Journal of Critical Reviews

ISSN- 2394-5125 Vol 7, Issue 7, 2020

PREDICTION OF DECAY MODES OF HIGGS BOSON USING CLASSIFICATION ALGORITHMS

Mohit Baliyan[1*],Akshat Bandooni[2], Sharad[3], Akshansh[4], R. Viswanathan[5], B. Mallikarjuna[6] , T.Edison[7]

[1,2,3,4] **IMSEC, AKTU University, Ghaziabad,Uttar Pradesh, India, 201 015.**
[5, 6,7] **School of Computing Science & Engineering, Galgotias University, Greater Noida, Uttar Pradesh, India, 203 201.**

[1]mohitbaliyan98@gmail.com , [2]Akshatbandooni333@gmail.com , [3]skumarchaudhary12345@gmail.com , [4]akshanshsingh97@gmail.com, [5]rvnathan06@gmail.com, [6]basettymalli@gmail.com, [7]edison.t@galgotiasuniversity.edu.in

Received: 03.02.2020 *Revised: 07.03.2020* *Accepted: 16.04.2020*

Abstract.
Computational algorithms have been implied to various problems in particle physics from explosion of applications in particles to event identification and reconstruction. The motive of this study is to propose a model which predicts the decay events of High Boson as “tau-tau ecay of High Boson” or "the background noise" after applying some features extraction on the CERN dataset. Therefore, eight algorithms namely “K” Nearest Neighbor, Artificial Neural Networks, Naïve Bayes algorithm, Logistic Regression algorithm, SVM , Random Forest Decision Tree, and Gradient Boosting are used in this experiment. The performances of all the algorithms are examined basis accuracy and computational time. Derived results show decision tree outperforms the best on the measures like Accuracy and Computational Time.

Keywords: Higgs Boson, Classifiers, Decay Modes, CERN, GPU

DOI: http://dx.doi.org/10.31838/jcr.07.07.48

INTRODUCTION

European Nuclear Research identified a particle around 126 MEV considered to be coherent and much awaited Higgs Boson and Later in March 2013 the finding of a new particle was established [1]. This ATLAS experiment takes place at the CERN’s large Hadrons Collider and LHC accelerator. To boost the energy of the particles, it has 27 km magnets. To guide the accelerator ring, strong magnetic fields that are produced by the superconducting electromagnets are used. Beams inside the accelerator ring collide at the position of ATLAS, CMS, ALICE and LHCB particle detectors. ATLAS is a general purpose finder at the LHC. Beams of particles from The Large Hadrons Collider collide at its center Deflection of new particles from the collision point fly in all directions [2]. Paths, momentum and the energy of the particles are recorded by the ring detector subsystems that are arranged in layers. To record the worth events, ATLAS generates a huge amount of flow data and uses an advanced trigger system. The existence of Higgs Boson is also confirmed on the basis of collisions at the ATLAS [3]. The Higgs is one of the 17 particles which explain the all known basic particles. For all physical forces and giving particles their mass, the Higgs particle is a Boson is responsible. It is difficult to detect Higgs Boson, as it is too vast unlike other particles and does not last less longer and obeys the law of Conservation of Energy.

Higgs Boson bifurcates into other particles, due to its short life-span and decays in various forms of elementary particles that carry forces. A decay of Higgs Boson into specific particles is called channels. Three distinct decay channels were observed in Higgs Boson which were all boson pairs. The next decay of Higgs Boson has been noticed into the fermions pairs. One of the ATLAS experiments showed as particles.

Tau particles called fermions, that are accountable for making the matter and tau is a Fermion. Fermions behave like a massive electron and can be one or any composite particle such as the proton. They obey the Pauli Exclusion Principle and have half integer spin. Fermions acquire mass in the same process, how gauge bosons acquire the mass. That’s why Higgs Boson could decay to either bosons or fermions. But the strength of this signal is small, thus it is buried in the background noise. The replicated data is created by ATLAS after detecting the features which are accountable to feature the events. So, each event is classified in two categories, one is "tau-tau decay of Higgs Boson'' and other is background noise.

This Paper firstly performed data analysis on this CERN dataset basis analysis it preprocessed the data and then applied several classifiers on this data to categorize events as decay of Higgs Boson into tau particles or as background noise and rate the performance of each classifier. Then finally performed time analysis and compared the GPU based computation for each classifier over training and test data.

LITERATURE REVIEW

A remarkable role is played by Machine learning in processing large data at particle colliders. Energy physics gets the advantage of Machine learning at two levels; the online filtering of streaming detector measurements and offline analysis of pre-recorded data. Particle colliders may not produce particles of interest at times. To differentiate the particles, supervised machine learning has been used. According to Whiteson, stochastic optimization techniques result in substantially better analysis [4]. Artificial neural networks have been used to detect the decay of Higgs Boson to tau leptons, as deep neural network structures are able to auto-discover high level features from the data. Bruce Denby stated that Machine Learning enhances statistical power more than the common high-level features designed by the physicists [5]. Machine Learning approaches are mostly used for searching the rare particles produced during the collision. Standard Machine Learning approaches such as shallow Machine Learning models used to linear inputs. On the contrary, according to Baldi, Sadowski, &Whiteson to learn multifaceted non-linear inputs, artificial neural networks are able to distinguish the signals of decay better than the background noise [6].

Stacking machine learning algorithms performed worse than DNN though with a less calculated cost. Stacked classifiers in a MVA showed a significant improvement. Alves recommends MVA tools to accumulate simpler and faster machine learning algorithms performs better complex algorithms [7].

METHODOLOGY

Dataset:

The dataset is permanently released at CERN Open Data Portal [8]. The raw data has 818238 events in 195.5 MB. The dataset

Higgs Boson Training Data which is used in our experiment is downloaded from Kaggle [9]. It is composed of two files, training file and test file.

Overview of the design of the dataset is described in Table 1. As the label column is not present in the Test data file as a outcome Training data file is used in this experiment.

Table 1: Dataset Description

Observations	Training file	Test file
Events	250000	550000
ID Column	Yes	Yes
Feature Column	30	30
Weight	Yes	Yes
Label Column	Yes	Yes

Data Preprocessing

Value -999.000 symbolizes the non-computed value in the dataset. Number of non-computed values in the different feature columns is illustrated in Table 2.

Table 2: Number of Non-Computed Values

Column Name	Number of non-computed values
D-m-M	38114
D-d-j-j	177457
D-m-j-j	177457
D-p-j-j	177457
DER-l-e-c	177457
P-j-l-p	99913
P-j-l-e	99913
P-j-l-p	99913
P-j-s-p	177457
P-j-s-e	177457
P-j-s-p	177457

Most columns presented in Table 2 include more than the half number of entries in non-computed form. To deal with non-computed values, firstly estimate the correlation and the dependencies of each column to the other. Fig. 1.shows the correlation amongst these columns. As shown in Fig. 1. Variable (D-m-M) DER_mass_MMC does not highly correlate with any other variable and the number of non-computed values in this variable is less than half of its total entries. We didn't drop the column of this variable at the cost of information loss therefore, we replaced its non-computed values by the mean of this column.

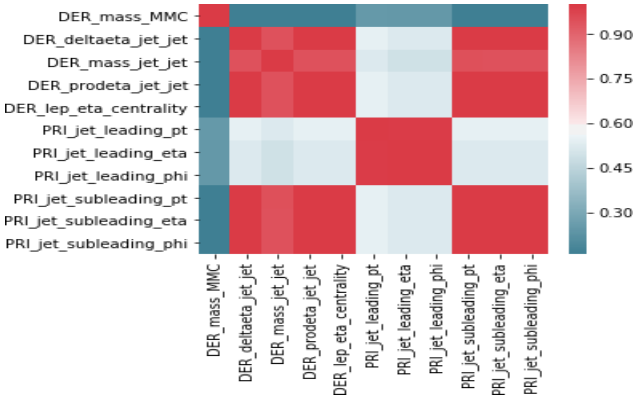

Figure 1: Correlation between these columns

(P-j-l-p) PRI_jet_leading_phi and (P=j-l-e) PRI_jet_leading_eta variables are extremely concurrent with PRI_jet_leading_pt and have non-computed entries in more than half no of rows so we dropped these two columns to avoid uncertainty during training and keep PRI_jet_leading_pt to avoid information loss.

Variables DER_lep_eta_centrality, D-m-j-j, D-p-j-j, PRI_jet_subleading_eta, P-j-s-p and DER_deltaeta_jet_jet extremely concurrent is to P-j-s-p and include more than half entries in non-computed form so we reduced them and keep PRI_jet_subleading_pt to keep the features information of dropped columns, wherein minimum value and maximum value in that feature vector.

The variables prefixed P-j-l-p, P-j-l-e, P-j-l-p, P-j-s-p, P-j-s-e are PRI primitives about the bunch collision as concluded by the detector, fundamentally the impetuses of atoms.
The derived variables prefixed D-m-M, D-d-j-j, D-m-j-j, D-p-j-j, D-l-e-c are DER primitive quantities. Fig 2 showed the total number of counts of "tau-tau decay of Higgs Boson" and "Background noise" in the target variable for each event where "s" refers the signal of decay and "b" refers the background noise.

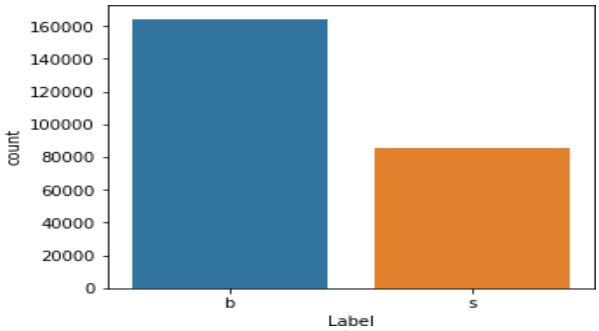

Figure 2: Total Count

There is an unstable allocation in event classification as we have a greater number of events which are categorized as "background noise" in comparison to "tau-tau decay of Higgs Boson" and label column is the target variable which is conditional. It has entries in two categories 's' for 'tau-tau decay of Higgs Boson' and 'b' for 'background noise'. Thus, we programmed the label column in the form of 0 and 1 to make it unconditional.

Arranges like logistic deterioration and neural networks work well on standardized inputs. In case of logistic deterioration and neural networks, it guarantees the constancy in convergence of weights and biasness during optimization of cost function. For k-nearest neighbor, it has two aspects, one is to use the Euclidean distance between two entries of different range to make our classifier uninformative and according to second characteristic if we didn't normalize the inputs for k-nearest neighbor, all nearest neighbors are aligned near the same axis that has smaller range and leads to incorrect categorization. But in our experiment K-nearest neighbor works better on second aspects, so we feed the normalized data into k-nearest neighbor classifier. We used a Min-Maxscaler to normalize our data whose value for any entry is given by equation 1.

$$x^{l} = (x - \min)/(\max - \min) \quad (1)$$

After data preprocessing, we have 24 features left for characterizing the events as "tau-tau decay of Higgs Boson" or "background noise" in our dataset. Feature vector removed after preprocessing refer with Table 3.

Table 3: Characteristics of Feature columns

Column Name	Characteristics
D-m-M	The projected mass of the Hiigs boson mH
D-d-j-j	A probabilistic of undefined if PRI$_j et_n um$=1.
D-m-j-j	To computes the jets undefined if PRI$_j et_n um$=1.
D-p-j-j	The product of two pseudorapidities jets undefined if PRI$_j et_n um$=1
D-l-e-c	The centrality of the pseudorapidity of two jets undefined if PRI$_j et_n um$=1
P-j-l-p	The transverse momentum undefined leading jet if PRI$_j et_n um$= 0.
P-j-l-e	The transverse pseudorapidity of the leading jetundefined if PRI jet num = 0.
P-j-l-p	The angle of leading jet undefined if PRI jet num = 0.
P-j-s-p	The second largest transverse momentum undefined if PRI$_j et_n um$1.
P-j-s-e	The subleading of pseudorapidity of jet undefined if PRI$_j et_n um$=1.
P-j-s-p	The subleading jet undefined if PRI$_j et_n um$=1.

Methods Used

Classifiers are the models which learn from the data to map the input pattern to a specific class. We use various classifiers to evaluate the performance of each classifier on our data which are given below:

K-Nearest Neighbor Classification (KNN)is one of the simplest and fundamental classifiers and performs exceptionally well when there is no prior or little information of data. K-nearest-neighbor classification difficult to determine and perform discriminant analys[10].

K-nearest neighborsmake the prediction by using the training data directly, so there is no need for learning. To make a prediction on any instance, it first computes the K number of nearest neighbors to that instance. Then it assigned the class to that instance which is the most common class assigned to their neighbors. The optimal choice to determine the value of K is to evaluate the performance of the model on various values of K before settling the value on one. It calculates Euclid's distance on the basis of all attributes of the instances. Sometimes not all of the attributes do not contribute to the classification of the instances. That's why it misleads the results in these cases.

Support Vector Machine

uses classical learning approaches which follow the empirical risk minimization (ERM) principal. According to this principal, learning is based on minimizing the error on the training dataset [11]. The SVM follows structural risk minimization (SRM) principle obtain better generalization abilities to the SVM [12]. According to this principal it minimizes error.

SVM can handle very large feature spaces because vectors does not perform [13]. The SVM perform task is given by Cristianini, Shawe-Taylor [14] as in equation 2

$$y_i f(x_i) = y_i(w^T x_i + b) >= 1 \quad (2)$$

If there is an M number of input samples, then w is a vector of M-dimension whereas b is scalar. The vectors w and scalar b vector are rummage-sale to define thir position of unraveling hyperplanes. f(x) is a separating hyperplane which is used to classify the input data. It performs better over other classifiers if we have unbalanced data or we have a huge difference in the number of positive and negative events. It represents a model using few examples.

Naive Bayes classifier works on a very simplified Bayesian probability model [15]. In the Bayesian probability model, the probability model for n number of attributes in a dataset, the naive Bayes classifier had made 2n! independent assumption. The probability that an input sample x with attributes vector < $x_1, x_2, .., x_n$ > belongs to class y1 is given by Bayes theorem as in equation 3.

$$P(y_1 |x) = P(x|y_1)P(y_1)/P(x) \quad (3)$$

Where the $P(y_1|x)$ is the posterior probability and $P(y_1)$ is prior probability. If this conditional independent assumption holds true, then naive Bayes classifiers performs well and learns rapidly in various supervised and classification problems. It is computationally fast and performs well on high dimensional data [16].

Artificial Neural Network
is the simulation of real-world neural networks found in animal brains. Scientists are approaching the work and capabilities of neurons found in the animal brain. Neural networks have the capabilities of learning based on the result from experience or training.

The artificial neural networks look like connected array of exceptional basic processor named as nurons. Multi-layer perceptron is the consist of an input layer, one or more number of hidden layers. Except for the set input layer, each neurons receives the signals from previous neurons. The neuron will later produces sigmoidal function and output function as defined by Sobajic & Pao [17].

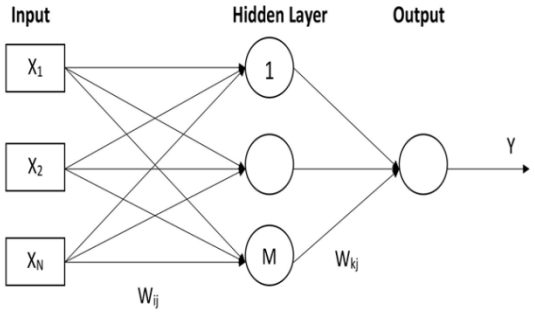

Figure 3: Multi-Layer Neural Network

ANN faces an issue of over fitting the data and sometimes performs poorly on new data. To overcome this, cross validation can be utilized for the early stopping of gradient descent learning or introducing regularization in the hypothetical function and loss function. Fig 3 represents the multi neural network containing input, one of the hidden and an output layers.

Logistic Regression
contains a dummy variable which is the dependent variable. It dependents on independent varible, Logistic regression is constant as illustrated by Midi, Sarkar, & Rana [18]. So a hypothetical function for logistic regression is slightly modified by applying an activation function on it which is given in equation 4 and 5. As to convert

$$z = wx + b \quad (4)$$

$$h(x) = f(z) \quad (5)$$

here w is weights, x is input, b is bias, h(x) is the hypothetical function for logistic regression and f is the activation function. We choose activation functions according to the range of our dichotomous dependent variable. Unlike linear regression, binary logistic regression uses binary cross entropy function for calculating the loss as we have discrete values in our output, mean squared errors cannot be used to calculate the loss. Due to its linear decision surface and limited hypothesis space. It can't solve nonlinear problems.

Decision Tree
classifier may follow any approach like optimal and look-up rule, multistage decision making. However, the principle idea behind decision trees is determine the best decision using a greedy approach to be on the safe side [19]. It follows the tree-like structures in which each internal nodes represents a test on an attributes where the branch represents the result of the test. A Decision taken for a class label taken after computing all attributes represented by a leaf node. The criterion for taking decision on each internal node may be based on Gini index or Information Gain. As for categorical attributes can use information gain and if attributes are continuous, can utilize Gini index. Entropy is given by equation 6. To avoid overfitting in decision trees, the growth of the tree can be stopped earlier, before it classifies the training data perfectly or we can also prune the tree even without stopping the growth of the tree earlier.

$$H(x) = -p(x)\log(p(x)) \quad (6)$$

Gradient Boosting
is an ensemble of various models which help in reducing the factors like bias and variance during predictions. It uses as prediction model and which is produced in the form of an ensemble of weak prediction model. It usually converts weak learner into a strong learner. As in case of Gradient Boosting, it first trains a tree and increase the weight of that observation which is difficult to classify and decrease weights of tree. This process of growing trees repeats for a specified number of iterations [20].

Random Forest
is based on the Decision Tree algorithm. As in a decision tree, the result data set is split as different as it can be from each other into tree branches and tries to bunch up as a similar result data set it can be. In Decision Tree there's only one tree but in Random Forest Tree there are groups of decision trees to make a forest which are uncorrelated to each other but will perform the same operation on individual models, it increases the probability of accurate result. Random forest can perform comprehensive calculations. When the number of observations are high with any number of variables, random forest generates multiple decision tree models with diverse samples and initial variables. It is magnificent in accuracy. Random forest can be used for future calculation on different data sets to obtain results. For the information regarding relation between the variables and classification computed by the prototypes. Random forest comparatively from other machine learning algorithms is less impacted by noise. It is more reliable even if a new data point is added into the data set, it will not affect the overall algorithm since the new data may impact one or two trees but it is very unlikely for it to impact all the trees. Random forest can handle missing values itself [21].

EXPERIMENTS AND RESULTS

System Specifications of the machine on which the model training and testing was performed on the GPU with the following specifications given in Table 4:

Table 4: GPU Specification

Configuration of the processor	Intel I3 Processor
Architecture	x86_64
Processor	72
CPU MHz	1000.713
Processor max MHz	3700.0000
Processor min MHz	1000.0000
RAM	128 GB
GPU	Nvidia Quadro P5000
GPU Memory	16 GB

Training Configuration

constitutes trained data which is on 2,40,000 classifiers events out of 2,50,000 and it is tested on 10,000 events. For k-nearest neighbor classifiers, as it is a lazy learning algorithm. It learns only during prediction. At the small values of k (neighbors), the model has a variance problem as it performs well on the training set as well as it did not do well on the test sets. During large k it has a bias problem as it did not even fit the training set well. When the value of K is equal to 7, it gave the best result [22]. In the case of SVM, the RBF used as kernel function as well as used scale coefficient for kernel functions which defines the influence of a single input pattern. As dealing with continuous data it can be concluded that features of events are distributed according to the Normal or Gaussian distribution. So Gaussian naive Bayes classifiers can be used for the experiment [23].

In the case of neural networks, a rectified linear unit activation function is used in the hidden layer. The reason behind using the rectified linear unit activation function in the hidden layer is to let the gradient be non-zero and recoverable during training. Sigmoid activation functions from the output layer neuron because we have to predict the probability as output. Stochastic gradient descent algorithms are used to learn the weight of neurons in which we set its batch size to 32. Generally, it is noisier than simple gradient descent because it takes a higher number of iterations to reach the global minima. But it is computationally less expensive than simple gradient descent. It does not feed all training examples in once to calculate the cost gradient on every iteration. So, in the case of large datasets as in this paper, it is better to use stochastic gradient descent for efficient optimization. The learning rate is initialized with 0.02 with a decay of e^{-6} to update the learning rate in each epoch. Several experiments were performed by changing neurons in the feed-forward layers and evaluating their performances on the training sets. But the best results were achieved when the numbers of neurons in the feed-forward layers were chosen same as like in the input layer.

In logistic regression when it started decreasing regularization strength by increasing the value of inverse regularization strength C, the classifier performs well. At C is equal to 200 it gave the best result on training and test set.

As in case of Decision tree and random forest classifier, this dataset has continuous data in independent variables Gini criterion is utilized for splitting the attributes at each internal node. This paper determines that Decision tree classifier, Gradient Boosting and Random Forest performed exceptionally well, as it achieved 100% accuracy on training as well as test set. Performance on the train and test set of all classifiers is listed in Table 5.

Table 5: Performance Measure

Classifier	Training accuracy	Test accuracy
K-Nearest Neighbor	92.58%	90.57%
Support Vector Machine	97.05%	96.51%
Naive Bayes	99.28%	99.26%
Artificial Neural Network	99.83%	99.81%
Logistic Regression	99.99%	100%
Decision Tree	100%	100%
Gradient Boosting	100%	100%
Random Forest	100%	100%

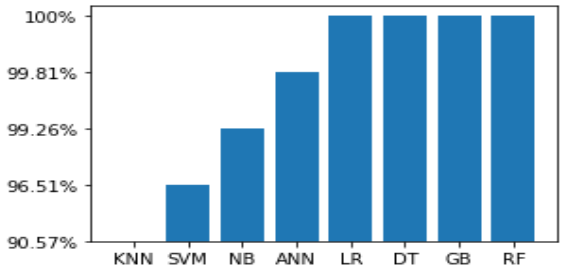

Figure 4: Test accuracy

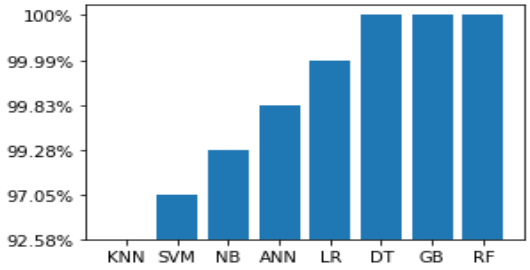

Figure 5: Training accuracy

Fig. 5. and Fig. 4. show performances of all classifiers on the training set and test set.

Time Analysis

In this experiment time analysis is performed on GPU based computation. Each classifier algorithm 5 times on training and test datas and calculated the average in value. The computation time taken by GPU over training data and test data for each classifier are listed in Table 6 and Fig. 6 and Fig. 7 show the graphical comparison of time analysis for all classifiers. In case of K-nearest neighbor, its training time is very less than the testing time due to its lazy algorithm-based principal, as it learns only during predictions. Naive Bayes classifier showed us the most optimizing behavior in contrast to training, but it took more time to predict than that of logistic regression and decision trees. logistic regression after well tuning of hyperparameters gave us impressive results during predictions, as its testing time is lowest in all classifiers. Time analysis showed that naive Bayes classifiers performed well on training as well as on test data for the time point of view.

Table 6: GPU based computation

Classifier	Training time	Testing time
K-Nearest neighbor	3.225s	7.964s
Support Vector Machines	1008.156s	9.718s
Naive Bayes	0.131s	0.006s
Artificial Neural Network	3589.567s	0.268s
Logistic Regression	8.456s	0.001s
Decision Tree	0.554s	0.003s
Gradient Boosting	20.164s	0.011s
Random Forest	4.760s	0.011s

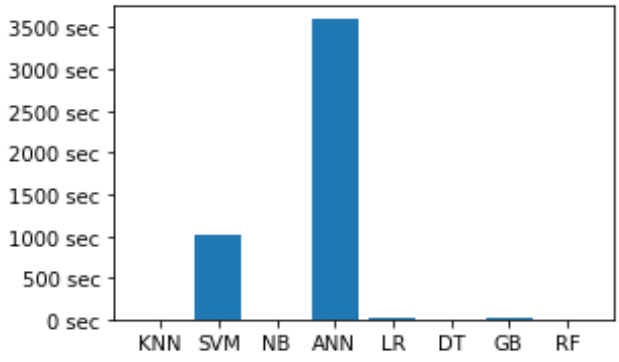

Figure 6: Training Time

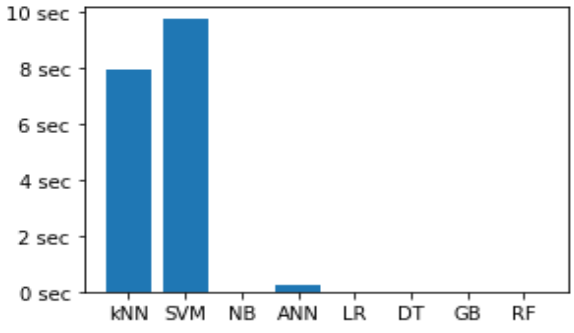

Figure 7: Testing Time

CONCLUSION

Machine learning has led to significant advances in many fields such as data analysis, web search, photo tagging and spam detector, etc. They are clearly useful in a wide range of application, including a large host of application in the natural science. The problems in high-energy physics are particularly made suitable for machine learning, by having large data set with complicated underlying structure in it. Results shows machine learning can be used for providing a powerful means as well as the best practical approach to analyzing particles collider data. On this specific basis of results, it clearly specifies that the most accurate classifiers are Logistic Regression, Gradient Boosting and Random Forest but there are some drawbacks when it comes to time analysis of the above classifiers. Turns out Naïve Bayes is the most optimized of them all but it took more time than Logistic Regression and Decision Tree. Therefore, we can say that Decision Tree is the most efficient classifier when it comes to inspecting such large data sets with accuracy and computational time.

Future Scope

Based on these results, maybe these algorithms might not perform well in general but assembling several algorithms can be performed to produce better results. In this case, Random Forest and Gradient Boosting were better than other algorithms but this was on pre-recorded data. However, we can generate more potentially powerful algorithms by ensemble technique which can be used to analyze the data and will help in online filtering of streaming detector measurements.

REFERENCES

1. New results indicate that new particle is a higgs boson. (2013, March). Retrieved 2013-03-14, from https://home.cern/news/news/physics/new-results-indicate-new-particle-higgs-boson
2. The large hadron collider. (n.d.). Retrieved from https://home.cern/science/accelerators/large-hadron-collider
3. Atlas. (n.d.). Retrieved from https://home.cern/science/experiments/atlas
4. Whiteson, S., Whiteson, D. (2009). Machine learning for event selection in high energy physics. Engineering Applications of Artificial Intelligence, 22(8), 1203–1217.
5. Denby, B. (1999). Neural networks in high energy physics: a ten-year perspective. Computer Physics Communications, 119(2-3), 219–231.
6. Baldi, P., Sadowski, P., Whiteson, D. (2014). Searching for exotic particles in high-energy physics with deep learning. Nature communications, 5, 4308.
7. Alves, A. (2017). Stacking machine learning classifiers to identify higgs bosons at the lhc. Journal of Instrumentation, 12(05), T05005.
8. Cern open data portal. (2014). Retrieved from http://opendata.cern.ch/
9. Higgs boson training data. (2014). Retrieved from https://www.kaggle.com/c/higgs-boson/data
10. Peterson, L. E. (2009). K-nearest neighbor. Scholarpedia, 4(2), 1883.
11. Boser, B. E., Guyon, I. M., Vapnik, V. N. (1992). A training algorithm for optimal margin classifiers. In Proceedings of the fifth annual workshop on computational learning theory (pp. 144–152).
12. Vapnik, V. (2013). The nature of statistical learning theory. Springer science & business media.
13. Cortes, C., Vapnik, V. (1995). Support-vector networks. Machine learning, 20(3), 273–297.
14. Cristianini, N., Shawe-Taylor, J., et al. (2000). An introduction to support vector machines and other kernel-based learning methods. Cambridge university press.
15. Wafa'S, A.-S., Naoum, R. (2009). Development of genetic-based machine learning for network intrusion detection. World Academy of Science, Engineering and Technology, 55, 20–24.
16. Dimitoglou, G., Adams, J.A., Jim,C.M. (2012). Comparison of the c4.5 and a naïve bayes classifier for the prediction of lung cancer survivability. arXiv preprint arXiv:1206.1121.
17. Sobajic, D. J., Pao, Y.-H. (1989). Artificial neural-net based dynamic security assessment for electric power systems. IEEE Transactions on Power Systems, 4(1), 220–228.
18. Midi, H., Sarkar, S. K., Rana, S. (2010). Collinearity diagnostics of binary logistic regression model. Journal of Interdisciplinary Mathematics, 13(3), 253–267.
19. Dattatreya, G., Kanal, L.N. (1985). Decision trees in pattern recognition. University of Maryland. ComputerScience.
20. Mallikarjuna, B., & Reddy, D. A. K. (2019). Healthcare Application Development in Mobile and Cloud Environments. In Internet of Things and Personalized Healthcare Systems (pp. 93-103). Springer, Singapore.
21. Mallikarjuna, B. (2020). Feedback-Based Resource Utilization for Smart Home Automation in Fog Assistance IoT-Based Cloud. International Journal of Fog Computing (IJFC), 3(1), 41-63.
22. Mallikarjuna, B. (2020). Feedback-Based Fuzzy Resource Management in IoT-Based-Cloud. International Journal of Fog Computing (IJFC), 3(1), 1-21.
23. Mallikarjuna, B., Reddy, D. A. K., & Sailaja, G. (2018). Enhancement of Railway Reservation System Using Internet of Things. In Proceedings of the 1st International Conference Universal Computing, Communication in Data Engineering (CCODE-2018).