

Using machine learning to diagnose relativistic electron distributions in the Van Allen radiation belts

S. Killey,¹★ I. J. Rae,¹ S. Chakraborty¹,^{ORCID} A. W. Smith,¹ S. N. Bentley,¹ M. R. Bakrania,² R. Wainwright,¹ C. E. J. Watt¹ and J. K. Sandhu¹

¹Department of Mathematics, Physics and Electrical Engineering, Northumbria University, Newcastle upon Tyne NE1 8ST, UK

²Mullard Space Science Laboratory, UCL, Dorking RH5 6NT, UK

Accepted 2023 August 4. Received 2023 July 20; in original form 2023 January 31

ABSTRACT

The behaviour of relativistic electrons in the radiation belt is difficult to diagnose as their dynamics are controlled by simultaneous physical processes, some of which may be still unknown. Signatures of these physical processes are difficult to identify in large amounts of data; therefore, a machine learning approach is developed to classify energetic electron distributions which have been driven by different mechanisms. A series of unsupervised machine learning tools have been applied to 7 yrs of Van Allen Probe *Relativistic Electron-Proton Telescope* data to identify six different typical types of plasma conditions, each with a distinctly shaped energy-dependent pitch angle distribution (PAD). The PADs at lower energies have shapes as expected from previous studies – either butterfly, pancake, or flattop, providing evidence that machine learning has been able to reliably classify the relativistic electrons in the radiation belts. Further applications of this technique could be applied to other space plasma regions, and data sets from inner heliospheric missions such as Parker Solar Probe and Solar Orbiter, to planetary magnetospheres and the JUICE mission. Understanding PADs across the heliosphere enables researchers to determine the physical mechanisms that drive pitch angle evolution and investigate their spatial and temporal dependence and physical properties.

Key words: radiation belts – electrons – machine learning – unsupervised; pitch angle distributions.

1 INTRODUCTION

The behaviour of energetic particles in the Van Allen radiation belts is difficult to diagnose due to their complicated dynamics. At times, radiation belt dynamics can be dominated by a multitude of loss, transport, and acceleration processes (Reeves et al. 2003; Baker et al. 2018; Li & Hudson 2019; Chu et al. 2021; Chakraborty et al. 2022) including magnetospheric shadowing (e.g. Herrera, Maget & Sicard-Piet 2016; Staples et al. 2020), and wave-particle interactions (e.g. Artemyev et al. 2016; Ripoll et al. 2020) involving pitch angle scattering (e.g. Summers & Thorne 2003; Chaston et al. 2018) and atmospheric precipitation (e.g. Rodger et al. 2007; Rae et al. 2018). These different physical processes drive relativistic electron behaviour at different energies and pitch angles, leading to differently shaped energy-dependent pitch angle distributions (PADs). Hence, PADs are essential in understanding the state of plasma regions (Bakrania et al. 2020a) that can have a pancake (peak at 90°), butterfly (electron flux minima at 90°), or flattop (flux plateau over a range of pitch angles centred on 90°) shape (Horne et al. 2003; Gannon, Li & Heynderickx 2007; Souza et al. 2016; Zhao et al. 2018; Chakraborty et al. 2022).

The Van Allen Probe mission (Mauk et al. 2013), provides over 7 years of extensive, high-quality observations of radiation belt particles (Baker et al. 2018) to analyse radiation belt physics. For

relativistic electrons (>1 MeV) alone, there are almost 20 million observations measured using the *Relativistic Electron-Proton Telescope* (REPT) instrument (Baker et al. 2013). Identifying physical processes that drive the behaviour of relativistic electrons in such big data is a difficult and lengthy process. Therefore, to understand relativistic electron behaviour, we must first understand the response of MeV electrons to changes to the magnetosphere by investigating their distributions. However, with 20 million observations, it is impossible to reliably identify PADs of similar shape by eye, which likely introduces significant bias to the results. Traditional PAD studies typically pre-define the shape of the distributions a priori (e.g. Liu et al. 2020; Chakraborty et al. 2022; Ozeke et al. 2022) at given energies. In this work, we adopt a different approach and consider PADs at all energies and without the assumptions of specific PAD shapes.

Machine learning classification methods have been found to be incredibly useful in space physics, including for investigating the different plasma regions in the Earth's magnetosphere (e.g. Breuillard et al. 2020; Innocenti et al. 2021) and other planetary magnetospheres (e.g. Cheng, Achilleos & Smith 2022; Yeakel et al. 2022), identifying solar wind types (e.g. Camporeale, Caré & Borovsky 2017; Amaya et al. 2020; Bloch et al. 2020), and solar wind characteristics (e.g. Bakrania et al. 2020b) and even space weather forecasting (e.g. Maimaiti et al. 2019; Smith et al. 2020). Machine learning methods have also increasingly been used to model the highly dynamic radiation belts (e.g. Bortnik et al. 2016; Chu et al. 2021; Wing et al. 2022).

* E-mail: s.killey@northumbria.ac.uk

Table 1. The random, shuffled 20:20:60 splitting of the training, validation, and testing sets, respectively, between 2012 September and 2019 July.

Set	Split (per cent)	Number of observations	Date range
Training	20	3 995 838	2012 September–2019 July
Validation	20	3 995 838	2012 September–2019 July
Testing	60	11 987 516	2012 September–2019 July

Recent classification works using machine learning by Bakrania et al. (2020a) have been performed on plasma sheet electron distributions that yield robust identifications of clusters of PADs. These clusters can then be analysed to identify mechanisms that result in different particle populations in the magnetotail (Bakrania et al. 2020a). Using a different machine learning technique, Souza et al. (2016) determined similarly shaped PADs of 1.8 MeV relativistic electrons using a month’s worth of REPT data, evidencing that machine learning can identify underlying relationships and classify large particle data sets. By adapting the method of Bakrania et al. (2020a), we present a new unsupervised learning technique (Section 3) applied to relativistic electron data to cluster similar energy-dependent PADs together (Section 6) for the duration of the Van Allen Probe mission. In this paper, we describe this new technique and its suitability to plasma PADs, using Van Allen Probe REPT data as an exemplar.

2 INSTRUMENTATION

The NASA Van Allen Probes follow an elliptical orbit of ~ 600 km \times $5.8 R_E$ with a $\sim 10^\circ$ inclination, meaning that the satellites collect observations from both the inner and outer radiation belts with an orbital cadence of 9 h (Mauk et al. 2013). We analyse relativistic radiation belt electron fluxes from 1 to 20 MeV measured by the Van Allen Probe REPT instrument (Baker et al. 2013) at a temporal resolution of the order of tens of seconds for the entire Van Allen Probe mission lifetime between 2012 September and 2019 July. In this paper, we apply the technique to Van Allen Probe B only, although this technique is valid for any spacecraft measurement.

REPT measured flux f as a function of 17 different pitch angles PA and 12 energies E , meaning that at each observation time t there are a total of 204 individual energy-pitch angle bins or ‘dimensions’. The flux, which we refer to as $f(PA, E, t)$, from REPT was then normalized with respect to the maximum flux of each observation, in order to focus on the shape of distribution rather than predominantly on the particle flux or density. As we are interested in understanding the shape of the PADs rather than the magnitudes, this step has minimal impacts. Although we note here that this technique can be adapted without issue if the magnitude of flux was of specific interest. The normalized three-dimensional (3D) array of size $17 \times 12 \times N_t$, where N_t is the number of observations, was flattened to a 2D array of size $204 \times N_t$ for easier processing in the early stages of machine learning. We consider PADs across all energies rather than a subset as an exemplar; the machine learning techniques of this study can be easily adapted to consider a smaller range of energies and therefore a lower number of dimensions.

Due to the multidimensionality of REPT data, we employ a 2-step process to reduce the number of dimensions to a more manageable number while considering the linear and non-linear trends within the data and retaining the most important information. Dimensionality reduction was first achieved by applying an autoencoder (AE; Section 4.1) to compress the REPT data from 204 dimensions to 102 dimensions, before then applying principal

component analysis (PCA; Section 4.2) to further compress the data into only three dimensions. The 3D REPT data was then classified by first using a mean shift algorithm (Section 5.1) to determine the number of clusters, before a k-means algorithm (Section 5.2) was applied to sort the relativistic electron data into this number of groups.

3 MACHINE LEARNING TECHNIQUES

3.1 Train – validation – test split for machine learning

Unsupervised machine learning is a valuable tool used to determine important characteristics in data without prior teaching (Kyan et al. 2014) by imposing as few a priori decisions as possible, e.g. the number of clusters. Standard practice for machine learning is to split the data into three: a training, validation, and testing set. This is done to ensure that the resultant model will not automatically return a biased output; as such the model should not be trained on the same data that it is applied to. To limit any biases in the unsupervised learning, we follow this standard practice by requiring both training and testing data sets, such that each of the data sets is excluded from each other. A validation set is used in order to validate the performance of the model during training.

To retain as many observations in the testing set as possible, a 20:20:60 training, validation, and testing split was applied to 19979192 flattened $f(204, t)$ REPT observations. The splitting of the data was achieved based on the randomly shuffled data points using the TRAIN_TEST_SPLIT Python tool (Pedregosa et al. 2011) with a random seed of 4. By splitting the data randomly, each of the data sets contains a mix of both quiet and active times. Each data set will contain observations for every day within the 7-yr window. Therefore, as geomagnetic storms are multiday events (Murphy et al. 2020), it is guaranteed that each data set will contain a mix of all geomagnetic conditions, which is critical for the machine learning tools to learn and distinguish between these different activity times. The numbers of observations included in each of the training, validation, and testing sets, respectively, are shown in Table 1.

When using a time series data set, splitting the data in this manner may result in the validation and training sets being correlated. This is because the dynamic flux time is larger than that of the resolution of the data (Ma et al. 2022). Therefore, adjacent flux measurements will effectively be the same. We acknowledge the limitation of this splitting method; however, as this is an unsupervised learning problem, we collect no accuracy metrics and therefore the potential autocorrelation between the training and validation sets is not important. We limit the effect by randomly shuffling the REPT observations before randomly splitting to remove any bias due to the data being ordered by timestamp. A validation set was provided only to satisfy the building arguments of the first stage of the machine learning, an AE (Section 4.1), and to determine if the model is performing as expected during its learning phase before being applied to the testing set.

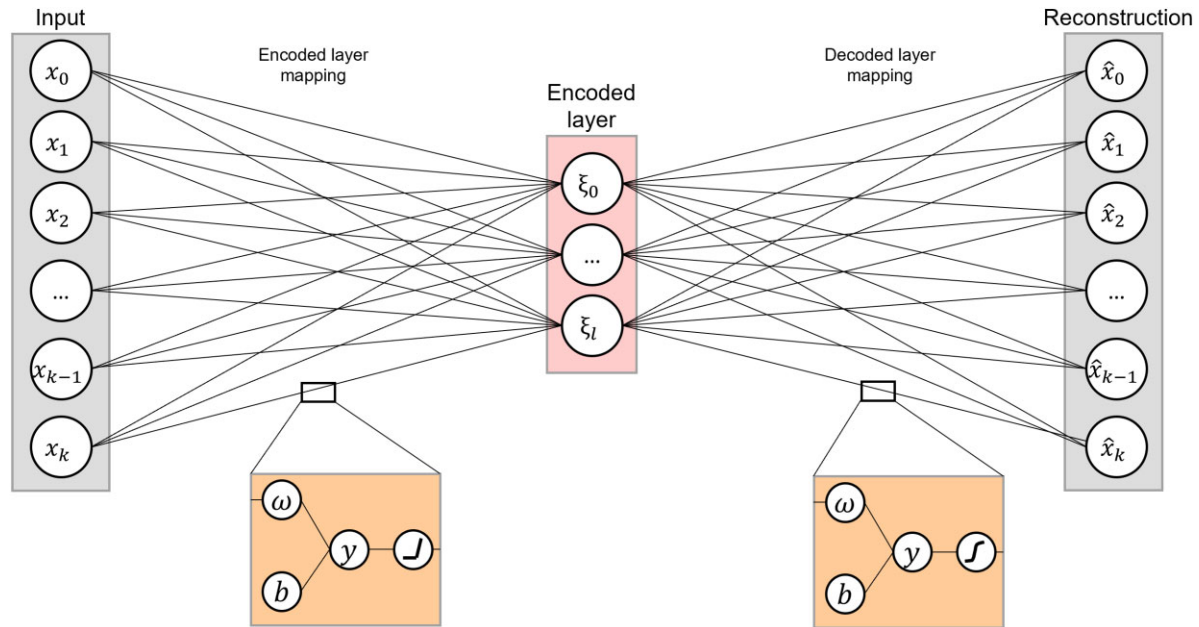


Figure 1. Schematic of the internal processes of an AE, where x_i denotes the input data with dimensions in the range of 0 to k , ω are the adjustable weights, b is the applied bias parameter, ξ_i represents the encoded layer neurons with dimensions in the range of 0 to l , where l is smaller than k , and \hat{x}_i is the reconstructed layer with dimensions the same as the input.

4 DIMENSION REDUCTION

Given that REPT flux measurements have a high number of dimensions, the data can be difficult for unsupervised classification tools to efficiently process (Bakrania et al. 2020a) and visualize. As some of the dimensions within the data will be dominated by noise, instrument sensitivity, and the instrument noise floor (Baker et al. 2013), to improve efficiency and denoise the data, the flattened 204D flux measurements underwent dimensionality reduction to compress the data into a more suitable representation. The compression of data to 3 dimensions was achieved in two stages: an AE (Section 4.1) and PCA (Section 4.2).

4.1 Autoencoders

An AE is a form of neural network that can be used to compress multidimensional data while retaining both the underlying linear and non-linear relationships within the data. The AE algorithm works by mapping each flattened data point (neuron) to a hidden layer of a lower number of dimensions (encoder) before then being mapped back to a layer of the same number of dimensions as the input (decoder). An example of an AE is shown in Fig. 1.

The mapping of each input neuron x_i to each new layer neuron y_j works through the multiplication of inputs and weights $y_j = \sum_i \omega_i x_i + b$. Here i is in the range of 0 to number of input dimensions k , j is of the range 0 to number of encoded dimensions l , ω refers to the weights associated with the mapping, and b is a bias unit which has been applied and adjusted to improve the performance of the AE (Bakrania et al. 2020a). The summation of weights and input neurons and the addition of a bias parameter results in an activation value y . The activation value is then transformed into a j th next layer neuron output by passing y_j through an activation function (Sharma, Sharma & Athaiya 2020). The AE repeats this process for a number of epochs (iterations) n , whereby the score of the previous epoch is used to update the weights and bias parameters. In this case the

score is the loss value, which should be minimized for an optimal solution (Chollet et al. 2015). The loss value is a measure of how much information is lost during the compression of the data and, in this case, is determined using the mean square error function $mse = \frac{1}{N} \sum (x_i - \hat{x}_i)^2$ of input x_i and reconstruction \hat{x}_i , where N is the number of data points. The AE algorithm was built and applied to the REPT flattened data using the Keras Python library (Chollet et al. 2015) with a single hidden layer, a batch-size of 256, a learning rate of 0.0005, 102 encoded dimensions, and an Adam optimizer. Batch-size determines the number of samples by splitting the data into batches. In order for the AE to be optimal, the batch-size needs to be as close to the number of dimensions of the data set. Given that the batch-size is given by a 2^n function where n is an integer (Bakrania et al. 2020a), we chose a batch-size of 256 (where $n = 8$) to most closely resemble the 204 dimensions of our data set. The learning rate modulates the change in learning rate of the data over time and has a default value of 0.001 (Chollet et al. 2015). The encoded dimensions refer to the smaller number of dimensions (neurons) in the hidden layer of the AE, and the Adam optimizer was used for computational efficiency on large data sets (Kingma & Ba 2014; Chollet et al. 2015). A stopping function was applied such that the AE would stop iterating when it had reached minimum loss value with a patience of 5, where patience refers to the number of epochs where the loss value has not improved (Chollet et al. 2015) and is used to prevent overfitting the training data. To consider the reproducibility of results, a TensorFlow (Abadi et al. 2016) random seed of 1 was applied.

To transform the activation values into next layer neuron outputs, we opted to use a Rectified Linear Unit (ReLU) activation function for the mapping between the input and encoded layers, and a sigmoid activation function for the mapping between the encoded and reconstruction layers.

ReLU is a popular non-linear activation function described by $\max(0, y)$. This means that for positive activation values, the function behaves as a simple linear function; however, for negative

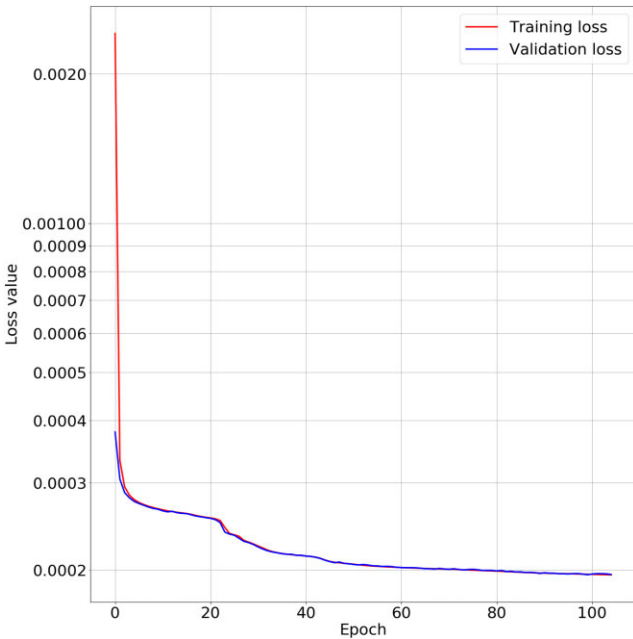


Figure 2. Resulting AE loss curve for training (red) and validation (blue) sets with optimum parameters of a learning rate of 0.0005, batch-size of 256 and 102 encoded dimensions.

inputs, the output will be saturated at 0 (Basodi et al. 2020). A sigmoid activation function was applied to the reconstruction layer to add non-linear and more complex relationships in the reconstruction in attempt to recreate the original input data. A sigmoid activation is a non-linear function described by $\frac{1}{1+e^{-y}}$, where y is the activation value (Dubey, Singh & Chaudhuri 2021).

The AE needed to be tested during its building due to the number of hyperparameters needed. A summary of the parameter tests and their loss curves are shown in Appendix A. The learning rate, batch-size, and encoded dimensions were determined to have the most optimum values of 0.0005, 256, and 102, respectively. These parameters collectively gave the most minimum validation loss of 0.02 per cent with a loss curve showing no signatures of over or underfitting and it is shown in Fig. 2. The validation curve closely

follows that of the training loss, suggesting that the model has not been overfitted. As both the training and validation curves decrease over time, this implies that the model was able to learn the training set and hence the model has not been underfitted either.

Once trained, the AE has then applied to our test data set. The reconstructed $f(PA, E, t)$ is then compared to that of the original input, an example of which is shown in Fig. 3. The overall shape of the reconstruction distribution (Fig. 3b) remains the same as the original (Fig. 3a) with no signatures of severe data loss or introduction of missing data or anisotropies. While there are minor differences, shown by the flux differences in Fig. 3c, the overall shape and magnitudes remain almost identical given the chosen discretization of the colour scale. However, it should be noted that there are small changes of less than 5 per cent resulting from the information loss, which we consider to be minimal. By comparing the original and reconstructed data which was created from the encoded layer with a lower number of dimensions, Fig. 3 gives confidence that the AE is functioning correctly and with a small loss of data that does not impact the shape of the distributions.

4.2 Principal component analysis

Once the REPT data was initially compressed to 102 dimensions, a PCA was applied to further compress the data to three dimensions following Bakrania et al. (2020a). PCA determines the data’s eigenvectors (or principal components) by calculating the covariance matrix of the data. The eigenvectors returned from the covariance matrix are hierarchical, meaning the top eigenvectors contain the most important information. As three dimensions was the desired output, the top three eigenvectors \mathbf{A} were determined. The 102D data x was then translated to 3D principal component space \mathbf{Z} through $\mathbf{Z} = x\mathbf{A}$ (Bakrania et al. 2020a).

The PCA algorithm was applied using the PCA Python tool from the scikit-learn package (Pedregosa et al. 2011). The top three principal components account for a total of 74.47 per cent of the information (45.06 per cent, 23.61 per cent, and 6.30 per cent, respectively), meaning that we can describe over 70 per cent of the 204D data with only three dimensions. A sensitivity test was performed in order to understand whether three components were sufficient to describe this data set. The fourth principle component yielded only an extra 4.69 per cent of information, therefore suggesting that the first three components were sufficient to describe the REPT data in our analysis.

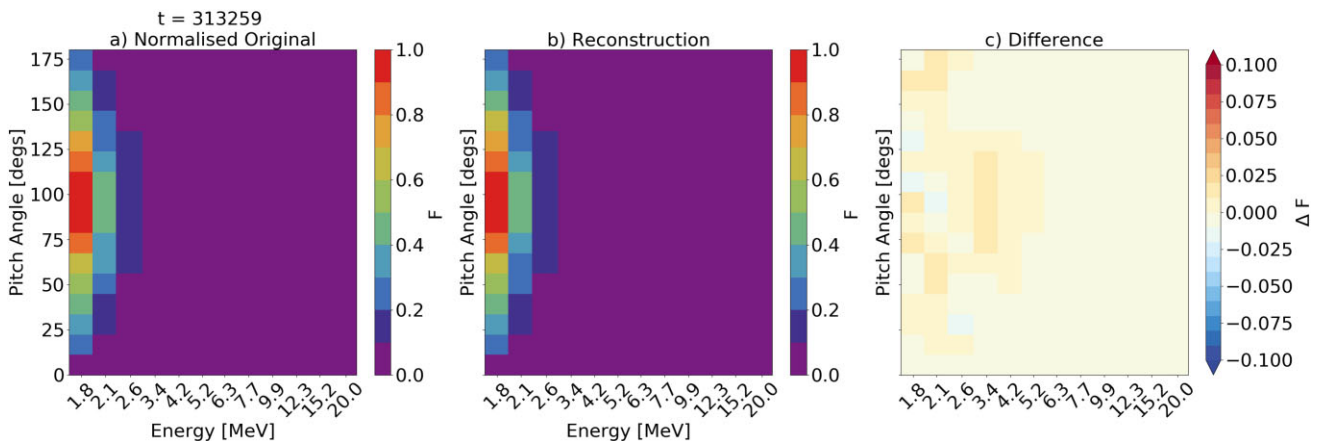


Figure 3. (a) Pitch angle – energy distribution example from one of the random observations in the testing set. (b) The resulting AE reconstruction of the input data. (c) The difference between the original and reconstruction. While there are no large local differences, there are slight differences globally. These differences are due to information loss.

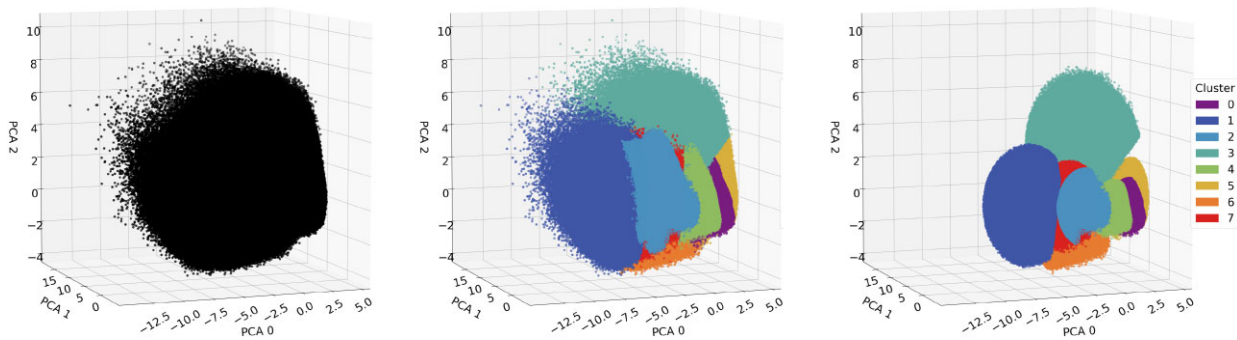


Figure 4. (a) 3D representation of Van Allen Probe B REPT electron flux data in principal component space. Each of the dimensions corresponds to one of the principal components which contain the most important information. (b) The 3D representation of REPT data in each of the three PCA eigenvector space, where observations have been partitioned into eight predicted clusters using k-means. Each cluster has been arbitrarily assigned a different colour. (c) The 3D REPT data in PCA space, with a 95 per cent confidence interval applied to each cluster to restrict the effect of ambiguous classifications at the boundaries due to small spatial separations between clusters.

We chose three dimensions in order to visualize the three most critical dimensions of the Van Allen Probe REPT data. The resulting 3D data are shown in Fig. 4a where each dimension is one principal component. The 2D projections are shown in Appendix D. Overall, the data set is still very dense, and it is not clear how to identify any data that have similar PAD shapes. Hence, we need to employ a classification technique in order to identify PADs that have common shapes.

5 CLUSTERING

Unsupervised machine learning is utilized due to the unlabelled nature of the data. Unlabelled data means that as well as unknowing the truths of our data, the number of classifications within the data first needs to be predicted before any classifications can be made. In order to classify these 12 million data points into meaningful physical categories, we use a mean shift algorithm (Section 5.1) to predict the number of clusters of data, before using a k-means algorithm (Section 5.2) to classify the data itself.

5.1 Mean shift

Mean shift is a non-parametric clustering technique based on the density function of the data (Fukunaga & Hostetler 1975), which in this case is 3D REPT data. Using the MEANSHIFT Python tool in the scikit-learn package (Pedregosa et al. 2011), we can apply a mean shift algorithm to predict the number of distinct classifications or clusters of data within the REPT data. The algorithm works by iterating between calculating and translating the mean shift (a vector that describes the distance and direction of the nearest density function maxima) of each data point until convergence onto a density function maximum. The predicted number of clusters is then given by the total number of density function maxima (Fukunaga & Hostetler 1975; Bakrania et al. 2020a). A mathematical description of the mean shift algorithm is given in Appendix B.

The mean shift algorithm requires a data-dependent bandwidth that defines the size of the window used to calculate the mean, which needs to be optimized for each data set. To optimize the bandwidth, the ESTIMATE_BANDWIDTH Python tool from the scikit-learn package (Pedregosa et al. 2011) was used, with 6000 samples, a quantile factor of 0.1, and a random state of 0. The algorithm runs on time-scales comparable to the quadratic of the number of samples (Pedregosa et al. 2011), therefore 6000 samples were chosen to consider computational expense while yielding a comparable bandwidth estimation to a test ran with 60 000 samples. The quantile factor relates to the

proportion of data that is considered in the estimation of the bandwidth where from experience, typical values range between 0.05 and 0.3 depending on the data set and computational resources. A summary of the bandwidth optimization is given in Appendix C with respect to the Calinski–Harabasz (CH) score and Davies–Bouldin (DB) index, where the CH score describes the variance between clusters and should be as high as possible. The DB index describes the similarity between clusters and should be as low as possible. Mathematical descriptions for these statistics are also supplied in Appendix C.

For the 3D representation of REPT data, with an optimal bandwidth of 1.949, the mean shift algorithm predicted eight clusters with a CH score of 1.990×10^6 and DB index of 0.941.

5.2 K-means

The 3D data can be partitioned into this mean shift predicted number of clusters using a k-means algorithm. We have applied k-means clustering using the KMEANS Python tool from the scikit-learn package (Pedregosa et al. 2011) with a random state of 0. K-means algorithms are computationally simple and fast to process, making them advantageous even on large data sets. Moreover, k-means only relies on a single parameter – the number of clusters k . K-means is based on minimizing the sum-of-squares (known as inertia) between data points in each cluster and works by assigning a k number of cluster centroid positions. The Euclidean distance $d_{\text{euc}} = \|x_i - C_j\|$ is calculated for each data point x_i within N observations and each cluster centroid C_j . Observations are then assigned to the closest cluster.

The centroid positions are recalculated by taking the mean position of the intra-cluster data points. The process iterates between determining the nearest centroids and recalculating the cluster means until the centroid position variance between iterations is zero. The algorithm iterates to locally make improvements to the partitions in order to minimize the inertia (Arthur & Vassilvitskii 2007; Pedregosa et al. 2011).

By using the k-means algorithm, the 3D REPT was partitioned into the eight predicted clusters from the mean shift (Section 5.1) with a CH index of 8.258×10^6 and DB index of 0.997. The resulting 3D representation in PCA space is shown in Fig. 4b, where each cluster is represented by a different colour.

To test the stability of the results, a k-means test was also performed with a random state of 4. The result yielded less than a 1 per cent difference in the number of observations in each cluster,

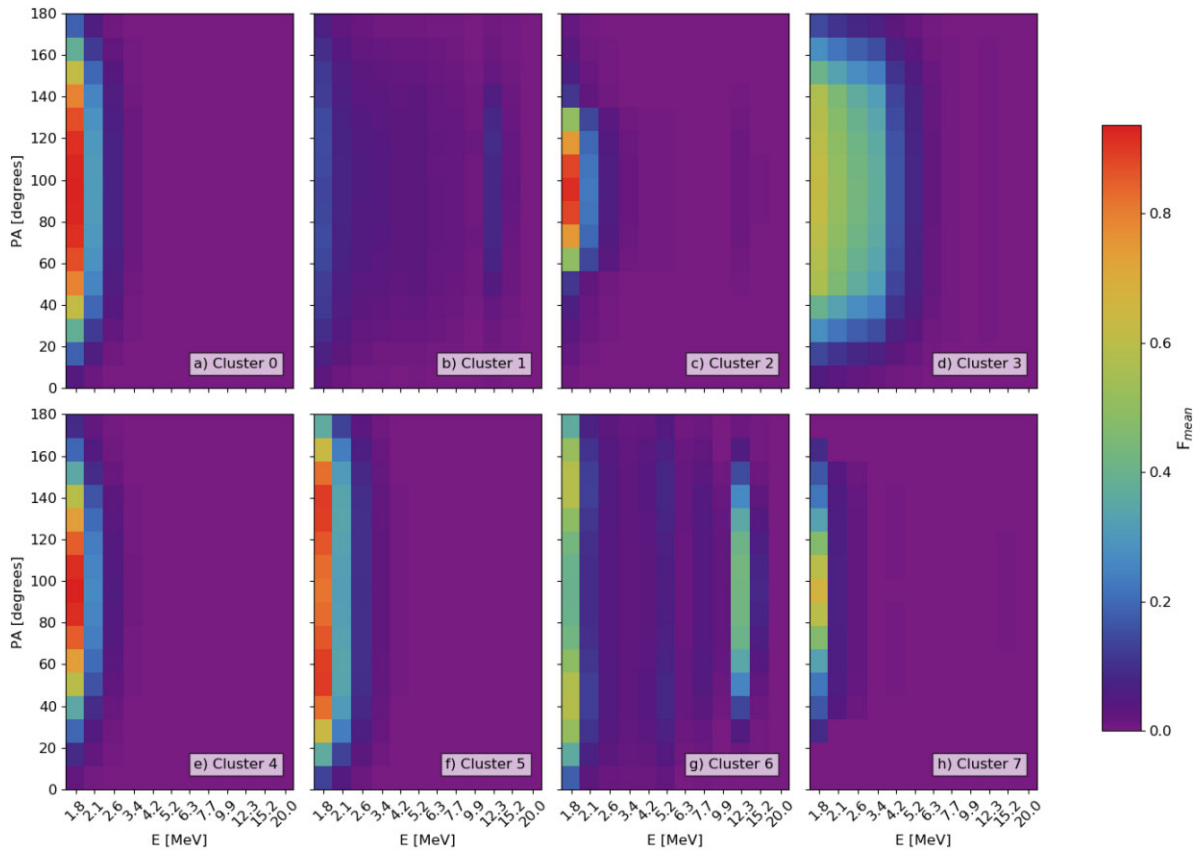


Figure 5. The mean PAD for each cluster across all energies. The colour bar represents the mean normalized flux. Each of the clusters shows a differently shaped distribution where for lower energies Cluster 0 (a) and Cluster 1 (b) show flattop like distributions where Cluster 1 is a significantly weaker distribution. (c) Cluster 2 is a narrow pancake distribution centred on 90° . (d) Cluster 3 is another type of flattop that extends to ultra-relativistic energies. (e) Cluster 4 is a broader pancake than Cluster 2. Cluster 5 (f) and Cluster 6 (g) have butterfly-like shapes at lower energies; however, at energies higher than 10 MeV Cluster 6 shows a flattop-feature. (h) Cluster 7 is the third pancake-like distribution which is much weaker than the distributions of Clusters 2 and 4.

evidencing that eight clusters is stable and valid regardless of the random sampling. This test is shown in Appendix E.

6 CLASSIFICATIONS

6.1 Cluster reduction

Fig. 4b shows the positions of the eight clusters in 3D PCA space. From Fig. 4b we can see that there are regions of well separated data points. However, at the borders of each cluster, there is little separation, meaning that in those regions, it is difficult to reliably determine which cluster the data point most closely resembles. Therefore, to interpret the clusters, we limit our classifications by focusing on observations more centrally concentrated within the cluster. K-means, with a Euclidean distance metric, attempts to find spherically shaped clusters within the data (Jain 2010) and calculated inertia on the assumption that the clusters are isotropic (Pedregosa et al. 2011). We therefore take a 2σ confidence interval around each cluster centroid by calculating the Euclidean distances of intra-cluster observations from the cluster centroids and taking the top 95 per cent of observations.

The resulting clusters in 3D PCA space are shown in Fig. 4c. These reduced clusters can be seen to be more spatially separated in PCA space, and therefore any potential ambiguous classifications at the boundaries have been restricted.

6.2 Pitch angle distributions

To recap, each data point within the clustered data corresponds to an individual relativistic electron energy-dependent PAD. To investigate the shape of the PADs, the mean and median fluxes of the normalized observations in each cluster were calculated. The mean and median PADs are shown in Fig. 5 and 6, respectively. We present both the mean and median PADs in order to demonstrate that our results are not skewed by statistical outliers. The mean and median PADs with respect to 1.8 MeV are shown in Appendix F.

Each of the mean PADs in Fig. 5 indicate a different shape of the distribution in energy–pitch angle space, which correspond to physically meaningful PADs resultant from processes operating in the radiation belts. Fig. 5 shows the mean flux as a function of PA (y-axis) and E (x-axis). We notice that each of the clusters display different distributions. Clusters 2, 4, and 7 (Figs 5c, e, and h) show different variations of a ‘pancake distribution’, which is described as a peaked distribution centred on 90 degree (e.g. West, Buck & Walton 1973; Horne et al. 2003; Gannon et al. 2007; Chakraborty et al. 2022). Fig. 5c shows a narrower pancake distribution for Cluster 2 than Cluster 4, whereas Cluster 7 (Fig. 5h) is a much weaker pancake than both Cluster 2 and 4. Clusters 5 and 6 (Figs 5f and g) show a butterfly-like distributions at low energies (<2.5 MeV), where there is a decrease in flux at 90 degree (e.g. West et al. 1973; Selesnick & Blake 2002; Gannon et al. 2007; Zhao et al. 2018; Chakraborty et al. 2022). Fig. 5g however also shows a high

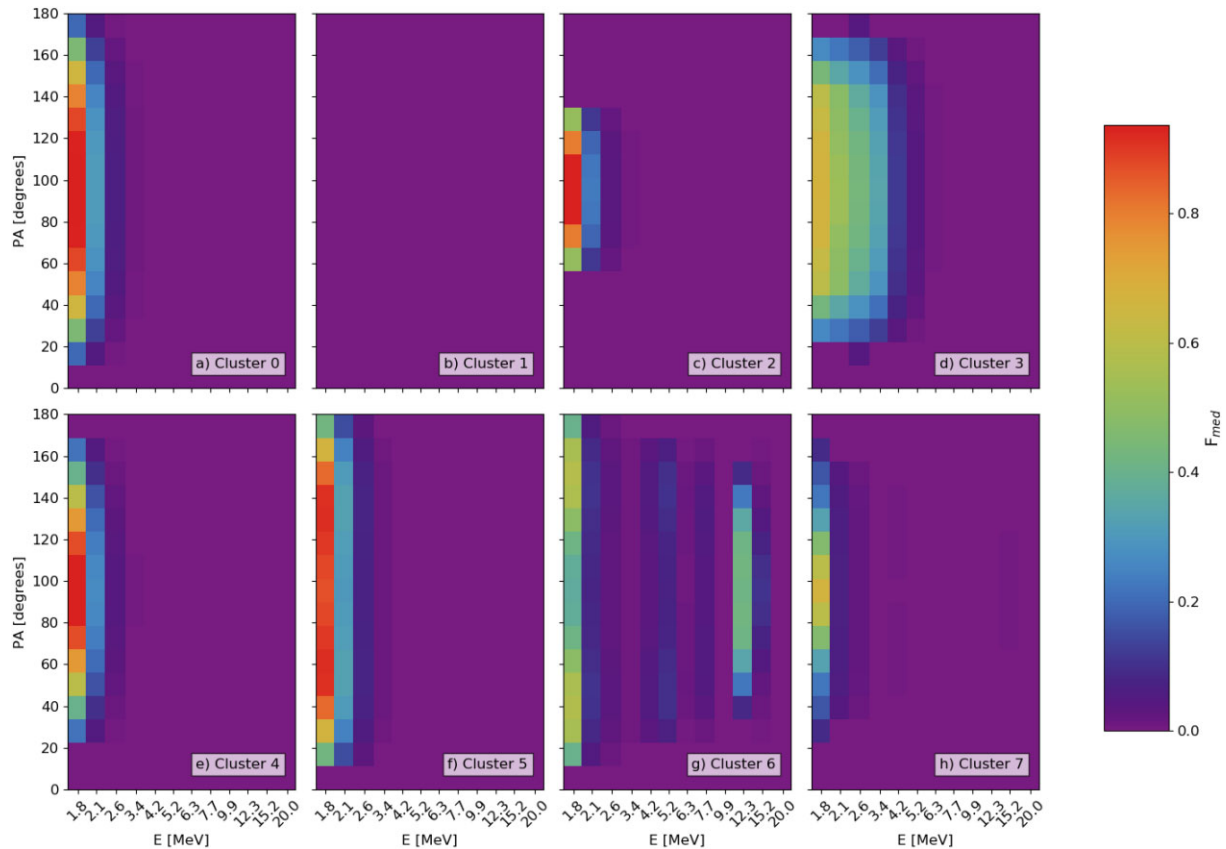


Figure 6. The median PADs for each cluster across all energies, with the colour bar representing the median normalized flux. The shapes of Clusters 0, 2, 3, 4, 5, 6, and 7 (a, c, d, e, f, g, and h, respectively) are the same as the mean distributions shown in Fig. 5. Cluster 1 (b) however shows a median normalized flux of 0 that is a result of partitioning noisy observations together.

energy (>10 MeV) flattop feature, a signature that is not dominant in the other cluster distributions. This high-energy PAD may have been caused by proton contamination in the inner radiation belt (e.g. Claudepierre et al. 2019). Cluster 0 (Fig. 5a) shows evidence of a flattop distribution which has a plateau of constant flux at pitch angles of approximately 60–130 degree (e.g. Horne et al. 2003; Zhao et al. 2018) and appears to be an intermediate distribution between Cluster 4 and Cluster 5 (Figs 5e and f) perhaps corresponding to evidence of a transition between pancake-like, flattop, and butterfly distributions (Horne et al. 2003; Gannon et al. 2007). Fig. 5b shows a low-flux flattop distribution for Cluster 1 whereas Cluster 3 (Fig. 5d) displays a flattop that extends to ultra-relativistic energy ranges (~ 7 MeV). Future applications of this work will entail looking more closely at these ultra-relativistic flattop observations to determine if these distributions have resulted from noise or been driven by a physical magnetospheric process.

In terms of the median distributions (Fig. 6), for Clusters 0, 2, 3, 4, 5, 6, and 7, the shapes of the distributions remain the same as their mean distributions in Fig. 5, with slight differences only in the values of the average flux. However, Cluster 1 (Fig. 6b) shows zero flux at all pitch angles and energies. When investigated further, each individual observation within this cluster displayed only signatures that corresponded to noisy or missing data. Similarly, observations in Cluster 7 (weak pancake, Fig. 5h), only contained observations of low counts, close to that of the noise floor. Hence, our technique has partitioned these noisy, low-count observations together within two clusters, which can then be neglected from further study. Our new technique has demonstrated that, by removing the clusters that

contained only noise and low counts, scientific data sets can be cleaned up and bad and/or missing data can be easily removed.

The distributions of the remaining six clusters at lower energies have shapes as expected from e.g. Chakraborty et al. (2022), Zhao et al. (2018), Souza et al. (2016), Gannon et al. (2007), and Horne et al. (2003). As the distributions are clearly different shapes, this work suggests that these different types of distributions have resulted from different magnetospheric drivers as discussed in e.g. Chakraborty et al. (2022), Chu et al. (2021), and Li & Hudson (2019). Therefore, future applications of this algorithm and data set include identifying the magnetospheric phenomena driving each individual cluster and investigating the properties and spatial and temporal dependence of the six physically meaningful clusters.

7 CONCLUSION

The behaviour of the radiation belts is difficult to diagnose due to their dynamic nature and unpredictability. We can begin to understand the plasma conditions of the radiation belts by looking at the energy-dependent PADs of the relativistic electrons. The Van Allen Probe mission provides over 7 yrs worth of relativistic electron observations with a temporal resolution of the order of tens of seconds (Baker et al. 2013). This means there are millions of PADs available to analyse, which is impossible by eye. However, by using a new amalgamation of unsupervised machine learning techniques adapted from Bakrania et al. (2020a), entire relativistic electron distributions over all energies and pitch angles have been characterized over the entire Van Allen Probe mission.

We use an AE and PCA to reduce the dimensions of Van Allen Probe REPT data $f(PA, E, t)$ to a more manageable number for visualization, in this case – three dimensions. The now 3D data was then able to be classified through unsupervised machine learning tools by using mean shift to predict the number of clusters and k-means to partition the data into the predicted number of classifications.

Using this method, a total of eight different clusters were identified. Upon investigating the average energy-dependent PADs, one of these clusters contained only low counts and spurious data with an average flux of zero and the other contained low counts of noisy, asymmetric pancake distributions – counts so low they can be considered negligible. The remaining six clusters displayed average PAD shapes as expected from previous distribution studies at energies lower than 2.5 MeV – either flattop, butterfly, or pancake (Gannon et al. 2007; Souza et al. 2016; Zhao et al. 2018; Chakraborty et al. 2022), with some clusters being wider in pitch angle space than others potentially revealing an evolution between distribution types. Future applications of this work will be to identify the physical processes that drive the behaviour of the relativistic electrons by investigating the properties and spatial and temporal dependence of the six clusters. Our findings show that this technique can be used to both classify and denoise large data sets of multidimensional data in space plasma physics, using Van Allen Probe relativistic electrons as an exemplar. Given that we are looking for specific particle distributions in large amounts of data, this machine learning technique could be used to mine data sets from other plasma missions, where data is either less plentiful or incomplete. These data sets could include new inner heliospheric missions such as Parker Solar Probe and Solar Orbiter, previous missions such with less data availability such as Mars Express and Cassini, and potentially new missions such as JUICE.

ACKNOWLEDGEMENTS

SK is indebted to Northumbria University and STFC grant 2597922 for PhD studentship support. IJR, SC, and JKS are funded in part by STFC grants ST/V006320/1, and NERC grants NE/V002554/2 and NE/P017185/2. RW was supported by a Royal Astronomical Society Summer Studentship. AWS was supported by NERC Independent Research Fellowship NE/W009129/1. MRB was supported by a UCL Impact Studentship, joint funded by the ESA NPI programme. CEJW is supported by NERC grant NE/V002759/1 and STFC grant ST/W000369/1.

DATA AVAILABILITY

Level-3 Van Allen Probe REPT data are publically available and can be accessed at <https://cdaweb.gsfc.nasa.gov/>. The authors share their unsupervised machine learning technique code and can be found at https://github.com/ASTRO-IOM/Clustering_VAP.

REFERENCES

- Abadi M. et al., 2016, preprint (arXiv:1603.04467)
 Amaya J., Dupuis R., Innocenti M. E., Lapenta G., 2020, *Front. Astron. Space Sci.*, 7
 Artemyev A., Agapitov O., Mourenas D., Krasnoselskikh V., Shastun V., Mozer F., 2016, *Space Sci. Rev.*, 200, 261
 Arthur D., Vassilvitskii S., 2007, Proc. 18th Annu. ACM-SIAM Symp. Discrete Algorithms, SODA '07, K-Means++: The Advantages of Careful Seeding. Soc. Ind. Appl. Math., USA, p. 1027
 Baker D. N. et al., 2013, *Space Sci. Rev.*, 179, 337
 Baker D. N., Erickson P. J., Fennell J. F., Foster J. C., Jaynes A. N., Verronen P. T., 2018, *Space Sci. Rev.*, 214, 17

- Bakrania M. R., Rae I. J., Walsh A. P., Verscharen D., Smith A. W., 2020a, *Front. Astron. Space Sci.*, 7, 80
 Bakrania M. R., Rae I. J., Walsh A. P., Verscharen D., Smith A. W., Bloch T., Watt C. E. J., 2020b, *A&A*, 639, A46
 Basodi S., Ji C., Zhang H., Pan Y., 2020, preprint (arXiv:2006.10560)
 Bloch T., Watt C., Owens M., McInnes L., Macneil A., 2020, *Sol. Phys.*, 295, 41
 Bortnik J., Li W., Thorne R. M., Angelopoulos V., 2016, *J. Geophys. Res.: Space Phys.*, 121, 2423
 Breuillard H., Dupuis R., Retino A., Le Contel O., Amaya J., Lapenta G., 2020, *Front. Astron. Space Sci.*, 7, 55
 Camporeale E., Caré A., Borovsky J. E., 2017, *J. Geophys. Res.: Space Phys.*, 122, 10,910
 Chakraborty S., Chakraborty D., Reeves G., Baker D., Rae I. J., 2022, *Front. Astron. Space Sci.*, 9, 986061
 Chaston C. C., Bonnell J. W., Halford A. J., Reeves G. D., Baker D. N., Kletzing C. A., Wygant J. R., 2018, *Geophys. Res. Lett.*, 45, 9344
 Cheng I. K., Achilleos N., Smith A., 2022, *Front. Astron. Space Sci.*, 9
 Chollet F. et al., 2015, Keras, <https://keras.io/> (accessed 2022 July)
 Chu X. et al., 2021, *Space Weather*, 19, e2021SW002808
 Claudepierre S. G. et al., 2019, *J. Geophys. Res.: Space Phys.*, 124, 934
 Comanicu D., Meer P., 2002, *IEEE Trans. Pattern Anal. Mach. Intell.*, 24, 603
 Davies D. L., Bouldin D. W., 1979, *IEEE Trans. Pattern Anal. Mach. Intell.*, PAMI-1, 224
 Dubey S. R., Singh S. K., Chaudhuri B. B., 2021, *Neurocomputing*, 503, 92
 Fukunaga K., Hostetler L., 1975, *IEEE Trans. Inf. Theory*, 21, 32
 Gannon J. L., Li X., Heynderickx D., 2007, *J. Geophys. Res.: Space Phys.*, 112, A05212
 Herrera D., Maget V. F., Sicard-Piet A., 2016, *J. Geophys. Res.: Space Phys.*, 121, 9517
 Horne R. B., Meredith N. P., Thorne R. M., Heynderickx D., Iles R. H. A., Anderson R. R., 2003, *J. Geophys. Res.: Space Phys.*, 108, 1016
 Innocenti M. E., Amaya J., Raeder J., Dupuis R., Ferdousi B., Lapenta G., 2021, *Ann. Geophys.*, 39, 861
 Jain A. K., 2010, *Pattern Recognit. Lett.*, 31, 651
 Kingma D. P., Ba J., 2014, preprint (arXiv:1412.6980)
 Kyan M., Muneesawang P., Jarrah K., Guan L., 2014, *Unsupervised Learning*, p. 1
 Li W., Hudson M., 2019, *J. Geophys. Res.: Space Phys.*, 124, 8319
 Liu C. M., Fu H. S., Liu Y. Y., Wang Z., Chen G., Xu Y., Chen Z. Z., 2020, *J. Geophys. Res.: Space Phys.*, 125, e2020JA027777
 Ma D. et al., 2022, *Space Weather*, 20, e2022SW003079
 Maimaiti M., Kunduri B., Ruohoniemi J. M., Baker J. B. H., House L. L., 2019, *Space Weather*, 17, 1534
 Mauk B. H., Fox N. J., Kanekal S. G., Kessel R. L., Sibeck D. G., Ukhorskiy A., 2013, *Space Sci. Rev.*, 179, 3
 Murphy K. R., Mann I. R., Sibeck D. G., Rae I. J., Watt C., Ozeke L. G., Kanekal S. G., Baker D. N., 2020, *Space Weather*, 18, e2020SW002477
 Ozeke L. G., Mann I. R., Olifer L., Claudepierre S. G., Spence H. E., Baker D. N., 2022, *J. Geophys. Res.: Space Phys.*, 127, e2021JA029907
 Pedregosa F. et al., 2011, *J. Mach. Learn. Res.*, 12, 2825
 Rae I. J. et al., 2018, *J. Geophys. Res.: Space Phys.*, 123, 1900
 Reeves G. D., McAdams K. L., Friedel R. H. W., O'Brien T. P., 2003, *Geophys. Res. Lett.*, 30, 1529
 Ripoll J.-F. et al., 2020, *J. Phys.: Conf. Ser.*, 1623, 012005
 Rodger C. J. et al., 2007, *J. Geophys. Res.: Space Phys.*, 112, A11307
 Selesnick R. S., Blake J. B., 2002, *J. Geophys. Res.: Space Phys.*, 107, 1265
 Sharma S., Sharma S., Athaiya A., 2020, *Intern. J. Engineer. Appl. Sci. Technol.*, 4, 310
 Smith A. W., Rae I. J., Forsyth C., Oliveira D. M., Freeman M. P., Jackson D. R., 2020, *Space Weather*, 18, e2020SW002603
 Souza V. et al., 2016, *Space Weather*, 14, 275
 Staples F. A. et al., 2020, *J. Geophys. Res.: Space Phys.*, 125, e27289
 Summers D., Thorne R. M., 2003, *J. Geophys. Res.: Space Phys.*, 108, 1143
 West H. I. J., Buck R. M., Walton J. R., 1973, *J. Geophys. Res.*, 78, 1064
 Wing S., Turner D. L., Ukhorskiy A. Y., Johnson J. R., Sotirelis T., Nikoukar R., Romeo G., 2022, *Space Weather*, 20, e2022SW003090

Yeakel K. L., Vandegriff J. D., Garton T. M., Jackman C. M., Clark G., Vines S. K., Smith A. W., Kollmann P., 2022, *Front. Astron. Space Sci.*, 9, 875985

Zhao H. et al., 2018, *J. Geophys. Res.: Space Phys.*, 123, 3493

SUPPORTING INFORMATION

Supplementary data are available at [RASTAI](https://doi.org/10.1002/rastai) online.

supp data

Please note: Oxford University Press is not responsible for the content or functionality of any supporting materials supplied by the authors. Any queries (other than missing material) should be directed to the corresponding author for the article.

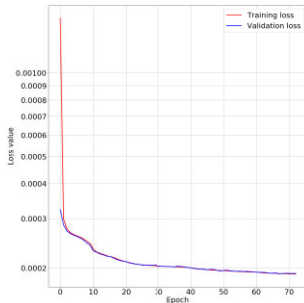
APPENDIX A: AE HYPERPARAMETER OPTIMIZATION

In order to optimize the AE, a series of parameter tests were performed. The test that produced the most minimum loss value and had the most desired loss curve (with no signatures of over or underfitting) was determined to be the most optimal set of parameters, a summary of the test results is shown in Table A1. The lowest overall loss was given by parameters of 102 encoded dimensions, batch-size of 256, and learning rate of 0.001 (test 0); however, the

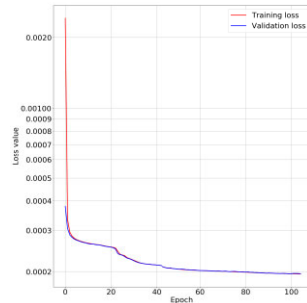
Table A1. AE parameter tests performed in order to determine the most optimum batch-size, number of encoded dimensions, and learning rate. The most optimum parameters are taken to be those that had most minimal loss values and smoothest loss curve. Some parameters gave the most minimal loss value but showed potential signatures of overfitting in their loss curves, for this reason those parameter combinations were not selected. The combination of parameters that gave the most minimum loss and had an agreeable loss curve were found to be 256, 102, and 0.0005 for the batch-size, number of encoded dimension, and learning rate, respectively.

Test	Batch-size	Encoded dimensions	Learning rate	Loss
0	256	102	0.001	0.000 191 129
1	256	102	0.0005	0.000 195 895
2	256	51	0.001	0.000 378 399
3	256	51	0.0005	0.000 377 934
4	512	102	0.001	0.000 198 566
5	512	102	0.0005	0.000 197 810
6	512	51	0.001	0.000 397 513
7	512	51	0.0005	0.000 397 850

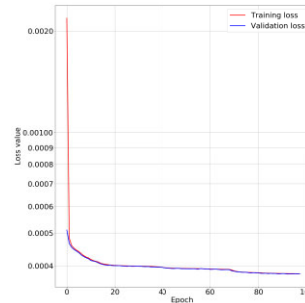
resulting loss curve (Fig. A1a) indicates a potential signature of overfitting. Therefore, we take the second most optimal parameters of 102 encoded dimensions, batch-size of 256, and learning rate of 0.0005 (test 1) as the resulting loss curve (Fig. A1b) does not show any noticeable signatures of overfitting.



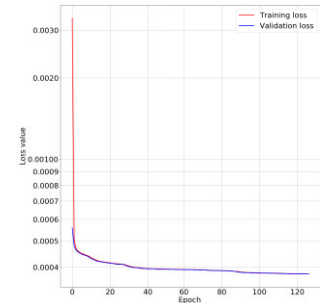
(a) Test 0: AE loss curve with parameters of 102 encoded dimensions, batch-size of 256 and learning rate of 0.001.



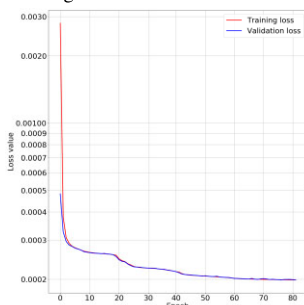
(b) Test 1: AE loss curve with parameters of 102 encoded dimensions, batch-size of 256 and learning rate of 0.0005.



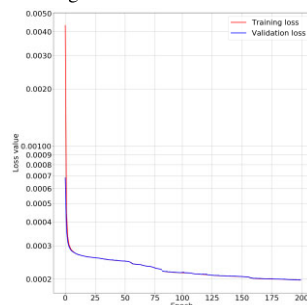
(c) Test 2: AE loss curve with parameters of 51 encoded dimensions, batch-size of 256 and learning rate of 0.001.



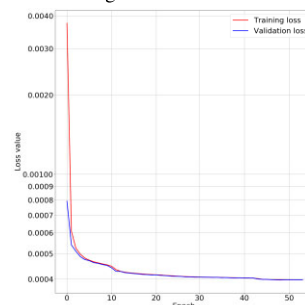
(d) Test 3: AE loss curve with parameters of 51 encoded dimensions, batch-size of 256 and learning rate of 0.0005.



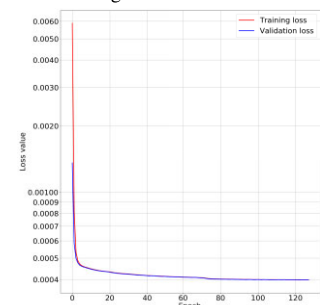
(e) Test 4: AE loss curve with parameters of 102 encoded dimensions, batch-size of 512 and learning rate of 0.001.



(f) Test 5: AE loss curve with parameters of 102 encoded dimensions, batch-size of 512 and learning rate of 0.0005.



(g) Test 6: AE loss curve with parameters of 51 encoded dimensions, batch-size of 512 and learning rate of 0.001.



(h) Test 7: AE loss curve with parameters of 51 encoded dimensions, batch-size of 512 and learning rate of 0.0005.

Figure A1. AE parameter test loss curves.

APPENDIX B: MEAN SHIFT MATHEMATICAL DESCRIPTION

The density function $f(x)$ is given by equation (B1), where N is the number of observations \mathbf{x}_i , n is the number of dimensions, K is a kernel function around the randomly assigned mean point \mathbf{x} and kernel bandwidth h (Fukunaga & Hostetler 1975; Bakrania et al. 2020a):

$$f(x) = \frac{1}{Nh^{n+1}} \sum_{i=1}^N K\left(\frac{x - x_i}{h}\right). \tag{B1}$$

To satisfy conditions of consistency and asymptotic unbiasedness, a Gaussian kernel k is applied to the K function (Fukunaga & Hostetler 1975). The kernel function weights of the nearest neighbour data points within the kernel window and is given by equation (B2), where c_k is a normalization constant (Bakrania et al. 2020a):

$$K\left(\frac{x - x_i}{h}\right) = c_k k\left(\left\|\frac{\mathbf{x} - \mathbf{x}_i}{h}\right\|^2\right). \tag{B2}$$

The density gradient of the density function (equation B1) with a Gaussian kernel (equation B2) is then derived to give

$$\nabla f(x) = \frac{2c_k}{Nh^{n+2}} \sum_{i=1}^N (\mathbf{x}_i - \mathbf{x}) \left(-k' \left(\left\|\frac{\mathbf{x} - \mathbf{x}_i}{h}\right\|^2\right)\right), \tag{B3}$$

where $-k'(X) = g(X)$ (Comaniciu & Meer 2002; Bakrania et al. 2020a). Rearranging and substituting in the mean shift $\mathbf{M}_h(\mathbf{x})$, equation (B3) becomes

$$\mathbf{M}_h(\mathbf{x}) = \frac{\sum_{i=1}^N x_i g\left(\left\|\frac{\mathbf{x} - \mathbf{x}_i}{h}\right\|^2\right)}{\sum_{i=1}^N g\left(\left\|\frac{\mathbf{x} - \mathbf{x}_i}{h}\right\|^2\right)} - \mathbf{x}, \tag{B4}$$

where the mean shift vector indicates the direction of the nearest maximum density for each data point x_i . The window function is then translated by shifting the mean, following equation (B5) (Bakrania et al. 2020a):

$$\mathbf{x}^{t+1} = \mathbf{x}^t + \mathbf{M}_h(\mathbf{x}^t). \tag{B5}$$

The mean shift recalculations and window functions translation will be iterated over t steps, until $M_h(x)$ converges onto a density function maximum and therefore when the density gradient is zero.

By repeating the process on each observation, each data point is shifted to converge on a density point maximum (Comaniciu & Meer 2002; Pedregosa et al. 2011; Bakrania et al. 2020a).

APPENDIX C: BANDWIDTH OPTIMIZATION

To optimize the bandwidth for the mean shift algorithm, a series of tests were performed. The tests were carried out using the scikit-learn ESTIMATE_BANDWIDTH tool (Pedregosa et al. 2011). This tool relies on defining specific parameters – number of samples, quantile factor, and number of parallel jobs that affect the performance of the function. A summary of the tests is shown in Table B1.

To evaluate the most optimum bandwidth, the CH score and DB index was calculated for each test. The test that returned the most maximum CH score and most minimum DB index was taken to be the optimum parameter.

The CH score is a statistical technique used when ground truths are not available. The index defines the variance ratio between clusters to describe their dispersion and separation following equation (C1), where the CH index (s) is ratio of the trace of the inter-cluster dispersion $tr(B_k)$ over the intra-cluster dispersion of all clusters $tr(W_k)$ for N number of observations and k number of clusters (Pedregosa et al. 2011):

$$s = \frac{tr(B_k) N - k}{tr(W_k) k - 1}. \tag{C1}$$

The larger the CH score, the denser and more separated the clusters are and the better the algorithm has performed. The CH index was used to optimize the bandwidth value, by selecting the bandwidth that resulted in a mean shift algorithm with the largest CH index.

The DB index was also used to optimize the bandwidth. The DB index describes the similarity R_{ij} between clusters (i) and their most similar cluster (j) with respect to their separation following equation (C2). The DB index measures the ratio of the cluster diameters (s) of clusters i and j (where i and $j = 1, \dots, k$ and $i \neq j$) from the average distance between the cluster’s centroid and each intra-cluster data point over the distance between the i and j centroids d_{ij} (Davies & Bouldin 1979; Pedregosa et al. 2011):

$$R_{ij} = \frac{s_i + s_j}{d_{ij}}. \tag{C2}$$

Table B1. Bandwidth optimization tests using the scikit-learn ESTIMATE_BANDWIDTH function showing the number of samples $n_samples$, quantile factor Q , number of parallel jobs n_jobs used in each test, the resulting bandwidth BW estimation, and number of clusters n , with their CH scores and DB indexes. In all cases, only a Q score of 0.1 yielded a result with more than one clusters.

n_samples	Q	n_jobs	BW	n	CH score	DB index
600	0.1	8	1967 550	8	1967 550	0.93 523
6000	0.05	2	1.466 269	20	1544 917	1.01 811
6000	0.1	2	1.94 923	8	1990 235	0.94 146
6000	0.2	2	2.647 787	1	NaN	NaN
6000	0.3	2	3.243 912	1	NaN	NaN
6000	0.05	4	1.466 269	20	1544 917	1.01 811
6000	0.1	4	1.949 230	8	1990 235	0.94 146
6000	0.2	4	2.647 787	1	NaN	NaN
6000	0.3	4	3.243 912	1	NaN	NaN
6000	0.05	8	1.466 269	20	1544 917	1.01 811
6000	0.1	8	1.949 230	8	1990 235	0.94 146
6000	0.2	8	2.647 787	1	NaN	NaN
6000	0.3	8	3.243 912	1	NaN	NaN
6000 000	0.1	8	1.95 022 898	8	1990 849	0.94 157

The maximum similarity (equation C3) for each cluster is then taken to be the DB index \bar{R} and relates to the appropriateness of the partitions within the data set without the dependence upon clustering method (Davies & Bouldin 1979):

$$\bar{R} = \frac{1}{N} \sum_{i=1}^N \max_{i \neq j} R_{i,j}. \quad (\text{C3})$$

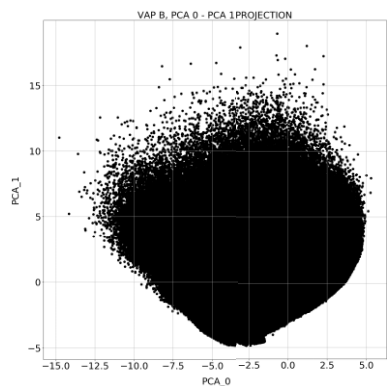
A DB index close to 0 indicates that no clusters are similar, and thus is the best possible score.

By using the ESTIMATE_BANDWIDTH function, the ambiguity of determining an optimal bandwidth manually based of their statistics has been limited and yielded the most optimal result when the

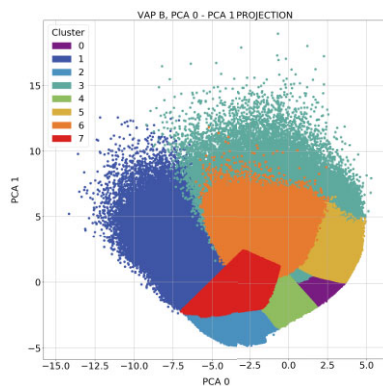
function was considered with 6000 samples and a quantile of 0.1. As there was no dependence on the number of parallel jobs, two jobs were applied to match the number of jobs used in the mean shift algorithm itself.

APPENDIX D: 2D PROJECTIONS

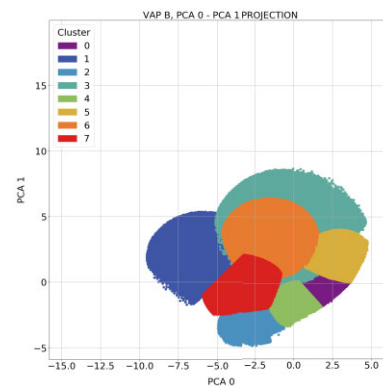
Fig. D1 shows the projections of the 3D representations of REPT data after PCA (left-hand panel), k-means (central panel), and 95 per cent confidence interval (right-hand panel) has been applied for the PCA 0–PCA 1 (top row), PCA 1–PCA 2 (middle row), and PCA 0–PCA 2 (bottom row) projections.



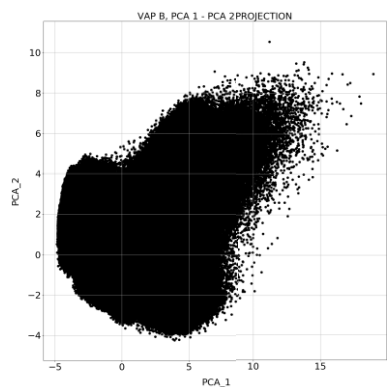
(a) PCA 0 - PCA 1 projection of the 3D representation of REPT data.



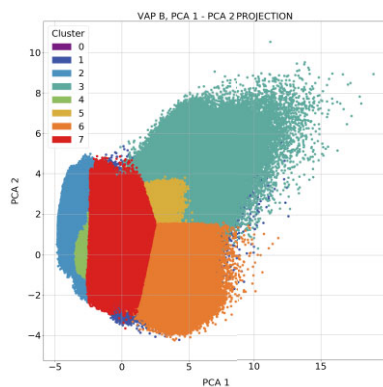
(b) PCA 1 - PCA 2 projection of the 3D representation of REPT data coloured by the assigned cluster number determined through k-means.



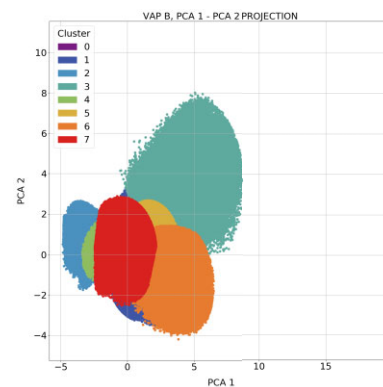
(c) PCA 0 - PCA 1 projection of the 3D representation of REPT data coloured by the assigned cluster number determined through k-means with an applied 95% confidence interval to remove ambiguous classifications at the boundaries of each cluster.



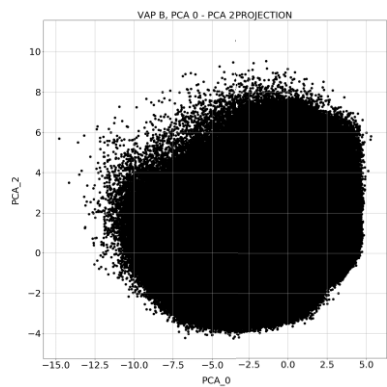
(d) PCA 1 - PCA 2 projection of the 3D representation of REPT data.



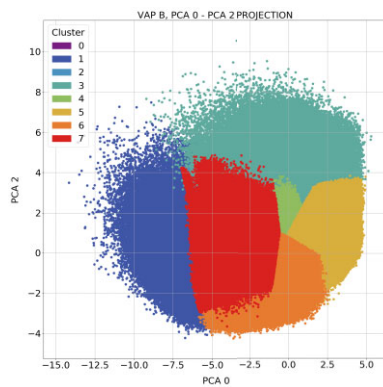
(e) PCA 1 - PCA 2 projection of the 3D representation of REPT data coloured by the assigned cluster number determined through k-means.



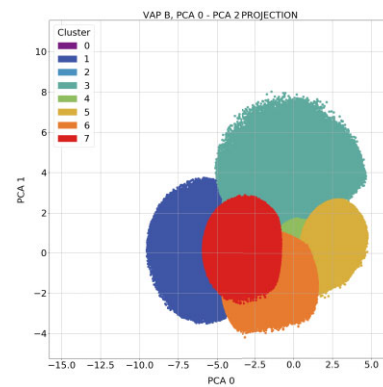
(f) PCA 1 - PCA 2 projection of the 3D representation of REPT data determined through k-means with an applied 95% confidence interval to remove ambiguous classifications at the boundaries of each cluster.



(g) PCA 0 - PCA 2 projection of the 3D representation of REPT data.



(h) PCA 0 - PCA 2 projection of the 3D representation of REPT data determined through k-means.



(i) PCA 0 - PCA 2 projection of the 3D representation of REPT data determined through k-means with an applied 95% confidence interval to remove ambiguous classifications at the boundaries of each cluster.

Figure D1. The 2D projections of the 3D representations of the REPT data in principal component space. The top row reflects the PCA 0–PCA 1 projection, the middle row reflects the PCA 1–PCA 2 projection, and the bottom row reflects the PCA 0–PCA 2 projection. The columns reflect the 3D observation, the resulting k-means classifications, and an applied 95 per cent confidence interval, respectively.

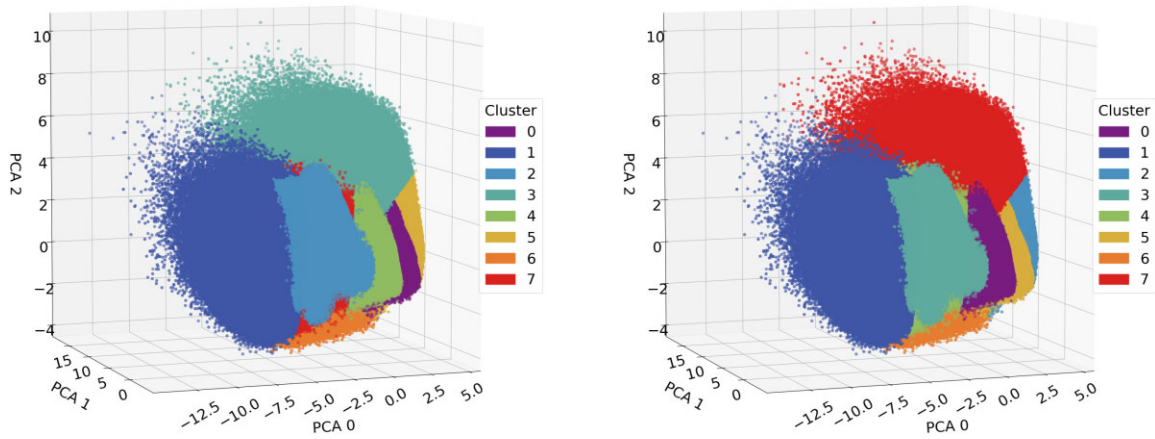


Figure E1. The 3D representation of REPT data in each of the three PCA eigenvector space, where observations have been partitioned into clusters using k-means. Left-hand panel: The resulting k-means classifications using a random state of 0, the result used throughout this study. Right-hand panel: The resulting k-means classification test using a random state of 4. Though the cluster numbers are different due to the random placement of the k-means centroids, the number of observations within each cluster was within 1 per cent of the result from this study.

APPENDIX E: K-MEANS STABILITY TEST

To verify the stability of the k-means result, we ran a k-means test with a random state of 4. The results of the test show only a 1 per cent difference to the results of this study. The cluster numbers are different due to the random positioning of the k-means centroids and the resulting 3D k-means test compared to the results of this study are shown in Fig. E1.

APPENDIX F: 1.8 MEV-DEPENDENT PADS

Fig. F1 shows the mean PAD of each cluster with respect to the first energy bin (1.8 MeV). Fig. F2 shows the median PAD of each cluster with respect to the first energy bin (1.8 MeV). The energy-dependent PADs more clearly see the shapes of the distributions that are described in Section 6.

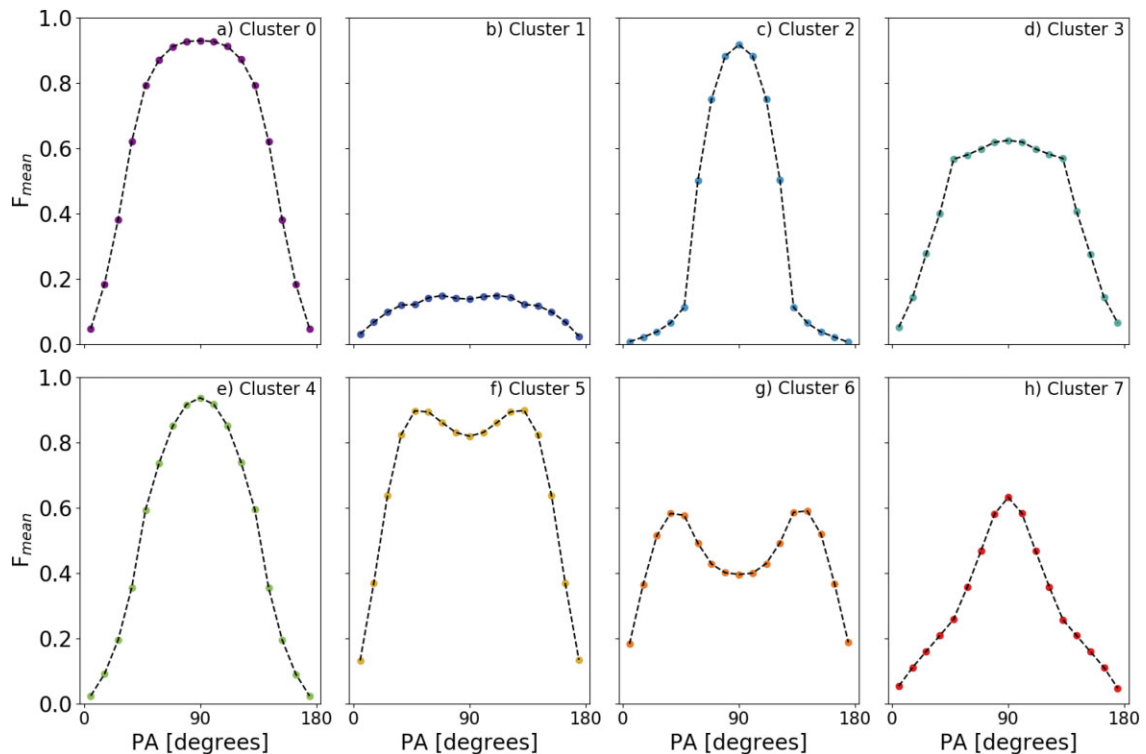


Figure F1. The mean PAD for each cluster at 1.8 MeV. Each of the eight clusters shows a differently shaped distribution. (a) Cluster 0 is a flat-topped peak centred across 90° , (b) Cluster 1 shows a significantly weak flat-topped peak, (c) Cluster 2 is a narrow peak, (d) Cluster 3 is a flat-topped peak, (e) Cluster 4 is a peak centred on 90° , (f) Cluster 5 is a butterfly distribution, (g) Cluster 6 also shows a butterfly distribution at lower energies but a flat-topped peak at energies higher than 12 MeV, and (h) Cluster 7 is a low flux narrow peak centred on 90° .

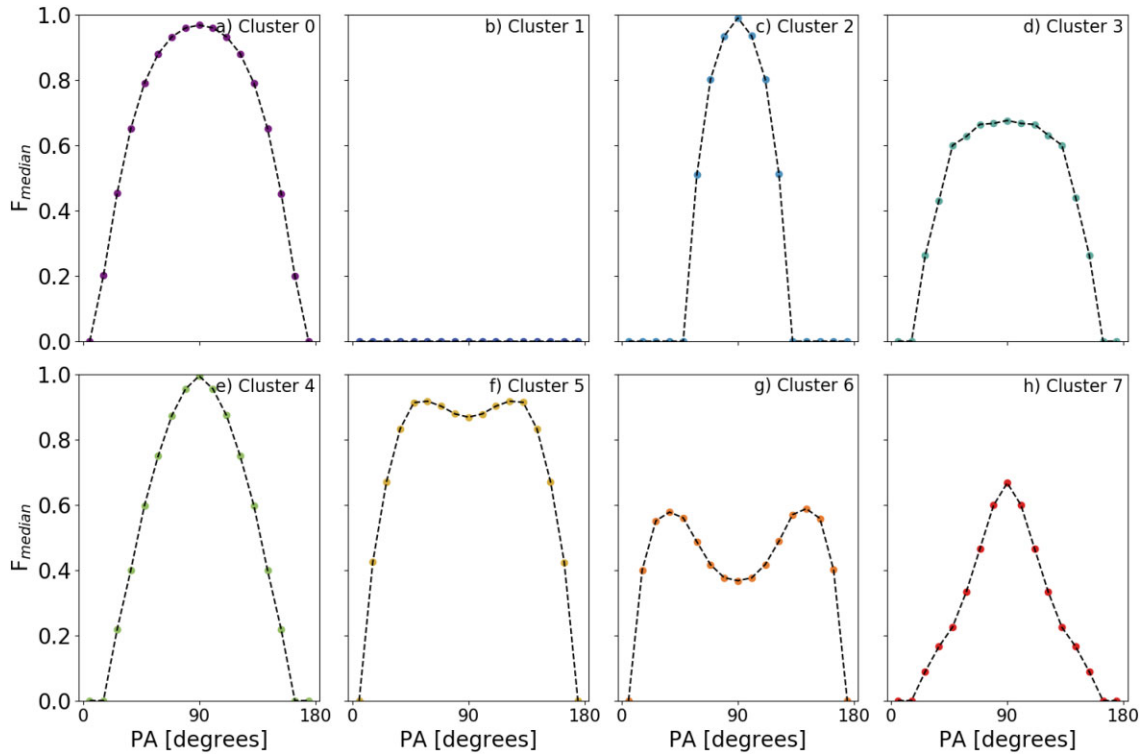


Figure F2. The median PAD for each cluster at 1.8 MeV. Each of the clusters shows a differently shaped distribution. (a) Cluster 0 is a flattop centred across 90° , (b) Cluster 1 shows a net flux of 0, (c) Cluster 2 is a narrow pancake, (d) Cluster 3 is flattop shaped, (e) Cluster 4 is a pancake centred on 90° , (f) Cluster 5 is a butterfly distribution, (g) Cluster 6 also shows a butterfly distribution at lower energies but a flattop at energies higher than 12 MeV, and (h) Cluster 7 is a low flux narrow peak centred on 90° .

This paper has been typeset from a $\text{\TeX}/\text{\LaTeX}$ file prepared by the author.