A Study on Higgs Boson Detection Using Multilayer Perceptron and Support Vector Machines

Asha Guruvayurappan
asha.guruvayurappan@city.ac.uk

Abstract
This paper attempts to evaluate the predictions of Multi-layer Perceptron and Support vector Machines in a binary classification problem, to identify if a signal is Higgs boson or background. Both these models are seen to be capable of learning and classifying signals with a great degree of accuracy. The best classification from each model was compared to an unaltered, unprocessed test to evaluate performance using confusion matrix.

DESCRIPTION AND MOTIVE

An exotic energy field is found to exist in every region of the universe accompanied by a fundamental particle known as Higgs boson. On investigating the structure of this matter and the laws that govern its interaction, this field strives to discover the fundamental property of physical universe. Observing these particles' properties may provide vital insights into the very basis of matter [1]. Identifying these rare particles lead to a significant challenge in differentiating the Higgs signal from background signal, Machine-learning classifiers, such as neural networks, offer a powerful solution to this challenge [1].

The objective of this paper is to critically evaluate two models to determine if the energy is Higgs signal or background signal. This analysis is done using IPython and Scikit-learn Deep Learning Library to build a Multi-Layer Perceptron (MLP) and Support Vector Machine (SVM) with various modifications, two of the most popular neural networks models. Both the models are robust and can handle huge amount of data even with noise to make accurate predictions [2].

EXPLORATORY DATA ANALYSIS

The Higgs boson dataset was obtained from OpenML [3] also available in UCL [4]. The dataset contains 29 continuous attributes and 98050 records. Monte Carlo simulation has been used to generate the data. The first 21 features are kinematic properties determined by the accelerator's particle detector. The last seven feature are calculated functions developed by physicists derived from the first 21 features. The target variable is a binary class that states 0 - background, 1 - Higgs signal. The dataset is balanced with 52.8% as higgs signal and 47.2% as background signal.
As a first step, the data was split to training and test sets in the ratio of 80 - 20 and the test set was untouched during analysis or processing; the test set was introduced only to the best trained models to get the accuracy of the best trained model. On preprocessing the training data, few special characters were observed, and few missing values were present, these records were dropped. Few columns are positively skewed for which the training data was normalized when training. Presence of outliers were identified using a boxplot, a z-score was computed with training data, relative to mean and standard deviation to remove outliers; The z-score was chosen such that, the data scale are preserved and patterns are not changed also to minimize loss of data [Figure 1].

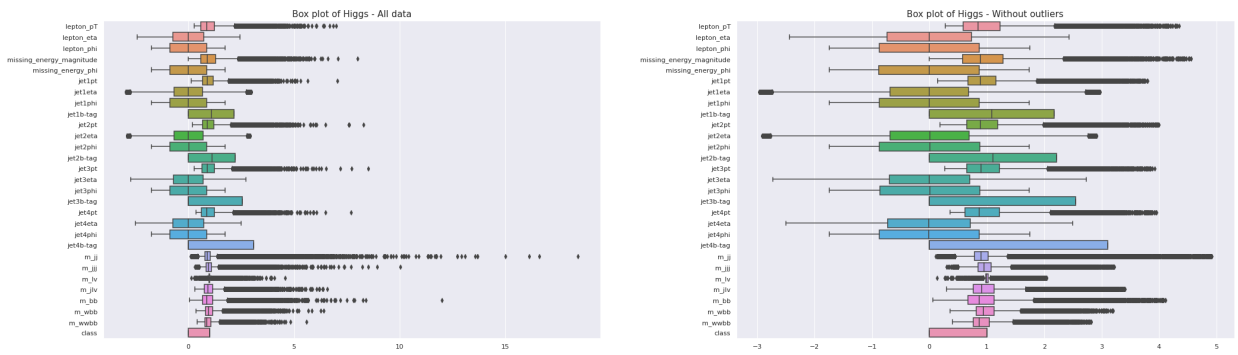

Figure 1 : Higgs pre-processing - Whole Data vs Without Outliers

To perceive the relationship of each variable with its adjacent variable parallel coordinates were plotted, it can be observed that the last 7 variables have a similar distribution. [Figure 2] highlights efficiently that Higgs signal (class - 1) have low value for all these attributes.

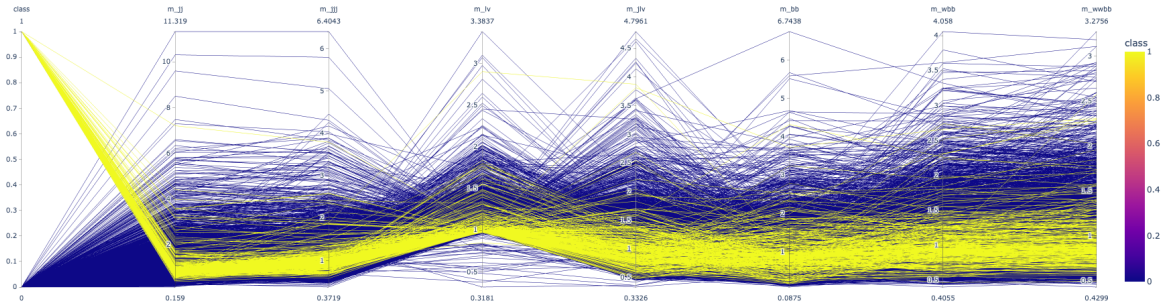

Figure 2 : Parallel coordinates - Calculated attributes

DESCRIPTION OF MODELS

Multi-Layer Perceptron

A Multi-Layer Perceptron is a feedforward Artificial Neural Network (ANN). that consists of three types of layers: an input layer, hidden layer(s), and output layer [5]. It is the most often used neural network model in deep learning. A perceptron is an artificial neuron, also known as a node, that passes information from one node to another, much like the human brain. In a fully connected network each node is connected to every other node in the subsequent layers. The input layer distributes data to the subsequent layers; Linear activation function and no thresholds are used in input nodes. In addition to the weights, each hidden unit node and each output node have thresholds connected with it. The outputs have linear activation functions, but the hidden unit node have nonlinear activation functions. [6]

Advantages	Disadvantage
• MLP neural networks are well suited to the classification. • Works well with large datasets and also can achieve same accuracy with smaller datasets [7] • Works well with complex non-linear data • Training time is quick	• Computationally expensive • The quality of the training data determines how well the model works. • Generalization issues arise when the model does not function effectively [7].

Support Vector Machine

Support Vector Machines are a set of supervised learning methods used for both classification and regression problems, it can also be used to identify outliers [8]. The objective of SVM algorithm is to find a hyperplane in n-dimensional space that distinctly classifies the data points into the dimension that has a clear dividing margin between classes. SVM with the use of kernels can be used to classify non-linear classification problems. A kernel is a similarity function that determines how similar two inputs are [9].

Advantages	Disadvantage
• SVM minimizes overfitting [9] • Works well with small clean datasets • Can be used for linear and non-linear classification problems. • Works well with structured and semi-structured data like image, texts [9]	• Choosing a good kernel is not easy • Training a SVM or performing a GridSearch is time consuming. • Hyperparameter tuning is not easy; and is difficult to estimate the impact. [9]

HYPOTHESIS

(H1) The initial hypothesis is that, even though MLP captures short-term dependencies it will outperform SVM even after fine tuning hyperparameter.
(H2) The subsequent hypothesis is that, feature extraction and feature importance will result in best performance; this will result best results when compared to tuned models.

METHODOLOGY

The methodology that was adapted was to split train and test data as 80/20 for model implementations and comparisons; the test data was untouched and was not included in any analysis. The training data was further split into training and validation sets (holdout 10% of training data), as the dataset was huge we still had enough data to train on.

Feature Selection

It was observed that the derived features were highly correlated with one another, also refer [Figure 2]. *A good feature subset is one that contains features highly correlated with (predictive of) the class, yet uncorrelated with (not predictive of) each other*. [10] For this reason, it was necessary to explore this aspect. A component analysis was also carried out covering various ranges of features, i.e. coverage of 95%, 97%, 99% to observe which yielded better results.

Models and Parameters

Multiple models were trained and validated for MLP before reaching the final best model. To start with, a basic MLP model with one hidden layer of 2 nodes was created and validated to understand the basic computational cost of and time taken for training. The next couple of models with 5 node hidden layer and activation function as logistic and relu respectively was created to check if there is any difference in the accuracy obtained on validation set. A grid search on activation function was performed out of which the best two activation function was picked. With these two activation functions multiple grid search to obtain optimal number of hidden layers and nodes was done like one hidden layer with 5, 10, 25, 50, 100 nodes, two hidden layers each with 5 and 30, 10 and 30, 25 and 30 was done. [11] highted the importance of data scaling for implementing MLP or any perceptron based algorithms and hence this aspect of Standardizing data was explored to check if it had any impact on models. In an attempt to try and improve model, a component analysis was carried out with different percentage of coverage of data with and without scaled data. With all the results obtained, a grid search for other parameters like learning rate which controls the step size in updating weights, alpha a L2 regularization, random state and maximum iterations was performed. As this grid search was running for a very long time, the data was subsampled to 8000 records to obtain the best set of parameters. An attempt to replicate [1] with 5 hidden layers with 300 nodes each, learning rate at 0.005 and relu activation with other set of best parameters was implemented to check the model performance. Another approach to create a model with only 7 calculated fields which has high correlation with the best set of parameters was also implemented.

As for SVM, the first model was trained with linear kernel on the whole training set and accuracy was obtained on the validation set. This was done to understand computational time of SVM model. This basic model alone took up to 48 minutes to train as the dataset was huge, around 72,000 records. For the purpose of coursework 10% of data was subsampled from training set in order to reduce computational Time. A second SVM with kernel as linear, C = 0.1 and gamma = 0.1 was trained to check the change in model performance. Observing the result a grid search on the sampled data was performed over linear, rbf, poly and sigmoid to get the best result. Using the best kernel, a hyperparameter were tuned to get the best C and gamma values. C is a hyperparameter to control error by creating a nominal decision boundary and gamma gives the curvature in the decision boundary.
Another approach was to perform component analysis by reducing the components by giving various percentage of coverage like 0.95, 0.97, 0.99. An approach to evaluate model performance before and after data standardization, with standardization and component analysis was also implemented. [12][13] suggested that for a linear SVM gamma need not be implemented and so gamma was excluded from

the training. Cross validation on the best set of parameters was implemented to check overfitting of model.

ANALYSIS AND EVALUATION

Comparing the two models helped gain a lot of information on the models and data. With respect to configuration MLP was faster and most of evaluations were done on the whole training set. The basic model yielded an overall accuracy of 65.3% on validation set. Grid search to tune hidden layers and activation function resulted in a slightly better accuracy. Performing a component analysis with different coverages did not make a huge difference, but according to [11] a standardised data yield better results for MLP; a component analysis before and after standardization showed significant improvement on validation set of 70.36%. [Figure 3]

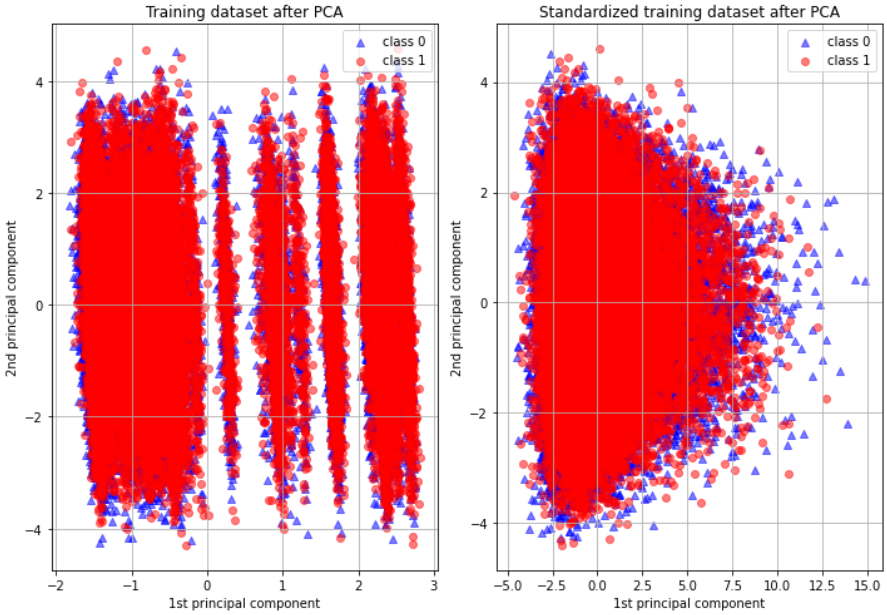

Figure 3 : MLP training data before and after Standardization

Running a grid search to identify best parameters on this standardized data and reduced attributes, obtained the best set of parameters which was cross validated to avoid over fitting. Best set of parameters in Table 1. An attempt to replicated [1] with 5 hidden layers gave only 63% accuracy on validation set, assuming some of the parameters were not discussed in the paper. Also, a model created with only the 7 calculated columns ['m_jj', 'm_jjj', 'm_lv', 'm_jlv', 'm_bb', 'm_wbb', 'm_wwbb'] on validating resulted 70% accuracy, but considering the corelation between these attributes [Figure2], it is not recommended to train with only these attributed.

MLP		SVM	
Activation Function	relu	kernel	linear
Hidden Layers	2 hidden layers with (25, 30) nodes	C	10
alpha	0.001		
learning rate	0.001		
max iterations	100		
random state	10		

Table 1 : Best set of Parameters

	MLP	SVM
Accuracy	57.8	64.4
Precision - class 1	0.59	0.63
Recall - class 1	0.66	0.82
fi-score - class 1	0.62	0.71
Time	13 minutes	135 minutes

Table 2 : Scores - MLP vs SVM

On the other hand, training SVM with the whole training set was computationally inefficient, the basic model alone trained for 48 minutes. Considering the time constrain, the subsampled (10%) of data was utilized to get best results and the final model was tuned with these results and the whole training set. The basic model gave 64% as accuracy on validation which was almost similar to accuracy obtained from MLP basic model. A grid search to tune the best kernel resulted in linear, another grid search to

tune C and gamma was implemented, though it is suggested that gamma is not required for linear SVM, to experiment on the results gamma was included in grid search. An attempt to improve performance by standardizing data before and after component analysis did not yield any better results which contradicts our hypothesis (H2) that important features and standardised data will give better results in SVM.

The best models from MLP and SVM was tested on unseen test data to observe the results. The test data for MLP was standardized and reduced to 27 features which was the best training model results. Though the best MLP on validations gave 71% accuracy, it gave only 58% accuracy on unseen and unprocessed data. [Figure 4]. This could be the result of overfitting on training data.
SVM' s best model trained for about 2 hours 10 minutes on the whole training set yielded 66% accuracy; when tested on unseen unprocessed data gave 64% accuracy.

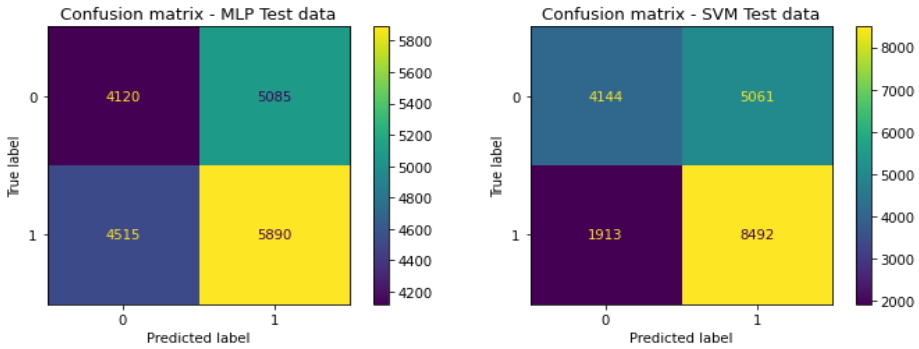

Figure 4 : Confusion Matrix - MLP vs SVM

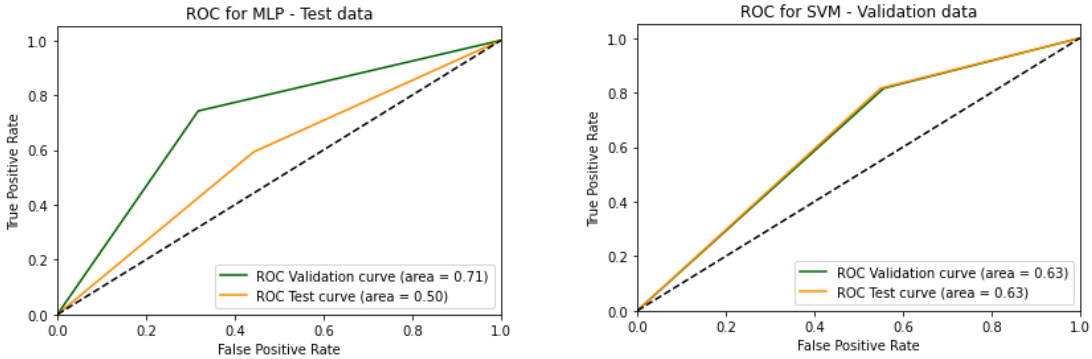

Figure 5 : ROC - MLP vs SVM / Validation vs Test

CONCLUSION

In this paper we compared the performance of Multilayer Perceptron and Support Vector Machines for a binary classification problem. MLP performed well on validation data after standardization and extracting the principle components giving 71% accuracy, but when tested on unseen data the accuracy came down to 58.3% showing that the model was overfitting in training data. Precision and recall are low on identifying background signals. On the other hand SVM's best model obtained 66% on training data and 64.4% on test unseen data which actually gives a better performance compared to MLP and contradicts our first hypothesis that MLP will out perform SVM [Figure 5].
The second hypothesis that feature selection and scaling will improve performance was true for MLP but in case of SVM the accuracy of validation set decreased when data was either standardized or when principle components were extracted. Thus, SVM in this dataset works best with all the features.

Future work could be to improve MLP model from overfitting on training data. We could also focus on improving the overall performance of both the models. For MLP, model can be experimented with different solver other than adam and also with momentum. For SVM, perform hyperparameter tuning on whole training set rather than 10% of data.

REFERENCES

[1] P. Baldi, P. Sadowski, and D. Whiteson, “Searching for Exotic Particles in High-Energy Physics with Deep Learning,” *Nat Commun*, vol. 5, no. 1, p. 4308, Sep. 2014, doi: 10.1038/ncomms5308.

[2] E. A. Zanaty, “Support Vector Machines (SVMs) versus Multilayer Perception (MLP) in data classification,” *Egyptian Informatics Journal*, vol. 13, no. 3, pp. 177–183, Nov. 2012, doi: 10.1016/j.eij.2012.08.002.

[3] Daniel Whiteson, “HIGGS Data Set.” [Online]. Available: https://archive.ics.uci.edu/ml/datasets/HIGGS

[4] Daniel Whiteson daniel’@’uci.edu", Assistant Professor, Physics, Univ. of California Irvine, “higgs.” [Online]. Available: https://www.openml.org/search?type=data&status=active&id=4532

[5] F. Amato, N. Mazzocca, F. Moscato, and E. Vivenzio, “Multilayer Perceptron: An Intelligent Model for Classification and Intrusion Detection,” in *2017 31st International Conference on Advanced Information Networking and Applications Workshops (WAINA)*, Taipei, Taiwan, Mar. 2017, pp. 686–691. doi: 10.1109/WAINA.2017.134.

[6] Nazzal, Jamal and El-Emary, Ibrahim and Najim, Salam, “Multilayer Perceptron Neural Network (MLPs) For Analyzing the Properties of Jordan Oil Shale,” *DOSI Publications, 2008*, vol. 5, [Online]. Available: https://www.researchgate.net/publication/239580128_Multilayer_Perceptron_Neural_Network_MLPs_For_Analyzing_the_Properties_of_Jordan_Oil_Shale

[7] Akkaya, Berke and Çolakoğlu, Nurdan, “Comparison of Multi-class Classification Algorithms on Early Diagnosis of Heart Diseases.” 2019. [Online]. Available: https://www.researchgate.net/publication/338950098_Comparison_of_Multi-class_Classification_Algorithms_on_Early_Diagnosis_of_Heart_Diseases

[8] “Support vector machines.” [Online]. Available: https://scikit-learn.org/stable/modules/svm.html

[9] “204.6.8 SVM : Advantages Disadvantages and Applications”, [Online]. Available: https://statinfer.com/204-6-8-svm-advantages-disadvantages-applications/

[10] Hall, Mark Andrew and others, “Correlation-based feature selection for machine learning,” *Citeseer*, 1999.

[11] T. P. Oliveira, J. S. Barbar, and A. S. Soares, “Multilayer Perceptron and Stacked Autoencoder for Internet Traffic Prediction,” in *Advanced Information Systems Engineering*, vol. 7908, C. Salinesi, M. C. Norrie, and Ó. Pastor, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2014, pp. 61–71. doi: 10.1007/978-3-662-44917-2_6.

[12] A Man Kumar, “C and Gamma in SVM,” 2018, [Online]. Available: https://medium.com/@myselfaman12345/c-and-gamma-in-svm-e6cee48626be

[13] “The effect of gamma value on support vector machine performance with different kernels,” *IJECE*, vol. 10, no. 5.

APPENDIX

Glossary

Higgs boson. An exotic signal observed on many parts of the planet but difficult to differentiate from background signal, also these provide vital insights into the very basis of matter.

accuracy. The match between a sample and the target population is referred to as accuracy. It also indicates how close a value obtained from prediction is equal to the actual value.

activation function. A gate between input to current neuron and its output to next layer.

binary classification. The given attributes belong to one of the two classes

correlated attributes. Correlation is a statistical measure that expresses the extent to which two variables are linearly related, meaning correlated predictors change together at a constant rate.

features. Each feature is a column that represent measurable data which can be used for analysis.

Overfitting. A overfitting model performs exceptionally on training data but fail to perform on test or unseen data.

hyperparameter. A parameter from prior distributions. A result obtained from various parameters

positive correlation. Relationship between two variables in which both move in same direction, i.e. when one increases the other increases.
kernel. Map original observation to high dimensional space where they are seperable
standardization. Data standardization is the process of bringing data into a uniform format.
Data is received from various sources in various formats, necessitating standardisation for analysis.

$$X\_new = (X - mean) / Std$$

target. The target is a feature or column in the dataset that is classified or predicted based on all other features. These values can be both categorical or numerical.

Formulas

$$\text{Precision} = \frac{True\ Positive}{True\ Positive + False\ Positive}$$

$$\text{Recall} = \frac{True\ Positive}{True\ Positive + False\ Negative}$$

$$\text{F1 Score} = 2 * \frac{Precision * Recall}{Precision + Recall}$$

IMPLEMENTATIONS

Data Set Information

The data has been produced using Monte Carlo simulations. The first 21 features (columns 2-22) are kinematic properties measured by the particle detectors in the accelerator. The last seven features are functions of the first 21 features; these are high-level features derived by physicists to help discriminate between the two classes. There is an interest in using deep learning methods to obviate the need for physicists to manually develop such features. Benchmark results using Bayesian Decision Trees from a standard physics package and 5-layer neural networks are presented in the original paper. The last 500,000 examples are used as a test set.

Attribute Information

The first column is the class label (1 for signal, 0 for background), followed by the 28 features (21 low-level features then 7 high-level features): lepton pT, lepton eta, lepton phi, missing energy magnitude, missing energy phi, jet 1 pt, jet 1 eta, jet 1 phi, jet 1 b-tag, jet 2 pt, jet 2 eta, jet 2 phi, jet 2 b-tag, jet 3 pt, jet 3 eta, jet 3 phi, jet 3 b-tag, jet 4 pt, jet 4 eta, jet 4 phi, jet 4 b-tag, m_jj, m_jjj, m_lv, m_jlv, m_bb, m_wbb, m_wwbb. For more detailed information about each feature see the original paper.

Model Information

Multiple models were created with MLP and SVM algorithms to identify higgs signal from background. This was implemented in python and sklearner deep learning libraries in Google Collab Notebook Pro.