Searching for exotic particles in 4 bottom quarks-antiquarks final states with machine learning techniques at the LHC

Punnawich Chokeprasert and Chayanit Asawatangtrakuldee *

Department of Physics, Faculty of Science, Chulalongkorn University, Bangkok 10330, Thailand

*Corresponding author's e-mail: Chayanit.Asawatangtrakuldee@cern.ch

Abstract. The standard model (SM) has been a highly successful theory in explaining fundamental particles and their interactions among themselves. However, the SM has not yet explained several phenomena, and many beyond the standard model (BSM) have been introduced to solve these unexplained phenomena. One example is the bulk Randall-Sundrum (RS) model, which proposed a new higher dimensional mechanism for solving the hierarchy problem and predicted the existence of a hypothetical particle, bulk graviton. In this study, we investigate supervised machine learning methods to search for the bulk graviton decays into a pair of the SM Higgs bosons, and each Higgs boson decays into a pair of bottom anti-bottom quarks ($G^*_{KK}\rightarrow hh\rightarrow b\bar{b}b\bar{b}$). We train machine learning models to classify events between $G^*_{KK}\rightarrow hh\rightarrow b\bar{b}b\bar{b}$ (signal) and QCD 4b multi-jet (background) processes. The evaluation metrics are calculated in the highest score to compare the classification efficiency between Adaptive Boosting and Neural Networks algorithms after performing feature importance and hyperparameter tuning techniques to optimize the models. The results show that the Neural Networks better classify our signal versus background events with the AUC score of 0.9836, compared to the Adaptive Boosting model of 0.9741. Furthermore, the signal significance is also predicted and scaled to the integrated luminosities of Run 2, Run 3 and HL-LHC, data-taking periods of the LHC. The predictions are obtained at 1.952, 2.858 and 9.037 for the Neural Networks and at 1.968, 2.881 and 9.111 for the Adaptive Boosting.

1. Introduction

Many beyond the Standard Model (BSM) theories have been introduced to address several phenomena not explained by the Standard Model (SM), for example, hierarchy problems regarding the particle masses, neutrino oscillation, dark matter and dark energy, etc. An example of our interest is the bulk Randall-Sundrum (RS) [1], which predicts a rich spectrum of excited states of the graviton, the hypothetical particle responsible for mediating the gravitational force. In this model, the gravitational force is modified by an extra dimension, leading to a warped geometry that can explain the relatively weaker strength of gravity compared to the other fundamental forces. The RS model predicts a spin-2 Kaluza-Klein graviton G^*_{KK}, a particle that carries gravitational force and has different energy states in the extra dimension. Throughout this analysis, a particularly interesting aspect in connection to the $G^*_{KK}\rightarrow hh\rightarrow b\bar{b}b\bar{b}$ [2] the final state is an enhancement of the production cross-section of the graviton decays to the two Higgs bosons, corresponding to the SM mass of 125 GeV, and each Higgs boson decays to two b-quarks. The analysis focuses on the resolved channel, in which each of the four b-quarks

from the Higgs bosons leads to an individually reconstructed jet, a spray of quarks and gluons produced in high energy particle collisions.

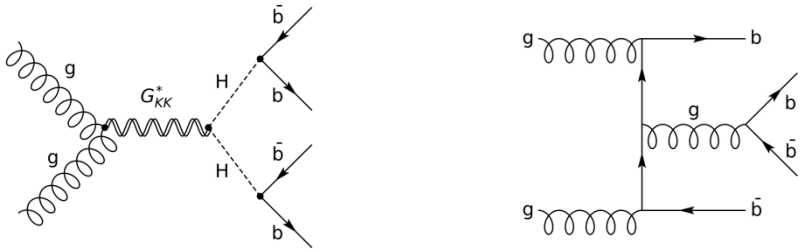
The Large Hadron Collider (LHC), the world's largest and most powerful particle accelerator, of CERN, where two beams of hadron (either protons or lead ions) collide to create a massive amount of energy, up to 13 TeV, to study the origin of the Universe. Based on the latest experimental study by the CMS experiment at the LHC, a similar search was performed using the traditional cut-based analysis method with 2016-2018 data at 13 TeV, with respect to an integrated luminosity of 138 fb^{-1}. The results show no significant excess of new exotic particles.

Therefore, in this study, we are interested in developing an advanced analysis method to perform multivariate data analysis using a more sophisticated tool, namely machine learning techniques. This method is the most effective way of dealing with high dimensional data. We mainly study supervised machine learning models to classify signal and background events in the search for bulk graviton with a mass of 1200 GeV produced from proton-proton collisions at the LHC. Nonetheless, the mass of the bulk graviton is ambiguous, and we expect the trained models to be applied to any masses.

2. Methodology

Simulated data (signal and background) are generated by MadGraph5_aMC@NLO 3.4.0 generator [3] to mimic hard scattering processes, occurring from proton-proton collisions at a center-of-mass energy of 13 TeV at the LHC, and supplemented with parton showering and hadronization. Particles from the collisions are then detected by the CMS detector using Delphes 3.5.0 [4], a fast detector response simulation framework.

The characteristic signature of the signal process is the final state of two pairs of highly energetic b-tagged jets, which are produced from the decay of the two Higgs bosons, initiated from the decay products of the bulk graviton as shown in figure 1 (left). Background contributions arise from the SM processes, dominated by QCD 4b multi-jet production [5]. This process leaves a similar signature of four b-tagged jets in the final state as the signal process, presented in figure 1 (right). The b-tagged jet is a cluster of reconstructed particles identified to come from bottom-quarks hadrons.



(a) (b)

Figure 1. (a) Feynman diagram of the signal process ($G^*_{KK} \rightarrow hh \rightarrow b\bar{b}b\bar{b}$) and (b) background process (QCD 4b multi-jet)

After simulating MadGraph-Delphes data, at least 4 jets and two of them are b-tagged required for each event. Each jet should also have transverse momentum (p_T) ≥ 30 GeV and pseudo-rapidity ($|\eta|$) < 2.4, the region where b-tagging is highly possible. Before training the models, we add labels of 0 and 1 as target variables to all simulated events belonging to background and signal processes, respectively. The dataset contains 69 input features divided into 23 low-level and 46 high-level features. The low-level features represent physics variables directly measured by the CMS detector. The high-level features, on the other hand, are calculated from the low-level features using four-momentum vector to reconstruct parent particles such as the Higgs bosons and the bulk graviton. Furthermore, we selected 50 input features by scrutinizing characteristic differences between signal and background processes using histograms for a higher possibility of being classified more efficiently by machine learning.

The overall dataset is then separated into 80% training and 20% testing datasets. We balance a proportion of 0 (background) and 1 (signal) labeled events in the training dataset using the synthetic minority oversampling technique (SMOTE) [6]. This method uses interpolation to avoid the risk of majority-biased prediction in our machine learning models and creates synthetic data points between the two nearest data points joined by a straight line. The training data is further split into 80% training and 20% validation datasets to validate the models and prevent bias before employing the models in the testing dataset. In this study, we are interested in two types of algorithms: Tree-Based Adaptive Boosting [7], and Artificial Neural Network algorithms [8]. Adaptive Boosting is a boosting method that sequentially builds a weak decision tree, the most basic model of Tree-Based algorithms that can be a base estimator for more complex ensemble methods and tries to correct the net error of the preceding tree. The Neural Networks model, on the other hand, resembles the human brain based on non-linear activation functions. It consists of node layers, including input, multiple hidden, and output layers. Each neuron connects to another neuron in the next layer and is assigned weights and biases. The well-known non-linear activation functions include rectified linear units (ReLU) and sigmoid.

In addition, each input feature passes Gaussian standardization [9] with a mean value of 0 and a variance of 1, which is applied before the training. Finally, we build and train the two models using Scikit-learn [10] and TensorFlow [11] libraries in Python. The feature importance method is applied to reduce computational time and disk space while improving the predictive performance of both models. In particular, the permutation importance technique [12] is performed for Neural Networks by randomly shuffling the values of one feature at a time and measuring a decrease of model's accuracy accordingly. This process is repeated for all features and the feature importance is ranked by their impact on accuracy values. On the other hand, the impurity-based feature importance [13] measures the reduction in the impurity of Adaptive Boosting by splitting each feature. Impurity refers to the degree of uncertainty in the classification in a decision tree node. The importance of given feature is then calculated as the total impurity reduction achieved by using the feature to split the data.

The models are also optimized through hyperparameter tuning [14]. For Neural Networks, hidden layers, hidden nodes, dropout rate and learning rate are looped through several selected values to search for the best values which maximize model's accuracy and AUC (area under Receiver Operating Characteristic (ROC) curves) scores [15]. The results show that a multi-layer perceptron with 2 hidden layers, each layer contains 100 hidden nodes, and 0.1 dropout rate and 0.001 learning rate in optimizer is the best set of hyperparameters. ReLU is selected as activation function for hidden layers and sigmoid function is applied to output layer to provide output score in terms of probability. Regarding Adaptive Boosting model, a maximum depth of base estimator of 200 and learning rate of 1.0 are selected to guarantee the best performance.

Lastly, the evaluation metrics, including precision, recall, F1 score, and AUC scores, are calculated based on the two models' classification predictions from the testing dataset. Precision is the fraction of correctly identified signal events among all events identified as signals, recall is the fraction of correctly identified signal events among all actual signals, and the F1 score is the harmonic mean of precision and recall. Additionally, the signal significance is also predicted and scaled to the integrated luminosities of Run 2, Run 3 and HL-LHC, data-taking periods of the LHC.

3. Results

Table 1 shows the top-ranked important features for Neural Networks and Adaptive Boosting models, resulting in different orders. The rank includes low-level features, such as number of jets (**njets**), number of b-tagged jets (**nbjets**), scalar sum of transverse momentum of all jets (**ht**), invariant mass of the first and second leading jets ordered by p_T (**mj1, mj2**), transverse momentum of the first, second, third, and fourth jets (**ptj1, ptj2, ptj3, ptj4**), pseudo-rapidity of the fourth jet (**etaj4**), and a binary possibility of b-tagged jets of the first and third jets (**btagj1, btagj3**). There are also some high-level features in the rank which are di-jet invariant mass (**m12, m13, m23**), and transverse momentum of the di-jet (**pt12, pt13, pt14, pt23**). The variables of di-jet are calculated from a pair of jets, assuming they are the decay products of the Higgs boson. Additionally, the delta-R (dR) or difference in the azimuthal angle (ϕ) and

the pseudo-rapidity between the two leading jets is also used (**dR13, dR14, dR23, dR24, dR34**). Lastly, there is the transverse momentum of the four jets (**pt1234**), which is assumed to be a feature of the bulk graviton.

Table 1. The top-ranked input features from Neural Networks versus Adaptive Boosting models.

Model	Number of input features	Input features (ordered)
Neural Networks	19	ht, mj1, dR23, pt14, mj2, nbjets, dR13, ptj2, m12, dR14, m23, m1234, ptj1, btagj3, ptj3, pt13, dR24, pt1234, pt23
Adaptive Boosting	19	njets, ht, mj1, m23, m12, mj2, m14, pt13, nbjets, m13, ptj3, dR24, btagj1, etaj1, pt12, ptj1, ptj4, dR34, pt1234

Comparison of the AUC scores from the two models considering the ranked input features on both training and testing datasets is shown in figure 2. In this study, we first calculate and compare the AUC scores for the two models by using the default threshold value of 0.5 (0.0) for Neural Networks (Adaptive Boosting), and the scores can obtain good predictions. Table 2 also indicates that Neural Networks yield the AUC score of 0.9836, while the Adaptive Boosting provides the AUC score of 0.9741.

Additional performance metrics including precision, recall and F1 score are further evaluated at various threshold values to seek for the optimal value of each model. The results from the highest F1 score show that the cut value of 0.3794 is optimal for the Neural Networks model, while the Adaptive Boosting is optimal at the threshold of -0.0001 which is very close to the default value. In both cases, we found no significant difference of the AUC scores with the default threshold values.

Moreover, the signal significance is also predicted and scaled to the integrated luminosities of Run 2, Run 3, and HL-LHC of the LHC, as listed in table 2. By assuming a simple counting experiment and applying the best-cut value of 0.3794 for Neural Networks and -0.0001 for Adaptive Boosting, signal significances yield at 1.952, 2.858, and 9.037 for the Neural Networks and at 1.968, 2.881, and 9.111 for the Adaptive Boosting. Figure 3 compares signal efficiency (blue), background efficiency (red) and signal significances as a function of the cut value of Neural Networks (a) and Adaptive Boosting (b) classifiers. The primary y-axis provides the efficiency values, while the secondary y-axis takes care of the signal significance. Note that the ranges of the cut value on x-axis are different among the two models. The cut values of Adaptive Boosting mainly gather around the default value of 0.0, while Neural Networks distribute along the value of 0 to 1. This is because the Adaptive Boosting uses a decision function which is the weighted sum of the predictions from all weak classifiers to produce output score or weight for each event. Therefore, the model is less confident in its predictions than the Neural Networks.

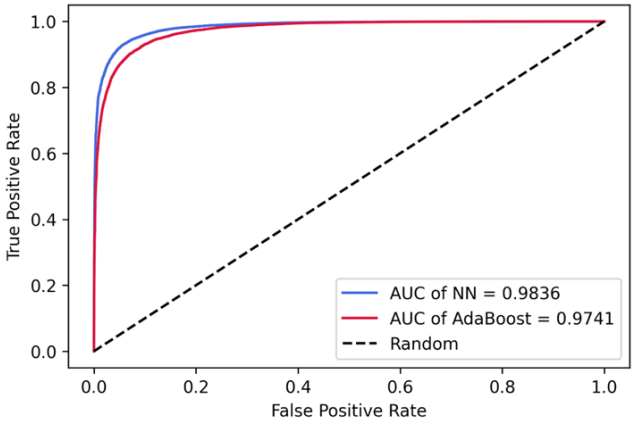
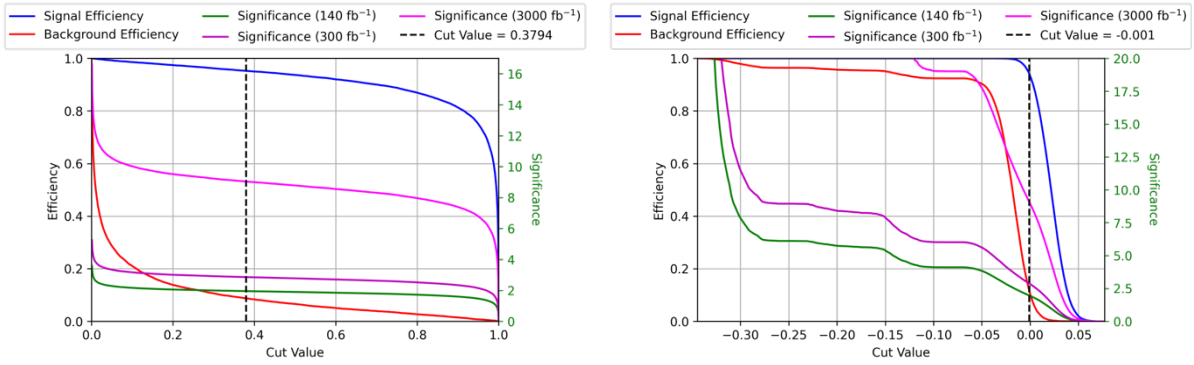


Figure 2. ROC curve and AUC of Neural Networks (blue) and Adaptive Boosting (red)

Table 2. Comparisons of the AUC scores and signal significances for the integrated luminosities of Run 2, Run 3 and HL-LHC of the LHC on the two models when predicting on the testing dataset.

Model	AUC	Signal significance		
		Run 2	Run 3	HL-LHC
Neural Networks	0.9836	1.952	2.858	9.037
Adaptive Boosting	0.9741	1.968	2.881	9.111



(a) (b)

Figure 3. Signal efficiency (blue), Background efficiency (red) and Signal significances for integrated luminosities of Run 2 (green), Run 3 (purple), and HL-LHC (magenta) as a function of the cut value of Neural Networks (a) and Adaptive Boosting (b) classifiers. The dashed vertical black line represents the best-cut value of each model.

4. Summary and discussion

The results show Neural Networks and Adaptive Boosting models perform comparably well on our signal versus background classification. This is because both models can optimize their model parameters by learning from the error of prior predictions. The predictions of signal significance simply provide the degree of confidence in observing our signal in the simple counting experiment. Note that the uncertainties are not yet included and would remarkably affect the signal significances and final sensitivity. Subsequently, the classifiers generated by each model can be used to predict events from actual collision data and compare them with well-estimated background events to search for possible excess of the signals. More effectively, these models can be adapted for statistical analyses searching for the bulk graviton in a wide mass range and different decay channels of the Higgs boson. For example, the models can be trained for bulk graviton signals in the mass range of a few hundreds to thousand GeV, given they have functions for automatically monitoring and setting the parameters to the best performance. Then, the output classifiers are used to perform a maximum likelihood fit with the real data and search for any potential excess above the SM predictions. A selection of which algorithm will be based on the analysis's preference and the sensitivity of the search. According to their structures, the Neural Networks algorithm will benefit for analyses with high similarity of background to signal events. While Adaptive Boosting is simpler and requires fewer computational resources.

Acknowledgements

This research has received funding support from the NSRF via the Program Management Unit for Human Resources & Institutional Development, Research and Innovation [**B37G660013**] and financially supported by Sci-Super IX fund from Faculty of Science, Chulalongkorn University.

References

[1] Randall L and Sundrum R 1999 Large mass hierarchy from a small extra dimension *Phys. Rev. Lett.* **83** 3370–73
[2] CMS Collaboration 2018 Search for resonant pair production of Higgs bosons decaying to bottom quark-antiquark pairs in proton-proton collisions at 13 TeV *J. High Energy Phys.* **08** 1–38
[3] Alwall J *et al* 2014 The automated computation of tree-level and next-to-leading order differential cross sections, and their matching to parton shower simulations *J. High Energy Phys.* **07** 1–157
[4] Favereau J D *et al* 2014 DELPHES 3: a modular framework for fast simulation of a generic collider experiment *J. High Energy Phys.* **02** 1–26
[5] Behr J K *et al* 2016 Boosting Higgs pair production in the $b\bar{b}b\bar{b}$ final state with multivariate techniques *Eur. Phys. J. C* **76** 1–31
[6] Chawla N V *et al* 2002 SMOTE: synthetic minority over-sampling technique *J. Artif. Intell. Res.* **16** 321–57
[7] Chengsheng T *et al* 2017 AdaBoost typical Algorithm and its application research *MATEC Web Conf.* **139** 00222
[8] Schmidhuber J 2015 Deep learning in neural networks: An overview *Neural Netw.* **61** 85–117
[9] Ali P J M and Faraj R H 2014 Data normalization and standardization: a technical report *Mach. Learn. Tech. Rep.* **1** 1–6
[10] Pedregosa F *et al* 2011 Scikit-learn: Machine learning in python *J. Mach. Learn. Res.* **12** 2825–30
[11] Abadi M *et al* 2016 TensorFlow: Large-scale machine learning on heterogeneous distributed systems *Preprint* arXiv:1603.04467
[12] Altmann A *et al* 2010 Permutation importance: a corrected feature importance measure *Bioinformatics* **26** 1340–47.
[13] Scornet E 2021 Trees, forests, and impurity-based variable importance in regression *Preprint* arXiv:2001.04295
[14] Yu T and Zhu H 2020 Hyper-parameter optimization: A review of algorithms and applications *Preprint* arXiv:2003.05689
[15] Bradley A P 1997 The use of the area under the ROC curve in the evaluation of machine learning algorithms *Pattern Recognit.* **30** 1145–59