LIP-STUDENTS-22-02

Anomaly Detection in all hadronic boosted final states

Junda Tong[1,a]

[1] Department of Physics, University of Liverpool, UK
[2] Laboratório de Instrumen-tação e Física Experimental de Partícula, Lisbon, Portugal

Project supervisor: N.Castro[2], R.Pedro[2]

October 10, 2022

Abstract. In this report, the machine learning technique of supervised Deep Neural Network(DNN), Graph Attention Network(GAT) and semi-supervised deep Auto-Encoder(AE) was explored, for the purpose of finding the possible new physics in the current experiment data, by using the simulated standard model events and beyond standard model events from ATLAS experiment. The performance of each model has been presented, and a comparison between supervised learning DNN and the anomaly detection approach of semi-supervised deep AE is discussed.

Keywords:ATLAS, Anomaly detection, Machine learning, Deep neural network, Graph attention network, Deep Auto-Encoder, DNN, GAT, AE, Optimization, Hyper-parameters

1 Introduction

The Standard Model (SM) in particle physics is known as the most successful model. The model can give an accurate prediction of most of the phenomena from the current experiments. However, SM is not perfect, as the SM describes three of the four fundamental forces, and there is evidence of the existence of dark matter by the observation of unexpected speed on the galaxy rotation curve.[1] For these mysteries beyond SM, the searches of Beyond Standard Model (BSM) have been conducted at many facilities, for example, the using Higgs boson to search dark photons in ATLAS.

The concerns about generic search in the experiment of ATLAS and CMS, suggest that the BSM signal could be omitted due to the insufficient sensitivity of the current strategy. Therefore, a possible improvement to the sensitivity in the collider is proposed that the Anomaly Detection (AD) method of machine learning may be sensitive to such signal.[2] Where the AD method refers to a type of machine learning model, where the model is trained by ordinary data, but able to isolate abnormal data from ordinary data. The proportionality of ordinary data for such a model is normally considered to have a larger portion than abnormal data. Thus, the application of the AD method in the classification of SM and BSM is expected to be trained by SM events only, and the model is able to separate the BSM events from SM events without knowing their details.

2 Machine learning models and simulated data

In the training of machine learning models, the simulated data were used, the data consist of simulated SM events as background and simulated BSM events as the signal, where in general, there are more background events than signal events. Based on the training set of simulated data, the supervised model of Deep Neural Network (DNN) and Graph Attention Network (GAT) was explored, and the AD method Auto-Encoder (AE) was explored as well. The supervised models were used for the comparison with AE.

The data were from simulated events of boosted top-quark with hadronic decay, some of the features can be described in figure 1. Signal events have large missing energy from undetected particles but largely overlapped top quark masses with background events and some discrepancy in the momentum information.

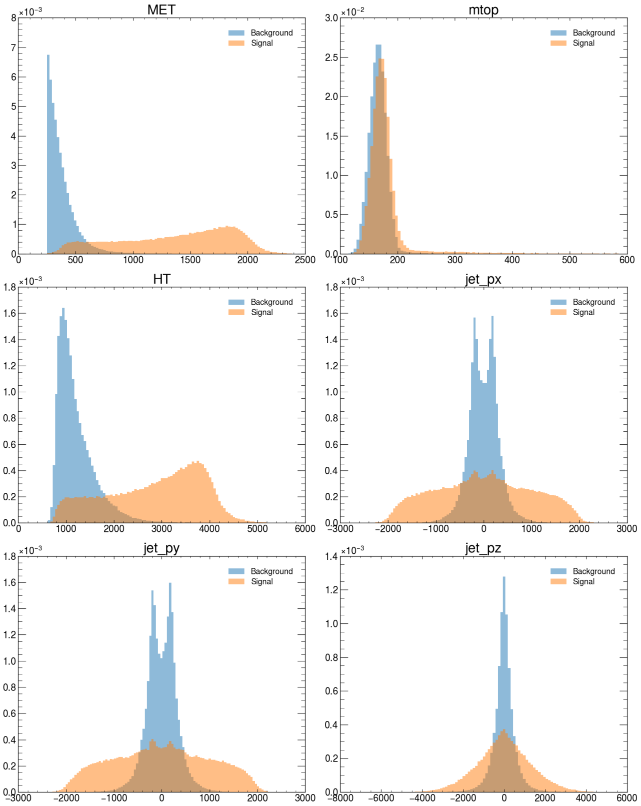


Figure 1: plot of mainly used features of simulated data.

[a] e-mail: psjtong3@liverpool.ac.uk

In the DNN model, the model learns the input features from the hidden layers for a regression task, where the hidden layers are the layers between input and output. The DNN model can be described by $y = f_{NN}(x)$, where x is the input, f_{NN} is the vector function on the hidden layer, which has the form:

$$\boldsymbol{f}_l(\mathbf{z}) \overset{\text{def}}{=} g_l\left(\mathbf{W}_l\mathbf{z} + \mathbf{b}_l\right) \tag{1}$$

Where l represents the layer index, z is the input, g_l is the activation function, matrix W_l and vector b_l are parameters that will be learned by the model algorithm. As the raining objective for DNN is to classify the signal and background events, the model output is given as a probability for an event to be signal or background.[3]

2.1 Graph Attention Network (GAT)

The architecture of GAT can be considered as a graph composed of nodes, and each node has features with the same dimensionality, to produce a new set of nodes with features in the graph attentional layer as its output. The model combined the idea of neural network and conversational network, with additional graph attention. In the graph attentional layer, the information from the node is passed to the neighbour node and computed the attention coefficient by:

$$e_{ij} = a\left(\mathbf{W}\vec{h}_i, \mathbf{W}\vec{h}_j\right) \tag{2}$$

Where W is weight matrix, a represents the shared attention mechanism $a : \mathbb{R}^{F'} \times \mathbb{R}^{F'} \rightarrow \mathbb{R}$. The attention coefficient normalized by the Softmax function is used for the corresponding linear combination in the neural network, therefore, the parameters of layers that are attached to the neural network can be learned by the model algorithm. Since the training objective of GAT was the same as DNN, the output of GAT is the same as DNN.[4]

2.2 Deep Auto-Encoder

The Deep AE have a symmetric architecture, mainly consisting of two part for the compression of and decompression of data. For the encoder in the compression part, the number of units in each layer is always decreasing as the index of the layer increases, the compressed data through a bottleneck layer, which connect the compressed part and decompressed decoder part, with the bottleneck layer as the intermediate layer, always have a smaller dimension than the original data sample. In the decoder, the number of units always increases as the index of layer increases, and finally, have the same dimensionality as the original data sample. The decoder is aiming to reconstruct the compressed data from the encoder therefore, by using the cost function to compute the error between the original data and reconstructed data, the error can be used as the anomaly score in the neural network.[3]

3 Model implementation, training and optimization

The simulated data set was split into train, validation and test set with predefined random state. The data consist of 97 features including the Monte-Carlo information, in all the stages of implementation, the Monte-Carlo information was excluded for the unbiased outcome. In the implementation of all three models, the input features were selected with only contain the basic information of four-momentum, and the condition of the jet, all the other information was avoided for the test of performance in the condition of limited information.

3.1 Training features

In the training of the DNN and Deep AE, the model was trained with the same features for the consistency of the comparison. The trained features with a total of 15 features and were standardized(see table 1).

Table 1: Features used in DNN and Deep AE.

DNN/Deep AE	
nGoodJets	Number of good jets
MET	Missing energy(both x,y direction)
mtop	Mass of top quark
HT	scalar Pt sum of all objects in the event
fjet_p(x,y,z)	Front jet 3-momentum
jet_p(x,y,z)	Jet 3-momentum
ljet_p(x,y,z)	Large 3-momentum
MET_p(x,y)	Missing energy components

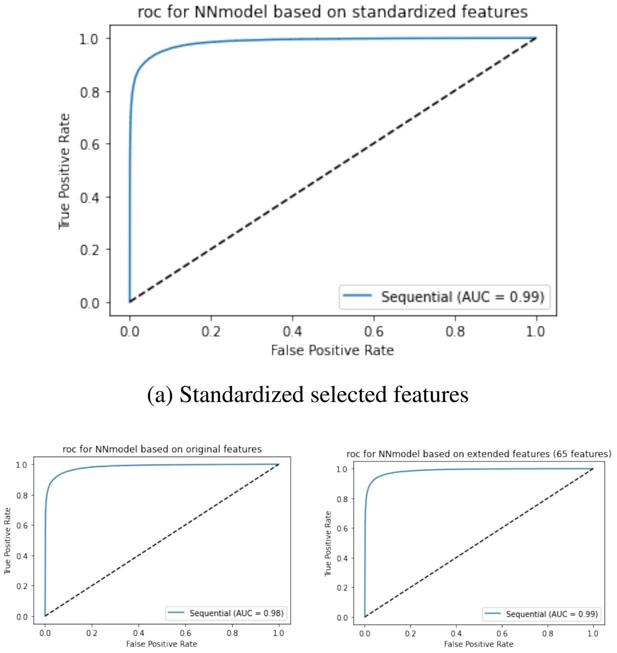
And due to the architecture of GAT, the trained features were differently. Four nodes were assigned in the GAT model, with four features under each node(see table 2). In the MET node, the MET_pt was used as a dummy feature, for the nodes can remain the same property, and ttbar_category was repeatedly applied on all the nodes. the model was trained by 16 features, but 13 unique features were used.

Table 2: Features used in GAT.

MET	jet	fjet	ljet
MET_pt	jet_pt	fjet_pt	ljet_pt
MET_phi	jet_phi	fjet_phi	ljet_phi
MET_eta	jet_eta	fjet_eta	ljet_eta
ttbar_cate	ttbar_cate	ttbar_cate	ttbar_cate

3.2 Models and optimization

In the implementation of DNN, GAT and Deep AE, the model was implemented in TensorFlow 2.9.1 with Keras 2.9.0. For the optimization of hyper-parameters, the Optuna package 2.10.1 was used. In the implemented DNN, the model was able to converge within a few epoch therefore, in order to avoid over-fitting, 100 epoch was used



(a) Standardized selected features

(b) Selected original features

(c) Full original features

Figure 2: ROC of DNNs, trained with different type of features.

with an early stop if the loss of the model does not have a significant improvement in 20 epoch. The DNN has also trained with full 65 features and selected unstandardized features. The ROC curve of three setups in figure 2, shows that the performance of input features does not change the classification of the model significantly, hence only the standardized selected features were optimized.

For DNN and AE, the detail of optimization of hyper-parameters is shown in table 3. In deep AE optimization, due to the symmetric architecture, the number of units was not directly optimized with optuna loop but correlated with the number of layers with $5 * i^3$, where i is layer index, the index reversed at the decoder layer. The optimizer Adam was used for all the trials, due to the Adam provided the best performance in the preliminary test. The best hyper-parameters configuration can be found in table 4

Table 3: Considered hyper-parameter in DNN and deep AE in the Optuna optimization. Search of number of layers and units and learning rate are in form of [initial value, maximum value].

Hyper-parameter	DNN range	AE range
Number of layer	[1, 8]	[2 ,5]
Number of unit	[2. 1024]	
Learning rate	[10^{-4}, 10^{-2}]	[10^{-4}, 10^{-2}]
Optimizer	[Adam, RMSprop]	[Adam]
Activation	[relu, selu]	[linear, selu, elu]
Batch size	[256, 512, 1024]	[256, 512, 1024]
Number of trial	50	200

Table 4: Best hyper-parameter configuration of DNN and AE.

Hyper-parameter	DNN	AE
Number of layer	8	5
Number of unit	879	
Learning rate	0.004825	0.004981
Optimizer	Adam	Adam
Activation	relu	selu
Batch size	256	256

In the optimization of GAT, the optimizer Adam was used for all trials, and the number of units in the hidden layer was correlated with number of layers with $6 * i^3$, where i is the layer index. The detail and best hyper-parameter can be found in the table5.

Table 5: Considered hyper-parameter and best result in GAT in the Optuna optimization. Search of number of layers and units and learning rate are in form of [initial value, maximum value].

Hyper-parameter	Range	Best
Number of layer	[2, 5]	5
GAT unit	[2, 1024]	56
number of unit	[2, 1024]	835
learning rate	[10^{-4}, 10^{-2}]	0.000159
Activation	[linear, selu, elu]	selu
Batch size	[256, 512, 1024]	512
Number of trial	50	

4 Results

4.1 Supervised learning

For the supervised learning DNN and GAT, the model shows high accuracy even if tested by unseen data in the training phase. By comparing DNN and GAT in the figure 3 and 4, the DNN shows better performance than GAT. However, the features used in the GAT are less than DNN, but GAT still can give similar performance, but due to requirements of GAT of data format, the simulated data-set was unable to provide more correlated features.

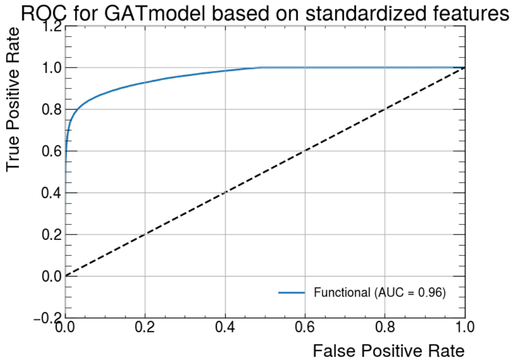


Figure 3: ROC of GAT by best hyper-parameter.

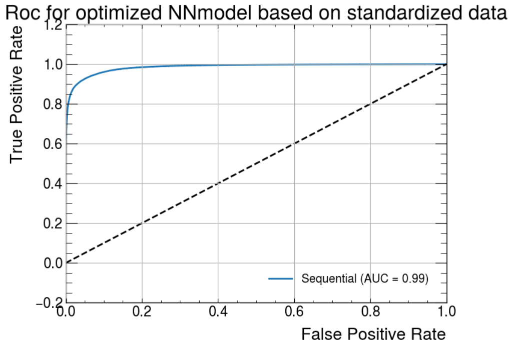


Figure 4: ROC of DNN by best hyper-parameter.

Furthermore, the GAT shows higher stability in the training phase than DNN. In the training of GAT and DNN, the model was trained in 100 epochs with the early stop of 20 epochs. By comparing the loss at GAT and DNN in figure 5 and 6, the DNN validation loss shows a different trend with training loss and diverged since around 30 epochs.

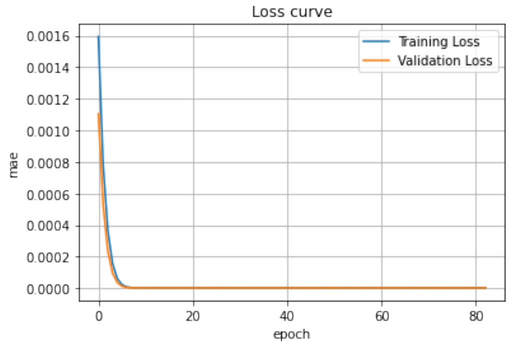


Figure 5: GAT loss in the training phase.

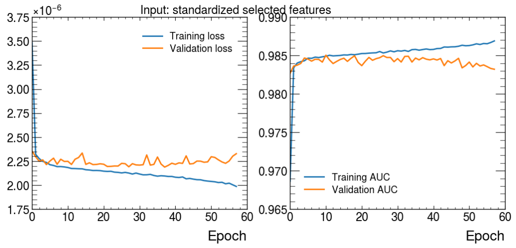


Figure 6: DNN loss and ROC in the training phase.

And from the output of DNN and GAT in figure 7 and 8, the GAT have better performance on the classification of background events, this is mainly due to the extra features ttbar_category in the training features. The performance of DNN shows an equivalent accuracy in the classification of background and signal events.

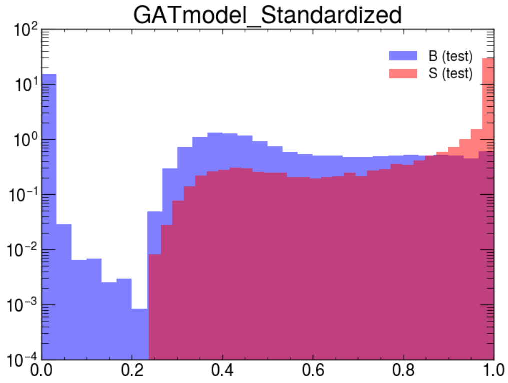


Figure 7: GAT output.

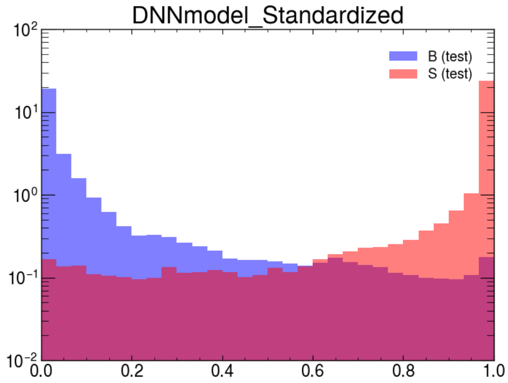


Figure 8: GAT output.

4.2 Semi-supervised learning

In semi-supervised learning deep AE, the model was trained with the same features as DNN, according to the classification report created by the Scikit-learn package in table 6, the model was trained only with the background events, 0.0 refer to background, 1.0 refer to signal. The model can isolate the majority of the signal events but is less accurate in the prediction of background events, overall the model gives a weighted average precision of 72%.

Table 6: Classification report of deep AE with best hyper-parameters.

	precision	recall	f1-score	support
0.0	0.60	1.00	0.75	70498
1.0	0.89	0.04	0.08	49029
Accuracy			0.61	119527
Macro avg	0.75	0.52	0.42	119527
Weighted avg	0.72	0.61	0.48	119527

In the reconstructed data by AE in figure9 and 10, the AE was able to reconstruct the majority of data within a certain region but lacked the ability to reconstruct the data with higher deviation. Especially in the case of fjet data in figure 11, the reconstruction has only matched a small amount of data in a limited range.

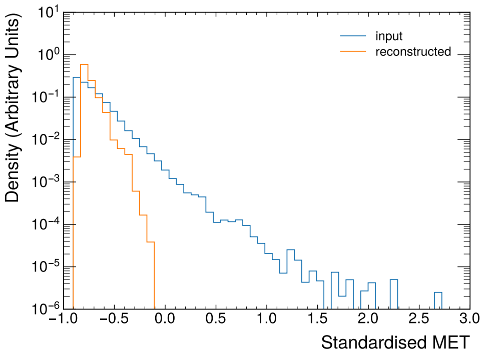


Figure 9: Reconstructed MET by deep AE.

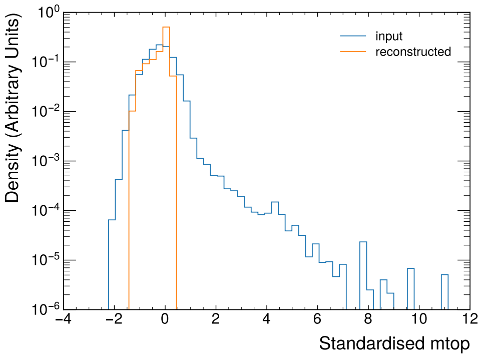


Figure 10: Reconstructed mtop by deep AE.

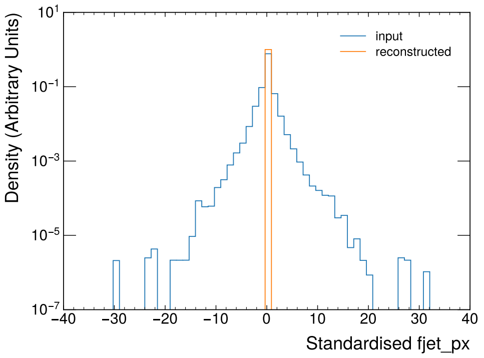


Figure 11: Reconstructed fjet_px by deep AE.

5 Discussion and conclusion

In the trained supervised learning DNN and GAT, the models show high performance in the classification of SM events and BSM events. To compare the DNN with the AD method deep AE, the performance of DNN was much better than AE, however, the supervised model is expected to have less precision if the test set is from non-correlated data with the training set, And AE is expected to perform a general equivalent precision for all kind of data-set. In the reconstructed data of AE, the AE shows the ability to reconstruct the majority of the data, but lack of precision in the fjet data, this is may due to the unexpected bias of the data-set or solely due to the scale of the fjet data was much larger than others.

In conclusion, DNN and GAT show promising results in the classification of SM and BSM even tested by unseen data in the training phase. If the data set can have compatible properties of features, GAT could have better performance than DNN. The deep AE is able to recognise signal events that were not seen during the training phase. The performance is less than supervised models but provided relatively reliable reconstructed data, and reasonable precision in the classification of BSM, therefore, deep AE can be used as a generic signal classifier. However, the model needs to be parallel compared with other AD method models, the comparison between DNN can not provide a sufficient result.

Acknowledgements

I wish to show my appreciation to my supervisors, Nuno Castro and Rute Pedro for all the support and feedback throughout the internship. I believe the experience in this internship will deeply affect my future study.

References

[1] E. Corbelli, P. Salucci, Monthly Notices of the Royal Astronomical Society (2000), [Online] https://academic.oup.com/mnras/article-lookup/doi/10.1046/j.1365-8711.2000.03075.x

[2] M.C. Romao, N.F. Castro, R. Pedro, The European Physical Journal C (2021)

[3] A. Burkov, *The Hundred-Page Machine Learning Book* (2019), http://ema.cri-info.cm/wp-content/uploads/2019/07/2019BurkovTheHundred-pageMachineLearning.pdf

[4] P. et al. V, *Graph Attention Networks*, ICLR 2018 (2018)