USING MACHINE LEARNING TO IMPROVE THE PERFORMANCE OF THE MU2E COSMIC RAY VETO

Mohit V. Srivastav
University of Virginia, Charlottesville, VA 22904

April 9, 2022

Abstract—**The Mu2e experiment at Fermilab is searching for the direct neutrinoless conversion of a muon to an electron. The experiment requires an extremely efficient Cosmic Ray Veto to detect cosmic muons and ignore electrons produced by them that can be confused with real direct conversions. We found that using a deep neural network improved upon the current Cosmic Ray Veto algorithm in terms of both the induced cosmic-ray background and the deadtime, yielding much promise for future exploration.**

Index Terms—**Mu2e, CRV, Neural Network, deadtime, induced background**

I. THE MU2E EXPERIMENT

Although the Standard Model of particle physics is well-tested in many areas, it appears to be incomplete. Even though Charged Lepton Flavor Violation (CLFV) (a transition between taus, muons, and electrons that does not conserve lepton family number) is not explicitly forbidden in the Standard Model of particle physics, it is greatly suppressed. Beyond the Standard Model, there exist predictions for observable CLFV rates, and searches of incidences of this have greatly increased in the past few years.[1]

One example of CLFV is the neutrinoless conversion of a muon to an electron within the Coulomb field of a nucleus. The Mu2e experiment, expected to start in 2025, will be looking for evidence of such a conversion. The experiment is a multinational project consistent of multiple labs and universities, mounted at the Fermi National Accelerator Laboratory near Chicago.[1]

To search for the direct neutrinoless conversion of a muon to an electron, a proton pulse hits a production target every 1.7 μs, where it will produce a beam of low-energy negatively charged muons which will be transported to and stopped at a series of thin foils known as the stopping target by the transport solenoid, which selects the particle's momentum and avoids a direct line of sight from production to the stopping target. At the stopping target, the individual muons will be captured in atomic orbits. The produced conversion electron's momentum, energy, and a variety of other attributes will be recorded by the tracker and calorimeter.[2] The apparatus is pictured in Fig. 1, and the production of a conversion electron from a muon hitting the stopping target in Fig. 2.

The signal window for conversion is approximately 1000 ns, with 700 ns to the next proton pulse. Within this time window, if the muon converts to an electron without emitting a neutrino, and the experiment detects it, then a direct muon-to-electron conversion has been found.

II. THE COSMIC RAY VETO

The occurrence of a neutrinoless conversion of a muon to an electron is extremely rare, if it occurs at all. A large barrier to achieving the desired sensitivity is the cosmic-ray background, induced by cosmic-ray muons. Each minute, approximately one cosmic-ray muon hits the Earth's surface per square centimeter. These muons are expected to produce, on average, one event a day (in this case, event means an electron that has the same characteristics as a real conversion electron) that cannot be distinguished from a successful conversion electron.[3]

The rate of such an occurrence has to be reduced by a factor of 10,000 in order to reduce the background to less than one event.[3] The solution to this problem is to surround the Mu2e detection apparatus with a detector that identifies cosmic-ray muons and rejects, or "vetoes", time windows around cosmic-ray muons that produce conversion-like backgrounds during the offline analysis.

The Cosmic Ray Veto (CRV), displayed in Fig. 3, consists of four layers of extruded polystyrene scintillators (a material that "scintillates", or emits light, when excited by ionizing radiation) counters with embedded wavelength shifting fibers, read out with Silicon Photomultiplier (SiPM) photodetectors.[3] These counters range from 900 to 6600 mm long, and have a cross section of 50×20 mm^2.[3] These detectors sense when charged particles enter the CRV, and will be used to detect muons and veto the events associated.

An track stub consists of at least three adjacent strips of the CRV with signals over a certain threshold within a 5 ns time window, signifying a real track localized in both space and time. These tracks are reconstructed using an algorithm developed by Dr. Ralf Ehrlich. Once a track stub is recorded within the CRV, the CRV reconstructs various numeric variables for the track stub using the algorithm by Dr. Ehrlich (such as position, light yield, and time), and the tracker itself also records variables pertaining to the interaction of the conversion electron, (such as the momentum, track quality, and the time recorded). Note that particles can enter the tracker without hitting the CRV and still be considered electron events.

Given a good track stub, a 200ns time window around the respective event recorded by the tracker will be vetoed during offline analysis to prevent consideration of any possible electrons produced by a cosmic-ray muon.

III. STUDY GOALS/STRATEGY

The CRV currently employs a "time window cut" to veto potential conversion electrons produced by cosmic-ray muons. All electron events outside this time window are classified as possible conversion electron (CE) events, and all those within the window are classified as background. Currently, the CRV has a very high efficiency rate in identifying cosmic-ray muons.

However, the CRV produces false track stubs from random coincidences due to the beam-induced background in the scintillator counters at high beam intensities. Deadtime, or the fraction of the time that the CRV spends vetoing events, maxes out at approximately 50% at the highest expected beam intensity with the current algorithm. Such behavior is not ideal, as the greater the deadtime the greater the running time of the experiment to achieve a given sensitivity.

The aging of the CRV will also reduce the capability of the CRV to suppress the background of cosmic-ray muons, as the light yields of the SIPMs decrease over time in the scintillator counters. A new algorithm that is

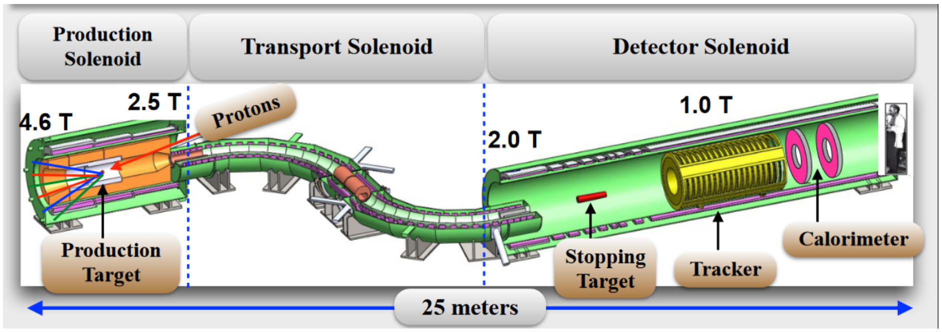

Fig. 1: The Mu2e apparatus, with the proton beam, transport solenoid, and detector solenoid separated and pictured. To prevent magnetic bottles which trap muons, the magnetic field of the solenoids is graded (hence the magnetic field strengths in Teslas in the figure).[2]

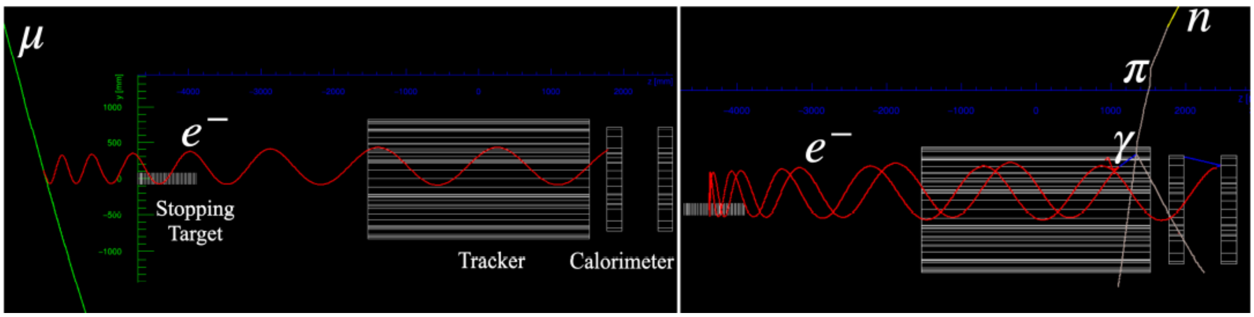

Fig. 2: Left: An event produced by a cosmic ray muon that knocks out a conversion-like electron in the Detector Solenoid. Right: A cosmic-ray neutron is incident from the upper right and interacts in the apparatus to produce an upstream-going electron. This electron reverses direction in the Detector Solenoid magnetic mirror and passes again through the tracker. This event is not vetoed by the CRV, because the neutron is a neutral particle, but can be vetoed by the tracker.

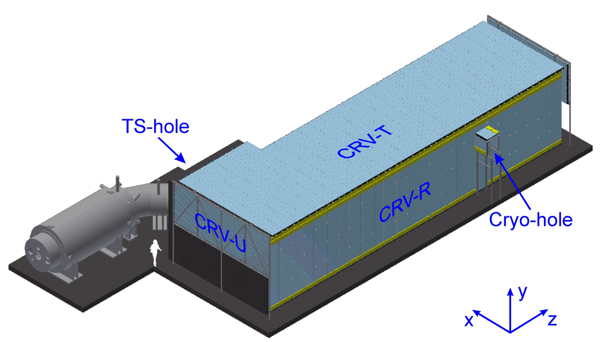

Fig. 3: A drawing of what the CRV will look like, along with its coordinate axes, with a human shown for scale below the CRV-U in white.

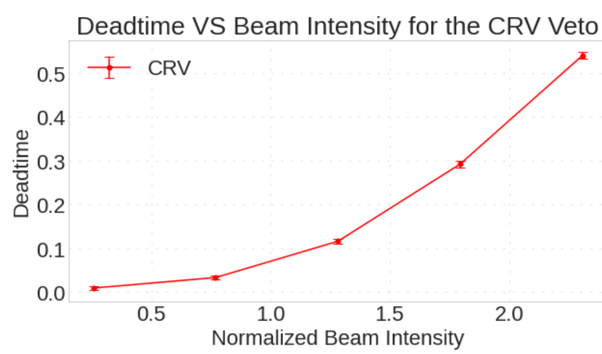

Fig. 4: The deadtime produced by the CRV vs the beam intensity. Beam intensity is relative to the nominal value of 3.9×10^7 protons per pulse.

both able to decrease the deadtime and reduce reliance on the light yield of the CRV is being investigated from many angles. Even though the current CRV veto algorithm is quite good at correctly identifying cosmic-ray muons, it is quite crude. It is the goal of this study to make a more sophisticated veto algorithm using the current algorithm developed by Dr. Ralf

Ehrlich supplemented by a deep neural network at the last step instead of the time window cut that is used currently .

IV. Simulated Datasets

Data were generated using GEANT4 (GEometry And Tracking Iteration 4), a tool for modeling the passage of elementary particles through matter.[4] The software for Mu2e instantiates a solid model of the Mu2e apparatus within the GEANT4 framework.[5] Three different data sets were generated by Dr. Yuri Oksuzian, a scientist at Argonne National Laboratory. The CE/noise dataset was produced by simulating conversion electrons, overlaid with noise produced by the Mu2e beam, which consists of neutrons and gammas. Overall, this dataset represented events that should not be vetoed, lest they contribute to the deadtime.

The CRY3 (Cosmic Ray Shower Generator) dataset corresponds to cosmic-ray muons that produced an electron-like track that was successfully reconstructed in the tracker. The CRY4 dataset was produced in nearly exactly the same way as the CRY3 dataset, but with a slightly updated generation algorithm with greater statistics, along with an updated shielding geometry. The sample also contained samples at differing light yields, in order for the algorithm to be tested on multiple light yields. The CRY3 and CRY4 samples represent events that should be vetoed, lest they contribute to the cosmic-ray induced background.

V. Important Definitions

The following section serves to define terms used for the rest of the paper.

A. Cut Terminology

There were a variety of cuts used in the study as well. These cuts, based off of the attributes of particles that enter the tracker, have been optimized over the course of multiple Mu2e studies to both help suppress cosmic-ray muons alongside other backgrounds, and to separate noise from real conversion electrons.

Cut	CRY3	CE/Noise	CRY4
No Cut	1,842,456	2,385,473	15,157,304
Loose Box Cuts	220,918	1,814,739	2,059,156
Box Cuts	133,307	1,806,498	1,280,300
Loose Cuts	75,175	1,579,431	786,184
Extended Momentum Cut	11,577	1,457,507	130,903
Physical Momentum Cut	847	893,575	9,863

TABLE I: A table of how many events remain for the three datasets after the given cuts defined. Each cut is more restrictive than the last going from top to bottom, as they all build off of each other.

The different cut sets are utilized for different purposes the stricter they are, and both the stricter and looser cuts were used throughout the study. In order of strictness, the cuts were as following: Loose Box Cuts, Box Cuts, Loose Cuts, Extended Momentum Cut, Physical Momentum Cut. The number of events remaining for each dataset after each type of cut is detailed in Table 1.

CRV Time Window Cut

The CRV Time Window Cut is applied after either of the Kinematical Cuts (for the actual experiment it would be the Physical Momentum Cut, but for this study it could be either of the cuts, depending on how good the statistics are). It is a 200 ns window determined by the time recorded in the CRV and the time recorded in the tracker. The window for the cut is $-50 < \Delta\mathrm{T} = \mathrm{T}_{\mathrm{Tracker}} - \mathrm{T}_{\mathrm{CRV}} < 150$ns. The current Mu2e veto algorithm finds track stubs in the CRV using an algorithm developed by Dr. Ralf Ehrlich, then uses the CRV Time Window Cut Veto, where CE-like events in the tracker that pass the CRV Time Window Cut are assumed to be cosmic-ray induced.

B. Deadtime

The deadtime of the experiment is the fraction of data removed by the action of CRV Veto. Part of the deadtime comes from the vetoing of cosmic-ray muons, whereas the other portion comes from false coincidences in the

CRV induced by neutrons and gammas from the exposed beam, which dominates. Real conversion electron events that the experiment is looking for may be vetoed if they look similar to a cosmic-ray muon. Similarly, the deadtime that was viewed for this study was the "harmful" deadtime, or the time spent vetoing false coincidences. Due to this, every reference to "deadtime" in this study refers to the deadtime induced by conversion electrons and noise.

This value must be minimized, as the greater the deadtime, the longer the experiment's anticipated runtime for a given sensitivity. Since the CRV vetoes the entire 200 ns window surrounding an event that it believes to be a cosmic muon, a deadtime of, for example, 50% would correspond to a 100% increase in the time required for the experiment, as the CRV would be unable to detect anything else during that window. Decreasing the deadtime of the experiment would then decrease the anticipated experimental runtime.

C. *Cosmic-Ray Induced Background*

The cosmic-ray induced background should be reduced to well less than one expected event over the course of the experiment. Cosmic-ray muons that pass through the CRV can produce a background in the form of a cosmic-ray induced electron. Due to the simulated nature of the data produced, the number of events had to be normalized to the expected "livetime" for one run of the Mu2e experiment, since the overall background over a run is heavily dependent on the livetime of the experiment. Run One of Mu2e's livetime is equal to 3.46×10^6 seconds.

The simulated livetime for the low energy muons was $1.36 \times 10^8 + 5.09 \times 10^7 = 1.869 \times 10^8$ seconds (the livetime was divided into two portions because the low energy muons were simulated in two different batches). The simulated livetime for the high energy muons was 3.64×10^6 seconds. The expected cosmic background, and its uncertainty, over the course of one run of the experiment is below:

$$\text{deadtime} = 1 - \frac{\text{Identified}_{\text{CE}}}{\text{Total Possible}_{\text{CE}}}, \tag{1}$$

$$\Delta\text{deadtime} = \frac{\sqrt{\text{Identified}_{\text{CE}}}}{\text{Total Possible}_{\text{CE}}}, \tag{2}$$

$$\text{bkg} = \frac{\text{\# cosmic events not vetoed}}{\text{livetime}_{\text{sample}}} \times \text{livetime}, \tag{3}$$

$$\Delta\text{bkg} = \frac{\sqrt{\text{\# cosmics not vetoed}}}{\text{livetime}_{\text{sample}}} \times \text{livetime} \tag{4}$$

VI. BUILDING THE MODEL

The Machine Learning model was built using the Keras[6] sequential programming interface for deep neural networks, which is a package for Python, the programming language used. This package allows for easily customizeable deep neural networks to be built and used. NumPy and Pandas, packages for data management and manipulation were also used throughout the study. The variables selected to be input into the model, referenced in Table 2, were at the recommendation of Dr. Yuri Oksuzian.

A. *The Existence of Two Models*

There are two types of events within all of the simulated data: events that produce hits in the CRV, and those that do not. Events that do not produce hits do not have CRV variables, but are still recorded by the tracker, and thus still have the variables from the tracker.

At the beginning of the study, there was only one model - no matter the type of event. However, the existence of CRV variables correlated with whether an event was a CE/Noise event or a cosmic muon, diluting the actual correlations of the variable within the dataset. This phenomenon ended up producing suboptimal results, and so a change was necessary.

Every dataset was then split in two, known as the "noCRV" and "CRV" datasets, with their corresponding models named the same. The "noCRV" dataset is that without CRV variables, and the "CRV" dataset is the one with

CRV Model Variables	noCRV Model Variables
Downstream number of hits	Downstream number of hits
Particle ID	Particle ID
Track quality	Track quality
Starting z-value of track	Starting z-value of track
Distance from z-axis	Distance from z-axis
Recorded CRV z position	N/A
Recorded CRV ΔT	N/A
PE Yield of primary scintillators	N/A

TABLE II: A table of variables used for the models. The CRV model used 3 more variables than the noCRV model.

CRV variables. Fig. 5 shows the two datasets and their correlations. An event without CRV variables, for example, could be an electron produced by a muon that went through the TS-Hole, depicted in Fig. 3, as the CRV does not cover there.

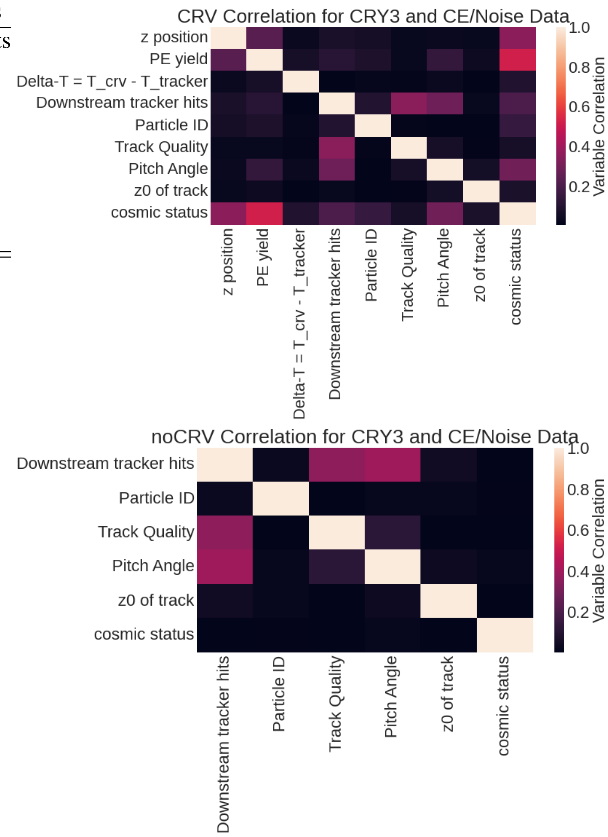

Fig. 5: Correlations for the CRV (top) and noCRV (bottom) data, which consists of the mixed CRY3 and CE/Noise data from Fig. 12.

B. Final Model Structure

The model structure that was chosen was a width of 4 times the number of input variables, with 8 hidden layers (layers that are neither input nor output layers) no matter the number of input variables. Inside of the middle 6 hidden layers there was a dropout rate of 0.2. The first hidden layer did not have dropout functionality because it was directly connected to the input layer, and the last one did not have dropout because it connected directly to the output. The batch size chosen was 100 events per batch for the CRV model, 10 events per batch for the noCRV model was, as the training sample size was much smaller for the noCRV sample.

The variables used for each model are detailed in Table 2, and a graphical rendition of the two models in Fig. 6.

VII. Training the Model

A. The Alpha Metric

Neural networks consist of weight matrices for every neuron, as a way to actually evaluate the inputs they are given. These matrices can be analyzed to provide insight into the model itself, without needing access to the training data.[7]

The α metric is an exponent determined by a model's weights such that the spectral density of the weight matrices, $\rho(\lambda)$, is approximately equal to λ^α for a given layer, where λ is an eigenvalue of the layer's weight matrix.

An α value less than 2 corresponds to the overtraining of a layer. As such, the average α over the every layer of the model was examined at the end of every epoch, and if the value of α_{avg} was less than 2.1, training would stop, as the model was close to being overtrained. The analysis of this metric was done using the "WeightWatcher" package for Python.[8]

B. Training Procedure

Both the noCRV and CRV models were trained on the CRY3 cosmic dataset alongside the training/validation subset of the CE/Noise sample, as stated before. The optimizer used

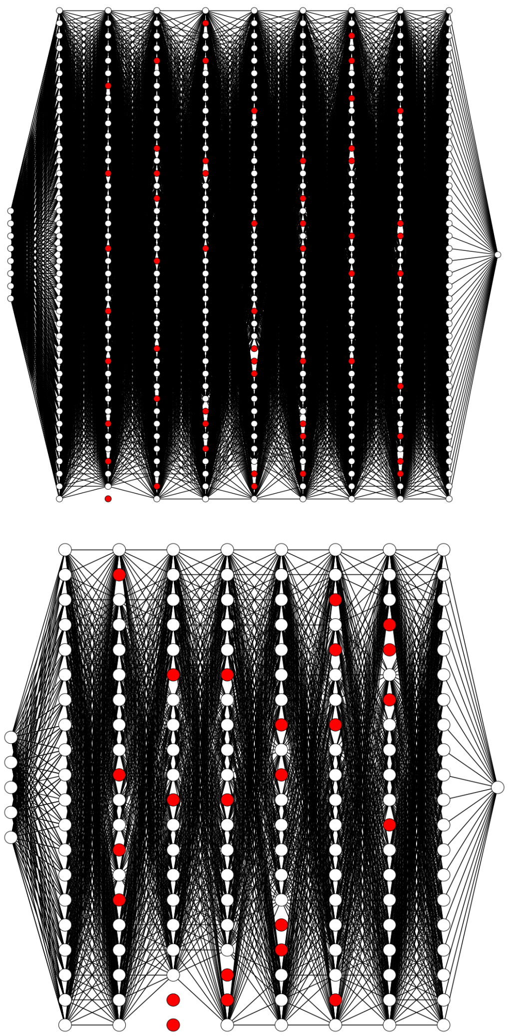

Fig. 6: The models for the CRV(top) and the noCRV(bottom) models. The red dots represent dropped out neurons, as a representation of how many neurons would be dropped out. The input is on the left and output on the right for each model. Note that the first and last hidden layer do not have dropout functionality. The dropout rate for the hidden layers with dropout is 0.2.

was the “adam” optimizer, which is a standard optimizer from the Keras package, and the kernel was initialized to a normal distribution.[6]

VIII. Model Predictions

The testing data, which was separated into the entire CRY4 dataset and the testing subset of the CE/Noise dataset, was scaled using the same values used to normalize the data for each variable to having a mean of 0 and a standard deviation of 1. These scalers put the testing data in the same relative scale, meaning that while the testing data would not have a mean of 0 and a standard deviation of 1, the data would be relative in those metrics to the data the model was trained on.

All of the predictions from the model are between 0 and 1, and a cutoff has to then be defined, where everything greater than the cutoff is classified as a cosmic, and everything less than the cutoff classified as a CE.

IX. Prediction Analysis

A. The Performance of the Two Models

The true positive rate compared to the false positive rate of the model is shown in the Receiver Operating Characteristic (ROC) curves in Fig. 7. The true positive rate is the rate at which cosmic-ray muon events are correctly identified, and the false positive rate is the rate at which CE/Noise events are identified as cosmic-ray muon events.

The performance of the noCRV model is significantly worse, as none of the variables in the noCRV model really correlate to cosmic status, making prediction quite difficult. The role of the noCRV model was simply to separate at least some of the cosmic events from the rest, not necessarily to optimize the separation. The real classification power came from the CRV model, where a majority of the cosmic muons were.

B. Providing a Classification Cutoff

The value of the classification cutoff is the largest factor in determining whether an event is a cosmic muon or a CE, as it is the final determining cut for the ML Veto Algorithm.

The final classification cutoff was crafted in such a way that the cosmic background would be minimized. Since the vast majority of the cosmic-ray muon events are handled by the CRV model, and a large number of

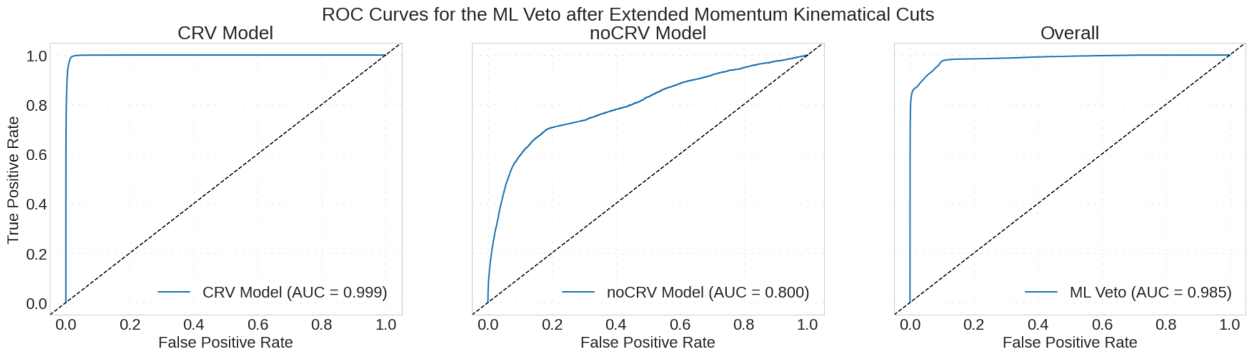

Fig. 7: The ROC curves for both models, with the CRV model on the left and the noCRV model in the middle. The ROC curve for the models combined is on the right.

CE/Noise events are handled by the noCRV model, the cutoff was established at 0.5 for the noCRV model. Other cutoff values for the noCRV model were explored, but it was settled that a value of 0.5 was the best.

For the CRV model, the cutoff value was set to match or improve upon the CRV Time Window Veto in terms of performance with regards to the cosmic-ray induced background. The three cutoffs looked at in depth were at values of 0.001, 0.005, and 0.010. A cutoff greater than 0.010 resulted in a cosmic background that was too large.

X. Physical Metrics

A. Cosmic Background

The ML Veto performs consistently better than the CRV Time Window Veto at the cutoff value of 0.001 in terms of both cosmic-ray induced background and deadtime. For every light yield except the highest at 17,000, the other two cutoff values selected does either better or within one Δ of the CRV Time Window Veto, with Δ being calculated using Eqs. 5 and 10. The cosmic-ray induced background vs light yield for the three cutoffs alongside the CRV Time Window Veto are plotted in Fig. 8 below.

One notable trait of the ML Veto is its success in regards to low and high energy muons. The ML Veto often performs worse than the CRV Veto for high energy muons for the higher cutoff values at higher light yields, and the ML Veto performs consistently better than the CRV Time Window Veto for low energy muons at that scale. This can be seen in the plots in Appendix B.

B. Deadtime

The deadtimes for the CRV Time Window Veto and the ML Veto Cutoff values are detailed below in Fig. 9. In reality, the beam produced by the production solenoid will have a set beam intensity, and Fig. 9 displays the deadtime for the ML Veto Algorithm versus the beam intensity. The ML Veto has less deadtime than the CRV Time Window Veto for every beam intensity except the lowest one. Even for the lowest beam intensity all the deadtimes are within each other's uncertainty bound.

There is a relatively large difference between the deadtime of the CRV Time Window Veto and the ML Veto for every cutoff available. This difference can be explained the by ΔT distribution for the ML Veto versus the current CRV Veto. Unlike the CRV Time Window Veto, the ML Veto lets through a number of CE events that are inside of the CRV Veto's time window while still identifying and vetoing enough cosmic muons to remain relevant as a viable veto for the CRV itself, which produces a much lower deadtime for the ML Veto even at the strictest cutoff value available.

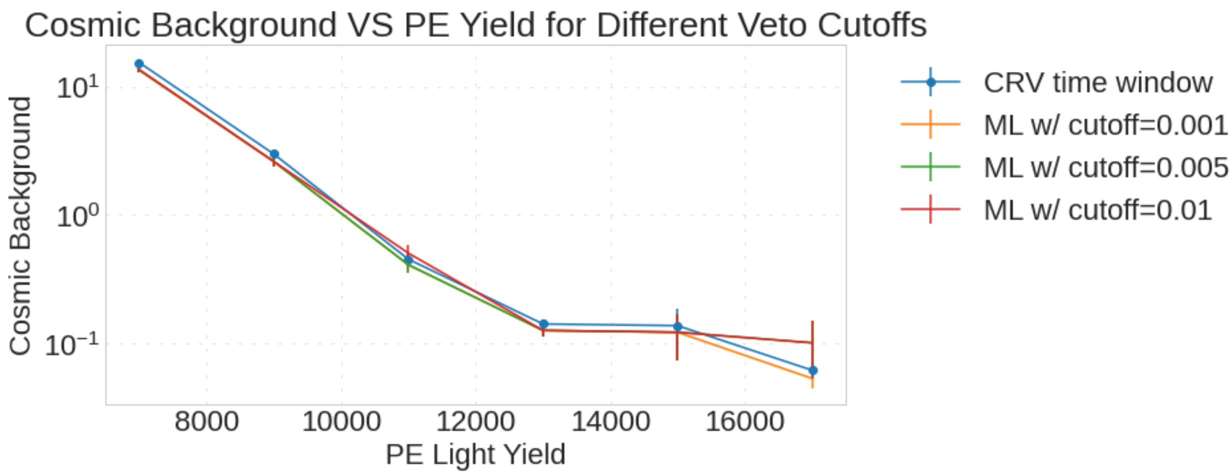

Fig. 8: The cosmic-ray induced backgrounds for different light yields. The noCRV model cutoff was kept at 0.5, with only the CRV cutoff changing.

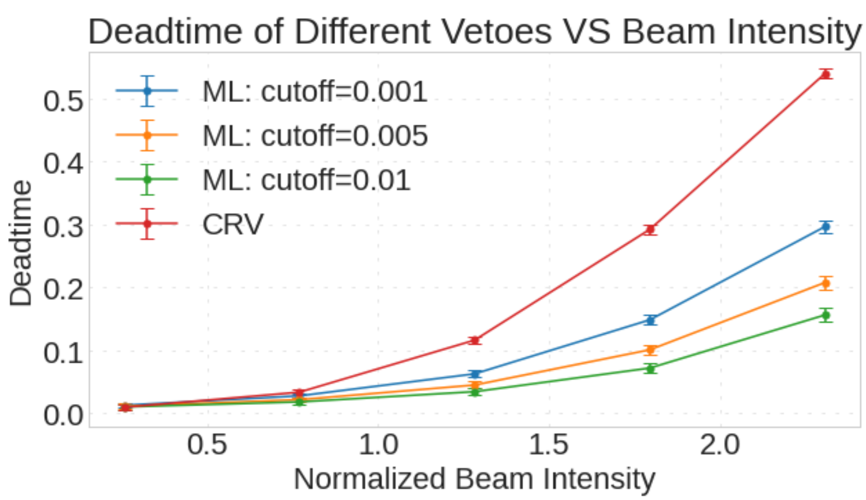

Fig. 9: The deadtime of the ML Veto and CRV Time Window Veto versus the beam intensity. Beam intensity is relative to the nominal value of 3.9×10^7 protons per pulse.

XI. Concluding Thoughts

A. Study Conclusion

This study is an investigation into whether a deep-neural network could perform better than the current CRV Time Window Veto. The ML Veto performs either the same or better than the CRV Time Window Veto in terms of both reducing the cosmic-ray muon induced background and the deadtime. For the strictest cutoff of 0.001 for the ML Veto, both the cosmic background and deadtime were improved upon at every light yield, which is very promising for the future of using machine learning for the Mu2e CRV.

B. Further Study

Any model developed needs to be further tested for robustness and further tested against other backgrounds, such as the DIO (muons which Decay In Orbit) background, and other such backgrounds. Even though the cuts applied remove a large portion of these backgrounds, it would still be pertinent to check. This would be the only way to actually beat the CRV Time Window Veto, as the current veto algorithm is also optimized for these backgrounds.

Similarly, it would possibly be prudent to see a machine learning model's reaction to changing the reconstruction thresholds: for instance, changing the threshold for a valid muon track stub to 2/4 layers hit instead of 3/4, to allow for a higher efficiency. This could help gain more cosmic-ray induced CEs, but may also backfire, so it would be an interesting test to do in the future.

Currently the performance of the ML Veto, similar to the CRV Time Window Veto, is very dependent on the light yield of the sample. This should be improved upon due to aging concerns in the CRV. Looking into methods making the machine learning model more resilient towards changes in the light yield is a point for further study.

XII. Acknowledgements

This work was done under the supervision and guidance of Dr. Yuri Oksuzian and Prof. E. Craig Dukes. Feedback and guidance was also provided by members of the Mu2e Collaboration over the course of the study.

XIII. References

[1] Fermi National Accelerator Laboratory, "Mu2e for physicists," https://mu2e.fnal.gov/public/index.shtml, 2021, [Online; accessed 2021].

[2] Y. Oksuzian, "A cosmic ray veto detector for the mu2e experiment at fermilab," https://indico.cern.ch/event/361123/contributions/856188/attachments/1135881/1625309/DPF2015_CRV_Overview.pdf, 2017, [Online; accessed 2022].

[3] E. C. Dukes, "Cosmic rays are a pain: The mu2e cosmic ray veto," http://muse.lnf.infn.it/wp-content/uploads/2017/08/mu2e_crv_seminar_2017_08_fnal.compressed.pdf, 2017, [Online; accessed 2021].

[4] CERN, "Geant4: A simulation toolkit," https://geant4.web.cern.ch/, 2022, [Online; accessed 2022].

[5] Y. Oksuzian, "Report on the 2020 LDRD expedition project: "improvements to cosmic muon identification at mu2e," 2020, [Online; accessed 2021].

[6] Keras, "Keras: the python deep learning api," https://keras.io/, 2021, [Online; accessed 2021].

[7] C. H. Martin, T. Peng, and M. W. Mahoney, "Predicting trends in the quality of state-of-the-art neural networks without access to training or testing data," *CoRR*, vol. abs/2002.06716, 2020. [Online]. Available: https://arxiv.org/abs/2002.06716

[8] C. Martin, "weightwatcher 0.5.5," https://pypi.org/project/weightwatcher/, 2021, [Online; accessed 2021].