupervised scenarios. The supervised method is using a classification algorithm but it has proved to be too conservative to find PISNe in a reliable way. The unsupervised method is using an anomaly detection algorithm and is a powerful tool to isolate high signal PISNe. This work represents an initial stage of a science module which can be used in the Fink broker to detect long term transients like PISNe. The work done here will serve as a basis for future developments and implementations in the broker, enabling the discovery of rare astrophysical objects.

Contents

1 Introduction **4**

2 Background **5**
2.1 Measurements in astronomy . . . 5
2.1.1 Spectroscopy . . . 5
2.1.2 Photometry . . . 5
2.2 The PLAsTiCC data challenge . . . 7
2.3 Pair-instability supernova . . . 7

3 Processing the data **8**
3.1 PLAsTiCC data . . . 8
3.1.1 Data format . . . 8
3.1.2 Additional metadata . . . 9
3.2 Filtering the data . . . 10
3.2.1 Why filtering . . . 10
3.2.2 Adding/removing PISN . . . 11
3.2.3 Extra-galactic filter . . . 11
3.2.4 Cadence filter . . . 11
3.2.5 Passband filter . . . 11
3.2.6 Detection filter . . . 11
3.2.7 Partial curve filter . . . 12
3.2.8 Data transformation . . . 12
3.2.9 Final "completeness" filter . . . 12
3.2.10 Result of the filtering process . . . 12
3.2.11 Discussion about code optimisation . . . 13
3.3 Feature extraction . . . 13
3.3.1 What is feature extraction . . . 13
3.3.2 Fitting a model . . . 13
3.3.3 Extra parameters . . . 14
3.3.4 Result of the feature extraction . . . 14

4 Data analysis **15**
4.1 Methods used . . . 15
4.1.1 Random forest . . . 15
4.1.2 Isolation forest . . . 16
4.2 Sample used . . . 16
4.2.1 WFD sample . . . 17
4.2.2 DDF sample . . . 19
4.3 Results on complete light curves . . . 19
4.3.1 Random forest result . . . 19
4.3.2 Isolation forest result . . . 20

5 Conclusion **24**

A Pair instability supernovae in the Deep Drilling Field **27**

1 Introduction

For millennia, stars have been guides to locate ourselves on earth. They seem so immutable at human scale that they make the perfect compass for us. But in fact, not every celestial object has a static nature. Some of them change brightness over time and are the subject of extensive astronomical research.
We distinguish two kind of evolving objects. First there are the variables, sources for which luminosity vary periodically over time. Then there are transients, objects that quickly brighten, fade over time, and are never seen again. Among them are the famous supernovae, the final stage of evolution of massive stars. There are many types of transient and variable objects but we measure them all in the same way : by taking repeated pictures of the sky and observing the luminosity evolution.

Some telescopes in charge of this continuous survey work are currently operating. The Zwicky Transient Facility (ZTF) [2] is a great example [8]. Since 2017 it is searching for transient objects such as supernovae or gamma ray burst, among others. To do so it is taking a template picture of the sky and if, later, the next pictures of the same spot of the sky shows a significant difference in luminosity, an alert will be send for astronomers to analyze it. Figure 1 shows the subtraction process ZTF is currently producing up to 1 200 000 alerts per night[17] , which is about 10 times more than it's predecessor [4].

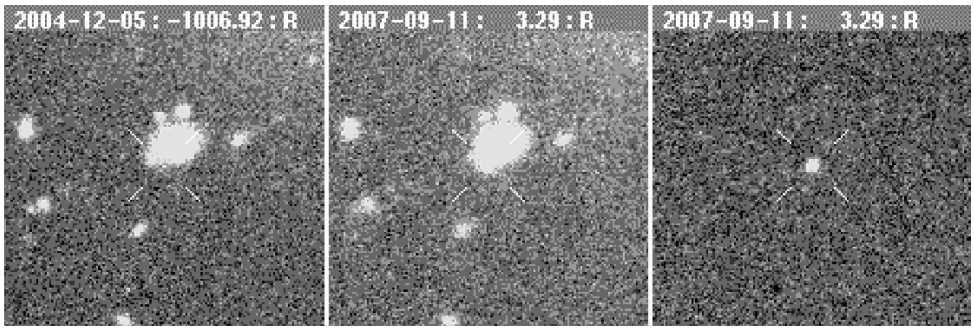

Figure (1) Example of image subtraction procedure to detect transient objects. **Left:** background image. **Center**: science image. **Right:** subtraction where the transient appears clearly. [figure from 15].

But things are about to get even bigger as the Vera Rubin Observatory Large Survey of Space and Time (LSST) is about to become operational [6], and will be one of the biggest telescopes of all times. It should be operating on 2021 but unfortunately, due to Covid19, it has been rescheduled to start operations in 2023 [19]. It is designed to observe transient and variable events at a larger scale, taking a complete picture of the austral sky every few days. Such a telescope will produce 200 times more alerts than ZTF [11]. Given such a large data set, most of the data produced by LSST will never be directly seen by humans.

In this context, it is necessary to build tools able to find interesting observations hidden in such a large data base. Indeed considering the amount of objects we are observing, it is reasonable to think that we will observe some hypothetical objects for the first time in history. However, we are certainly not going to encounter them by chance : future astronomical discoveries will only happen if they are planned. Computational and algorithmic methods can help us scan, filter and select the

data, so that experts end up looking at the most interesting objects.

Astronomy brokers have the job to receive the data stream coming from a series of image subtraction analysis (Figure 1), identify interesting objects and re-distribute this information to the community who will use them for scientific studies. The Fink project [14] is one of these brokers. It started at "Laboratoire de Physique de Clermont" (LPC) a couple of years ago and now is a complete system built to treat data from LSST. In the mean time, it treats data from the Zwicky Transient Facility [ZTF, 8]. One of the strengths of Fink is its machine learning based classifiers, which help the broker to select which detection to sent to which user.

This is where this work takes place. It aims at discovering never observed but theoretically predicted objects – more specifically, pair instability supernovae [7]. But the the framework is general enough to be applied to other target classes. In this work we constructed a machine learning based pipeline which implements machine learning techniques in the treatment of simulated LSST data in both, supervised and unsupervised scenarios. We used simulated data to develop a model which can be easily incorporated in the broker, thus helping astronomers to search for these very rare objects. All the code used to produce these results are publicly available at https://github.com/erusseil/PISN-classification.

2 Background

2.1 Measurements in astronomy

There are various objects in our universe that emits light and are thus observable from earth. We characterize them by measuring the photon flux coming from the source. This measurement can happen using two different techniques: spectroscopy and photometry. In what follows we give further details about each one of them.

2.1.1 Spectroscopy

A spectroscope divides the incoming light into fine bins in wavelength, thus providing a detailed measurement of the flux in many regions of the wavelength spectrum. This is how we obtain the example curve in Figure 2.
As we can see, the curve is so detailed that we can observe absorption lines. It gives us information about the chemical composition of the object that has emitted the light. Expert astronomers may then classify the source using such a spectra.
Unfortunately spectroscopy has a big limitation : it is time consuming [12]. Allowing our detectors to receive enough photons for such detailed observation requires long periods of integration (pointing the telescope to the same point in the sky). Moreover, the technique requires extremely good observation conditions and relatively bright objects. Therefore spectroscopy is a powerful tool to characterise objects but its application to large scale surveys is limited.

2.1.2 Photometry

Photometric measurements summarize the light collected from a region in the sky into larger wavelength passbands. It requires considerably less integration time than spectroscopy and allow us to measure the brightness of many objects simultaneously in a given passband. Therefore photometry is very powerful tool to scan quickly a lot of objects, however, is it way harder to determine

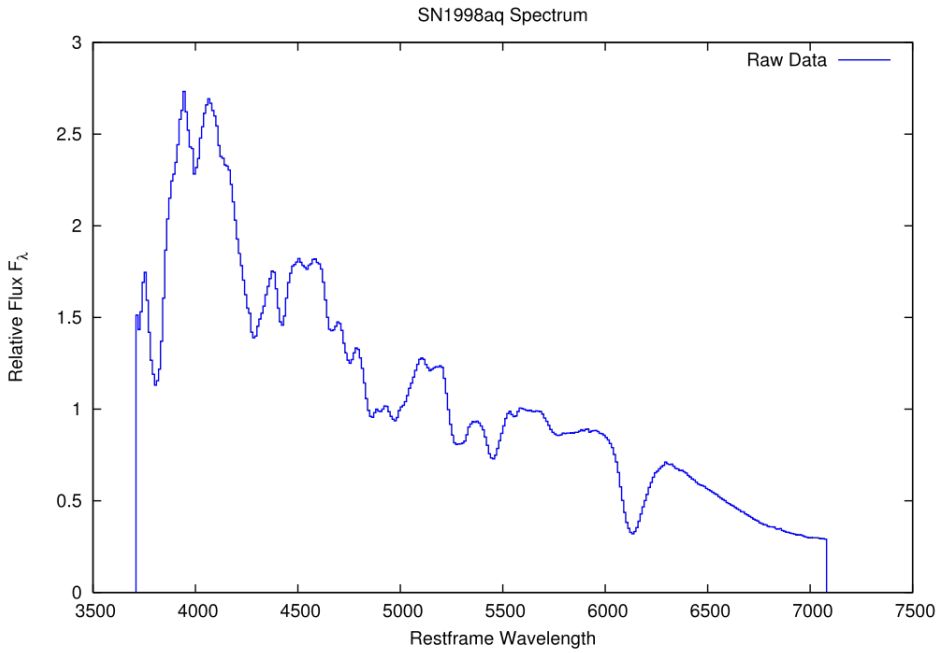

Figure (2) Example of a spectra of a type Ia supernova [3]

.

the nature of the object using such low resolution information. Photometry is used to study the evolution of brightness as a function of time by taking repeated observations of the same area in the sky in different epochs.

LSST will be a photometric telescope, which explains the amount of data it will produce. In it's case, photometric observations will divide the incoming light into 6 passbands. Figure 3 shows transmission and width of the 6 passbands of LSST.

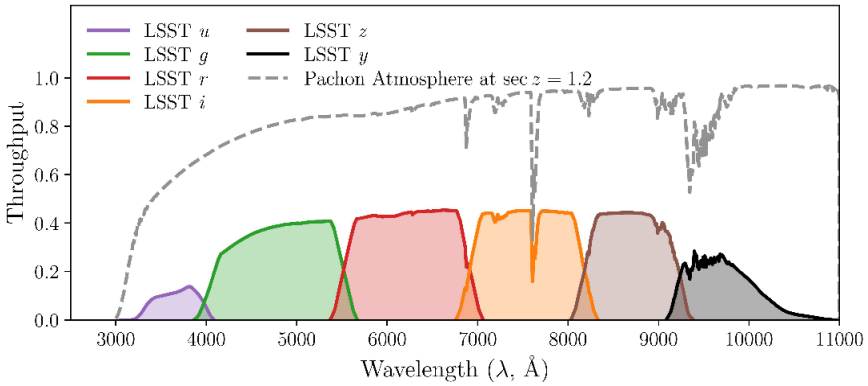

Figure (3) Passband transmission efficiencies for all filters of the Vera Rubin Observatory Large Survey of Space and Time (LSST) [figure from 15].

2.2 The PLAsTiCC data challenge

In 2018, in order to prepare for LSST massive incoming data, a classification challenge has been hosted on Kaggle. A database of what LSST could produce in 3 years of operation has been simulated for the occasion : it is called the "Photometric LSST Astronomical Time-series Classification Challenge" [(PLAsTiCC) 20].

The PLAsTiCC complete simulated data has been created using the models and their expected rates shown in table 1 – it consists of more than 100 million objects. However, most of these were too faint to be detected by LSST, leaving the final dataset with 3.5 million objects in total. Each objects has been measured several times, in 6 passbands, we therefore reach a total of ≈453 million observations.

Since the goal was to train a machine learning model, two datasets were created : a training sample, provided to the contestants and a testing sample, used to evaluate the performance of their algorithms. To mimic real conditions, the training sample was rather small since it was composed of 7,846 already known objects considered to be spectroscopically confirmed. On the other hand, the testing sample was made of 3,492,888 objects, corresponding to a photometric-only sample. The training sample contained 14 classes of objects, while the test sample had 14+1, the extra class corresponding to rare or theoretically predicted objects still lacking observation.
We can quickly see how difficult the task of training a predicting model will be, since these datasets break the most basic assumption of machine learning : training and testing should be representative of each other. Indeed there is a statistical difference of three order of magnitude and never seen objects in the testing sample to make things harder.

But people worked hard on this challenge and managed to create algorithms that were good at predicting the type of objects based on their light curves. Unfortunately, even though people were encouraged to classify every unknown objects under a common type "unknown", nobody did. Indeed the incentives of the challenge weren't suited for that since the score was based on the number of accurately predicted type of objects. Because of the very low number of objects from the "other" class in the testing sample, it wasn't really worth identifying them. In this work we will try to identify one of these theoretical objects: pair instability supernovae.
The challenge ran from September/2018 to December/2018. After the competition finished, the organizers released the complete data set, including the corresponding labels, to the public. This complete post-challenge data set was the one used in this work.

2.3 Pair-instability supernova

A couple of hundred million years after the Big Bang, the vast inert gas that was our universe had collapsed into itself, creating the first stars. Those stars are called population III stars and have very unique behavior [10]. Indeed, during the nucleosynthesis, 3 minutes after the Big Bang, the quasi totally of nuclei produced were hydrogen and helium[5]. At that time, metals (which, for astronomers, are elements heavier than helium) were not part of our universe yet. Consequently, population III are very low metallicity stars but as big as a 300 solar masses [10].

A star can remain stable thanks to an equilibrium of forces : the gravitational pressure, pushing inward and the radiation pressure coming from the nuclear fusion, pushing outward. If the balance is broken, the star will collapse and "die". In very massive and low metallicity stars (between 130 and 250 solar masses), the production of a pair electron/positron may occur [7]. They come

from a very energetic gamma ray that collides into a nuclei. Soon, the pair will collide back into a gamma photon. But during this time, the gamma ray didn't exist and the internal pressure of the star dropped a little. The star has therefore contracted, rising it's internal temperature. An higher internal temperature means more electron/positron pair production: the star enters an inevitable loop that will result in a very violent collapse. It will finally explode in an event called a Pair Instability Supernovae (PISN). It is one of the most violent event in our universe, it will completely destroy the progenitor, leaving nothing behind (whereas classical supernovae leaves a neutron star or a black hole).

The fact that these events took place very early in the universe history gives information about the primordial environment they inhabited thus providing important insights into the first stages of cosmic evolution. Since the redshift of these objects will be significantly high we expect to measure very stretched light curves (duration in the order of years). In the PLAsTiCC data set there are 1172 PISN and all of them are contained in the testing sample.

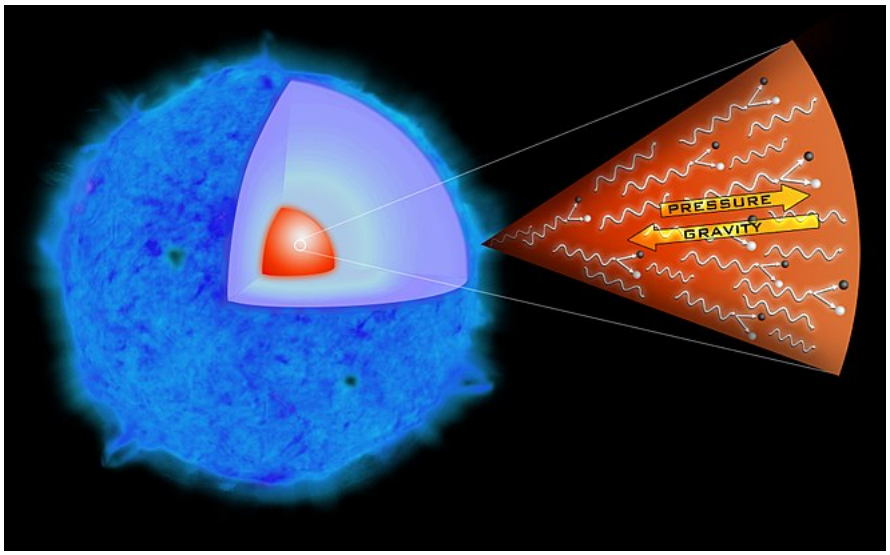

Figure (4) Illustration of a pair instability supernova [figure from 1].

3 Processing the data

3.1 PLAsTiCC data

3.1.1 Data format

The data is available as CSV files and their format is illustrated in Figure 5.

	object_id	mjd	passband	flux	flux_err	detected_bool
0	13	59798.3205	2	-1.299735	1.357315	0
1	13	59798.3281	1	-2.095392	1.148654	0
2	13	59798.3357	3	-0.923794	1.763655	0
3	13	59798.3466	4	-4.009815	2.602911	0
4	13	59798.3576	5	-3.403503	5.367328	0

Figure (5) Extract of the PLaSTiCC data as available in its zenodo files.

model class num[a]: name	model description	contributor(s)[b]	Nevent Gen[c]	Nevent train[d]	Nevent test[e]	redshift range[f]
90: SNIa	WD detonation, Type Ia SN	RK	16,353,270	2,313	1,659,831	< 1.6
67: SNIa-91bg	Peculiar type Ia: 91bg	SG,LG	1,329,510	208	40,193	< 0.9
52: SNIax	Peculiar SNIax	SJ,MD	8,660,920	183	63,664	< 1.3
42: SNII	Core Collapse, Type II SN	SG,LG:RK,JRP:VAV	59,198,660	1,193	1,000,150	< 2.0
62: SNIbc	Core Collapse, Type Ibc SN	VAV:RK,JRP	22,599,840	484	175,094	< 1.3
95: SLSN-I	Super-Lum. SN (magnetar)	VAV	90,640	175	35,782	< 3.4
15: TDE	Tidal Disruption Event	VAV	58,550	495	13,555	< 2.6
64: KN	Kilonova (NS-NS merger)	DK,GN	43,150	100	131	< 0.3
88: AGN	Active Galactic Nuclei	SD	175,500	370	101,424	< 3.4
92: RRL	RR lyrae	SD	200,200	239	197,155	0
65: M-dwarf	M-dwarf stellar flare	SD	800,800	981	93,494	0
16: EB	Eclipsing Binary stars	AP	220,200	924	96,572	0
53: Mira	Pulsating variable stars	RH	1,490	30	1,453	0
6: μLens-Single	μ-lens from single lens	RD,AA:EB,GN	2,820	151	1,303	0
991: μLens-Binary	μ-lens from binary lens	RD,AA	1,010	0	533	0
992: ILOT	Intermed. Lum. Optical Trans.	VAV	4,521,970	0	1,702	< 0.4
993: CaRT	Calcium Rich Transient	VAV	2,834,500	0	9,680	< 0.9
994: PISN	Pair Instability SN	VAV	5,650	0	1,172	< 1.9
995: μLens-String	μ-lens from cosmic strings	DC	30,020	0	0	0
TOTAL	Sum of all models		117,128,700	7,846	3,492,888	—

Table (1) Transient classes and their respective rates present in the PLAsTiCC data set [table from 12].

- **object_id**: An identification number associated to each unique object;
- **mjd**: Modified Julian Date of the observation;
- **passband**: LSST passband used for the measurement, such that $[u, g, r, i, z, y] \rightarrow [0, 1, 2, 3, 4, 5]$;
- **flux**: The measured flux;
- **flux_err**: The uncertainty in flux measurement;
- **detected_bool**: If 1, the point is a valid observation with at least 3σ above the noise level.

We can visualize the available measurements for a given object by plotting it's light curve : that is it's flux against time for each passband (Figure 6).

3.1.2 Additional metadata

In addition to the data described above, the PLaSTiCC dataset also contains metadata files. These hold extra information about each object, such as position in the sky, redshifts, etc. Here, we are going to cover only the information used in this work: redshift, deep drilling field identification and class.

- **true_target**: An integer associated to each class;
- **true_z**: Value of the spectroscopic redshfit (float). Since speed inside a galaxy is negligible compared to the speed of the galaxy itself, objects in our galaxy are fixed at 0 redshift.
- **ddf_bool**: A boolean identifying observation strategy. The PLAsTiCC data was simulated considering two different strategies for repeatedly sampling different areas of the sky. Most of the sky will be observed with sparse observations (long intervals between two consecutive observations of the same object). This strategy is called Wide Fast Deep (WFD). The second strategy, called Deep Drilling Field (DDF) will cover small areas of the sky with a high cadence, thus providing light curves with a gap of only a few days between points. The division of the sky and the associated observation strategy is illustrated in Figure 7. If **ddf_bool** is 1 the object was observed following the DDF strategy, and if 0, following WFD.

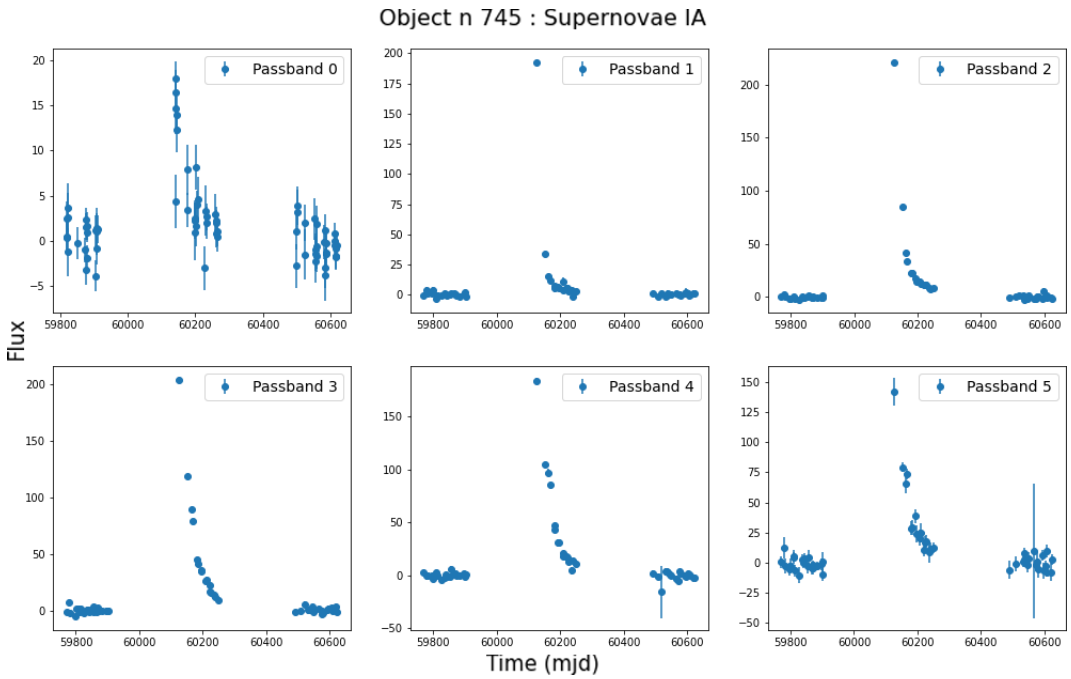

Figure (6) Example light curve from the PLAsTiCC data set. It shows the flux evolution of a supernova Ia in the 6 different LSST passbands.

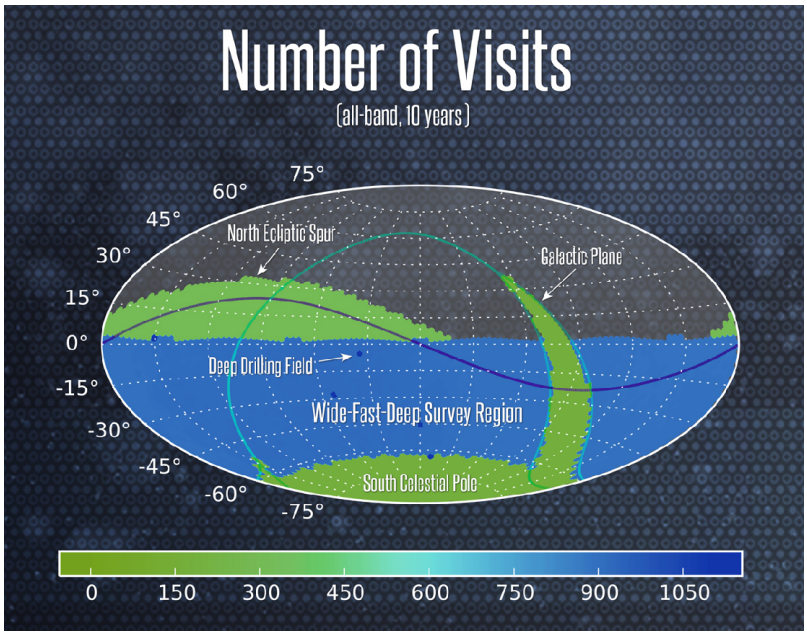

Figure (7) Observation strategy of the Vera Rubin Observatory Large Survey of Space and Time [figure from 15].

3.2 Filtering the data

3.2.1 Why filtering

The PLaSTiCC database constitutes a massive source of information. We want to use it to prepare machine learning based models that can distinguish PISN from other classes. Since we know what we are looking for, we can filter obvious non-interesting classes and avoid misleading the algorithm.

For example, we know that PISN are not inside our galaxy so we won't bother looking at objects inside the Milky Way.
In order to apply the different filters, we are mainly using pandas[16] and numpy[9] libraries for python which are powerful and fast (when used properly) tools to analyse data. We have created a complete function that takes the PLaSTiCC dataset as an input and outputs a clean and filtered database.

3.2.2 Adding/removing PISN

Before filtering anything from the original sample we have added a way to control the number of PISN in the data. Since there are no PISN in the original PLAsTiCC training sample, we have the possibility to add some. It is also a way to evaluate the performances of the algorithm with different number of PISN. This constitutes the only feature of the code specifically targeted towards our problem, all other stages can be easily applied to various cases.

3.2.3 Extra-galactic filter

Since PISN originated in the early universe, we know that they must be very old objects. There is also another hypothesis that some PISN might have happened more recently in very empty region of the universe, where large clouds of hydrogen could still have existed long after the first stars have formed [22]. In any case, PISN can not exist in our galaxy and we have added an option to keep only extra-galactic objects. To do so, we are only selecting objects with a redshift larger than zero.

3.2.4 Cadence filter

As mentioned previously, PLAsTiCC simulates two different LSST observation strategies. The DDF set is very small since it is made of about 30,000 objects, but with better sampled light curves. The WFD is about a hundred times larger but with much less detailed measurements. Our code features an option to choose which cadence to use. We analysed the data using only one set at a time in order to isolate the impact of cadence in our results.

3.2.5 Passband filter

Since PISNe are expected to be found at high redshift, we expect a very important redshift effect that will concentrate the information in the redder (high wavelengths) passbands. Therefore we have added a function that allows you to filter all the unwanted passbands.

3.2.6 Detection filter

As shown in Figure 8, points with the `detect_boolean` $= 1$ are points with considerable signal-to-noise-ratio (SNR > 3). However, even points with `detect_boolean`$= 0$ contain valuable information. They indicate epochs when the object was not bright enough to produce a significant imprint in our detector. Our code accommodates an option to filter according to this flag.

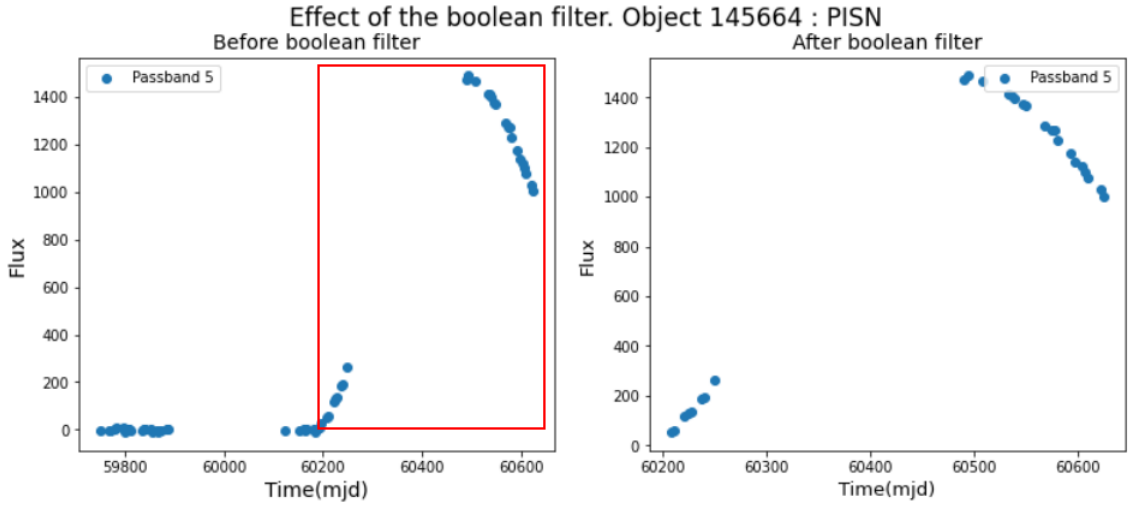

Figure (8) Example of detected boolean filter in a light curve from a PISN at redshift $z = 0.382$.. **Left:** Complete light curve. The points highlighted by the red box correspond to `detect_bool`$= 1$. **Right**: Panel shows only points with `detect_bool`$= 1$.

3.2.7 Partial curve filter

Ideally, we would like our model to to be able to identify objects of interest before the peak of the light curve in order to spectroscopically analyse and confirm them. To do so, we need algorithms which can work on partial information. This is why we have added an option that allows us to keep only the points before the peak maximum.

3.2.8 Data transformation

In the PLAsTiCC data set, the time of each observation is given in modified Julian dates [MJD, 13]. These are very big numbers which can cause numerical problems in optimization routines. Since the absolute time of measurement has not meaning by itself, but only in relation to other measurements in the same light curve. Consequently we have implemented an option to use the first observation of each light curve as a starting point for time measurement.

We also identified the maximum observed flux and added an option to normalize each flux and flux error value by this maximum. The absolute value of maximum flux is stored in a new file whenever the normalization option is chosen, to avoid loosing any available information regarding brightness

3.2.9 Final "completeness" filter

Once every filter has been applied, many light curves are going to end up unexploitable. Therefore we have implemented an option that removes every light curves containing less than 3 points (this number can be modify by the user). In addition, and to keep everything coherent, every object that misses a light curve in any of the previously asked passbands will be remove from the final sample.

3.2.10 Result of the filtering process

This process ensures a fast and clean way to obtain a robust database for your analysis. It gives great flexibility which allows us to design filters for targeted problems. In addition, executing the algorithm returns real time information about the state of your database so that you can deduce which filters are causing what effect. Figure 9 shows an example of the outputted information.

```
 CREATION OF THE TRAINING DATA BASE

We start with  7848 objects and 1421705 measurements

After we add/remove PISN we have 8434 objects and 1496647 measurements
--> There are  586 PISN in the dataset

After EXTRA-GALACTIC and DDF we have 1580 objects and 519192 measurements
--> There are  4 PISN in the dataset

After PASSBANDS we have 1580 objects and 338260 measurements
--> There are  4 PISN in the dataset

Total time to translate mjd 0.1 sec

Total time to normalise flux 0.1 sec

Total time to check completness 0.1 sec

After COMPLETNESS we are left with 1580 objects and 338260 measurements
--> There are  4 PISN in the dataset
```

Figure (9) Example of output from our pipeline after a filtering process.

3.2.11 Discussion about code optimisation

The code to filter the data is entirely written in `Python` [21]. Using loops is an extremely slow process in Python. This is unnoticeable when working on a small amount of data but it can become a real problem in our case. Fortunately the libraries `Pandas` [16] and `Numpy` [9] used in this work solve the problem because they are implicitly using C++ to do the calculations. Therefore, in some cases and when used correctly, one can manage to get rid of the loops and obtain a fast running code. It is the case of our filtering function which take less than a minute to compute 300 thousand objects, thus allowing to easily generate a large databases.

3.3 Feature extraction

3.3.1 What is feature extraction

The idea of feature extraction is to describe each light curve by a fixed number of features determined via mathematical modelling. In a way we are compressing the complete curve information into few values, while at the same time ensuring that each object is described by the same number of parameters. Figure 10 is an example of fitting where we are using a quadratic polynomial model. Further information on the fit is given in Section 3.3.2.
We have implemented a function that takes a filtered database as an input, and that automatically extracts the features of every object in every passband. Similarly to the filtering function, we have added options that give control over the result we want to produce.

3.3.2 Fitting a model

Any mathematical model, requiring any number of parameters can be inputted inside the fitting function. It will then perform the fit using the least square method from numpy. We need to keep in mind that a model using N parameters will need at least N data points to produce a meaningful fit.
For a given object, each passband will be fitted independently, thus multiplying the number of features that describes one object. For example using a model requiring 3 parameters in all passbands, we will produce $3 \times 6 = 18$ parameters for each object.

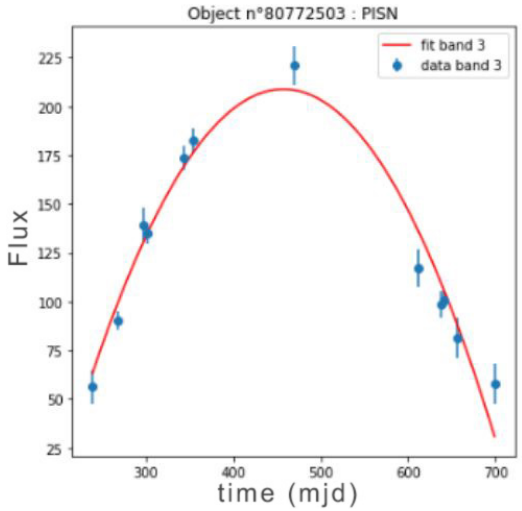

Figure (10) Example of polynomial fit to a pair instability supernova light curve.

In this work we kept the mathematical model simple and focused on the quadratic polynomial model :

$$f(x) = a \times x^2 + b \times x + c. \tag{1}$$

Although this function is not able to capture the details of all the light curves shapes present in our data set, it is well suited to describe long events, which is exactly the case for PISNe. Thus, the parameters extracted from PISNe will occupy a different region of the parameter space when compared to features derived from light curves of other classes. This is enough to give the machine learning algorithms a way to differentiate between classes.

3.3.3 Extra parameters

Getting the parameters from the fit is a first step in describing an object but we can augment the parameter list with some meaningful values. This is why we have implemented options to add up to three extra parameters for each passband :

- **Fitting error**: When applying a least square method, you are minimizing the sum of squared residuals. Adding this value as a parameter gives an important information as a machine learning algorithm might not give the same weight to a good fit than to a catastrophic fit.
- **Number of points**: Adding the number of points used for the fit carries complementary knowledge with the previous error parameter, since it indicates the amount of information used in the fit.
- **Maximum flux**: this is where the previously saved table of all the maximum flux for every light curves is used. Adding it as a new parameter ensures we do not loose information about intrinsic brightness of the object in observer frame, even after normalization.

3.3.4 Result of the feature extraction

Starting from a filtered database, we obtained a clean rectangular table of features. Each line corresponds to a given object and the two first columns describe the ID and the class of the object. Every

other columns are parameters extracted. Figure 11 is an example of such a table (see Table 1 for a correspondence between physical classes and the code number shown here).

			Passband 1					Passband 2		
	object_id	target	0	1	2	3	4	5	6	7	...
0	713	88	-8.091144e-07	-0.000860	0.501113	1.683054	56.0	10.529041	-1.352675e-06	-0.000403	...
1	730	42	7.966541e-07	-0.000238	0.004927	1.240684	52.0	20.994711	3.594613e-07	0.000072	...
2	745	90	-9.798144e-07	0.000830	-0.037826	0.463179	56.0	220.795212	-1.546852e-06	0.001319	...
3	1124	90	4.360581e-07	-0.000047	-0.005754	0.791369	58.0	106.671692	2.813419e-07	0.000026	...
4	1598	90	6.155099e-07	-0.000396	0.028149	0.697264	58.0	1289.851440	6.379362e-07	-0.000409	...
...	...	...	...	...	...	...	...	...	...	...	...

Figure (11) Shape of feature extracted data matrix.

4 Data analysis

4.1 Methods used

We describe below the two machine learning algorithms used in this work: a supervised and an unsupervised method based on decision trees. We used the `scikit-learn` [18] implementation of these routines in our pipeline.

4.1.1 Random forest

Decision trees are supervised machine learning algorithms based on conditional control statements. Figure 12 is a simple illustration of a decision tree. A tree is composed of decision nodes represented as ellipses. They are conditions that will divide your sample into two sub-samples . The tree is built in such a way that the decisions are decreasingly discriminant and it can be as long as necessary.

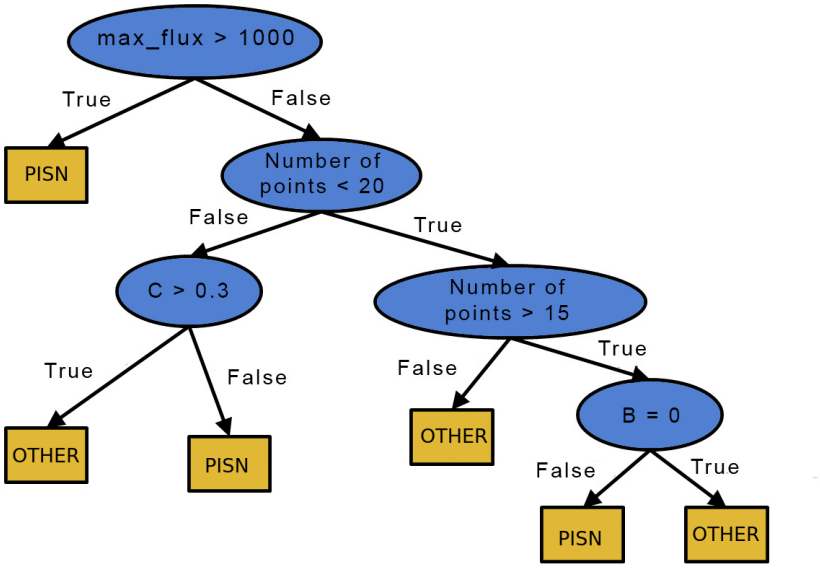

Figure (12) Simple fictional illustration of a decision tree.

After a succession of decision nodes we eventually reach a state where a node isolates objects of the same class, generating an end node or leaf – represented here as rectangles. Once the tree is constructed using the training sample, an input value from the test sample is passed through the tree until it reaches a lead node.The algorithm return the corresponding class of this leaf node as the predicted class. Decision trees can be automatically generated from a dataset made of input values features, expected results (labels) and parameters to be used for the construction of the trees (e.g. maximum depth).

Decision trees are great tools because they are easy to understand but they lack stability. Indeed, changing just a little bit the data can result in a completely different tree shape which would perform well in the training data but fail in the test. This is called overfitting.

Overfitting can be avoiding by considering a larger set of possible trees. This is called a random forest algorithm. The idea is to generate a lot of trees from the same data (with an additional small randomicity). When an input value is passed to the forest, each tree will output an answer. Each output counts as a vote and we consider that the definitive output of the forest is the most voted answer. This method constitutes a robust supervised method for classification and we are going to use it for our analysis. We chose to use the function `RandomForestClassifier` from the `scikit-learn` library.

4.1.2 Isolation forest

Isolation trees are very similar to decision trees but serve other purposes. They are an unsupervised machine learning algorithm that is used to evaluate how different an object is from the rest of a given dataset. An anomaly score is attributed to each object and we can then compare them.

An isolation tree is build by randomly and repeatedly choosing a parameter and a decision node and splitting the data in consequence. Once a point get isolated, it counts the number of nodes that was needed to isolate it and attributes an anomaly score inversely proportional to this number. Consequently points that need relatively few nodes to get isolated get a higher anomaly score meaning they must be far from other points in the parameter space.

Just like a decision tree, isolation trees are prone to overfitting and therefore we need to use an isolation forests that uses many trees and performs an average over their answers. We will be using this method because it is suited for the study of a low number of interesting objects hidden in a massive data set. We chose the `IsolationForest` method from the `scikit-learn` library
.

The anomaly score returned by this implementation ranges between -1 and 1, where -1 corresponds to the most abnormal score possible and 1 to the most normal score possible. Consequently high anomaly scores counter intuitively means that the object is normal and vice versa.

4.2 Sample used

All the analysis done for this work are based on two different initial filtering processes leading to two distinct samples.

4.2.1 WFD sample

The first sample is obtained by applying relatively low restriction filters and is therefore very large.

- Adding/removing PISN : As we will not be using the training sample directly, we chose let the PISN distribution untouched.
- Extra-galactic filter : As discussed before, if PISN exist, they are necessarily located outside our galaxy. Therefore we chose to keep only extra-galactic objects.
- Cadence filter : This sample will use WFD objects only, which constitute most of the objects. The counterpart is that each light curve has a lower number of points.
- Passband filter: For the passband we chose to exclude the two first bands as the redshift will erase most of the information in those bands. Consequently we are using the bands : $[r, i, z, y]$.
- Detection filter: This option is more subtle and we chose not to apply this filter. Indeed, even though "zero points" carry little information about the object itself, they do have some meaning. Looking and finding nothing gives us information about when the objects started to become significantly bright.
- Partial curve filter: The first step is to make sure that we obtain satisfying results using entire light curves. We might then think about increase the difficulty by splitting the curves. For now we won't be using this filter.
- Data transformation: We are homogenising the data by shifting the MJD and normalizing flux values. We are also applying the final "completeness" filter.

Eventually, we end up with 3069681 objects (among them are 1166 PISNe) which corresponds to 320 million measurements. The population of each class within this sample is detailed in Figure 13. Additionally one may notice that the number of measurements in each passband isn't constant as illustrated on the left panel of Figure 14.
Finally we have represented the redshift distribution of the sample on the left panel of Figure 15. The large size of this data set is a technical difficulty in itself since is is hard to manipulate in a single pandas data frame. Therefore, we need to keep the initial configuration and separate it in 10 data frames.

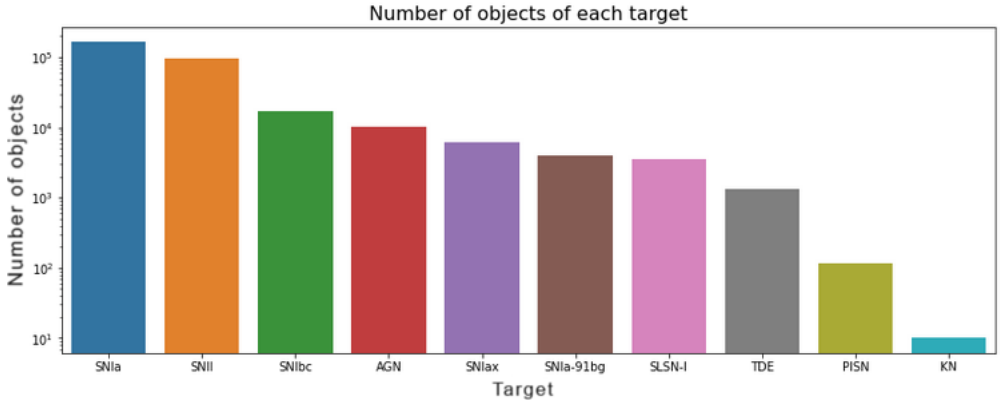

Figure (13) Distribution of number of object per target class for the WFD sample.

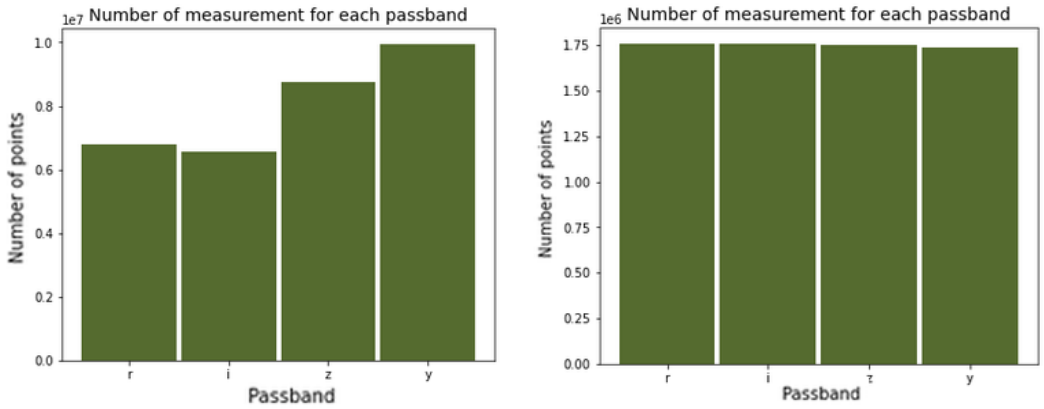

Figure (14) Number of measured epochs per passband. **Left:** Wide Fast Deep sample. **Right:** Deep Drilling Field sample.

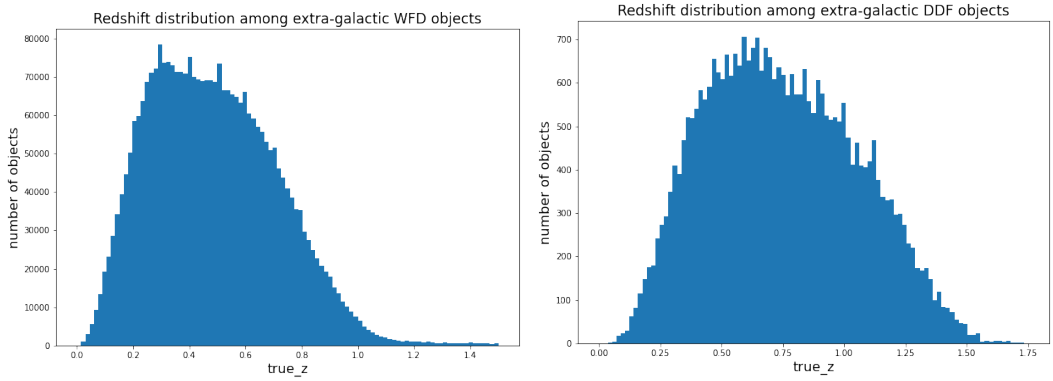

Figure (15) Redshift distribution among objects . **Left:** Wide Fast Deep sample. **Right:** Deep Drilling Field sample.

4.2.2 DDF sample

The second sample that we have created is smaller. It is exactly identical except for the cadence filter that is chosen to be DDF instead of WFD. This sample has several advantages compared to the larger WFD one. First, it's small size allows for faster computation time and makes it a powerful tool to work with since it can be contained in a single data frame. It's second strength is that each light curve contains more measured values, resulting in better fits.

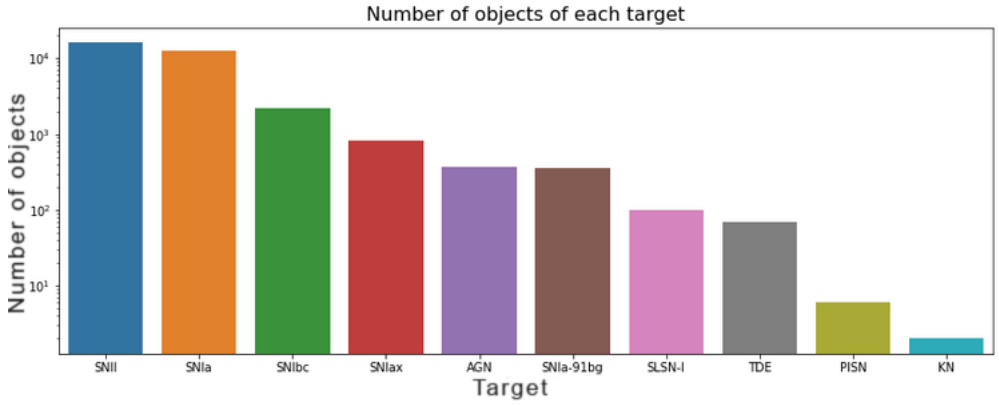

Figure (16) Distribution of number of objects per target class for DDF sample.

The dataset is composed of 32699 objects and a total of 7008772 measurements. Among those objects are only 6 PISNe. It is about 46 times smaller than the WFD sample but the relative number of each class is similar, as shown in Figure 16 . We can also notice that the number of measurement for each passband is constant as illustrated on the right part of Figure 14. Finally the right panel of Figure 15 shows the redshift distribution in the sample. We can notice that the average redshift is significantly higher than the WFD sample.

4.3 Results on complete light curves

In this section we summarize the results we found. At first, we approached the problem with a supervised learning strategy. However, given the low frequency of PISNe in the data (see figures 13 and 16), we also applied an unsupervised learning strategy.

4.3.1 Random forest result

We have run the random forest algorithm on the WFD sample with a feature extraction collecting 3 parameters from the polynomial fit plus the 3 extra metadata parameters. We then used 70% of the sample for the training and 30% for the testing. We performed a binary classification : PISN/non-PISN using 1000 trees. At first sight the results seems unrealistically good since we obtain 99.96 % of correct predictions using the model. Unfortunately this is only due to the relative proportion of PISN compared to the other class. As illustrated in Figure 17 we see that our model tends to classify everything as non-PISN to obtain good results. This was to be expected, considering the problem an anomaly detection solution should be the natural way of proceeding.

The random forest classifier is in fact taking decision on an assumption that deserves to be questioned. Indeed the algorithm is in fact returning probabilities for an object to belong to the class PISN or non-PISN. So intuitively, one would choose the decision threshold to be at 50%. But in fact this value should only be used in case we have balanced classes, which is not the case of our data.

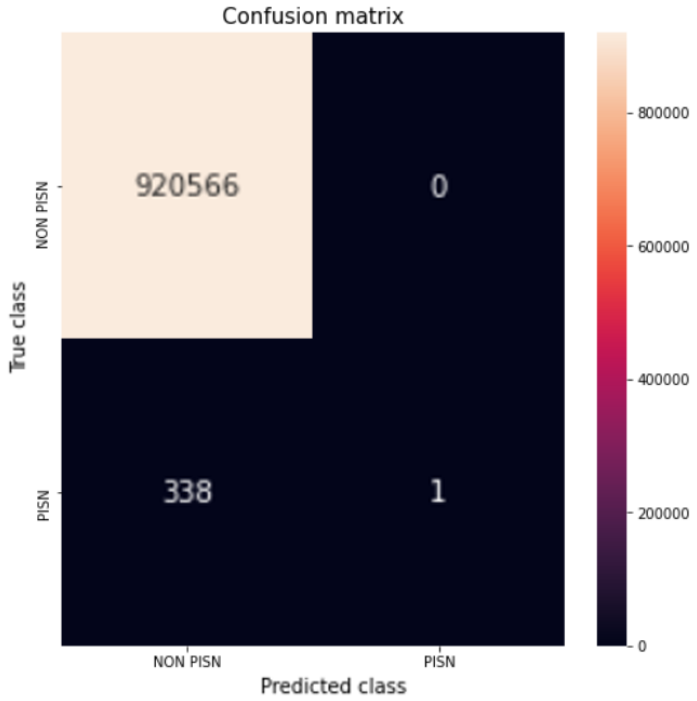

Figure (17) Random forest confusion matrix : Wide Fast Deep sample with 0.5 threshold

We can have a better idea of how we came to this result by looking at Figure 18, which shows the distribution of probabilities of being PISN for all true PISN in the WFD sample. It is good that the one object above 50% is in fact a true PISN with high signal – but this result is too conservative. We would like to discover more PISNe. In the next section, we will use a different method, that will allow us to discover more PISNe.

4.3.2 Isolation forest result

We will now focus on the unsupervised method that seems more suited for the problem. In fact, the classes of the objects are never specified in the model and therefore PISN are not explicitly targeted by the isolation forest. We expect the method to be able to identify a large set of not so common objects, among which there will be a large number of PISNe. As for the random forest, the feature extraction has been performed to extract 3 parameters from the polynomial fit plus the 3 extra parameters offered by the function.

The first extraction has been done on the DDF sample. It was a first attempt to see if our model did well at isolating PISN and since there are only 6 PISN in this sample, it was possible to look at them one by one. Figure 22 shows all the light curves and their corresponding fit. It is crucial to notice right away that only two of them do have a clear signal. The rest have little or no signal and can be considered as carrying no information about the nature of the object. We then performed an isolation forest algorithm using a 1000 trees and order the anomaly scores in increasing order. We consider anomalous objects in the top 1% with higher anomaly score. For the DDF this means we will only be analyzing 327 objects out of the initial 32 699.

Anomaly detection methods are known to have a high rate of false positives. They recognize any statistical anomaly, like errors in measurements or light curves with a bad fit, beyond the really

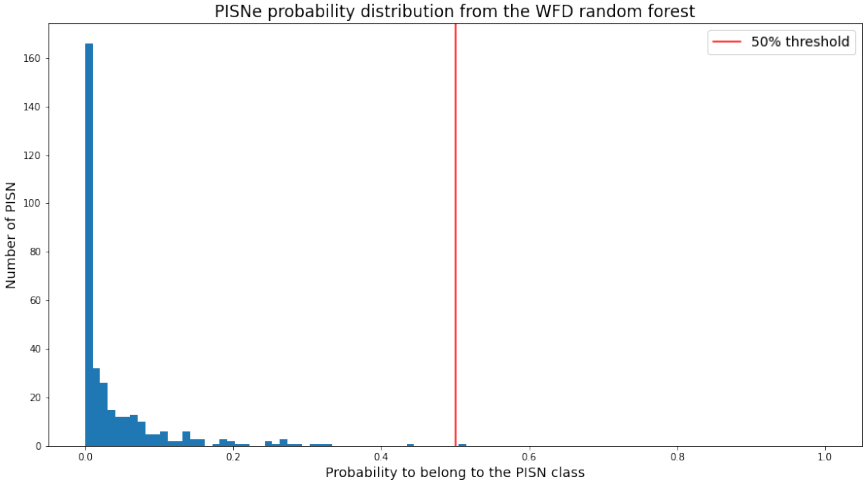

Figure (18) Distribution of probabilities given by the Random Forest classifier. The red vertical line shows the threshold 50% decision threshold.

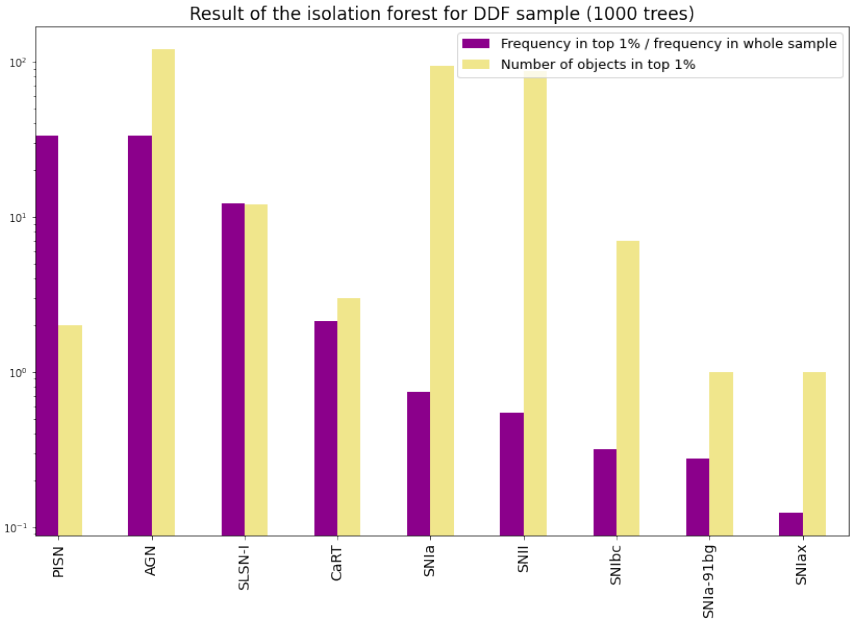

Figure (19) Population comparison for the DDF sample. The purple bars show the frequency of each class in the top 1% objects with highest anomaly score divided by the frequency of that class in the entire target sample. The yellow bars show the number of objects within the 1% most anomalous per class.

interesting ones. So, we do expect to find more false positives than in the previous section. But we also hope to find a larger incidence of PISNe in the top 1% than we would in entire data set.

Figure 19 shows the frequency of objects from each class in the top 1% divided by their frequency in the entire population (purple bars). Even if there are only 2 PISNe in the top 1% with highest anomaly scores they represent a larger portion of this sub sample then the entire sample. In addition, we observe that the PISNe classified as anomalies are the PISNe with high signal previously mentioned.

Moreover, large part of the other objects in the top 1% are active galactic nuclei (AGN) and other types of supernovae (SNII and SNIa). In the context of the broker, we should be able to identify a considerable fraction of the AGNs already cataloged along side their host galaxies. It is also interesting that we have some CART models within the 1%, which are also part of the anomalous class on PLAsTiCC and is of high interest to the astronomers. We can conclude that the first test on the small DDF sample is a great success since it managed to classify every meaningful PISN as anomalies. However, we do need to test if such results persist for a larger sample..

This is the reason why we performed a second extraction on the large WFD sample. As mentioned previously it contains about 3 million objects and among them are 1166 PISNe. Due to the large amount of data we are only using 100 trees in order to reduce the computational time. Figure 20 shows the same distribution of scores as previously but for the WFD sample.

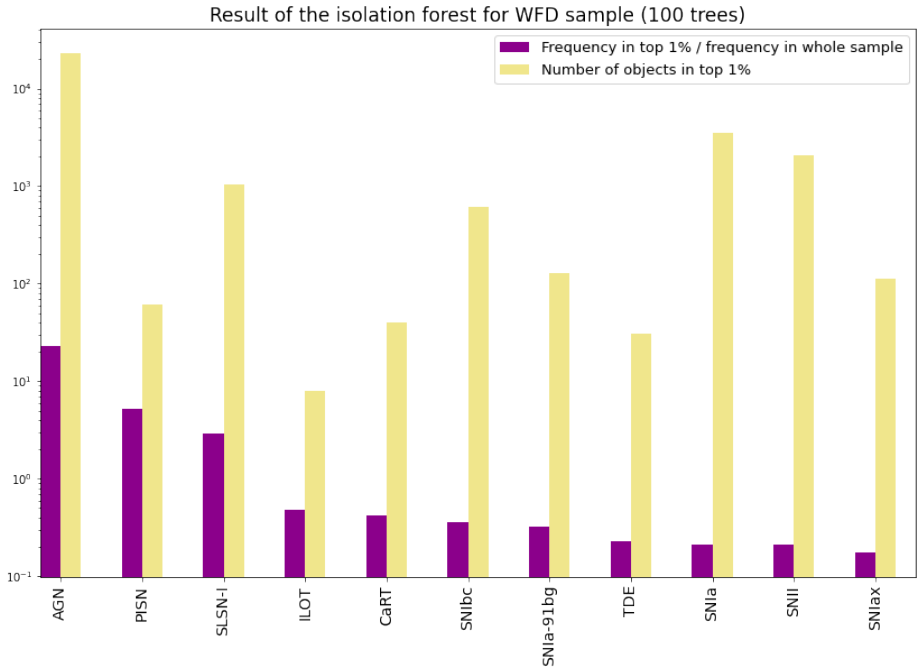

Figure (20) Population comparison for the WFD sample. The purple bars show the frequency of each class in the top 1% objects with highest anomaly score divided by the frequency of that class in the entire target sample. The yellow bars show the number of objects within the 1% most anomalous per class.

In this attempt, 62 PISNe are within the top 1% on the most abnormal objects. Once again we can see that the "frequency ratio score" of PISNe is very high even though AGNs hold the highest score . The only other class that obtains "frequency ratio score" above 1 is the SLSN-1, which means that every other classes are overall considered as normal by the isolation forest model.

The top 1% that we chose to define the anomalies is an arbitrary choice and other thresholds might

be tried depending on the goal of the model. Figure 21 illustrates the distribution of scores for PISNe and shows how the threshold divides the objects. In addition we observe that our isolation forest model is attributing low anomaly scores to most of the PISNe.

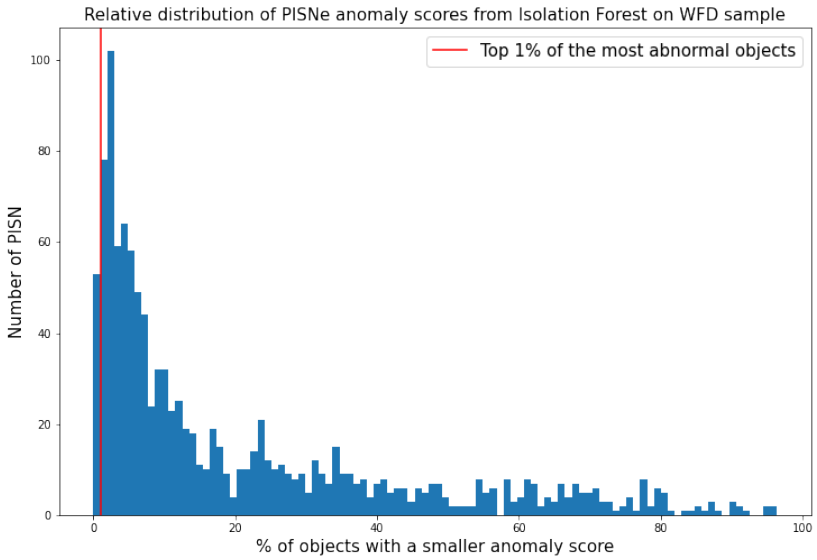

Figure (21) Distribution of the anomaly scores among pair instability supernovae

Finally, as previously, we need to investigate manually the light curves to make sure that higher score ones have low quality signal. And indeed after looking at the lowest score PISNe, we see that they all have a very neat signal. On the other hand, the highest score PISNe have no signal at all.

Once again the isolation forest model offers very good results and it confirms the success observed with the DDF sample. By choosing an appropriate threshold, on may use this algorithm to filter away most of the objects from this sample and be left with a smaller dataset that has high chances of containing high signal PISNe.

5 Conclusion

The overall method that has been developed is a success and is very promising for the future of pair instability supernova research.
The filtering function that we have implemented is fully operational and performs very well, offering enough freedom for it to be use in other contexts. We used it to generate two data bases with two statistical powers : a small DDF data base and a large WFD one.
The feature extraction function is providing correct results and since it can take any fitting function as an input, one can use it in other contexts as well. We successfully applied the function using a quadratic fit on both created data bases to extract 6 parameters per object per band.
The assembly of the filtering and feature extraction functions constitutes a complete pipeline from raw data to ready to process data.
Applying a random forest algorithm offered mitigated results. Since the method isn't suited for anomaly detection the model is too conservative. It is predicting only one PISN and thus achieves 100% purity but misses 99.71 % of the PISNe. Using this model in real conditions might not provide any positive results.
On the other hand the isolation forest algorithm performs very well since it is capable of isolating a small fraction of a whole dataset that is very likely to contain high signal PISNe. With this method we can reasonably aim at selecting only 1 % of the original data base. It is definitely a powerful tool and is promising for real condition tests.
The next step is to complexify the task of the classifiers by splitting each objects into two randomly chosen chunks of 1 year. This will allow us to test the framework in a more similar data situation to that which will be experienced by LSST.
In the end, we will consider adding the whole pipeline to the broker Fink [14] and put it to the test against real incoming data. Hopefully one day this work will open the path to the discovery of the first pair instability supernova.
All the code used to produce these results are publicly available at `https://github.com/erusseil/PISN-classification`.

References

[1] Pair instability supernova. https://en.wikipedia.org/wiki/Pair-instability_supernova. Accessed: 2021-04-24. pages 8

[2] Zwicky transient facility. systematic exploration of the dynamic sky. https://www.ztf.caltech.edu/. Accessed: 2021-05-28. pages 4

[3] Mark Armstrong. Sn 1998aq. https://en.wikipedia.org/wiki/SN_1998aq#/media/File:SN1998aq_max_spectra.svg. Accessed: 2021-06-02. pages 6

[4] Yi Cao, Peter E. Nugent, and Mansi M. Kasliwal. Intermediate palomar transient factory: Realtime image subtraction pipeline. *Publications of the Astronomical Society of the Pacific*, 128(969):114502, Sep 2016. pages 4

[5] Alain Coc and Elisabeth Vangioni. Primordial nucleosynthesis. *International Journal of Modern Physics E*, 26(08):1741002, Aug 2017. pages 7

[6] The LSST Science Collaboration. Lsst science book, version 2.0, 2009. pages 4

[7] C. L. Fryer, S. E. Woosley, and A. Heger. Pair-instability supernovae, gravity waves, and gamma-ray transients. *The Astrophysical Journal*, 550(1):372–382, Mar 2001. pages 5, 7

[8] Matthew J. Graham, S. R. Kulkarni, Eric C. Bellm, Scott M. Adams, Cristina Barbarino, Nadejda Blagorodnova, Dennis Bodewits, Bryce Bolin, Patrick R. Brady, S. Bradley Cenko, and et al. The zwicky transient facility: Science objectives. *Publications of the Astronomical Society of the Pacific*, 131(1001):078001, May 2019. pages 4, 5

[9] Charles R. Harris, K. Jarrod Millman, Stéfan J. van der Walt, Ralf Gommers, Pauli Virtanen, David Cournapeau, Eric Wieser, Julian Taylor, Sebastian Berg, Nathaniel J. Smith, Robert Kern, Matti Picus, Stephan Hoyer, Marten H. van Kerkwijk, Matthew Brett, Allan Haldane, Jaime Fernández del Río, Mark Wiebe, Pearu Peterson, Pierre Gérard-Marchant, Kevin Sheppard, Tyler Reddy, Warren Weckesser, Hameer Abbasi, Christoph Gohlke, and Travis E. Oliphant. Array programming with NumPy. *Nature*, 585(7825):357–362, September 2020. pages 11, 13

[10] A. Heger and S. E. Woosley. The nucleosynthetic signature of population iii. *The Astrophysical Journal*, 567(1):532–543, Mar 2002. pages 7

[11] Mario Jurić, Jeffrey Kantor, K-T Lim, Robert H. Lupton, Gregory Dubois-Felsmann, Tim Jenness, Tim S. Axelrod, Jovan Aleksić, Roberta A. Allsman, and et al. The lsst data management system, 2015. pages 4

[12] R. Kessler, et al., LSST Dark Energy Science Collaboration, and Transient and Variable Stars Science Collaboration. Models and Simulations for the Photometric LSST Astronomical Time Series Classification Challenge (PLAsTiCC). *Publications of the Astronomical Society of the Pacific*, 131(1003):094501, September 2019. pages 5, 9

[13] Dennis D. McCarthy. The Julian and Modified Julian Dates. *Journal for the History of Astronomy*, 29:327, November 1998. pages 12

[14] Anais Möller, Julien Peloton, Emille E. O. Ishida, Chris Arnault, Etienne Bachelet, Tristan Blaineau, Dominique Boutigny, Abhishek Chauhan, Emmanuel Gangler, and et al. FINK, a new generation of broker for the LSST community. MNRAS, 501(3):3272–3288, March 2021. pages 5, 24

[15] Gautham Narayan. The plasticc astronomy starter kit. `https://www.kaggle.com/michaelapers/the-plasticc-astronomy-starter-kit`. Accessed: 2021-05-20. pages 4, 6, 10

[16] The pandas development team. pandas-dev/pandas: Pandas, February 2020. pages 11, 13

[17] Maria T. Patterson, Eric C. Bellm, Ben Rusholme, Frank J. Masci, Mario Juric, K. Simon Krughoff, V. Zach Golkhou, Matthew J. Graham, Shrinivas R. Kulkarni, George Helou, and et al. The zwicky transient facility alert distribution system. *Publications of the Astronomical Society of the Pacific*, 131(995):018001, Nov 2018. pages 4

[18] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830, 2011. pages 15

[19] The LSST project. Coronavirus update. `https://project.lsst.org/coronavirus-update`. Accessed: 2021-05-28. pages 4

[20] The PLAsTiCC team, The LSST Dark Energy Science Collaboration, The LSST Transients, and Variable Stars Science Collaboration. The Photometric LSST Astronomical Time-series Classification Challenge (PLAsTiCC): Data set. *arXiv e-prints*, page arXiv:1810.00001, September 2018. pages 7

[21] Guido Van Rossum and Fred L. Drake. *Python 3 Reference Manual*. CreateSpace, Scotts Valley, CA, 2009. pages 13

[22] Daniel J. Whalen, Joseph Smidt, Alexander Heger, Raphael Hirschi, Norhasliza Yusof, Wesley Even, Chris L. Fryer, Massimo Stiavelli, Ke-Jung Chen, and Candace C. Joggerst. Pair-instability Supernovae in the Local Universe. ApJ, 797(1):9, December 2014. pages 11

A Pair instability supernovae in the Deep Drilling Field

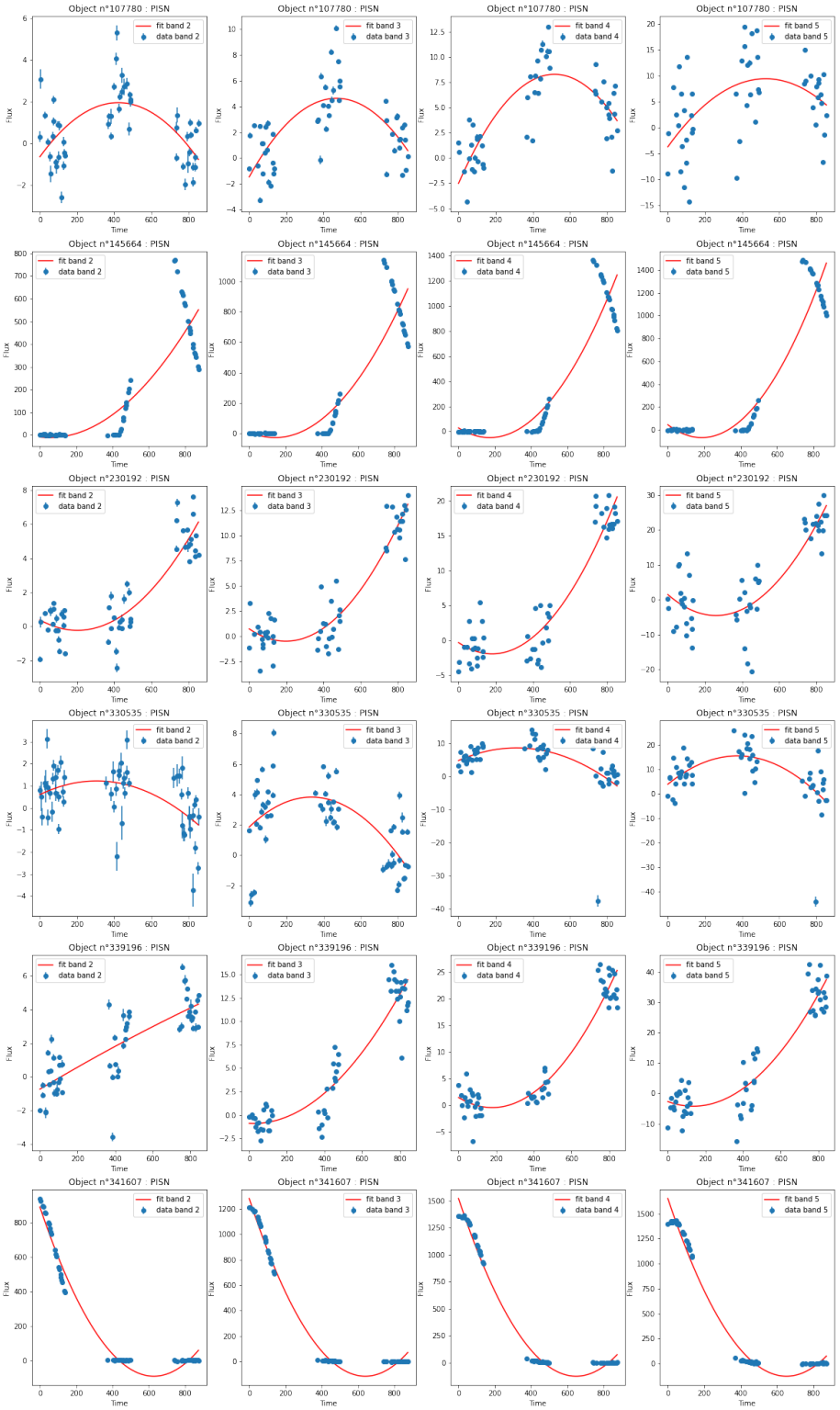

Figure (22) All pair-instability supernovae in the deep drilling field. Blue dots show measured points and red lines show results from the polynomial fit.