Training dark matter classifiers with machine learning using ATLAS Monte Carlo data

Introduction

Dark matter is generally considered a large scale feature of physics however it is possible to look for it within elementary particle interactions, more precisely the theoretical dark matter particle called a "weakly interacting massive particle" (WIMP). This project is to perform searches for the dark matter + mono-Z process ($pp \rightarrow \chi\bar{\chi} + Z$) and is identified by missing energy (due to the dark matter particles) and two opposite charged leptons (See Fig. 1), which is detected by the ATLAS detector in the LHC.

The goal of this project was to develop machine learning models called binary classifiers, which are trained to identify dark matter events from the background events. This was then used in an interactive Jupyter notebook designed to teach six form students the fundamental concepts of machine learning and how it can be used in high energy physics, worked on by me and the team at the University of Sussex (See Fig. 2). The models where developed in python using sciKit learn.

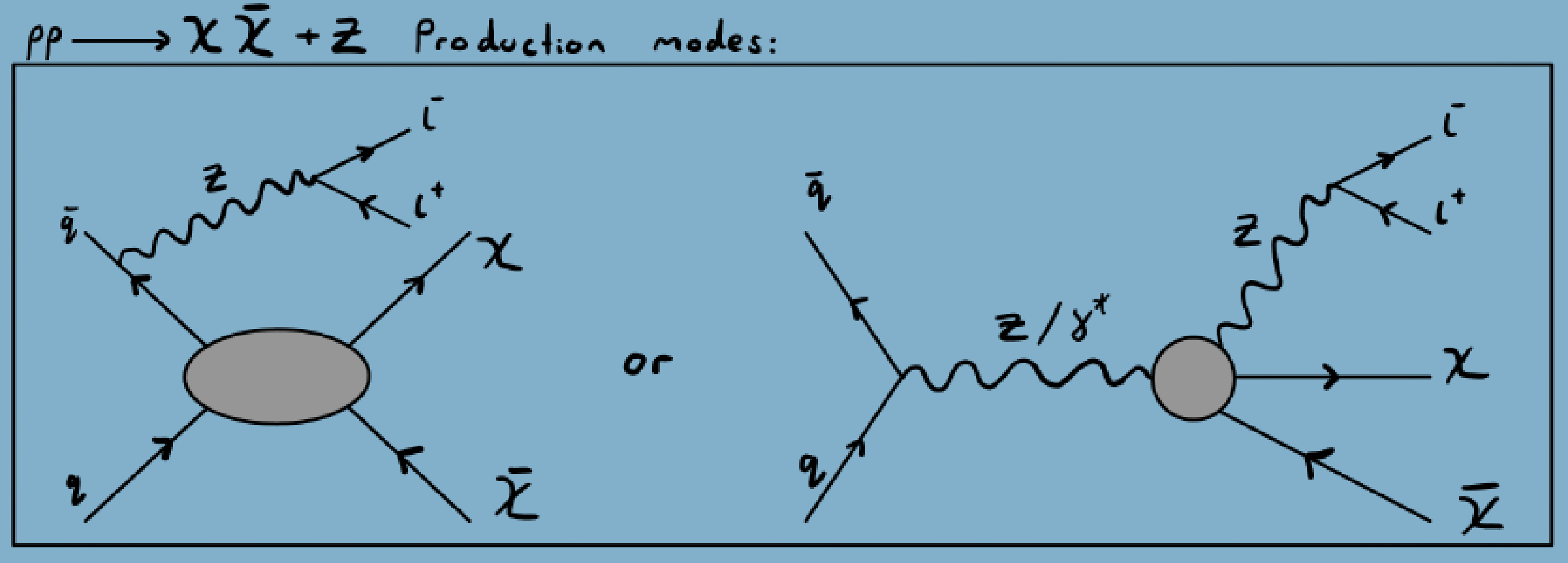
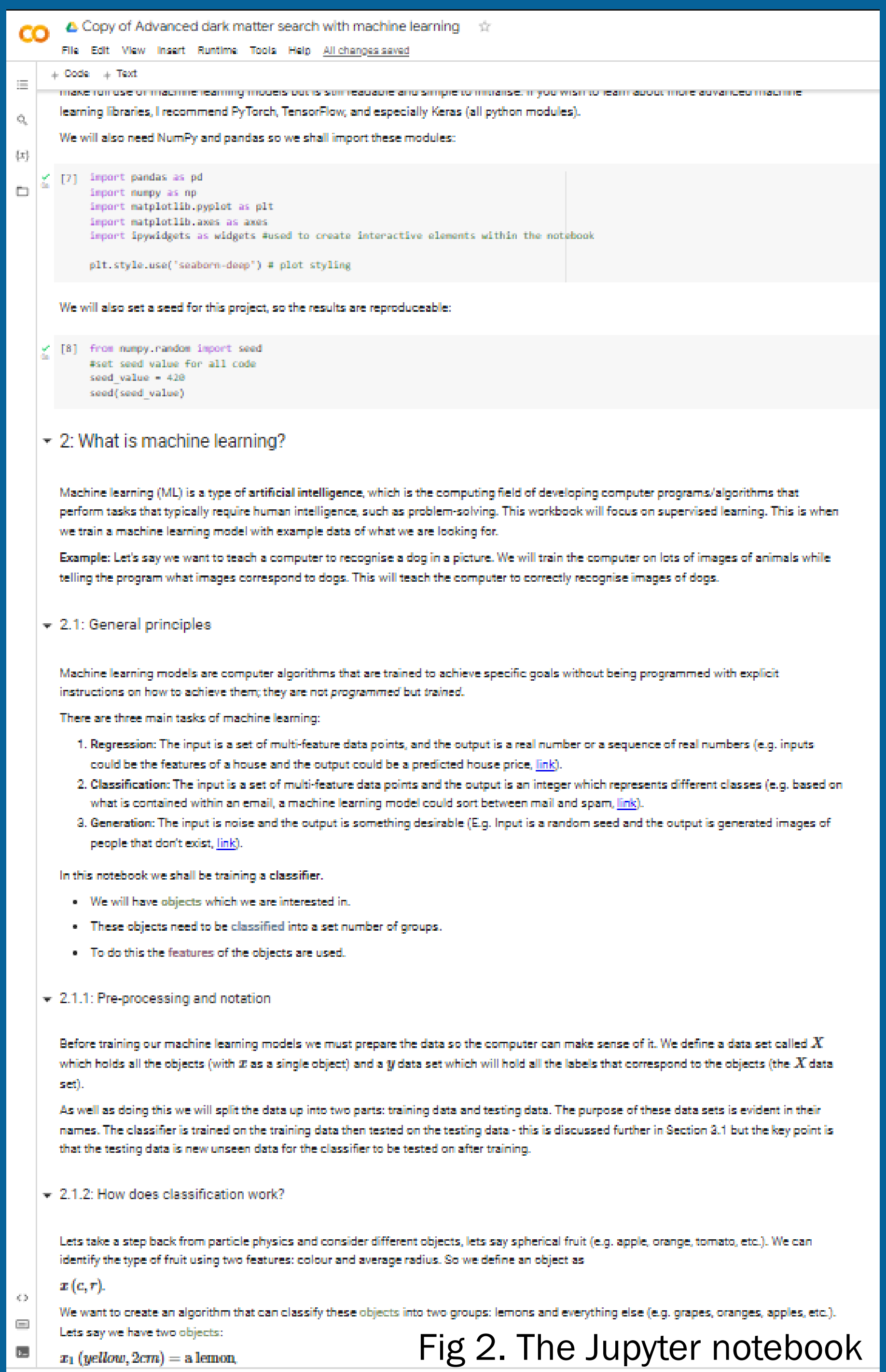


Fig 1. Possible Feynman diagrams for the dark matter + mono-Z process



Copy of Advanced dark matter search with machine learning

make full use of machine learning models but is still readable and simple to initialise. If you wish to learn about more advanced machine learning libraries, I recommend PyTorch, TensorFlow, and especially Keras (all python modules).

We will also need NumPy and pandas so we shall import these modules:

```
[7]  import pandas as pd
     import numpy as np
     import matplotlib.pyplot as plt
     import matplotlib.axes as axes
     import ipywidgets as widgets #used to create interactive elements within the notebook

     plt.style.use('seaborn-deep') # plot styling
```

We will also set a seed for this project, so the results are reproduceable:

```
[8]  from numpy.random import seed
     #set seed value for all code
     seed_value = 420
     seed(seed_value)
```

2: What is machine learning?

Machine learning (ML) is a type of **artificial intelligence**, which is the computing field of developing computer programs/algorithms that perform tasks that typically require human intelligence, such as problem-solving. This workbook will focus on supervised learning. This is when we train a machine learning model with example data of what we are looking for.

Example: Let's say we want to teach a computer to recognise a dog in a picture. We will train the computer on lots of images of animals while telling the program what images correspond to dogs. This will teach the computer to correctly recognise images of dogs.

2.1: General principles

Machine learning models are computer algorithms that are trained to achieve specific goals without being programmed with explicit instructions on how to achieve them; they are not *programmed* but *trained*.

There are three main tasks of machine learning:

1. **Regression:** The input is a set of multi-feature data points, and the output is a real number or a sequence of real numbers (e.g. inputs could be the features of a house and the output could be a predicted house price, link).
2. **Classification:** The input is a set of multi-feature data points and the output is an integer which represents different classes (e.g. based on what is contained within an email, a machine learning model could sort between mail and spam, link).
3. **Generation:** The input is noise and the output is something desirable (E.g. Input is a random seed and the output is generated images of people that don't exist, link).

In this notebook we shall be training a **classifier**.

- We will have objects which we are interested in.
- These objects need to be classified into a set number of groups.
- To do this the features of the objects are used.

2.1.1: Pre-processing and notation

Before training our machine learning models we must prepare the data so the computer can make sense of it. We define a data set called X which holds all the objects (with x as a single object) and a y data set which will hold all the labels that correspond to the objects (the X data set).

As well as doing this we will split the data up into two parts: training data and testing data. The purpose of these data sets is evident in their names. The classifier is trained on the training data then tested on the testing data - this is discussed further in Section 3.1 but the key point is that the testing data is new unseen data for the classifier to be tested on after training.

2.1.2: How does classification work?

Lets take a step back from particle physics and consider different objects, lets say spherical fruit (e.g. apple, orange, tomato, etc.). We can identify the type of fruit using two features: colour and average radius. So we define an object as

$x(c, r)$.

We want to create an algorithm that can classify these objects into two groups: lemons and everything else (e.g. grapes, oranges, apples, etc.). Lets say we have two objects:

$x_1(yellow, 2cm) =$ a lemon,

Fig 2. The Jupyter notebook

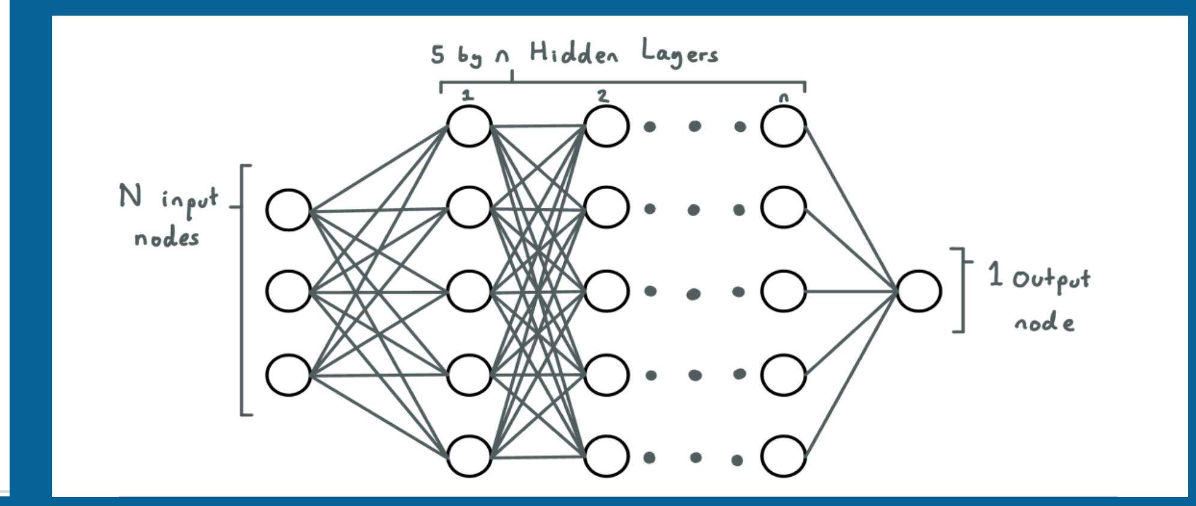
ATLAS open data:

ATLAS Open Data provides data from Monte Carlo simulation of the proton-proton collision data collected by the ATLAS experiment for public use, this contains a huge number of labelled collision events including dark matter used in this training.

Why?

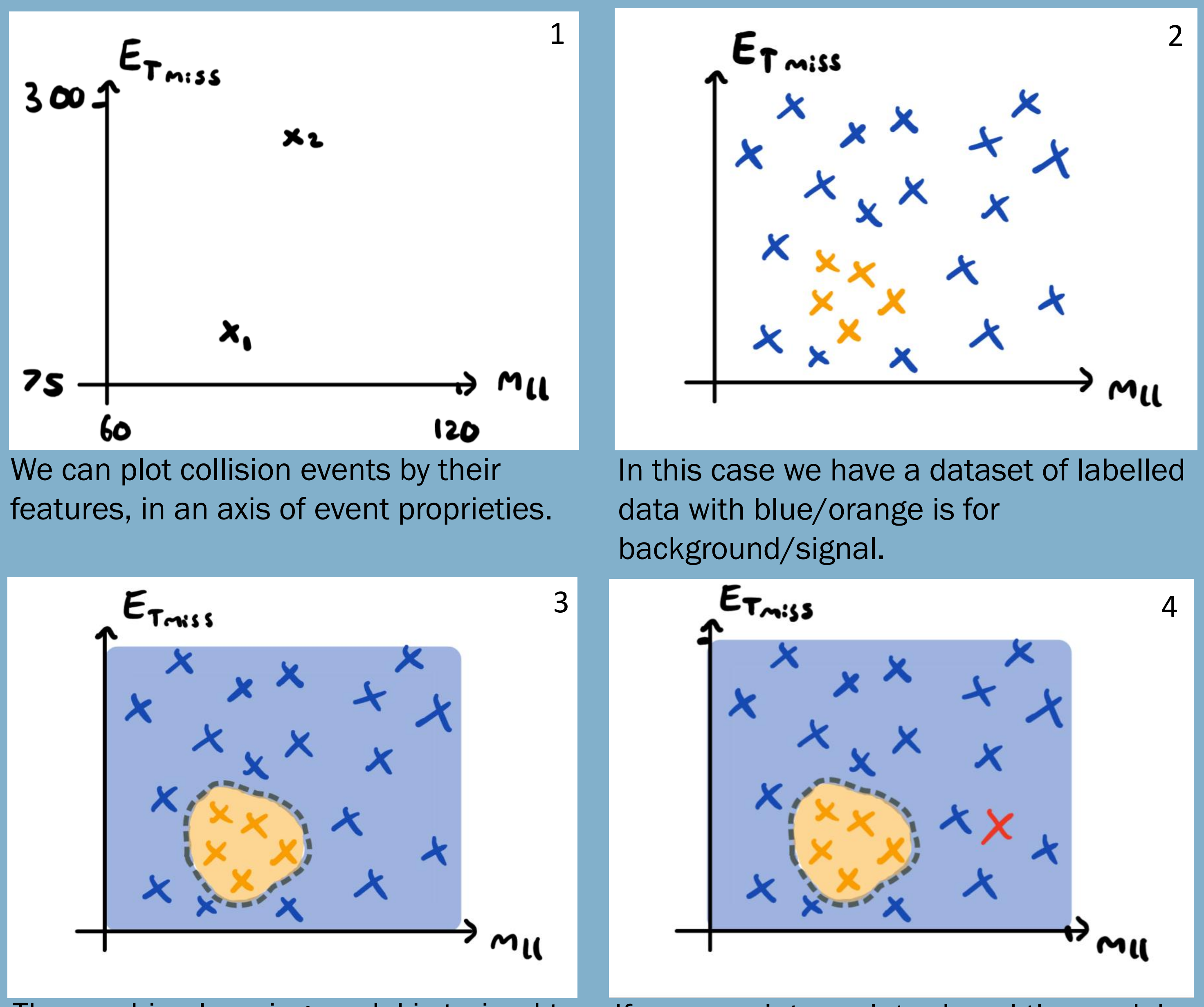
- Experimental particle physics involves a lot of data analysis. The LHC produces up to **1 billion proton-proton collisions** per second.
  - Around one petabyte per day, or **106 gigabytes**
- The efficiency and speed of machine learning methods also results in a massive decrease in computing time compared with manual methods

Neural Network Diagram



Random Forests and neural networks where the machine learning models used in this project.

How does classification work?



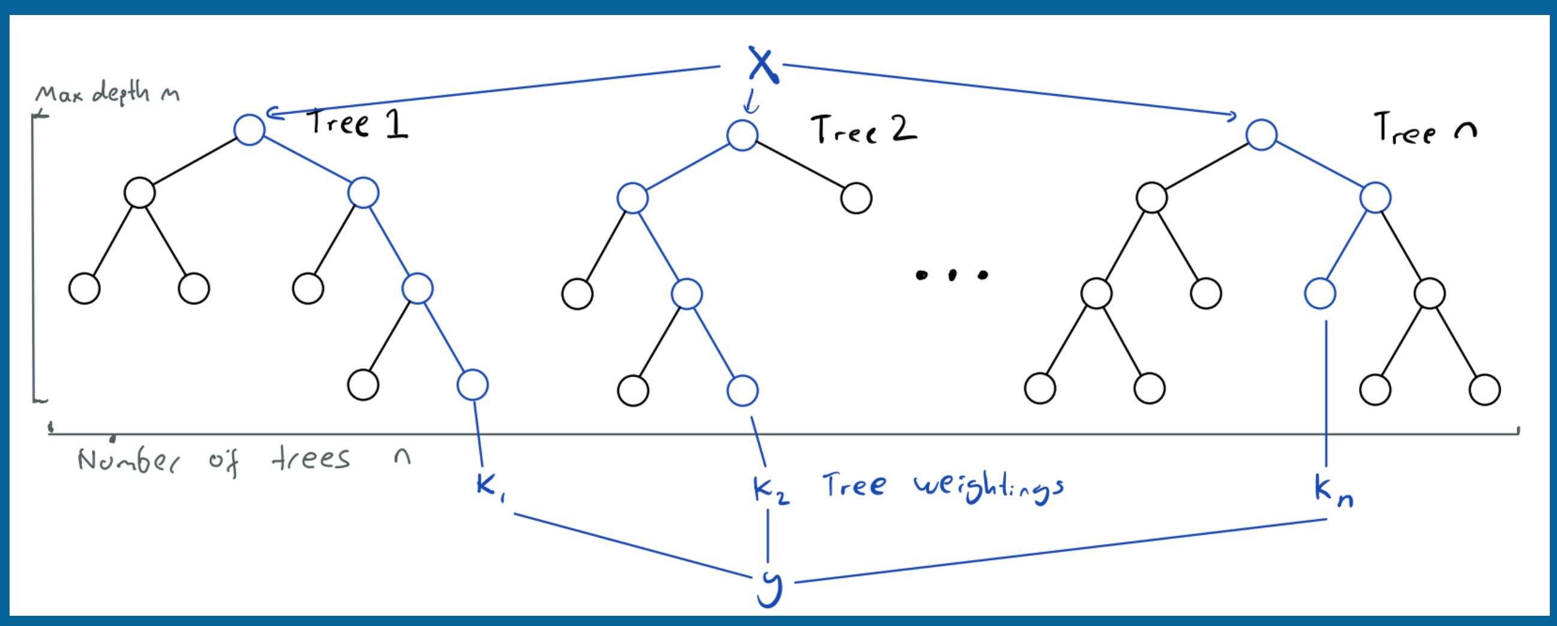
We can plot collision events by their features, in an axis of event proprieties.

In this case we have a dataset of labelled data with blue/orange is for background/signal.

The machine learning model is trained to produce boundaries classifying the points.

If a new point was introduced the model could then predict what it was, here it would predict background.

Random Forest Diagram



Significance Graphs →

A range of significance is found by performing a scan cut on the classifier response (Fig. 3&4). This measures the significance of the dark matter signal over the background.

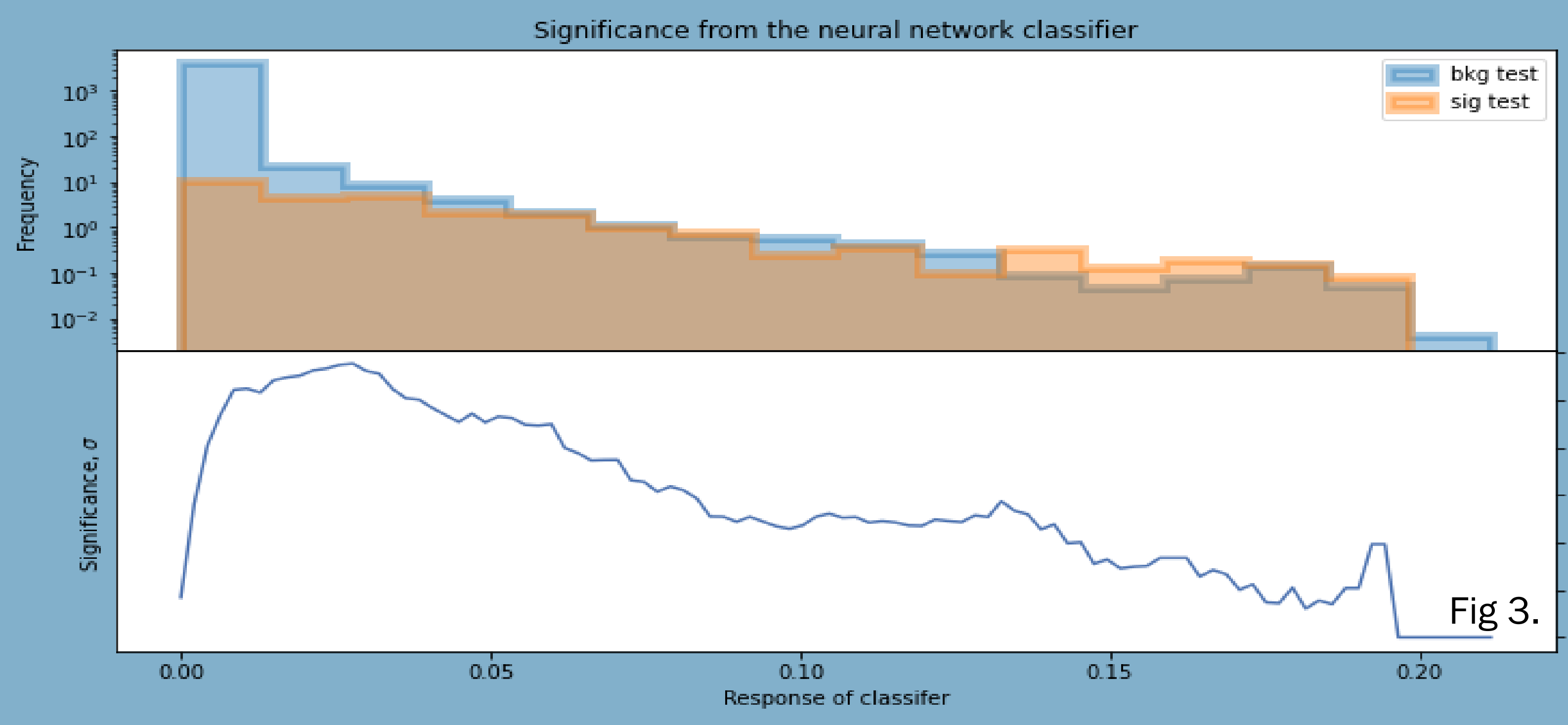


Fig 3.

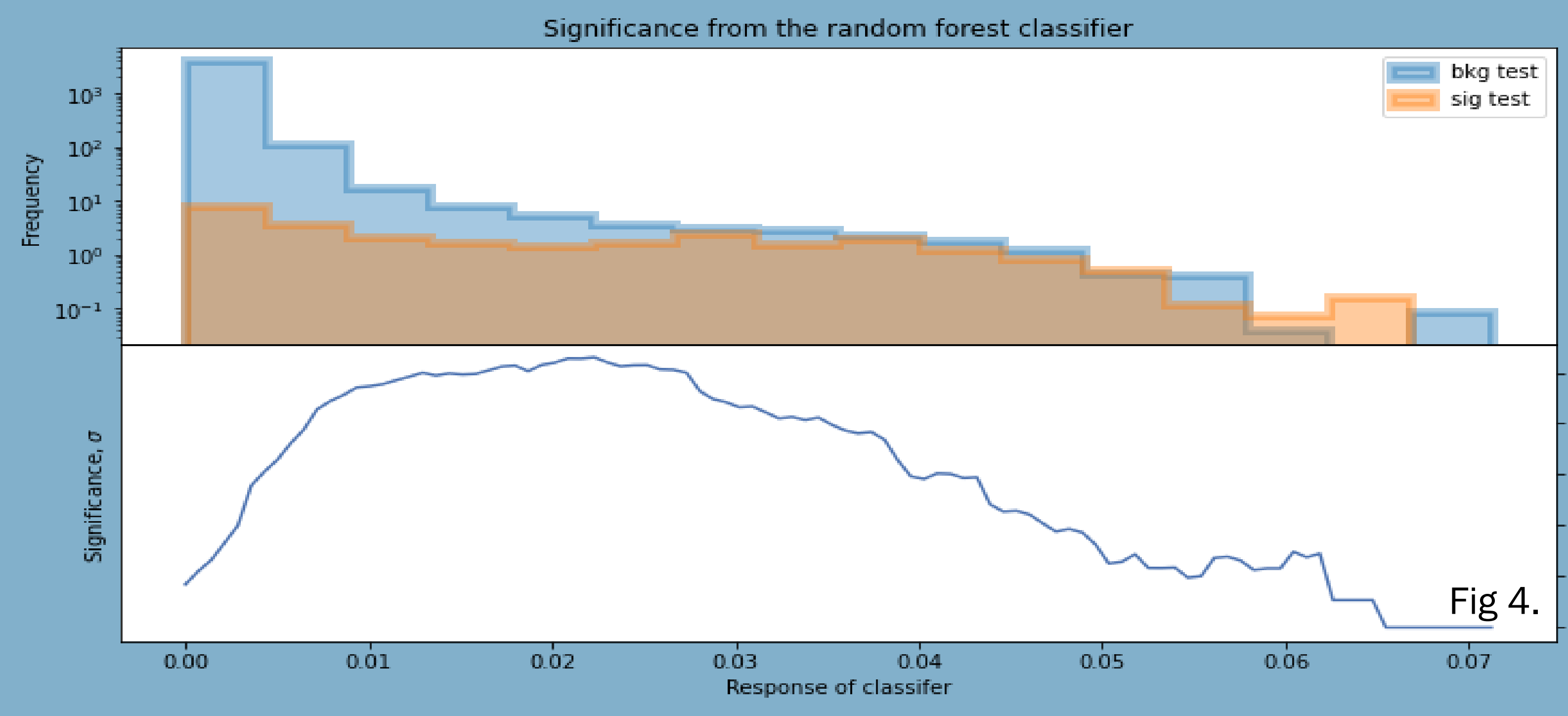
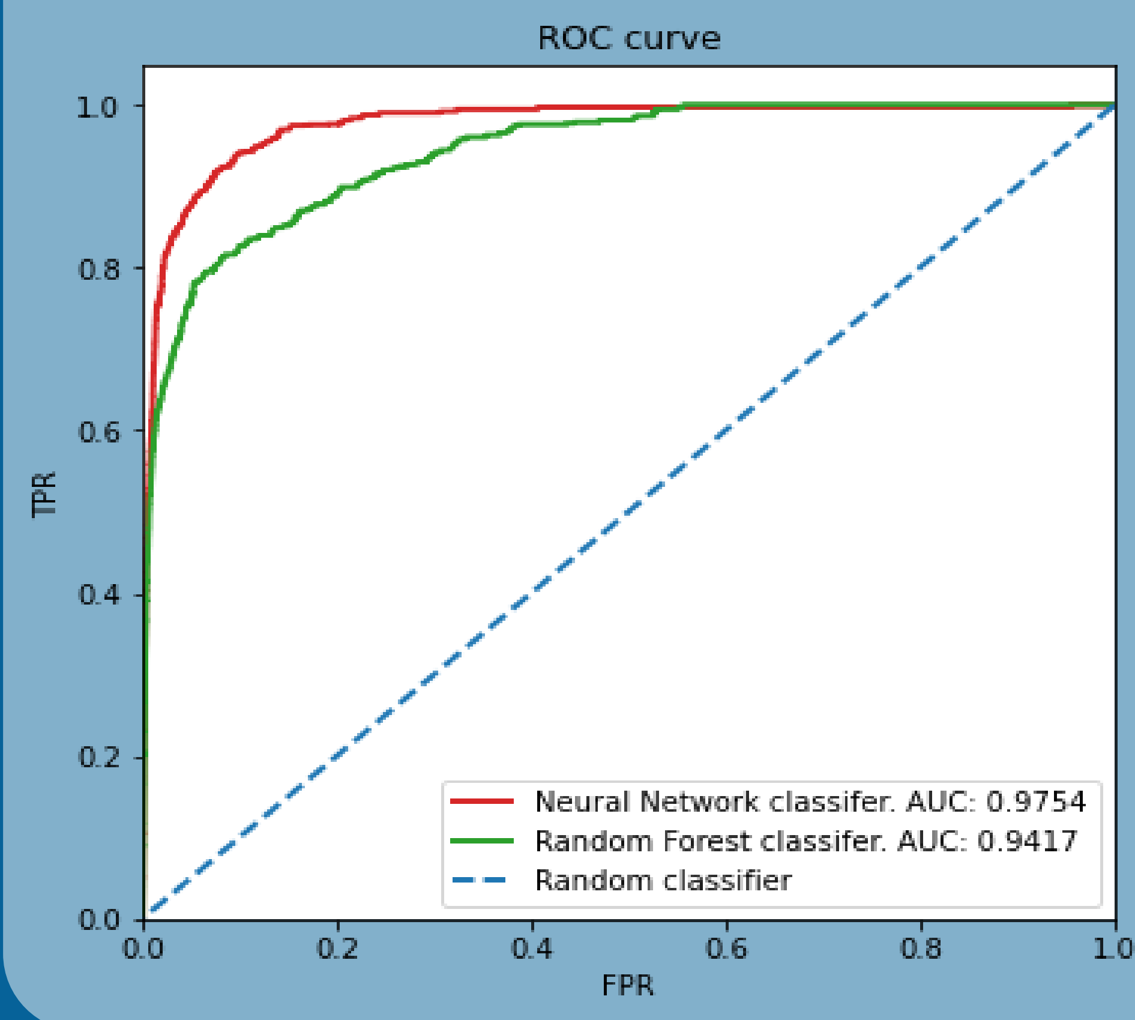
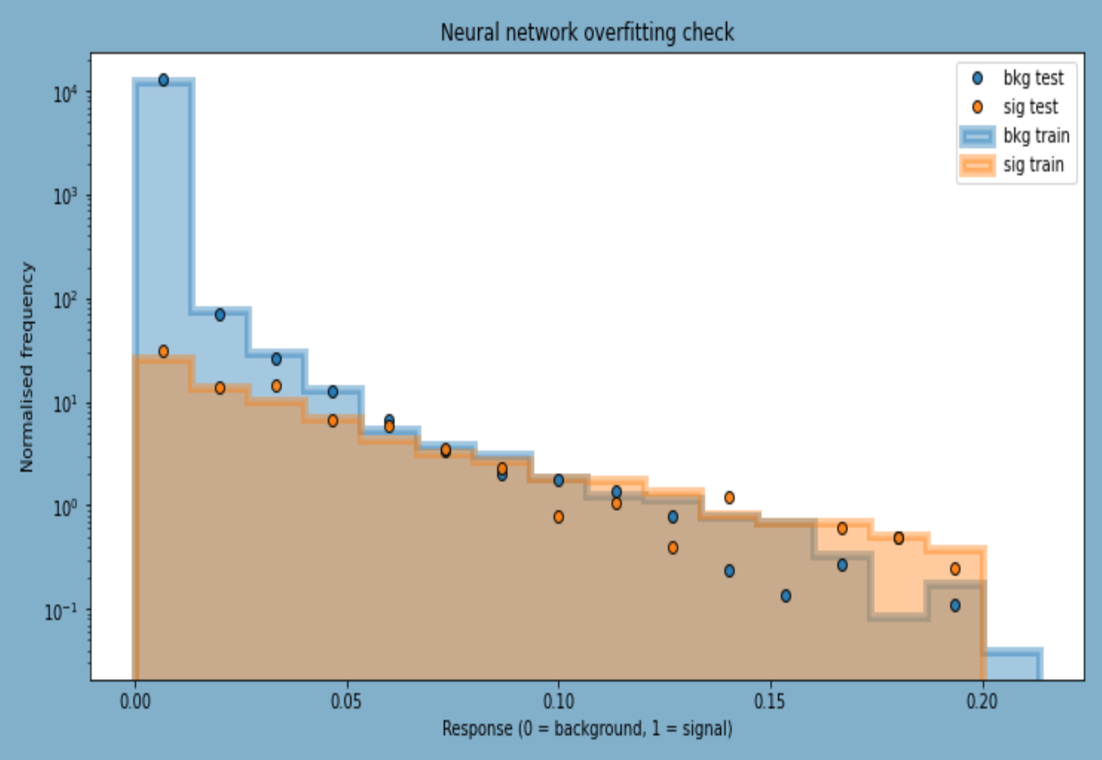


Fig 4.

Model Analysis:

The machine learning models are analysed by plotting model signal and background responses, generating a ROC curve and checking overfitting including a Kolmogorov–Smirnov test (metric measuring match or mis-match between two distributions).





ROC: Receiver operating characteristic

AUC : Area under curve (of ROC)

Conclusions:

1. Machine learning can be taught in an intuitive way using real scientific applications.
2. Neural networks were found to be the most effective in signal classification.
3. Simplistic CPU bound machine learning models can show the principles of AI driven data analysis and statical significance.
4. Using neural networks and random forests dark matter signal significances of 2.37σ and 2.26σ were found respectively, with limited overfitting and acceptable ROC curves with AUC of 0.9754 and 0.9417.

Oscar Jackson - Physics with Space Science (MPhys) – 4th year
okcj1g19@soton.ac.uk

Workbook (Available https://www.sussex.ac.uk/epp/outreach) Credits:

Project Lead: Kate Shaw
Project Supervisors: Meirin Oan Evans, Zöe Earnshaw, Thomas Stevenson
Developers: Oscar Jackson, Christopher Comiskey-Erazo, Iago Rosetto