Accelerated Machine Learning for Higgs Boson Classification Using NVIDIA RAPIDS and DASK

***Abstract*— High-energy physics experiments produce immense volumes of data and require efficient manipulation for the revelation of scientifically useful insights. Thus, this study proposes a machine-learning approach to classifying Higgs boson events based on salient kinematic features from particle collision events at the Large Hadron Collider (LHC). Preprocessing steps include normalization, skewness correction, outlier treatment, and Principal Component Analysis (PCA) for dimensionality reduction. Random Forest (RF), XGBoost, and Support Vector Machines (SVM) are being examined for possible implementations. This study also lays emphasis upon the advantages rendered by GPUaccelerated processing using NVIDIA RAPIDS and DASK to achieve leap time reductions in the computations. The aim of the study is to create an efficient and scalable machinelearning pipeline for Higgs boson classification using accelerated compute engines.**

***Keywords—M*achine learning, GPU Acceleration, RAPIDS, Higgs Boson, Principal Component Analysis (PCA).**

I. Introduction

2012 witnessed the achievement of probably the greatest milestone in particle physics: the discovery of the Higgs boson inside the Large Hadron Collider (LHC), which validated the mechanism for mass acquisition by elementary particles as anticipated in the Standard Model. Identification/classification of Higgs boson events within the vast amounts of data being generated from high energy collision experiments pose awesome challenges with regards to computation, due to the high dimensionality of the data, weak correlations between features as well as the great consumption of resources for computation.

Most researchers are using machine learning (ML) tools to confront some of these challenges in their works. For instance, Alves (2016) [1] researched the use of stacked generalization, which is an ensemble ML technique, as a technique in the identification of Higgs bosons at the LHC. The study made use of deep neural networks (DNNs) as well as boosted decision trees for performance comparison against stacking, and the results showed among other findings that while stacks performed marginally worse than DNNs, it took significantly less computational effort and outperformed boosted decision trees in certain scenarios.

Brain-Inspired Bayesian Confidence Propagation Neural Networks (BCPNN) were researched by Svedin et al. (2021) [2] for classification of Higgs boson samples.

Their experimental setup, termed StreamBrain, was geared towards the Higgs Boson high-energy physics dataset for background/signal classification, attaining a performance score of up to 69.15% accuracy and 76.4% Area Under the Curve (AUC), exemplifying the kind of potential a different learning rule brings to ML applications for particle physics.

High Performance Machine Learning Classifiers: Deep Neural Networks, Boosted Trees, Random Forest, and Support Vector Machines were tested by Lasocha et al. (2018) [3] to measure the CP state of the Higgs boson in the H to tau tau channel. This research stresses the value of putting constraints on non-measurable outgoing neutrinos if one was to improve on the classification sensitivity. All these pieces of research highlight the efforts directed towards improving the classifying of events pertaining to the Higgs boson through innovative ML methods. Based on this research effort, our research gives an advanced yet interpretable machine learning pipeline enabled with GPU acceleration. The pipeline will be addressing data preprocessing, feature engineering, and model selection with the help of frameworks like NVIDIA RAPIDS and DASK for supersonic data handling and training acceleration.

This paper is structured as following : the paper reviews relevant previous research in Section 2 regarding particle classification and GPU-accelerated computing. Section 3 then portrays the different computational challenges that occur in high-energy physics data processing. Section 4

presents the dataset and related techniques used for exploratory data analysis (EDA) and preprocessing. Section 5 describes the machine learning methodology from model selection, hyperparameter tuning to evaluation metrics. and finally, Section 6 gives the results, challenges faced in the study, and future direction**s**.

II. Background

High-energy physics experiments, such as those conducted at the Large Hadron Collider (LHC), typically generate a huge amount of data, which causes significant problems regarding their storage, processing, and analysis. For instance, these problems must be resolved for a further understanding of elementary particles like the Higgs boson. This section describes previous work on the classification of the Higgs boson and also discusses its challenges in areas such as data storage and machine learning application.

A. Previous Work

1) Storage Solutions:

Data storage solutions for the increasing amount of data generated in particle physics undergo intense scrutiny. Traditional storage systems could not keep pace with the increasing data volumes.

Bashyal et al. (2022) [4] evaluated the use of HDF5 as a possible storage technology for high-energy physics experiments regarding aspects such as scalability and efficiency of processing large data amounts in highperformance computing environments.

Data management for lattice QCD calculations also covers aspects such as the amount of data involved in such computations. McNeile (2000) [5] addressed data management issues that turn research strategies into robust data handling strategies.

2) Machine Learning Applications in Higgs Boson Classification:

Machine learning techniques have been broadly applicable in the classification of events in particle physics. Modification of classifiers for Higgs boson identification has been experimentally evaluated by Nelakurti and Hill (2024) [6] to characterize detection accuracies and classification issues.

Adam-Bourdarios et al. (2015) [7] put forward models for gradient boosting (GBM) for event selection and indicated that it has certain hurdles towards selection of features and interpretation of models. The work highlighted on how to find associations in weak features and on dimensionality in the Higgs boson classification tasks.

In addition, Roy (2022) [8] focused on the FAIR data and AI models in high-energy physics research, emphasizing accessibility and reproducibility of data as two points being significant in machine learning applications.

B. Challanges

1) Data Storage and Management:

The massive volumes of data that HEP experiments generate pose serious challenges in terms of storage. Often, traditional storage systems are unable to match the requirements of scalability and efficiency, creating bottlenecks in the data retrieval and analysis processes. Scalable storage solutions like HDF5 are being looked into for addressing such challenges, while others remain, such as ensuring compatibility with existing data processing frameworks and maintaining data integrity [4], [9].

2) High-dimensional Data and Feature Correlation:

Higgs boson event classification involves the analysis of high-dimensional datasets characterized by a large number of features. Difficulties arise in the identification of relevant features and the management of weak correlations among various features, which complicate the development of effective classification models. Dimensionality-reducing techniques, such as Principal Component Analysis (PCA), are employed. however, determining how many components should be chosen and interpreting those may prove quite challenging.

3) Processing Speed and Computational Resources:

The computational demand for processing and analyzing large-scale HEP data is beyond the capacity of classic CPU-based systems. While GPU-accelerated computing frameworks like NVIDIA RAPIDS and DASK present possibilities for alleviation by allowing parallel processing, imparting these technologies into existing workflows and tuning their performance on specific tasks entails a lot of work and expertise.

Addressing these challenges present crucial steps toward the advancement of Higgs boson classification and high-energy physics large-scale complex datasets in general.

III. Consideration

Usually, data science & machine learning activities require computations that go beyond the limits of traditional CPU. For large-scale data processing and complex model training - these are the hallmarks of particle discovery, healthcare analytics, and, of course, many other fields. These computing activities can be accelerated with frameworks designed mainly for CPUs, especially those utilizing GPUs. Below are factors considered when using such frameworks with specific attention to how they improve performance in parallel computing, distributed systems, and so on.

A. Parallel Computing

When multiple tasks are accomplished at once concurrently, that is what one means by parallel computing. This involves significantly saving computation time. This is the kind for computing that can be used for large datasets or very expensive participant operations-both of which fall into the categories of machine learning and simulations in particle physics.

Classic serial computer runs processes one at a time, which proves to be inefficient for operations dealing with data. In summary, however, parallel computer divides up tasks into smaller sub-tasks, some of which can be done concurrently. Hence, parallel computing speeds things up in terms of data processing and model training. GPUs, having thousands of cores, have been designed for parallel execution, thus performing computations much faster than a CPU for a parallelizable workload.

B. Distributed Systems

Distributed computing refers to many computers, usually coupled by a computer network, using their resources to jointly tackle a given problem. Indeed, the ability to go beyond the single machine limits that these distributed systems can give is counterbalanced by additional problems, such as network latency and data synchronization, among other things, as well as failure tolerance. GPU-fitted accelerated computing frameworks can be instrumental in improving accelerated efficiency within a distributed system by taking weight from heavy computations away from the system.

In this way, if GPU applied in distributed systems can perform parallel computations locally at each node, the data transmitted between nodes is reduced to minimum amount. Hence, it can enhance overall throughput while minimizing the latencies which are frequently observed in CPU-based distributed systems

C. DASK

DASK is an open-source parallel computing library that lets Python libraries (such as NumPy, pandas, and scikitlearn) function in environments larger than memory or a distributed environment. Scaling computations across many cores or machines are the forte of DASK, specifically in big data applications.

Nothing seems to bother DASK concerning data management. it just effectively divides a large data set into smaller chunks which can be processed in parallel. It is also capable of utilizing GPUs in order to scale processing, thereby being the ideal instrument for handling such complex workflows in distributed systems.

D. RAPIDS Framework

RAPIDS is intended to run data science and machine learning completely on the GPU, and it includes the whole ecosystem of tools for accelerating manipulation, applications of machine learning, and data visualization tasks. Important RAPIDS libraries include:

- cuDF: A data frame library that utilizes the GPU and is similar to pandas, for the efficient handling of large datasets.
- cuML: The machine-learning library accelerated by the GPU, with a wide variety of algorithms:

classification, regression, clustering, dimensionality reduction, and on and on.

- cuGraph: A library for GPU-accelerated graph analytics.

RAPIDS uses the GPU for speedup of data processing tasks ordinarily handled by the CPU-bound libraries. Hence, it would allow much faster data manipulation, model training, and analytics. With the use of GPU-accelerated libraries such as cuDF and cuML, a data scientist can drastically minimize the time spent on exploratory data analysis, feature engineering, model development, and training

E. GPU/CPU Comparison

The goal of CPU designs is to deliver outstanding performance on tasks that involve significant singlethreaded execution. The goal of GPU architectures is to facilitate the parallel execution of thousands of threads for many thread-parallel tasks.

CPUs do not outshine GPUs in execution because of the task at hand. Although some tasks such as matrix operations and image processing may require parallelism, among the other activities that require training machine learning models, GPUs do perform better than CPUs. Since GPUs interact with jobs in parallel, they can yield high throughputs as compared to CPU processing during training deep learning models or matrix multiplications.

IV. Data Description

Written especially for the purpose of research in highenergy physics, the HIGGS Dataset is a Monte Carlo simulation dataset. The dataset classifies the events as signal (1) representing Higgs boson production or background (0).

The dataset consists of Low-Level Features directly derived from the measurements of a particle detector and High-Level Features introduced by physicists to improve the discrimination power between the classes.

1) Key Features of the Dataset

a) Low-Level Features (21 columns)

- Lepton pT, Lepton eta, Lepton phi.
- Missing energy magnitude, Missing energy phi.
- Jet 1-4 properties: pt, eta, phi, b-tag (for each jet).

b) High-Level Features (7 columns)

- Invariant mass variables: m_jj, m_jjj, m_lv, m_jlv, m_bb, m_wbb, m_wwbb.

c) Target Variable:

- 1 = Signal (Higgs boson production).
- 0 = Background.

2) Dataset Characteristics

- Size: 11,000,000 instances.
- Features: 28 numerical features + 1 target variables.
- Class Distribution: Balanced dataset (53:47) as shown in fig 1.
- Missing Data: None (dataset is complete).
- File Format: CSV.

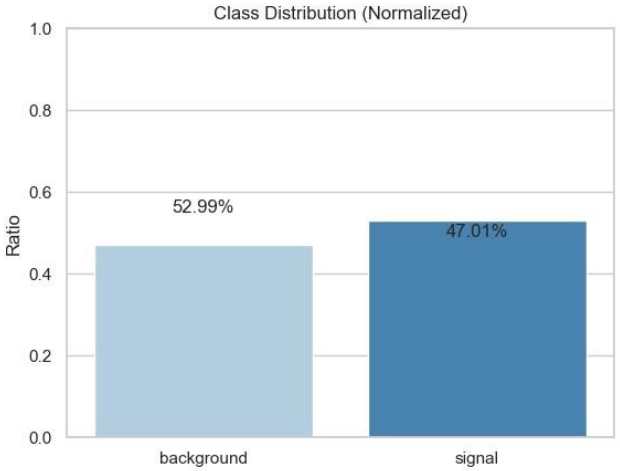

Fig. 1. Target Class Distribution

B. Exploratory Data Analysis (EDA)

In this section we perform an EDA on the dataset. This step is conducted to gain insights into the dataset, identify patterns, and inform preprocessing decisions. This section highlights the findings from visualizing feature distributions, outliers, and correlations with the target variable.

1) Data Distribution

When using histogram to display data as seen in fig 2, we observed some issues that need to be addressed before model training.

a) Left Skewness: features like lepton_pT, missing_energy_magnitude, jet_1_pT, jet_2_pT, jet_3_pT, jet_4_pT, m_jj, m_jjj, m_lv, m_jlv, m_bb, m_wbb, and m_wwbb show left skewed distribution. This skewness may negatively impact the learning process.

b) Data Values Range: Values for features in this dataset range extremely. For instance, missing_energy_magnitude and jet pT values have significantly higher values than the other features, probably leading to an unexpected result during model training. Thus, feature scaling will be performed to normalize the values across all features.

The rest of the features follow a bell-shaped distribution, which is ideal for training an ML model. but we still need to normalize the values to the [0, 1] range.

2) Data Balance

While the dataset exhibits a 53:47 class distribution (fig 1) between Higgs boson events (positive class) and background events (negative class), Such minor imbalances are unlikely to require balancing techniques. Studies indicate that under such conditions, where the difference involves slight imbalances, the classifiers do not really

suffer in performance when used in high-energy applications.

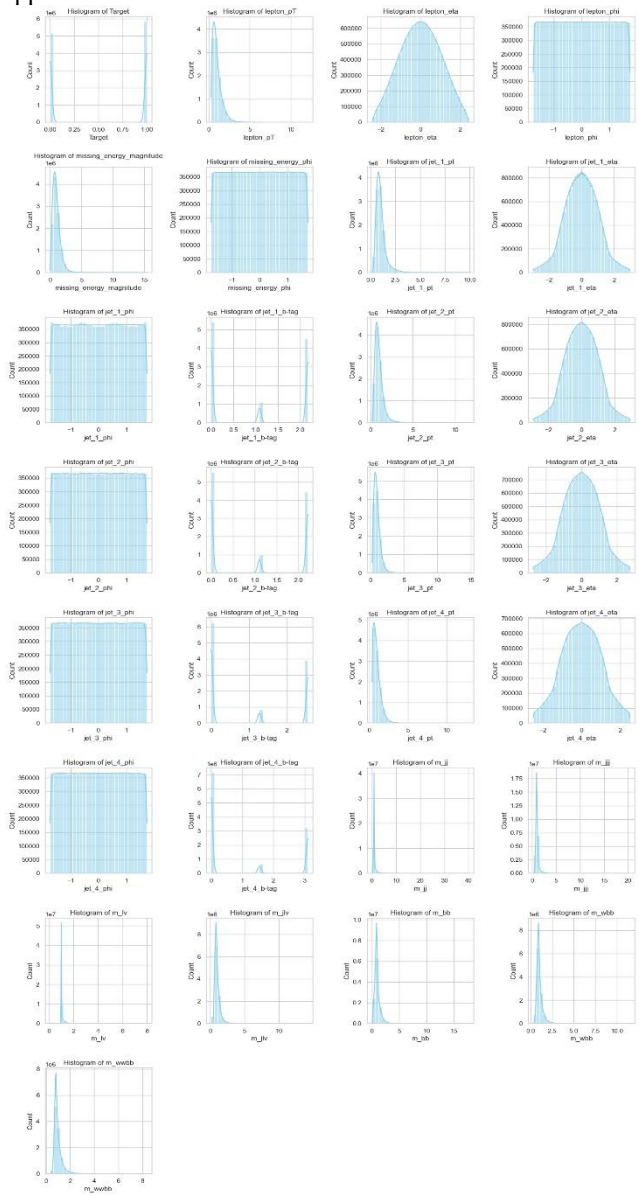

Fig. 2. Histograms of Data Distributions

Murphy (2019) [10] discussed imbalance in high-energy physics and instances where it does not strongly affect classifier performance. Similarly, in the work by Chen et al. (2021) [11] on a data set that describes Higgs boson decays, the observation is that while the existence of large imbalances is fairly widespread, the occurrence of small imbalances, such as that found in our dataset, does not require correction measures. Hence, the standard procedures for training should be adequate without the use of any balancing interventions [12].

3) Features Correlation

By examining Fig. 4, we conclude that the features in this dataset do not share high correlation between themselves or with the target. This means that the dataset bears less redundancies and hence is more informative, but the presence of such complex underlining relationships enables the application of models like Random Forests into such datasets.

4) Outliers

Extreme values can badly influence the model training as stated in here [13]. In fig 3 we can see that the dataset has many outliers across different features. However, we can expect that most outliers are due to skewness left in the dataset, so by fixing the skewing in the features, most of the outliers will be delt with. For the remaining outliers we need to use imputation to remove them.

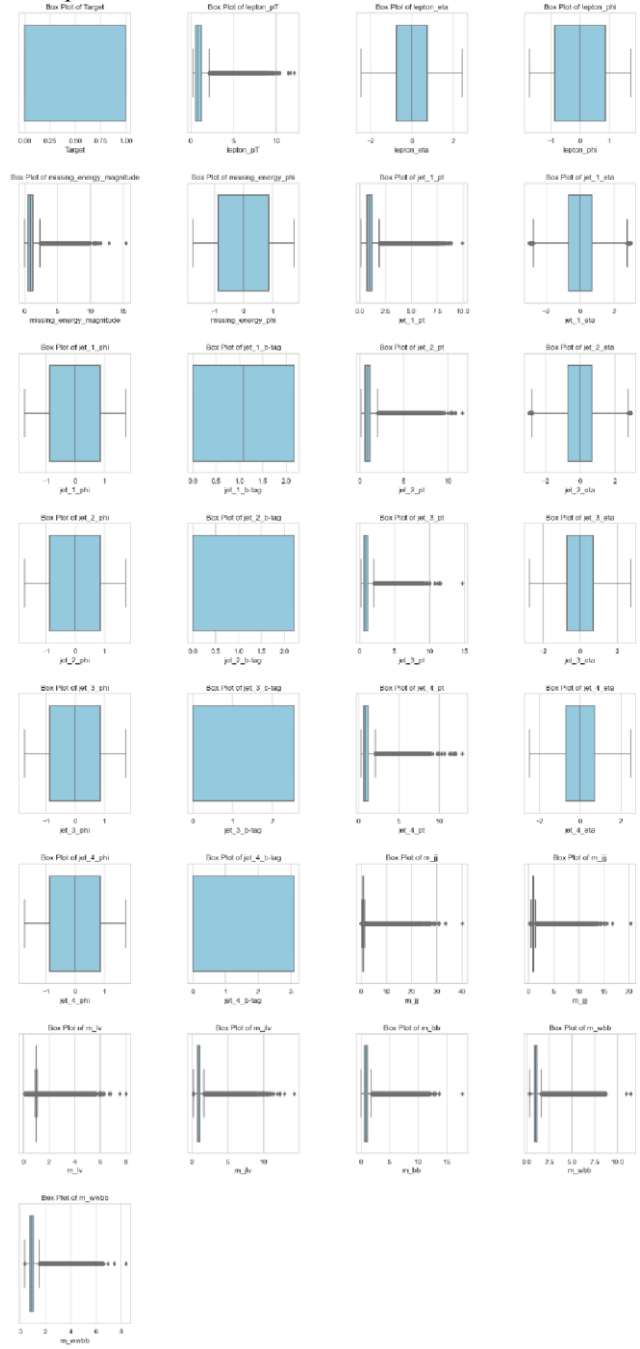

Fig. 3. Outliers Box Plot

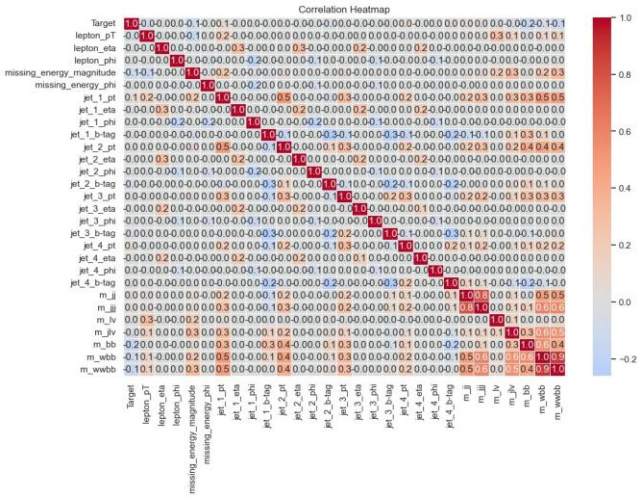

Fig. 4. Correlation Heatmap

V. METHODOLOGY

Given the challenging nature of the dataset provided, we are going to apply many preprocessing steps before selecting the model and using the dataset to train it.

In this section we discuss Data Preprocessing, Models selection and Models Evaluation Metrics and how to solve the challenges provided in the dataset.

A. Data Preprocessing

We are going to use several methods to address many issues in the dataset.

1) Skewness Adjustment: Features with significant skewness will be log-transformed to reduce skewness and make distributions more normal. We are using logtransformed for better results and fast implementation [14]

2) Feature Scaling: we will use Z score scaling to standerize the features as suggested in the research paper by Fei et al., (2021) [15].

3) Outlier Handling: after using log-transform and Z score we will remove any outliers in the dataset to reduce negative influence during training.

4) Feature engineering: Given the weak correlations observed in the feature set, we shall explore in more detail the most relevant variance features, reproducing the representation of the data in a more refined manner. This has been stated by Aslam et al.'s 2023 definition of PCA [16] as "A challenge to the evolution of dimensionality reduction, PCA converts the data into a principal component defining the maximum variance the data captures. In other words, it provides a lower dimensional representation while minimizing information loss, particularly in data-rich geometries." PCA will enhance redundancy minimization while saving the important variations, thus improving computational power use and possibly improved classification efficiency.

Moreover, emerging studies delving into the performance of principal component analyzers in such regards include Ogbuanya (2020) [17], who presented a revolutionary approach in PCA and demonstrated how it reduced dimensionality and also improved computational efficiency for many datasets. Similarly, in the same spirit, there was maximally correlated principal components with Feizi and Tse (2017) [18] which sought to realize a nonlinearity features popular in traditional PCA in terms of correlation detection, further indicating that PCA methods could be functional in complexity-laden datasets.

5) Splitting Dataset: for the final step of the data preprocessing we are going to split the data into 3 subdatas. Train which contain 80%, validation 10% and test 10%. We split data so that we can see how the model perform in an unseen data. We use the validation dataset to messur the fiting of the model. And we test the model on the test dataset to see how it performs.

B. Model Selection and Evaluation

Three models were selected for their unique advantages, in this section we will discuss the reasons why we chose those models and what are their strengths and weaknesses, and what result we can expect from them.

1) Model Selection

a) Random Forest: Random Forest is the strong model of the ensemble machine learning technology, which creates a multitude of decision trees used for predictions. It is mainly worth its advantage, as it can capture the complex interaction among features in the data and complicated nonlinear relationships, making this method flexible for such datasets with complex patterns like the Higgs dataset. Moreover, it has a strong resistance to overfitting, especially while using a relatively large number of trees. It can also glean a neat insight into the relative importance of feature variables which is important in knowing what really drives the prediction, as in particle physics data.

However, computational cost is very high, as numerous trees are involved in storing and processing memoryintensive data structures. In this sense, it has good generalization. However, with it comes the disadvantage of lesser interpretability as compared to simple models like Logistic Regression, which might restrict use in contexts needing transparency, such as healthcare or scientific research (Breiman, 2001) [19].

b) Support Vector Machine (SVM): A classifier which works in high dimensional spaces. It creates hyperplanes that maximize a margin between classes, and make it more effective when a dataset has a clear separation between classes. Simplicity and interpretability make SVM a strength: that is important in clinical or scientific settings where model understandability is key (Cortes & Vapnik, 1995) [20].

SVM only assumes that a dataset is linearly separable, which can be a limitation in existence of more complex relationship in datasets such as that in Higgs. The kernel trick allows SVM to define non-linear borders. however, it will still be unable to manage highly non-linear patterns unless the right kernel is chosen.

c) XGBoost: A popular implementation of gradient-boosted decision trees. It has become very famous because of the ability of treating very large datasets and at the same

time being very effective at capturing complex feature interactions and very complicated non-linear relationships. It works quite good in the case of the Higgs dataset, where feature interactions and non-linearity play very important roles. Moreover, it has shown great performance and combats overfitting with its regularization techniques, which is very important for high-dimensional datasets such as the ones in the current environment (Chen & Guestrin, 2016) [21].

Yet, like Random Forest, XGBoost has a high cost in terms of computation power while tuning its hyperparameters. Moreover, the interpretability of the model is less when compared to simpler models, thus making it difficult to comprehend the role of individual features.

d) Expected Result: Models like Random Forest, SVM, and XGBoost were chosen because, in the context of the high dimensionality and weak correlation between features of the Higgs dataset, they will be able to yield valuable insights with respect to feature importance and non-linear relationships within the data. Among them, XGBoost should achieve the best performance on account of its ability to model complex interactions effectively in a more computationally efficient manner when dealing with large datasets. Random Forest will also perform well but will be computationally more expensive. SVM instead might not be able to capture very comfortably the nonlinearities of data but would certainly provide a good base benchmark.

2) Hyperparameter Selection

For the best optimization of our model, we will systematically explore several combinations of hyperparameters using grid search and keep the best one for each model. Especially since the Higgs dataset is very complex and large, with 11 million records, and has very low correlation amongst its features, tuning must be done very carefully to harmonize accuracy versus computation versus generalization. The following hyperparameters are proposed to be effective as preliminary starting ones. For Random Forest, we chose n_estimators=500 for stability while taking advantage of parallel computation, set max_depth=30 as a control against tree growth and overfitting, and min_samples_split=10 to maintain generalization by requiring sufficient samples for splitting. For Support Vector Machine (SVM), we used radial basis function (RBF) as the kernel to capture non-linear relationships. C was set to 1.0, a balanced default. and gamma="scale", which will automatically set the kernel coefficient based on the distribution of the dataset. Because of the high computational cost of SVM on large datasets, these initial values are thus intended to provide a rough balance between accuracy and efficiency. For XGBoost, we set n_estimators=1000 with an early-stop mechanism to avoid overfitting, max_depth=8 for complexity control while being able to capture interactions, learning_rate=0.05 to allow for small steps that improve generalization, subsample=0.8 for randomness, and colsample_bytree=0.8 to restrict the feature number for each boosting round, lowering the risk of overfitting. These starting values will guide grid search so as to give the best fit of our models to the Higgs dataset while being computationally plausible.

3) Evaluation Metrics

The dataset will be split into 3 sets train, validation and test sets. The training set will contain 80% of the data and the rest will be split between test and validation (10% each).

Once the model is trained, we will use Receiver Operating Characteristic Area Under Curve (ROC-AUC), precision, recall, and F1-score to evaluate the model performance. [22] state that using these metrics will give more information since they follow the characteristics of datasets by focusing on both class separability (ROC-AUC) and the evaluation of minority class performance (precision, recall, and F1-score). These metrics provide a comprehensive understanding of the model's effectiveness in correctly identifying and classifying the minority classes.

REFERENCES

[1] A. Alves, “Stacking machine learning classifiers to identify Higgs bosons at the LHC,” Journal of Instrumentation, vol. 12, pp. T05005–T05005, 2016, [Online]. Available: https://api.semanticscholar.org/CorpusID:14727229

[2] M. Svedin, A. Podobas, S. W. Der Chien, and S. Markidis, “Higgs Boson Classification: Brain-inspired BCPNN Learning with StreamBrain,” CoRR, vol. abs/2107.06676, 2021, [Online]. Available: https://arxiv.org/abs/2107.06676

[3] K. Lasocha, E. Richter-Was, D. Tracz, Z. Was, and P. Winkowska, “Machine learning classification: Case of Higgs boson CP state in H→ττ decay at the LHC,” Physical Review D, 2018, [Online]. Available: https://api.semanticscholar.org/CorpusID:119106065

[4] A. Bashyal, P. van Gemmeren, S. Sehrish, K. Knoepfel, S. Byna, and Q. Kang, “Data Storage for HEP Experiments in the Era of High-Performance Computing,” 2022. [Online]. Available: https://api.semanticscholar.org/CorpusID:247451085

[5] C. McNeile, “Data storage issues in lattice QCD calculations,” ArXiv, vol. hep-lat/0003009, 2000, [Online]. Available: https://api.semanticscholar.org/CorpusID:33550160

[6] R. Nelakurti and C. Hill, “Evaluating Modifications to Classifiers for Identification of Higgs Bosons,” 2024. [Online]. Available: https://api.semanticscholar.org/CorpusID:272694091

[7] C. Adam-Bourdarios, G. Cowan, C. Germain, I. Guyon, B. Kégl, and D. Rousseau, “The Higgs boson machine learning challenge,” in Proceedings of the NIPS 2014 Workshop on High-energy Physics and Machine Learning, G. Cowan, C. Germain, I. Guyon, B. Kégl, and D. Rousseau, Eds., in Proceedings of Machine Learning Research, vol. 42. Montreal, Canada: PMLR, Mar. 2015, pp. 19–55. [Online]. Available: https://proceedings.mlr.press/v42/cowa14.html

[8] A. Roy, “FAIR Principles for data and AI models in high energy physics research and education,” Proceedings of 41st International Conference on High Energy physics — PoS(ICHEP2022), 2022, [Online]. Available: https://api.semanticscholar.org/CorpusID:254044690

[9] S. Campana, A. Di Girolamo, P. Laycock, Z. Marshall, H. Schellman, and G. A. Stewart, “HEP computing collaborations for the challenges of the next decade,” 2022. [Online]. Available: https://api.semanticscholar.org/CorpusID:247447088

[10] C. W. Murphy, “Class imbalance techniques for high energy physics,” SciPost Physics, 2019, [Online]. Available: https://api.semanticscholar.org/CorpusID:141459371

[11] Y. Chen et al., “A FAIR and AI-ready Higgs boson decay dataset,” Sci Data, vol. 9, 2021, [Online]. Available: https://api.semanticscholar.org/CorpusID:236924419

[12] R. Nelakurti and C. Hill, “Evaluating Modifications to Classifiers for Identification of Higgs Bosons,” 2024. [Online]. Available: https://api.semanticscholar.org/CorpusID:272694091

[13] H. Wang, M. J. Bah, and M. Hammad, “Progress in Outlier Detection Techniques: A Survey,” IEEE Access, vol. 7, pp. 107964– 108000, 2019, doi: 10.1109/ACCESS.2019.2932769.

[14] P. Fergus and C. Chalmers, Applied Deep Learning: Tools, Techniques, and Implementation. in Computational Intelligence Methods and Applications. Springer International Publishing, 2022. [Online]. Available: https://books.google.co.uk/books?id=eJv5zgEACAAJ

[15] N. Fei, Y. Gao, Z. Lu, and T. Xiang, "Z-Score Normalization, Hubness, and Few-Shot Learning," in 2021 IEEE/CVF International Conference on Computer Vision (ICCV), 2021, pp. 142–151. doi: 10.1109/ICCV48922.2021.00021.

[16] S. Aslam and T. F. Rabie, "Principal Component Analysis in Image Classification: A review," in 2023 Advances in Science and Engineering Technology International Conferences (ASET), 2023, pp. 1–7. doi: 10.1109/ASET56582.2023.10180847.

[17] C. E. Ogbuanya, "Improved Dimensionality Reduction of various Datasets using Novel Multiplicative Factoring Principal Component Analysis (MPCA)," ArXiv, vol. abs/2009.12179, 2020, [Online]. Available: https://api.semanticscholar.org/CorpusID:221949214

[18] S. Feizi and D. Tse, "Maximally Correlated Principal Component Analysis," ArXiv, vol. abs/1702.05471, 2017, [Online]. Available: https://api.semanticscholar.org/CorpusID:12116408 [19] L. Breiman, "Random Forests," 2001.

[20] C. Cortes and V. Vapnik, "Support-vector networks," Mach Learn, vol. 20, no. 3, pp. 273–297, 1995, doi: 10.1007/BF00994018.

[21] T. Chen and C. Guestrin, "XGBoost: A Scalable Tree Boosting System," CoRR, vol. abs/1603.02754, 2016, [Online]. Available: http://arxiv.org/abs/1603.02754

[22] J. Hancock, T. M. Khoshgoftaar, and J. M. Johnson, "Informative Evaluation Metrics for Highly Imbalanced Big Data Classification," in 2022 21st IEEE International Conference on Machine Learning and Applications (ICMLA), 2022, pp. 1419–1426. doi: 10.1109/ICMLA55696.2022.00224.