10-2022

# Unsupervised Machine Learning Approaches to Nuclear Particle Type Classification

Daniel Ruiz
*TRAC*, daniel.ruiz@nps.edu

Nicholas Liebers
*USMA*, nicholas.liebers@westpoint.edu

Jacob Huckelberry
*USMA*, jacob.huckelberry@westpoint.edu

David Fobar
*USMA*, david.fobar@westpoint.edu

Peter Chapman
*USMA*, peter.chapman@westpoint.edu

# Unsupervised Machine Learning Approaches to Nuclear Particle Type Classification

Nicholas Liebers, Jacob Huckelberry, Daniel Ruiz, David Fobar, and Peter Chapman

2022 IEEE Long Island Systems, Applications and Technology Conference (LISAT) | 978-1-6654-6596-0/22/$31.00 ©2022 IEEE | DOI: 10.1109/LISAT50122.2022.9924043

***Abstract*—Historically, nuclear science and radiation detection fields of research used Pulse Shape Discrimination (PSD) to label gamma-ray and neutron interactions. However, PSD's effectiveness relies greatly on the existence of distinguishable differences in an interaction's measured pulse shape. In the fields of machine learning and data analytics, clustering algorithms provide ways to group samples with similar features without the need for labels. Clustering gamma-ray and neutron interactions may mitigate PSD's pitfalls, since clustering methods view the total waveform rather than just the area under the tail and the total area under the pulse. However, traditional clustering methods, such as the k-means clustering algorithm, suffer from poor performance on high dimensional data. This study explores unsupervised machine learning methods using Deep Neural Networks (DNN) to cluster gamma-ray and neutron interaction measurements collected with an organic scintillation detector, in order to perform binary labeling of gamma-rays and neutrons. Using various network architectures, this research demonstrates the effectiveness of using autoencoder-based neural networks to cluster gamma-ray and neutron interactions when compared to shallow clustering algorithms. The results reveal the effectiveness of autoencoders on high energy gamma-ray and neutron pulses with an energy deposit greater than 0.80 MeVee whilst greatly outperforming k-means comparatively in all cases.**

***Index Terms*—Particle Discrimination, Neural Networks, Unsupervised Machine Learning**

## I. Introduction

Previously, particle identification in scintillation detectors has been dominated by Pulse Shape Discrimination (PSD). However, PSD's performance is highly dependant on the existence of meaningful differences between particles' measured pulses [1]. Particularly with low energy pulses, it can be difficult to discern the difference between particle pulses using PSD, especially regarding the area under the tail and the total area under the pulse. [1]. In the field of data science, categorizing similar data points using a metric, such as the Euclidean distance, is an effective way to discover similarities in data. Unsupervised learning methods do not require data labels, and categorizing particle pulses in an unsupervised manner may help mitigate the uncertainty associated with PSD. Unfortunately, traditional clustering methods, such as k-means, do not scale well with large data sets [2]. Additionally, these classical clustering methods tend to perform poorly on high-dimensional data [3]. High dimensionality refers to the presence of multiple attributes per sample in a dataset, thus rendering each datapoint impossible to visualize in a two or three dimensional plot. For visualization and extraction of key features, such data requires some form of dimensionality reduction before effective clustering is possible. In the case of the waveforms being processed, each waveform is comprised of 100 analog time-series attributes. With this observation in mind, methods to reduce dimensionality become crucial and highly researched. Modern dimensionality reduction methods, including principal component analysis (PCA), local linear embedding (LLD), and t-SNE, reduce the dimensions of data but drop in effectiveness as complexity increases [4], [5], [6]. Deep neural networks (DNNs) can also perform non-linear dimensionality reduction, which will be the focus of his paper.

This work was performed under the auspices of the Defense Threat Reduction Agency's (DTRA) Nuclear Science and Engineering Research Center (NSERC) and the Cyber Research Center (CRC) at West Point, NY.

N. Liebers (nicholas.liebers@westpoint.edu), J. Huckelberry (jacob.huckelberry@westpoint.edu), and D. Ruiz (daniel.ruiz@westpoint.edu) are members of the Department of Electrical Engineering and Computer Science, United States Military Academy, West Point, NY. D. Fobar (david.fobar@westpoint.edu), and P. Chapman (peter.chapman@westpoint.edu) are members of the Department of Physics and Nuclear Engineering, United States Military Academy, West Point, NY.

## II. Background on ML Particle Discrimination

The research question explored through this study follows from prior research that investigates the effectiveness of supervised machine learning for particle discrimination [1]. In that research, the authors deploy a dense neural network to classify gamma-rays and neutrons. In the supervised approach to this problem, the dense neural network is provided with both the features, digitized waveforms from an EJ309 organic liquid scintillator exposed to a $^{252}$Cf source, and the classifications of each gamma-ray and neutron interaction. The classification of each sample from the set is determined through a process the authors defined as time-correlated pulse-height and time-correlated pulse shape parameter screening (TCPH-TCPSP screening). In testing their neural network, they use five different training sets. One set has pulse features described by both the pulse shape parameter and time of flight, another set only has pulse features described by the pulse shape parameter, and the final three sets have pulses described by the same features as the first set mentioned except they exclude energy deposits below 0.20 MeVee, 0.40 MeVee, and 0.80 MeVee, respectively.

The results from the first set of testing result in 99.7% classification accuracy while their results from the second set produce 99.6% accuracy [1]. Although these are strong results, they are entirely dependent on the quality of labels provided for the training and test data sets. As mentioned before, the process of labeling these gamma-ray neutron data sets can be a challenge, as the primary methods for labeling these sets rely heavily on the pulses having distinguishable shapes. The efforts in this paper aim to remedy this issue

by offering a unsupervised machine learning method for labeling these particle data sets without the pitfalls of PSD. By utilizing unsupervised learning, we are able to reduce the dimensionality of the observed features of the gamma-ray and neutron pulses, thus enabling individual particle separation in a two-dimensional visualization.

## III. Merits of Unsupervised Learning

### A. The Why Behind Unsupervised Methods

Previous research concerning the use of unsupervised machine learning methods for clustering shows promise for the application of such a model to our problem. In their article "Analysis of Classification by Supervised and Unsupervised Learning", Sapkal et. al. explain the methodology behind unsupervised learning methods. The authors state that models created from unsupervised techniques, in comparison to supervised models, are not as effective in pure classification, although there are use cases that make them a vital resource in data science. These use cases include remedying the costly nature of labeling large sets of samples and extracting features that are useful for categorization [7]. Labeling samples in data sets usually requires manual labor to accomplish, and depending on a data set's size, it can make building labeled data sets difficult. As mentioned previously, the current processes used to calculate labels for neutron and gamma-rays provide a limited degree of certainty, especially for lower energy events.

### B. Strengths of Unsupervised Methods

In their article "A Survey of Clustering With Deep Learning: From the Perspective of Network Architecture", Xie et. al. investigate various deep unsupervised architectures and their strengths as compared to their shallow counterparts [8]. Before the widespread use of neural networks, clustering was largely done with shallow methods such as k-means [2]. As a result, k-means is regularly used in contemporary publications as a benchmark for clustering performance when compared to state-of-the-art methods [9], [10], [11], [12]. The k-means algorithm categorizes data into one of $k$ clusters. By calculating the euclidean distance between a data point and a cluster center, the algorithm forms clusters such that the average euclidean distance between a cluster and its points is minimized for $k$ clusters. As mentioned earlier, these methods are unable to process high-dimensional data due to weak similarity measures between two points in a high dimensional space [8]. Neural networks remedy this issue because of their ability to produce unique feature representations and perform dimensionality reduction. The authors cite autoencoders as "one of the most significant algorithms in unsupervised representation learning" [8]. This distinction comes from the architecture's ability to distill only the most important features of the data through dimensionality reduction [8]. An important architecture in the development of deep clustering and a stepping stone for this study is the Deep Embedded Clustering (DEC) architecture [9]. This model is a neural network that is specifically designed for clustering data. This architecture set the stage for many more models like Variational Deep Embedding (VaDE) and others [10], [11], [13]. With the strengths of deep unsupervised clustering and the recent advancements due to models such as the DEC, an autoencoder based architecture provides a viable approach to clustering our gamma-ray and neutron pulse data set due to the dimensionality of each interaction sample.

## IV. Source of data set

The data set in this study consists of a collection of over 5 million gamma-ray and neutron pulses. These pulses are collected from the fission events of a $^{252}$Cf source using lanthanum-bromide and EJ-309 detectors. For each sample, the pulse shape parameter (PSP) is calculated and included in the metadata of the samples. Using this same data set, supervised machine learning with an artificial neural network resulted in over 99% accuracy above 0.20 MeVee [1]. Readers will note the data labels from this data set are only used for verifying the efficacy of unsupervised clustering. These labels are not presented to the model during training. This data set was used in the published work described in our Background on ML Particle Discrimination section.

## V. Architecture - Network Design

In order to create a viable clustering environment for the k-means algorithm, we used a neural network to reduce the data's dimensionality. To assist readers in fully understanding this process, we briefly introduce deep neural networks, recurrent neural networks, our model's architecture in this section.

### A. An Introduction to Deep Neural Networks

A deep neural network (DNN) is a structured algorithm that applies a series of linear and non-linear transformations to some input data. In doing so, multiple layered representations of the input data are created in an effort to approximate a complex underlying function. As the network trains, a "loss function" evaluates a network's predictions with respect to its targets [14]. The output of this loss function is then minimized via an algorithm known as backpropagation, which adjusts the network's internal parameters by computing the partial derivative of the loss function with respect to each parameter [14]. However, DNNs that rely solely on fully connected layers of neurons do not take advantage of the sequential information found within some data sets, such as the order of values present in our radiation detector waveforms [14].

### B. Neural Network Architectures for Sequential Data

Recurrent neural networks (RNNs), a popular network architecture when training on sequential data, are similar to DNNs since both network types consist of layers of transformations that aim to minimize a loss function; however, RNNs are different in how they process the input data. As seen in Fig. 1, the current state of a neuron in an RNN is a function of the input values into the neuron, and its own prior output. This reinjection of historical output allows RNNs to better capture meaningful information from sequential data. While RNNs offer increased capabilities for sequential data compared to DNNs, RNNs suffer from exploding and vanishing gradients:

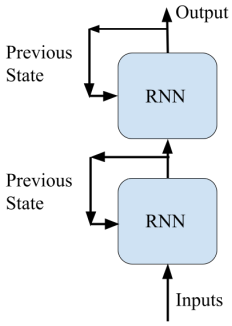



Figure 1. A recurrent neural network computes its next state as a function of an input and the node's previous state. Two recurrent neural network nodes are computing outputs based on an input and their respective previous states.

an effect where the network's parameters become unstable, causing backpropagation to compute nonsensical updates that effectively halt the training process [15]. To combat exploding and vanishing gradients, Long Short-term Memory (LSTM) neurons were introduced [15]. LSTMs incorporated additional parameterized mechanisms for preserving information extracted from earlier parts of a sequence, which made them a popular choice for sequential and time-series problems. However, this increased performance tends to come at a cost of increased training time, due to the increased number of trainable parameters in the model.

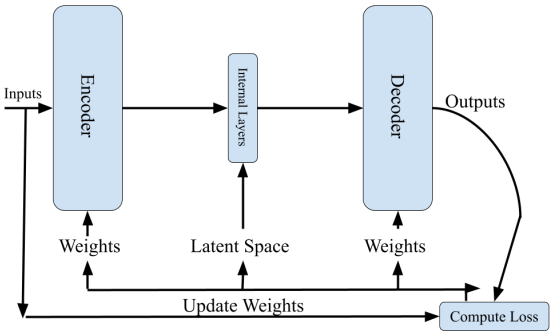



Figure 2. A simple autoencoder that contains an encoder, a latent space, and a decoder. This loss function compares the similarity between the inputs and the generated outputs.

### C. Dimensionality Reduction with Autoencoders

As mentioned previously, traditional clustering methods, such as k-means, struggle to cluster data with high dimensionality. As referenced in Fig. 2, autoencoders (AE) offer a non-linear approach to reducing the dimensions of input data while still preserving important features of the data. AEs are a relatively simple network architectures consisting of an encoder and a decoder separated by a bottleneck layer consisting of fewer neurons than its neighbors [16]. The goal of an AE is to receive an input and reconstruct it after computing a low-dimensional representation via the bottleneck. In the context of a simple AE, its loss–the function the network is attempting to minimize–computes the difference between the AE's reconstructed outputs and its inputs [12]. A key characteristic of an AE is the purposeful dimension reduction between the encoding and decoding layers [16]. Since there is a reduction in dimensions in the middle of the network, the AE must learn the most important features of the input data to be used for reconstruction [9]. The inputs mapped to the latent space of the bottleneck are referred to as the "latent representation" of the inputs. Extracting the latent representation allows us to use clustering algorithms, like k-means, on the lower dimensional data with higher degrees of effectiveness [8].

### D. A Variational Recurrent Auto-encoder Architecture for Clustering Sequential Data

The architecture used in this paper is Variational Recurrent Auto-encoder (VRAE)–an LSTM-based autoencoder [17]. Fig. 3 diagrams the model's structure and pertinent layers.

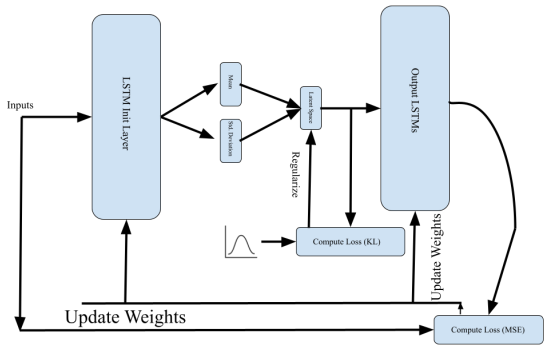



Figure 3. An illustration of the network used to create a latent space suitable for clustering energy pulses. This model contains two loss functions: Mean Squared Error and Kullback-Leiber divergence loss [17].

*1) VRAE Description:* Like the standard implementation of an AE, the VRAE contains two main parts: an encoder and a decoder. As seen in Fig. 3, in the encoder, input data is forward propagated through the LSTM layers. Since this model is a variational AE, its encoding layer does not encode inputs into individual datapoints in the latent space. Instead, the encoder maps inputs to a distribution described by a mean $\mu$ and standard deviation $\sigma$. To decode a sample distribution from the latent space, a random sample from the selected latent distribution is chosen and passed into the decoder, and the decoder proceeds to reconstruct the input based on the sample [17]. The main benefit of this approach compared to a regular AE is that a VRAE helps to prevent overfitting to the input data [16].

*2) VRAE Loss Functions:* VRAEs are trained using two loss functions: the mean squared error (MSE) loss function between encoded inputs and decoded outputs and Kullback-Leiber divergence (KL-divergence) loss function of the learned

latent distribution and a normal distribution. The former ensures the VRAE is accurately reconstructing the inputs, while the latter creates a consistent latent space that also mitigates overfitting. KL-divergence is a way to measure the similarity between two distributions, and minimizing the difference between the latent distribution and a normal distribution regularizes the latent space during training [18], [16]. This model is fully implemented in Pytorch [19], [17].

To cluster the data, we extract the latent representation of the data from the latent space. We then use the k-means clustering algorithm to categorize similar events.

### E. Hyperparameters Used During Training

The VRAE was trained on the data set for 30 epochs with a batch size of 512. Training for longer than 30 epochs created an inconsistent latent space during testing. The latent space dimensions are set to 20, the learning rate is 0.0005, and the dropout rate is 20%

## VI. Measures of Success

In order to determine overall clustering performance, clustering label accuracy is used. This metric is popular among other studies regarding clustering [9], [10], [11], [13], [12].

### A. Accuracy

Clustering Accuracy (ACC) is defined as follows:

$$ACC = \max_{m} \frac{\sum_{i=1}^{n} 1\{l_i = m(c_i)\}}{n}, \tag{1}$$

where $l_i$ is the ground-truth labels, $c_i$ is the cluster assignment, and $m$ iterates over all one-to-one mappings between each cluster and labels [9]. To evaluate the clustering performance of the VRAE, we compare the VRAE's clustering ACC to k-means clustering ACC. In this case, k-means serves as a baseline for VRAE's performance.

## VII. Data Preparation

To prevent an inherent bias towards neutrons or gamma-rays, the dataset was balanced to contain $30,000$ gamma-ray events and $30,000$ neutron events. While the data set contains over 5 million pulses, the training time with LSTMs is costly when using large amounts of data. As a result, the sample size was reduced to promote faster training on available hardware. The data set was randomly shuffled, and $30,000$ pulses for each category were saved for training and ACC evaluations.

Additionally, to investigate how training on different energy deposition regimes impacts clustering ACC, subsets of the main data set containing different minimum energy deposit interactions were trained and tested. Nine different balanced data sets of $60,000$ total interactions were created using the main data set. Each data set version introduces a new threshold that filters out certain particle interactions based on their respective energy deposit values.

Fig. 4 plots the percentage of the total data set available for randomly sampled selection for training and testing as lower-energy events are filtered out. The available samples in the main data set diminishes quickly as lower energy events are cut from training. This constraint prevents the investigation of clustering performance on higher energy samples since there are relatively few in the main data set from which to sample.

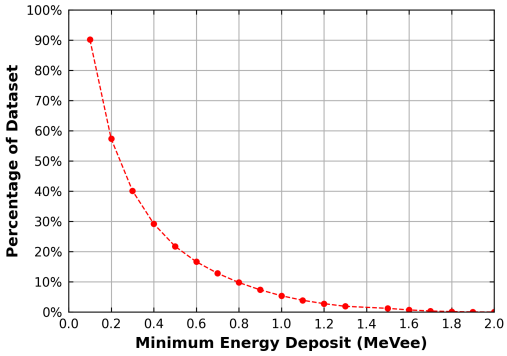



Figure 4. Percentage of the main data set available for sampling for training as a function of the minimum energy deposit cut.

### A. Cases

In order to investigate our VRAE's cluster effectiveness compared to k-means, 3 cases are required: Case 1 trains the VRAE on the 9 balanced data sets and tests only within the same energy regime the model was trained on. Case 2 trains the VRAE on a single balanced data set of events greater than 0.80 MeVee energy deposited and, the results are tested against all test data sets from case 1. This allows for the degradation in model performance to be understood in energy regimes the model was not trained on. Case 3 uses the model from Case 2 and validates its results on an unbalanced data set. Case 3 explores the VRAE's ability to not only generalize to energy deposit values that it has never seen, but it also evaluates the model's ability to generalize to an unbalanced data set. In the real world, a data set of gamma-rays and neutron interactions would be unbalanced. Therefore, it is of interest of how well our model performs on the unbalanced data cut.

## VIII. Results

### A. Case 1

Table I details gamma-ray and neutron labeling accuracy for the VRAE and k-means as the minimum energy deposit of a particle interactions increase. This table and figure demonstrate our model's increased performance over k-means as lower-energy pulses are removed from the training data. This performance gain over k-means continues to increase as the low energy cut-off is increased. For each iteration, the VRAE is trained and tested on the same minimum energy deposit value measured in MeVee.

### B. Case 2

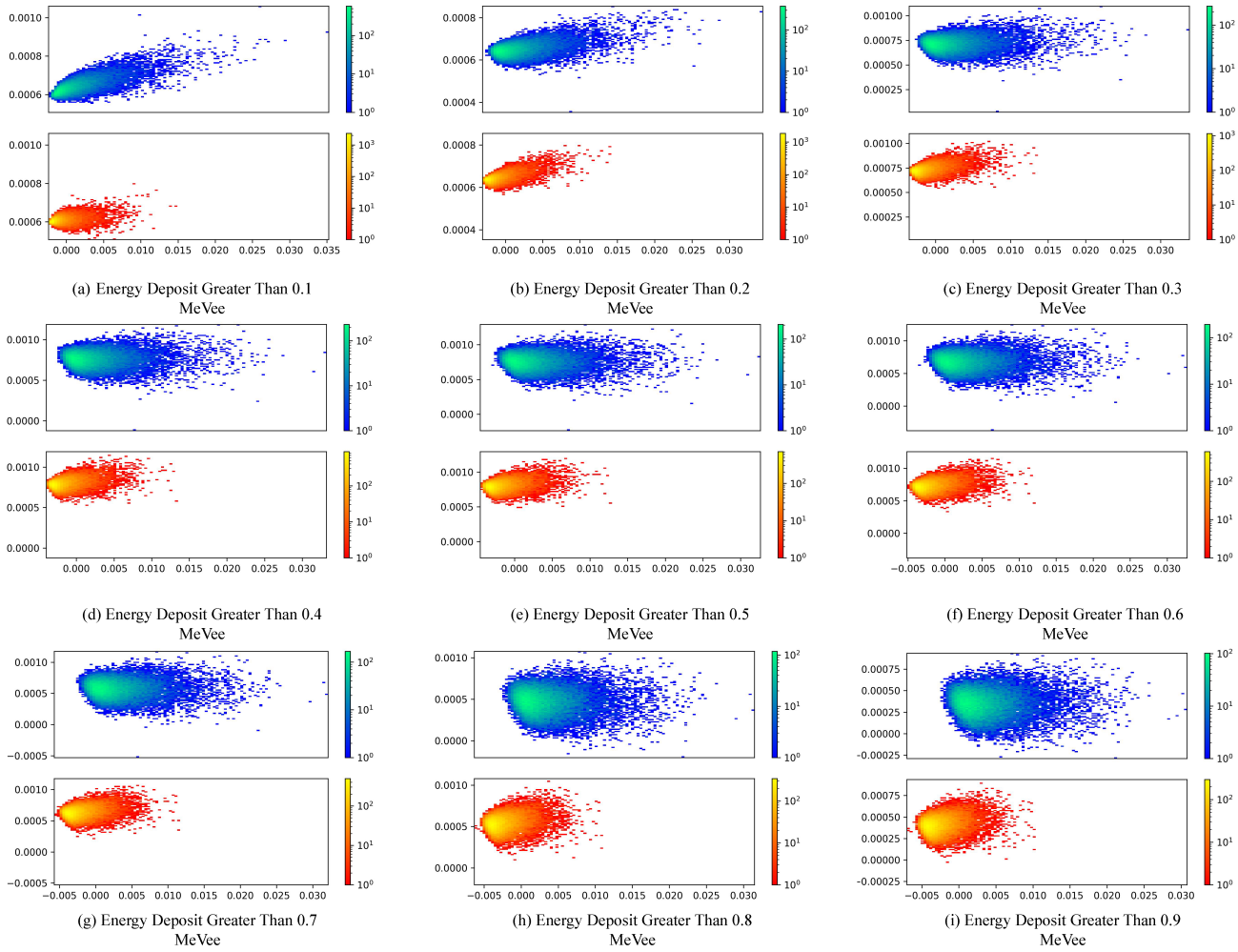
Table II details gamma-ray and neutron labeling accuracy when the VRAE is trained solely on particle interactions with a energy deposit value greater than 0.80 MeVee. In

Table I
Comparison of clustering accuracy as minimum energy deposit sampled is increased in Case 1

| Energy Deposited [MeVee] | 0.1 | 0.2 | 0.3 | 0.4 | 0.5 | 0.6 | 0.7 | 0.8 | 0.9 | 1.0 | 1.1 | 1.2 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| K-means | 50.04% | 50.16% | 50.12% | 50.14% | 50.32% | 50.45% | 50.21% | 50.50% | 51.67% | 52.68% | 53.84% | 55.57% |
| VRAE | 50.12% | 63.09% | 64.54% | 66.83% | 71.93% | 75.74% | 79.13% | 82.94% | 87.46% | 90.78% | 92.81% | 94.85% |

Table II
Comparison of clustering accuracy as minimum energy deposit sampled is increased in Case 2

| Energy Deposited [MeVee] | 0.1 | 0.2 | 0.3 | 0.4 | 0.5 | 0.6 | 0.7 | 0.8 | 0.9 | 1.0 | 1.1 | 1.2 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| K-means | 50.12% | 50.16% | 50.12% | 50.14% | 50.32% | 50.45% | 50.21% | 50.50% | 51.67% | 52.68% | 53.84% | 55.57% |
| VRAE | 60.05% | 62.42% | 64.77% | 67.47% | 70.91% | 73.89% | 77.35% | 82.00% | 87.09% | 90.81% | 93.65% | 95.19% |



(a) Energy Deposit Greater Than 0.1 MeVee
(b) Energy Deposit Greater Than 0.2 MeVee
(c) Energy Deposit Greater Than 0.3 MeVee
(d) Energy Deposit Greater Than 0.4 MeVee
(e) Energy Deposit Greater Than 0.5 MeVee
(f) Energy Deposit Greater Than 0.6 MeVee
(g) Energy Deposit Greater Than 0.7 MeVee
(h) Energy Deposit Greater Than 0.8 MeVee
(i) Energy Deposit Greater Than 0.9 MeVee

Figure 5. Latent representations mapped to 2-dimensions using PCA of gamma-rays and neutrons from experiment 2's model. Axis values are arbitrary due to the dimensionality of the latent space. Colorbars scale logarithmically based on frequency of particle interactions in 2-d space.

order to visually inspect the latent representations, we can use a shallow dimension-reducing function, such as PCA, to map the data into 2 dimensions so it can be plotted. In Fig. 5, the various latent representations are projected to 2 dimensions. As the minimum energy deposit increases, notice the clusters, or the high density areas of the clusters, diverge from one another. Increased cluster separation creates a better environment for k-means to distinguish between each cluster. This model's weights create consistently round clusters. Case 2's model performs well compared to Case 1's data even though Case 2's model was not trained on any interactions with a energy deposit less than 0.80 MeVee. Fig. 6 plots this model's performance compared to case 1's model as a function of minimum energy deposit measured in MeVee.

### C. Case 3

Fig. 7 demonstrates Case 2's model's ability to generalize to an unbalanced data set. While the difference in accuracy

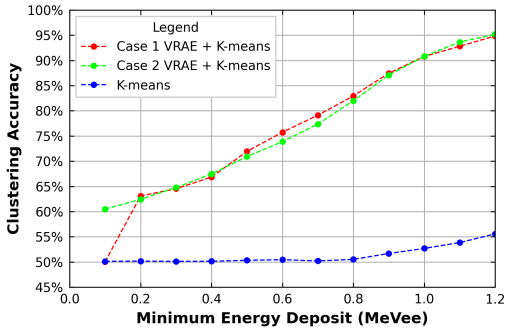



Figure 6. Clustering Accuracy as a function of minimum energy deposit for Cases 1 and 2.

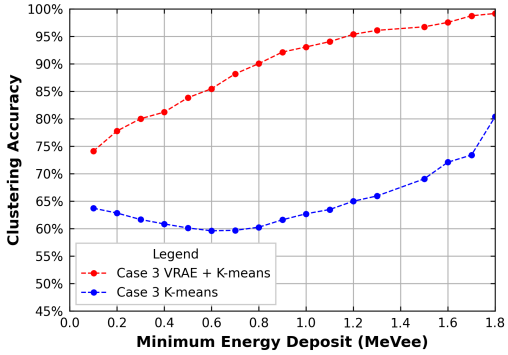



Figure 7. Clustering Accuracy as a function of minimum energy deposit for Case 3.

between the VRAE and k-means is less in Case 3, the VRAE model still greatly outperforms k-means. Notice that k-means establishes a baseline at about 62% on the unbalanced data set. This specific balance demonstrates k-means is effectively guessing randomly. About 62% of the unbalanced data set was previously labeled as gamma-rays with the remainder being neutrons that are produced during $^{252}$Cf spontaneous fission.

## IX. Conclusion/Future Work

The results produced by our VRAE model show promise for clustering and labeling particle data using unsupervised machine learning techniques. However, for energy deposits below 100 keVee, while our VRAE still outperforms k-means, the clustering accuracy is not good enough for any meaningful labeling, suggesting more research should be conducted. Potential solutions include applying a 1D convolutional variational autoencoder with Kullback-Leiber divergence loss to creating a latent space for k-means clustering [18]. LSTM layers are effective on this data set as a result of the temporal nature of the gamma-ray and neutron pulses, but 1D convolutional layers may provide a better alternative. Traditional convolutional layers take advantage of the spatial relationships found in image data, and a 1D convolutional layer may work to a similar effect. The ability of these layers to take advantage of both the spatial and temporal aspects of data make them a strong candidate for creating a novel model for clustering with this data set. Mainly, 1D convolutional layers would train faster, thus mitigating the major pitfall associated with using LSTMs.

## References

[1] D. Fobar, L. Phillips, A. Wilhelm, and P. Chapman, "Considerations for training an artificial neural network for particle type identification," *IEEE Transactions on Nuclear Science*, vol. 68, no. 9, pp. 2350–2357, 9 2021.

[2] J. A. Hartigan and M. A. Wong, "Algorithm AS 136: A K-Means Clustering Algorithm," *Applied Statistics*, vol. 28, no. 1, p. 100, 1979.

[3] L. Steinbach Michael }and Ertöz and K. Vipin, "The Challenges of Clustering High Dimensional Data," in *New Directions in Statistical Physics: Econophysics, Bioinformatics, and Pattern Recognition*, L. T. Wille, Ed. Berlin, Heidelberg: Springer Berlin Heidelberg, 2004, pp. 273–309. [Online]. Available: https://doi.org/10.1007/978-3-662-08968-2_16

[4] S. Wold, K. Esbensen, and P. Geladi, "Principal component analysis," *Chemometrics and Intelligent Laboratory Systems*, vol. 2, no. 1-3, pp. 37–52, 1987.

[5] S. T. Roweis and L. K. Saul, "Nonlinear dimensionality reduction by locally linear embedding," *Science (New York, N.Y.)*, vol. 290, no. 5500, pp. 2323–2326, 12 2000. [Online]. Available: https://pubmed.ncbi.nlm.nih.gov/11125150/

[6] L. van der Maaten and G. Hinton, "Visualizing Data using t-SNE," *Journal of Machine Learning Research*, vol. 9, no. 86, pp. 2579–2605, 2008.

[7] S. Sapkal, S. Kakarwal, and P. Revnakar, "Analysis of Classification by Supervised and Unsupervised Learning," *International Conference on Computational Intelligence and Multimedia Applications 2007*, pp. 280–284, 2007.

[8] E. Min, X. Guo, Q. Liu, G. Zhang, J. Cui, and J. Long, "A Survey of Clustering With Deep Learning: From the Perspective of Network Architecture," *IEEE Access*, vol. 6, 2018. [Online]. Available: http://www.ieee.org/publications_standards/publications/rights/index.html

[9] J. Xie, R. Girshick, and A. Farhadi, "Unsupervised Deep Embedding for Clustering Analysis," *33rd International Conference on Machine Learning, ICML 2016*, vol. 1, pp. 740–749, 11 2015. [Online]. Available: https://arxiv.org/abs/1511.06335v2

[10] Z. Jiang, Y. Zheng, H. Tan, B. Tang, and H. Zhou, "Variational Deep Embedding: An Unsupervised and Generative Approach to Clustering," 11 2016. [Online]. Available: http://arxiv.org/abs/1611.05148

[11] B. Yang, X. Fu, N. D. Sidiropoulos, and M. Hong, "Towards K-means-friendly Spaces: Simultaneous Deep Learning and Clustering," 10 2016. [Online]. Available: http://arxiv.org/abs/1610.04794

[12] R. McConville, R. Santos-Rodríguez, R. J. Piechocki, and I. Craddock, "N2D: (Not Too) Deep Clustering via Clustering the Local Manifold of an Autoencoded Embedding," in *2020 25th International Conference on Pattern Recognition (ICPR)*, 2021, pp. 5145–5152.

[13] M. M. Fard, T. Thonet, and E. Gaussier, "Deep $k$-Means: Jointly clustering with $k$-Means and learning representations," 6 2018. [Online]. Available: http://arxiv.org/abs/1806.10069

[14] F. Chollet, *Deep Learning with Python, Second Edition*, 2nd ed. Shelter Island, NY: Manning Publications Co, 10 2021.

[15] S. Hochreiter and J. Schmidhuber, "Long Short-Term Memory," *Neural Computation*, vol. 9, no. 8, pp. 1735–1780, 11 1997.

[16] F. Chollet, "Building Autoencoders in Keras," 2016. [Online]. Available: https://blog.keras.io/building-autoencoders-in-keras.html

[17] T. Lodaya, "Timeseries clustering," 12 2019. [Online]. Available: https://github.com/tejaslodaya/timeseries-clustering-vae

[18] S. Kullback and R. A. Leibler, "On Information and Sufficiency," *The Annals of Mathematical Statistics*, vol. 22, no. 1, pp. 79–86, 1951. [Online]. Available: http://www.jstor.org/stable/2236703

[19] "PyTorch." [Online]. Available: https://pytorch.org/