**Google Summer of Code 2021**
Project Proposal
Deep autoencoders for ATLAS data compression


**Name:** Georgios Dialektakis
**Affiliation:** Aristotle University of Thessaloniki
**Program:** Postgraduate Student, Data & Web Science
**Mentors:** Caterina Doglioni, Alex Gekow, Antonio Boveia, Baptiste Ravina, Lukas Heinrich
**Email:** geo4diale@gmail.com
**Skype:** gdialektakis
**Linkedin:** www.linkedin.com/in/george-dialektakis-404830181/
**Address:** Gymnasiarchou Dimitriadou 5, Thessaloniki, 54655
**Phone:** +30 694 997 2822


Google Summer of Code

# Contents


**1 Abstact** **3**

**2 Project Information** **3**
2.1 Data Compression . . . . . 4
2.2 Related Work . . . . . 4

**3 Project Contribution** **4**
3.1 Benchmarking . . . . . 4
3.2 Variational and Adversarial Autoencoders . . . . . 5
3.3 Combination of Different Compression Algorithms . . . . . 6
3.4 Anomaly Detection . . . . . 6
3.5 Expected Results . . . . . 7
3.6 Feasibility of Deliverables . . . . . 7

**4 Project Timeline** **8**
4.1 Bonding period (May 17th - June 7th) . . . . . 8
4.2 Coding period (June 7th - August 23rd) . . . . . 8
4.3 Scheduled Conflicts . . . . . 9

**5 Benefits to Community** **9**

**6 Biographical Information** **10**
6.1 Academic . . . . . 10
6.2 Open-Source Projects . . . . . 10
6.3 Programming Background . . . . . 10
6.4 Personal Motivation . . . . . 10

**7 References** **11**

# 1 Abstact

Storage is one of the main limiting factors to the recording of information from proton-proton collision events at the Large Hadron Collider (LHC), at CERN in Geneva. Hence, the ATLAS experiment at the LHC uses a so-called trigger system, which selects and transfers interesting events to the data storage system while filtering out the rest. However, if interesting events are buried in very large backgrounds and difficult to identify as a signal by the trigger system, they will also be discarded together with the background. To alleviate this problem, different compression algorithms are already in use to reduce the size of the data that is recorded. One of those state-of-the-art algorithms is an autoencoder network that tries to implement an approximation to the identity, $f(x) = x$, and given some input data, its goal is to create a lower-dimensional representation of those data in a latent space using an encoder network. Then using this latent representation and a decoder network, the model can reconstruct the input.

The goal of this project is to experiment with autoencoders for data compression in-depth and optimize their performance in reconstructing the ATLAS event data. For this reason, except for the standard autoencoders, I propose two different families of autoencoders, the variational and the adversarial autoencoder which are more complex than the traditional one. Furthermore, various combinations of different compression algorithms are proposed to evaluate their ability to compress and construct accurate representations of the input data. Finally, the studied autoencoders are implemented to detect anomalies in the data. The proposed implementations will be a decisive contribution towards future testing and analysis for the ATLAS experiment at CERN and will assist overcome the obstacle of needing much more storage space than in the past due to the increase in the size of the data generated by the continuous proton-proton collision events in CERN's Large Hadron Collider.

# 2 Project Information

At CERN's Large Hadron Collider (LHC) [6], proton collisions are performed to study the fundamental particles and their interactions. To discover and record the outcome of these collisions, the ATLAS detector [5] is used, which is one of many detectors that have been developed at the LHC. Each second, there are roughly $10^9$ events or collisions occurring inside the ATLAS detector and storage is one of the main limiting factors to the recording of information from these events, since it is impossible to save all those events. To keep the most relevant information, the ATLAS experiment uses trigger systems, which selects and transfers interesting events to the data storage system while filtering out the rest. Storage of these events is restricted by the amount of information to be stored and a decrease of the event size can allow for an analysis of the events that was not previously possible.

The structure of this proposal is organized as follows, the rest of the section presents the data compression method using Autoencoders and some previous work on this project. In section 3, my proposed contribution to this project is presented along with the

implementation and the expected results. In section 4, the project timeline is presented, section 5 contains some benefits of this project to the community, and finally, the last section of this proposal presents biographical information and personal motivation regarding this project and the participation in this year's Google Summer of Code.

### 2.1 Data Compression

Data compression is the process of encoding information using fewer dimensions or less size than the original representation [15]. One of the current state-of-the-art data compression techniques is deep compression using Autoencoders [14]. Typically, an Autoencoder (AE) is an Artificial Neural Network that tries to implement an approximation of the identity function. It consists of an encoder that performs data compression by projecting the input high dimensional data to a lower-dimensional latent space, and a decoder that performs the reversal of the process (decompression) by reprojecting the latent space back to the dimensions of the original data, also known as the reconstruction phase. The latent space representation can then be used as a compressed representation of the input and can be stored along with the decoder network to reconstruct the data.

### 2.2 Related Work

Previous work [17] examines the use of deep neural autoencoders to compress data for jets, which is the most common type of particle. Different autoencoder variants are tested with different widths and depths. The work shows promising results as the AEs manage to successfully compress and decompress simple hadron jet data and preliminary results indicate that the reconstruction quality is good enough for certain applications where high precision is not paramount. The work in [13], further investigates the use of AEs for compression of trigger level analysis (TLA) data as used by the ATLAS experiment, while showing that it may however be difficult to generalize between different datasets. Moreover, the use of different compression methods used sequentially, by so-called float truncation then followed by autoencoder compression is evaluated.

## 3 Project Contribution

### 3.1 Benchmarking

The first step of this project will be to benchmark an existing autoencoder to compress ATLAS data. Benchmarking includes testing different autoencoder architectures regarding the number of hidden layers and their size, the input-output, and the latent space dimensions, while also fine-tuning all necessary hyper-parameters of the network, such as the learning rate, the optimizer, the loss function, etc. This procedure aims at optimizing the AE to provide meaningful and accurate representations of the input data while also providing high-quality decompressed data with respect to original inputs. Moreover, during the benchmarking procedure, the costs of training and decompressing the data in the context of a resource-constrained system such as the ATLAS trigger will be examined.

## 3.2 Variational and Adversarial Autoencoders

Following the benchmarking, I intend to use different families of autoencoders for the ATLAS data compression task to examine their performance compared to the standard autoencoder and their quality of representations they produce. First, I propose the use of the Variational Autoencoder (VAE) [1]. The VAE differs from traditional autoencoders, as it performs a regularization technique in the latent encoding. Specifically, every input of the VAE is encoded as a distribution over the latent space rather than a single point. Next, a point from the latent space is sampled from that distribution, which is then fed to the decoder to compute the reconstruction error and backpropagate it through the network. The VAE ideally wants to produce encodings that are as close as possible to each other while still remaining separate. In this way, it allows smooth interference and enables the generation of new samples. The generation of new synthetic data may be very important for the ATLAS experiment, as it provides more data for further physics analysis without the need of performing numerous proton-proton collisions and storing their information.

Secondly, I propose the use of Adversarial Autoencoders (AAE) which are based on Generative Adversarial Networks (GANs) [9]. Adversarial Autoencoders (AAE), proposed by Makhzani et al. [11], were initially designed for semi-supervised classification, unsupervised clustering, and disentangling style in the image domain. However, in my diploma thesis [8] as an undergraduate student, I introduced the AAE for dimensionality reduction of the Belief State Space which serves as input to a Dialogue system. The main advantage of this work was to produce lower-dimensional robust representations of the input vectors, which augmented the ability of two Reinforcement Learning algorithms to achieve state-of-the-art performance. The AAE is a generative autoencoder similar to a standard AE; however, an adversarial network, which we refer to as the discriminator, is added on top of the autoencoder's latent space vector, which guides the encoder's output to meet the prior distribution of the input data. The architecture of the AAE is illustrated in figure 1 [12], where $q(z|x)$ denotes the output of the encoder for an input $x$, $z$ is the latent encoding drawn from $q(z|x)$, $z'$ is the real input of the discriminator with the prior distribution, $p(x|z)$ is the decoder output given $z$, and $x\_$ is the reconstructed input. The AAE, similar to the VAE, can be used for both data compression and the generation of new synthetic data

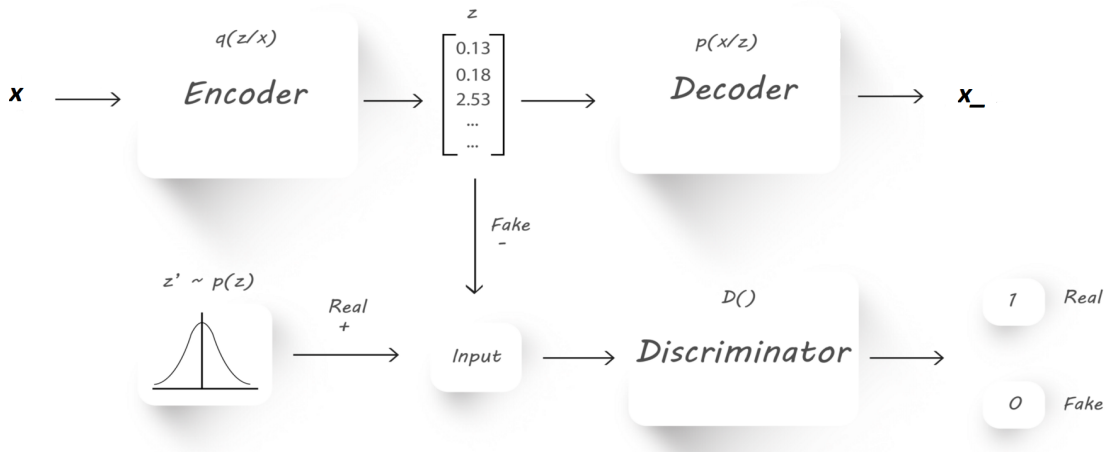


Figure 1: Adversarial Autoencoder architecture [12]

## 3.3 Combination of Different Compression Algorithms

The next approach I am going to follow for data compression of the ATLAS experiment is to try out various combinations of different compression algorithms and compare their performance to the methods proposed in the previous subsection where we only make use of autoencoders. In specific, I would like to examine the autoencoders combined with a Matrix Factorization algorithm, the Principal component analysis (PCA) [16] which is commonly used for dimensionality reduction by projecting each data point onto only the first few principal components to obtain lower-dimensional data while preserving as much of the data's variation as possible [10].

Even though PCA can be powerful, it often misses important non-linear structures in the data. For that reason, we can use Manifold Learning [4]. Manifold Learning can be thought of as an attempt to generalize linear frameworks like PCA to be sensitive to non-linear structure in data. Though supervised variants exist, the typical manifold learning problem is unsupervised: it learns the high-dimensional structure of the data from the data itself, without the use of predetermined classifications. A manifold learning algorithm I aim to use is Spectral Embedding [3] which is an approach to calculate a non-linear embedding by implementing Laplacian Eigenmaps, which finds a low dimensional representation of the data using a spectral decomposition of the graph Laplacian. Therefore, in this task I would like to evaluate the performance of the autoencoders when combined with Spectral Embedding.

## 3.4 Anomaly Detection

The final task of this project is to utilize an Autoencoder for anomaly detection. To achieve this functionality, the AE is first trained on normal instances of data that is known not to be anomalous [7]. It uses the reconstruction error as the anomaly score. Data points with high reconstruction error are considered to be anomalies. After training, the AE will reconstruct normal data very well, while failing to do so with anomaly data

which the autoencoder has not encountered. Therefore, the first method I am going to evaluate is the application of a standard autoencoder as an anomaly detection algorithm.

Next, I would like to examine the Variational Autoencoder for anomaly detection. A VAE uses the reconstruction probability which is a probabilistic measure that takes into account the variability of the distribution of variables [2]. The reconstruction probability has a theoretical background making it a more principled and objective anomaly score than the reconstruction error of a traditional autoencoder.

## 3.5 Expected Results

The following list describes the expected results after the completion and the final submission of this project:

- An optimized standard autoencoder for data compression with documentation and figures demonstrating its performance, plots of the compressed-variables and plots of the performance on unseen data.
- An optimized Variational autoencoder for data compression with documentation and figures demonstrating its performance, plots of the compressed-variables and plots of the performance on unseen data.
- An optimized Adversarial autoencoder for data compression with documentation and figures demonstrating its performance, plots of the compressed-variables and plots of the performance on unseen data.
- A data compression model composed of the combination of a standard autoencoder with PCA, documentation and figures of its performance.
- A data compression model composed of the combination of a standard autoencoder with Spectral Embedding algorithm, documentation and figures of its performance.
- An optimized standard autoencoder for anomaly detection with documentation and plots demonstrating the performance of the algorithm.
- An optimized Variational autoencoder for anomaly detection with documentation and plots demonstrating the performance of the algorithm..

## 3.6 Feasibility of Deliverables

As far as it concerns the feasibility of the above deliverables, I have to mention that I have worked with deep autoencoders thoroughly in the past during my diploma thesis [8]. In specific, I implemented a Variational AE which took me about 2 weeks, an Adversarial AE which took me about 3 weeks, and their denoising variants of those two AEs. For more details, you can take a look at the code at: https://github.com/gdialektakis/Statistical-Dialogue-Systems-with-Adversarial-AutoEncoders and read my thesis [8]. Therefore, having implemented the above networks in the past, I can now reuse that code and adjust it to the ATLAS project needs for compression of the ATLAS data. Moreover,

for the anomaly detection using Variational AEs, I intend to rely on this specific work in [2] and implement the VAE as described using the reconstruction probability to detect anomalies in the given data.

# 4 Project Timeline

## 4.1 Bonding period (May 17th - June 7th)

The Community Bonding period consists of 3 weeks. During this period the mentors of this project and I are going to get to know each other. Moreover I will spend this period of time to learn more details about the CERN's ATLAS experiment and prepare all the necessary tools to be ready to start coding on the 7th of July.

## 4.2 Coding period (June 7th - August 23rd)

This year we are going to have 10 weeks for the coding period of 175 hours in total.

- **1st week** and **2nd week** (June 7th - June 21st)
  Benchmark the existing Autoencoder. I will create various plots demonstrating the Autoencoder's performance and plots of the compressed data. During this period I am going to conduct various tests and experiments to fine tune the network.
- **3rd week** and **4th week** (June 21st - July 5th)
  1. Implementation of Variational Autoencoder for data compression.
  2. Optimization and fine tuning.
  3. Documentation and plots of the performance of the Variational Autoencoder.
  4. Apply Variational Autoencoder on generation of synthetic data based on the input ATLAS data (optional).
- **5th week** and **6th week** (July 5th - July 19th)
  1. Implementation of Adversarial Autoencoder for data compression.
  2. Optimization and fine tuning.
  3. Documentation and plots of the performance of the Adversarial Autoencoder.
  4. Use Adversarial Autoencoder for synthetic data generation based on the input ATLAS data (optional).
- **7th week** and **8th week** (July 19th - August 2nd)
  Combinations of different data compression algorithms.
  1. Implement the combination of the standard Autoencoder with Principal Component Analysis algorithm.
  2. Manifold Learning (such as Spectral Embedding and Locally Linear Embedding) with the standard Autoencoder (optional).

3. Documentation of the implemented combinations.

- **9th week** (August 2nd - August 9th)

  1. Examine the performance of the standard Autoencoder as an anomaly detection algorithm.
  2. Implement a Variational Autoencoder for anomaly detection.
  3. Implement an Adversarial Autoencoder for anomaly detection. (optional)

- **10th week** (August 9th - August 16th)
  Final code optimizations, documention and final submission.

**Comment**: The above time schedule is a **worst-case** schedule. However, if I get stuck in some step that I do not expect at the moment, we will not move on the implementation of some optional deliverables. Of course if the coding exceeds my present expectations I will discuss with the mentors new additions and implementations.

### 4.3 Scheduled Conflicts

The only period during the 10 weeks of this project that I am going to have a scheduled conflict is the examination period (2 weeks): Monday 21/6/2021 – Monday 5/7/2021 in my postgraduate programme in Aristotle University of Thessaloniki. During this period I will be examined on 4 courses, 2 in the first week and 2 on the second week. Even though I will have to study for these 4 courses, I will still be able to work on this project and deliver the scheduled implementation of the Variational Autoencoder as proposed in the above project timeline.

## 5 Benefits to Community

The project is expected to be beneficial to both (a) educational and (b) research for the ATLAS experiment at CERN. For (a), university students, who are studying or are interested in High Energy Physics and machine learning, will have access to use cutting-edge machine learning techniques for both data compression and detection of anomalous events. For (b), this study can be treated as a proof-of-principle for future data compression methods for the ATLAS experiment. For the planned experimental upgrades in 2026, the techniques used in this work may assist to overcome the obstacle of needing much more storage space than in the past due to the increase in the size of the data generated by the continuous proton-proton collision events in CERN's Large Hadron Collider.

# 6 Biographical Information

## 6.1 Academic

- Postgraduate Student in Data and Web Science at Aristotle University of Thessaloniki.
- Undergraduate student in the Department of Electrical and Computer Engineering at Technical University of Crete.
  - Thesis: "Adversarial Learning in Statistical Dialogue Systems". (2020) [8]
    Advisor: Michail Lagoudakis (TUC)

## 6.2 Open-Source Projects

- Transfer Learning for Image Classification:
  https://github.com/gdialektakis/Transfer-Learning
- Sentiment Classification on Greek smartphone reviews:
  https://github.com/gdialektakis/Sentiment-Classification-on-Greek-smartphone-reviews
- Graph-Based-Movie-Recommendation:
  https://github.com/gdialektakis/Graph-Based-Movie-Recommendation
- Graph-Based-Movie-Recommendation:
  https://github.com/gdialektakis/Graph-Based-Movie-Recommendation
- Scalable Processing Of Dominance Based Queries:
  https://github.com/gdialektakis/Scalable-Processing-Of-Dominance-Based-Queries

## 6.3 Programming Background

- Highly experienced in Deep Learning and Autoencoders.
- Skillful in Python and Deep Learning libraries, such as Tensorflow, Keras, PyTorch.
- Completed successfully various projects in my Postgraduate programme in Data Science. (more on my GitHub page https://github.com/gdialektakis?tab=repositories)

## 6.4 Personal Motivation

I am specially interested in machine learning algorithms and their applications in real world problems. That is the reason I am staying active by developing my skills and learning about state-of-the-art Deep Learning models and libraries such as Pytorch.

This year's Google Summer of Code is a great opportunity to continue and significantly upgrade my skills and experience by developing deep learning models and applying them in an exciting real world experiment, such as the ATLAS experiment for data compression with Deep Autoencoders. The main reason behind my selection of this subject is my great and deep knowledge in Autoencoders for data compression and dimensionality reduction, as I have previously worked on a similar subject during my diploma thesis at the Technical University Of Crete for the diploma degree in Electrical and Computer Engineering [8]. Finally, I find the idea of contributing to a scientific project quite appealing, as I am going to cooperate with my mentors who are highly experienced in this field and I believe they can offer me quite a lot of knowledge and advice.

To find out more details about my skills and background you can visit my LinkedIn profile: https://www.linkedin.com/in/george-dialektakis-404830181/

# 7 References


[1] Jinwon An and Sungzoon Cho. Variational autoencoder based anomaly detection using reconstruction probability. *Special Lecture on IE*, 2(1):1–18, 2015.

[2] Jinwon An and Sungzoon Cho. Variational autoencoder based anomaly detection using reconstruction probability. *Special Lecture on IE*, 2(1):1–18, 2015.

[3] Mikhail Belkin and Partha Niyogi. Laplacian eigenmaps for dimensionality reduction and data representation. *Neural computation*, 15(6):1373–1396, 2003.

[4] Lawrence Cayton. Algorithms for manifold learning. *Univ. of California at San Diego Tech. Rep*, 12(1-17):1, 2005.

[5] CERN. The atlas detector. https://home.cern/science/experiments/atlas.

[6] CERN. The large hadron collider. https://home.cern/science/accelerators/large-hadron-collider.

[7] Zhaomin Chen, Chai Kiat Yeo, Bu Sung Lee, and Chiew Tong Lau. Autoencoder-based network anomaly detection. In *2018 Wireless Telecommunications Symposium (WTS)*, pages 1–5. IEEE, 2018.

[8] Georgios Dialektakis. Adversarial learning in statistical dialogue systems. School of Electrical and Computer Engineering, Technical University of Crete, Chania, Greece, http://purl.tuc.gr/dl/dias/61C5E296-DF97-49E6-9F34-7719775576DA, 2020. Diploma Work.

[9] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. In *Advances in neural information processing systems*, pages 2672–2680, 2014.

[10] Lorraine Li. Principal component analysis for dimensionality reduction. *Towards Data Science*, 2019.

[11] Alireza Makhzani, Jonathon Shlens, Navdeep Jaitly, Ian Goodfellow, and Brendan Frey. Adversarial autoencoders. *arXiv preprint arXiv:1511.05644*, 2015.

[12] Naresh Nagabushan. A wizard's guide to adversarial autoencoders: Part 2, exploring latent space with adversarial autoencoders.

[13] Erik Wallin. Tests of autoencoder compression of trigger jets in the atlas experiment, 2020. Student Paper.

[14] Yasi Wang, Hongxun Yao, and Sicheng Zhao. Auto-encoder based dimensionality reduction. *Neurocomputing*, 184:232–242, 2016.

[15] Wikipedia. Data compression. https://en.wikipedia.org/wiki/Data_compression.

[16] Svante Wold, Kim Esbensen, and Paul Geladi. Principal component analysis. *Chemometrics and intelligent laboratory systems*, 2(1-3):37–52, 1987.

[17] Eric Wulff. Deep autoencoders for compression in high energy physics, 2020. Student Paper.