ACAT-2022 IOP Publishing
Journal of Physics: Conference Series **3206** (2026) 012079 doi:10.1088/1742-6596/3206/1/012079

Uncertainty estimation in deep learning based-classifiers of High Energy Physics events using Monte Carlo Dropout.

R Pezoa[1,3], S Bórquez[1,3], W Brooks[2,3], L Salinas[1,3], C Torres[1,3]

[1] Departamento de Informática, Universidad Técnica Federico Santa María
[2] Departamento de Física, Universidad Técnica Federico Santa María
[3] Centro Científico Tecnológico de Valparaíso, Universidad Técnica Federico Santa María

E-mail: raquel.pezoa@usm.cl,sebastian.borquez@sansano.usm.cl, william.brooks@usm.cl, luis.salinas@usm.cl, claudio.torres@usm.cl

Abstract. Deep learning-based models are essential in High Energy Physics data analysis, demonstrating outstanding results when classifying events. However, high classifiers performance is as important as the uncertainty quantification of the model prediction. Deep learning systems only produce a point estimate which does not allow quantifying the uncertainty of each prediction. This work uses a neural architecture search tool to automatically select a neural network architecture, the Monte Carlo Dropout technique to approximate a predictive distribution and measure the uncertainty when classifying events using deep neural networks. The experiments show that the Monte Carlo Dropout method can estimate *predictive entropy* and *mutual information* measurements, which enables increasing trust in the classifications of high energy physics events.

1. Introduction

Separating *signal* events from the *background*, i.e., classifying High Energy Physics (HEP) events, is a critical analysis task in the HEP field and a foundational task in the search for new phenomena. The deep learning approach has been successfully applied to HEP event classification tasks, achieving superior results compared to traditional methods. These new tools can adeptly handle higher-dimensional and more complex problems than previously feasible. However, high classification performance is as important as the uncertainty quantification of the model prediction. Deep neural networks only produce a point estimate, which does not allow quantifying the uncertainty of each prediction, despite its critical importance in scientific applications [1].

In this work, we implement a Bayesian deep learning-based algorithm for measuring uncertainty when we classify HEP events using a deep neural network classifier. The work is focused on the use of the Monte Carlo Dropout (MC-Dropout) method, a variational inference technique proposed in [2] that is based on Dropout [3], a well-known regularization technique used to overcome overfitting. The Monte Carlo Dropout method allows the production of the posterior distribution of the network weights by training a dropout network that approximates Bayesian inference. Thus, a Bayesian deep neural network considers a distribution over network parameters instead of a single point. The traditional Dropout method randomly toggles off some neurons, with probability D_{rate} (or dropout rate) during the training stage. However, the MC-Dropout method toggles off neurons both during the training stage and also during the inference stage. In this work, we address the classification problem using the public dataset of the Higgs boson [4], which is the signal of interest, and the goal is to separate it from the background. The *keras* and *autokeras* frameworks are used in the paper. *Keras* is the well-known neural network Python library, and *autokeras* is also a Python library that allows us to automate the process of creating the artificial neural network architecture.

This paper is organized as follows. Section 2 introduces the MC-Dropout method. Section 3 describes the dataset and the classifiers configuration using the *autokeras* library. Then, in Section 4, we present the results. Finally, in Section 5, we present the conclusions and future work.

2. Monte Carlo Dropout

Monte Carlo Dropout [2] (MC-Dropout) is a Bayesian deep learning-based method that can estimate uncertainty in deep learning architectures. The uncertainty measures provide an objective measure of the model's certainty concerning its prediction. It is common to categorize uncertainty into two types: *epistemic* and *aleatoric* [5]. In this work, we will estimate epistemic uncertainty (also called model uncertainty) related to the lack of knowledge for model generation. It can be reduced as more training data is obtained. We compute the predictive entropy and mutual information, both described in [2], to measure the epistemic uncertainty using the MC-Dropout method.

Consider a deep neural model $\boldsymbol{F}(\boldsymbol{x}, \boldsymbol{\omega})$ with parameters $\boldsymbol{\omega}$, the training set $\mathcal{D}_{\text{train}} := \{\boldsymbol{X}, \boldsymbol{Y}\}$, where $\boldsymbol{X} := \{\boldsymbol{x}_1, \ldots, \boldsymbol{x}_N\}$ and $\boldsymbol{Y} := \{\boldsymbol{y}_1, \ldots, \boldsymbol{y}_N\}$ are the inputs and outputs, respectively. Bayesian models allow predictions on a new input point $\boldsymbol{x}^*$, predicting $\boldsymbol{y}^* = \boldsymbol{F}(\boldsymbol{x}^*, \boldsymbol{\omega})$ given the learned weights $\boldsymbol{\omega}$. Thus, the **predictive distribution**, i.e., the distribution of a prediction $\boldsymbol{y}^*$ is given by:

$$p(\boldsymbol{y}^*|\boldsymbol{x}^*, \boldsymbol{X}, \boldsymbol{Y}) = \int p(\boldsymbol{y}^*|\boldsymbol{x}^*, \boldsymbol{\omega}) \underbrace{p(\boldsymbol{\omega}|\boldsymbol{X}, \boldsymbol{Y})}_{\text{posterior dist.}} d\boldsymbol{\omega}. \quad (1)$$

The posterior distribution of Equation 1 is usually intractable. Instead, the posterior distribution $p(\boldsymbol{\omega}|\boldsymbol{X}, \boldsymbol{Y})$ can be approximated by a variational distribution $q(\boldsymbol{\omega})$, which can be achieved with the minimization of the Kullbak-Leibler (KL) divergence (details in [6]). However, this minimization is computationally expensive, so the predictive distribution can be approximated using the MC-Dropout method. This technique allows us to approximate this posterior distribution $p(\boldsymbol{\omega}|\boldsymbol{X}, \boldsymbol{Y})$ providing a *scalable* way by randomly toggling off neurons in a neural network during the training and inference stages. Each dropout configuration corresponds to a different sample from the approximate parametric posterior distribution. Thus, $\boldsymbol{F}(\boldsymbol{x}, \boldsymbol{\omega})$ with parameters $\widehat{\boldsymbol{\omega}}$ is used to approximate a predictive distribution $\widehat{p}(\boldsymbol{y}|\boldsymbol{x}, \mathcal{D}_{train})$ given by:

$$p(\boldsymbol{y}|\boldsymbol{x}, \mathcal{D}_{train}) \approx \frac{1}{T} \sum_{i=1}^{T} p(\boldsymbol{y}|\boldsymbol{x}, \widehat{\boldsymbol{\omega}}_t) = \widehat{p}(\boldsymbol{y}|\boldsymbol{x}, \mathcal{D}_{train}), \;\; \widehat{\boldsymbol{\omega}}_t \sim q(\boldsymbol{\omega}). \quad (2)$$

Here, $\widehat{p}$ is the average of the predictive distribution that is obtained with the MC-Dropout method, which is equivalent to performing T stochastic forward passes (see Figure 1) over the deep neural network during the inference process with dropout and then averaging the results. The approximated predictive distribution allows us to compute the predictive entropy and mutual information estimates of each classified HEP event of the testing dataset.

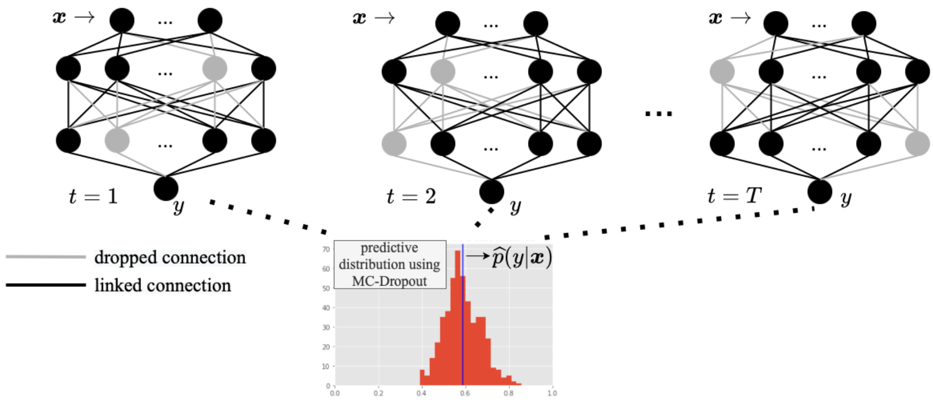


Figure 1. Monte Carlo Dropout diagram. $T = 500$ forward passes, $\boldsymbol{x_1} \in \mathbb{R}^{n=28}$ represents the feature vector of an event.

3. Classification and Uncertainty Estimation

3.1. Dataset

We use a publicly available *Higgs* dataset described in [4] and downloaded from OpenML [7]. Here, the signal is the new theoretical Higgs bosons, and the background is the process with identical decay products but different kinematic features. More precisely, the signal is the process defined by: $gg \rightarrow H^0 \rightarrow W^{\mp}H^{\pm} \rightarrow W^{\mp}W^{\pm}h^0 \rightarrow W^{\mp}W^{\pm}b\bar{b}$, where the signal process starts with the fusion of two gluons (gg), with intermediate Higgs and W bosons that finally decay into a pair of b quarks.

Each event is represented by a feature vector $x \in \mathbb{R}^{28}$ of 21 low-level features corresponding to physics properties measured by the detector and 7 high-level features derived from the previous ones. The low-level features consist of transverse momentum measurements, pseudorapidity, azimuthal angles, b-tagging information of each jet, and missing energy data. The high-level features correspond to reconstructed masses of lepton, neutrino, jets, and b-quarks. In addition, each event has the class label $y \in \{0, 1\}$, and from the machine learning point of view, this a binary classification problem, where events are classified as signal ($y = 1$) or background ($y = 0$). Note that the classifier's output is a score in the $[0, 1]$ range. Hence, a threshold t (commonly $t = 0.5$) is applied to the output to classify the event into the signal or background class.

3.2. Builiding the MC Dropout classifiers

The design of neural architectures is critical in achieving high classification performance, and the neural architecture search (NAS) approach [8] provides various tools for designing the architectures automatically. In this work, we build a deep learning-based classifier using *autokeras* [9] to identify a suitable architecture for the Higgs dataset. We used training, validation, and test sets in a 60%, 20%, and 20% split. The selected network architecture using *keras* comprises three hidden layers, with 128, 512, and 128 neurons in each layer, batch size equal to 8, the ReLu activation function in hidden layers, the sigmoid function in the output layer, and binary cross-entropy as the loss function.

Once the network architecture was defined, we implemented the MC-Dropout method, using $D_{rate} = \{0.2, 0.5, 0.9\}$ and $T = 500$ forward passes to produce the predictive distribution $\widehat{p}(y|\boldsymbol{x}, \mathcal{D}_{train})$. Thus, using MC-Dropout, i.e., toggling off neurons during the training and inference stage, the network's output is a distribution instead of a single point, allowing us to compute uncertainty measures. Different learning rates directly impact the distribution of the weights learned by the model during the training stage and hence the classifier performance.

4. Results

We computed the following performance metrics: accuracy (Acc), precision (Prec), recall (Rec), F1-score (F1), and the area under the ROC curve (ROC-AUC). Table 1 shows the results using different D_{rate} values and a baseline classifier, i.e., a traditional deep neural network without the MC-Dropout technique and hence, without uncertainty estimation.

D_{rate}	Acc	Prec	Rec	F1	ROC-AUC
-	0.77	0.79	0.77	0.78	0.85
0.2	0.76	0.77	0.78	**0.77**	**0.84**
0.5	0.74	0.76	0.74	0.75	0.82
0.9	0.53	0.53	1.00	0.69	0.50

Table 1. Classification performance metrics for the testing set, using the MC-Dropout method with dropout rates $D_{rate} = \{0.2, 0.5, 0.9\}$. Notice that higher dropout rates decrease classification performance.

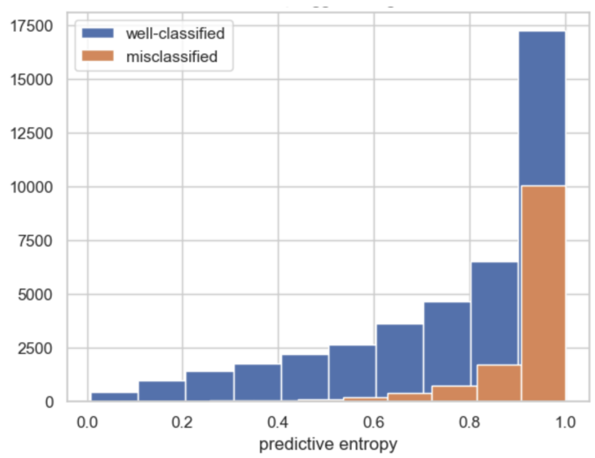
The baseline classifier achieved 0.78 in F1-score and 0.85 in ROC-AUC. We use the F1-score (the harmonic mean of precision and recall) and the ROC-AUC metrics to select the best MC-Dropout classifier. When MC-Dropout is used, we observed that the best classification performance was obtained when $D_{rate} = 0.2$, reaching a 99% of the baseline classifier, with 0.77 F1-score and 0.84 ROC-AUC. In addition, we observed that higher dropout rates decrease classification performance. This makes sense, considering that a high dropout rate means toggling off many neurons in the network, which slows the convergence rate of the model, jeopardizing the classification performance.

We computed the uncertainty measures using the classifier with $D_{rate} = 0.2$. Figure 2-(a) shows the histogram of the predictive entropy of well-classified and misclassified testing events, where entropy values are distributed primarily around one. This indicates that most of the events samples have an average predictive distribution $\widehat{p}(\boldsymbol{y}|\boldsymbol{x}, \mathcal{D}_{train}) \approx 0.5$ (Equation 2). It is expected to have prediction values close to 1 (for signal) or close to 0 (for background) to have a confident model's predictions. On the other hand, Figure 2-(b) depicts the histogram of the mutual information, and we observe uncertainty values around zero. This result indicates that the model's output is consistent during all the forward passes ($T = 500$).

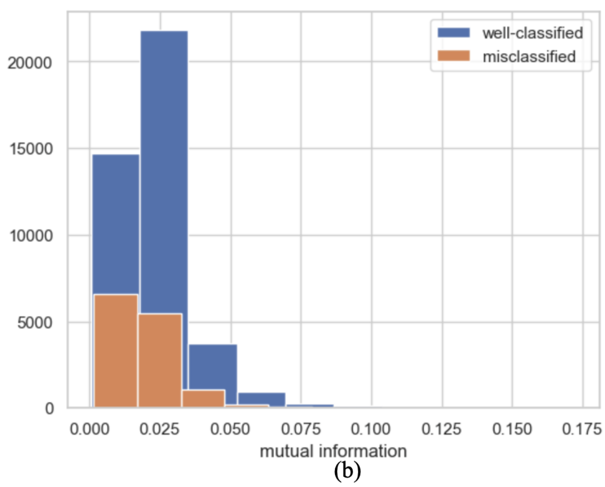
5. Conclusions

We have shown the application of the MC-Dropout method to estimate uncertainty when classifying HEP events and maintaining the baseline classification performance. We used *autokeras* tool to automatically select an artificial neural network architecture. The MC-Dropout method was directly integrated into a deep neural network. The best classification performance was achieved using $T = 500$ forward passes and $D_{rate} = 0.2$. We computed epistemic uncertainty measures and observed high predictive entropy and low mutual information in classifying Higgs events, which shows predictive entropy performance must be improved.

This work shows that MC-Dropout methods have the potential for increasing trust when complex deep learning-based models classify HEP events. Future work includes improving the network architecture and integrating the uncertainty estimations in the training stage (active learning) to increase classification performance confidently. In addition, we plan to combine Bayesian Deep learning methods with eXplainable Artificial Intelligence (XAI) techniques [10] to understand the model's prediction and increase trust in classifying particle physics events.



(a)



(b)

Figure 2. Uncertainty estimation of well-classified and misclassified testing set, using $D_{rate} = 0.2$. (a) shows high predictive entropy values, indicating average $\widehat{p}(\boldsymbol{y}|\boldsymbol{x}, \mathcal{D}_{train}) \approx 0.5$, and (b) depicts uncertainty values around zero, showing consistent predictions during the $T = 500$ forward passes.

The code is available in the GitHub repository: https://github.com/rpezoa/MCDropout_HEP_classif.

6. Acknowledgements

We gratefully acknowledge the financial support from ANID FONDECYT Postdoc Project N. 3190740 and ANID PIA/APOYO AFB230003. Powered@NLHPC: This research was partially supported by the supercomputing infrastructure of the NLHPC (ECM-02).

References

[1] Aishik Ghosh, Benjamin Nachman, and Daniel Whiteson. Uncertainty-aware machine learning for high energy physics. *Phys. Rev. D*, 104:056026, Sep 2021.
[2] Yarin Gal. *Uncertainty in Deep Learning.* PhD thesis, Department of Engineering, University of Cambridge, September 2016. Gonville and Caius College.
[3] Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. Dropout: a simple way to prevent neural networks from overfitting. *The journal of machine learning research*, 15(1):1929–1958, 2014.
[4] Pierre Baldi, Peter Sadowski, and Daniel Whiteson. Searching for exotic particles in high-energy physics with deep learning. *Nature communications*, 5(1):1–9, 2014.
[5] Armen Der Kiureghian and Ove Ditlevsen. Aleatory or epistemic? does it matter? *Structural safety*, 31(2):105–112, 2009.
[6] Yarin Gal and Zoubin Ghahramani. Dropout as a bayesian approximation: Appendix, 2015.
[7] Joaquin Vanschoren, Jan N Van Rijn, Bernd Bischl, and Luis Torgo. OpenML: networked science in machine learning. *ACM SIGKDD Explorations Newsletter*, 15(2):49–60, 2014.
[8] Pengzhen Ren, Yun Xiao, Xiaojun Chang, Po-yao Huang, Zhihui Li, Xiaojiang Chen, and Xin Wang. A comprehensive survey of neural architecture search: Challenges and solutions. 54(4), 2021.
[9] Haifeng Jin, Qingquan Song, and Xia Hu. Auto-keras: An efficient neural architecture search system. In *Proceedings of the 25th ACM SIGKDD international conference on knowledge discovery & data mining*, pages 1946–1956, 2019.
[10] Alejandro Barredo Arrieta, Natalia Díaz-Rodríguez, Javier Del Ser, Adrien Bennetot, Siham Tabik, Alberto Barbado, Salvador García, Sergio Gil-López, Daniel Molina, Richard Benjamins, et al. Explainable artificial intelligence (xai): Concepts, taxonomies, opportunities and challenges toward responsible ai. *Information fusion*, 58:82–115, 2020.