Astronomy ESFRI & Research Infrastructure Cluster

ASTERICS - H2020 - 653477

Automated Solar Radio Bursts Detection through Machine Learning

ASTERICS GA DELIVERABLE: D3.22

Document identifier:	ASTERICS-D3.22.doc
Date:	**17-05-2019**
Work package:	**WP3 - OBELICS**
Lead partner:	**S&T**
Document status:	**Final**
Dissemination level:	**Public**
Document link:	www.asterics2020.eu/documents/ ASTERICS-D3.22.pdf

Abstract

This report describes the applicability of using Machine Learning techniques to automatically detect Solar Radio Bursts in dynamic spectra. It describes the activities perfomed, the results and outlines next steps to be taken.

I. COPYRIGHT NOTICE

II. DELIVERY SLIP

	Name	**Partner/WP**	**Date**
From			
Author(s)	Bruno Rino	S&T / WP3	2019-05-14
Reviewed by	Erwin Platen	S&T / WP3	2019-05-15
Approved by	Edo Loenen AMST	S&T/WP3 ASTRON	2019-05-15 2019-05-16

III. DOCUMENT LOG

Issue	**Date**	**Comment**	**Author/Partner**
1	2019-04-24	Initial version	Bruno Rino / S&T
2	2019-04-29	Full revision	Bruno Rino / S&T
3	2019-04-30	Review	Erwin Platen / S&T

4	2019-04-30	Final Review for Release	Edo Loenen / S&T
5	2019-05-14	Revision after comments	Bruno Rino / S&T
6	2019-05-15	Review	Erwin Platen / S&T
7	2019-05-15	Final Review for Release	Edo Loenen / S&T

IV. APPLICATON AREA

This document is a formal deliverable for the GA of the project, applicable to all members of the ASTERICS project, beneficiaries and third parties, as well as its collaborating projects.

V. TERMINOLOGY

A complete project glossary is provided at the following page:
http://www.asterics2020.eu/glossary/

VI. PROJECT SUMMARY

ASTERICS (Astronomy ESFRI & Research Infrastructure Cluster) aims to address the cross-cutting synergies and common challenges shared by the various Astronomy ESFRI facilities (SKA, CTA, KM3Net & E-ELT). It brings together for the first time, the astronomy, astrophysics and particle astrophysics communities, in addition to other related research infrastructures. The major objectives of ASTERICS are to support and accelerate the implementation of the ESFRI telescopes, to enhance their performance beyond the current state-of-the-art, and to see them interoperate as an integrated, multi-wavelength and multi-messenger facility. An important focal point is the management, processing and scientific exploitation of the huge datasets the ESFRI facilities will generate. ASTERICS will seek solutions to these problems outside of the traditional channels by directly engaging and collaborating with industry and specialised SMEs. The various ESFRI pathfinders and precursors will present the perfect proving ground for new methodologies and prototype systems. In addition, ASTERICS will enable astronomers from across the member states to have broad access to the reduced data products of the ESFRI telescopes via a seamless interface to the Virtual Observatory framework. This will massively increase the scientific impact of the telescopes, and greatly encourage use (and re-use) of the data in new and novel ways, typically not foreseen in the original proposals. By demonstrating cross-facility synchronicity, and by harmonising various policy aspects, ASTERICS will realise a distributed and interoperable approach that ushers in a new multi-messenger era for astronomy. Through an active dissemination programme, including direct engagement with all relevant stakeholders, and via the development of citizen scientist mass participation experiments, ASTERICS has the ambition to be a flagship for the scientific, industrial and societal impact ESFRI projects can deliver.

VII. EXECUTIVE SUMMARY

Solar Radio Burst events are known to have a wide range of impact on (civil) services such as satellite communication, GPS, airport radars and many more. Automatically detecting Solar Radio Bursts (SRBs) on archival data provides the ability to reconstruct historical records of SRBs, which are of scientific interest. Applying the same technique on real-time data enables warning services to 'now-cast' of ongoing strong SRB events.

Automatically detecting SRBs is not a straight forward problem to solve. In order to have a reliable detector, it must be able to distinguish SRBs from other strong radio signals, like radio transmissions, radar signals, or interference. In this report we explore the possibilities of using Machine Learning (ML) techniques to develop such an automated SRB detection system. No evidence in literature has been found that this approach has been attempted before.

We developed a prototype identification system using a catalogue of SRBs that spanned four days with known high SRB activity. The catalogue was built by first applying a heuristic filtering and clustering algorithm. These clustered events were manually inspected by an expert to provide a sanitized and categorized list of events.

In the second part of the project, we used the sanitized list to train the machine learning algorithms. Here we applied a Machine Learning architecture designed for image categorisation (convolution neural networks) to detect those events.

Due to the short time frame for this activity, only a small catalogue of SRBs could be created to train the network, and this is the main issue to address going forward. With less than 100 Type III SRBs (both strong and weak), both the learning and validation phase of the Machine Learning model was less reliable.

However, despite the limited training data we show that the Machine Learning prototype is able to provide a good prediction of strong SRB events, and that automatic detection of SRBs using Machine Learning is a promising technique to investigate further.

Table of Contents

1. Introduction

Our current high-technology reliant society is fundamentally vulnerable for havoc causing space weather events. These space weather events can arise from Solar Outbursts like Solar Flares, Coronal Mass Ejections (CME) or Solar Radio Bursts (SRBs). They may affect almost all aspects of modern society, as for example the loss of Global positioning systems due to Solar Flares, has been estimated for the UK only up to 5 billion pounds over a five days' period. Solar Flares often (but not always) have a radio counterpart called Solar Radio Bursts. Broadly speaking, these Radio Outbursts can be considered both culprits (blocking Radio communications that affect Aviation, Radar and GPS) and proxies (certain types of SRB are part of the driving process that happen during a Solar Flare, that may lead to a CME) for Space Weather Forecasting Events.

LOFAR and SKA are undertaking key science programs with the aim of studying Solar Physics. A simple publication search shows that this is currently an active field of research with the goal to uncover the physical processes that lead to the development of Solar Outburst within the Solar Atmosphere. Solar Radio Bursts are particularly relevant for the radio domain. These are well known phenomena and have an extensive scientific history. The observation of SRBs with radio telescopes in combination with other, optical, UV, X-ray based, solar observations (multi-wavelength astronomy) allows to uncover the specific and complex plasma physical conditions and phenomena that are the causes of the various types of Solar Outbursts.

Solar Radio Outbursts are transient events, and to be able to study and monitor them with the current and future infrastructure (e.g. LOFAR and SKA) these require triggered Virtual Observatory (VO-)Events that are derived from dedicated systems that constantly monitor the Sun. The latter is currently only possible with smaller telescopes or single antenna observations, and the current infrastructure is not able to constantly monitor the high frequency regime (relevant for civil applications). Currently, ASTRON is actively researching the possibility to create a constant solar monitoring system using phased arrays. To be able to exploit such a monitoring system that is both of scientific and of civic interest we identify the following *problem statement*:

> "There must be an automated Software System that can produce accurate and trust-worthy SRB event recordings, to be used for scientific and civil purposes."

The *goal* of the project is to deliver a prototype implementation for the following end-product:

> "A Machine Learning based pipeline that analyses a given input stream of radio spectrum data and provides an output prediction of Solar Radio Weather Events, in particular to determine the detection and classification of an ongoing Solar Radio Burst."

The *aim* is to provide these products back as a service both to scientific and civil communities such that they can be used:

- To implement automatic event broadcasting. Such events are both of scientific interest for follow-up observations with for example LOFAR or SKA, and are of civil interest for early space-weather detection and prediction.

- To be able to reprocess historical archives and provide a sample of historically accurate classified Solar Radio Bursts.

The approach is based on Machine Learning (ML) Techniques. That is, previous classifications are used to train a ML-architecture.

2. Project plan

The following plan was devised to achieve the goal outlined above:

1. Identify suitable test datasets from archival Solar Radio Observations and archival sets of Solar Radio Bursts (classified or unclassified). For example, datasets from e-Callisto, Bleien Solar Radio Telescopes are all available online.

2. Retrieve, sanitize and homogenize the datasets to be able to serve as the ideal dataset for training of Machine Learning Algorithms.

3. Using the ideal dataset develop automated detection algorithms, that may involve:

    - Simple heuristic thresholding.

    - Adaptive thresholds (incl. filtering of the data).

    - Neural Network (NN) algorithms (Anomaly detection NNs, Multi-resolution NN.)

4. Apply the developed algorithm to a larger set of data. Cross-correlate with archival Solar Radio Burst events (such as provided by NOAA) for validation.

5. Iterate steps 3 to 4 to improve if necessary.

6. [Optional] Further improve the algorithm or create a different algorithm to classify the Solar Radio Burst. For example, algorithms such as Support Vector Machines allow for the ability to provide an automatic unsupervised classification.

3. Overview of performed activities

Dataset collection and sanitation

Supervised machine learning algorithms use previous classifications to train a ML-architecture. For our purposes, this consist of two datasets:

- Solar radio (sweep) observations from existing radio telescopes. This is the data in which the ML algorithm needs to detect the SRBs.
- An SRB catalogue used to train the ML algorithm on where SRBs occur in the data, including the following metadata:
    - time range (start and stop)
    - frequency
    - type

The e-Callisto project (http://e-callisto.org/) provides a very comprehensive set of observations, and provides data from many stations around the world. The data goes as far back as 1978, but is most consistent starting around 2013. It is freely available thanks to the Institute for Data Science FHNW Brugg/Windisch, Switzerland.

The U.S. National Oceanic and Atmospheric Administration (NOAA) provides a list of "solar and geophysical events" (https://www.swpc.noaa.gov/products/solar-and-geophysical-event-reports) which include SRBs with all the aforementioned metadata. The SRB event list is based on the Radio Solar Telescope Network, which is operated by the US military.

For this project, we focused on the time window between 2017-09-06 and 2017-09-09. This period was selected because it is a fairly recent period (thus ensuring good data quality) known for significant solar activity (maximizing the number of SRBs in the sample data set).

In order to train the ML algorithms on these data sets, a list of SRB events is needed. In the initial round, it was attempted to cross-match the e-Callisto data with the NOAA SRB event catalogue. Surprisingly, these two data-sets did not correlate well with each other. Although some matches were found, there were also a large number of mismatches (SRBs either only being detected in the data, or only present in the NOAA list).

Within the scope of this activity, there was no opportunity to fully understand where this unexpected dichotomy comes from. The subject, however, does merit further investigation, since the NOAA data-set is often used for statistical analysis of SRB distribution function (e.g. Nita et al 2001). The initial conclusions from the work performed here suggest that the observed mismatches are due to some of the characteristics of the NOAA catalogue, which makes it less ideal for our purposes:

1. The time resolution is 1 minute. This is sub-optimal: in the cases where an event lasts only a few seconds, the sampled data would consist mostly of non-eventful data.

2. Some events are listed with the same time stamp for the start and stop time. Looking at a few examples, it would be required to add 1 minute before and after the defined time in order to capture the event. This would however also capture a lot of non-eventful data. See Figure 1 for examples.

3. False negatives: several obvious events are not present in the NOAA dataset. This greatly reduces the ability to build an accurate ML model. For the weaker events this can be attributed to the inherent 500 SFU (solar flux unit) limit of the SRB event broadcasts (see Figure 2).

The figures below show several examples of e-Callisto data. E-Callisto data consists of radio flux density measurements over a 15 minute period along the wavelengths specific for each instrument. Events present in the NOAA dataset are marked with a small white triangle on the bottom of the plot. If the event has a set duration, a rectangle is displayed around the event.

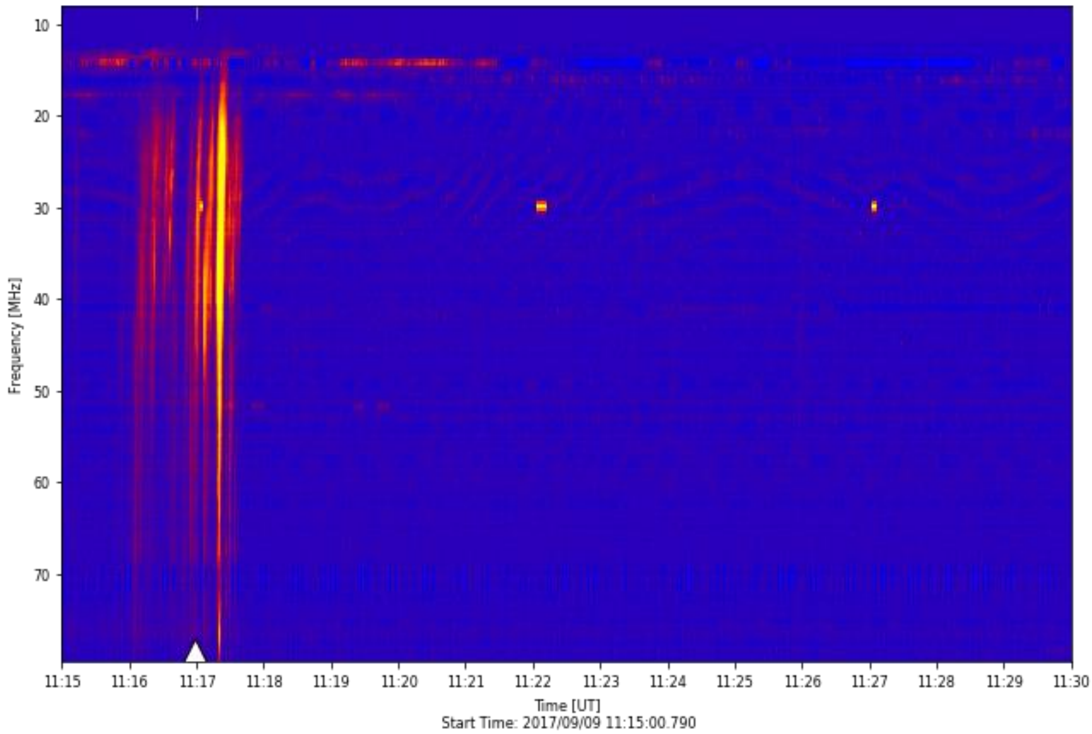

Figure 1: An example showing an event "without duration". The white triangle at the bottom of the graph marks the beginning and end of the burst as found in the NOAA SRB catalogue. The spectrogram data is from e-Callisto. In this case the event as reported on the event list occurs halfway of the burst as visible in the spectrogram.

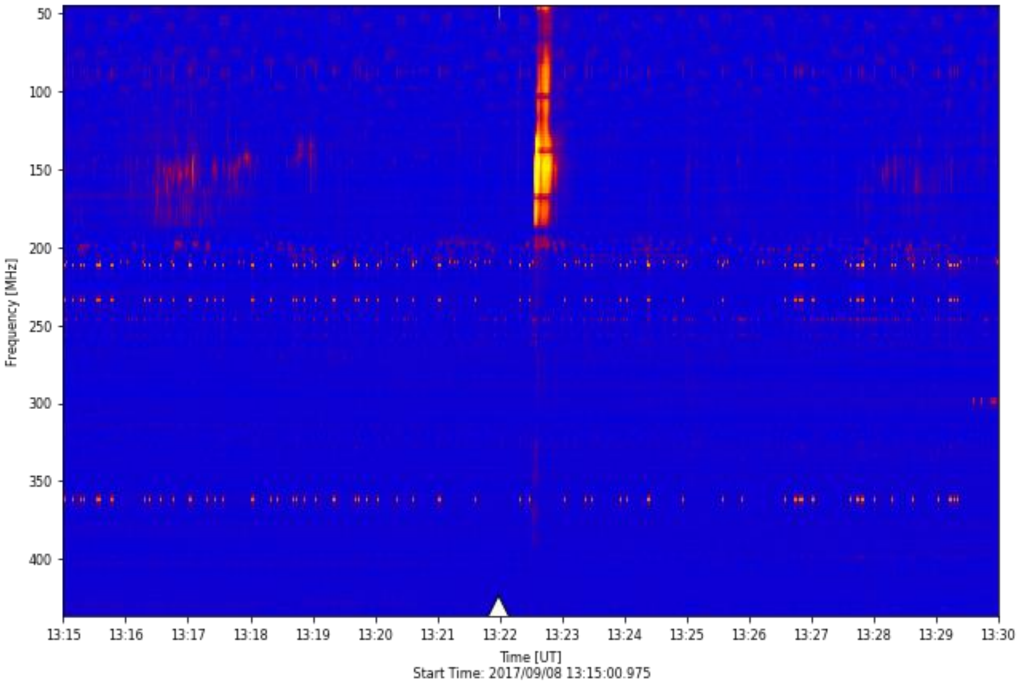

Figure 2: Another example of an event "without duration". In this case the event as reported on the event list occurs before the burst as visible in the spectrogram (but within one minute).

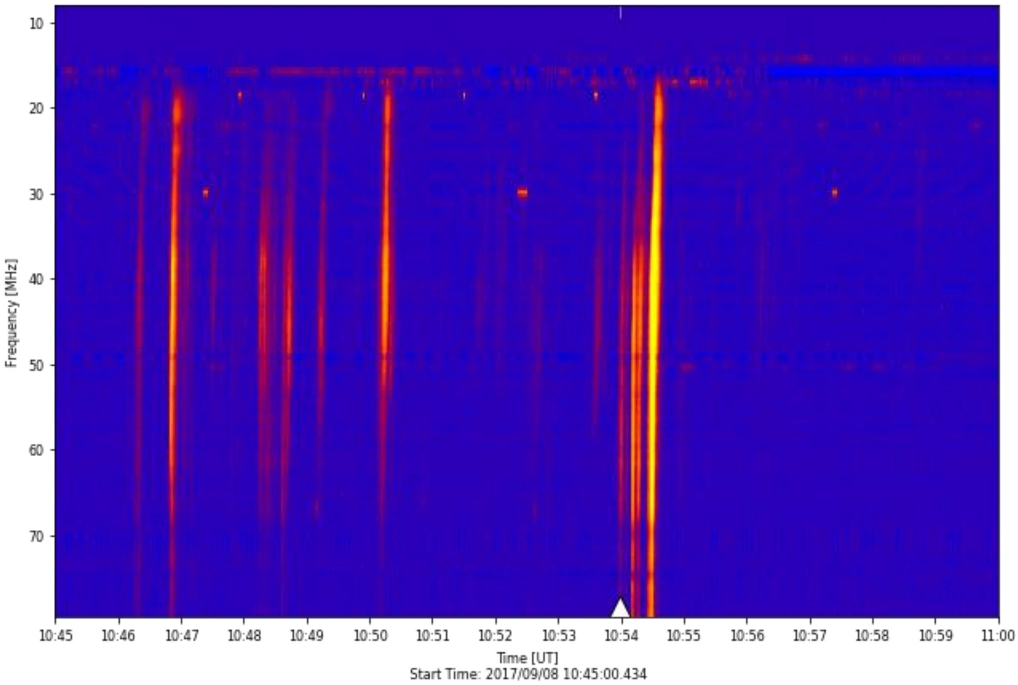

Figure 3: An example showing false negatives. The figure shows events that are detected in the data, but that were missed in the NOAA catalogue. Although the plot has one matching event, the image clearly shows multiple events that were not registered in the NOAA catalogue.

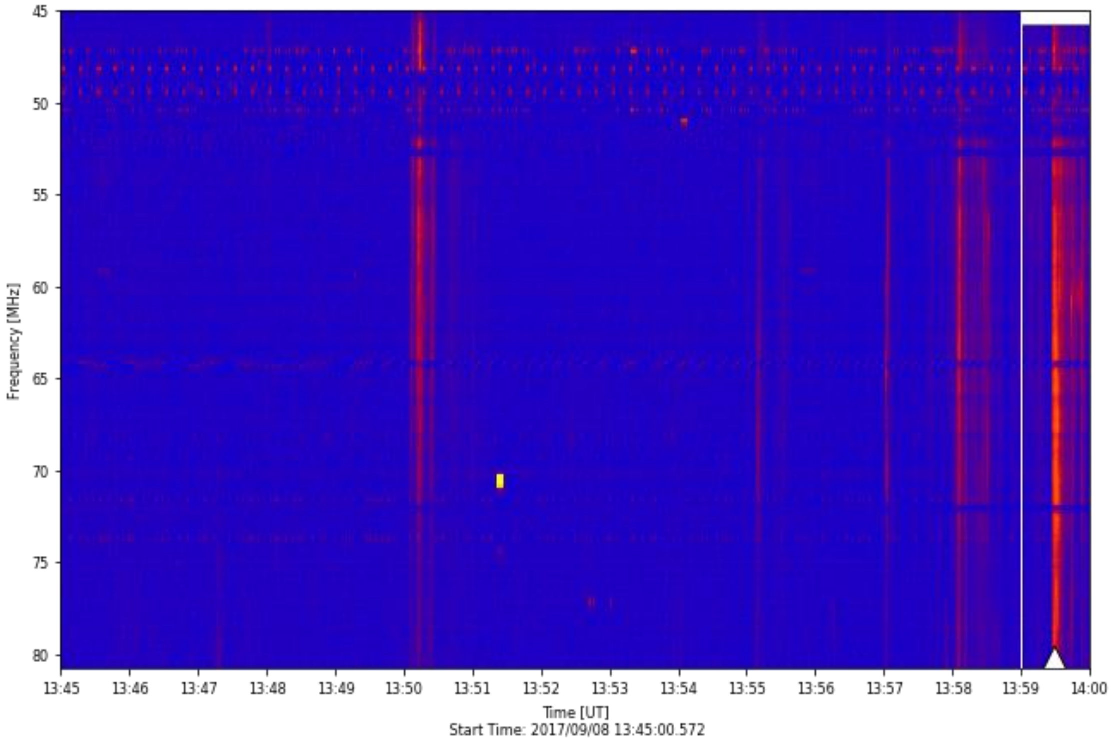

Figure 4: Another example showing false negatives. In this case, the plot has one matching event that lasts one minute, but previous events were not registered in the NOAA catalogue.

Given the NOAA dataset turned out to be unsuitable for the objectives of this activity, we had to resort to build our own dataset suitable for using machine learning algorithms.

Our approach was the following: first provide a coarse-grained list of events, which was then sanitized by manual inspections. In this first phase we build a heuristic-based clustering detection algorithm to identify potential SRBs events. This algorithm was initially set up to detect Type III SRBs, since they are the easiest to detect due to their characteristic spectral features. As can be seen in Figure 5, Type III SRBs are typically short lived, and span a relatively large spectral range. This simple spectral “shape” makes them easy to filter from the data.

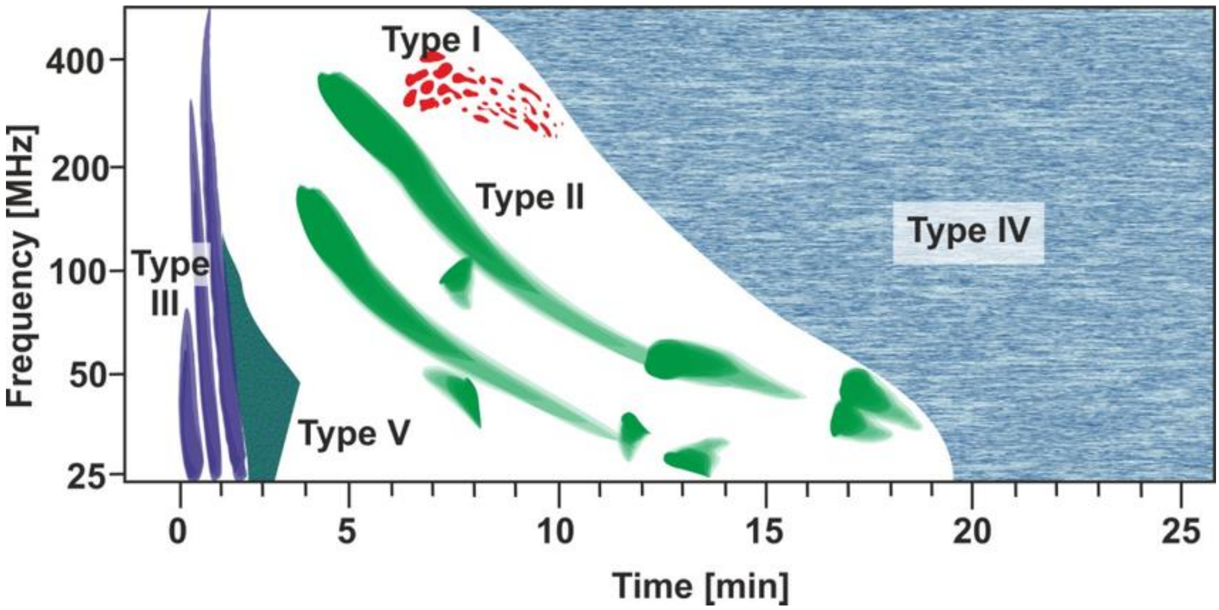

Figure 5: Illustration of the spectral features of different types of SRBs.

Based on these spectral characteristics, the algorithm used the following approach:

1. Find flux values higher than the median by some factor (typically 1.5) of the standard deviation. The median is determined within each frequency range within a predefined interval.

2. Cluster high flux values to form the SRB event outline within the spectrogram.

To find high flux values, one of the following heuristic was used:

- *highlighter_stddev_freq*: The standard deviation is calculated for each frequency range within the spectrum.

- *highlighter_stddev_total*: The standard deviation is calculated using all flux values in the spectrum.

- *highlighter_stddev*: Keep the standard deviation low by removing high flux values from median and standard deviation calculation (this to remove outliers, caused by interference, from the data).

- *highlighter_stddev_autodetectnoise*: Keep the standard low by removing high flux values from median and standard deviation calculation (as above), as well as frequencies with more than 15% of high flux values (spectral channels severely affected by interference).

To cluster high flux values, the following heuristics were used (derived from typical Type III SRB properties):

- The frequency span of a cluster should be higher than some threshold (typically 10 Mhz).
- The time span of a cluster should be longer than some threshold (typically 3 seconds).
- Allow some tolerance (i.e. there might be small gaps in a cluster): for up to 2 pixels gap within the span

Using a selection of e-Callisto instruments that are expected to be among the more reliable (BIR_02, BIR_03, BLEN5M_03, BLENSW_01, BLENSW_02, GLASGOW_59, HUMAIN_59), the detector algorithm was applied.

The default parameters for the algorithm were:

- highlighter: data should be more than 1.5 times the standard deviation, using the *highlighter_stddev* routine.
- time threshold: 3 seconds
- frequency threshold: 10 MHz

For some instruments, some customisation was needed to get the best results:

- BIR_02:
    - highlighter: more than 2 times the standard deviation, using the *highlighter_stddev* routine.
- BIR_03:
    - remove frequencies above 385 Mhz (they are consistently noisy)
- BLENSW_01 and BLENSW_02:
    - highlighter: Use the *highlighter_stddev_autodetectnoise* routine, with a standard deviation threshold of 1.
- HUMAIN_59:
    - frequency threshold: 30 MHz

This resulted in a list of potential SRB events for each instrument, to be confirmed by manual inspection.

The manual inspection resulted in a clean dataset, where the following types of event were identified:

- Type III
- Type IIIs (several Type III events in a single detection)
- Noise storm
- RFI (Radio Frequency Interference)

Other types of events were not labelled, either because none were present or because the heuristic-based detector was not setup to identify them.

The manual inspection proved to be very time consuming. It took several working days for a SRB specialist to inspect the first two instruments (BIR_02 and BIR_03). For this reason, only those two instrument were inspected:

	BIR_02	**BIR_03**
Type IIIs	6	1
Type III	88	84
RFI	82	3739
Noise Storm	485	505

Table 1: Number of events for each instrument, by type. The heuristic-based clustering detection settings were more stringent for BIR_02 than BIR_03 and that shows: a lot more candidates were labelled as RFI for BIR_03.

For each instrument, the final data set contained fewer than 100 SRBs, which is usually quite low to train a ML algorithm. A common rule of thumb is that 1000 samples is needed for a first run. Ultimately the initial results of the training will determine what the required number is.

Given the limited time available for the activity, it was unfortunately not possible to create a better dataset. However, as will be shown in Section 4 - Analysis of the results, the results are still very promising despite the limited data set.

Detection algorithm

Given that the SRB catalogue only contains Type III events, the ML model was also targeted to identify Type III events.

Given the parameters of the heuristic-based detector algorithm, the shortest events are 3 seconds long. This was taken as the sample size to feed to the ML. A small sample size like this has several benefits:

- In a real-time scenario, a prediction can be produced within little more than 3 seconds after the event started.
- Given the small number of SRBs in the catalogue, longer events can be “sliced” into several events to artificially increase the number of samples.

There is, however, also a downside:

- Slicing the events may remove characteristics that might be important for detection.

The detection algorithm is built using a Convolutional Neural Network (CNN), a ML architecture commonly used in image classification. In a CNN, the features that define the classification of the samples are not pre-determined, but rather the CNN itself learns how to extract those features. This has a big advantage: there is no need to a human expert to define and/or choose what features might be interesting. The learned features are very abstract however, and it is not possible for a human to always understand exactly what drives the classification.

In a CNN, the input layer to the network consists of a node for each pixel. The nodes in the convolutional (hidden) layers are not simply connected to all of the previous layer nodes like in a regular Neural Network. Because they are optimised for images, CNN nodes compose their input into small tiles of neighbouring pixels. Each node then learns some feature of the image. In the first hidden layer, edges and patches of light are common features. Features in subsequent layers are generally more abstract and undecipherable. Features are learnt by means of a filter: the filter allows the tiles that contain the feature, while other tiles are blocked by the filter. The output of the filter is then transformed by an activation function, which output the probability that the feature is present in the input. The output of each convolutional layer is then downsampled by means of a pooling layer, constraining the computational cost and overfitting in subsequent layers.

A final (fully connected) layer aggregates the activations of each learned feature and distils them into a probability that the image provided as input to the CNN is a positive match.

A good introduction to CNNs can be found here: https://cs231n.github.io/convolutional-networks/

The architecture that was implemented consists of three convolutional layers:

- Layer 1: 16 filters
- Layer 2: 32 filters

- Layer 3: 64 filters

All filter tiles are 3x3. Each convolutional layer is followed by a max-pooling layer with a 2x2 window. The final (fully connected) layer with ReLU activation with 512 hidden units follow, and the output layer uses a sigmoid function, which produces the confidence level of the classification.

4. Analysis of the results

The ML models achieved at best 95% accuracy for BIR_02 and 90% accuracy for BIR_03 on the validation data. This applies to slices of events. Since strong Type III events contribute more slices, this accuracy might be biased towards strong Type III events.

To analyse the accuracy in a more real setting, we saved one day for testing the model and trained the model on the remaining 3 days of data. This provides a more realistic testing, even if it is less statistically significant. Using a random test dataset would be a more statistically significant approach, however that advantage would be undermined by the small size of our dataset (less than 100 SRBs).

The testing showed that:

- The BIR_02 model was quite accurate at predicting strong events, but had quite a few false positives and false negatives for fainter events. Tweaking the confidence level of the model to 95% greatly reduced the false classifications. The images below show the detection (left) versus the SRB catalogue (right).

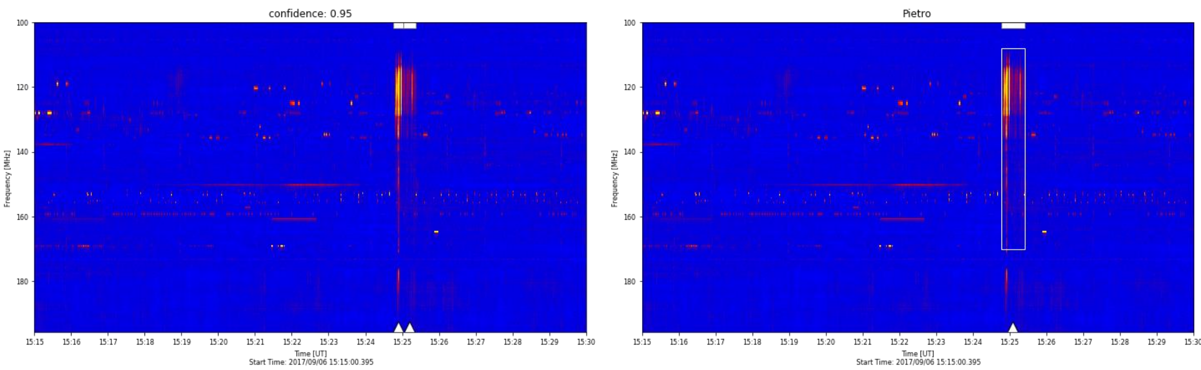

Figure 6: Strong event detected, the left image shows the ML classified events (indicated by arrows) and the right hand side shows the manual classification of the SRB event.

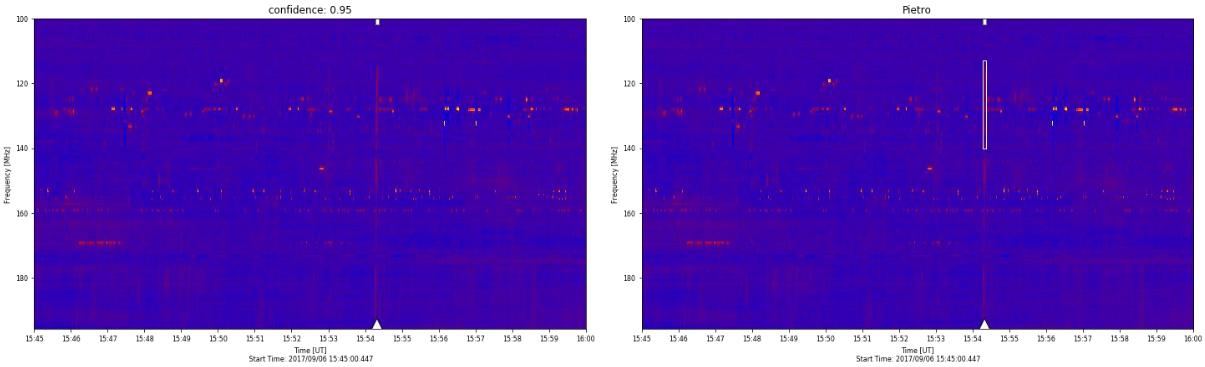

Figure 7: Very faint event detected. Similar to Figure 6, the left image shows the ML classified detection and the right hand image the manually classified example.

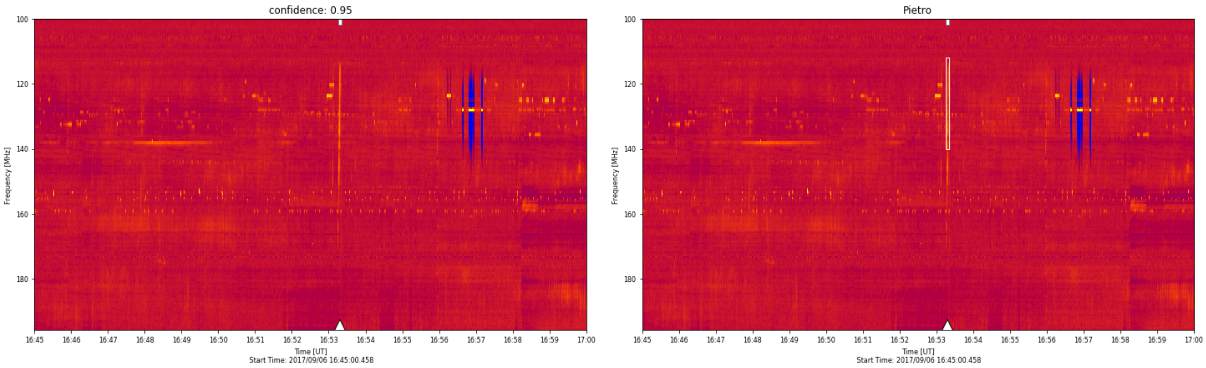

Figure 8: Weak event detected. Again, the left image shows the ML classified detection and the right hand image the manually classified example.

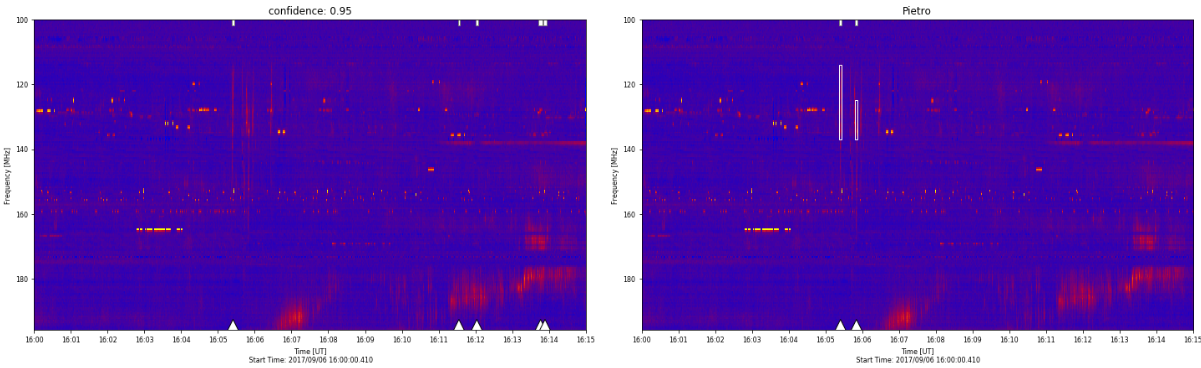

Figure 9: Weak event detected, but false positives (arrows on the right hand side of the image) and negatives (missed the 2nd event) are also present.

- The BIR_03 model was not as accurate. This is likely due to high levels of noise of the BIR_03 dataset. A much larger dataset would help reduce the impact of noise on the data.

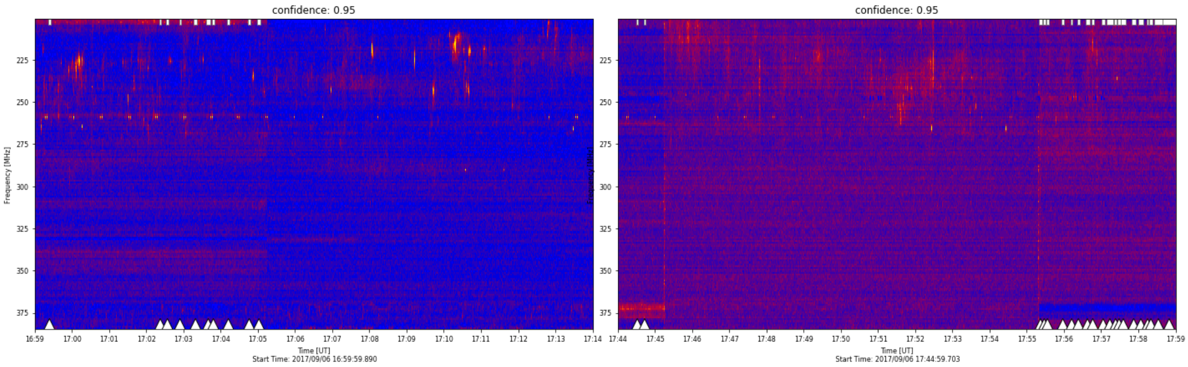

Figure 10: Two different samples showing noise and artefacts present in the BIR_03 data. It shows that many of those are incorrectly identified as SRBs.

Note that we have not used a formal definition for strong versus faint events: strong events are easily discernible, given their high flux values and duration. In our test dataset, BIR_02 has 9 such events, and all but one of the shortest was correctly identified.

During the analysis of results, it emerged that on some events were wrongly labelled (noise storm was labelled as Type III). We have not cross-validated the entire dataset, but the presence of misclassifications on the few samples observed reduced our confidence in our SRB catalogue. It also highlights the importance of quality assuring this dataset in future endeavours.

5. Deviations from the original project plan

In our original plan, we had intended to use the NOAA catalogue as input for the ML training. Unfortunately, when we inspected the event catalogue in more detail, it turned out to be of low-resolution and not very accurate when directly compared to the e-Callisto data-set. Therefore, we had to construct an SRB catalogue from scratch and start our own bootstrapping procedure.

The heuristic detection algorithm worked quite well at providing the first filtering of potential events. And, while it can be used on its own for SRB detection, we used it as the first step for bootstrapping the ML training process. Thereafter, a manual identification process was required to build a high confidence set of SRB events. Though the process was successful, it was more time-consuming then anticipated a priori, and less data could be accumulated to train the ML algorithms.

6. Summary of project results

The ML prototype for detecting SRBs proved quite accurate at detecting strong Type III events, especially considering the limited training data-set used for the prototype. The current detection-algorithm becomes less efficient for detecting weaker events. This can be expected since under very noisy conditions, any model will start to perform poorly. Our current training sample set is not large enough to provide exhaustive conclusion on the limits of the current approach.

Our analysis provides a general way forward on how to bootstrap the machine learning approach for SRB detection. First build a coarse-grained detection algorithm to provide a list of candidate events on a small set of data. This set can be further refined by manual inspection. This will yield a small but robust data-set. Then the ML algorithm can be trained on that small data set. Thereafter, use this on a larger data-set with the trained algorithm and start an error correcting procedure on these detections by manual inspection (or use external lists for validation if available). The latter step should be less time-consuming since the algorithm will already exclude most of the RFI.

7. Next steps

The current project has shown that a ML based SRB detection can be accurate, but is depending on good input/training data. The ML detection model would therefore benefit from a better SRB catalogue:

- Larger training set: more sample events are needed to achieve better accuracy
- More accurate training set:
    - our approach of an automated first pass produced a dataset that ignored some of the weaker events
    - the tedious nature of manual classification lends itself to human errors
- more event types, and subtypes (e.g. a strong type III is much different from a weak type III): this would allow us to fine-tune the algorithms

These recommendations and their relative importance will vary depending on the application and use-cases. Broadly speaking the applications that are in scope for this approach are both for scientific and for civil applications.

For scientific purposes it is of interest to detect all types of events, even the weakest events. These can be detected with our approach, however the training data-set needs to be expanded to be able to accurately start detecting these events.

For civil and commercial applications, such as a real-time warning service, ***only*** the strongest events are important. In this situation we can confirm that our current approach is able to classify and detect these strong events, given that the detection of the strong events was already 95% to 90% accurate with the limited dataset we used. However, we need to build up the statistics for stronger events, since these are also the rarest events. So also for this use-case a reliable input set with strong events is required.

In particular, attention should be paid to make the SRB catalogue less tedious to produce, both to improve its accuracy and to allow to create a larger SRB catalogue. This may require a graphical classification application to be created, which allows easy human feedback to be able to further refine the SRB model.

It will be interesting to use measurements from several similar instruments (and possibly variations of those measurements, e.g. different calibrations) that operate on the same wavelengths to train a single ML model. Given enough variety of instruments, it is expected that the resulting model could be used to accurately classify events on new but similar instruments.

8. Conclusions

Although the time available for this project was brief, and some challenges needed to be overcome (in particular related to the training data), the overall outcome of the project is very positive. We have successfully developed a ML algorithm to detect SRBs, and have shown that the novel approach of using Machine Learning based detection algorithms to automatically detect SRBs is feasible. This result opens up the possibility to construct new, reliable, automatic detectors for SRBs.