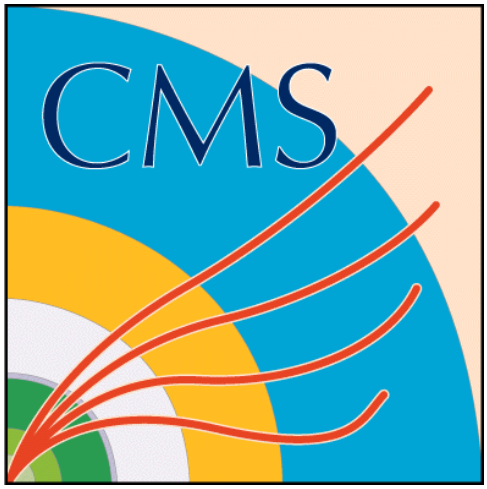
Project Proposal for Google Summer of Code 2021

End-to-End Deep Learning Regression for Measurements with the CMS Experiment



Mentors:

- **Davide DiCroce** (University of Alabama)
- **Darya Dyachkova** (KGI Minerva)
- **Shravan Chaudhari** (BITS Pilani Goa)
- **Emanuele Usai** (Brown University)
- **Michael Andrews** (Carnegie Mellon University)

Anis Ismail
Lebanese American University
anis.ismail@lau.edu

Project Design and Implementation

1. Introduction

Experiments conducted at the Large Hadron Collider (LHC) are the source of the most important discoveries in new physics. One of the most prominent experiments is the Compact Solenoid (CMS), whose results rely on the reconstruction and detection of particles, jets, and topologies from low-level detector data using an End-to-End Deep Learning (E2E) model. This project intends to investigate the use of multiple Deep Learning architectures to learn the properties of simulated top quark pair events and successfully integrate the best performing architecture into the E2E CMSSW prototype.

1.1 Synopsis

Discovering new science at the CMS experiment at LHC depends on extracting particle events and their properties from their corresponding background signals. In fact, traditional approaches for measurement might lose information over the process and limit the exhaustive search for physics beyond the Standard Model (SM). This project will make use of simulated top quark events to train Deep Learning models for particle properties prediction, such as top-quark mass. The top-quark mass (mt) is a crucial parameter of the Standard Model. A precise measurement of this quantity is an important input to global electroweak fits constraining the properties of the Higgs boson and other models for physics beyond the SM.
The simulated data can take the form of image matrices. Each channel belongs to the hits on the layers of the CMS apparatus, such as the electromagnetic calorimeter (ECAL) and brass and scintillator hadron calorimeter (HCAL).The image-like structure of the data allows the use of novel Deep Learning architectures such as Convolutional Neural Networks. These networks can benefit from the large size of the data generated to accurately learn the data structure and successfully predict the top-quark mass among other properties.
In summary, this project intends to develop an end-to-end trained Deep Neural Network that accurately performs regression for top-quark properties measurement give simulated events. Furthermore, this project investigates different deep learning architectures then propose the most accurate model for future integration in the E2E CMSSW prototype.

2. Deliverables

2.1 Mandatory Deliverables

The following are the mandatory deliverables of the project to be completed within the duration of the project.

- **Trained weights** of the end-to-end regression model that estimate the properties of a simulated top quark pair event
- **Extended package** of the E2E CMSSW inference prototype with the components needed to run inference with the model.
- **A jupyter notebook** documenting the training and testing process of the multiple deep learning models.

3. Implementation

3.1 Data Exploration

A first important step in developing the neural network is to understand the data in hand. This step might include multiple techniques:

- **Understanding the statistical distribution of the input features**: this step investigates the features to understand the type of scaling needed at the processing phase (Standardization, Normalization, Logarithmic transformation
- **Data visualization:** This step gives more insights on the data and help understand the effect of data sparsity on the choice of the neural network **architecture later.**
- **Features correlation with the target feature:** This step investigates the correlations between the features and the outcome and is a first step towards feature selection. This is crucial to narrow the feature space and might help produce a simpler and faster model.

3.2. Data Preprocessing

Based on the knowledge acquired from the first step, several transformations will be applied to the data in hand, including feature scaling, selection, and reshaping (reshaping is applied for model compatibility).
This step also includes splitting the data into training, validation and testing sets. This will help check whether the models have learned a valid hypothesis that can generalize well (i.e. check whether the model learned the true relationship between the features and the output).
Setting the random seed to a fixed value is crucial for code reproducibility

3.3 Framing the Problem

Based on the structure of the data and the task in hand, the problem can be framed as a **classification** (distinct target classes) or **regression** (continuous target variable) task.
It should be noted that the evaluation metrics should be properly defined based on the task. The evaluation metrics could be the average class accuracy, precision and recall for classification tasks or root mean square error for regression tasks.

3.4. Model Training and Selection

At this stage, multiple models based on the literature will be compared. Possible architectures include:

- Fully Connected Neural Networks (FCNNs): The simplest type of neural network, expects data samples to be 1-dimensional each.
- Convolutional Neural Networks (CNNs): A novel type known for superior performance compared to FCNNs on image-like structures.
- Resnet Architecture: Deeper model with residual networks that achieves lower errors than CNNs for image-like data.

3.5 Hyperparameter tuning

For each of the selected models, a hyperparameters search will be executed to find the parameters that maximize the chosen metrics on the validation set. The model with the best parameters is then selected to be tested on the test set.

3.6 Evaluating the Performance of the Model

After selecting the model maximizing the metric on the validation set, this model will be tested on the test data to make sure the model is able to perform well on unseen data.

3.7 Wrapping the Model for Inference

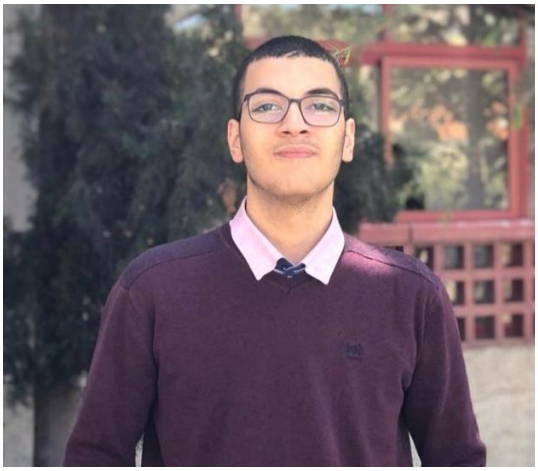
Once the best model is selected, it needs to be integrated into the E2E CMSSW inference engine. This process requires saving the weights of the best model then refactoring the training and inference code to follow the CMS naming, coding, and style rules for packaging in Python.

4. Timeline

Week Nb	Duration	Objective
Week 0	May 17 - June 6	- Community Bonding - Finalize Phase 1 Objectives with mentors
Week 1	June 7 - June 13	- Read the Literature for Deep Learning in CMS experiment - Explore the data - Preprocess the data
Week 2	June 14 - June 20	- Frame the problem - Develop the code for the architectures chosen - Train the chosen models
Week 3	June 21 - June 27	
Week 4	June 28 - Jul 4	- Perform Hyperparameter Tuning for the chosen architectures - Test the best models
Week 5	Jul 5 -Jul 11	
Phase 1 Evaluations		
Week 6	Jul 12 - Jul 18	- Refactor the code inside the jupyter notebook and explain the steps - Prepare the components while following the guidelines of CMSSW
Week 7	Jul 19 - Jul 25	
Week 8	Jul 26 - Aug 1	- Integrate the components as an extension to the E2E CMSSW package - Write unit tests for the package extension
Week 9	Aug 2 - Aug 8	- Write Documentation for the project - Write Report Comparing the chosen architectures
Week 11	Aug 9 - Aug 15	
Final Week		
Week 12	Aug 16 - Aug 23	- Submit Package Extension - Submit Documentation - Submit Mentor Evaluation

About Me

1. Background



Name: Anis Ismail
LinkedIn: https://www.linkedin.com/in/anisdismail
Github: https://github.com/anisdismail
Medium: https://medium.com/@anisismail09
Email: anis.ismail@lau.edu
Phone Number: +96170637488

Exploring the vast field of Artificial Intelligence was always a topic of interest for me. I am always fascinated by how Machine Learning can be applied to improve a multitude of domains. This passion has been reflected in my career choices and academic research. I am currently a **Senior Computer Engineering student** at the **Lebanese American University** with an emphasis in **Software and Communication** and a **GPA of 3.82/4.0**.

My journey at LAU strengthened my interest in ML through a multitude of courses, competitions, and lectures. For instance, I have been working as an **ML researcher** at the Electrical and Computer Engineering Department at LAU for **more than two years**. My research focuses on **optimizing the decisions of Unmanned Aerial Vehicles (UAVs)** in Vehicular Connected Networks to minimize packet delivery delay. I developed a deep reinforcement learning algorithm that **outperformed the previous solutions with 60% improvement** (the research paper is currently pending publication). Also, I am currently working on my undergraduate research project on the automated generation of chemical reactions with Generative Language Models.

My passion for Machine Learning led me to join **BMW Group as a Machine Learning and Computer Vision intern** in November 2020. I am working closely with the BMW Techoffice Munich to develop a **Deep Learning optical character recognition API** to be released open-source on their Github repository. Furthermore, I am also working with

Idealworks, a new BMW Spinoff, to develop **3D Collision Avoidance Components to be integrated into the navigation stack** of the Smart Transport Robot.

Being part of a supportive AI community has been a crucial part of my Artificial Intelligence journey. I am currently a **Technical Lead at Beirut AI**, one of the most active AI communities in the MENA region, and I am responsible for supervising and guiding instructors in preparing their workshops for Beirut AI quarterly AI Weekends. **I also delivered the Introduction to Deep Learning with Pytorch workshop** in the last AI Weekend in November 2020. Being part of Beirut AI since my early days in AI has left a deep influence on my work, as it has inspired me to give back to the community and helped me understand the power of open-source activities. This experience has led me to become the **president of the AI Club at my university** for the last academic year, organizing so far **more than 10 AI-centered events and workshops**.

I came across the ML4SCI initiative last year at GSOC 2020, when I was inspired to discover how Artificial Intelligence is being applied at the CMS Experiment at CERN. Although I was not able to apply for last year's edition, I worked on a short project about **muon momentum prediction using Deep Learning** which was published on my blog and TowardsDataScience publication. I believe that I am now more equipped for this year's edition of GSOC to apply my Machine Learning knowledge to the exciting problems of ML4SCI.

2. My Machine Learning Experience

Detecting Muon Momentum in the CMS Experiment at CERN using Deep Learning

- **Duration**: Aug–Nov 2020
- **Description**:
  Developed multiple Deep Learning approaches to improve the precision of muon detector systems and reached an accuracy of 85% when testing on Monte-Carlo simulated data.
- **Technologies Used**: Pandas, Numpy, Matplotlib, Tensorflow

Automated Generation of Chemical Reactions with Generative Language Models

- **Duration:** Jan 2020–Present
- **Description**:
  - Tested and compared multiple generative language models architectures for augmenting existing chemical reactions datas.
  - Investigated the effect of Generative Adversarial Models (SeqGAN, RankGAN and LeakGAN) on the quality of the data generated.
- **Technologies Used**: Pandas, Numpy, Matplotlib, Nvidia Rapids, Pytorch

Scheduling the Movement of Unmanned Aerial Vehicles in Connected Vehicular Networks Using Deep Reinforcement Learning

- **Duration:** Sep 2019–Present
- **Description**: Developed a Dueling Deep Q-Network in **Python and Java** to minimize average packet delay in UAVs assisted VANETs and reached a **60% reduction** in average packet delay in single UAV assisted VANETs case.
- **Technologies Used**: Numpy, Tensorflow, Java

Plant Images Segmentation with Deep Learning

- **Duration**: Jul–October 2020
- **Description**:
    - Worked in my internship at Zaka on an image segmentation project to identify weeds and their stems from crops.
    - Experimented with multiple Convolutional Neural Networks architectures to select the best performing one.
    - Developed domain-specific metrics to evaluate the performance of multiple architectures on training and testing data.
- **Technologies Used**: Numpy, Tensorflow

3. Extra-Curricular Activities

- **President** of the Artificial Intelligence Club at LAU
- **Technical Instructor Lead** at Beirut AI
- **Member** of IEEE Student Club at LAU

4. Awards and Honors

- **Distinguished List** of the School of Engineering since Fall 2017
- **Member of LAU Honors Program**
- **Full Merit Scholarship recipient**
- Won **first runner-up** in BMW Group Beirut AI Hackathon 2019 at LAU
- **Ranked among top ten** in the Lebanese Collegiate Programming Competition 2019
- **Ranked among top five teams** in LAU Collegiate Programming Competition 2019