Tackling the muon identification in water Cherenkov detectors problem for the future Southern Wide-field Gamma-ray Observatory by means of Machine Learning

B.S. González[a,b], R. Conceição[a], M. Pimenta[a], B. Tomé[a], A. Guillén[b]

[a] *Laboratório de Instrumentação e Física Experimental de Partículas (LIP) - Lisbon, Av. Prof. Gama Pinto 2, 1649-003 Lisbon, Portugal and Instituto Superior Técnico (IST), Universidade de Lisboa, Av. Rovisco Pais 1, 1049-001 Lisbon, Portugal*
[b] *Computer Architecture and Technology Department, University of Granada, Granada, Spain*

Abstract

This paper presents several approaches to deal with the problem of identifying muons in a water Cherenkov detector with a reduced water volume and 4 PMTs. Different perspectives of information representation are used and new features are engineered using the specific domain knowledge. As results show, these new features, in combination with the convolutional layers, are able to achieve a good performance avoiding overfitting and being able to generalise properly for the test set. The results also prove that the combination of state-of-the-art Machine Learning analysis techniques and water Cherenkov detectors with low water depth can be used to efficiently identify muons, which may lead to huge investment savings due to the reduction of the amount of water needed at high altitudes. This achievement can be used in further research to be able to discriminate between gamma and hadron induced showers using muons as discriminant.

Keywords: Machine learning, Neural Networks, Feature engineering, Gamma rays, Cosmic rays, Information representation, Signal processing

1. Introduction

The irruption of Deep Learning (DL) models using Convolutional Neural Networks (CNNs) has been earth-shaking for the research in machine learning by setting the standard outperforming human capabilities in many tasks. It all started in image applications with [1] but, rapidly, these methods have been applied to many other domains [2, 3]. Outside the image landscape, the principal application of these models has been biomedical signals like ElectroCardioGrams (ECG) and ElectroEncephaloGrams (EEG) [4, 5, 2].

For the concrete case of analysing cosmic rays measurements, DL has been successfully applied to a few experiments. Some of them resemble the application done in image processing as images can be generated in a quite straight-forward way from the experimental devices as in Imaging Atmospheric Cherenkov Telescope Arrays (IACTAs). However, a different approach is required to process other type of inputs in other detectors, for instance, the reconstruction of the main characteristics of Extensive Air Showers (EAS) in observatories like HAWC or Pierre Auger [6, 7, 8] or the detection of neutrinos with the IceCube observatory [9].

In this research field, the detection of very-high-energy (VHE) gamma-rays is essential to investigate the sources of the incoming electromagnetic radiation produced by some of the most extreme, non-thermal, phenomena taking place in the Universe, e.g., fast-spinning neutron stars or supermassive black holes [10]. One of the possible techniques to indirectly detect gamma-rays are the EAS arrays, which cover large areas with particle detectors at high altitude. This method reconstructs the main characteristics (e.g. energy or direction) of the primary gamma-ray by means of the detection of secondary particles produced during the air shower development. EAS arrays are able to perform long term observations of variable sources and allow the search for emissions in extended regions [11].

However, although the indirect methods are effective in the GeV-TeV region, an important disadvantage is the presence of a huge background from charged particles produced in showers originated by cosmic-rays, which is three orders of magnitude larger than the signal. Several techniques have been proposed to perform the gamma/hadron separation, that is, rejecting the hadronic background. At high energy, muons, a clear signal of hadronic interactions, begin to reach the ground in sufficiently high numbers so that they can be used to discriminate gamma from hadronically induced showers. Thus, the identification of muons in water Cherenkov detectors (WCD) is explored in this work.

Using light detectors, WCDs measure the Cherenkov light emitted by charge particles in water. The production of the Cherenkov light in WCDs is closely related to its depth. Since electromagnetic particles get attenuated at the top of the station, using a detector unit with enough depth ensures a good efficiency in the identification of muons. That is the case of HAWC observatory, which has a good discrimination power in single muons events (those events with only muons and no electromagnetic con-

tamination) [12, 13] by using 300 cylindrical WCDs with a depth of 5 meters and a diameter of 7,5 meters. However, it must be taken into account that at mountain altitudes the resources of liquid water are reduced, then, the detector size must be optimised to reduce the necessity of water at those altitudes. For this reason, in this paper, the problem is tackled from a new WCD design with low water depth and a good muon tagging efficiency. The proposed detector uses a set of four Photomultipliers Tubes (PMTs), whose positions inside the WCD have been optimised to maximise the signal asymmetric of muons vertically crossing the detector.

The challenge in this work is to tackle the problem of identifying if a muon has crossed the WCD by means of the PMTs' gathered information. Once it is known if there are muons in the event, this information could be used for an ulterior classification between γ and hadron induced showers which could be straight applied to new observatories. To do so, it is necessary to establish the whole machine learning pipeline as well as determine how the input information will be given to the models. Thus, the rest of the paper is organised as follows: in Section 2, the data for the analysis is described. Section 3 presents the different approaches considered to tackle the problem. Section 4 performs an experimental comparison of the previous approaches. Finally, the conclusions and a discussion are presented in Section 5.

2. Data description and preprocessing

The data analysed in this work has been simulated using CORSIKA (version 7.5600) [14] and the detector using the Geant4 toolkit (version 4.10.05.p01) [15, 16, 17]. The observation level for the simulations was set at 5 200 m above the sea level and a WCD array covering an area of $80\,000\,\mathrm{m}^2$ and a fill factor of $\sim 80\%$ was considered which are the operational characteristics that the future Southern Wide field Gamma-ray Observatory (SWGO) [18] will have.

The set of EAS generated was conducted using proton-based particles with energies between $4 - 6\,\mathrm{TeV}$ and a zenith angle $\theta_0 \in [5°; 15°]$. This range makes muons to fall vertically towards the station, making its presence clearer and adding some variability to the data set instead of having to introduce noise to make them more realistic. The azimuth angle of the primary particle was uniformly distributed over ϕ. The WCD unit used in this study is a cylinder 1.7 m height with a diameter of 4 m (see Figure 1a). The station has 4 PMTs at the bottom, equally separated and distancing from the WCD center by 1.5 m. These were the dimensions encountered to guarantee a good signal uniformity and maximise the signal asymmetry between PMTs for vertically entering muons, as for instance the case shown in Figure 1b.

2.1. Data curation and preprocessing

A crucial stage in the experimentation is to define a procedure to arrange the data registered by the detectors into a classical machine learning problem. First of all, for each EAS, the simulator considers all the stations that are hit by any particle. However, it is necessary to be cautious and select only the events that are interesting for our purpose. For instance, in the case that a muon doesn't cross the detector entirely, few Cherenkov light could be detected. To overcome possible unfavorable situations a threshold of 300 photoelectrons in the light collected is established when selecting the events.

The next step is to build the data sets necessary for the experimentation. As the muon identification is carried out at station level, to ensure the maximum fairness in our analysis, the EAS are separated beforehand to avoid having events of the same shower in different data sets.

Afterwards, the data is partitioned into three independent subsets:

- **Train**: necessary for setting a value for model hyperparameters.
- **Validation**: used to select the best final model. To this end, a 20% is taken from the training data set.
- **Test**: applied to perform an unbiased evaluation of the fittest models after the training/validation process.

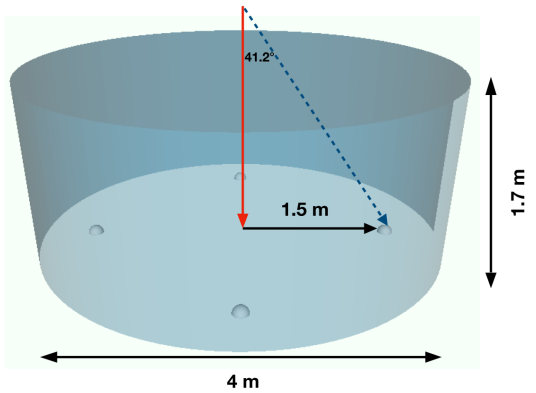
Table 1 details the number of instances of each partition.

	Data sets	
	Training	**Test**
EAS	2 000	1 637
S.M. stations	17 244	14 808
E.M. stations	340 007	289 354
Instances	357 251	304 162
Muonic prop.	4.83 %	4.87 %

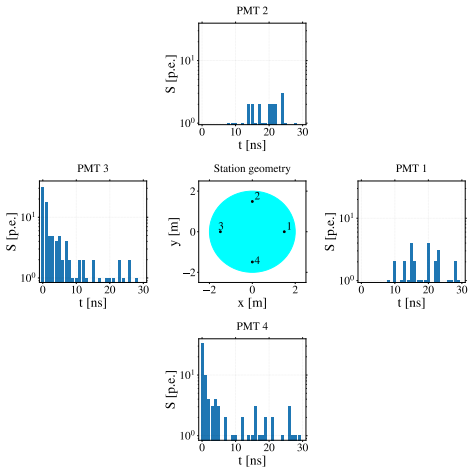
Table 1: Description of the data sets. Number of station events with *Single Muons* (S.M. stations) and without muons (E.M. stations) from proton-induced showers with $E_0 \in [4; 6]\,\mathrm{TeV}$ and $\theta_0 \in [5°; 15°]$.

Muons constitute a small fraction of the total number of particles that reach the Earth's surface. Thus, as shown in Table 1, the number of stations with muons per EAS is low (in comparison with those with electrons or photons). Since the subsets are partitioned by separating the EAS, the proportion of stations with muons will remain. For this reason, different preprocessing techniques have been explored in order to balance the classes before the training stage, which is a must if we want to identify any muon [19].

- **Random oversampling**: random repetition of the samples available from stations with muons.
- **Random undersampling**: random elimination of the samples available from stations without muons.



(a) Single-layered WCD design.



(b) Water Cherenkov detector crossed by a *Single Muon*. The WCD drawing is surrounded by the correspondent PMT signal time traces.

Figure 1: Scheme of the single-layered WCD.

- **SMOTE** (*Synthetic Minority Over-sampling TEchnique*): creation of synthetic new samples of stations with muons [20].

3. Machine Learning algorithms: problem approach and model design

This section presents the different approaches considered to solve the problem as well as the details regarding the design methodology for each type of technique.

3.1. Problem encoding and feature engineering

As it is desired to know the probability that the WCD analysed has been crossed by a muon, instead of approaching the task as a classification problem (where output would be {0,1}), regression models are the adequate to provide such information. By having the probability, it will be possible to tune the model towards a more sensible or specific output by defining thresholds [21].

The nature of the data allows using different techniques when handling the information collected by the PMTs. In this paper, the following strategies have been considered:

- *Signals* ($\vec{S} = \vec{S_1}, \vec{S_2}, \vec{S_3}, \vec{S_4}$): it corresponds to the signal trace captured of the event during 30 ns, which allows us to explore the temporal features of the events and may be crucial for identifying, in future work, if more than one particle has crossed the station. This is specially interesting considering that analysing the whole signal could make possible to identify reflections of light generated by the tank walls. Each $\vec{S_i}$ is highly dependant on the total energy of the event, therefore, the raw signals are normalised by the sum of the integrals of all PMTs during 30 ns.

- PMT Integrals (I_1, I_2, I_3, I_4): instead of having such a large amount of data, it is possible to project the traces into a number by computing the integral of the signal recorded by each detector. By doing this, it is possible to consider classical approaches which would not been able to process the raw signals described above. As there are 4 PMTs recording the Cherenkov light, it is still possible to conserve some spatial information although the temporal component is lost. For the sake of clarity in the muon identification, as they reach the PMTs before the electromagnetic component, a threshold of 10 ns has been set to compute these values.

- Total light I_t: it represents the sum of the PMT integrals, therefore, $I_t = \sum_{i=1}^{4} I_i$. The reason to consider I_t is due to the intrinsic uniform properties of muons, which are confined in a concrete range of values of this variable.

- Asymmetry A_T: Cherenkov light in muonic events is mainly produced in a reduced area of the WCD, whilst in electromagnetic events the light is spread inside the tank. Under these circumstances, it is to be expected that a large asymmetry is present in muonic events when comparing the amount of light collected by the PMT with the maximum signal and the one in front of it. Let A_T be the total asymmetry of an event, defined by equation (1), and let I_{max} and I_{opp} be the integral of the signals in the PMT with maximum signal and the one in front of it respectively, so that $A_T \in [0, 1]$.

$$A_T = \frac{I_{max} - I_{opp}}{I_{max} + I_{opp}} \tag{1}$$

- PMT Integrals Normalization ($In_1, ..., In_4$): Although the integral by itself could be a good projection of the event, it is possible to engineer the information encoding by normalising the amount of signal of each PMT considering the total signal recorded to obtain a magnitude independent range of values. Thus, the integrals for each PMT are normalised using I_t. Let, $In_i = I_i/I_t$.

As a summary, Table 2 shows the input given to the models and its classification.

3.2. Model design methodologies

This subsection first describes a first approach on CNNs application to the problem using the traces available. Afterwards, it is detailed the process followed with boosted tress which are not able to process the raw signals. Finally, in this section, it is commented the possibility of combining those different approaches by means of an ensemble.

3.2.1. Models using Convolutional Neural Networks

To process the signal trace recorded by the PMTs 1-dimensional convolutional neural networks (CNNs) have been designed. As stated previously, CNNs are one of the state-of-the-art machine learning algorithms, whose relevant applications in different fields have proven its great potential. The algorithm is composed of two blocks of hidden-layers: a first set of convolutional layers extract the features from the raw data by means of the convolution, afterwards, a second set of hidden-layers uses the previous information to perform the classification or regression and provide the output. Therefore, CNNs are capable of fusing the tasks of feature extraction from complex data (signals or images) and classification/regression in a single process, which is the main advantage of using this algorithm [22]. Before hyperparameters can be tuned, it is required to find a way of introducing such input. As stated previously, different kind of inputs appear after analysing the information captured by the detectors. With the use of this algorithm we intend to squeeze all the information available, both spatial and temporal.

On the one hand, convolutional layers will be used to extract characteristics from the signal traces. For each considered station we will have 4 signal traces to store (one per PMT). In order to respect as much as possible the temporal information that the problem provides, the approach consist on using a channel for each of these signals, in this way, each signal trace value will be attached to the nanosecond when it was registered by its position in the array which stores the data.

On the other hand, after focusing on the temporal characteristics, we must consider how to feed the CNN with the other engineered variables as well. To this end, the second set of hidden layers is fed with the rest of the variables, that is, the dense layers which are used to perform the regression.

When designing a CNN, one of the critical stages is the definition of the convolutional layers. Nowadays, no standard approach exists to set the configuration of a CNN. There are some rules of thumb for image classification, but, since the problem tackled in this paper belongs to another domain, those rules could not be adequate. To overcome this problem, the hyperparameter space has been clustered based on the experience tackling similar problems like [23, 24], so that a finite number of values are evaluated. Finally, after evaluating different topologies with the validation data set, the fittest one was composed of three convolutional layers which were enough to cover the inner complexity of the data and extract relevant features from the raw data. ReLU [25] activation function was used in these layers. The size of the filters was selected according to the mean of the signal traces registered, which unveils that the first pulse of Cherenkov light usually arrives within the first 3 nanoseconds of signal. Thus, small filters are proposed. The fittest configuration used filters of size 2, with a stride = 2 in the first layer to highlight the arrival of the direct Cherenkov light. The number of filters in these convolutional layers was 20, 15 and 10. Pooling layers have been discarded since they reduce the average specificity, though higher average sensitivity was achieved when using them.

Regarding the architecture of the CNN, the configuration chosen is: three dense layers (with 30, 15 and 10 neurons) and a final output layer were used to perform the regression from the previous extracted characteristics and the introduced variables. Sigmoid activation function was used for the three dense layers and for the final neuron. ReLU and Tanh activation functions were considered as well but performance obtained was smaller.

All CNNs configurations (using different input variables) were trained using Adam learning algorithm [26], which is a stochastic gradient-based method and has been broadly used in many deep learning applications. The adjustment of its parameters was done following those recommended in [26], that is: betas = (0.9, 0.999) and epsilon = 10^{-7} and learning rate of 0.001. The models were trained during 200 epochs with a batch size of 512, smaller values were showing high differences between training and validation and higher values increased training errors. Regarding the loss function to be minimised after each epoch it was chosen Mean Squared Error (MSE).

Figure 2 shows the topology of the fittest CNN found using temporal and spatial features, whose parameters have been detailed previously.

3.3. Models using decision-tree based algorithm

Two state-of-the-art decision-tree based algorithms have been used to process exclusively the signal traces integrals which enable us to study the spatial information present in the WCDs: *Extreme Gradient Boosting* (XGBoost) and *Random Forest*. Random Forest [27] is an algorithm that combines several decision trees and arises from the modification of the *bagging* process after decorrelating the trees

Variable #	Variable Name	Variable Type	Meaning
1	$\vec{S}$	Simulated	PMTs' signal traces. First 30 ns of the signal trace of each PMT. Normalised using the total signal.
2	I_t	Engineered	WCD's total signal. Sum of the integrals of each PMTs' signal using 10 ns. Total Cherenkov light in the WCD.
3	$I_i, i = 1...4$	Engineered	Integral of the PMT's signal using the first 10 ns.
4	$In_i, i = 1...4$	Engineered	Integral of the PMT's signal using the first 10 ns. The variable is normalise by the WCD total signal.
5	A_t	Engineered	Asymmetry. A normalised difference of signal between the PMT with the greatest signal and its opposed in the WCD. Defined in equation (1).

Table 2: Variables used in the different approaches. Variable number 3 is actually 4 variables, one for each PMT in the tank. In the same way, variable number 1 consists of 4×30 values corresponding to the trace registered by each of the 4 PMTs.

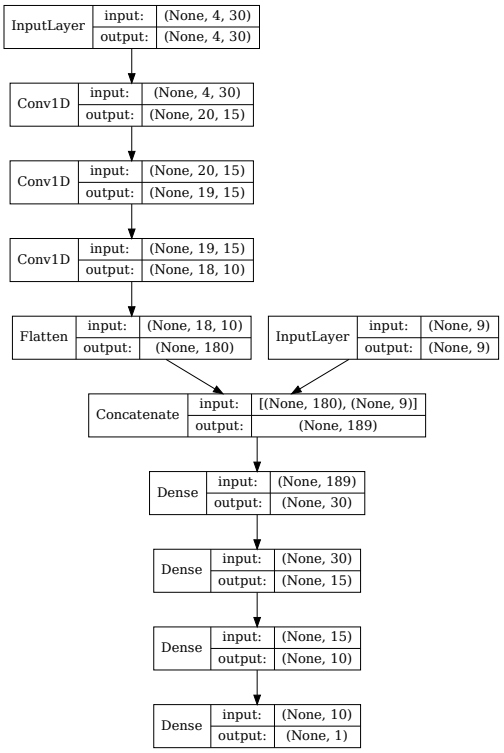


Figure 2: Topology of the fittest CNN found.

generated in the process. The bagging technique is meant to reduce the variance from the combination of models. During the process, the resampling technique *bootstrapping* is used. With this technique new data sets are created from the extraction of samples (with repetition) of the training data set. So, in a regression problem, a model is trained with each of the new training sets and the prediction of the total model will be the combination of the output of each of the trained models. The parameters to adjust after the training stage have been clustered and evaluated with the validation data set:

- Number of estimators that will be built. Evaluated values: [50,100,500,1000]. Best value found: 100.
- *Maximum depth* for the trees built. Evaluated values: [10,15,20]. Best value found: 20.

The second approach considered is the Extreme Gradient Boosting (XGBoost) algorithm [28, 29] which is based on the *boosting* method. This method is similar to bagging technique, but in this case the trees are built sequentially. That is, each of the trees are built using information from those previously built. In this way, "robust" trees are built from "weaker" ones. This is achieved by using the Gradient descent algorithm as the optimiser. Thus, during each iteration of the training process, the parameters of the weakest models are adjusted in order to minimise the loss function established for the problem (for instance, the root of the mean square error in the case of a regression), which will be given by the result of the model. Some parameters must be established when training this algorithm, for which we have followed the same strategy used previously. The parameters which were evaluated using the validation data set are the following:

- *Maximum depth* for the trees built. Evaluated values: [8,9,10]. Best value found: 10.
- *Learning rate*. Evaluated values: [0.01, 0.1, 0.2, 0.7]. Best value found: 0.2.
- *Objective*: define the kind of output that will be provided by the algorithm. In our case, a probability is wanted, thus, "binary logistic" is chosen.
- Scale_pos_weight: is used to control the balance of positive and negative weights, which is useful for unbalanced classes. Introducing the ratio between the classes N_{em}/N_{μ} which is present in the preprocessed training data set helped in improving the result.
- Maximum number of rounds when training. Selected value: 1000.
- Early stopping rounds: applied to find the optimal number of boosting rounds. If the error is not reduced in the validation data set after a fixed number of rounds the training will stop. Selected value: 20.

3.4. Ensemble approach

The previous approaches have a partial overlap in the variables used as input, however, there might be some features discovered by the CNN that tree models are not able to compute. At the same time, due to the smaller input data and the efficient optimization strategy, tree models might obtain more accurate results. With this two facts in mind, the possibility of combining both outputs in an ensemble is discussed.

Two ensembles are designed using the CNN as the main algorithm and a decision-tree based algorithm as the second algorithm, whose outputs will be obtained using the equation (2). Let $P_{\mu,\mathrm{pred}}(w)$ be the probability obtained by the ensemble and let $P_{\mu,\mathrm{m_1}}$ and $P_{\mu,\mathrm{m_2}}$ be the probabilities given by the two best models found for each approach respectively. Additionally, let us propose the weight w to adjust the influence that each algorithm will have when determining the ensemble probability, so that $P_{\mu,\mathrm{pred}}(w) \in [0,1]$.

$$P_{\mu,\mathrm{pred}}(w) = w \cdot P_{\mu,\mathrm{m_1}} + (1-w) \cdot P_{\mu,\mathrm{m_2}} \tag{2}$$

In principle, there are infinite possibilities when combining the outputs of the models depending on the weight w. However, we can adjust and get an optimum value of w in order to minimise the error. A possible approach to optimise the weight of each ensemble is to find the value which minimises the mean squared error after predicting the samples of the validation data set [30], that is, minimizing the equation (3), where n is the total number of samples in the validation data set, i is the index of the sample, $P^{(i)}_{\mu,\mathrm{pred}}$ is the probability obtained by the ensemble and $P^{(i)}_{\mu}$ the real "probability" and label of that sample (1 if there is a muon in the WCD or 0 otherwise). To solve the global optimization problem *Differential Evolution* algorithm from the *Scipy Optimize* Python package is implemented, which is a stochastic population based method [31].

$$\mathrm{MSE} = \frac{1}{n}\sum_{i=1}^{n}\left(P^{(i)}_{\mu,\mathrm{pred}}(w) - P^{(i)}_{\mu}\right)^2 \tag{3}$$

With the previous approach, when there is a disagreement between the models, one of them will have more weight than the other. This is translated in that, if one model predicts muon and the other does not the ensemble will predict muon unless the weight given to the other model is very small. This situation is undesirable as it is a requirement to be sure about when muons appear so two other approaches to combine the model's outputs have been considered as well:

$$P_{\mu,\mathrm{pred}} = P_{\mu,\mathrm{m_1}} \cdot P_{\mu,\mathrm{m_2}} \tag{4}$$

$$P_{\mu,\mathrm{pred}} = \sqrt{P_{\mu,\mathrm{m_1}} \cdot P_{\mu,\mathrm{m_2}}} \tag{5}$$

$$P_{\mu,\mathrm{pred}} = \frac{1}{\sqrt{2}}\sqrt{P^2_{\mu,\mathrm{m_1}} + P^2_{\mu,\mathrm{m_2}}} \tag{6}$$

By using the product, in case of clear disagreement, the ensemble will not provide a high probability. This value will be high only when both models agree. The ensemble of equation (6) was designed with the aim of giving greater importance to the algorithm that provides the highest probability.

4. Experiments and discussion

This Section will show first, how relevant are the new variables proposed in Section 2. Afterwards, the performance of the ensemble models is analysed. Finally, an experiment on solving the classification task using the probability is presented with the models presented in the paper.

As stated in Section 2, the data set is highly imbalanced so, after making some trials with the cited strategies, the best one (evaluated using validation error) was Random oversampling with ratio $N_\mu/N_{\mathrm{e.m.}} = 0.5$, where N_μ and $N_{\mathrm{e.m.}}$ are the number of stations with muons and without muons respectively. The following experiments, unless stated explicitly, are made after applying this procedure to the training data set.

Regarding the implementation details, Python 3.7 using SciKit-learn [32], Numpy [33] and Keras Framework [34] were used to develop the neural networks and random forest. Regarding the XGBoost, the implementation available at [35] was chosen to carry out the experiments. Due to the high volume of data (each EAS can require up to hundreds of MiB), the infrastructure used to compute was a cluster at LIP with the characteristics: Intel(R) Xeon(R) Silver 4110 CPU @ 2.10GHz processors, 46 GiB of RAM and two GPUs: NVIDIA GP102 TITAN Xp, NVIDIA TU102 GeForce RTX 2080 Ti.

For the sake of reproducibility and transparency, all the code and data are available at `https://github.com/borja-sg/Muon-identification-WCD`.

4.1. Analysis of engineered variable's importance and approximation error

To determine how important the variables are, two methods were applied. The first one is based on the ranking that the XGBoost algorithm builds during the training stage. The second is based on the approximation error achieved by each model when training with different subset of variables (given the fixed model architecture of Section 3.2.1).

Figure 3 shows the weights that the XGBoost has assigned after carrying out the training. The most important variables are the integrals after the normalisation process followed by the new proposed index for the asymmetry. For the CNN approach, in Table 3 it is shown the mean Root Mean Squared Error (RMSE, $\sqrt{MSE}$) obtained after performing a cross validation with 5 folds. By doing this, it is guaranteed that the process is not biased by the initial shuffle to select train and validation data. As expected, the cross validation does not affect the model's

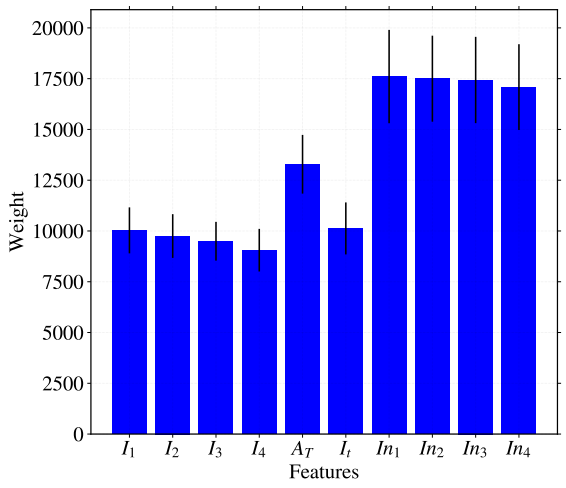


Figure 3: Mean feature importance given by XGBoost using 5 different seeds to select train and validation data sets. The black lines correspond to the standard deviation after training with the different seeds. Importance type: *weight*, which is the number of times the feature was used to split the data across all trees.

learning capabilities. In this case, the combination of variables that provides the best validation results is the one that uses all the proposed variables but the asymmetry. It is fair to say that the variables proposed always improved the approximation capabilities of the CNN architecture as the CNN using only the raw normalised signals is the one that provides the worst validation error. Table 3 also shows the error obtained by the tree based models. Although these models are able to learn properly the data, they tend to overfit, obtaining worse validation and test errors. Another reason to see some difference between training and validation might probably be due to the fact that the training data was balanced meanwhile validation and test are not.

4.2. Comparing the ensemble approach

From the previous experiment, we have seen that the features proposed were able to improve the probability prediction. Once the best models have been identified, it is time to see if they can combine knowledge assembling them. Table 4 shows the obtained RMSE for the different strategies towards creating the ensemble's output. The two models were chosen considering the best RMSE in validation for the models using the PMT signal's (CNNs) as well as for the tree based models. It is interesting to observe how, due to the overfitting of the XGBoost, it receives much more weight than the CNN although it performs worse in validation. The weight computation is also affected by the training data preprocessing and it's important to notice how the balancing still seems necessary as it provides better validation and test errors. The most remarkable fact of this table is that, regardless of the strategy used to combine both model's outputs, it always improves the results obtained isolated. Taking a close look, it is easy to conclude that when combining probabilities, it is better to perform the product than the weighted sum. The product of both probabilities seems the most adequate as it provides significant improvement in all metrics.

4.3. Tank Classification

Once it was obtained the most accurate model to predict the probability of having a muon in the analysed WCD, this experiment will show if this prediction is reliable enough to estimate whether muon has crossed the detector or not. In this task of saying if a tank has been crossed by a muon, it has to be set a threshold to decide if the given probability is high enough. If the value is higher than the threshold, the tank is classified as positive. Otherwise, it is considered as with no muon. To determine the threshold, a loop applying cuts to the obtained probability was used with a resolution of 0.01 and choosing the best one according to a metric.

Four classification metrics have been used: Area under the *Receiver Operating Characteristic* (ROC) curve, F1-score, Accuracy, and an Ad-hoc criterion. The first considered is based on the (ROC) curves of each proposed model to explore which ratio between the *True Positive Rate* (TPR) and the *False Positive Rate* (FPR) is suitable to determine whether there is a muon in the stations. Considering using this information in an ulterior system to classify γ/hadron EAS is necessary to ensure a low rate of false positive rate. In such a task, the ROC curves can be effective to analyse the behavior of the models when selecting different thresholds. In addition, the *Area Under the Curve* (AUC) will allow us to compare the overall performance of the models. Figure 4 shows the results obtained for each model for the two test data sets. According to the results obtained, there is a match between the CNN and the ensemble obtaining and excellent result for *Single Muons*. XGBoost has a good performance but it does not achieve as good results.

For the other three metrics, Tables 5 and 6 show the results for the three models. Table 5 shows results for the thresholds that maximise F1-score and Accuracy metrics respectively for the balanced Train data set. In this case, the ensemble outperforms in accuracy and F1-score, however, for the last metric, the CNN show a better behaviour for validation and test data sets probably due to the effects of the overfitting of the XGBoost, which shows very poor performances in both validation and tests.

The last metric is the criterion that could be used for an ulterior classification of the γ vs hadron EAS and to identify the type of particle. In this case, it is very important to not to missclassify the T_γ, this is, the tanks where a muon has not passed through and, it is enough to be able to identify some muons (T_μ). Thus, these values were computed and the cut has been set where the T_γ rate is near 99.9 %. As this implies new threshold values, the other two metrics have been computed as well to have a fair comparison of the models using different criteria. From Table 6 the performance of both the CNN and ensemble

Input Variables	Train	Validation	Test
$\vec{S}, I_i, In_i, I_t, A_T$	0.2118 (0.0049)	0.2361 (0.0058)	0.2341 (0.0059)
$\vec{S}, I_i, In_i, I_t$	0.2127 (0.003)	**0.2302 (0.0059)**	**0.2285 (0.0055)**
$\vec{S}, In_i, A_T, I_t$	0.2196 (0.0031)	0.2316 (0.0058)	0.2292 (0.0060)
$\vec{S}, I_i, A_T, I_t$	0.2136 (0.0034)	0.2340 (0.0036)	0.2318 (0.0044)
$\vec{S}$	0.2221 (0.0122)	0.2484 (0.0109)	0.2467 (0.0107)
$\vec{S}, A_T$	0.2166 (0.0076)	0.2447 (0.0104)	0.2438 (0.0109)
$\vec{S}, I_i$	0.2049 (0.0035)	0.2331 (0.0070)	0.2312 (0.0077)
$\vec{S}, In_i$	0.2194 (0.0068)	0.2425 (0.0108)	0.2409 (0.0106)
$\vec{S}, I_t$	0.2187 (0.0064)	0.2424 (0.0084)	0.2397 (0.0086)
$\vec{S}, I_i, I_t$	0.2098 (0.0032)	0.2362 (0.0050)	0.2340 (0.0060)
$\vec{S}, In_i, I_t$	0.2182 (0.0043)	0.2351 (0.0078)	0.2329 (0.0074)
XGBoost	**0.1052 (0.0121)**	0.2340 (0.0040)	0.2342 (0.0032)
RF	0.2301 (0.0010)	0.2606 (0.0013)	0.2600 (0.0004)

Table 3: Mean RMSE (and standard deviation) for models trained using 5 different seeds to select train and validation data sets.

Model 1	Model 2	Strategy	w optimisation	w	Train	Val.	Test
CNN $(\vec{S}, I_i, In_i, I_t)$	XGB	$w \cdot P_{\mu,\mathrm{m}_1} + (1-w) \cdot P_{\mu,\mathrm{m}_2}$	Train (Balanced)	0.2323	0.1288	0.2092	0.2102
CNN $(\vec{S}, I_i, In_i, I_t)$	XGB	$w \cdot P_{\mu,\mathrm{m}_1} + (1-w) \cdot P_{\mu,\mathrm{m}_2}$	Train (Original, imbalanced)	0.1833	0.1285	0.2131	0.2141
CNN $(\vec{S}, I_i, In_i, I_t)$	XGB	$P_{\mu,m_1} \cdot P_{\mu,m_2}$	-	-	**0.1026**	**0.1872**	**0.1883**
CNN $(\vec{S}, I_i, In_i, I_t)$	XGB	$\sqrt{P_{\mu,m_1} \cdot P_{\mu,m_2}}$	-	-	0.1193	0.1897	0.1901
CNN $(\vec{S}, I_i, In_i, I_t)$	XGB	$\frac{1}{\sqrt{2}} \cdot \sqrt{P^2_{\mu,m_1} + P^2_{\mu,m_2}}$	-	-	0.1623	0.2133	0.2126

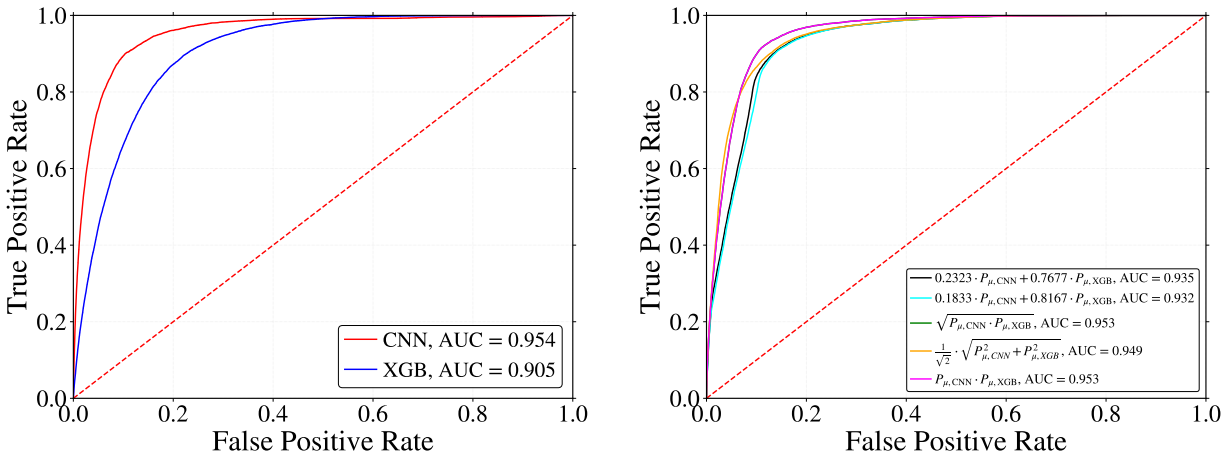
Table 4: RMSE error for different ensemble strategies combining the best CNN with XGBoost. Notice that the approaches using the product do not require the w optimisation parameter.

remains quite similar although in the test sets, for the F1-score, the ensemble does an improvement in comparison with the other approaches. For the Ad-hoc criterion, we have two non-dominant solutions, this is, from the two objectives that we are aiming, T_μ and T_γ, there is no model that perform better in both at the same time. If a more accurate T_μ is desired, the ensemble should be chosen but a 0.2% of error will be increased when misclassifying T_γ.

After comparing the different models using several metrics, it is quite obvious that the best approach in general is the one that receives the input the traces and the engineered variables, this is, CNN. From this fact, it is possible to devise that the temporal and spatial components of the signals are crucial when trying to discriminate the presence of muons. One reason for this might be the fact that the traces are seeing 30 ns from the signal that includes the direct light and the first reflection, meanwhile the integral is computed considering the first 10 ns to avoid too much interference with the electromagnetic component. Thus, a tank that maximises reflections should be considered when designing the new observatory. Another interesting element to discuss is inability of models to learn the engineered variables. This makes sense as the engineered variables are almost impossible to be obtained just by performing convolutions but it is a call of attention to DL practitioners so they do not rely uniquely on the model but they have to put special attention to the problem domain and the expert knowledge available.

5. Conclusions

The muon identification is a hot topic in cosmic ray research due to its implications in γ-ray observation, multi-messenger studies and Ultra High Energy Physics. Within this context, a new observatory based on indirect methods, like Water Cherenkov Detectors, is being studied. This paper has dealt with the problem of determining the probability of muon presence in the signal recorded by a water Cherenkov detector prototype design. To do so, several machine learning techniques have been compared and new engineered variables have been proposed. The results have shown that the models that behave the best considering the characteristic of the problem are the ones that process the complete signal and that introduce the variables engineered by researchers in Physics and Computer Science by means of a combination of Convolutional Layers with Dense layers. Although the improvement is small, it is magnified when using an ensemble of models combining CNNs and XGBoost. Therefore, the possibilities that this new methodology open to identify Extensive Air Showers with γ-rays are wide open and allow evaluating the different detector designs before deploying the future observatory. Future studies will concentrate in considering other convolutions for the CNN and to consider methods that do not suffer from overfiting as much as the XGBoost does with these data sets.



(a) Test using CNN and XGB.

(b) Test using ensembles.

Figure 4: Result of the ROC curves and AUC for test data set (a) single models (b) ensembles.

Accuracy					
Algorithm	Threshold	Train	Train (Original, imbalanced)	Validation	Test
CNN	0.27	94.72	94.04	92.70	92.73
XGBoost	0.83	99.67	99.59	94.60	94.48
CNN · XGBoost	0.18	96.03	97.74	94.68	94.66
F1-score					
Algorithm	Threshold	Train	Train (Original, imbalanced)	Validation	Test
CNN	0.27	92.39	60.81	52.16	52.36
XGBoost	0.83	99.51	95.95	27.86	27.89
CNN · XGBoost	0.16	93.92	78.54	50.14	50.44

Table 5: Maximum values for two error measures when classifying the appearance of muons in the WCDs. This is the best value obtained for any possible threshold to determine the class separation.

Accuracy								
Algorithm	Threshold	Train		Train (Original)		Validation		Test	
CNN	0.96	70.70		95.59		95.61		95.53	
XGBoost	0.94	95.69		99.14		95.04		94.97	
CNN · XGBoost	0.82	87.05		98.04		95.47		95.48	
F1-score									
Algorithm	Threshold	Train		Train (Original)		Validation		Test	
CNN	0.96	22.13		21.26		22.39		20.71	
XGBoost	0.94	93.10		90.47		16.10		16.41	
CNN · XGBoost	0.82	75.94		75.08		21.67		23.04	
Ad-hoc criterion									
Algorithm	Threshold	Train		Train (Original)		Validation		Test	
		T_μ	T_γ	T_μ	T_γ	T_μ	T_γ	T_μ	T_γ
CNN	0.96	12.49	99.81	12.35	99.81	13.13	99.79	11.99	99.81
XGBoost	0.94	87.25	99.91	84.11	99.91	9.86	99.36	10.14	99.31
CNN · XGBoost	0.82	61.33	99.90	61.24	99.90	12.99	99.65	13.89	99.66

Table 6: Error measures (accuracy and F1-score) and percentage of correct classification (Ad-hoc criterion) when classifying the appearance of muons in the WCDs. These are the best values to ensure a minimum of $T_\gamma \sim 99.9\%$ using the balanced Train data set.

Acknowledgements

We would like to thank to A. Bueno for all the support and useful discussions during the development of this work. The authors thank also for the financial support by OE - Portugal, FCT, I. P., under project PTDC/FIS-PAR/29158/2017. R. C. is grateful for the financial support by OE - Portugal, FCT, I. P., under DL57/2016/cP1330 /cT0002. B.S.G. is grateful for the financial support by grant LIP/BI - 14/2020, under project IC&DT, POCI-01-0145-FEDER-029158.

References

[1] Y. LeCun, B. Boser, J. Denker, D. Henderson, R. Howard, W. Hubbard, L. Jackel, Handwritten digit recognition with a back-propagation network, Advances in neural information processing systems 2 (1989) 396–404.

[2] S. Kiranyaz, O. Avci, O. Abdeljaber, T. Ince, M. Gabbouj, D. J. Inman, 1d convolutional neural networks and applications: A survey, Mechanical Systems and Signal Processing 151 (2021) 107398. doi:https://doi.org/10.1016/j.ymssp.2020.107398.
URL http://www.sciencedirect.com/science/article/pii/S0888327020307846

[3] L. Erhan, M. Ndubuaku, M. Di Mauro, W. Song, M. Chen, G. Fortino, O. Bagdasar, A. Liotta, Smart anomaly detection in sensor systems: A multi-perspective review, Information Fusion 67 (2021) 64 – 79. doi:https://doi.org/10.1016/j.inffus.2020.10.001.
URL http://www.sciencedirect.com/science/article/pii/S1566253520303717

[4] A. Peimankar, S. Puthusserypady, Dens-ecg: A deep learning approach for ecg signal delineation, Expert Systems with Applications 165, cited By 0 (2021). doi:10.1016/j.eswa.2020.113911.
URL https://www.scopus.com/inward/record.uri?eid=2-s2.0-85090839248&doi=10.1016%2fj.eswa.2020.113911&partnerID=40&md5=182b1e5063d318fa2da7cd4713910de2

[5] M. Manzano, A. Guillén, I. Rojas, L. J. Herrera, Deep learning using EEG data in time and frequency domains for sleep stage classification, in: I. Rojas, G. Joya, A. Català (Eds.), Advances in Computational Intelligence - 14th International Work-Conference on Artificial Neural Networks, IWANN 2017, Cadiz, Spain, June 14-16, 2017, Proceedings, Part I, Vol. 10305 of Lecture Notes in Computer Science, Springer, 2017, pp. 132–141. doi:10.1007/978-3-319-59153-7_12.
URL https://doi.org/10.1007/978-3-319-59153-7_12

[6] A. Guillén, A. Bueno, J. Carceller, J. Martínez-Velázquez, G. Rubio, C. T. Peixoto, P. Sanchez-Lucas, Deep learning techniques applied to the physics of extensive air showers, Astroparticle Physics 111 (2019) 12 – 22. doi:https://doi.org/10.1016/j.astropartphys.2019.03.001.
URL http://www.sciencedirect.com/science/article/pii/S0927650518302871

[7] A. Guillén, J. Martínez, J. M. Carceller, L. J. Herrera, A comparative analysis of machine learning techniques for muon count in uhecr extensive air-showers, Entropy 22 (11) (2020). doi:10.3390/e22111216.
URL https://www.mdpi.com/1099-4300/22/11/1216

[8] T. Capistrán, I. Torres, L. Altamirano, New method for gamma/hadron separation in hawc using neural networks, arXiv preprint arXiv:1508.04370 (2015).

[9] N. Choma, F. Monti, L. Gerhardt, T. Palczewski, Z. Ronaghi, P. Prabhat, W. Bhimji, M. Bronstein, S. Klein, J. Bruna, Graph neural networks for icecube signal classification, in: 2018 17th IEEE International Conference on Machine Learning and Applications (ICMLA), IEEE, 2018, pp. 386–391.

[10] A. De Angelis, M. Pimenta, Introduction to particle and astroparticle physics: multimessenger astronomy and its particle physics foundations, Springer, 2018.

[11] P. Assis, R. Conceição, M. Pimenta, B. Tomé, A. Blanco, P. Fonte, L. Lopes, U. B. de Almeida, R. Shellard, B. D. Piazzoli, et al., Lattes: a novel detector concept for a gamma-ray experiment in the southern hemisphere.

[12] A. Zuñiga-Reyes, A. Hernández, A. Miranda-Aguilar, A. Sandoval, J. Martínez-Castro, R. Alfaro, E. Belmont, H. León, A. P. Vizcaya, Detection of vertical muons with the hawc water cherenkov detectors and its application to gamma/hadron discrimination, arXiv preprint arXiv:1708.09500 (2017).

[13] A. Barber, D. Kieda, W. Springer, H. Collaboration, et al., Detection of near horizontal muons with the hawc observatory, in: ICRC, Vol. 301, 2017, p. 512.

[14] D. Heck, J. Knapp, J. Capdevielle, G. Schatz, T. Thouw, A monte carlo code to simulate extensive air showers, Report FZKA 6019 (1998).

[15] S. Agostinelli, J. Allison, K. a. Amako, J. Apostolakis, H. Araujo, P. Arce, M. Asai, D. Axen, S. Banerjee, G. . Barrand, et al., Geant4—a simulation toolkit, Nuclear instruments and methods in physics research section A: Accelerators, Spectrometers, Detectors and Associated Equipment 506 (3) (2003) 250–303.

[16] IEEE Transactions on Nuclear Science 53 No. 1 (2006) 270–278.

[17] Nuclear Instruments and Methods in Physics Research A 835 (2016) 186–225.

[18] Southern Wide field Gamma-ray Observatory (SWGO) swgo.org.

[19] B. S. González, R. Conceição, B. Tomé, M. Pimenta, L. J. Herrera, A. Guillen, Using convolutional neural networks for muon detection in WCD tank, Journal of Physics: Conference Series 1603 (2020) 012024. doi:10.1088/1742-6596/1603/1/012024.
URL https://doi.org/10.1088%2F1742-6596%2F1603%2F1%2F012024

[20] N. V. Chawla, K. W. Bowyer, L. O. Hall, W. P. Kegelmeyer, Smote: Synthetic minority over-sampling technique, J. Artif. Int. Res. 16 (1) (2002) 321–357.

[21] A. Guillén, C. Todero, J. C. Martínez, L. J. Herrera, A preliminary approach to composition classification of ultra-high energy cosmic rays, in: International Conference on Applied Physics, System Science and Computers, Springer, 2018, pp. 196–202.

[22] S. Kiranyaz, O. Avci, O. Abdeljaber, T. Ince, M. Gabbouj, D. J. Inman, 1d convolutional neural networks and applications: A survey, arXiv preprint arXiv:1905.03554 (2019).

[23] F. Carrillo-Perez, L. J. Herrera, J. M. Carceller, A. Guillén, Improving classification of ultra-high energy cosmic rays using spacial locality by means of a convolutional dnn, in: International Work-Conference on Artificial Neural Networks, Springer, 2019, pp. 222–232.

[24] M. Manzano, A. Guillén, I. Rojas, L. J. Herrera, Combination of eeg data time and frequency representations in deep networks for sleep stage classification, in: International Conference on Intelligent Computing, Springer, 2017, pp. 219–229.

[25] I. Goodfellow, Y. Bengio, A. Courville, Deep Learning, MIT Press, 2016, http://www.deeplearningbook.org.

[26] D. P. Kingma, J. Ba, Adam: A method for stochastic optimization, arXiv preprint arXiv:1412.6980 (2014).

[27] L. Breiman, Random forests, Machine learning 45 (1) (2001) 5–32.

[28] J. H. Friedman, Greedy function approximation: a gradient boosting machine, Annals of statistics (2001) 1189–1232.

[29] T. Chen, C. Guestrin, Xgboost: A scalable tree boosting system, in: Proceedings of the 22nd acm sigkdd international conference on knowledge discovery and data mining, 2016, pp. 785–794.

[30] M. Shahhosseini, G. Hu, H. Pham, Optimizing ensemble weights and hyperparameters of machine learning models for regression problems, arXiv preprint arXiv:1908.05287 (2019).

[31] R. Storn, K. Price, Differential evolution–a simple and efficient heuristic for global optimization over continuous spaces, Journal of global optimization 11 (4) (1997) 341–359.

[32] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, E. Duchesnay, Scikit-learn: Machine learning in Python, Journal of Machine Learning Research 12 (2011) 2825–2830.

[33] T. E. Oliphant, Python for scientific computing, Computing in Science Engineering 9 (3) (2007) 10–20. doi:10.1109/MCSE.2007.58.

[34] F. Chollet, et al., Keras, https://github.com/fchollet/keras (2015).

[35] xgboost developers, XGBoost Python Package, https://xgboost.readthedocs.io/en/latest/python/index.html (2020).