

Thesis Summary

The **Standard Model of Cosmology** predicts that the present Universe began some 14 billion years ago through a mathematical singularity, the Big Bang, and started expanding. Matter and radiation were, at the beginning, a hot plasma in thermal equilibrium. Due to the expansion, this primordial plasma gradually cooled down and, after around 380,000 yr, it was cold enough for atoms to form. The sharp decrease in the cross section of Thomson scattering permitted the free propagation of photons. To date, many experiments have been able to measure these primordial photons, also known as **Cosmic Microwave Background** (CMB).

The space mission **Planck**, by the *European Space Agency*, has given to date some of the most accurate measurements of CMB anisotropies, differences in temperature of the photons across the sky. Planck implemented two instruments, which probed the sky in two different frequency ranges: **LFI** covered the 30–70 GHz range, and **HFI** the 100–850 GHz range. LFI, *low frequency instrument*, used radiometers kept at 20 K. HFI, *high frequency instrument*, consisted of 48 bolometers, nearly perfect absorbers kept at 0.1 K and connected to solid state thermometers that measure changes in temperature of the absorber itself.

While both instruments were prone to systematics due to **cosmic ray** hits, this was particularly true for HFI. When cosmic rays managed to reach HFI bolometers, they induced a signature in the temperatures acquired by the thermometers. These signatures are labeled as **glitches**. The Planck/HFI collaboration already managed to remove these glitches from the raw data using an algorithm based on the *rms* measured in the timelines.

The target of my thesis is to analyze and classify the signature of cosmic rays in the timelines using an alternative approach, based on **Convolutional Neural Networks**.

A **Neural Network** is made of a set of *layers*, and produces an output starting from some input, in a deterministic way. The network computes the output by means of numbers, called *weights*, associated to each layer. The values of the *weights* are set through a process called **learning**. Some **training data**, which is classified by hand, is given to the network, which produces the output. In order to learn, the output is compared to the expected result, and a *loss score* is computed. An *optimizer* changes the weights to get a better score. The training is repeated many times, until some convergence criterion is reached. If the process is successful, the network will be able to produce sensible output from new unseen data, provided that its features are similar to that of the training data.

I chose a **Convolutional Neural Network** because of the following advantageous feature: *translational invariance*. This means that, when the network learns a pattern in one part of a timeline, it will be able to recognize it anywhere else. My CNN is a classification network, as its purpose is to classify its input (a timestream of temperatures) as *glitches* or not.

I personally prepared the test dataset manually, classifying 2000 datstreams acquired by HFI during the first days of the Planck mission. Of these, 400 were used as *validation*: their purpose was to check the performance of the network on data not utilized in the training. This method permits to detect *overfitting*, a situation in which the network performs well on the training set but poorly on unseen data, because of an excessive specialization.

I built the **Convolutional Neural Network** using 6 layers. I optimized the structure of the CNN by training it multiple times on different configurations, until I found the one that maximized the accuracy. The final **accuracy** reached, both on the training set and the validation set, is 97.71%. In the image below (Fig. ??) I show two timeseries classified correctly.

At last, I classified 100 timelines for testing the network on new data. To compare it with the approach followed by the Planck/HFI collaboration, I wrote a program that identifies the presence of a glitch using an *rms* threshold. This algorithm divides each timeline into 7 batches (number obtained through some optimization), it takes the highest *rms* between the 7 divisions and compares it with a threshold. To discriminate, I used the lowest of the values computed from timelines classified as *glitches*. On these new 100 testing timestreams, the *rms* program classified correctly 97 of them, while the network managed to label all of them as I did.

In conclusion, in this thesis I showed a different type of approach to glitch classification. My **Convolutional Neural Network** managed to achieve a result above 95% on every instance it was deployed when recognizing *glitches*, even if trained on a small sample of data. My work is important in the context of future CMB missions based on bolometers, like the JAXA **LiteBIRD** mission, adopted in May 2019 and scheduled for launch in 2027.

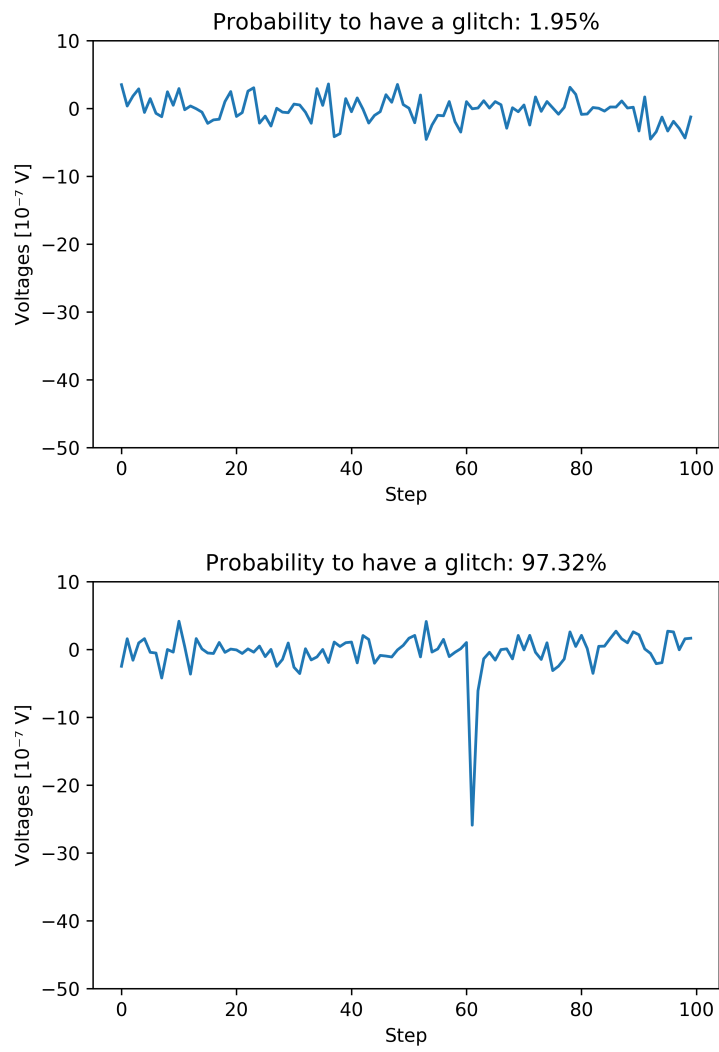


Figure 1