

1 **Using Machine Learning Techniques for Data Quality**
2 **Monitoring at CMS Experiment**

3 by

4 Guillermo A. Fidalgo Rodríguez

5 A thesis presented for the degree of

6 BACHELLOR'S OF SCIENCE??

7 in

8 Physics

9 UNIVERSITY OF PUERTO RICO
10 MAYAGÜEZ CAMPUS

11 2018

12 Approved by:

13 _____
14 Sudhir Malik, Ph.D.

15 President, Graduate Committee
Date

16 _____
17 Héctor Méndez, Ph.D.

18 Member, Graduate Committee
Date

19 _____
20 Samuel Santana Colón, Ph.D.
21 Member, Graduate Committee

Date

22 _____
23 Rafael A. Ramos, Ph.D.
24 Chairperson of the Department

Date

²⁵ Abstract

²⁶ The Data Quality Monitoring (DQM) of CMS is a key asset to deliver high-quality
²⁷ data for physics analysis and it is used both in the online and offline environment. The cur-
²⁸ rent paradigm of the quality assessment is labor intensive and it is based on the scrutiny of
²⁹ a large number of histograms by detector experts comparing them with a reference. This
³⁰ project aims at applying recent progress in Machine Learning techniques to the automa-
³¹ tion of the DQM scrutiny. In particular the use of convolutional neural networks to spot
³² problems in the acquired data is presented with particular attention to semi-supervised
³³ models (e.g. autoencoders) to define a classification strategy that doesn't assume previ-
³⁴ous knowledge of failure modes. Real data from the hadron calorimeter of CMS are used
³⁵ to demonstrate the effectiveness of the proposed approach.

³⁶ *Keywords:* [DQM, online, offline, Machine Learning]

³⁷ **Acknowledgments**

³⁸ I wish to thank United States State Department and University of Michigan for pro-
³⁹ viding the opportunity to work abroad at CERN during the 2018 Winter Semester. I also
⁴⁰ wish to thank CERN staff, CMS Experiment , Texas Tech University, and the University
⁴¹ of Puerto Rico at Mayagüez, with special thanks to Dr. Federico de Guio for his local
⁴² mentorship and Dr. Nural Akchurin, and Dr. Sudhir Malik for their guidance. A very
⁴³ special thanks to Dr. Jean Krisch for accepting me for this great research opportunity and
⁴⁴ Dr. Steven Goldfarb for being a wonderful host and overall local guidance at CERN.

⁴⁵ List of Figures

46	2.1 CMS Detector	4
47	2.2 The trajectory of a particle traveling through the layers of the detector leaving behind it's signature footprint	5
49	4.1 Occupancy maps with 5x5 affected regions	11
50	4.2 Weights and Biases	12
51	4.3 Gradient Descent algorithm	13
52	4.4 Loss Function surface	14
53	5.1 Occupancy Maps with 1x1 bad regions. A) Good image B) Dead image C) Hot image	15
55	5.2 Two Convolutional Layers Model	17
56	5.3 Confusion Matrix results and Learning curve for 5×5 damaged area with on the same location for all trials	17
58	5.4 Confusion Matrix results and Learning curve for 5×5 damaged area with on the random location for all trials	18
60	5.5 Confusion Matrix results and Learning curve for 5×5 damaged area with an extra class to identify with random location for all trials	18
62	5.6 Jet multiplicity and the associated S_γ scale factor in the loose photon con- trol region before any corrections are applied.	19
64	5.7 N_{jet} distribution in the tight $\mu\mu$ control region after S_γ corrections.	20
66	5.8 p_T^γ (left) and $p_T^{\gamma(miss)}$ (right) distributions after applying the $S_\gamma(N_j)$ scale factor. Comparing to ??, an improvement in the agreement between data/MC can be observed.	20
68	5.10 H_T and m_{T2} distributions applying the $S_\gamma(N_j)$ scale factor.	21
69	5.9 N_b distribution before (left) and after (right) applying the $S_\gamma(N_j)$ scale factor.	21
71	5.11 Shown are data/MC comparisons for the p_T^{miss} (left) and H_T (right) distri- butions after applying both the N_j -dependent shape corrections (S_γ) and the global normalization scale factor (R_{norm}).	22
74	5.12 Shown are data/MC comparisons for the N_j (left) and N_b (right) distribu- tions after applying both the N_j -dependent shape corrections (S_γ) and the global normalization scale factor (R_{norm}).	23
77	5.13 Systematic uncertainty in the final prediction, as a function of the search bin, associated to the MC statistics.	24
79	5.14 $Z \rightarrow \nu\bar{\nu}$ background prediction for all search bins, including the break- down of the various uncertainties.	25

81 **Contents**

82	Abstract	i
83	Acknowledgments	ii
84	List of Figures	iii
85	1 Introduction	1
86	2 The CMS Experiment	3
87	3 Data Collection and Data Quality Monitoring	6
88	3.1 What is Data Collection for CMS?	6
89	3.2 What is Data Quality Monitoring?	7
90	4 What is Machine Learning?	9
91	4.1 Developing the Algorithm	10
92	4.2 Teaching the Algorithm	12
93	5 Results	15
94	5.1 SL Models for known anomalies in the HCAL data for DQM	16
95	5.1.1 Two Convolutional Layers for binary classification	16
96	5.1.2 Normalization Correction Using the tight $Z \rightarrow \mu^+ \mu^-$ Control Sample	21
97	5.2 Results	23
98	5.2.1 Systematics	23
99	5.2.2 $Z \rightarrow \nu \bar{\nu}$ Estimation for the Search Bins	25
101	6 References	26

¹⁰² **Chapter 1**

¹⁰³ **Introduction**

¹⁰⁴ The work for this thesis was performed at CERN on CMS Experiment. CERN stands
¹⁰⁵ for European Organization for Nuclear Research. It was founded in 1954 and is located
¹⁰⁶ at the Franco-Swiss border near Geneva. At CERN, physicists and engineers are probing
¹⁰⁷ the fundamental structure of the universe. They use the world's largest and most complex
¹⁰⁸ scientific instruments to study the basic constituents of matter – the fundamental parti-
¹⁰⁹ cles. The instruments used at CERN are purpose-built particle accelerators and detectors.
¹¹⁰ Accelerators boost beams of particles to high energies before the beams are made to col-
¹¹¹ lide with each other or with stationary targets. Detectors observe and record the results
¹¹² of these collisions. The accelerator at CERN is called the Large Hadron Collider (LHC),
¹¹³ the largest machine ever built by humans and it collides particles (protons) at close to the
¹¹⁴ speed of light. The process gives the physicists clues about how the particles interact, and
¹¹⁵ provides insights into the fundamental laws of nature. Seven experiments at the LHC use
¹¹⁶ detectors to analyze particles produced by proton-proton collisions. The biggest of these
¹¹⁷ experiments, ATLAS and CMS, use general-purpose detectors designed to study the fun-
¹¹⁸ damental nature of matter and fundamental forces and to look for new physics or evidence
¹¹⁹ of particles that are beyond the Standard Model. Having two independently designed de-
¹²⁰ tectors is vital for cross-confirmation of any new discoveries made. The other two major
¹²¹ detectors ALICE and LHCb, respectively, study a state of matter that was present just
¹²² moments after the Big Bang and preponderance of matter than antimatter. Each experi-

123 ment does important research that is key to understanding the universe that surrounds and
124 makes us.

125

126 [Chapter 2](#) presents a basic description of the Large Hadron Collider and CMS Detector

127

128 ?? gives a brief motivation

129

130 ?? is dedicated to a study optimizing

131

132 ?? ptimated.

133

134 ?? details an improvarger production cross-section than Z+jets process used before.

135

136 The conclusions and results of each chapter are presented in the corresponding chap-
137 ter.

138

139 This thesis work has been presented at several internal meetings of the CMS Experi-
140 ment and at the following international meetings and conferences:

141 1. **Andrés Abreu** gave a talk “*Estimation of the Z Invisible Background for Searches*

142 *for Supersymmetry in the All-Hadronic Channel*” at “APS April 2018: American
143 Physical Society April Meeting 2018, 14-17 Apr 2018”, Columbus, OH

144 2. **Andrés Abreu** gave a talk “*Phase-2 Pixel upgrade simulations*” at the “USLUA

145 Annual meeting: 2017 US LHC Users Association Meeting, 1-3 Nov 2017”, Fer-
146 milab, Batavia, IL

¹⁴⁷ Chapter 2

¹⁴⁸ The CMS Experiment

¹⁴⁹ The Compact Muon Solenoid (CMS) detector is a general purpose particle detector
¹⁵⁰ designed to investigate various physical phenomena concerning the SM and beyond it,
¹⁵¹ such as Supersymmetry, Extra Dimensions and Dark Matter. As its name implies, the
¹⁵² detector is a solenoid which is constructed around a superconducting magnet capable of
¹⁵³ producing a magnetic field of 3.8 T. The magnetic coil is 13m long with an inner diameter
¹⁵⁴ of 6m, making it the largest superconducting magnet ever constructed. The CMS detector
¹⁵⁵ itself is 21m long with a diameter of 15m and it has a weight of approximately 14,000
¹⁵⁶ tons. The CMS experiment is one of the largest scientific collaborations in the history
¹⁵⁷ of mankind with over 4,000 participants from 42 countries and 182 institutions. CMS is
¹⁵⁸ located at one of these points and it essentially acts as a giant super highspeed camera
¹⁵⁹ that makes 3D images of the collisions that are produced at a rate of 40 MHz (40 million
¹⁶⁰ times per second). The detector has an onion-like structure to capture all the particles that
¹⁶¹ are produced in these high energy collisions most of them being unstable and decaying
¹⁶² further to stable particles that are detected. CMS detector was designed with the following
¹⁶³ features (as shown in [Figure 2.1](#)) :

- ¹⁶⁴ 1. A **magnet** with large bending power and high performance muon detector for good
¹⁶⁵ muon identification and momentum resolution over a wide range of momenta and
¹⁶⁶ angles.

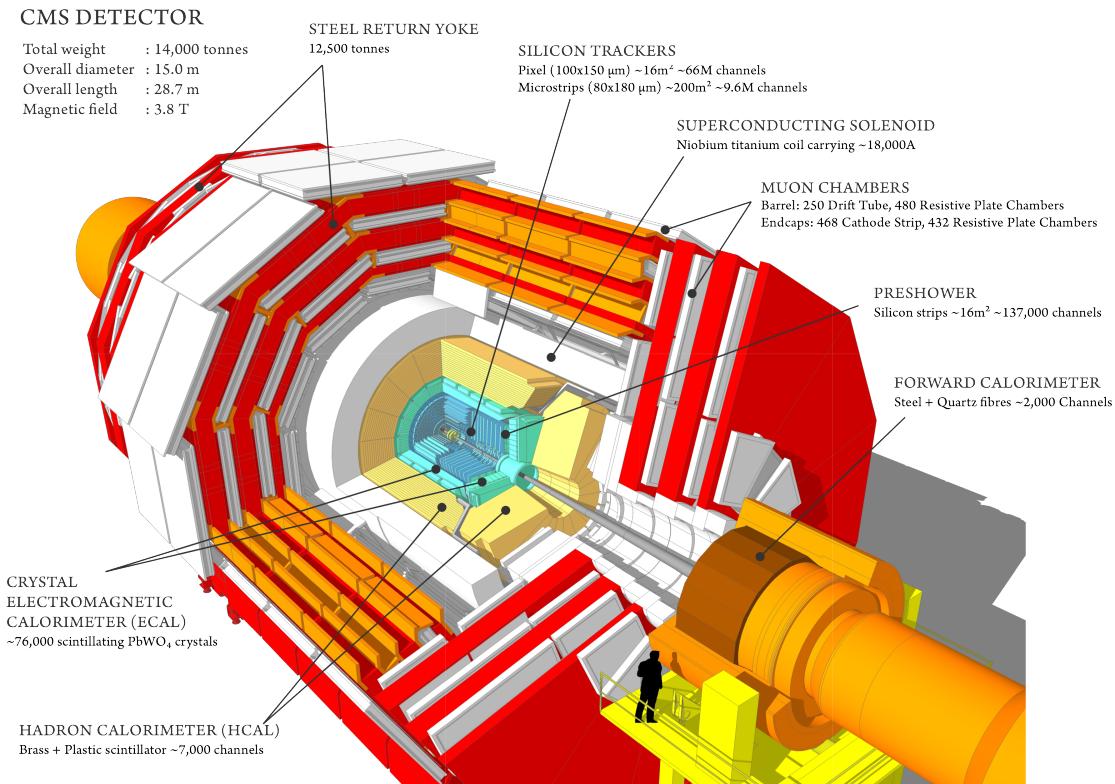


Figure 2.1: CMS Detector

- 167 2. An **inner tracking system** capable of high reconstruction efficiency and momen-
- 168 tum resolution requiring **pixel detectors** close to the interaction region.
- 169 3. An **electromagnetic calorimeter** able to provide good electromagnetic energy res-
- 170 olution and a high isolation efficiency for photons and leptons.
- 171 4. A **hadron calorimeter** capable of providing precise missing-transverse-energy and
- 172 dijet-mass resolution.

173 A property from these particles that is exploited is their charge. Normally, particles
 174 produced in collisions travel in a straight line, but in the presence of a magnetic field,
 175 their paths are skewed and curved. Except the muon system, the rest of the subdetectors
 176 lie inside a 3.8 Tesla magnetic field . Due to the magnetic field the trajectory of charged
 177 particle produced in the collisions gets curved (as shown in [Figure 2.2](#)) and one can
 178 calculate the particle's momentum and know the type of charge on the particle. The
 179 Tracking devices are responsible for drawing the trajectory of the particles by using a
 180 computer program that reconstructs the path by using electrical signals that are left by

the particle as they move. The Calorimeters measure the energy of particles that pass through them by absorbing their energy with the intent of stopping them. The particle identification detectors work by detecting radiation emitted by charged particles and using this information they can measure the speed, momentum, and mass of a particle. After the information is put together to make the “snapshot” of the collision one looks for results that do not fit the current theories or models in order to look for new physics.

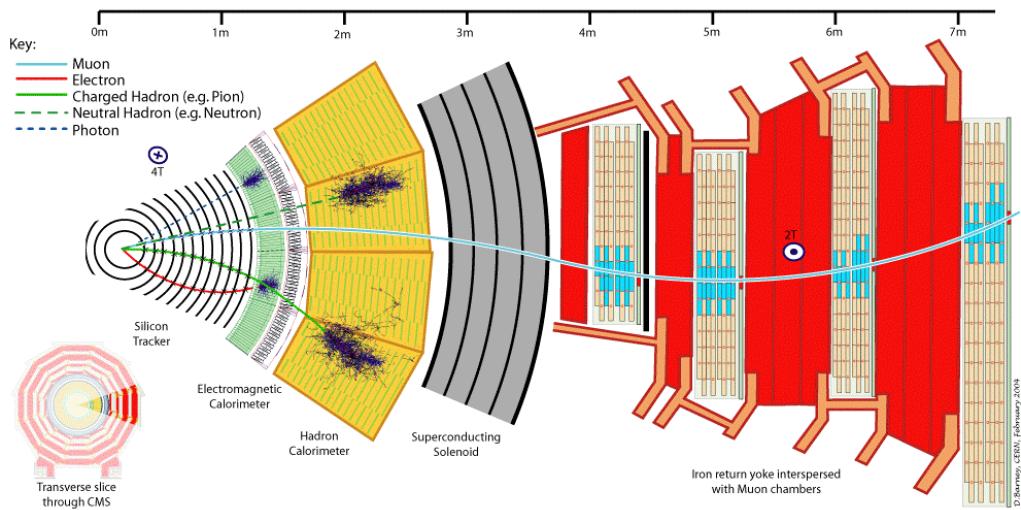


Figure 2.2: The trajectory of a particle traveling through the layers of the detector leaving behind its signature footprint

The project focusses specifically on data collected from one of the Calorimeters, - the Hadron Calorimeter (HCAL). The HCAL, as its name indicates, is designed to detect and measure the energy of hadrons or, particles that are composed of quarks and gluons, like protons and neutrons. Additionally, it provides an indirect measurement of the presence of non-interacting, uncharged particles such as neutrinos (missing energy) . Measuring these particles is important as they can tell us if new particles such as the Higgs boson or supersymmetric particles (much heavier versions of the standard particles we know) have been formed. The layers of the HCAL are structured in a staggered fashion to prevent any gaps that a particle might pass through undetected. There are two main parts: the barrel and the end caps. There are 36 barrel wedges that form the last layer of the detector inside the magnet coil, there is another layer outside this, and on the endcaps, there are another 36 wedges to detect particles that come out at shallow angles with respect to the beam line.

200 Chapter 3

201 Data Collection and Data Quality

202 Monitoring

203 3.1 What is Data Collection for CMS?

204 During data taking there are millions of collisions occurring in the center of the de-
205 tector every second. The data per event is around one million bytes (1 MB), that is produced
206 at a rate of about 600 million events per second [1], that's about 600 MB/s. Keeping
207 in mind that only certain events are considered "interesting" for analysis, the task of de-
208 ciding what events to consider out of all the data collected is a two-stage process. First,
209 the events are filtered down to 100 thousand events per second for digital reconstruction
210 and then more specialized algorithms filter the data even more to around 100 200 events
211 per second that are found interesting. For CMS there is a Data Acquisition System that
212 records the raw data to what's called a High-Level Trigger farm which is a room full
213 of servers that are dedicated to processing and classify this raw data quickly. The data
214 then gets sent to what's known as the Tier-0 farm where the full processing and the first
215 reconstruction of the data are done. [2]

216 3.2 What is Data Quality Monitoring?

217 To operate a sophisticated and complex apparatus as CMS, a quick online feedback on
218 the quality of the data recorded is needed to avoid taking low quality data and to guarantee
219 a good baseline for the offline analysis. Collecting a good data sets from the collisions
220 is an important step towards search for new physics as deluge of new data poses an extra
221 challenge of processing and storage. This all makes it all the more important to design
222 algorithms and special software to control the quality of the data. This is where the Data
223 Quality Monitoring (DQM) plays a critical in the maintainability of the experiment, the
224 operation efficiency and performs a reliable data certification. The high-level goal of
225 the system is to discover and pinpoint errors, problems occurring in detector hardware
226 or reconstruction software, early, with sufficient accuracy and clarity to maintain good
227 detector and operation efficiency. The DQM workflow consists of 2 types: **Online** and
228 **Offline**.

229 The **Online** DQM consists of receiving data taken from the event and trigger his-
230 tograms to produce results in the form of monitoring elements like histogram references
231 and quality reports. This live monitoring of each detector's status during data taking gives
232 the online crew the possibility to identify problems with extremely low latency, mini-
233 mizing the amount of data that would otherwise be unsuitable for physics analysis. The
234 scrutiny of the Online DQM is a 24/7 job that consists of people or shifters that work at the
235 CMS control center constantly monitoring the hundreds of different plots and histograms
236 produced by the DQM software. This consumes a lot of manpower and is strenuous work.

237 The **Offline** DQM is more focused on the full statistics over the entire run of the
238 experiment and works more on the data certification. In the offline environment, the
239 system is used to review the results of the final data reconstruction on a run-by-run basis,
240 serving as the basis for certified data used across the CMS collaboration in all physics
241 analyses. In addition, the DQM framework is an integral part of the prompt calibration
242 loop. This is a specialized workflow run before the data are reconstructed to compute and
243 validate the most up-to-date set of conditions and calibrations subsequently used during

²⁴⁴ the prompt reconstruction.

²⁴⁵ This project aims to minimize the DQM scrutiny by eye and automate the process so
²⁴⁶ that there is a more efficient process to monitor the detector and the quality of the data by
²⁴⁷ implementing Machine Learning techniques.

²⁴⁸ **Chapter 4**

²⁴⁹ **What is Machine Learning?**

Machine Learning (ML) can be defined as an application of Artificial Intelligence that permits the computer system to learn without being told explicitly. In ML a computer program is said to learn from experience E with respect to some class of tasks T and performance measure P, if its performance at tasks in T, as measured by P, improves with experience E [3]. ML has made tremendous strides in the past decades and has become very popular recently due to its multifaceted applications. It is being used on social media, marketing, and in the scientific community as well. Some examples of ML applications are: the algorithms used on application in smartphones to detect human faces, self-driving cars, computer games, stock prediction, and voice recognition. An interesting characteristic of ML algorithms is that the more data one inputs the better is the performance. The ML application has a very wide spectrum covering almost every aspect of human endeavor that involves a lot of data. Scientific analysis today generates enormous data and is hence a perfect use case to apply ML techniques. In this work we use enhanced ML techniques based on progress in the recent past.

In general, there are two main categories to classify machine learning problems: **Supervised Learning** (SL) and **Unsupervised Learning** (UL). SL is the most used ML approach and has proven to be very effective for a wide variety of problems. Examples of common SL problems are: spam filters, predicting housing prices, identifying a malignant or benign tumor, etc. These types of problems are characterized by providing a

“right answer” as a reference. For example, spam filter algorithms identify emails that are spams by training on a dataset that has examples of such emails. In case of predicting house prices, the algorithm is trained on a dataset of houses involving features like the area of the house, number of rooms, and the selling price of the house.

UL algorithms are different in the sense that they do not have the “right answers” given to the machine. Instead, UL algorithms are used for finding patterns and make clusters from the given data. That is what also forms the basis of a search engine (e.g. Google news). Clicking on a link to a news article, one gets many different stories of different journals that have some correlation with the article searched. This happens because the ML algorithm is capable of learning features and shared patterns from a bunch of data without being given any specifics. Another interesting UL problem is the so-called “cocktail party” that involves distinguishing the voice of two people recording on two microphones located at different places. The ML algorithm is able to separate the sources of the voices in the recordings by learning the voice features that correspond to each person, showing the power of the UL algorithm.

In this study, I have focused on an SL approach and a variant of the UL approach, called the **Semi-Supervised Learning** approach (SSL). The SSL is named so because the data involves looking at images that are already known to be “Good” but one doesn’t necessarily know every possible situation that produces a “Bad” image. The purpose is to define a metric for a “good” image and subsequently decide if an image is “bad” in case it deviates too much from an acceptable value.

4.1 Developing the Algorithm

To develop an ML algorithm the following are taken into consideration, what is the task? and what is the method to approach the task? In our case, we are looking into images that have information about the activity that the channels in the HCAL are detecting. These images are called ”occupancy maps” and they are a visual way of monitoring the health of the detector itself (see [Figure 4.1](#)). There are two common problems that can be

identified by viewing occupancy maps which are called "dead channels" and "hot towers". They are referred to as "**dead**" and "**hot**" respectively in the rest of this document. Dead channels mean that on a certain place in the occupancy map there is not any readout from the channels on the HCAL and hot channels mean that there are channels that are being triggered by noise or are damaged in a way that makes them readout too much activity.

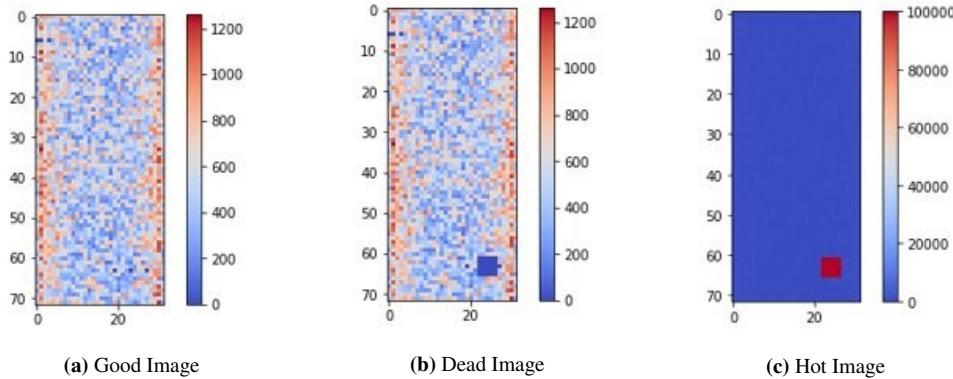


Figure 4.1: Occupancy maps with 5x5 affected regions

The problem is the following, to create a model that can detect and classify what type of scenario is occurring on each occupancy map. For this, we want to go with a SL approach which means that we will give the model the images as the input and it will train on these images by learning to identify patterns or features in the image and try to do a "fit" from the images to their corresponding labels. After the training, the algorithm will be given a testing set for us to evaluate the model's ability to correctly detect if there is a problem with the image and what type of problem is being detected. The output of the model will be the predicted class of the test image. The predictions are based on the labels and their corresponding images that were given to the model during training. This means that if the model was trained with 3 different types of images with their corresponding label the model will only work well for images that present similar patterns or characteristics to those presented in the training. For example, if we only train the model to distinguish between "good" and "hot" then when the model encounters images that aren't either of these two, like an image labeled "dead", then the model will not know what to do with this image and will give it an incorrect label. After the SL model has been tested the next step is trying an SSL model. The term semi-supervised

simply means that there isn't a ground truth label that is being given to the model during training because either there isn't necessarily a ground truth, or we don't know what the ground truth is. What we do know, is what is considered as a "good" image and what this approach hopes to accomplish is to use the error in the reconstruction of the input image and use that information to discriminate between the "good" vs the "bad" images.

4.2 Teaching the Algorithm

The way an ML algorithm learns is by an iterative process called an optimization algorithm in which the predicted output value of the model is compared to the desired output (See [Figure 4.2](#)) and the weights and biases of the model are adjusted such that the predicted output is closer to the desired output.

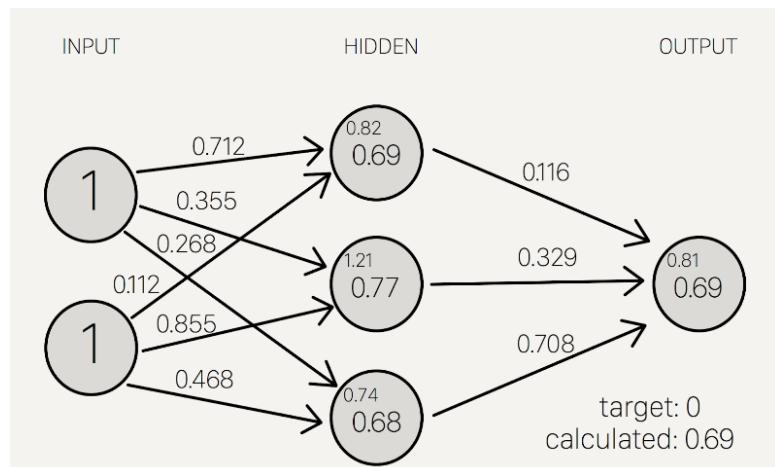


Figure 4.2: Weights and Biases

"Optimization algorithms helps us to **minimize** (or **maximize**) an **Objective function** (*another name for Error function*) $E(x)$ which is simply a mathematical function dependent on the Model's internal **learnable parameters** which are used in computing the target values(Y) from the set of *predictors*(X) used in the model. For example - we call the **Weights(W)** and the **Bias(b)** values of the neural network as its internal learnable *parameters* which are used in computing the output values and are learned and updated in the direction of optimal solution i.e. minimizing the **Loss** by the network's training process and also play a major role in the **training** process of the Neural Network Model." [4].

Gradient Descent

The “Learning” in Machine Learning.

Update the values of X (punish) it when it is wrong.

$$X = X - \eta \nabla(X)$$

X: weights or biases

η : Learning Rate (typically 0.01 to 0.001)

η :The rate at which our network learns. This can change over time with methods such as Adam, Adagrad etc.  (hyperparameter)

$\nabla(X)$: Gradient of X

We seek to update the weights and biases by a value indicating how “off” they were from their target.

Gradients naturally have increasing slope, so we put a negative in front of it to go downwards

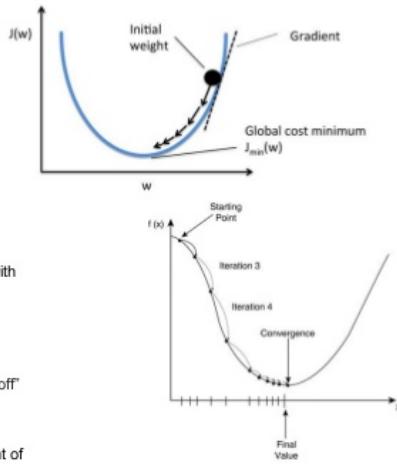


Figure 4.3: Gradient Descent algorithm

- 335 The most basic and probably the most used optimizer is called Gradient Descent (GD).
- 336 GD is based on the concept of using the gradient of a loss or cost function and moving
- 337 the weights and biases of the ML model so that the predicted value is taking a step in the
- 338 decreasing direction of this error function (See [Figure 4.3](#)). In general, the “terrain” of the
- 339 loss function is not a smooth bowl-shaped surface like the one present in the image. The
- 340 most general form of the surface is more similar to a rocky mountain (See [Figure 4.4](#)),
- 341 which presents a problem when using simple optimizers like GD.

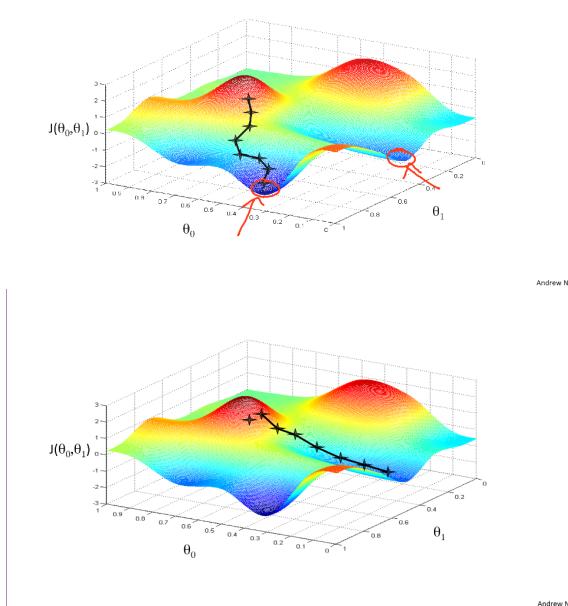


Figure 4.4: Loss Function surface

³⁴² Chapter 5

³⁴³ Results

³⁴⁴ Here first the limitations of Scikit-learn predefined ML models - Logistic Regres-
³⁴⁵ sion(LR) and Multi-Layer-Perceptron(MLP), are described. The Logistic Regression
³⁴⁶ Model seems to work almost perfectly with all 3 classes when the bad region size is
³⁴⁷ 5x5 (as in [Figure 4.1](#)) with either the same or randomized location. When the bad region
³⁴⁸ size is 1x1 like in [Figure 5.1](#) the LR Model performs poorly with an accuracy of approxi-
³⁴⁹ mately 20%. The MLP does not seem to work in any of the used cases that are studied as
³⁵⁰ it always performs poorly with an accuracy of $\approx 40\%$.

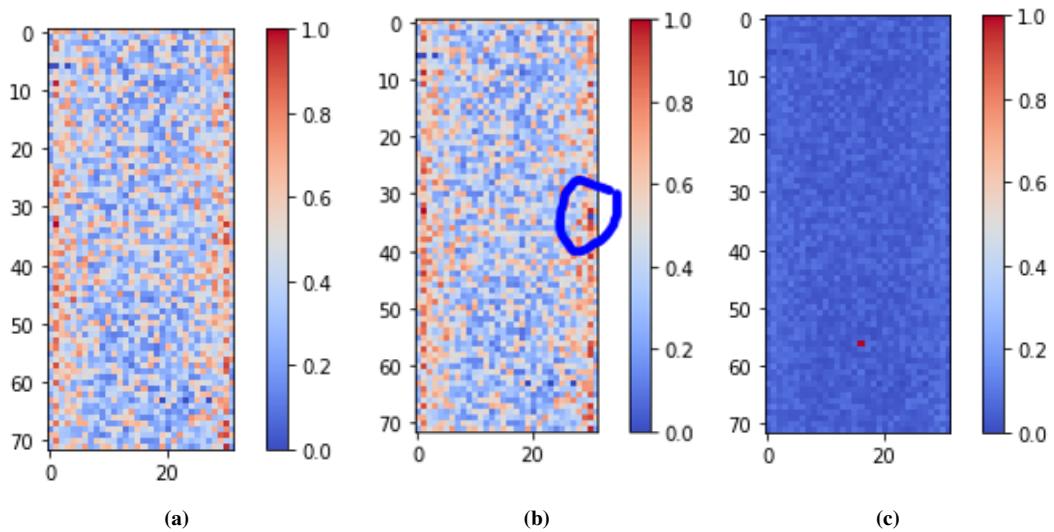


Figure 5.1: Occupancy Maps with 1x1 bad regions. A) Good image B) Dead image C) Hot image

³⁵¹ Also, the use of Scikit-learn's library is limited in comparison with the Keras module
³⁵² since one cannot customize the structure of the ML model with detail. Moreover, Keras is

353 an ML library designed for developing deep neural networks. Hence it was decided to use
354 Keras primarily for the creation of the model. With the Keras library, numerous models
355 were designed with both, SL method and SSL learning method. Using SL method, we are
356 interested in detecting anomalies and classifying what type of anomaly is seen. With SSL
357 method, we are interested in looking at the error of the reconstruction of an image to give
358 an idea that the image given can be considered good or that it might have some unseen
359 anomalies

360 **5.1 SL Models for known anomalies in the HCAL data
361 for DQM**

362 We considered three SL Models for classification of known anomalies in the HCAL
363 data for DQM. These models are based on Convolutional Neural Networks and differ
364 in the number of layers utilized, their ordering and number of units in each layer. The
365 Models and the corresponding results are described below.

366 **5.1.1 Two Convolutional Layers for binary classification**

367 Several variations of the two Convolutional Layers Model were tested and optimized
368 on the DQM data. This led to an optimal value of 8 units/neurons in the Convolutional
369 layers. The detail of selecting the number of units per layer is of great importance to find
370 a balance between efficiency and complexity of a model. More complex models (more
371 layers and connections) are “heavy” to train in terms of computational cost, provide better
372 results and are prone to “overfitting” to the training data. Simpler models (fewer layers
373 and connections) are quicker to train, efficient and computationally economic. However,
374 simpler models are more likely to “underfit” to the data. The [Figure 5.2](#) below shows a
375 code snippet with this model. [Figure 5.3](#) below shows the learning curve for this model
376 trained with Good and Hot images for fixed 5x5 location and the corresponding Confusion
377 Matrix.

```

model = Sequential([
    BatchNormalization(input_shape=input_shape),
    Conv2D(8, kernel_size=(3, 3), strides=(2, 2), activation='relu'),
    Conv2D(8, kernel_size=(3, 3), strides=(2, 2), activation='relu'),

    Dropout(0.25),

    Flatten(),

    Dense(2, activation='softmax')]

])

```

Figure 5.2: Two Convolutional Layers Model

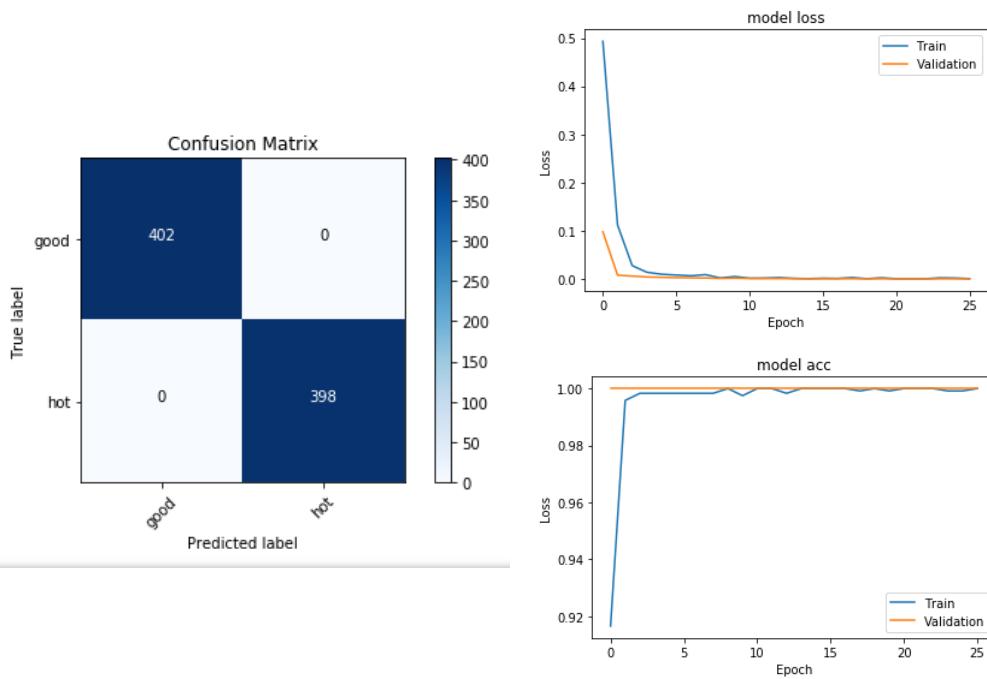


Figure 5.3: Confusion Matrix results and Learning curve for 5×5 damaged area with on the same location for all trials

378 [Figure 5.4](#) shows the learning curve for this model trained with Good and Hot images
 379 for fixed 5×5 location and the corresponding Confusion Matrix.

380 [Figure ??](#) shows the learning curve for this model trained with Good, Hot and Dead
 381 images for random 5×5 location and the corresponding Confusion Matrix
 382 on contribution typically arises from leptons (mostly electrons) whose tracks are not
 383 properly reconstructed, yet leave energy measurements in the ECAL.

384
$$f = \frac{fake}{prompt+fake} ,$$

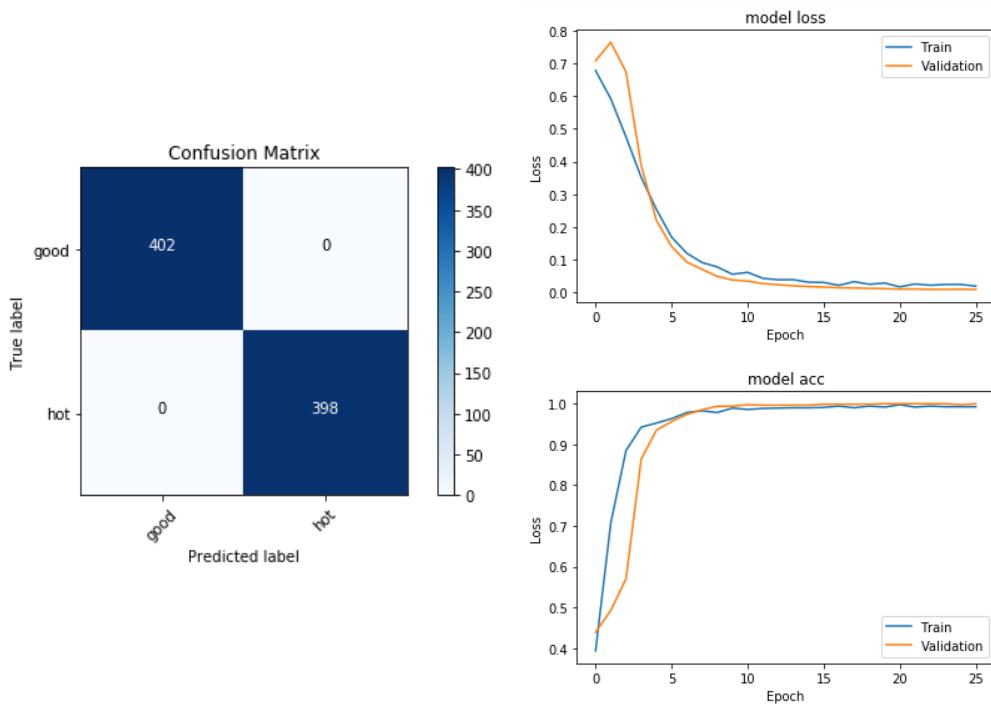


Figure 5.4: Confusion Matrix results and Learning curve for 5×5 damaged area with random location for all trials

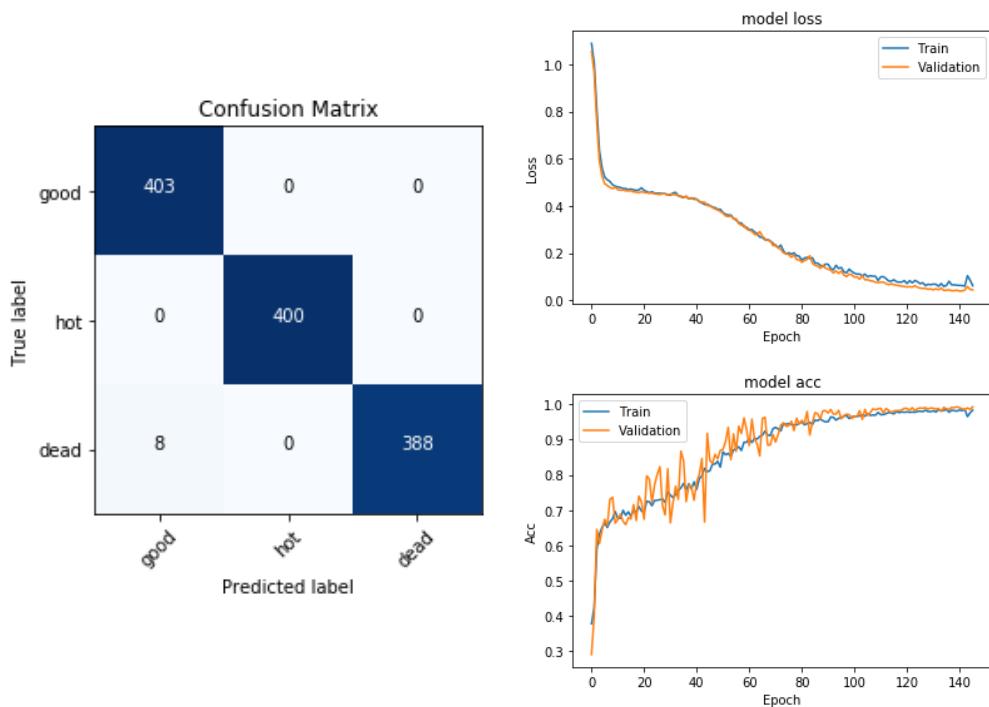


Figure 5.5: Confusion Matrix results and Learning curve for 5×5 damaged area with an extra class to identify with random location for all trials

385 ?? shows the purity and fakerate for photons that pass the loose ID/selection, have a
 386 $p_T > 200$ GeV and are within the ECAL acceptance range. A sample is obtained in which
 387 77% of the photons are direct, 12% are fragmentation and 11% are fakes. This implies an
 388 average purity of $\sim 89\%$ for this sample, well within the value that is expected. ?? shows
 389 the same ratios for the loose $\gamma +$ jets control region described in ?? . Although the amount
 390 of statistics has decreased due to the additional cuts, a similar trend can be observed.

$$391 S_{\gamma}^i = \frac{\text{Data}^i - \text{MC}_{\text{other}}^i}{\text{MC}_{\gamma+\text{jets}}^i},$$

392 where i denotes any given bin in the N_j distribution. The shape correction factors S_{γ}^i
 393 are displayed graphically in Figure 5.6 (right) for each N_j bin. These factors correct for
 394 differences in the jet multiplicity shape, while the overall normalization is estimated from
 395 the tight $\mu\mu$ control region. Figure 5.7 shows the N_j distribution in the tight $\mu\mu$ control
 396 region after the calculated scale factors have been applied. The S_{γ} correction will be
 397 applied to the $Z \rightarrow \nu\bar{\nu}$ simulation final prediction for each of the analysis search bins. The
 398 uncertainty associated with the scale factor is estimated from the event yields in the loose
 399 photon control region. This uncertainty will form part of the total systematic uncertainty
 400 in the final prediction.

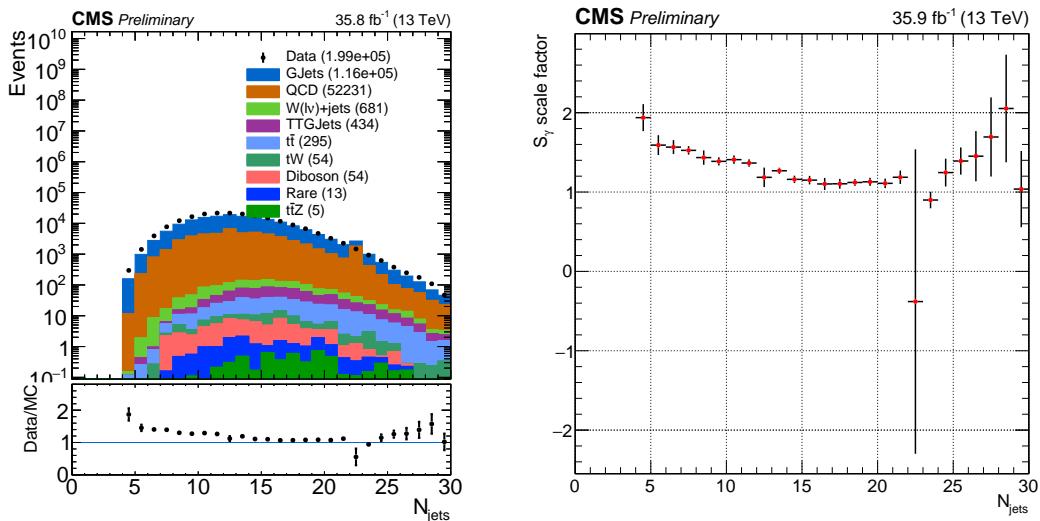


Figure 5.6: Jet multiplicity and the associated S_{γ} scale factor in the loose photon control region before any corrections are applied.

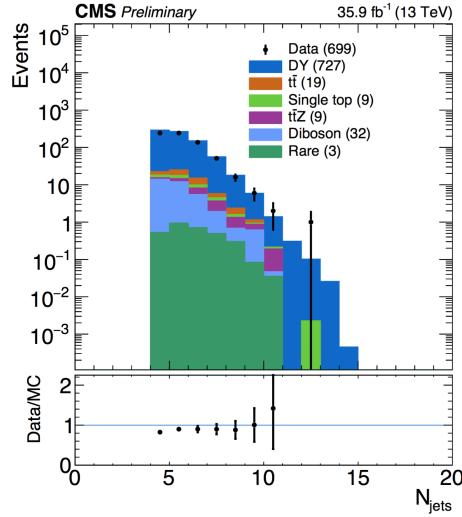


Figure 5.7: N_{jet} distribution in the tight $\mu\mu$ control region after S_γ corrections.

401 The effect of the $S_\gamma(N_j)$ scale factor is shown for various distributions. These results
 402 show that the overall agreement between data and simulation improves after applying the
 403 corresponding shape corrections.

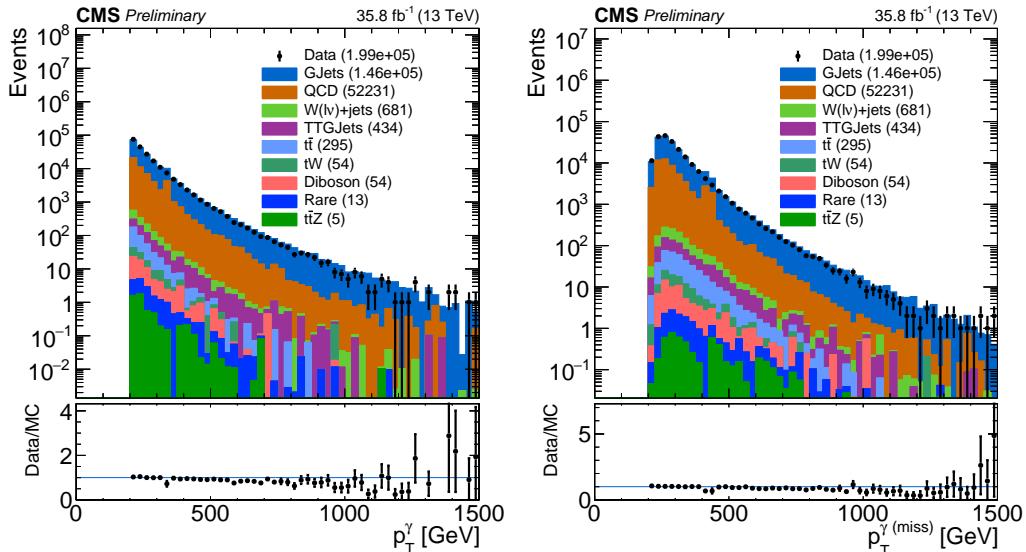


Figure 5.8: p_T^γ (left) and $p_T^{\gamma(\text{miss})}$ (right) distributions after applying the $S_\gamma(N_j)$ scale factor. Comparing to ??, an improvement in the agreement between data/MC can be observed.

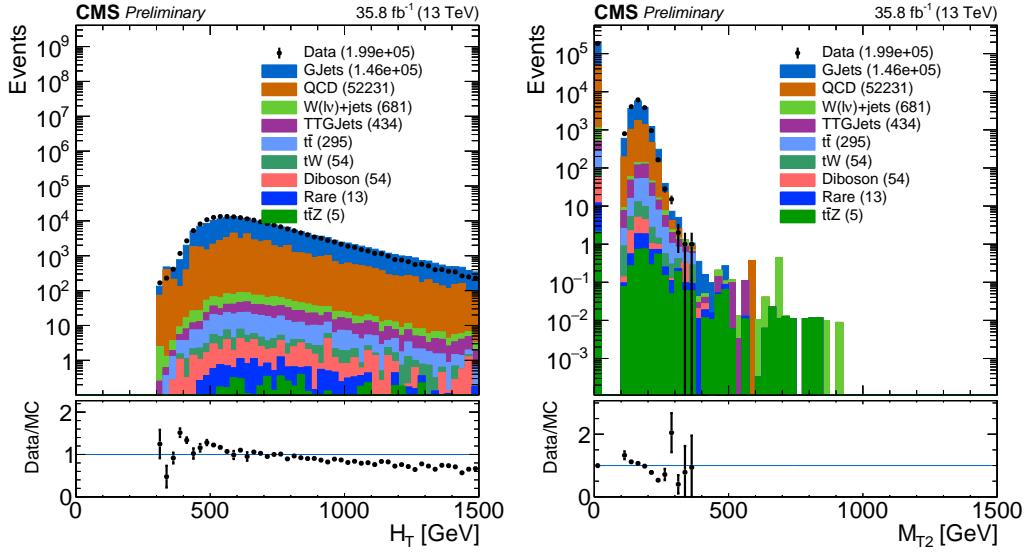


Figure 5.10: H_T and m_{T2} distributions applying the $S_\gamma(N_j)$ scale factor.

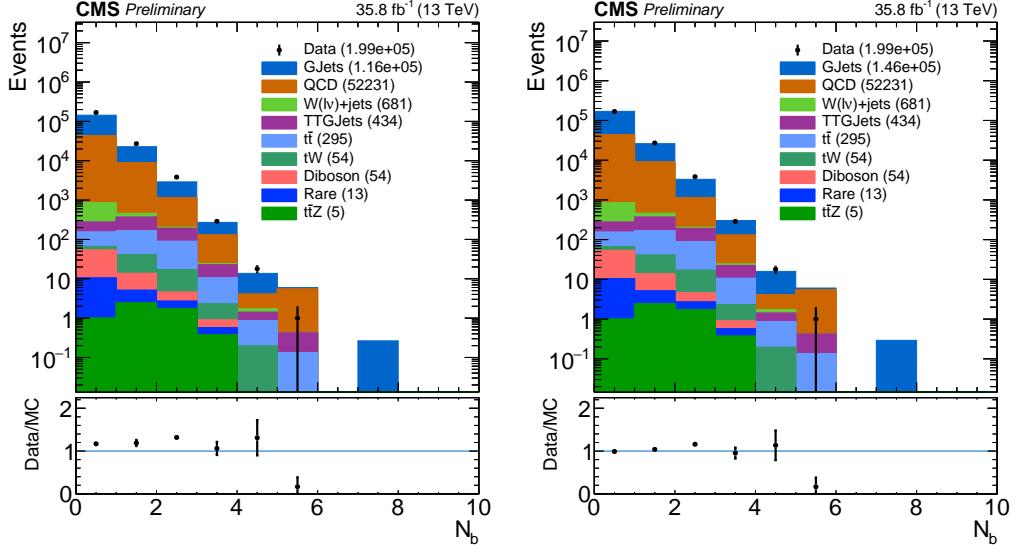


Figure 5.9: N_b distribution before (left) and after (right) applying the $S_\gamma(N_j)$ scale factor.

404 5.1.2 Normalization Correction Using the tight $Z \rightarrow \mu^+ \mu^-$ Control

405 Sample

406 In order to constrain the normalization of the $Z \rightarrow \nu \bar{\nu}$ simulation sample, a normal-
 407 ization correction factor R_{norm} is calculated from the tight $\mu\mu$ control region defined in
 408 subsection 5.1.2. Two categories are considered: the zero b-tagged jet category ($N_b = 0$),
 409 and the ≥ 1 b-tagged jet category ($N_b \geq 1$). Both of these categories are statistically
 410 consistent with each other but the inclusive region ($N_b \geq 0$) has a lower overall un-

certainty. The method used to calculate the normalization scale factor requires that the N_j -dependent shape correction factors already be applied. Then, the R_{norm} factor can be extracted from the ratio of the total event yield in data to that in the simulation. This factor is found to be:

$$R_{norm} = 1.070 \pm 0.085,$$

where the uncertainty includes only the associated statistical uncertainties on data and simulation. This uncertainty is found to be propagated to the final background prediction, see subsection 5.2.1.

419

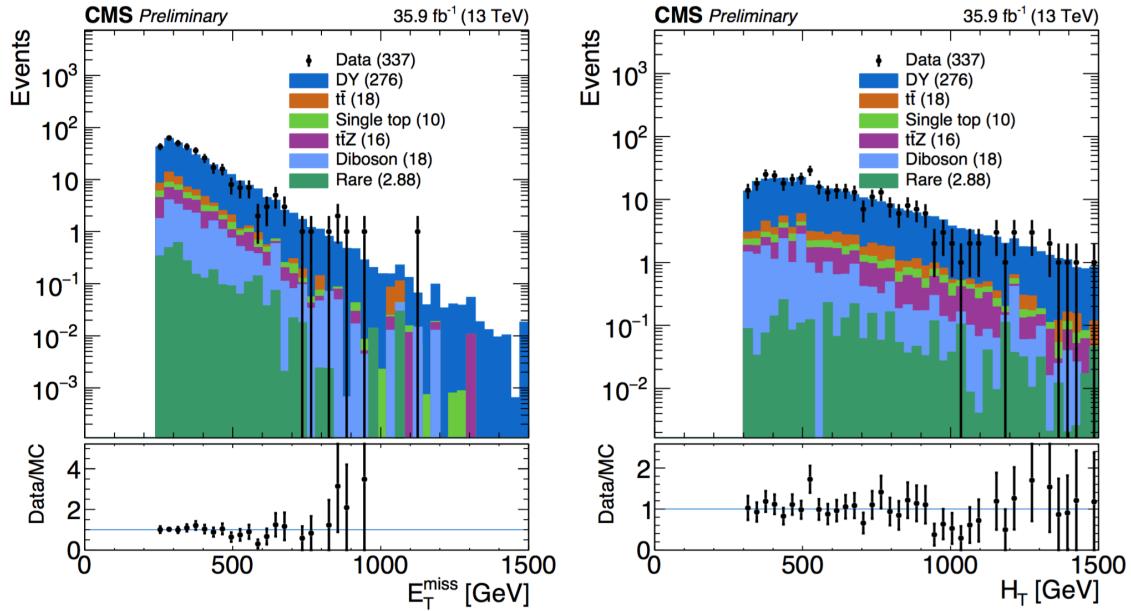


Figure 5.11: Shown are data/MC comparisons for the p_T^{miss} (left) and H_T (right) distributions after applying both the N_j -dependent shape corrections (S_γ) and the global normalization scale factor (R_{norm}).

Data/MC comparisons are shown in Figure 5.11 and Figure 5.12 after applying R_{norm} for several distributions in the study. With this final global scale factor all the required ingredients for the central value of the $Z \rightarrow \nu\bar{\nu}$ background prediction are obtained.

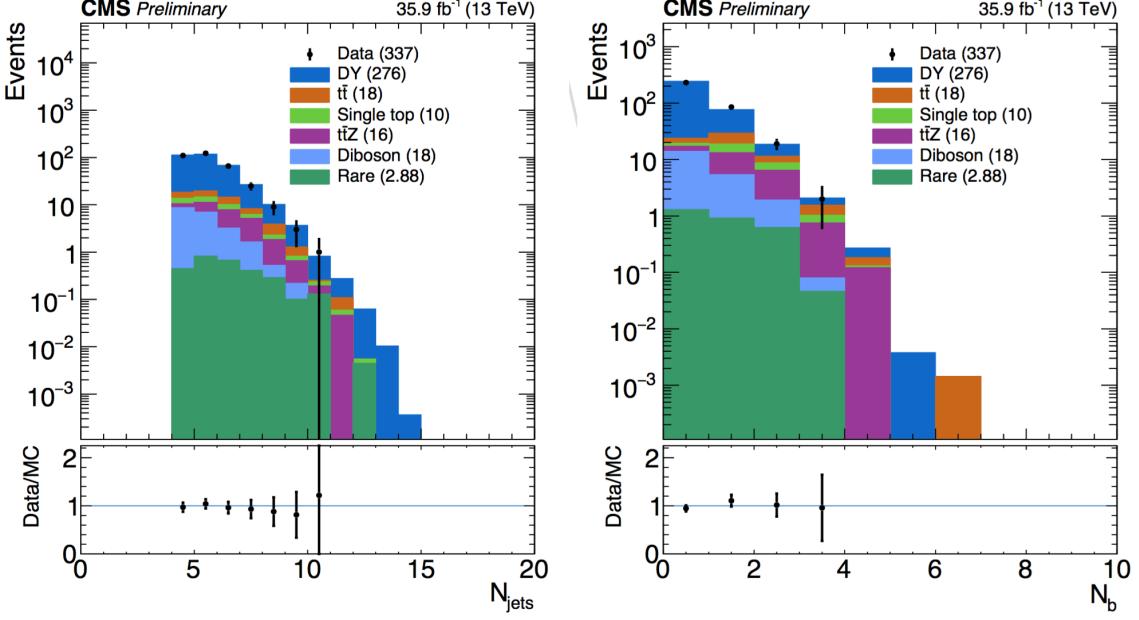


Figure 5.12: Shown are data/MC comparisons for the N_j (left) and N_b (right) distributions after applying both the N_j -dependent shape corrections (S_γ) and the global normalization scale factor (R_{norm}).

423 5.2 Results

424 In this section the results for the final estimation of the of the $Z \rightarrow \nu\bar{\nu}$ are presented.
 425 The current study includes preliminary results using only data obtained at the CMS detec-
 426 tor during 2016. The results for this study are intended to confirm the assumption that the
 427 additional $\gamma +$ jets control region introduced in this analysis reduce the overall uncertain-
 428 ties obtained in the 2016 analyses (described in ??). Furthermore, this study is intended
 429 as a benchmark for future analyses of the SUSY stop group based in Fermilab and will be
 430 the method used for the 2017 CMS data.

431 5.2.1 Systematics

432 Two categories of uncertainties for the $Z \rightarrow \nu\bar{\nu}$ prediction are considered: uncertain-
 433 ties that are associated to the use of MC simulation and the uncertainties specifically
 434 associated to the background prediction method. Several sources are acknowledged in the
 435 first category mentioned such as PDF and renormalization/factorization scale choices, jet
 436 and p_T^{miss} energy scale uncertainties b-tag scale factor uncertainties, and trigger efficiency
 437 uncertainties. Given that the simulation sample is normalized to data in the tight control

region, uncertainties associated with the luminosity and cross-section are excluded. In addition, the overall $Z \rightarrow \nu\bar{\nu}$ statistical uncertainty from MC simulation is also taken into account.

441

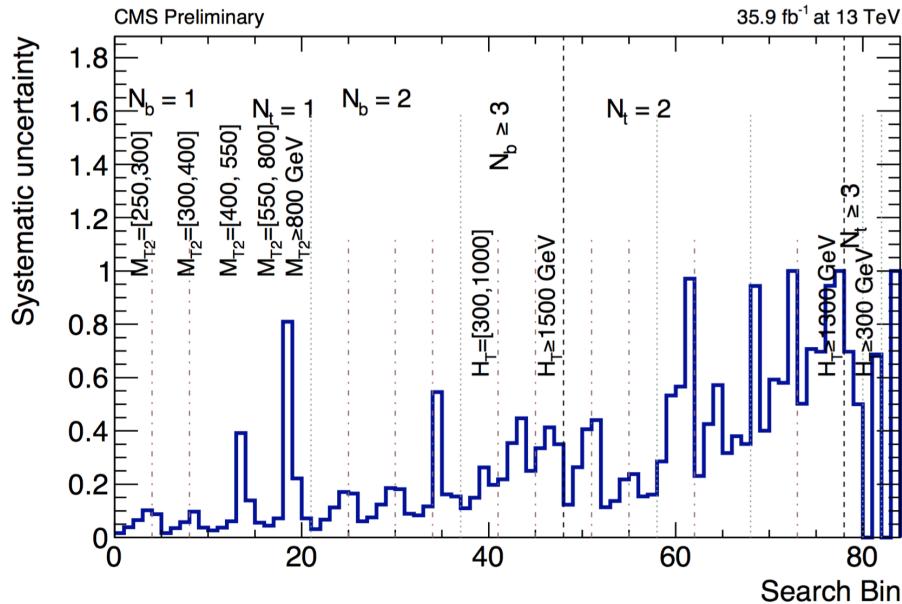


Figure 5.13: Systematic uncertainty in the final prediction, as a function of the search bin, associated to the MC statistics.

442 The statistical uncertainty associated with each bin in the MC is propagated as a sys-
 443 tematic uncertainty. The relative uncertainty per bin can be see in [Figure 5.13](#). It shows
 444 that the uncertainties for the MC vary from as low as 1% up to 81% and even 100% in
 445 some regions. Since the final estimation is scaled using the global normalization factor
 446 from the tight $\mu\mu$ control region (R_{norm}), the total uncertainty, due to limited amounts of
 447 events in data, is propagated in the final prediction. This is also true for the $S_\gamma(N_j)$ scale
 448 factor, in which the residual differences in search variables other than N_j are evaluated in
 449 the loose photon control region. Both the uncertainty arising from the N_j re-weighting
 450 as well as the residual differences are evaluated together. The uncertainty from R_{norm} is
 451 propagated as a flat value of 7.9% uncertainty per each search bin.

452 5.2.2 $Z \rightarrow \nu\bar{\nu}$ Estimation for the Search Bins

453 The final estimation for the $Z \rightarrow \nu\bar{\nu}$ background calculated for all 84 search bins is
 454 shown in [Figure 5.14](#). The statistical uncertainty in bins that have zero events is treated
 455 as the average weight (the sum of the weights squared over the weight) times the poisson
 456 error on 0 which is 1.8. This average weight is calculated on the basis of a relaxed cut in
 457 which $N_b \geq 2$ is required. For comparison, a cut in which $N_t > 2$ where two tops are
 458 fake for the $Z \rightarrow \nu\bar{\nu}$ is used.

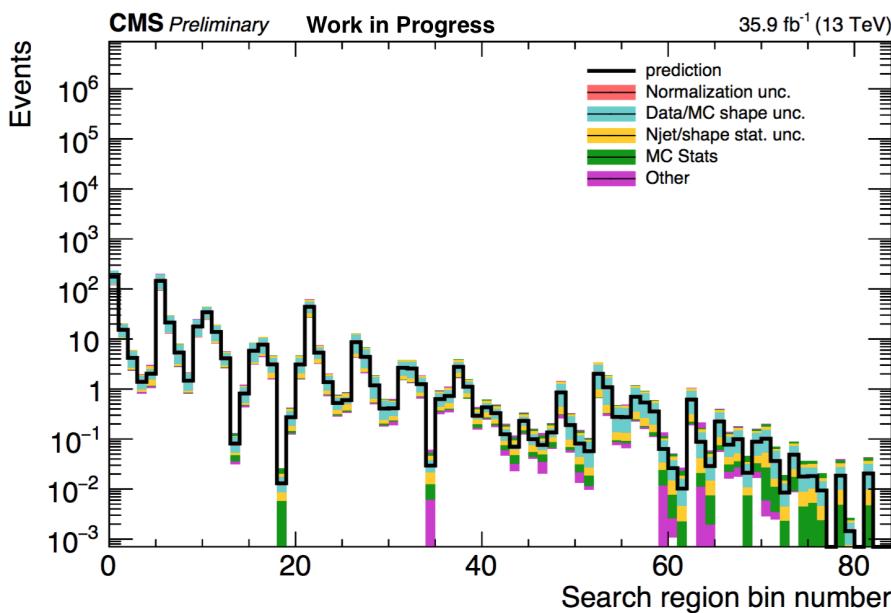


Figure 5.14: $Z \rightarrow \nu\bar{\nu}$ background prediction for all search bins, including the breakdown of the various uncertainties.

⁴⁵⁹ **Chapter 6**

⁴⁶⁰ **References**

⁴⁶¹ [1] CERN, “Processing what to record?,” 2018.

⁴⁶² [2] CMS, “The CMS Computing Project,” tech. rep., CERN, 2005.

⁴⁶³ [3] Coursera, “Machine learning,” 2018.

⁴⁶⁴ [4] A. S. Walia, “Types of optimization algorithms used in neural networks and ways to
⁴⁶⁵ optimize gradient descent,” 2018.