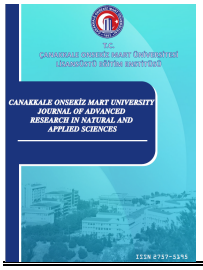
e-ISSN 2757-5195



Çanakkale Onsekiz Mart University

Journal of Advanced Research in Natural and Applied Sciences

Open Access

doi.org/10.28979/jarnas.1242840 2023, Vol. 9, Issue 3, Pages: 560-576 dergipark.org.tr/tr/pub/jarnas

An Educational Approach to Higgs Boson Hunting Using Machine Learning Classification Algorithms on ATLAS Open Data

Ayşe Bat [1,2,*]

[1]Bandırma Onyedi Eylül University, Faculty of Engineering and Natural Sciences, Engineering Sciences Department, 10200, Bandırma, Balıkesir, Türkiye
[2]Erciyes University, Faculty of Science, Department of Physics, 38280, Kayseri, Türkiye

Article History

Received: 26.01.2023

Accepted: 17.05.2023

Published: 20.09.2023

Research Article

Abstract – In this study, the performance of several classification algorithms that are used to separate the H → ττ signal from background is investigated. The data set came from the publicly available ATLAS data, which was utilized for the Machine Learning (ML) competition. The data was obtained from a full ATLAS simulation and originated from proton-proton collisions. There are 250 thousand events in the data set, and 70% of them were used to train the algorithms. The primary objective of this research is to identify the signal events from the background events by using various ML methods in the context of high-energy physics. In order to discover a solution to the binary classification problem that was discussed earlier, six distinct classification algorithms were utilized. This article also compares the performance of these classification algorithms, including Linear Support Vector Machines (SVM), Radical SVM, Logistic Regression, K-Nearest Neighbours, XGBoost Classifier, and the AdaBoost Classifier. The best results were obtained using the XGBoost Classification method, which had an AUC of $0.84 \pm 1.9 \times 10^{-3}$ followed by the AdaBoost Classifier with an AUC of $0.82 \pm 2.5 \times 10^{-3}$.

Keywords *– ATLAS Experiment, binary Ccassification, LHC, machine Llarning, XGBoost classifier.*

1. Introduction

ATLAS (Armstrong et al., 1994) is a general-purpose particle physics experiment located at the Large Hadron Collider (LHC) at CERN. The detector consists of many-layered subdetectors that are centered around the collision point to record particle kinematic properties. The ATLAS studies a wide range of physics topics to search for the exotic particles coming from proton-proton (pp) collisions at the center of the detector. One of these research leads to the discovery of the Higgs boson, which is the LHC's overarching goal. This discovery completed much of the missing part of the Standard Model (SM), and thus the theory that explained how matter initially gained mass was confirmed. ATLAS (Aad et al, 2012) and CMS (Chatrchyan et al, 2012) made public the discovery of the Higgs, also known as the God particle, about a decade ago. After all, simply discovering the Higgs is not the end of the story; researchers will continue to examine the Higgs in depth in order to evaluate the features it has. (A detailed map of Higgs boson, 2022; A portrait of the Higgs boson, 2022).

The Higgs bosons produced during the high-energy collisions at the LHC decays immediately to other fundamental particles. As a result, the detector is designed in such a way that the signature of the Higgs boson cannot be detected directly; instead, the detector observes and analyzes the decay products of the Higgs boson. In the LHC, the Higgs boson can decay in the bosonic decay channel ($\gamma\gamma$ WW, ZZ) (Aaboud et al., 2018a; 2018b; 2019a) or fermionic decay channel ($\tau\tau, b\bar{b}$) (Aaboud et al., 2019b) depending strongly on its mass (Flechl, 2015). Although the Higgs to tau-tau (H→ττ) decay channel is the most sensitive channel among leptonic Higgs boson decays, due to its high branching rate. This decay channel is experimentally difficult to study two reasons; first, the presence of neutrinos (υ) in the final state makes it difficult to estimate the mass of the Higgs

[1] abat@bandirma.edu.tr
[*]Corresponding Author

candidate, as the detector is not sensitive enough to measure υ. Second, the Z boson also decays into two τ, which is far more common than the Higgs decay, and their masses are close to each other, which makes it difficult to separate them (Aad, 2012).

As part of the research, Machine Learning (ML) techniques were developed to distinguish signal events in which the Higgs boson decays into two τ from background events. The open data that was provided as part of the ATLAS ML competition will serve as the data set that is used for this research (ATLAS Collaboration, 2014). The dataset consists of H→ττ labeled as signal events, and three different background events labeled as background that can mimic a signal in this channel. The topology is considered in signal events in which the Higgs decays into two τ: events in which one of the τ decays into an electron (e) or a muon (μ) and two υ, and the other τ decays into hadrons and a υ. The following is a list of background events that were taken into consideration throughout the simulation (Adam-Bourdarios, 2015).

- Z boson decays into two τ,
- Events involving a pair of top quarks (q), which can decay into a lepton and a hadronic τ,
- Events involving the decay of the W boson, in which an e or μ and a hadronic τ can occur simultaneously, this can occur only through defects in the particle identification procedure.

The process for mass generation of fermions in the SM can only be observed by the direct decay of the Higgs boson into fermions (Aad et al., 2015). The fermionic decay channels with the highest branching rates of the Higgs boson are states where it decays into a pair of b-quarks, $H \rightarrow bb$, and a pair of τ leptons, $H \rightarrow \tau\tau$. Although it is quite difficult to eliminate the $H \rightarrow bb$ channel background events, the $H \rightarrow \tau\tau$ decay channel is the most promising decay channel for measuring Higgs boson coupling to fermions (Aad & Abbott, 2022). Besides, the decay channel $H \rightarrow \tau\tau$ (Aadetal, 2022; Tumasyanetal, 2022) is used for direct measurements of Yukawa coupling of Higgs into fermions and to understand the CP violation nature of the Higgs thanks to angle distributions of the decay products of τ leptons (Aad et al., 2020).

In this study, $H \rightarrow \tau\tau$ signal events, and background events were identified using multiple ML classification techniques using ATLAS open data. A comprehensive analysis of the data is provided in the next section of the article. Information on the data may be found in Section 2.2. Exploratory data analysis is discussed in Section 2.2. Data pre-processing steps are explained in Section 2.3. The evaluation of the solution to the problem is discussed in Section 2.4, along with the ML algorithms and parameters that were employed in the evaluation. In the third chapter, the results are presented together with the corresponding interpretations.

2. Data and Machine Learning Model

2.1. Data

The ATLAS full detector simulations were used to generate the dataset. After the initial phase of the simulation, which involves simulating collisions between protons, the simulated particles are then followed through a model of a virtual detector. As an outcome of this, one generates a dataset of simulated events with statistical attributes that are equivalent to those of real events (Adam-Bourdarios et al., 2015). There are thirty feature vectors that each represent an event that was produced by the collider, and the dataset has a total of two hundred and fifty thousand events. The initial name PRI stands for the raw data measured by the detector, such as the speed of the particles. The initial name DER stands for the properties found by processing this raw data. Each of the detailed feature vector descriptions can be found in the reference (ATLAS Colloboration, 2014).

In terms of the ML approach, the problem can be solved as a binary classification problem. Here, the challenge of separating the signal events specifically, the decays of $H \rightarrow \tau\tau$ from the background was investigated.

2.2. Exploratory Data Analysis

Exploratory Data Analysis (EDA) is the first and most crucial step before implementing ML algorithms, which may help us identify patterns by statistically and visually examining the data. Figure 1 in the left shows the estimated mass of the Higgs boson (m_H) candidate. This estimate was obtained through a probabilistic phase space integration. Figure 1 in the middle shows the absolute value of the pseudorapidity (η) separation

between the two jets ($|\eta_{jet1} - \eta_{jet2}|$). If there is only one jet or none, then the value of this variable is undefined. Figure 1 in the right shows the ratio of the transverse momenta (p_T) of hadronic τ and the lepton. On the left side of Figure 2 is a representation of the invariant mass of the two jets. The invariant mass of the 4-momentum sum is used to calculate the invariant mass of two particles. Figure 2 in the center shows the R separation between the hadronic τ and the lepton ($\sqrt{(\eta_\tau - \eta_l)^2} + \sqrt{(\phi_\tau - \phi_l)^2}$), and Figure 2 on the right shows the sum of the p_T of the hadronic τ, the lepton, and the leading jet. Figure 3 illustrates the pseudorapidity (η), phi (ϕ), and p_T ($\sqrt{p_x^2 - p_y^2}$) of the leading jet (Fernow, 1983), respectively. Figure 4 shows the η, ϕ, and p_T of the hadronic τ, respectively. Figure 5 shows the η, ϕ, and p_T of the lepton respectively.

The jets are sprays of particles that arise when a parton (quark or gluon) undergoes hadronization in a pp collision. The leading jet is referred to as the one with the largest p_T. Different techniques are used to reconstruct jets, which are defined by calorimetry and tracking data. Cones are used by the reconstruction algorithm to iteratively go over each particle in the detector and combine them according to a set of criteria (Atkin, 2015). All methods employ the cone of radius R to rebuild jets, while various algorithms may have different criteria. With careful consideration, this value was established for the CMS experiment as well as for ATLAS for calculating the mass and energy of the jet. Determining a larger jet radius (R) is crucial for effectively collecting enough of the hadronized particle. R separation is important for jet definition since it provides for a distinct separation of particles that are and are not part of the jet. This is significant because it improves in distinguishing the signal (particles created by the high-energy parton) from the background (particles produced by other sources such as the underlying event or pile-up collisions (Butterworth, 2008).

Among the features in the dataset, some selected distributions that contribute the most to signal and background separation are shown in the figures. The plots shown here separate signal and background event distributions on these features. The units for energy, mass, and momentum are all GeV, although there are no units specified for any of the other variables. The azimuthal angle ϕ is defined as the angle formed by the $-\pi$, and $+\pi$ ranges.

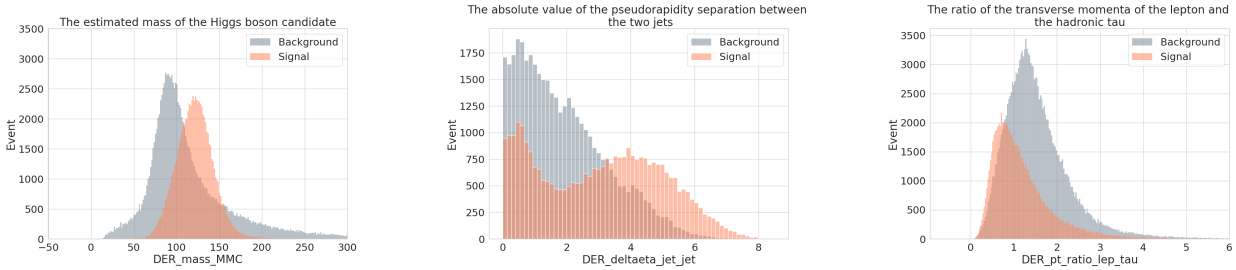


Figure 1. The estimated m_H candidate (left), the absolute value of the η separation between the two jets (middle), and the ratio of p_T of the lepton and the hadronic τ (right). The signal events are depicted in orange, while the background events are shown in gray.

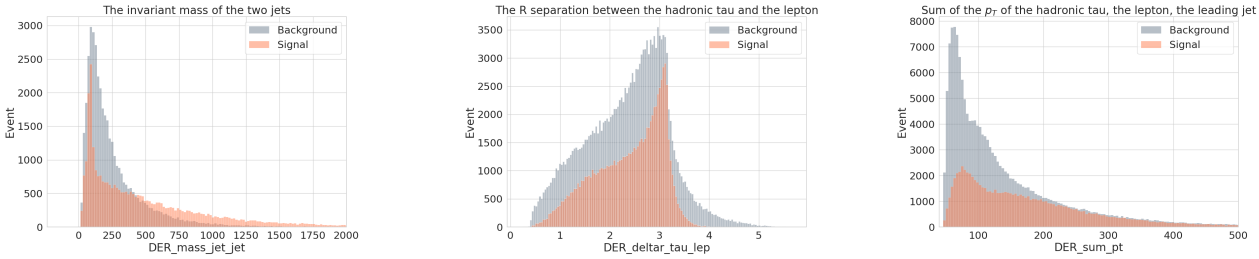


Figure 2. The invariant mass of the two jets (left), the R separation between the lepton and hadronic τ (middle), sum of the p_T of the leading jet, the lepton, and the hadronic τ (right). The signal events are depicted in orange, while the background events are shown in gray.

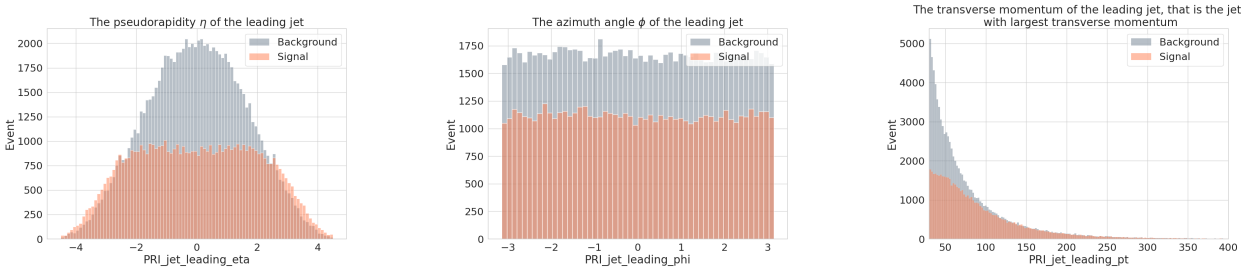


Figure 3. The η (left), ϕ (middle) and the p_T of the leading jet (right). The signal events are depicted in orange, while the background e events are shown in gray

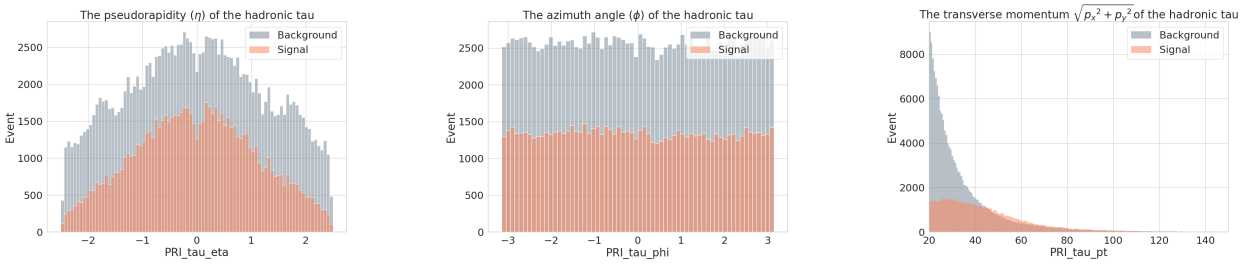


Figure 4. The η (left), ϕ (middle) and the p_T of the hadronic τ (right). The signal events are depicted in orange, while the background events are shown in gray.

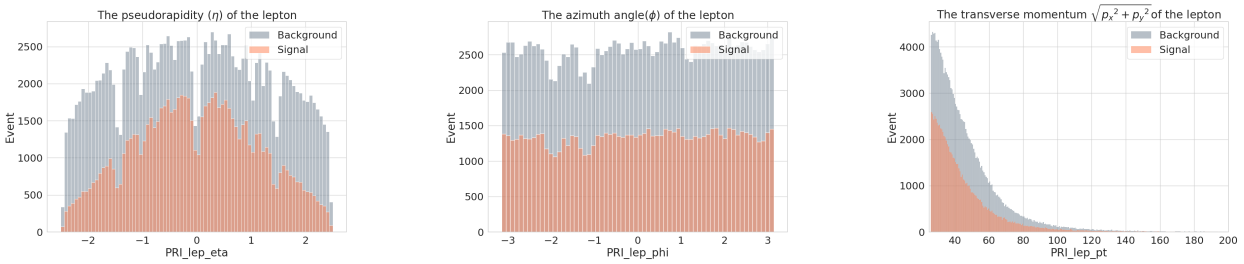


Figure 5. The η (left), ϕ (middle) and the p_T of the lepton (right). The signal events are depicted in orange, while the background events are shown in gray.

2.3. Data Pre-processing

The process of ML begins with the study of missing data and continues with the removal of chunks of data from the dataset. Both processes are essential. The handling of missing data may be done in a few different ways. The most common practice is to replace the missing data with a proper value. In the data set, there is a data group that expresses the number of jets (PRI_jet_num) produced in the events, as well as other data groups that represent the properties of these jets. Since there cannot be a "leading jet" in the absence of a jet (PRI_jet_num = 0) the primitive quantities associated with the jet are structurally undefined, as well as in properties derived from them (η, ϕ, p_T). So, the missing values in the dataset originated from the event that has no jet in it. The missing data is defined in the dataset as a large meaningless number (-999), and those definitions do not subsequently affect the ML classification (Müller, 2016).

A data point that significantly deviates from the rest of the data is referred to as an outlier. Outliers change the mean, standard deviation, and median because they throw off how the data are usually spread out statistically. Extreme values may be deleted from the data set to provide algorithms with more conventional information. The interquartile range approach was used to display the data and identify outliers using a box plot (Müller, 2016). The data was cleaned up by eliminating outliers highlighted by the boxplot. The Table 1 shows the summary of the cut implemented to remove the outlier. This process led to the deletion of 0.12% of the original data set.

Table 1
The table displays the cut implemented following the outlier procedure.

DER_mass_transverse_met_lep < 300
DER_pt_ratio_lep_tau < 12
Der_pt_tot < 400
DER_mass_MMC < 800
DER_mass_vis < 500
DER_ph_h < 700
PRI_tau_pt < 400

Before ML techniques can be used, the data must be changed into a suitable format. When looking at the information for this purpose, it becomes clear that transforming the feature that displays the number of jets into a category would be more helpful. Using the Sklearn library's OneHotEncoding function (Pedregosa, F., et al, 2011), one can classify the feature showing the number of jets in each event by altering the data type. This method creates new features in as many categories as there are, and their existence or absence is denoted by 1 or 0. To encode ML category properties, this function is by far the most popular choice. For the reason that it enables the discovery of patterns in the data collection. For the remaining numerical values in the data set, the Sklearn library's StandardScaler transformation was used. StandardScaler removes the feature's mean value from each feature variable and scales it to the feature's standard deviation (Scikit Learn, 2023b). Normalizing the dataset is a common step for ML estimators, and the resulting data looks like standard normally distributed data. The label data in the data set is denoted by 's' and 'b'. Using the Sklearn library's LabelEncoder (Scikit Learn, 2023a), signal events are encoded as *1* and background events as *0*. The entire data set contains around 40% signal events and 60% background events.

Feature engineering is a good way to figure out which characteristics are best for a given classification system (Müller, 2016). For each model of classification, the estimated mass of the Higgs boson has been seen as the most important candidate. In addition to this, the number of jets in an event, the transverse mass (m_T) between missing transverse energy (MET) and the lepton, the invariant mass of the hadronic τ and the lepton, the R separation between the hadronic τ and the lepton, and the hadronic features all show strong separation power. For feature selection, the correlation between each characteristic in the data set was analysed. The Pearson correlation coefficient technique (Nettleton, 2014), which is a typical methodology, essentially displays the linear relationship between two characteristics. If the correlation is too strong, the data may become overfit. The Figure 6 shows Pearson correlation coefficient for each feature vector. It is discovered that there is a high correlation in a few parameters related to jets and p_T. As a result, these strongly correlated characteristics were not implemented to train the model. Figures 7 to 13 are three-dimensional representations of selected features in the space of the estimated m_H and number of jets. The m_H gives the best separation power, and the number of jets is used as a categorical variable in the training data, which is the only categorical feature in the data set. Looking at the selected features in this two-feature space could help with understanding the separation power. The figures are also plotted for background and signal to see the different distributions for the selected feature.

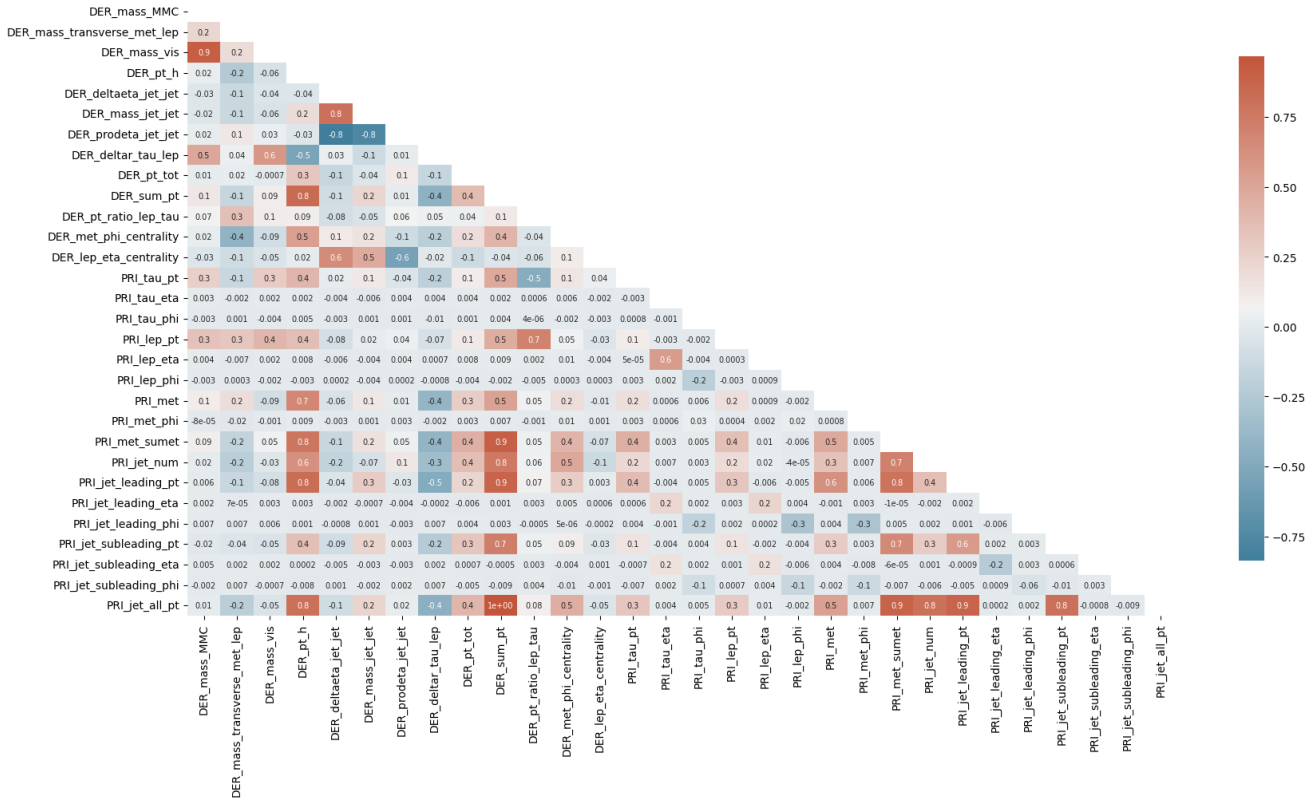


Figure 6. The figure shows Pearson correlation coefficient between each feature vector.

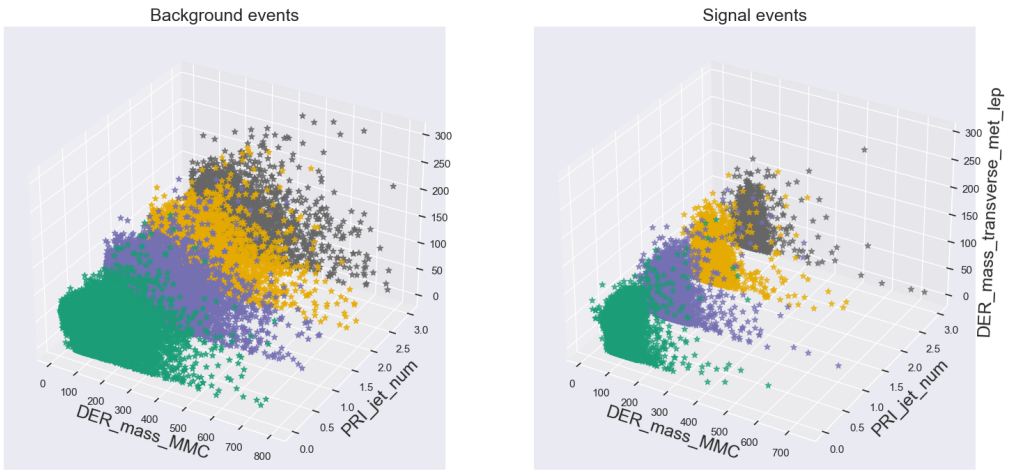


Figure 7. The m_T between MET and lepton feature is examined in the space of the estimated m_H and number of jets.

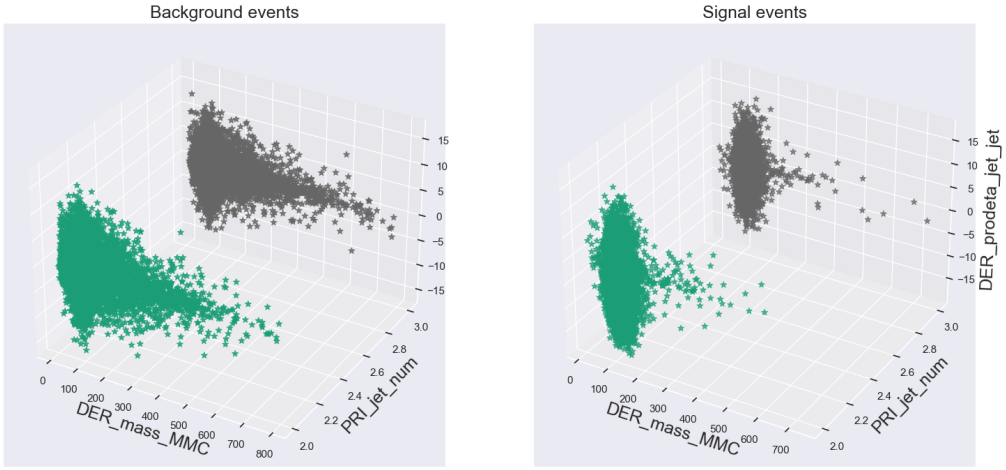


Figure 8. The yield of the η of the two-jets feature is examined in the space of the estimated m_H and number of jets.

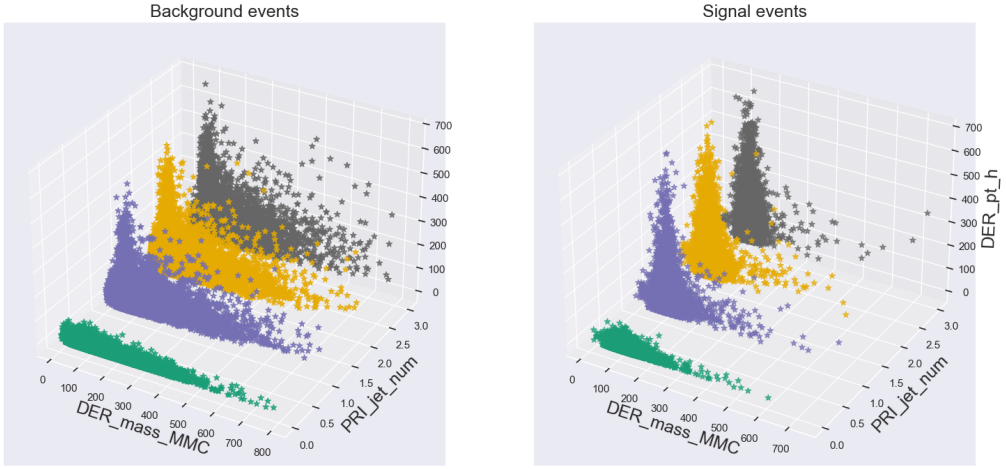


Figure 9. The vector sum of the p_T of the lepton, the hadronic τ, and the MET vector. This feature is examined in the space of the estimated m_H and number of jets.

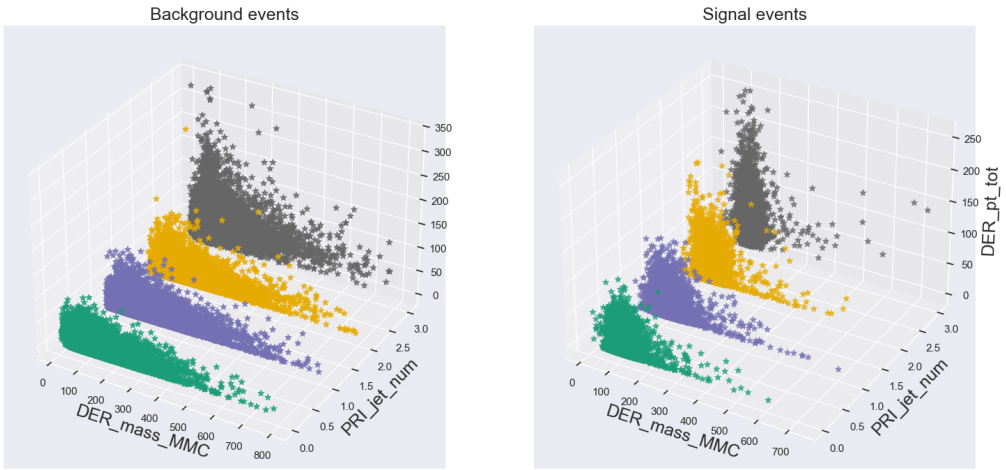


Figure 10. Total p_T, taking into account the missing p_T and p_T of the lepton, and hadronic τ. The leading jet is included if there are more than one jets, and the subleading jets are included if there are more than two jets. Total p_T feature is examined in the space of the estimated m_H and number of jets.

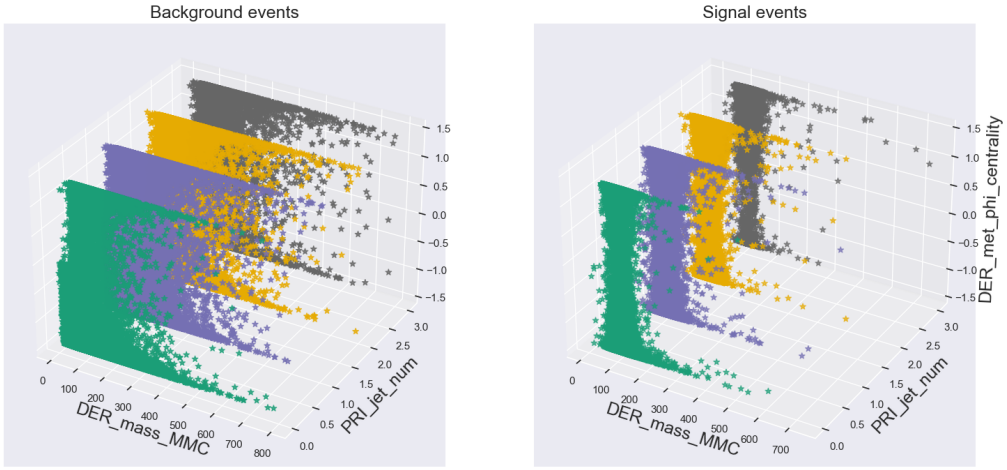


Figure 11. The features represent the ϕ of MET vector in relation to the hadronic τ and the lepton. The figure shows this feature in the space of the estimated m_H and number of jets.

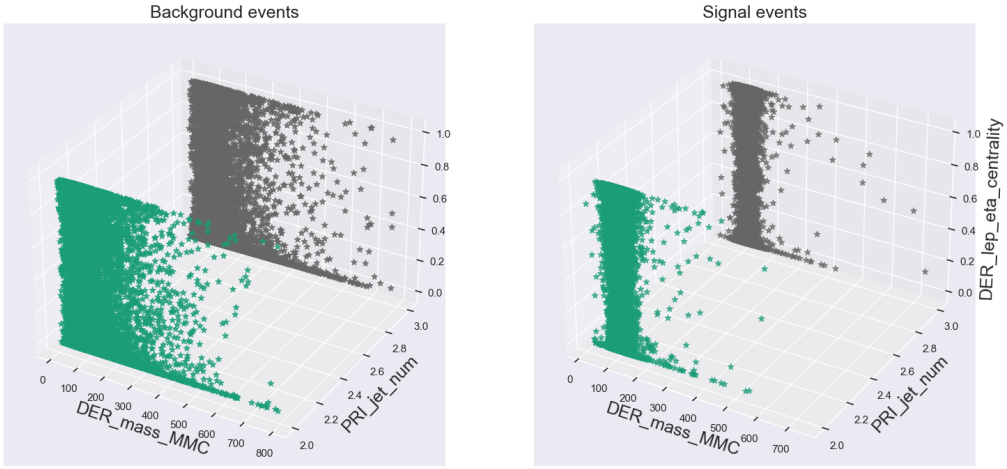


Figure 12. The features represent the η of MET vector in relation to the hadronic τ and the lepton (undefined if PRI_jet_num ≤ 1). The figure shows this feature in the space of the estimated m_H and number of jets.

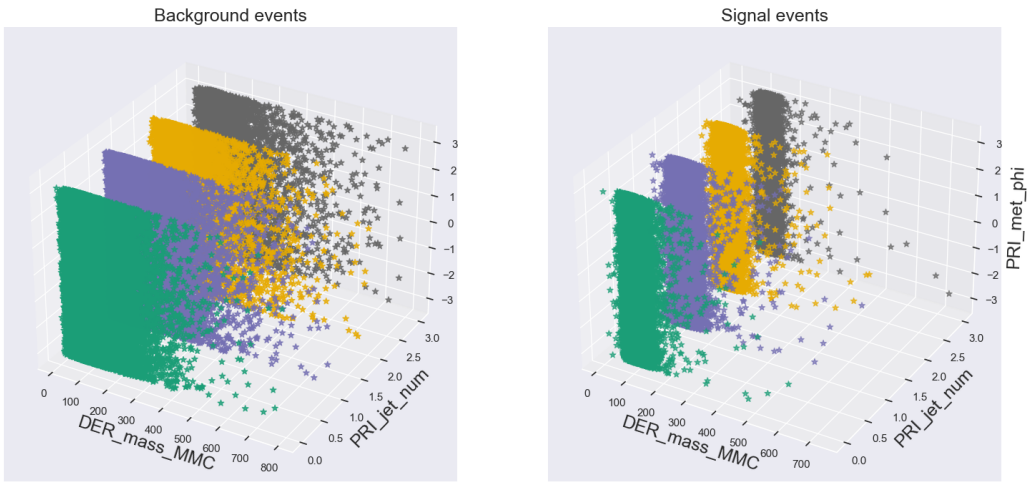


Figure 13. The ϕ of the MET feature in the space of the estimated m_H and number of jets.

Firstly, EDA studies on feature vectors were used to evaluate the data set. The outlier data values have been excluded from the data set at this particular stage. To identify the relationship between the feature vectors, the Pearson correlation coefficient technique was utilized. When training the models, features with a strong correlation were not implemented. Each feature in the estimated m_H and jet number space was examined before these features were selected and the process proceeded with features that had strong signal and background separation abilities. When training the models, the feature vectors listed in the Table 2 are taken into account.

Table 2
The features of the vectors that were selected for training the model.

'DER_mass_MMC', 'DER_mass_transverse_met_lep', 'DER_prodeta_jet_jet', 'DER_pt_h', 'DER_deltar_tau_lep', 'DER_pt_tot', 'DER_pt_ratio_lep_tau', 'DER_met_phi_centrality', 'DER_lep_eta_centrality', 'PRI_tau_pt', 'PRI_tau_eta', 'PRI_tau_phi', 'PRI_lep_pt', 'PRI_lep_eta', 'PRI_lep_phi', 'PRI_met_phi', 'PRI_jet_num', 'PRI_jet_leading_pt', 'PRI_jet_subleading_eta', 'PRI_jet_subleading_phi', 'PRI_jet_leading_eta', 'PRI_jet_leading_phi'

2.4. Machine Learning Model

The goal of applying classification algorithms to training data is to group together pre-labeled features that are like one another. To achieve this goal, data training is performed with the use of supervised learning algorithms. These algorithms make use of predefined classifications within the data. This article makes use of the following classification algorithms: Logistic regression, Linear Support Vector Machines (SVM), Radial SVM (Cortes & Vapnik, 1995), K-Nearest Neighbours (KNN) (Mucherino, 2009), XGBoost Classifier (Chen & Guestrin, 2016), and AdaBoost Classifier. The models such as Decision Trees and Gaussian Naive Bayes were also tested, both of which are great candidates for addressing classification issues; nonetheless, the models did not yield results that were as good as those achieved by other techniques. Because of this, we decided to concentrate on the aforementioned six classification methods. With the use of physics data, the purpose of this article is to experiment with various ML algorithms and then describe how to choose a statistically stable model based on how the results are interpreted. This is the overarching objective of the project.

The logistic regression model learns a linear relationship from the submitted dataset and then delivers a nonlinear relationship using a sigmoid function, allowing for categorizing data. This classification issue is simple to implement and identifies classes quickly; however, it uses linear bounds. The reason we present a comparison to the other model is to evaluate whether there is any nonlinearity in the dataset. The SVM functions to find a hyperline that splits the data into classes. This approach is commonly used for categorization. The algorithm provides a kernel option for implementing nonlinear data. The "RBF" kernel option makes it possible to utilize the model with nonlinear data. Both linear and radial SVMs are used to demonstrate how model training performs in both kernel selection scenarios. The "linear" kernel option of SVM was expected to perform similarly to Logistic regression. KNN is a distance-based classifier and begins by defending an unknown data point. Then, compute the distance between this unknown data and every instance of train data. The goal is to discover the closest data point and proclaim it as the nearest neighbour. The shorter the distance, the more related they are. This approach was used to see how distance-based classifiers performed on the data. However, employing and working with excessive parameter optimization may lead the machine to function to the extreme. The first results of this method performed rather well, but not well enough to suggest a detailed hyperparameter adjustment. XGBoost is an effective ensemble approach that combines numerous decision tree-based algorithms to enhance accuracy. The approach also employs parallel computing, which is an ensemble method in which learners are created simultaneously. The power of the method is derived from the combination of the decision tree algorithm, although this may result in an overfitting problem. The L1 and L2 regulation options are provided to avoid overfitting, which adds a penalty term to the loss function to avoid overfitting. L1 regularization, also known as "lasso regression," adds the "absolute magnitude" of the coefficient as a penalty term, whereas L2 regularization, also known as "ridge regression," adds the "squared magnitude" of the coefficient. AdaBoosting is an ensemble approach that employs sequential ensemble methods. As a result, the learners are created progressively, with each learner impacting the next. In general, the ensemble technique combines several separate models to form a master model. AdaBoosting is a model that helps to increase the

accuracy of weak models; however, it is typically used to classify text and pictures rather than binary classification issues. Because outliers and noisy datasets have a negative impact on the model, the method must be carefully prepared for model training.

A confusion matrix (CM) included in the Sklearn package will be utilized to assess the accuracy of the predictions made by a classification algorithm. The true positives, false positives, true negatives, and false negatives are the metrics used to evaluate model performance. These metrics, which are also shown in Figure 14, have been interpreted according to the solution of the problem to be concluded with this article. In the case described here, signal events are considered positive cases.

- True Positives (TP): Signal events predicted as a signal.
- False Positives (FP): Background events predicted as a signal.
- True negatives (TN): Signal events predicted as background.
- False Negative (FN): Background events predicted as background.

The parameters to be calculated with the classification report are precision, recall, and f1-score. Each class's predictions require a unique calculation of these parameters, which are done independently. While the precision shows the ratio of true signal events within the predicted signal events (TP/TP+FP), the recall shows the ratio of true signal events to all signal events (TP/ TP+FN). The f1score uses to compare classification models and calculated as a weighted harmonic mean of precision and recall (2* precision * recall / precision + recall) (Bonnin, 2017).

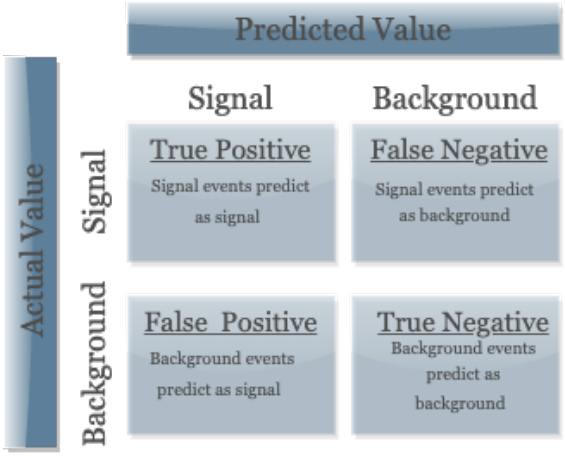


Figure 14. A CM is used to evaluate the performance of the classification models.

2.4.1. Hyper Parameter Tuning

Hyperparameter tuning is the process of identifying an optimum set of parameters for models in order to reach a final optimized model output. The *GridSearchCV* package is used for this, and *CV* stands for cross-validation (Scikit Learn, 2023c). This library searches a grid of hyperparameter values for the optimal collection of hyperparameters. As a result, each model was trained for each permutation of a particular parameter. It took a lengthy time to compute the entire set of parameters. The table 3 provides the parameter set that delivers the hyperparameter space as well as the parameter sets that give the best accuracy result. The KNN model caused the overfitting problem, so the most basic parameter set, the default values, was used to train the model. As a result, the default variables for the KNN model are n_neighbors=5, weights=uniform, algorithm=auto, leaf_size=30, p=2, and metric=minkowski. Though the model is simple, the algorithm tends to overfit. In other words, the supplied train set estimates the train more accurately than the test data since it learns so much. Hyperparameter adjustment for the KNN model is not done because of this issue.

Table 3
Hyper parameter list for each model and final parameter set used for the train the models.

Model Name	Hyperparameter Space	Selected Parameter
Logistic Regression	*'solver': 'penalty'* *'lbfgs'* : ['l2', None] *'liblinear'* : ['l1', 'l2'] *'newton-cg'* :['l2', None] 'newton-Cholesky' : ['l2', None] *'sag'* : ['l2', None] *'saga'* : ['elastic net', 'l1', 'l2', None] *'C'* : [used to numpy logspace function create a logarithmically spaced 21 points between 10^{-10} and 10^{+10}]	*'C'* : 10.0 *'penalty'*:'l2' *'solver':* 'liblinear'
Linear SVM	*'C':* [0.1, 1,10,100] *'gamma'*: [0.01, 0.1, 1]	*'C': 100,* *'gamma': 0.1,* *'kernel': 'linear'*
Radical SVM	*'C':* [0.1, 1,10,100] *'gamma'*: [0.01, 0.1, 1]	*'C': 100,* *'gamma': 0.01,* *'kernel': 'rbf'*
XGBoost	*'eta'*: [0. 0.2 0.4 0.6 0.8 1.0 *'max_depth'*: [6,8,10] *'gamma'*: [0.5, 1, 2, 5]	*eta:0.2* *max_depth:6* *gamma:5*
AdaBoost	*'n_estimators':* [10, 50, 100, 200, 500] *'learning_rate'*: [0.0001, 0.01, 0.1, 1.0, 1.5]	*'learning_rate'* : 1.0, *'n_estimators'* : 500

3. Result

In this study, the classification algorithms Linear SVM, Radical SVM, Logistic Regression, KNN, XGBoost, and AdaBoost Classifier were utilized to distinguish signal events from background events. The performance of these classification algorithms was assessed by calculating their Area Under Curve (AUC) scores, as well as their Receiver Operation Characteristics (ROC), CM and classification reports.

The AUC score represents the classification powers of each algorithm, whereas ROC is a probability curve. The "*predic_proba*" function from Sklearn libraries was utilized for model prediction, which returns the likelihoods associated with a classification label (Bruce, 2020). After that, these predictions are compared with the actual classification. Each algorithm's trained models are used to separately predict the class in the test dataset and the training dataset. Test dataset model results are taken into consideration as model performance as they were not previously used to train the model, whereas train dataset model results are used to assess overfitting. Furthermore, cross-validation (CV) was used to demonstrate that the AUC score is statistically consistent and to prevent overfitting (Browne, 2000). In this approach to resampling, the dataset is broken up into ten several distinct sections that are then used to test and train a model at various iterations. CV uses nine samples to train the data and one sample as the test sample. The XGBoost Classifier has provided the highest value with 0.84 ± 1.9x10-3 its AUC value compared to the other classification methods that were employed. AdaBoost Classifier comes in second with an AUC score of 0.82 ± 2.5 x 10^{-3}, while Radical SVM comes in third with an AUC value of 0.81 ± 2.7x10^{-3}. The AUC, ROC, and CV performance, along with its statistical variation, are presented in Table 4 for every classification algorithm. The train and test values for each algorithm are shown separately for both AUC, ROC, and CV. Only the KNN model showed a tendency to overfitting. Although the default model is used, the model seems to have learned a lot from train data. This problem can be solved by adding more data to the dataset. A model that was trained using train data yields results that are comparable when predicting the train and test data. This is a sign that the training model's results is acceptable.

Table 4
The AUC, ROC, and CV score for train and test sample of classification models. The CV displays the model's accuracy mean value of the ten fold and standard deviation.

Model Name	AUC		ROC		Cross-Validation	
	Train	Test	Train	Test	Train	Test
Logistic Regression	74.08	73.8	79.7	79.5	$0.74 \pm 2.97x10^{-4}$	$0.74 \pm 2.7x10^{-3}$
Linear SVM	73.5	73.2	78.9	78.6	$0.73 \pm 3.94x10^{-4}$	$0.73 \pm 3.4x10^{-3}$
Radial SVM	80.7	80.3	87.2	86.7	$0.81 \pm 2.50x10^{-4}$	$0.81 \pm 2.7x10^{-3}$
KNN	84.0	76.4	91.5	81.6	$0.84 \pm 2.82x10^{-4}$	$0.77 \pm 2.1x10^{-3}$
XGBoost Classifier	85.3	83.8	92.5	90.7	$0.85 \pm 3.97x10^{-4}$	$0.84 \pm 1.9x10^{-3}$
AdaBoost Classifier	81.9	81.7	88.6	88.2	$0.82 \pm 3.29x10^{-4}$	$0.82 \pm 2.5x10^{-3}$

The Figure 15 shows the the ROC curve for the classification algorithms. The TP rate is plotted along the y-axis of the ROC curve, while the FP rate is plotted along the x-axis of the curve. The area below this curve shows the AUC value. In addition to this, it illustrates the model in its optimal state, in which the real positive value approaches 1 and the false positive value approaches 0. Classifiers that produce curves more nearby to the top-left region perform better. The figure 16 shows the ROC curve that indicates that the XGBoost Classifier performs best, with the AdaBoost Classifier coming in second. The figure16 shows the CV accuracy values for each fold from the CV study. In the CV investigation, a total of ten kFolds were applied. In other words, the data was separated into ten equal pieces, with one serving as a test. In the data separation section, stratified Kfold was employed (Scikit Learn, 2023d). to make sure the proportional distribution of signal and background events is taken into account. This process was performed ten times. Consequently, each model yielded ten distinct findings. The figure16 depicts the min, max, and mean values for this outcome. The difference between the min and max values and the large box are interpreted as statistically more uncertainty in the model. The XGBoost model results in the most accurate and most consistent results. AdaBoost Classifier is the second-best model and produces more consistent results than Radical SVM, which comes in third place behind AdaBoost Classifier.

The final step was to interpret the outcomes of the algorithms using the CM and classification report. The CM allows for the evaluation and comparison of TP, FP, FN, and TN results, and the classification report shows the result for the precision, recall, and f1-score parameters. When the outcomes of these parameters are compared, conclusions can be drawn about which models are more beneficial for distinguishing between signals and which are more useful for distinguishing between backgrounds. Figure 17 shows the result of the CM for each classification model. The positive case indicated here is background events. XGBoost and Radial SVM are the models that classify background events as background and so get the best results for the TP measure. Again, this pair of models have the smallest possible margin of error when it comes to classifying FN. By a wide margin, the XGBoost model has the best FP. In other words, this algorithm is the one that recognizes signals from background events with the least amount of mistakes possible. Finally, the XGBoost has the highest performance for TN values that symbolize the signal predicted as a signal.

The precision, recall, and f1-score values obtained by classification reporting are shown in Table 5. Precision represents the model's accurate predictive capacity since it is the ratio of true positive events to all positive predictions. The recall is defined as the proportion of TP cases to actual positive events. Since the f1-score is the harmonic average of these two parameters, it can be used to evaluate model performance. Table 5 shows the results of each parameter for signal and background separately. According to these outcomes, each model's ability to separate background events is greater than its ability to separate signal events. The XGBoost classifier outperformed all other classifiers in distinguishing signal and background events. When the AUC score, the results of the ROC curve, and the classification report that was provided earlier are examined, it can be observed that the XGBoost method has the best classification power for resolving this problem.

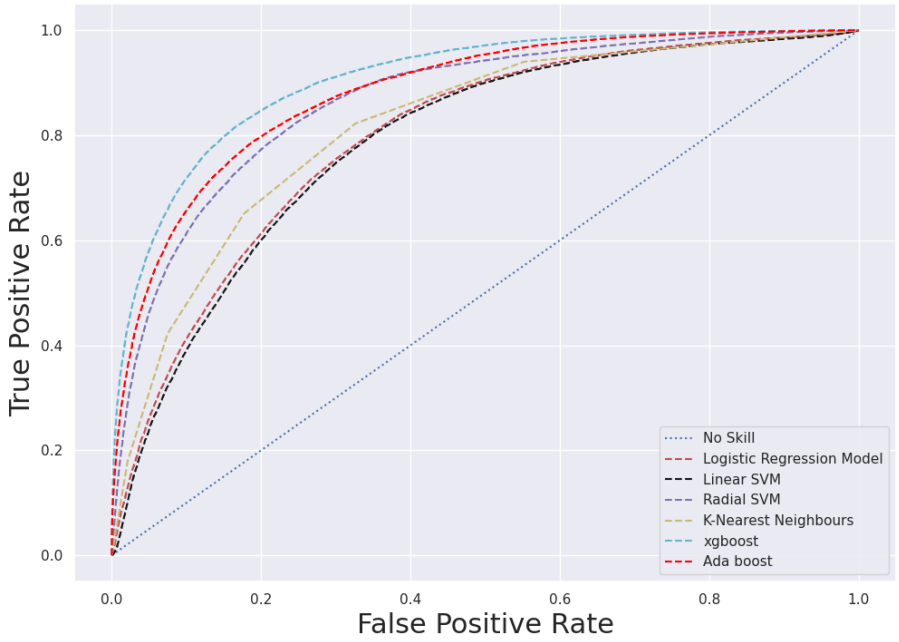


Figure 15. The Receiver Operation Characteristics curve for the classification algorithms.

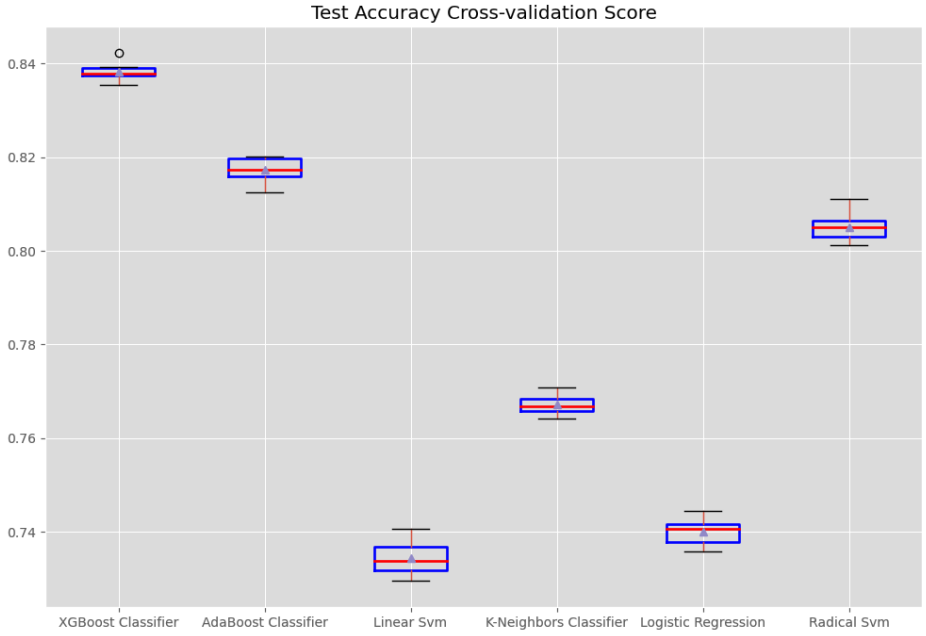


Figure 16. The cross-validation accuracy values for each fold from the cross-validation result. The red line indicates the mean of all result of accuracy.

.

.

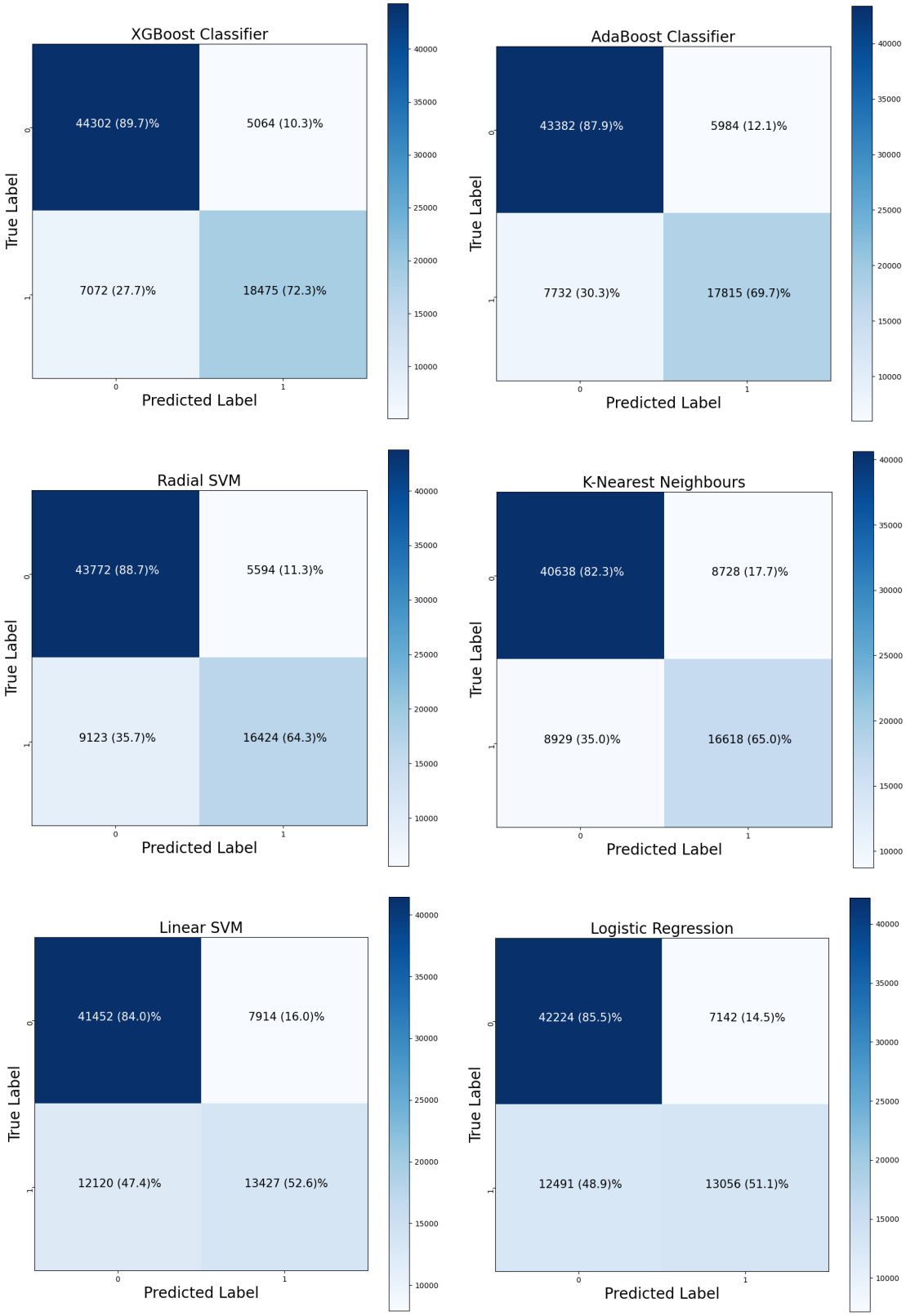


Figure 17. The result of each algorithm is according to the CM. TP at the bottom right, FN on the bottom left, FP on the top right, and a TN on the top left. The value shows how many instances of each parameter in a particular parameter were predicted.

Table 5
The table shows the performance of each classification model in terms of precision, recall, and f1-score. The table displays the findings for train and test data separately.

Model Name	Prediction Type	Precision		Recall		F1-Score	
		Train	Test	Train	Test	Train	Test
Logistic Regression	Background	0.77	0.77	0.86	0.86	0.81	0.81
	Signal	0.66	0.65	0.51	0.51	0.58	0.57
Linear SVM	Background	0.77	0.77	0.84	0.84	0.81	0.81
	Signal	0.64	0.63	0.53	0.53	0.58	0.57
Radial SVM	Background	0.83	0.83	0.89	0.89	0.88	0.86
	Signal	0.78	0.75	0.75	0.64	0.76	0.69
KNN	Background	0.87	0.82	0.89	0.82	0.88	0.82
	Signal	0.78	0.66	0.75	0.65	0.76	0.65
XGBoost Classifier	Background	0.87	0.86	0.91	0.90	0.89	0.88
	Signal	0.78	0.78	0.75	0.72	0.78	0.75
AdaBoost Classifier	Background	0.85	0.85	0.88	0.88	0.87	0.86
	Signal	0.76	0.75	0.70	0.70	0.73	0.72

4. Conclusion

This study uses the ML based data set from the ATLAS experiment at CERN to distinguish between background and signal events, namely those in which the Higgs boson decays into two tau particles. In this analysis, a variety of ML techniques were used, including a linear support vector machine, a radial support vector machine, a logistic regression, a k-nearest neighbours classifier, an XGBoost classifier, and an AdaBoost classifier. The performance of such classification algorithms may be measured by reports, ROC curves, and AUC score results. With an AUC of 0.84 ± 1.9 x 10-3, the XGBoost Classifier achieved the best accuracy, consistency, and stability, followed by the AdaBoost Classifier with an AUC of 0.82 ± 2.5 x 10-3. This research highlights the significance of selecting proper ML algorithms for high-energy physics classification tasks and demonstrates the use of the XGBoost and AdaBoost algorithms in this context. The research also looks at the models' performance in terms of true positive, false positive, false negative, and true negative metrics. The XGBoost Classifier has been shown to be the best efficient at distinguishing signals from background events while making the minimum of mistakes possible. Overall, the findings of this study attempt to develop a practical technique for working with high energy physics data using a ML approach.

Acknowledgements

The machine learning calculations reported in this paper were fully performed at TUBITAK ULAKBIM, High Performance and Grid Computing Center.

Conflicts of Interest

The authors declare no conflict of interest.

References

Aaboud, M., et al. (ATLAS Collaboration). (2018a). Measurement of the Higgs boson mass in the $H \rightarrow ZZ^* \rightarrow 4\ell$ and $H \rightarrow \gamma\gamma$ channels with s=13 TeV pp collisions using the ATLAS detector. *Physics Letters B*, 784,345-366. https://doi.org/10.1016/j.physletb.2018.07.050

Aaboud, M., et al. (ATLAS Collaboration). (2018b). Measurement of the Higgs boson coupling properties in the $H \rightarrow ZZ^* \rightarrow 4\ell$ decay channel at $\sqrt{s}$=13 TeV with the ATLAS detector. *J. High Energ. Phys*, 95. https://doi.org/10.1007/JHEP03(2018)095

Aaboud, M., et al. (ATLAS Collaboration). (2019a). Measurements of gluon–gluon fusion and vector-boson fusion Higgs boson production cross-sections in the $H \rightarrow WW^* \rightarrow e\nu\mu\nu$ decay channel in pp collisions at $\sqrt{s}$=13 TeV with the ATLAS detector. *Physics Letters B*, 789, 508-529. https://doi.org/10.1016/j.physletb.2018.11.064

Aaboud, M., et al. (ATLAS Collaboration). (2019b). Cross-section measurements of the Higgs boson decaying into a pair of τ leptons in proton-proton collisions at $\sqrt{s}$=13 TeV with the ATLAS detector. *Phys. Rev. D*, 99,072001. https://doi.org/10.1103/PhysRevD.99.072001

Aad, G., et al. (ATLAS Collaboration). (2022). Measurements of Higgs boson production cross-sections in the $H \rightarrow \tau^{+}\tau^{-}$decay channel in pp collisions at $\sqrt{s}$=13 TeV with the ATLAS detector. *JHEP*, 08, 175. https://doi.org/10.1007/JHEP08(2022)175

Aad, G. et al. (ATLAS Collaboration). (2020). Test of CP invariance in vector-boson fusion production of the Higgs boson in the H → ττ channel in proton–proton collisions at $\sqrt{s}$=13 TeV with the ATLAS detector. *Phys. Lett. B*, 805, 135426. https://doi.org/10.1016/j.physletb.2020.135426

Aad, G. et al. (ATLAS Collaboration). (2015). Evidence for the Higgs-boson Yukawa coupling to tau leptons with the ATLAS detector. *JHEP*, 117. https://doi.org/10.1007/JHEP04(2015)117

Aad, G., et al. (ATLAS Collaboration) (2012). Observation of a new particle in the search for the Standard Model Higgs boson with the ATLAS detector at the LHC. *Physics Letters B*, 716(1), 1-29. https://doi.org/10.1016/j.physletb.2012.08.020

Adam-Bourdarios, C., Cowan, G., Germain, G., Guyon, I., Kegl, B., Rousseau, D., (2015). The Higgs boson machine learning challenge. 664, s. 072015. *J. Phys.: Conf. Ser.*, DOI 10.1088/1742-6596/664/7/072015

Armstrong, W., et al. (ATLAS Collaboration). (1994). *ATLAS: technical proposal for a general-purpose pp experiment at the large hadron collider at CERN*. ATLAS Collaboration. doi:Retrived from: doi: 10.17181/CERN.NR4P.BG9K.

ATLAS Collaboration. (2014). *Dataset from the ATLAS Higgs Boson Machine Learning Challenge 2014. January 2022 tarihinde opendata*. Open Data. Retrived January 16, 2023, from http://opendata.cern.ch/record/328.

ATLAS Collaboration. (2022). A detailed map of Higgs boson interactions by the ATLAS experiment ten years after the discovery. *Nature*, 607, 52–59. https://doi.org/10.1038/s41586-022-04893-w.

Atkin, R. (2015). Review of the reconstruction algorithms. *J. Phys.: Conf. Ser.*, 645 012008. DOI: 10.1088/1742-6596/645/1/012008

Bonnin, R., (2017). *Machine Learning for Developers: Uplift your regular applications with the power of statistics, analytics, and machine learning*. Packt Publishing (First publish).

Butterworth, J.M., Davison, A.R., Salam, G.P., (2008). Jet Substructure as a New Higgs-Search Channel at the Large Hadron Collider. *Phys. Rev. Lett.,* 100,242001. doi.org/10.1103/PhysRevLett.200.24001

Browne, M.W. (2000). Cross-Validation Methods. *Journal of Mathematical Psychology*. 44-p 108-132. https://doi.org/10.1006/jmps.1999.1279.

Bruce, P., Bruce, A., Gedeck, P., (2020). *Practical Statistics for Data Sciences* (Nicole, T.). (Second Edition). O'Reilly Media.

Chatrchyan, S., et al. (CMS Collaboration) (2012). Observation of a New Boson at a Mass of 125 GeV with the CMS Experiment at the LHC. *Phys. Lett. B*, 716, 30--61. https://doi.org/10.1016/j.physletb.2012.08.021

Chen, T., & Guestrin, C. (2016). XGBoost: A Scalable Tree Boosting System. In Proceedings of the 22nd ACM SIGKDD. *International Conference on Knowledge Discovery and Data Mining*, (pp. 785–794). New York, NY, USA: ACM. https://doi.org/10.1145/2939672.2939785

CMS Collaboration. (2022). A portrait of the Higgs boson by the CMS experiment ten years after the discovery. *Nature*, 607, 60–68. https://doi.org/10.1038/s41586-022-04892-x

Cortes, C., Vapnik, V., (1995). Support-vector networks. *Machine Learning,* 20, 273–297. https://doi.org/10.1007/BF00994018

Flechl, M., (2015). Higgs physics: Review of recent results and prospects from ATLAS and CMS. *J. Phys. Conf. Ser.*, 631(1), 012028. https://doi.org/10.1088/1742-6596/631/1/012028

Fernow, R.C., (1983). Introduction to Experimental Particle Physics. *Cambridge University Press*. DOI: 10.1017/9781009290098.

Mucherino, A., Papajorgji, P.J., Pardalos, P.M. (2009). k-Nearest Neighbor Classification. In: Data Mining in Agriculture. *Springer Optimization and Its Applications, vol 34. Springer*, New York, NY. https://doi.org/10.1007/978-0-387-88615-2_4

Müller, A.C., Guido, S. (2016). *Introduction to Machine Learning with Python*. O'Reilly. ISBN: 9781449369897.

Nettleton, D. (2014). *Commercial Data Mining-Chapter 6 - Selection of Variables and Factor Derivation*. p 79-104. https://doi.org/10.1016/B978-0-12-416602-8.00006-6

Pedregosa, F., et al., (2011). Scikit-learn: Machine Learning in Python. *Journal of Machine Learning Research*. 12, p2825-2830. https://doi.org/10.48550/arXiv.1201.0490

Rao, A. S., Vardhan, B. V., and Shaik, H. (2021). Role of Exploratory Data Analysis in Data Science. *6th International Conference on Communication and Electronics Systems (ICCES),* Coimbatre, India, 2021, pp. 1457-1461. https://doi.org/10.1109/ICCES51350.2021.9488986

Schapire, R. E. (2013). Explaining AdaBoost. In: Schölkopf, B., Luo, Z., Vovk, V. (eds) *Empirical Inference. Springer*, Berlin, Heidelberg. https://doi.org/10.1007/978-3-642-41136-6_5

Scikit Learn. (2013a). *sklearn.preprocessing.LabelEncoder.* Sklearn. Retrived January 16 , 2023, https://scikit-learn.org/stable/modules/generated/sklearn.preprocessing.LabelEncoder.html.

Scikit Learn. (2013b). *sklearn.preprocessing.StandardScaler*.Sklearn. Retrived January 16, 2023, from https://scikit-learn.org/stable/modules/generated/sklearn.preprocessing.StandardScaler.html.

Scikit Learn. (2013c). *sklearn.model_selection.GridSearchCv*. Sklearn. Retrived January 16, 2023, https://scikit-learn.org/stable/modules/generated/sklearn.model_selection.GridSearchCV.html (Accesed: May 2023)

Scikit Learn. (2013d). *sklearn.model_selection.StratifiedKFold*. Sklearn. January 16 Retrived, 2023, https://scikit-learn.org/stable/modules/generated/sklearn.model_selection.StratifiedKFold.html

Tumasyan, A. a. (2022). Measurement of the inclusive and differential Higgs boson production cross sections in the decay mode to a pair of τ leptons in pp collisions at $\sqrt{s}$=13 TeV. *Phys.Rev.Lett*., 128, 081805. https://doi.org/10.1103/PhysRevLett.128.081805

Vinutha, H.P., Poornima, B., Sagar, B.M. (2018). Detection of Outliers Using Interquartile Range Technique from Intrusion Dataset. In: Satapathy, S., Tavares, J., Bhateja, V., Mohanty, J. (eds) Information and Decision Sciences. *Advances in Intelligent Systems and Computing*, vol 701. Springer, Singapore. https://doi.org/10.1007/978-981-10-7563-6_53