Classification of Higgs boson particle using accelerated machine learning

*Abstract*— In this paper, the use of machine learning algorithms in detecting the Higgs boson particle has been discussed. This study explores the challenges associated with detecting the particle and how machine learning can help overcome these challenges. The paper also discusses the use of accelerated computing frameworks and specialized hardware to improve the efficiency and speed of machine learning algorithms. The Higgs boson dataset has been analyzed using exploratory data analysis (EDA) techniques, and data pre-processing techniques has been also discussed. The paper suggests potential machine learning (ML) algorithms for classification problems. The ML algorithms which has been proposed for the detection of Higgs boson particle are Random Forest, Gradient Boosting, Support Vector Machine (SVM) and Naïve Bayes. The study also provides guidelines for selecting hyperparameters and evaluating model performance of each model. Overall, the paper highlights the potential of machine learning and accelerated computing for the identification of Higgs boson particle.

Keywords—Machine Learning, Higgs boson, particle, Accelerated Computing, Parallel Computing, Distributed Systems, GPUs, Exploratory Data Analysis (EDA), Data pre-processing, Missing Values, highly correlated features, Outliers, Libraries, APIs, Model selection, Evaluation metrics, Random Forest, Gradient Boosting, Support Vector Machines, Naive Bayes, classification.

# Introduction

Particles are the building blocks of matter. They have physical properties like mass, shape, and size. Particles can have different kinematic properties based on how they move. According to [1], Kinematic properties are numerical values that describe how an object moves, such as its position, speed, acceleration, momentum, and energy. In particle physics, the goal of measuring kinematic properties is to learn more about the basic properties of subatomic particles and the forces that control how they move and interact with each other. In particle physics, kinematic properties are used to explain how electrons, protons, and neutrinos move and interact with each other. Furthermore, it's important to measure kinematic properties for useful things like figuring out which particles come from high-energy collisions or how much energy a particle beam has.

The history of finding particles is long and interesting. It goes all the way back to the early 20th century, when quantum mechanics was invented, and the electron was found. Some of the most important discoveries in particle physics are the ones that led to the "Standard Model." However, scientists are still looking for a basic understanding of matter and the forces that control it. Among the most important questions in particle physics today are about the Higgs boson, dark matter, neutrino oscillations, and other things [2].

The Higgs boson is the last part of the Standard Model, so finding it was one of the most important things to happen in particle physics. In 2012, experiments at the Large Hadron Collider (LHC) proved that the Higgs boson, a particle that gives other particles mass, had been found [3][4]. Physicist [5], model the anticipated signal and background processes using advanced statistical methods and computer simulations to differentiate the Higgs boson signal from the background processes. Then, they look at the data on particle collisions to see if the recorded events match what was expected for the signal and the background. However, they must be used with experimental data and theoretical models because they are not perfect and can only help us understand the basic forces and particles of the universe.

According to [6], Methods of machine learning can play a significant role in overcoming the obstacles connected with the detection of the Higgs boson. These methods can be used to analyse data from particle collisions to identify patterns that are unique to the Higgs boson signal. Hence, the Higgs boson may be located with more precision. These techniques can also aid in finding and reducing data noise. In addition, [7], in the field of particle physics, machine learning can be applied to enhance the methods for designing experiments and analysing data. Using machine learning methods, for instance, the configuration of particle detectors and the selection of data processing techniques can be adjusted to improve the sensitivity and accuracy of Higgs boson discovery. Overall, the application of machine learning techniques to particle physics research has the potential to substantially improve our ability to find and investigate the Higgs boson.

The use of machine learning methods for the detection of the Higgs boson particle requires a large dataset of experimental data from particle collisions. The dataset must include a significant number of Higgs boson production events and background events, which are other particle productions that can counterfeit the Higgs boson signal [8]. As mentioned in [9], the Higgs boson dataset presents numerous challenges, including a substantial amount of noise. This noise may be caused by cosmological radiation, electronic noise, or poor detector calibration. Due to the noise, it may be difficult to distinguish the Higgs boson signature from background events. According to [10], the data's high dimensionality adds an extra difficulty. The dataset includes numerous characteristics, including the energy and momentum of the particles generated by the impact. These features are closely interconnected, making it difficult to analyse them using normal statistical approaches. However, to address these challenges there are several techniques. According to [11], enabling distributed computing, the data will be distributed over numerous nodes or servers, the storage requirements can be reduced with high-performance.

In machine learning, one way to use traditional machine learning approaches like SVM (Support vector machine), Gradient boosted tree and Random Forest [12] [13] for the classification of Higgs boson particle. Although, it requires more computational power and time to handle large amount of data, however these issues can be resolved by using accelerated computing [11]. Another way is to use deep learning algorithms such as convolutional neural networks (CNNs) and recurrent neural networks (RNNs), which can autonomously acquire complicated features from the data and minimise the impacts of noise [14]. In addition, [15], the precision and sensitivity of Higgs boson detection can be improved by adjusting particle detector design using reinforcement learning approach.

The format of the paper is as follows: the second part of the paper will discuss about the background and literature review of the detection of Higgs boson particle. Third part of the paper discusses why accelerated computing frameworks and machine learning approaches can provide a better solution. The Higgs boson dataset's data description and the necessary Exploratory Data Analysis (EDA) are covered in the fourth part. A design strategy for data pre-processing, model architecture selection, discussion, and justification, model training, and evaluation metrics are included in the methodological section's fifth part. Finally, there will be a conclusion paragraph in the last part of the paper.

# Background

The 2012 discovery of the Higgs boson by the Large Hadron Collider (LHC) experiments at CERN was a milestone in the field of particle physics. The LHC, the world's largest and most powerful particle accelerator, is required for the detection of the Higgs boson particle, among other techniques and equipment [16], in addition to the ATLAS (A Toroidal LHC Apparatus) and CMS (Compact Muon Solenoid) detectors [17], which are enormous instruments designed to detect the various particles created by LHC collisions. These detectors contain several sub-detectors, including as tracking detectors, calorimeters, and muon detectors, which are used to monitor the energy, momentum, and charge of particles [17]. Additionally, machine learning (ML) contributed significantly to the discovery of the Higgs boson, as it is an effective data analysis and classification method. In recent years, there has been an increasing interest in the use of accelerated machine learning techniques to improve the Higgs boson discovery. With the discovery of the Higgs boson, which established the existence of the Higgs field, which gives particles mass, significant progress was achieved in particle physics [3]. The identification of the Higgs boson was a difficult task, and machine learning techniques were utilized to enhance the signal-to-noise ratio of the data [12]. The discovery of the Higgs boson, a basic particle predicted by the Standard Model of particle physics, was a major advancement in the science of particle physics. For the identification of the Higgs boson, traditional machine learning algorithms have been applied extensively with deep learning approaches. In this study, I examined some current machine learning-based research on the detection of the Higgs boson.

The authors of this study [12], compares four Machine Learning (ML) approaches for the Higgs Boson Classification Issue using the Pyspark environment. ML algorithms employed included Logistic Regression (LR), Decision Tree (DT), Random Forest (RF), and Gradient Boosted Tree (GBT). Higgs dataset UCI and Higgs dataset Kaggle were utilised. AUC and Accuracy were utilised as the measures. The results demonstrated that the Gradient Boosted Tree (GBT) technique had the highest AUC and accuracy metrics. Using the Higgs dataset Kaggle, it achieved 83% accuracy and 82% AUC, whereas on the Higgs dataset UCI, it achieved 70% accuracy and 70% AUC. The ranking of the classifiers' ability to predict the future was confirmed by tuning and repeated cross-validation.

In another study, authors looked at how a neural network could be used to find situations where the Higgs boson was present. The authors used simulated data from the LHC to train a multi-layer perceptron (MLP) neural network to put events into groups based on what they were like. They found that the neural network was very good at identifying Higgs boson events. This shows how ML could be used in particle physics. The authors learnt the neural network to identify Higgs boson events by giving it simulated data from the LHC. In particular, the neural network was able to predict Higgs boson events with an accuracy of 80% and a recall of 77%. The authors also found that the neural network was able to identify Higgs boson events that had not been seen before [18].

In a recent study [13], an experiment was done to examine the performance of three machine learning algorithms in distinguishing Higgs boson occurrences from the signal. The evaluated methods were histogram gradient boosted decision trees, support vector machines, and neural networks. In terms of identification accuracy, runtime, and memory consumption, the histogram gradient boosted decision tree had the best performance. The histogram gradient boosted decision tree (BDT) had an average test accuracy of 83.88% percent, whereas the SVM and neural network had average test accuracy of 80.35% and 82.4%, respectively. The BDT had the greatest area under the curve (AUC=0.9) of the three examined models, indicating it was the most accurate.

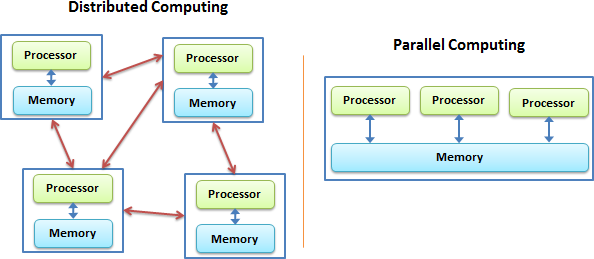
The identification of the Higgs boson at the Large Hadron Collider (LHC) creates a vast amount of data, requiring significant storage capacity and computational time. Often, machine learning techniques require the storage of even greater quantities of data, which might worsen this issue. This can be problematic for data centers and computer resources, as the cost of storing and handling massive databases might become prohibitive. Using data compression techniques to reduce data storage requirements is one answer to this issue. The use of principal component analysis (PCA) or other dimensionality reduction techniques, for instance, can drastically reduce the storage requirements of the data while keeping key features for machine learning applications. Using data streaming techniques, which enable real-time analysis of data without the need to store massive datasets, is another option. In addition, the utilization of distributed computing and cloud-based resources can aid in resolving the data storage issue. By distributing data across numerous nodes or servers, storage requirements can be lowered while high-performance processing is still possible [11]. Overall, the challenge of data storage related with machine learning algorithms for the detection of the Higgs boson can be solved by combining data compression techniques, data streaming, and distributed computing. These technologies can aid in reducing the expense and complication of managing massive datasets, while still enabling high-performance machine learning.

In conclusion, machine learning techniques have been extensively utilized for the discovery of the Higgs boson in LHC data. This study demonstrates that using accelerated computing frameworks, and ML algorithms such as random forests, SVM, and gradient boosted decision trees can achieve good performance in identifying Higgs boson events from background events.

# Considerations

Accelerated computing frameworks can provide a better way to identify Higgs boson particles due to its capacity to expedite the processing time of machine learning algorithms and manage massive datasets. This is important for the identification of the Higgs boson particle, which creates a massive amount of data that requires substantial storage capacity and processing power.

Accelerated machine learning is the use of specialised hardware such as graphics processing units (GPUs) and field-programmable gate arrays (FPGAs) to expedite the training and inference of machine learning models. Parallel computing, which allows the concurrent execution of various tasks over several processors or cores, is an essential component of accelerated computing frameworks. Each CPU may handle a piece of the data in parallel, which can substantially accelerate the processing of big datasets. Alternatively, distributed systems can be used to accelerate machine learning further by distributing the training or inference workload among numerous machines [19]. This approach may be applied via distributed systems, in which numerous computers collaborate to perform data processing tasks. The Apache Spark framework, for instance, offers a distributed system method to manage large-scale data processing jobs, which could be valuable for the investigation of Higgs boson data.



**Figure – 1** Distributed vs Parallel Computing [20]

Figure 1 shows that parallel computing is done by connecting multiple processors or cores to a shared memory architecture. Distributed computing, on the other hand, is done by connecting multiple machines to a network, each of which has its own memory and processing power. Parallel computing is used for tasks that require a lot of work to be done on a single problem, while distributed computing is used for tasks that need to work with large datasets or process many smaller tasks at the same time. The main difference between the two is how the work is split up and how the resources are shared.

Another important part of accelerated computing frameworks is the use of specialised hardware, like graphics processing units (GPUs) and central processing units (CPUs), which can do calculations in parallel and speed up processing time by a lot. For instance, the RAPIDS suite of libraries, which includes the cuML (library for machine learning algorithms) and cuDF (library for data analysis) libraries, is optimised for GPUs and can help machine learning tasks run much faster. This can be very helpful when looking at data about the Higgs boson, where machine learning methods are often used.

Researchers at CERN found that using GPU-accelerated computing frameworks like CUDA (general-purpose computing on NVIDIA GPUs) and OpenCL can speed up the processing of machine learning algorithms by up to 20 times compared to traditional CPU-based systems [21]. Another study by [22] demonstrates that the distributed computing framework Dask can be used to speed up the analysis of large datasets in particle physics experiments by spreading them out across multiple computing nodes. This was done by finding Higgs boson particles. In a similar way, using RAPIDS, a set of GPU-accelerated data science libraries, has been shown to improve machine learning tasks in many fields, including particle physics, by a large amount. These studies show that accelerated computing frameworks may be able to help find Higgs boson particles faster and more efficiently.

Lastly, accelerated computing frameworks can include tools for managing and pre-processing data, which can help reduce the amount of space needed to store large datasets and make data processing tasks run more quickly. For example, Dask is a framework for distributed computing that has tools for processing large datasets in parallel. This can help reduce the amount of space needed to store Higgs boson data.

Overall, accelerated computing frameworks are a better way to find the Higgs boson particle because they can handle large amounts of data, do complex calculations faster, and include tools for managing data and pre-processing it. Parallel computing, distributed systems, specialised hardware like GPUs, and tools like Dask and RAPIDS can all help make the analysis of Higgs boson data faster and more accurate.

# Data Description

The HIGGS dataset is a simulated set of data from collisions of particles at the Large Hadron Collider (LHC). It is made up of 11 million events, with 28 features and a binary target variable for each one ( the class label ,1 for signal process, 0 for background process). The first feature is the transverse momentum of the leading lepton, which is shown by the lepton pT. The second feature is the lepton eta, which shows how fast the leading lepton is moving. The third feature is the angle of the leading lepton, which is shown by the lepton phi. The fourth feature is the size of the missing transverse momentum, which is shown by the size of the missing energy. The fifth feature is the missing energy phi, which shows the angle of the missing transverse momentum along the direction of motion.

The next 16 features explain what the four top jets in the event are like. The leading jet's transverse momentum, pseudo rapidity, and azimuthal angle are shown by the jet 1 pt, jet 1 eta, and jet 1 phi features. The jet 1 b-tag property is the leading jet's b-tag value, which is a measure of how likely it is that the jet came from the decay of a b quark. The sub leading jet's jet 2 pt, jet 2 eta, jet 2 phi, and jet 2 b-tag all have the same properties. The jet 3 pt, jet 3 eta, jet 3 phi, and jet 3 b-tag features represent the same properties for the third jet. The fourth jet has the same properties for the jet 4 pt, jet 4 eta, jet 4 phi, and jet 4 b-tag.

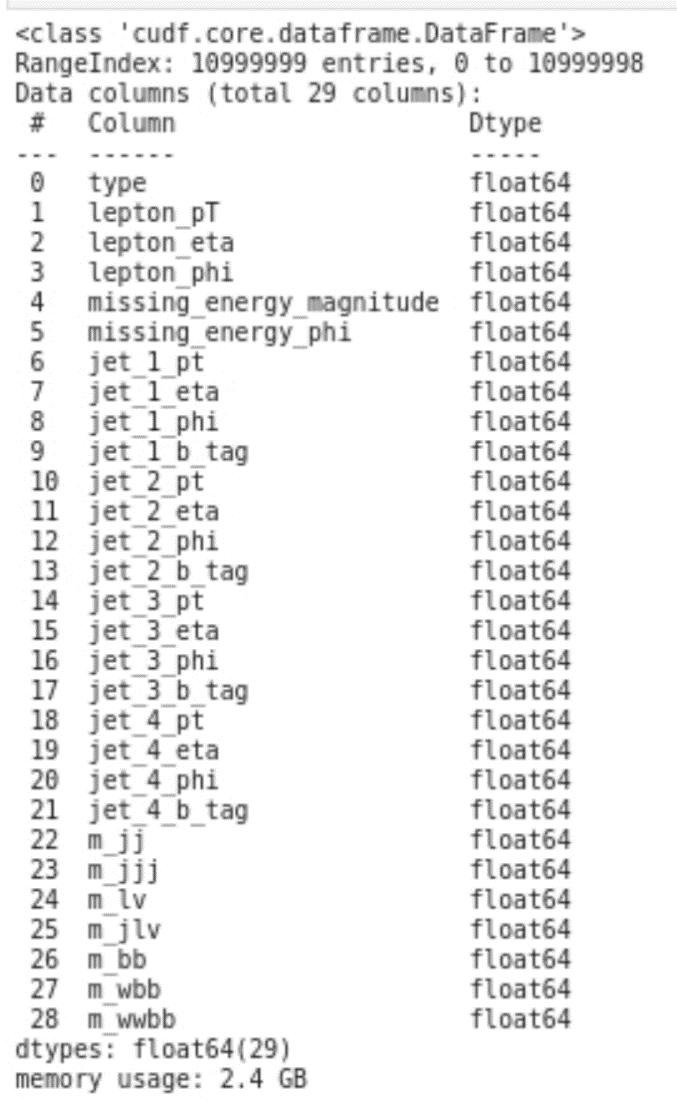
The last seven features talk about the masses that don't change for different combinations of particles in the event. The m\_jj feature shows the mass of the two leading jets that doesn't change. The m\_jjj feature shows the mass of the three leading jets that doesn't change. The m\_lv property stands for the lepton's constant mass and the transverse momentum that is missing. The m\_jlv feature shows the constant mass of the leading jet and the lepton, as well as the transverse momentum that is missing. The m\_bb feature shows the constant mass of the two jets whose b-tag value is the highest. The m\_wbb feature shows the constant mass of the system made up of the lepton, the missing transverse momentum, and the two jets with the highest b-tag value. The m\_wwbb feature shows the system's constant mass, which is made up of all the visible particles in the event.

Lastly, the target variable is a binary variable that shows whether the event produced the Higgs boson or not. If the value is 1, the event produced the Higgs boson. If the value is 0, the event did not produce the Higgs boson. Overall, the Higgs dataset has a lot of data that can be used to train machine learning models for analysing and exploring particle physics.

## Exploratory Data Analysis

EDA (Exploratory Data Analysis) is the process of understanding datasets to analyze their main features. This is often done with the help of statistical and visualization techniques. EDA is used to find patterns, trends, outliers, relationships, and insights in the data. It is also used to find potential problems and areas that need more analysis. Python is a popular language for EDA because it is easy to use and has many libraries for analyzing and displaying data. In this paper, Python will be used to check what the Higgs dataset could tell us. Let’s check the insights by importing the dataset in python.

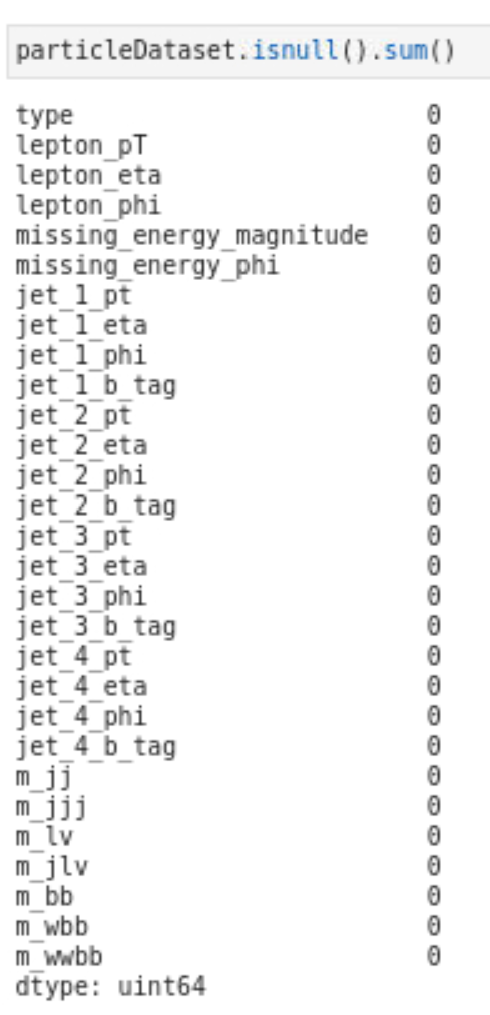
**Dataset Information**



**Figure – 2** Dataset columns and datatypes list

In Figure 3, there is a list of columns of the dataset, with their data types. All the 29 columns are numeric having data type float. The first column of the dataset is a target variable. It can be seen from the figure that there are total number of 10999999 observations or entries in the dataset.

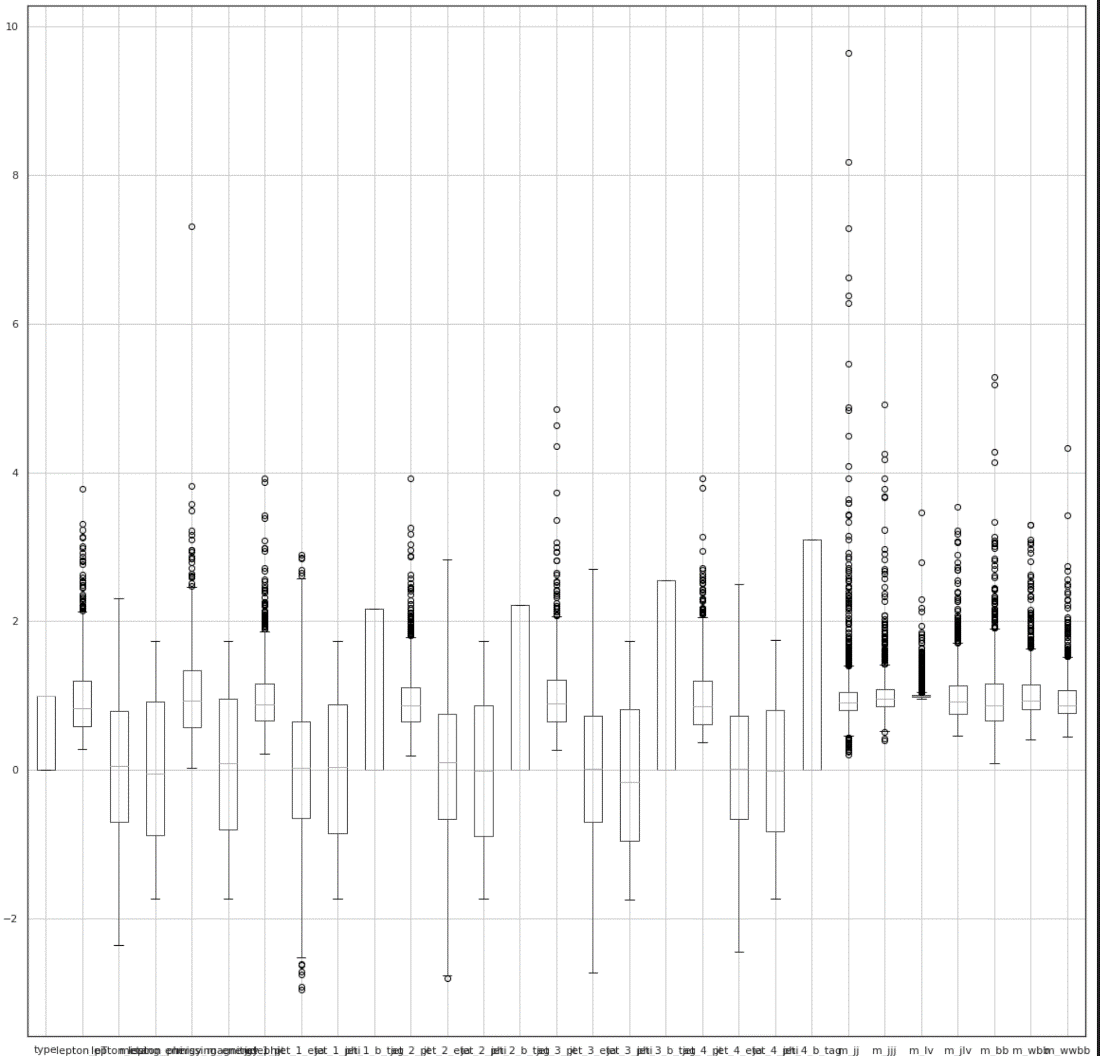
**Missing Values**



**Figure – 3** Missing values in the dataset

Figure 3 shows that there is no missing value or missing cell in the datasets, means there is no need to perform any missing values handling techniques.

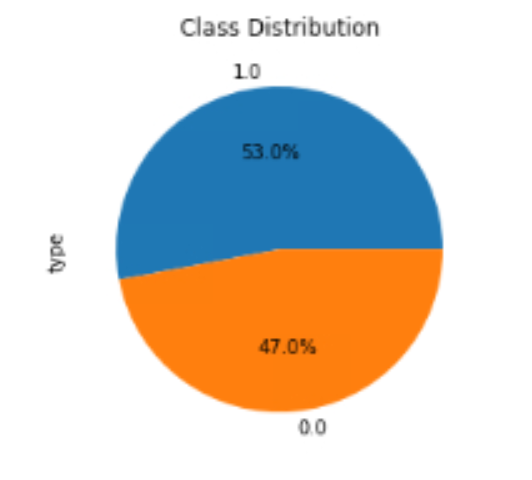
**Outliers**



**Figure – 4** Outliers in the dataset

There are many ways to identify outliers in dataset, through mathematical functions and from visualization. Visualization can be done by plotting box plot, scatter plots and histogram. In the above figure 4, boxplot has been used to show the outliers in the dataset. When analyzing a box plot, data points outside the box plot's whiskers are referred to as an outlier. From the above boxplots, "\*" these data points are outliers. In the next part “Data pre-processing”, will discuss the techniques to remove outliers.

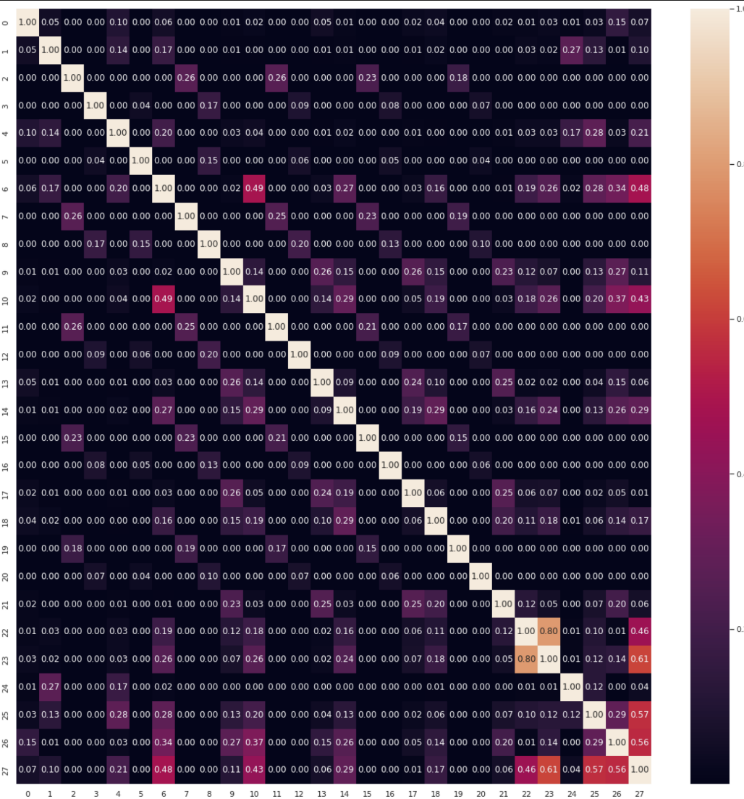
**Class Imbalance**



**Figure – 5** Imbalance classes

Fig 6 represents the classes of target variable “type” in the dataset. The above portion of the pie chart shows the percentage of signal process (which produces higgs boson particle) data which is 53% from overall dataset, and the portion below shows the percentage of background process data, 47%, from overall dataset. These classes are not balanced because the number of observations of each class is different from each other. Class balance is significant because it facilitates model training by preventing the model from becoming biased towards one class.

**Correlation**



**Figure – 6** Correlation between variables or features

Correlation is a statistical measure that describes the degree of association between two variables. When two features have a high correlation, they provide redundant information to the model, which can lead to overfitting, reduced model performance, and increased computation time. Figure 6 shows the correlation between all the variables. The correlation between variables ranges from 0.00– 0.80. The Higgs boson particle dataset has 28 features, and dropping highly correlated features can help to reduce overfitting and improve the accuracy of machine learning models.

**Data Skewness**

Table

Description automatically generated

**Figure – 7** Data skewness

Figure 7 demonstrates that, in terms of skewness we have negative, positive and zero values. Negative values indicates that the data is skewed to the left whereas positive values indicates that the data is skewed to right. For a perfectly symmetrical distribution skewness value should be zero. Handling the data skewness is a good practice for the machine learning problems. However, some algorithms such as decision trees (e.g., Gradient boosted decision tree) and Random Forest etc. are less sensitive to skewness and not require any handling data skewness technique.

# Methodology

## Data Pre-processing

Data pre-processing is an important step in the machine learning pipeline. It involves changing raw data into a format that machine learning algorithms can use to learn from. The main goal of data pre-processing is to clean, normalise, and transform the data to get rid of any errors, missing values, or inconsistencies and make it easier for machine learning algorithms to use.

**Handling Missing Values**

When there are missing values in a dataset, they can cause bias and affect how well the model works. For building a model, the loss of values may contain important data or insights. So, there are a few ways to deal with the missing values, including list-wise deletion, simple imputation, and regression imputation. For this set of data, we handle missing values with simple imputation.

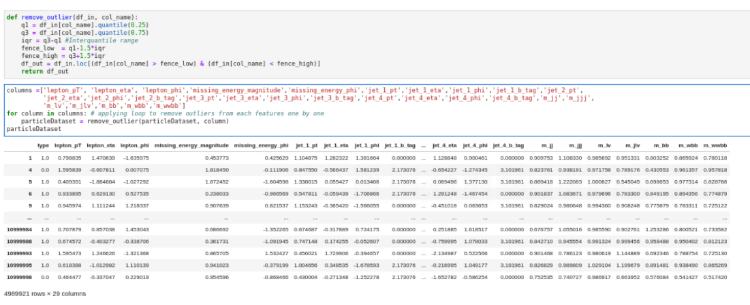
From figure 3, there is no missing value in this dataset, so nothing required.

**Drop features having high correlation**

Dropping features that have high correlation is a common technique in data pre-processing for machine learning. A commonly used threshold is a correlation coefficient of 0.9 or higher. From figure 6, correlation between the features can be seen. The highest correlation value in higgs boson dataset is 0.8, lesser than 0.9, meaning none of the features will be dropped. Dropping too many features or important features can lead to underfitting and reduced model performance. It's always a good idea to experiment with different correlation thresholds and evaluate the impact on model performance before making a final decision. During the implementation part, experiments will be done on dropping features to evaluate the impact on model performance.

**Handling Outliers**

There are different ways to deal with outliers in a dataset, such as deleting observations, changing values, imputation, and treating each one separately. It is important to figure out what the outliers are and whether they are clearly wrong or don't have anything to do with the problem. If the outliers are obviously wrong or not important, it may be best to take them out of the data set or change the values to make them less important. But if the outliers are important and tell us something important about the problem, it may not be right to change or get rid of them.



**Figure – 8** Remaining observations after removing outliers

Figure 8 shows that after removing outliers from the dataset almost 5 million records left from 11 million records. This means there are lots of outliers in the dataset. In the future work, autoencoder and PCA (Principal Component Analysis), techniques will be used to handle outliers. The impact of each technique will be analyzed on machine learning models to determine the overall performance of the model. After that decision will be made which technique is suitable for the handling of outliers for this type of dataset.

**Class Balancing**

As discussed before in figure 5, there is an imbalance of all the classes contained in dataset. Class imbalance results in models that have poor predictive performance. Class balancing can be handled using resample the training set. There are two methods of re-sampling, under-sampling and over-sampling.

The Python libraries cuDF and cuML provide support for both under-sampling and over-sampling techniques, which can be used to balance the class distribution of a dataset. As we can see from figure 5, there is a 6% difference between the classes and, it’s a huge dataset so an under-sampling technique has been selected for the future work. In cuML, I will use the cuML.preprocessing.RandomUnderSampler class to apply random under-sampling to this cuDF DataFrame.

## Libraries and APIs

For the identification of the Higgs boson particle as a classification problem, you can utilise a range of GPU-accelerated machine learning packages and APIs. Common accelerated machine learning libraries and application programming interfaces (APIs) for classification problems include the following.

**Dask:** Dask is a Python toolkit for parallel computing that enables CPU and GPU parallelism. It offers APIs for parallel DataFrames, arrays, and machine learning methods that can be utilised to rapidly pre-process big datasets.

**Rapids:** Rapids is a suite of open-source data science and analytics libraries built on NVIDIA's CUDA platform. Among others, it contains libraries for data preparation (cuDF), machine learning (cuML), graph analytics (cuGraph), and visualisation (cuXfilter).

**cuDF:** cuDF is a GPU-accelerated DataFrame toolkit that offers an API similar to Pandas for processing huge datasets on GPUs. It provides multiple dataframe operations, such as filtering, merging, and joining, which might be beneficial for preprocessing huge datasets. As previously discussed, I have utilised this package for EDA (Exploratory Data Analysis) of the Higgs boson dataset.

**cuML:** cuML is a GPU-accelerated Python library for machine learning built on the CUDA platform. It offers a number of algorithms for classification, regression, clustering, and other tasks, and it supports evaluation metrics such as precision, recall, F1 score, and ROC-AUC.

**Scikit-learn:** Scikit-learn is a popular machine learning package that contains a variety of classification, regression, and clustering techniques, among others. In addition, it offers numerous evaluation metrics for classification, regression, and clustering tasks, including as accuracy, precision, recall, F1 score, mean squared error, and R-squared, among others.

For **data visualization** in accelerated machine learning, the following libraries has been used:

**Matplotlib:** Matplotlib is a widely used Python toolkit for creating static, interactive, and animated visualisations. It can be used to create, among other visualisations, line charts, scatter plots, histograms, and heatmaps.

**Seaborn:** Based on Matplotlib, Seaborn is a Python library for visualising data. It has a high-level interface for making statistical graphics. It works with different kinds of plots, like heatmaps, scatterplots, line plots, and bar plots.

These libraries can help speed up the machine learning process for classifying Higgs boson particles. This means that models can be trained faster and with more accuracy.

## Model Selection

When using accelerated machine learning to solve the Higgs boson classification problem, there are several factors to think about, such as the size and complexity of the dataset, the hardware resources that are available, and the needs of the machine learning algorithms to solve the kind of the problem. Here are some of the machine learning algorithms I will be using to solve this classification problem.

**Random Forest**: Random Forest is an ensemble learning method that works by building a lot of decision trees at training time and outputting the class that is the average of the classes of the individual trees. It is a popular algorithm for solving classification problems, and it can work well with large datasets. The Random Forest algorithm can be used with Scikit-learn and cuML.

**Gradient Boosting:** Gradient Boosting is an ensemble method that improves the accuracy of predictions by using a series of decision trees. It is a popular algorithm for solving classification problems, and it can work well with large datasets. Gradient Boosting is a popular algorithm, and XGBoost and LightGBM are two popular ways to use it.

**Support Vector Machines (SVMs):** SVMs are a common way to solve classification problems, and they can work well with large datasets. Scikit-learn and cuML both have SVM implementations that can be sped up by a GPU.

**Naive Bayes:** The Naive Bayes algorithm is a simple way to solve classification problems that works well with large datasets. Scikit-learn has several implementations of the Naive Bayes algorithm that can be sped up by both the CPU and the GPU.

Here is a brief **justification** for each of the selected algorithms:

**Random Forest:** Random Forest is often used to solve classification problems because it is easy to use, can handle large datasets with many dimensions, and doesn't tend to overfit. It also lets you rank the importance of features, which can help you understand how the data is structured. The Random Forest algorithm is known to work well on many different types of datasets, including the Higgs boson dataset.

**Gradient Boosting:** Gradient Boosting is a powerful algorithm that improves the accuracy of predictions by adding decision trees to an ensemble repeatedly. It is a good choice for classification problems because it can handle complex interactions between features and handle missing data well. Gradient boosting is a popular technique, and XGBoost and LightGBM are two popular implementations of it. They are known for being fast and scalable, which makes them a good choice for large datasets.

**Support Vector Machines (SVMs):** SVMs are often used to solve classification problems because they can handle both linear and nonlinear relationships between features and work well with high-dimensional datasets. They are known for being able to handle large datasets quickly and are often used to solve text classification, image classification, and other types of classification problems.

**Naive Bayes:** The Naive Bayes algorithm is a simple way to solve classification problems that works well on large datasets. It works quickly, can grow, and can handle many kinds of data. It works well in practice even though it assumes that all features are independent, which may not be true in many real-world datasets.

## Model Hyperparameters

How to choose the best hyperparameters for each of the above models will depends. Here are some general rules to follow when choosing the hyperparameters for each model:

**Random Forest:** When training a Random Forest model, the main hyperparameters to think about are the number of trees in the forest (n estimators), the maximum depth of each tree (max depth), and the minimum number of samples required to split an internal node (min samples split). The size and complexity of the dataset will determine the best values for these hyperparameters. It may be necessary to try out different values for these hyperparameters and use cross-validation to find the best settings.

**Gradient Boosting:** When training a Gradient Boosting model, the most important hyperparameters to think about are the number of trees in the sequence (n estimators), the rate of learning (learning rate), and the maximum depth of each tree (max depth). The minimum number of samples required to split an internal node (min samples split) and the minimum number of samples required to be at a leaf node (min samples leaf) are also important hyperparameters. The size and complexity of the dataset will determine the best values for these hyperparameters.

**Support Vector Machines:** When training an SVM model, the main hyperparameters to think about are the type of kernel function (such as linear, polynomial, or radial basis function), the regularization parameter (C), and, for non-linear kernels, the kernel coefficient (gamma). The size and complexity of the dataset will determine the best values for these hyperparameters.

**Naive Bayes:** The Naive Bayes algorithm is easy to understand and has few hyperparameters that need to be changed. The most important hyperparameters to think about are the type of Naive Bayes variant being used (such as Gaussian, Bernoulli, or Multinomial) and any smoothing parameters that may be used to change the probabilities. Most of the time, the best values for these hyperparameters can be found through analysis or simple experimentation.

The hyperparameters need to be tuned because if the best values are not found, it can affect how well the model works. There are many ways to tune the hyperparameters, such as GridSearch and RandomSearch. GridSearch searches the hyperparameters by trying all of the possible combinations of values for the hyperparameters, while RandomSearch picks values at random for a certain number of iterations. Consequently, I'll be using RandomSearch to tune the hyperparameters instead of GridSearch because GridSearch can be hard to run on large datasets and it’s computationally intensive.

## Model Evaluation

I will use the following evaluation metrics (accuracy, precision, recall, F1-score, and area under the ROC curve) to evaluate how well the chosen models do at classifying Higgs boson particle. Furthermore, cross-validation will also be used to estimate the generalization performance of the model.

Here is a short summary of how to evaluate the chosen models:

**Random Forest:** I will evaluate the Random Forest model's performance utilizing metrics such as accuracy, precision, recall, F1-score, and AUC-ROC.

**Gradient Boosting:** I will evaluate the Gradient boosted decision tree model's performance utilizing metrics such as accuracy, precision, recall, F1-score, and AUC-ROC.

**Support Vector Machines (SVMs):** I will evaluate the SVM model's performance utilizing metrics such as accuracy, precision, recall, F1-score, and AUC-ROC.

**Naive Bayes:** I will evaluate the Naïve Bayes model's performance utilizing metrics such as accuracy, precision, recall, F1-score, and AUC-ROC.

# Conclusion

In the field of particle physics, machine learning methods and accelerated computing frameworks can be utilized to find the Higgs boson particle. The paper explores the challenges of finding the particle using conventional methods and how machine learning might help overcome these obstacles. The document also examines the significance of pre-processing data, how to handle missing values and outliers, and the impact of correlation and class imbalance on the performance of machine learning models. The research proposes machine learning techniques that could be used to tackle the Higgs boson particle classification problem and offers hyperparameters that can be utilized to enhance the overall performance of machine learning models. It also gives evaluation measures for testing the performance of machine learning models. In summary, the report demonstrates how accelerated frameworks for machine learning and faster computation could benefit particle physics research.

##### References

[1] Libretexts, “Physics Boundless.” Accessed: Feb. 25, 2023. [Online]. Available: https://phys.libretexts.org/Bookshelves/University\_Physics/Book%3A\_Physics\_(Boundless)/2%3A\_Kinematics/2.1%3A\_Basics\_of\_Kinematics

[2] M. Walter and A. W. Wolfendale, “Early history of cosmic particle physics,” *European Physical Journal H*, vol. 37, no. 3. pp. 323–358, 2012. doi: 10.1140/epjh/e2012-30020-1.

[3] The ATLAS Collaboration, “Observation of a new particle in the search for the Standard Model Higgs boson with the ATLAS detector at the LHC,” Jul. 2012, doi: 10.1016/j.physletb.2012.08.020.

[4] S. P. Martin, “A Supersymmetry Primer,” 2016.

[5] S. Chatrchyan *et al.*, “Observation of a new boson at a mass of 125 GeV with the CMS experiment at the LHC,” *Physics Letters, Section B: Nuclear, Elementary Particle and High-Energy Physics*, vol. 716, no. 1, pp. 30–61, Sep. 2012, doi: 10.1016/j.physletb.2012.08.021.

[6] D. Noel, “Machine learning based searches for new physics at the ATLAS experiment,” 2022.

[7] “Triggering in ATLAS in Run 2 and Run 3 Takuya Nobe \* on behalf of the ATLAS collaboration,” 2021. [Online]. Available: https://pos.sissa.it/

[8] “CLASSE DI SCIENZE Corso di Perfezionamento in Fisica.”

[9] M. Abbas, A. Khan, A. Saeed Qureshi, and M. W. Khan, “Extracting Signals of Higgs Boson From Background Noise Using Deep Neural Networks.”

[10] G. Louppe, M. Kagan, and K. Cranmer, “Learning to Pivot with Adversarial Networks,” Nov. 2016, [Online]. Available: http://arxiv.org/abs/1611.01046

[11] Innovation News Network, “17\_The Big Data challenge at the Large Hadron Collider”, Accessed: Mar. 06, 2023. [Online]. Available: https://www.innovationnewsnetwork.com/big-data-challenge-large-hadron-collider/11359/

[12] M. Azhari, A. Abarda, B. Ettaki, J. Zerouaoui, and M. Dakkon, “Higgs Boson Discovery using Machine Learning Methods with Pyspark,” in *Procedia Computer Science*, 2020, vol. 170, pp. 1141–1146. doi: 10.1016/j.procs.2020.03.053.

[13] “A Comparison of Machine Learning Algorithms in Identifying Higgs Boson Events From the Background.” [Online]. Available: www.summersteminstitute.org.

[14] A. N. Charan, “Particle identification with the Belle II calorimeter using machine learning,” Jan. 2023, doi: 10.1088/1742-6596/2438/1/012111.

[15] X. C. Vidal, L. D. Maroñas, and Á. D. Suárez, “How to use Machine Learning to improve the discrimination between signal and background at particle colliders,” Oct. 2021, doi: 10.3390/app112211076.

[16] Terrones & Terrones, “19\_Harold Walter Kroto\_ Physics Today\_ Vol 69, No 9”, Accessed: Mar. 06, 2023. [Online]. Available: https://physicstoday.scitation.org/doi/10.1063/PT.3.3306

[17] “The ATLAS Experiment at the CERN Large Hadron Collider.” [Online]. Available: http://www.iop.org/EJ/jinst/

[18] B. J. Presperín, “Machine learning for t ¯ tH mechanism Higgs boson detection from CERN ATLAS data.”

[19] M. Wang, W. Fu, X. He, S. Hao, and X. Wu, “A Survey on Large-scale Machine Learning,” Aug. 2020, [Online]. Available: http://arxiv.org/abs/2008.03911

[20] Pattamsetti, “22\_Parallel versus distributed computing \_ Distributed Computing in Java 9”, Accessed: Mar. 06, 2023. [Online]. Available: https://learning.oreilly.com/library/view/distributed-computing-in/9781787126992/7478b64c-8de4-4db3-b473-66e1d1fcba77.xhtml

[21] S. C. Tognini *et al.*, “CompF2-Beam and Detector Simulation Celeritas: GPU-accelerated particle transport for detector simulation in High Energy Physics experiments \*,” 2022. [Online]. Available: http://energy.gov/downloads/doe-public-access-plan

[22] N. Eich, M. Erdmann, P. Fackeldey, B. Fischer, D. Noll, and Y. Rath, “Fast Columnar Physics Analyses of Terabyte-Scale LHC Data on a Cache-Aware Dask Cluster,” Jul. 2022, [Online]. Available: http://arxiv.org/abs/2207.08598