Predicting Solar Energetic Particle Event Energy Spectra Using Machine Learning Methods

Master's thesis
University of Turku
Physics
2026
B.Sc. Valtteri Waenerberg
Examiners:
Dr. Nina Dresing
Prof. Tapio Pahikkala

The originality of this thesis has been checked in accordance with the University of Turku quality assurance system using Turnitin Originality Check service.

UNIVERSITY OF TURKU
Department of Physics and Astronomy

Waenerberg, Valtteri Predicting Solar Energetic Particle Event Energy Spectra Using Machine Learning Methods

Master's thesis, 44 pp.
Physics
June 2026

The magnetic activity of the Sun can trigger various energy-releasing events, such as solar flares and coronal mass ejections (CMEs). These energy bursts can lead to acceleration and ejection of matter into interplanetary space, where they can be observed as solar energetic particle (SEP) events. On this thesis the focus is on predicting the proton peak energy fluxes observed during these events. It has been shown that the properties of the SEP event energy spectra are related to the properties of the associated phenomena, such as flares, CMEs and the solar wind conditions during the event.

In the last years, machine learning (ML) models have also been applied to predicting SEP events. In their 2024 paper [1], Liu et al. research team presented an iterative decision tree model for predicting SEP event energy spectra. In this thesis our first goal was to recreate this ML model and then expand our approach to other classical machine learning methods. Using SEP event data with associated flare, CME and solar wind speed data we created machine learning models with ridge, K-nearest neighbor and decision tree regressors for predicting the SEP event energy spectra. In this thesis we also introduced feature cost analysis for the different input feature combinations. As the flare, CME and solar wind speed data are obtained from different instruments, we explored whether just some of these properties would be sufficient in predicting the SEP events.

Our results indicate that the ridge and KNN regression models seem to be somewhat equal in prediction performance and overall better than the decision tree regression. The decision tree regressor performed very poorly in predicting the SEP energy spectra. However, the prediction performances across all model types were overall lacking. For the feature set cost analysis, the results show that models trained with flare strength parameters perform better than other models and models trained with only flare strength parameters seem to perform nearly as well as models with additional parameters. Based on these results, further research with more robust models and if possible, larger datasets, is encouraged.

Keywords: solar energetic particles, machine learning

Contents

Introduction

The magnetic activity of the Sun can trigger various energy-releasing events on its surface such as solar flares and coronal mass ejections (CMEs). These bursts of energy can also cause the acceleration and ejection of solar energetic particles (SEPs) into the interplanetary space, where the particles can be observed in so-called SEP events. The particle bursts consist of electrons, protons and to a lesser extent of heavier elements up to zinc [2]. In this thesis our focus is proton flux observations of these SEP events.

The proton fluxes of SEP events can be observed in multiple different energy channels in order to construct energy spectra for the events. The energy of the protons that are usually detected in the interplanetary space can vary from 5 MeV up to several hundred MeV [3] but in principle the lowest energy protons have their energies at the keV levels. The SEP events usually contain a majority of lower energy particles, which means that the events have higher peak fluxes in the lower energy channels than in the high energy channels. However, the shape and the intensities of the energy spectra do vary event by event. It has been shown that the SEP events are closely related to the solar flares and the CMEs. Also, they seem to have some relation to the solar wind conditions at the time of the event, due to it changing the trajectories of the SEPs [2]. These properties of the associated flare and CME and the solar wind conditions have been shown to have an effect on the observed SEP event energy spectra.

Modeling the relationships between the SEP event energy spectra and its associated parameters is not straightforward. During the last decades, empirical models for predicting SEP event energies have been built. These models use historical data of SEP events and associated phenomenon data, such as flare, CME and solar wind properties in order to predict the SEP event energies [1]. During the last years, machine learning (ML) techniques have also been applied in different areas of space

physics, including SEP event observations. One of these studies in the recent years was conducted by Liu et al. in 2024 [1], which used in iterative decision tree algorithm in order to predict the energy spectra based on flare, CME and solar wind properties. However, there haven't been many previous studies on the SEP event prediction based on more classical ML methods. In this thesis our first goal was to explore the technique presented by the Liu et al. research team in their 2024 paper [1]. Our main goal that this thesis addresses to was to expand on this research by employing different ML methods on the SEP energy spectra prediction and also study whether using different associated properties in these ML models would have an effect on the performance of the models. These properties include observational data regarding the flares, the CMEs and the solar wind which are obtained using different instruments on board different spacecraft. Hence if we could get sufficiently good results using just some of these properties in our model, it could also reduce the cost of creating the models, since less observational data would be needed.

In this thesis we will first introduce the basic physics of the solar energetic particles and the theory behind the machine learning models used. Then we will introduce our data set and the machine learning methods we used in predicting the SEP event energy spectra. We have built three different classical machine learning models (ridge regression, K-nearest neighbors regression and decision tree regressor) of which we compare the performances. We also train these models with different subsets of observable properties and compare the performances of the models with these different subsets.

1 Solar Energetic Particles

The Sun is the central object of our solar system. It is an active star and phenomena caused by its activity in the near-earth space are very important in the field of space physics. Among other phenomena, the activity on the Sun's surface can cause

energetic charged particles to be accelerated in bursts into the interplanetary space where they can be observed as solar energetic particle events. In this chapter we will take a brief look into the physics and observable properties of these SEP events. In addition to the physics of this phenomenon, in this chapter we will also take a brief look into the theory of the ML models that can be used in predicting the SEP spectra.

1.1 Solar Energetic Particle events

In this section we introduce the physical theory behind the solar energetic particle events. First we will introduce the main sources of these events: solar flares and coronal mass ejections. Then we will look at the solar particle event observations.

1.1.1 Solar Flares

The solar magnetic activity creates sunspots and active regions on the surface of the Sun. These are regions of strong magnetic fields and are created when the magnetic field lines produced below the Sun's surface emerge through the photosphere. These regions are visible in different wavelength ranges and can be observed via magnetograms. Notably the sunspots are visible in the visible spectrum wavelengths as the surface of the Sun is cooled around them and the spots are seen as darker areas [4].

There is an enormous amount of energy stored in the magnetic fields near the solar surface. This energy can be sometimes released through magnetic reconnection in events known as solar flares. Solar flares have been observed to occur in nearly all regions of the sun with the exception of the very high latitude regions. However, the strongest seems to occur in the active regions and also are more likely to be closer to the equator of the Sun [6].

The energy release in flares is impulsive and can be observationally defined

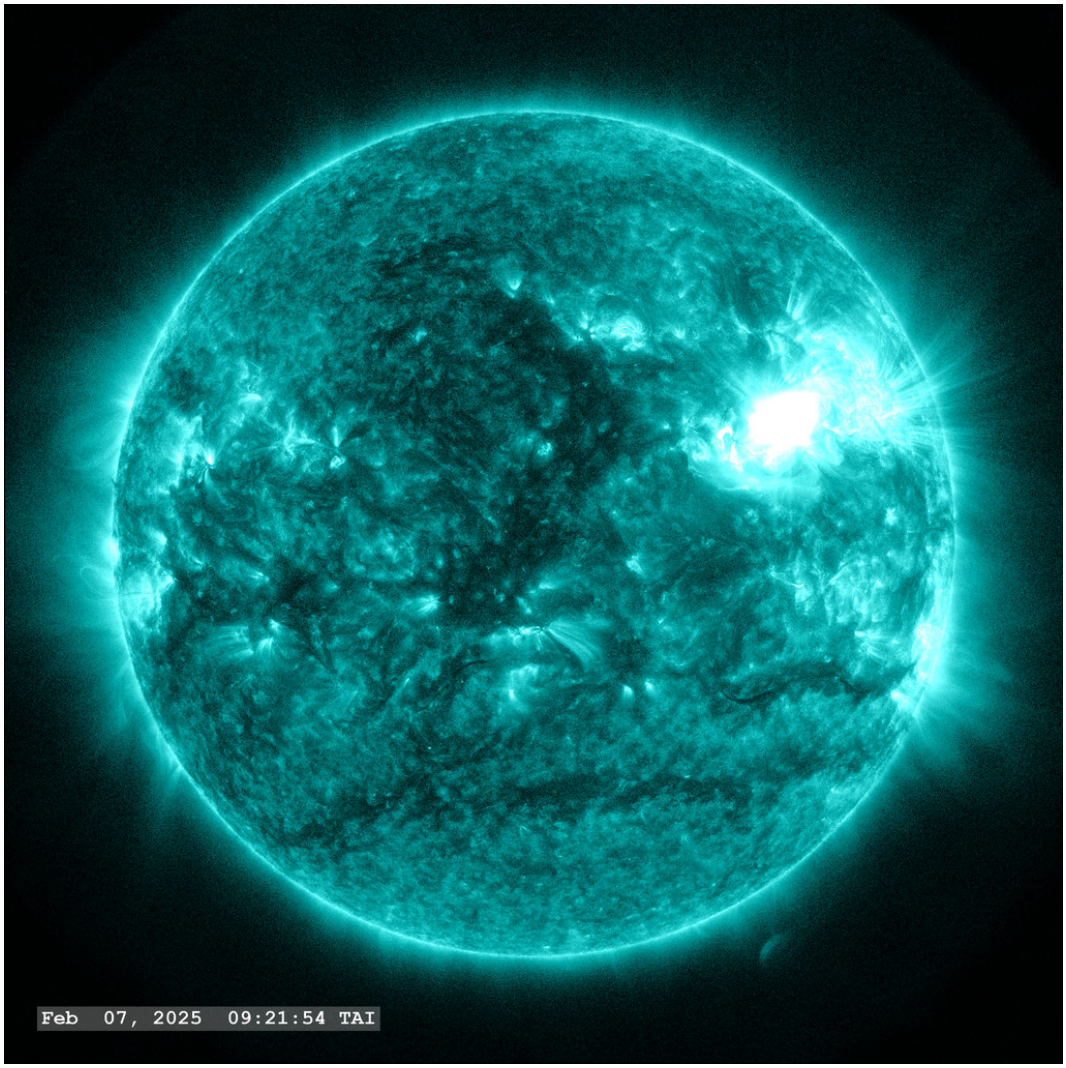

Figure 1. A M7.5 class solar flare imaged in EUV wavelength of 131 Å by Solar Dynamics Observatory on 7.2.2025 [5].

as a brightening of any electromagnetic radiation emission at low, minute-to-hour timescale. One example of such a bright flash can be seen on Fig. 1. During this release of energy a large part of the magnetic energy is transformed into kinetic energy of electrons and ions thus creating accelerated SEPs. However, the exact details of particle acceleration during the impulsive energy release is not yet certain [7]. Flares have been shown to be one of the main sources of SEPs along with coronal mass ejections [2]. Flares at the visible side of the Sun as seen from Earth are nowadays observed by the United States' National Oceanic and Atmospheric Administration's

(NOAA) Geostationary Operational Environmental Satellites (GOES). The GOES spacecraft conduct flare X-ray flux observation in the 0.1 to 0.8 nm passband. The 1-minute averaged peak intensity in this passband is used to classify the flare by intensity. The classification includes a letter representing the order of magnitude of the irradiance and a number within the order. From the lowest irradiance to highest, the letters are A, B, C, M and X. For example the classification X corresponds to a peak irradiance of 10^{-4} W m^{-2}. Thus for example a flare class of M5.8 would correspond to a peak irradiance of 5.8×10^{-5} W m^{-2} [8]. In addition to the intensity, the location of the flare on the solar surface is also important since it affects the probability of observing the accelerated particles in interplanetary space further away from the Sun. This relation of the location and the particle trajectory is gone through more in detail in the next section.

1.1.2 Coronal Mass Ejections

Along with radiation, the Sun can also eject mass into the space in events known as coronal mass ejections. They can be observed by coronagraphs on near-Earth spacecraft. The coronagraphs show the flow of particle matter from the Sun by observing Thomson-scattered sunlight in the solar corona and heliospheric plasma thus giving us a "plane of the sky" view of the emissions. In recent years, most of these observations have been made by the Large Angle and Spectrometric Coronagraph (LASCO) which is onboard the NASA/ESA joint operation Solar and Heliospheric Observatory (SOHO) spacecraft. One exampled of a CME event captured by LASCO C2 and C3 instruments is shown in Fig. 2.

The relation between the flares and CMEs was debated in the 20th century but nowadays they are thought to be different manifestations of a single magnetically-driven event [10]. This is also seen observationally as there is a correlation between the larger CMEs and the larger flares but both of these phenomena can also be

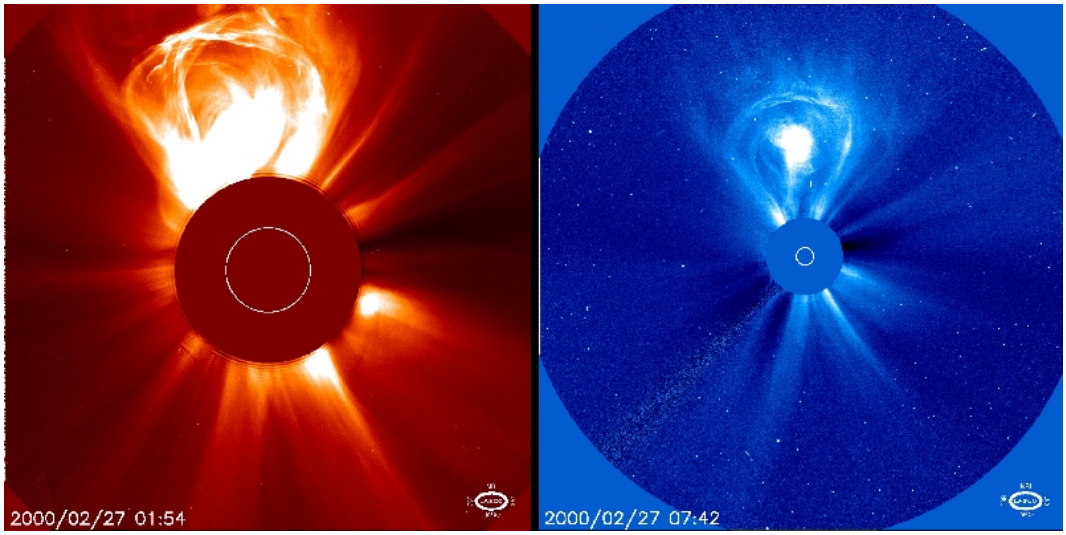

Figure 2. A lightbulb-shaped CME captured by LASCO C2 and C3 instruments on 27.2.2000. The direct view of the Sun is blocked and the size of the Sun's disc is shown as a white circle in the center. The C2 image on the left shows a smaller field of view of the inner solar corona and the C3 view has a larger field of view of 32 solar radii [9].

observed without the other. [2] CMEs generally expel large amounts of plasma and magnetic fields into the heliosphere. The expelled structures are generally much less-localized than the eruptions from flares. Most of the ejected CME mass comes from the lower parts of the solar corona. If the CME has a higher speed than the local Alfvén speed of the corona and the interplanetary medium, it can create a forward shock. These CME-driven shocks can then accelerate electron and ions producing SEPs. The CMEs also can cause disturbances in the interplanetary space leading to geomagnetic storms if they come into contact with the Earth's magnetosphere [10].

From the coronagraph observations one can deduce several properties of a CME that can be used to identify and measure the event. These properties include speed, acceleration, width, mass, kinetic energy and mechanic energy of the CME. However, these coronagraph observations only give us the "plane of the sky" view so to obtain the velocities and geometric values from these two-dimension views one has to make approximations and assumptions so there will always be uncertainty about the true

values. The leading edge of the CME may accelerate or decelerate near the Sun but have been found to have relatively constant velocity beyond a height of two solar radii. The CME speeds have quite a large variance reaching up to over 2500 km s^{-1} with the average speed being somewhere in the range of 300 to 500 km s^{-1} [10]. Measuring the width of the CME is also of importance as it has been shown that the angular width of the CME is important factor for determining whether the CME and its shock are connected to Earth [11]. The observed width depends largely on the direction of the CME as CMEs pointed directly towards at or away from the observer can be seen as halo CMEs where the CME surrounds the disk of the Sun completely. In non-halo CMEs the widths do vary with the average width being around 40° [10].

1.1.3 Solar Energetic Particles

SEP events have historically been classified into two major classes: impulsive and gradual. The impulsive events seem to be associated with flares and solar jets created by magnetic reconnection and not major CME events whereas the gradual events seem to relate to the shocks driven by major CMEs [2]. The difference between the origins of these two types of events is shown in the cartoon Fig. 3. The figure shows how flares induce a more narrow area of particle acceleration whereas CME shocks create a region where the particles are accelerated in a wider area. In addition to the width of the acceleration regions, these two types of events can also differ by many other properties such as elemental abundances, onset timing and duration. An example of gradual SEP event is shown in Fig. 4, where we see two SEP events in a timespan of 12 days, with the latter event being caused by the local crossing of the CME-driven shock at the spacecraft. In short, the impulsive events are small and brief, lasting for some hours, whereas the gradual events are larger, more energetic and intense with durations of hours to days. However, this kind of stark

classification is partly outdated as mixed-type events have been observed. Figuring out the contributions of flare and shock in a single event is still a problem in space physics [2].

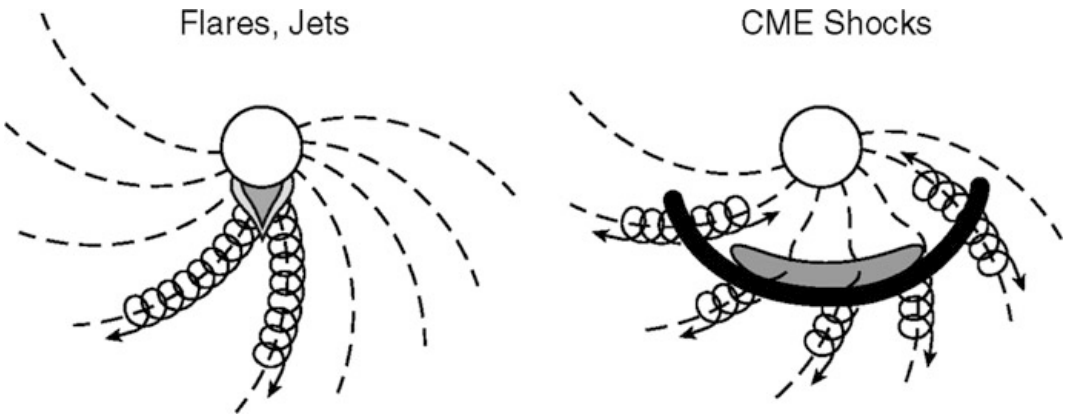

Figure 3. Impulsive, flare-induced source of a SEP event on the left and gradual, CME shock-induced source of a SEP event on the right. On the right-hand picture the CME is coloured as gray, the shock wave as solid black line. The particle trajectories are shown as spirals along the dashed magnetic field lines [2].

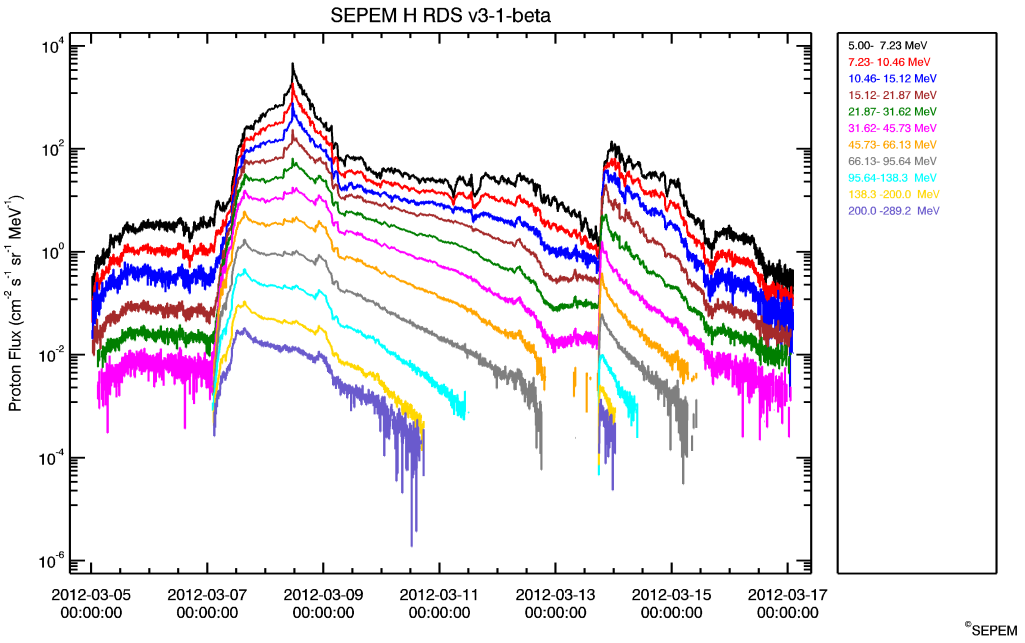

Figure 4. An example of proton flux profiles during a SEPEM reference event at 4.-17.3.2012 in the 11 lowest energy channels. Plot created with SEPEM application [12].

There are a large number of missions that observe SEP events. In this thesis we use data from SEP observations conducted by the GOES satellites. The GOES mission has been in operation since 1976 with new spacecraft launched every few years. The satellites measure proton fluxes in several energy channels from 4 MeV up to >700 MeV. This data can be accessed for example by using the Solar Energetic Particle Environment Modelling (SEPEM) application[1] created by The European Space Agency (ESA) which contains all the GOES proton data in addition to a reference SEP event list. An example plot of proton flux data during a reference event for the 11 lowest proton energies ranging from 5 MeV to 289 MeV channels is shown in Fig. 4 [12]. In our thesis the main interest is in the peak energy spectra of the SEP events. These spectra give us a concise look at the intensity and the profile of the SEP events. An example of a peak energy spectrum is shown in Fig. 5, which shows the observed peak particle flux in the same 11 energy channels as in Fig. 4. The energy spectra is plotted as a peak flux vs. energy graph in logarithmic scale.

When observing SEP events, the connection between the observer and the trajectory of the particles is also important. The trajectories are largely constrained by the curved magnetic field around the Sun, the Parker spiral. The Parker spiral is visualized in Fig. 6 with a reference flare event. This spiral configuration is caused by the solar wind that carries plasma and along it magnetic field outwards from the sun. The solar wind expansion can be approximated as radial but the rotation of the sun draws the field lines connected to a certain point on the solar surface into a spiral [2]. Due to the Sun's rotation from east to west points on the western hemisphere of the Sun are more likely to be connected via the Parker spirals to the Earth. Hence the longitude of the SEP origin (flare or CME) also affects the observation of the associated SEP event. The solar wind speed also varies over time and has an effect on how tightly the spiral structure winds. For an SEP event to be

[1] http://www.sepem.eu/

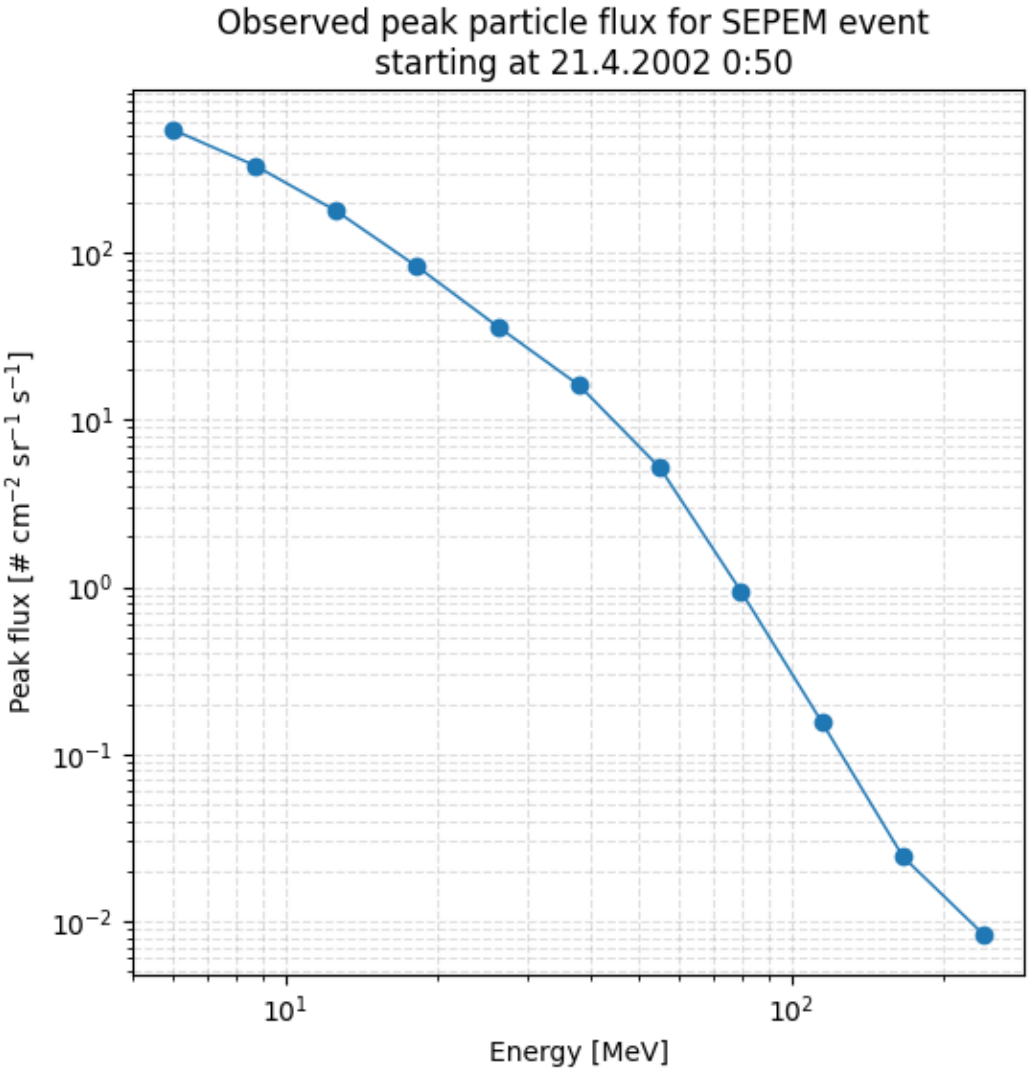

Figure 5. The observed peak particle flux of a SEPEM reference event showing the peak flux in 11 different energy channels in logarithmic scale.

observed in-situ by a spacecraft it has to be connected via the Parker spiral to the SEP triggering event site. As Fig. 6 shows, the Parker spiral field line connecting to the reference event at a certain longitude gets wound into a tighter spiral as the solar wind speed decreases.

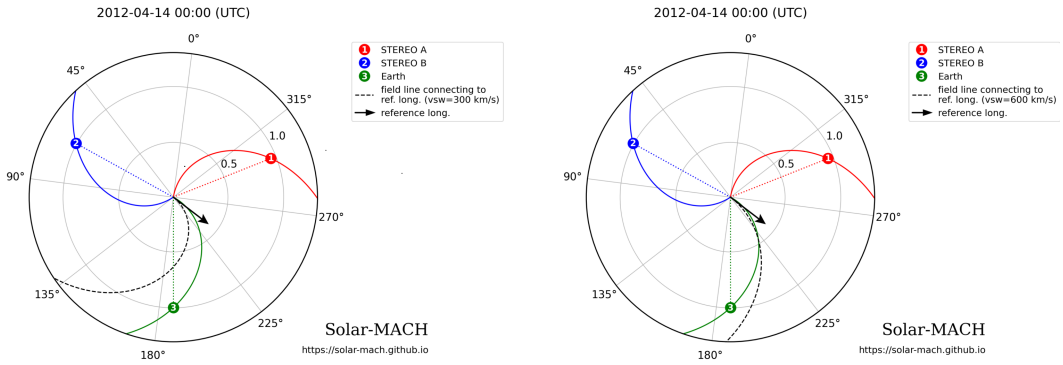

Figure 6. Parker spiral visualization of field lines connecting to Earth, STEREO A and STEREO B satellites along with a reference flare at longitude of 240° and latitude 0° with different solar wind speeds of 300 km s^{-1} (left) and 600 km s^{-1} (right). Figure created with the Solar-MACH tool [13].

2 Machine Learning

Machine learning (ML) is an umbrella term for describing computational pattern recognition methods. Machine learning can be divided into three main categories; supervised learning, unsupervised learning and reinforcement learning depending on how the ML model is trained on [14][2]. In supervised learning the model is given the input vector $\mathbf{x} = (x_1, x_2, ..., x_n)$ for n features with the output vector $\mathbf{y}$ as the response. The model then tries to find the best estimate for the connection between the outputs and inputs [15]. For example, in the case of our study, the input features would be the CME, flare and solar wind parameters and the output features would be the SEP peak fluxes at the different energy ranges.

Unsupervised learning differs from supervised learning in that the model isn't given output features as a response during its training and thus the model is not trying to find a connection between the output and input features. It often deals with finding the underlying connections between the features and/or the observations. One example of this is finding similar groups of observations in a process called

[2]In this thesis we are mainly interested in unsupervised and supervised learning. Basics of reinforcement learning can be found for example in Russell & Norvig's 2016 textbook [14].

clustering [15].

In this thesis the main focus is on supervised learning where an algorithm tries to approximate a function that maps the inputs to the outputs with the smallest error. It assumes that there is a relationship between the output features and the input features which can be shown in general form as

$$\mathbf{y} = f(\mathbf{x}) + \boldsymbol{\epsilon}, \tag{1}$$

where the $\mathbf{y}$ is the output vector, $f(\mathbf{x})$ represents the function mapping the input vector $\mathbf{x}$ to the outputs and $\boldsymbol{\epsilon}$ represents the error term independent of the input vector. Now the model tries to estimate the function that best approximates the relationship between $\mathbf{x}$ and $\mathbf{y}$. This estimation function can be described as:

$$\hat{\mathbf{y}} = \hat{f}(\mathbf{x}), \tag{2}$$

where $\hat{\mathbf{y}}$ is the model's prediction for the output $\mathbf{y}$ and $\hat{f}$ is the model's estimate for the function. Here the error term present in Eq. (1) is averaged to 0 since the error term is defined as having the mean value of 0 and thus is left out of the equation [15].

2.1 Training and Test Sets

The set of input-output pairs of data that is used to train the model to estimate the relationship f between the inputs and the outputs is called the training set. The set of data that is used to evaluate the model's performance after the training phase is called the test set. These sets can be formed from the original data set by a random split of desired ratio for the train-test data [16]. In the model training phase the model is given both the inputs and outputs of only the training set and learns the connection between them. During the training the model assesses how well the estimated function $\hat{f}$ approximates the true relationship f by calculating some training score. After the model has created the estimation function $\hat{f}$, it is

then applied to the yet-unseen test set. The test set input feature values are fed into $\hat{f}$ and the quality of the fit can be then measured by comparing the output estimates $\hat{y}_i$ given by the model to the real test set y_i values [15].

It is important to keep the training and test sets separate in order to get a realistic assessment on how the model performs on unseen data. Also, finding the ratio between the train and test set sizes is a problem in its own, since the model needs to have enough training data to learn the underlying patterns but also enough test data for giving a reasonable evaluation of the model's performance with data points unseen during the training phase [16]. Additionally, it is important to keep in mind the possibility of overfitting the model, where the model creates a very accurate estimate $\hat{f}$ for the training set. Traditionally, it has been thought that overfitted models perform very poorly with unseen data and thus overfitting has been something to avoid [15]. However, recent research about modern machine learning methods has been conducted on 'benign overfitting' that has shown that in some deep neural networks the overfitted models seem to perform remarkably well also on the unseen data [17].

2.2 Model Evaluation

The most common approach for measuring the closeness of the model and the real data in a regression problem is the least squares criterion. For this the model tries to find the appropriate coefficients to minimize the residual sum of squares (RSS). The RSS can be defined as

$$\text{RSS} = \sum_{i=1}^{n} e_i^2 \tag{3}$$

where the e_i-terms represent the ith residual, which is the difference between the ith observed output value and its value as predicted by the linear model. The equation for the residual is

$$e_i = y_i - \hat{y}_i, \tag{4}$$

where y_i is the true value of the ith output and $\hat{y}_i$ is the i:th predicted value of y.

After a model has been trained and used to create predictions on the test data, it is important to quantify how well the model fits the data. There are multiple ways to do this evaluation. For this, two commonly used quantities are the mean square error (MSE) and the R^2 score. The MSE is given by

$$\text{MSE} = \frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{f}(x_i))^2 = \frac{1}{n}\text{RSS}, \tag{5}$$

where n is the number of observations, y_i is the true value of the ith observation and $\hat{f}(x_i)$ is the model's estimate for the ith observation [15]. In this thesis we used the root mean square error (RMSE) as one of the evaluators of the ML models. The RMSE score is measured in the units of the output feature y which may help assess the absolute value of the error but can sometimes confound the assessment of what is a good RMSE value.

Alternatively, one can use a measure that is independent of the scale of the original feature and standardizes the measure. The R^2 score does this by using the proportion of the RSS and the total sum of squares, TSS. The TSS can be defined as the sum of all squared differences between the observations and their mean:

$$\text{TSS} = \sum_{i=1}^{n}(y_i - \bar{y})^2, \tag{6}$$

where the $\bar{y}$ represents the mean value of the observations. The formula for the R^2 score is

$$R^2 = \frac{\text{TSS} - \text{RSS}}{\text{RSS}} = 1 - \frac{\text{RSS}}{\text{TSS}} = 1 - \frac{\sum_{i=1}^{n}(y_i - \hat{y}_i)^2}{\sum_{i=1}^{n}(y_i - \bar{y})^2} \tag{7}$$

[15]. Usually the range of R^2 value is defined as $R^2 \in [0, 1]$ [15]. Interpreting the R^2 in this range is relatively simple. The closer the value is to 1 the better the model is at predicting the outputs. If $R^2 = 0$, the model performs poorly and the prediction result of the model can be interpreted being as good as one would get just by predicting the mean for all of the data points. However, from the eq. (7) we can see that if RSS $>$ TSS, then $R^2 < 0$. If the R^2 value is negative, it can be

interpreted as the model prediction being worse than predicting the mean value for all data points.

2.3 Model Bias and Variance

Both variance and bias are important properties of a ML model relating to how well the model performance generalizes on data outside of the training data [16]. Model variance tells us how stable the model's estimate $\hat{f}$ is related to the data used to train the model. If the variance is high, even a small change in the training data may lead to large changes in the estimate. The more flexible a model is (i.e. how well the model can fit different functional forms), the more variance it usually has. The bias of the model explains the error that is caused by approximating the real-life situation into a mathematical model and thus more flexible models usually result in less bias [15].

There are multiple ways to show how the bias-variance relationship relates to the expected prediction error of the model. We can show that the expected risk of the model can be composed of the bias, which is not dependent on the training sample, the variance which is dependent on the training sample and an error component, which isn't dependent on any model parameters. Without going into the detailed derivation, this can be shown mathematically as

$$\begin{aligned}\mathbb{E}_D\Big[\mathbb{E}_{\mathbf{xy}}[(\mathbf{y}-\hat{f}(\mathbf{x}))^2]\Big] = \mathbb{E}_{\mathbf{x}}\Big[\Big(\mathbf{y}^* - \mathbb{E}_D[\hat{f}(\mathbf{x})]\Big)^2\Big] + \mathbb{E}_{\mathbf{x}}\Big[\mathbb{E}_D[(\hat{f}(\mathbf{x}) - \mathbb{E}_D[\hat{f}(\mathbf{x})])^2]\Big] \\ + \mathbb{E}_{\mathbf{xy}}\Big[(\mathbf{y}-\mathbf{y}^*)^2\Big],\end{aligned} \tag{8}$$

where $\mathbf{y}^* = \mathbb{E}_{\mathbf{y}|\mathbf{x}}[\mathbf{y}]$ is the Bayes-optimal prediction for each point $\mathbf{x}$. The expected value $\mathbb{E}_D$ is taken with regard to all possible training sets, $\mathbb{E}_{\mathbf{x}}$ is taken with regard to the input vector of the new data point, $\mathbb{E}_{\mathbf{y}}$ is taken with regard to the output and the $\mathbb{E}_{\mathbf{xy}}$ is the joint expectation with regard to both the input vector and the output. In this formula the left-hand side term is the expected risk of the trained

model $\hat{f}$, the first term on the right-hand side is the bias of the model, the second term is the variance of the model and the third term is the noise term independent of the model parameters [18].

Now if we wanted to minimize the expected test error (risk), we would need a model that has both as low bias and variance as possible. This introduces the problem known as bias-variance trade-off since it is very easy to train a model with low bias but high variance or vice versa but finding the optimal point where both of these variables are low is very challenging [15].

2.4 Hyperparameters and Cross-Validation

Most ML models have hyperparameters (also known as tuning parameters) that are not determined by the model itself but set separately by the user when training the model. These parameters, such as the λ parameter for ridge regression and K parameter for K-nearest neighbor regression discussed in the following sections usually create some kind of restriction for the model to adjust its complexity in order to minimize the error [16]. Some models also allow using multiple hyperparameters. Choosing the most optimal combination of hyperparameters is not trivial. For this one can use cross-validation, which is one method of finding the best model among different models. An often used technique is the K-fold cross-validation, which is illustrated in Fig. 7. In this technique, the training set is divided into K different folds of equal sizes. The model is then trained with $K-1$ folds as the training set and the remaining set left as the validation set. The prediction error of the fitted model is then calculated with the validation set. This process is iterated until all K folds have acted as a validation set and the results of the K validation sets are averaged. This process is repeated for all different hyperparameter combinations and the best one can then be picked to be used with the test data [16].

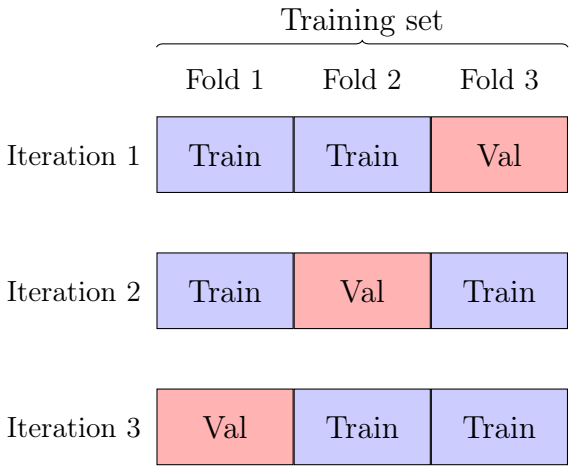

Figure 7. Training and validation folds in a $K = 3$ K-fold cross-validation situation. In each iteration the model is trained on the training folds (blue) and validated on the single validation fold (red).

2.5 Regression Models

Depending on the type of the output data the ML problem can be either classification (categorical output values) or regression (numerical and/or categorical outputs). In our case we are interested in proton peak fluxes in different energy ranges, which can in theory have any positive real number as their value, making our study a regression problem. There are many different ML techniques one can use to tackle a regression problem. For our study we have picked three commonly used techniques that are then compared with each other. As the techniques, we picked ridge regression, K-nearest neighbors (KNN) regression and a decision tree regression model. All of these models are quite simple to implement and also relatively easy to interpret. The decision tree model was also used previously in a similar task predicting proton fluxes from a similar data set as in this thesis by Liu et al in 2024 [1]. Choosing these models also lets us approach the regression problem in different ways since each of these models have a different method of solving the problem.

2.5.1 Linear regression

Ridge regression is a variant of a linear regression model. Linear regression models have a very straightforward approach for predicting the output based on inputs by assuming that there is approximately a linear relationship between them. Mathematically this linear relationship for a model with multiple predictors and a single output can be written as

$$y = \beta_0 + \sum_{j=1}^{n} \beta_j x_j + \epsilon, \tag{9}$$

where y is the output variable, β_0 is the intercept, n is the number of input variables, x_j represents the jth input variable and β_j is a coefficient specifying the association between the output and ϵ is the error term. The error term covers all the possible error sources that are missed by the model, including the non-linearity of the relationship, other variables associated with y and possible measurement errors. The error term is usually assumed to be independent of x [15].

For a situation with multiple outputs each output is thus calculated separately of each other. So in essence the goal of the model is to find the best values for the β_j-coefficients so that the resulting hyperplane of the regression function lies as close as possible to all of the data points used to train the model. The model aims to do this by finding the coefficients that minimize the RSS Eq. (3). For a multiple linear regression model with multiple input features and one output feature, the RSS equation becomes

$$\text{RSS} = \sum_{i=1}^{n} (y_i - \hat{y}_i)^2 = \sum_{i=1}^{n} (y_i - \hat{\beta}_0 - \sum_{j=1}^{p} \hat{\beta}_j x_{ij})^2, \tag{10}$$

where n denotes the size of the data sample, p is the number of different input parameters x and $\hat{\beta}_0, \hat{\beta}_j$ are the estimated coefficients. So in essence the simple linear regression model works by finding the coefficients β_0, β_j that minimize the RSS [15].

2.5.2 Ridge Regression

Ridge regression is a form of linear regression which employs an additional tuning parameter to shrink the regression coefficients towards zero. Instead of trying to minimize only the RSS eq. (10), it introduces a shrinkage penalty term to the function it tries to minimize:

$$\sum_{i=1}^{n}(y_i - \beta_0 - \sum_{j=i}^{p}\beta_j x_{ij})^2 + \lambda\sum_{j=1}^{p}\beta_j^2 = \text{RSS} + \lambda\sum_{j=1}^{p}\beta_j^2 \tag{11}$$

where $\lambda \geq 0$ is a tuning parameter and the sum term $\lambda\sum_{j=1}^{p}\beta_j^2$ is called the shrinkage penalty [15]. The value of the tuning parameter acts as a hyperparameter and can be chosen by cross-validation when training the model as described in Sect. 2.4.

If $\lambda = 0$, the Eq. (11) reduces to the ordinary RSS. When λ is increased, the shrinkage penalty term also increases resulting in the coefficients being restricted and starting to approach zero. As the model produces different coefficient estimates for each λ, finding a good value for it is important when training the models. The advantage of the ridge regression over the simple least squares-method is that by restricting the coefficients the model reduces the variance of the estimate with the cost of increasing the bias. If the data has a linear relationship, bias is often low but variance may be high and a least squares-estimate will result in high variability among the estimates. With ridge regression this variance is reduced [15].

2.5.3 K-Nearest Neighbors Regression

In case of non-linear relationship between the inputs and outputs, a non-parametric method which doesn't have any strong assumption about the form of the function relating the inputs and outputs is often a reasonable approach. The K-nearest neighbors (KNN) regression is one of the simplest and most-used non-parametric methods. In essence it works by identifying a certain number, K, training observations that are closest to a prediction point and then estimates the value of that

point using the average of the closest training responses. This can be expressed as

$$\hat{f}(x_0) = \frac{1}{K} \sum_{x_i \in \mathcal{N}_0} y_i, \tag{12}$$

where the $\hat{f}(x_0)$ is the value of the estimate for input feature values at a certain point x_0, K is the number of nearest neighbors, x_i are the observation points belonging to the set of the closest K observations $\mathcal{N}_0$ and y_i are the values of the output features of the training set [15].

The number of nearest neighbors, K, is a hyperparameter of the model. If a large K is used, the resulting function will be smoother as the points are averaged over larger amount of neighbors. With a lower K value the variability of the function increases and with $K = 1$ the resulting function is a rough step function as only the closest training point is considered. This also results in a complete overfit of the training data when $K = 1$. Generally said a non-parametric approach such as KNN regression is better than a linear regression method, when the relationship between the outputs and inputs is clearly non-linear. However, this is not always true as the KNN regression suffers from the curse of dimensionality, meaning that if there is a large number of different input features compared to the number of observations to be predicted, it might run into a situation where a given observation does not have sufficiently near neighbors leading to an inferior prediction. So generally in a situation with a low number of observations per input feature the non-parametric approaches tend to perform better [15].

2.5.4 Decision Tree Regression

Decision trees can be used for both classification and regression problems. They are relatively simple to interpret but still quite powerful. There are multiple ways to construct a decision tree. Here we focus on one of the most popular methods called Classification and Regression Trees (CART), that was also used with the SEP data in this thesis [16].

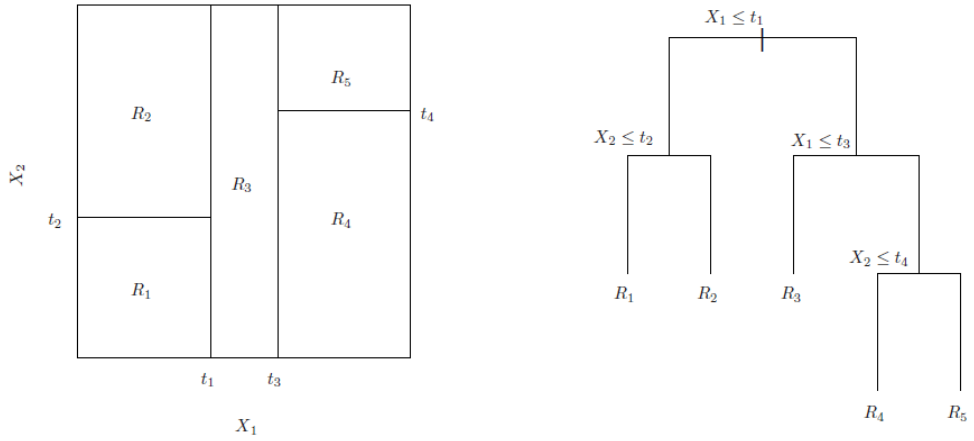

Figure 8. Example of a decision tree constructed from two-dimensional (two input features, X_1 and X_2) data using recursive binary splitting. The figure on the left shows the splitting of the data along split points t_1, t_2, t_3, t_4. The figure on the right side shows the resulting tree plot. Figure edited from [15].

Constructing a decision tree is relatively simple. A simple example of a deicsion tree construction is shown in Fig. 8. First the data is divided into smaller subsets by splitting the input feature (x_i) spaces into separate, non-overlapping regions R_j and then applying the same prediction to all the points inside the same region. The prediction is the mean of all of the output values inside the region. The algorithm tries to minimize the RSS, which is given in the regression tree as

$$\text{RSS} = \sum_{j=1}^{J} \sum_{i \in R_j} (y_i - \hat{y}_{R_j})^2, \tag{13}$$

where the $\hat{y}_{R_j}$ is the mean of the outputs inside the j:th region R_j and J is the total number of the regions. This splitting can be in theory done by any means resulting in irregular areas. However, it is often simplest to use splitting that is recursive and binary, meaning that each split divides the split region into two and each region may be split again, resulting in rectangle or box-shaped regions. During the splitting the split points are decided so that each split results in the largest possible reduction of RSS. The splits can also be in theory continued until each data point is in its own region, but normally some stopping criteria is decided in advance [15]. This stopping

criteria could be for example be a maximum or minimum number of observations per region [15] or a maximum depth of the tree.

After a tree is built there is often a risk of overfitting the data. To mitigate this, a pruning algorithm can be applied to the tree in order to optimize the test error of the model. Simplified, the pruning works by taking a large decision tree and then reducing the lower section splits in order to obtain a less complex subtree. This subtree comes with a better test performance with the trade-off of worse fit on the training data [15]. There are many different pruning algorithms which aim to find the best trade-off between the tree complexity and goodness of the fit, such as cost-complexity pruning [16], which was used in the 2024 article for predicting SEP energy spectra by Liu et al [1].

Decision trees are also a relatively easy to interpret and visualize, at least with a small data set. This is shown in Fig. 8, which visualizes how the splitting of the data happens in the input feature space and what the resulting tree structure looks like. However, they can sometimes be very non-robust and generally the simplest trees do not usually perform predictions as well as some other regression methods. This could be mitigated by aggregating many decision trees, but these methods are not in the scope of this thesis [15].

3 SEP Data

For creating ML models that can predict SEP energy spectra one needs a data set comprising of the energy spectra of as many clearly identified SEP events as possible. In addition to the energy spectra data, data associated with the SEP-generating events, the solar flares and the CMEs, are also essential. In addition, the connectivity of the events is of interest so solar wind speed data is also used in creating the predictions. In this section we will introduce the data sources and the input and output features that were used in training the ML models.

3.1 Data Sources

For the SEP event list we used a modified version of the latest SEPEM reference event list (Sect. 1.1.3), which contains peak proton flux and proton fluence data of verified SEP events from 1976 to 2017 [12]. The SEPEM list defines a reference event beginning when the proton flux in the $7.23 - 10.46$ MeV energy channel reaches 0.01 $\mathrm{cm}^{-2}s^{-1}\mathrm{sr}^{-1}\mathrm{MeV}^{-1}$ and ending when the flux drops below this value. However, if multiple flux enhancements were observed within 24 hours of the previous enhancement, the events were combined as one in the list, as it was observed that multiple enhancements in a short time span could be related to each other [3]. The modified SEPEM event list that was used for this thesis was provided us by the research team of Liu et al. who had previously used an earlier version of their modification of the original SEPEM event list in their 2024 article on predicting SEP energy spectra [1].

The edited version of the list was created by the Liu et al. research team by going through each entry on the SEPEM reference event list and separating the SEP events containing multiple flux enhancements as their own events, ending up with an initial list of 256 SEP events in the time range from 30.4.1976 to 6.9.2017. However, the earliest entries in the list were not used for the study, since the availability of CME feature data from LASCO restricts the time range of usable entries to start from the year 1996 [19], setting the SEP event on 4.11.1997 as the first entry on our list.

In addition to separating the entries containing multiple events, the Liu et al. research team had also identified an associated solar flare with each event and provided the flare features in their event list. The flare data was compiled from multiple sources with the primary source being the National Oceanic and Atmospheric Administration (NOAA) flare-CME-SEP event list[3], which contains 184 flare-matched SEP events. The events not found on the NOAA SEP event list were identified

[3]https://umbra.nascom.nasa.gov/SEP/

with flares one by one by using the following criteria: First, the flare must occur at maximum 10 hours before the SEP onset time in the 7.23 - 10.46 MeV channel. Second, if there are multiple flares occurring during this time window the largest flare is prioritized. And last, if multiple flares meet these two criteria, the ones located on the Sun's western side are prioritized. In addition to the NOAA event list, SEP catalogue by Papaionnau et al. [20] and the integrated Geostationary Solar Energetic Particle Events Catalog (GSEP) by Rotti et al. [21] were used to find the corresponding flares. If the flare identification using these resources failed, the SEP event was excluded. The associated CME featured data was obtained from the LASCO CME observations by the Liu et al. research team [1], based on the online CME catalog of the Coordinated Data Analysis Web[4].

In addition to the SEP, CME and flare data, the solar wind speed data was considered to be of importance by Liu et al. in predicting the SEP energy spectra [1]. The solar wind speed affects the Parker spiral configuration and thus the magnetic connection of the observing spacecraft to the Sun as described in Sect. 1.1.3. The solar wind speed data used were obtained from the NASA Wind spacecraft observations. Solar wind speed values at the flare start time for each event were obtained by us using the Solar-MACH Python interface [13].

3.2 Features

For our input feature selection, we decided on initially using the same set of seven features that were used by Liu et al. in their 2024 paper [1]. Smaller subsets of input features were split from these seven initial features. The input features associated with the flare were the latitude (Lat_{flare}) and longitude (Lon_{flare}) and the strength of the flare, which was measured with the peak and integral values of the soft X-ray flux ($F_{SXR_{peak}}$, $F_{SXR_{int}}$). The X-ray flux values were converted to logarithmic

[4]https://cdaw.gsfc.nasa.gov/CME_list/

scale before the analysis. The CME-associated features were the width of the CME (W_{CME}) as observed in degrees and the velocity of the CME (V_{CME}) in km s^{-1}. For the solar wind, the solar wind velocity (V_{SW}) in km s^{-1} at the flare onset was used.

Table I. Energy ranges of the six lowest energy channels in the SEPEM reference data set used in the analysis.

Peak number	Energy Range [MeV]
1	$5.00 - 7.23$
2	$7.23 - 10.46$
3	$10.46 - 15.12$
4	$15.12 - 21.87$
5	$21.87 - 31.62$
6	$31.62 - 45.73$

For the output features, the SEPEM reference proton data set includes data for fourteen different energy ranges ranging from 5 MeV to nearly 900 MeV [22]. In their paper, the Liu et al. research team omitted the three highest energy range channels from their analysis since the SEP events only rarely reach that high energies [1]. For our output features we initially worked with the same set of 11 peak proton flux energy ranges with the lowest range being $5.00 - 7.23$ MeV and the highest being $200.0 - 289.2$. However, after initial testing we opted to drop the 5 highest energy ranges since a large part of the events did not reach higher energies. The energy ranges of the different peak values used in our analysis are shown in Tab. I. The peak energy values were also converted to logarithmic scale before the analysis.

Sample values of our data set for the input features and output features are shown in Tabs. II and III, respectively.

Table II. Sample of our data set showing typical values of the input features. The flare strength values have been converted to logarithmic scale.

event id	V_{CME} [km s^{-1}]	W_{CME}[°]	Lon_{flare}[°]	Lat_{flare}[°]	$\log F_{SXR_{peak}}$	$\log F_{SXR_{int}}$	V_{SW} [km s^{-1}]
85	1192.0	197.0	89	12	0.000008	0.025008	474.00653
86	1328.0	360.0	-42	4	0.000048	0.060507	613.19070
88	459.0	192.0	49	12	0.000011	0.002951	418.80225

Table III. Sample of our data set showing typical values of the output features. The values have been converted into logarithmic scale.

event id	logPeak 1	logPeak 2	logPeak 3	logPeak 4	logPeak 5	logPeak 6
85	2.266600	1.238126	0.771123	0.265464	0.101225	0.031870
86	6.308496	4.996169	3.808877	2.332386	0.877850	0.161129
88	2.598996	2.040375	1.861524	1.204355	0.719405	0.357648

4 Predicting SEP Energy Spectra with Machine Learning

In this section we present the details of our ML models used to predict the SEP peak energy spectra. The goal of our analysis is to find out whether there is difference on the performance of the different machine learning approaches (ridge regression, KNN regression and decision tree regression) and how the different subsets of the input features compare to each other in predictive power. First we will introduce the data set and how it was preprocessed before the analysis. Then we will explore our different machine learning models and their respective parameters. We will also talk about how we picked our subsets of different feature combinations and how they were used in the predictions.

The first goal of our work was to reproduce the results of the iterative decision

tree regression model used by Liu et al. in their 2024 paper [1] After some tests with their algorithm, it was observed that our models couldn't produce as good results as their paper had indicated. After our trials on the iterative decision tree regression model, we proceeded to branch out to other more traditional machine learning methods and also incorporate the study of the effect of different input feature combinations on their performance.

4.1 Preprocessing

As described before in Sect. 3.2, after initial testing we opted to drop out the five highest peak energy ranges from the data provided by the Liu et al. research team and focus only on the lowest 6 energy ranges as our output features. This decision was motivated mostly by the same reasons that lead the Liu et al. research team to drop the highest 3 energy ranges out of the initial 14 channels provided in the SEPEM reference data set [1]. Due to the nature of the SEP events, the observations become increasingly rare at the highest energy channels. This means that in a typical SEP event there is often a cut-off at certain peak energy channel after which all intensity values at higher energies become 0. This high energy cut-off behavior of the data leads to a skewness towards values of 0 at the higher energy channels, which may have an effect on the prediction accuracy. This behavior in the data also means that there is some dependence between the higher energy channels, since no events had non-zero-valued observations in the higher energy channels if a value of 0 was observed in a lower energy channel. Our ML models work by treating the output features as completely independent, so the dependency between the higher energy channels isn't accounted for and may affect the performance. By exploring the data, we observed that the cut-off energy above which the event was not observed anymore happened often after the sixth energy channel which motivated us to work only on the six lowest energy channels.

Data scaling was also applied to both the input and output feature data. Scaling the data is essential for the ridge regression and the KNN regression models as their performances are tied to the scale of the data. In ridge regression the function (eq. 11) is minimized by finding the best estimates for the coefficients β_j and the values of the output features are then multiplied by their respective estimates for the coefficients. Hence the value of this multiplication, $\hat{\beta}x_i$ depends on both the scale of the ith input value x_i and the hyperparameter λ [15]. By transforming the feature values to the same scale, we eliminate this effect. For the KNN regression the distance between the different observation points is essential in determining the value of the prediction. This distance is also affected by the scale of the different input features and having features of different scales would distort the calculation of the predictions [15]. The decision tree model is not affected by the scale of the data and could have handled predicting the values using non-scaled data, but for the sake of consistency, the decision tree model was also used with scaled data. All of the input and output data were scaled to have a mean value of 0 and unit variance.

In addition to picking only the lowest energy channels for our analysis, we also followed the Liu et al. paper in converting the values of all of the output features and for the input features, the flare strength values ($F_{SXR_{peak}}$, $F_{SXR_{int}}$) to logarithmic scale [1]. The output features are the SEP peak intensities, which are usually treated on logarithmic scales. During the data exploration, 14 events with missing data on the CME widths and velocities and the flare strength values were excluded, leaving us with a total of 142 SEP events to use in our analysis.

4.2 Feature Selection

For our analysis, we created seven subsets of input feature that were used in training the ML models along with the set containing all input features. The input features contained in each set are presented in Tab. IV. The subsets were created with the

sources of the different input feature data in mind. The flare strength data $F_{SXR_{peak}}$ and $F_{SXR_{int}}$ and the flare location data Lon_{flare}, Lat_{flare} are usually obtained from single spacecraft observations. As with the CME features, V_{CME} and W_{CME} are also extracted from a observation by a single instrument. Additionally, the solar wind speed V_{SW} is usually observed by a spacecraft that does not conduct flare observations. In our data set, the flare measurements are obtained from GOES observations, which does not measure CME or solar wind parameters. The CME measurements are obtained from LASCO onboard the SOHO spacecraft, which does also have a solar wind measuring instrument also onboard [23]. However, for this data set we used solar wind speed data obtained from the WIND spacecraft.

Table IV. Input features included in each of the input feature sets.

Set	Input Features
1	V_{SW}
2	V_{CME}
3	Lon_{flare}, Lat_{flare}
4	$F_{SXR_{peak}}$, $F_{SXR_{int}}$
5	$F_{SXR_{peak}}$, $F_{SXR_{int}}$, V_{CME}
6	Lon_{flare}, Lat_{flare}, $F_{SXR_{peak}}$, $F_{SXR_{int}}$
7	$F_{SXR_{peak}}$, $F_{SXR_{int}}$, Lon_{flare}, V_{CME}
8	V_{SW}, W_{CME}, Lon_{flare}, Lat_{flare}, $F_{SXR_{peak}}$, $F_{SXR_{int}}$, V_{SW}

After the data was preprocessed as described in Sect. 4.1, a prediction model was set up for each of the machine learning models. For each machine learning model, a set of hyperparameters was also defined. For ridge regression, we had a single hyperparameter, λ, for which we had 16 different values ranging from $1 \cdot 10^{-15}$ to 100. For the KNN regression we also used a single hyperparameter K as the number of neighbors, where the possible values for K were $\{1, 3, 5, 9, 12, 15\}$. For the

decision tree hyperparameters we picked the max depth of the model with possible values of $\{5, 10, 15\}$, minimum samples per leaf $\{1, 5, 10\}$ and the minimum samples per split $\{2, 5, 10\}$. Due to the amount of models trained we opted to increase the efficiency of our process by not using the cost-complexity pruning with the decision tree algorithm, which was used in the Liu et al. 2024 paper [1]. During the training, for each model a grid search algorithm using 5-fold cross-validation was applied to find out the optimal hyperparameter combination for each model/feature set combination. The grid search algorithm works by training each model with each possible combination of hyperparameters and then uses the cross-validation technique to pick out the best-performing hyperparameter combination to train the final model.

After the machine learning models and the input feature sets were established, each model type was trained with all of the 8 different input feature sets and their performances were evaluated by calculating the training and testing set R^2 and root mean square error values. The output for all of these models are the peak intensity values for each event for the 6 first energy channels. Fig. 9 shows us one example event taken from one round of model training, with the true observed values and the predictions of the three different models. In the appendix, figures 12 and 13 show the observed and predicted spectra for each event for one randomly picked test set. For this example plot, the models were trained with the feature set containing $F_{SXR_{peak}}$, $F_{SXR_{int}}$ and V_{CME} as the input features. The event shown in Fig. 9 here is the same as depicted in Fig. 5.

During our initial testing, it was observed that all of our models presented very high variance when they were trained with different splits of training and testing data. To mitigate this, we trained each model over 100 different train-test data splits and calculated the average performance scores for each model/feature set combination. The train-test splits were done with a ratio of 0.2 for the test set size

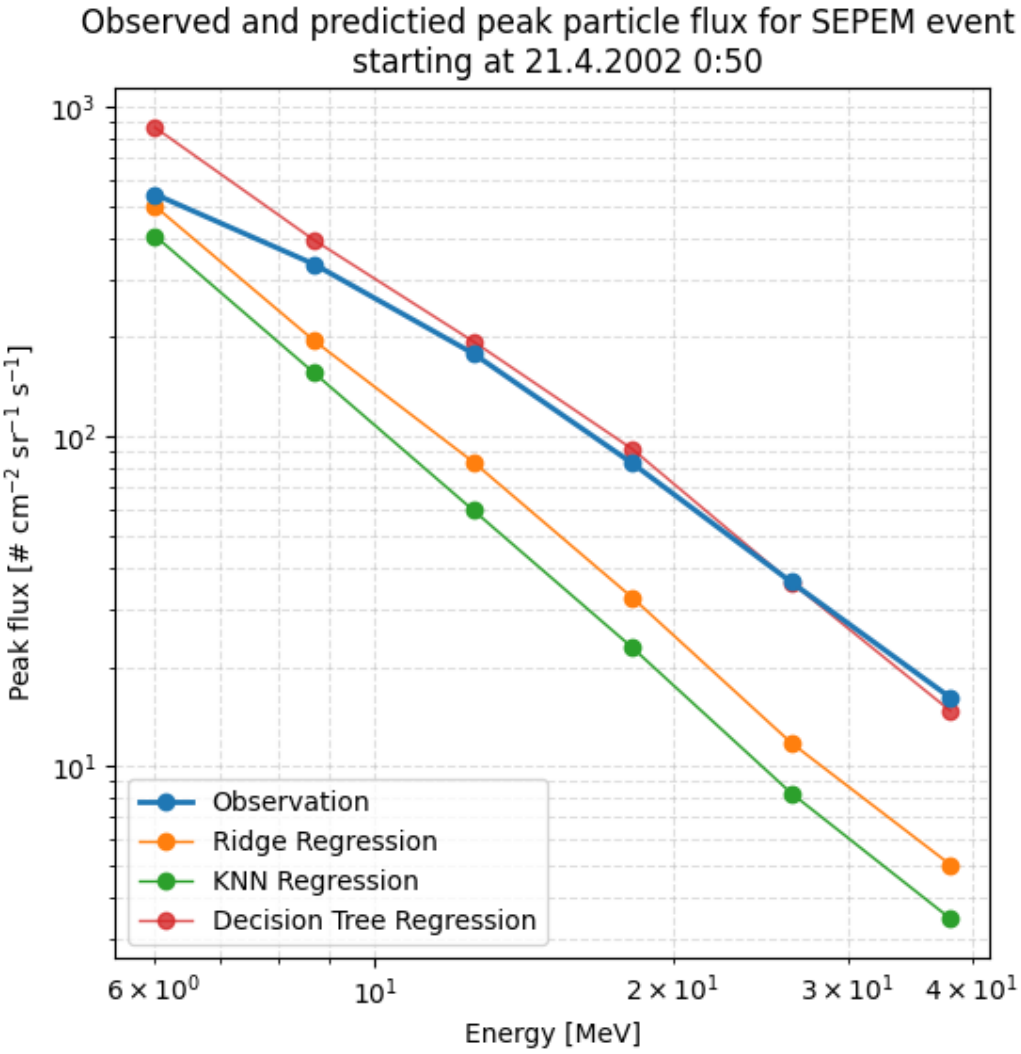

Figure 9. True observed values and the predictions created by the three different ML model types of an example SEP event peak particle flux. The models were trained with $F_{SXR_{peak}}$, $F_{SXR_{int}}$ and V_{CME} as the input features.

and 0.8 for the training set size. The averages were calculated in two ways. First the average peakwise R^2 and RMSE scores were calculated by taking the predictions of all of the peaks per predicted event and calculating the evaluation scores against the true values of the corresponding event. This gives us the *peakwise* evaluation scores, which were then averaged over each peak giving us a single value for test and train R^2 and RMSE per model/feature set combination. Additionally, we calculated a *global* R^2 and RMSE for each model/feature set combination by first pooling each

of the predicted values into one single vector, which was evaluated against similarly pooled vector of observed values. The pooled vectors were created by taking all of the values for each event for each peak and appending the peakwise values to the vector one after another, with the sixth peak values appended last. The R^2 values of these methods for the test set were used for evaluating the prediction performance on unseen data for the model/feature set combinations. The RMSE values could also be used for evaluation, but due to their dependence on the scale of the output features, we opted to use the R^2 scores since it is easier to evaluate the relative performances of different models with R^2 scores.

5 Results

The global averaged R^2 scores for the test set calculated over 100 train-test splits for each model and input feature set combination are shown in Fig. 10 and the peakwise averaged R^2 test scores for these combinations over 100 splits are shown in Fig. 11.

The resulting test set R^2 averages for the global scores are overall higher than the peakwise scores, with the the global R^2 score means ranging from about 0.35 to 0.50 for the ridge regression and KNN regression and from about 0.2 to 0.4 for the decision tree regression. For the peakwise test set R^2 averages we also observe negative values of R^2 for some model/input feature set combinations. For the peakwise scores, the test set R^2 score means range from about -0.1 to 0.2 for the ridge regression, about -0.2 to 0.2 for the KNN regression and about -0.3 to 0.0 for the decision tree regression.

Regardless of the averaging over 100 train-test splits, there is still quite a significant amount of variance in both the global and peakwise R^2 scores, which can be visualized in the figs. 10 and 11 as the error bars showing the standard deviation for each score. In all cases the standard deviation error bars for each model and

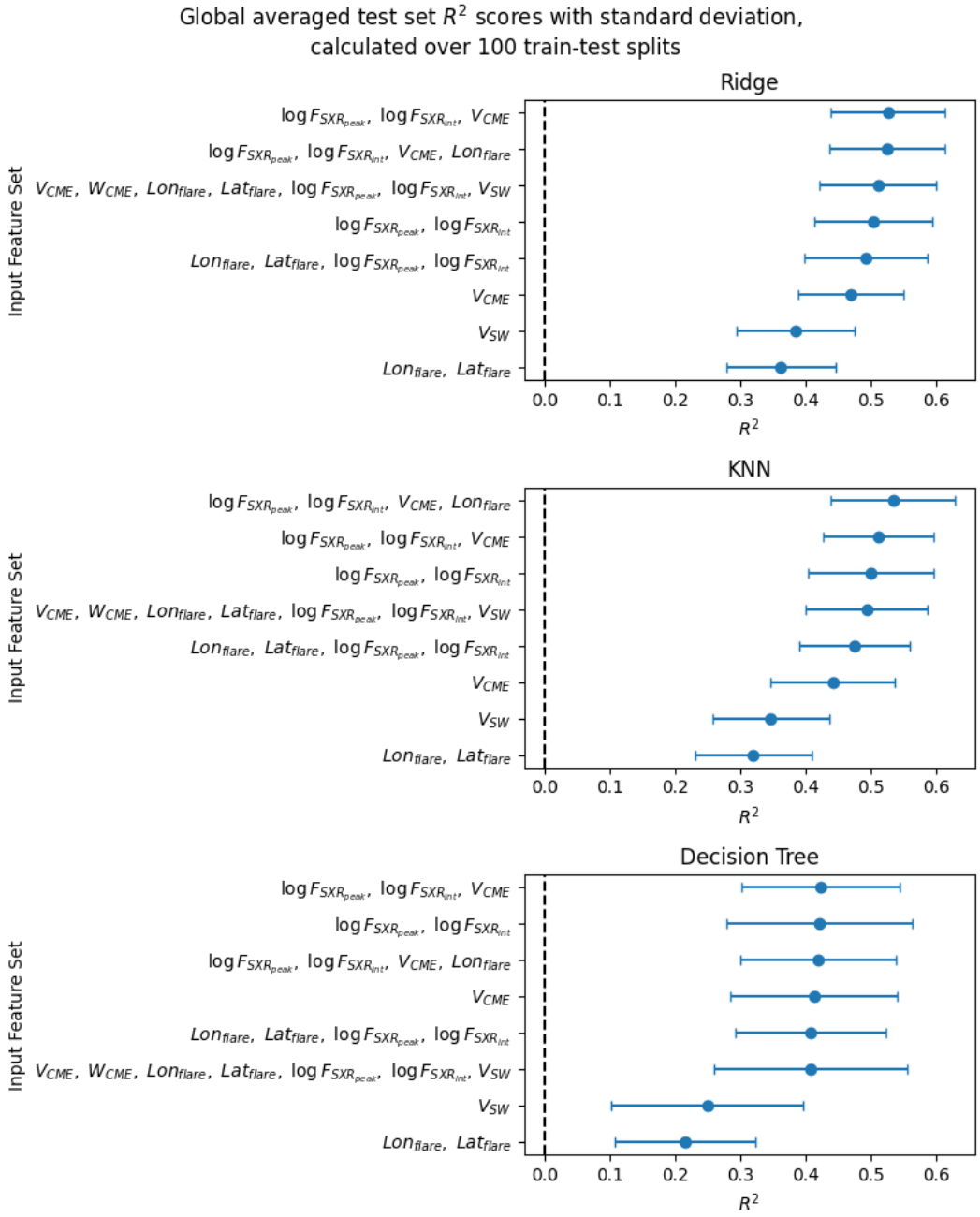

Figure 10. Global R^2 test score averages for each model/input feature set combination, calculated over 100 train-test splits.

feature set combination present overlapping. Especially in the case of the peakwise averaged R^2 scores, the decision tree models exhibit relatively even larger error bars than the other models do.

For ranking the different input feature subsets, we observe similar behavior for

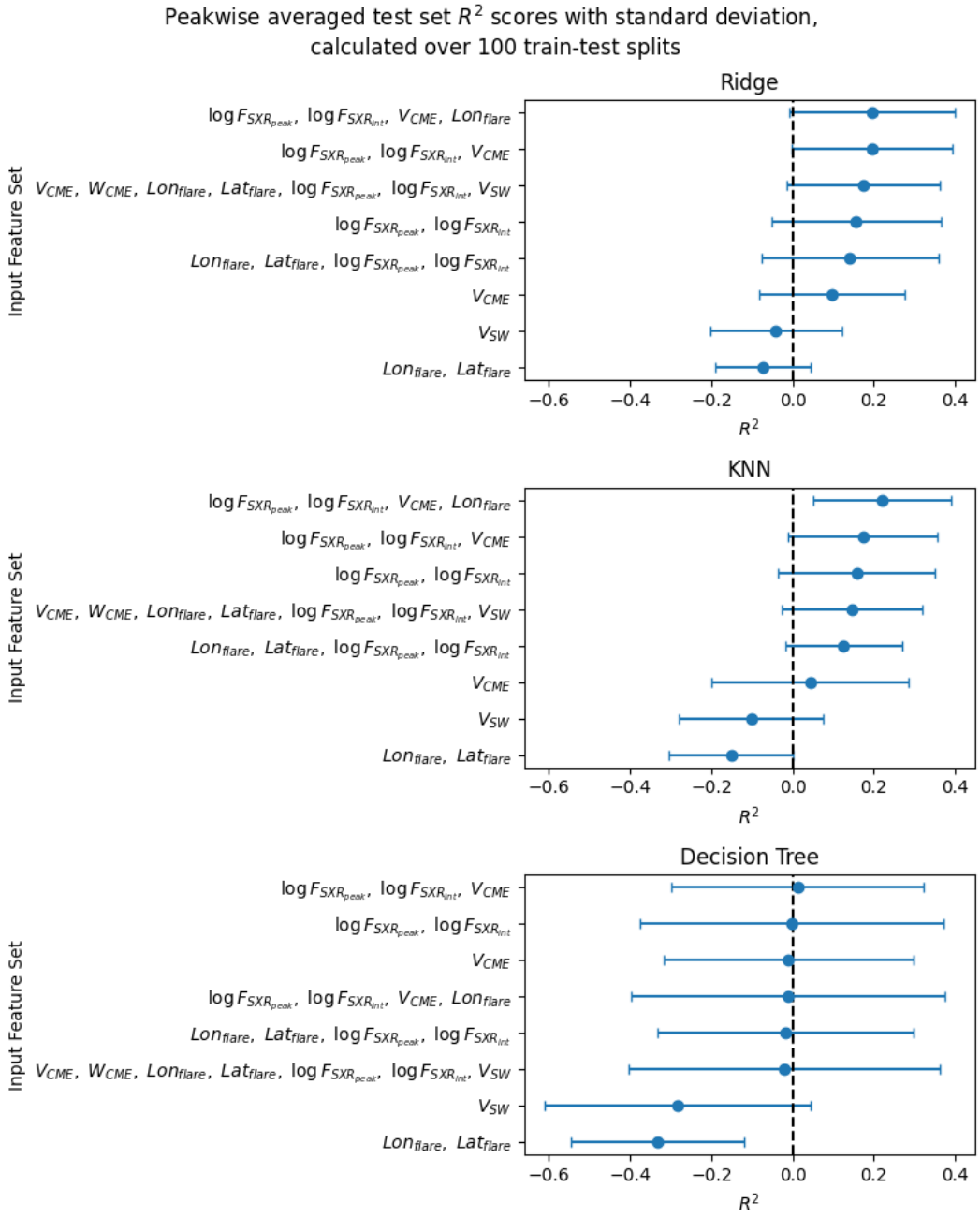

Figure 11. Peakwise R^2 test score averages for each model/input feature set combination, calculated over 100 train-test splits.

ridge regression and KNN regression for both the global and peakwise cases. In these rankings the bottom three subsets are $\{V_{SW}\}$, $\{V_{CME}\}$ and $\{Lon_{flare}, Lat_{flare}\}$ in this order from the third-worst to the worst model. The other five input feature subsets for these two models present similar scores among themselves. For the ridge

regression and KNN regression, in all cases the top two best performing input feature subsets are $\{F_{SXR_{peak}}, F_{SXR_{int}}, V_{CME}, Lon_{flare}\}$ and $\{F_{SXR_{peak}}, F_{SXR_{int}}, V_{CME}\}$. There is some variation of the rankings of the three other feature combinations for the ridge regression and KNN regression, but the values of the R^2 means of them are very close to each other in all cases.

For the decision tree regression, we observe a sort of division in two in the rankings of the input features. For both the global and peakwise test set R^2 mean scores, the worst-performing feature sets are $\{V_{SW}\}$ and $\{Lon_{flare}, Lat_{flare}\}$ with clearly a worse score than the other five input feature sets. However, the remaining five input feature set all produce very similar test set R^2 scores for both the peakwise and the global averages for the decision tree regressor. We also observe that for the peakwise averages, the five best-performing feature sets for the decision tree regressor have their mean test set R^2 scores very close to 0, with the two worse-performing models having their mean test set R^2 scores around -0.3.

6 Discussion

Based on the test set R^2 scores, the ridge regression and KNN regression models appear to be performing in a similar manner when their performances are compared over the different input feature combinations. For the decision tree regression, the scores are overall worse. The ranking of the input feature subsets are also very similar for the ridge regression and KNN regression models, with the same subsets being the top five for all these cases. These slight ranking differences could still be caused by the high variance in these models, since we can see that the standard deviations of the scores are still quite high when averaged over 100 train-test splits.

For the decision tree regression models the scores are overall worse and especially for the peakwise R^2 we observe the scores being of mean 0 and even worse for the input feature subsets $\{V_{SW}\}$ and $\{Lon_{flare}, Lat_{flare}\}$. This indicates that the

decision tree regression models perform very poorly in their predictions. Having a R^2 score of zero means that one would get the same result just by setting the predictions as the mean values of each energy channel for each observation. Having a negative R^2 score means that the prediction would be worse than predicting the mean for all observations, which indicates a very poorly performing model.

We have calculated the R^2 scores in two ways: the global R^2 score for the pooled predictions and the average peakwise R^2 score for the peak-by-peak predictions. As the R^2 score of zero can be compared to just using the mean value of the output feature for the predictions, the R^2 could be also interpreted as the mean square error relative to just using the mean value for prediction. For the global R^2 scores this mean value would be the mean of all energy values across all peaks and observations, which ignores the observed behavior of the energies in our data (higher energy channels tend to have lower peak fluxes). Therefore this global mean R^2 would be a worse predictor than the peakwise mean R^2. Thus when considering our test set R^2 scores, it is reasonable that the global test set R^2 scores are overall higher than the peakwise test set R^2 scores. However, when considering the real physics behind the SEP events, using the peakwise R^2 scores in evaluating the model performances would be more reasonable.

For the different input feature subsets, it can be observed that only using the solar wind speed (V_{SW}) or only the flare coordinates (Lon_{flare}, Lat_{flare}) as the input features, the models score overall worse than the other combinations. For the ridge and KNN regression models the feature set containing only the CME velocity (V_{CME}) also ranks among the three worst-performing models. For the other five input feature subsets, the scores are quite close to each other and the order of the subsets among the rankings vary only little. One notable element for these best-performing input feature subsets is that all of them contain the flare strength features ($\log F_{SXR_{peak}}$, $\log F_{SXR_{int}}$) and zero to five other input features. However,

the scores for these models are very close to each other, so it would seem like having just the flare strength properties as the input features would already be a reasonable basis for a ML model, as having additional input features does not seem to have a significant effect on the score. In this analysis, we decided to use the set of all input features and seven different subsets of these features in our comparison. However, for a set of seven different input features, one could create 126^5 smaller subsets of input features for the comparison. Although computationally more expensive, with more optimized code this would not be an impossible task. However, one needs to take in account the real-life source of observations when considering these subsets.

One of our main issues in predicting the SEP energy spectra was the small size of the data set, from which we included only 142 events in our analysis. We observed that our models had a very high variance regarding the randomly created train-test set splits, which is most likely caused by the size of the data set, as we only had 29 observations in our test sets. To mitigate this variance, we trained the models over 100 train-test splits and calculated the averages for evaluation scores. However, after this we still observed that these mean scores presented relatively high standard deviations from the means. One could always take a higher amount of these train-test splits to average over, but due to the small size of the data set, it would most likely produce diminishing results at the cost of computing power. One could also optimize the train-test split ratio, since our data set used the rather standard ratio of 20 % test set size to 80 % train set size.

For the decision tree regression, we observed very poor results for all of our input feature subsets. For this analysis we opted to use a simple decision tree regression model without any pruning algorithms. However, the tree construction was still regulated via hyperparameters. With a pruning algorithm implemented, the decision tree regression scores could possible be improved. However, in this kind

[5]The size of power set of 7 is 128, from which the empty set and the set of all features are subtracted.

of situation where we train multiple models over a large amount of train-test splits, introducing a pruning algorithm to the decision tree regressor greatly increases the computational power needed to train the models.

We also opted to overall very simple machine learning models for our analysis, since it is a good starting point when considering further research into the matter. With a combination of ridge regression, KNN regression and decision tree regression we do introduce linear and non-linear approaches to our regression problem. For this analysis, we kept the number of hyperparameters per model relatively low for the sake of simplicity, but with a larger array of hyperparameters, one could possibly find more suitable models, albeit again with the cost of requiring increased computational power.

We detected that the higher energy channels show dependencies on the other channels, since if a certain channel had the peak energy flux of zero in an event, all the higher energy channels above that would also have zero peak energy fluxes. This is not accounted for in any of our machine learning models, which treat all the predictions for the different energy channels as independent and the predictions are made separately for each channel. In our analysis, we tried to mitigate this by limiting the output features to the six lowest energy channels, where this kind of behavior is not as prominent. However, for future endeavors in this kind of task, one should consider using more robust and complex machine learning models that are capable of handling these kind of dependencies in the data. One possible avenue for future research could be using support vector machines for predicting the SEP peak fluxes.

Considering the overall evaluation scores of all of the models, we still see quite a large variance among the different subsets. The R^2 mean scores are still overall in the lower side, under 0.4 in all peakwise averaged test set R^2 scores, which would indicate that the prediction performance of the models is very modest at best. Based on these

results, it is difficult to give any definite answers on whether there are any optimal input feature subsets. However, based on our ranking of the different input feature subsets, it would seem like the flare strength features would be a good candidate for the most cost-effective features to use in training the models, since they are observed by the same spacecraft and seem to perform by themselves nearly as well as larger input feature subsets. However, one would need to construct better models in order to definitively find out whether any particular input feature combination is more cost-effective than others at predicting the SEP peak energy fluxes.

Use of AI in thesis

For this thesis, ChatGPT[6] was used extensively for creating the necessary code for implementing the machine learning methods and data processing. ChatGPT was used from September 2025 to May 2026 with the most recent free version available at the time of use. The versions used during this project were GPT-5, GPT-5.1, GPT-5 Mini, GPT-5.2, GPT-5.3, GPT-4.3 Mini and GPT-5.5. ChatGPT was mostly used to debug the code and help with the proper methods of handling the data in the correct formats. It was also used to help generate Fig. 7 using the TikZ package for Latex and with Latex formatting during the writing. ChatGPT was also used during the coding part to generate the initial model hyperparameter lists as advised by a thesis supervisor. AI was not used to direct the actual process or workflow of the model training neither it was not consulted to interpret or draw conclusion from the results.

Additionally, DeepL Translate[7] was used in the writing process from March 2026 to June 2026 in translating Finnish words and expressions into English.

[6]https://chatgpt.com/

[7]https://www.deepl.com/en/translator

References

[1] J. Liu, Z. Huang, J. Guo, Y. Wang and J. Liu, The Astrophysical Journal Letters **975**, L43 (2024) doi:10.3847/2041-8213/ad8bbc.

[2] D. V. Reames, *Solar Energetic Particles: A Modern Primer on Understanding Sources, Acceleration and Propagation*, Vol. 978 of *Lecture Notes in Physics* (Springer International Publishing, Cham, Switzerland, 2021) doi:10.1007/978-3-030-66402-2.

[3] P. T. Jiggens, S. B. Gabriel, D. Heynderickx, N. Crosby, A. Glover and A. Hilgers, in book *2011 12th European Conference on Radiation and Its Effects on Components and Systems* (IEEE, Sevilla, Spain, 2011), pp. 549–564 doi:10.1109/RADECS.2011.6131436.

[4] M. J. Aschwanden, *Physics of the solar corona: an introduction with problems and solutions*, *Springer-Praxis books in astronomy and space sciences*, 2nd ed. ed. (Springer, Berlin, 2006).

[5] S. Wiessinger and T. Bridgman, NASA Scientific Visualization Studio | M7.5 flare from Active Region 13981 - February 7, 2025, 2025, https://svs.gsfc.nasa.gov/5501/.

[6] W. Abdel-Sattar, R. Mawad and X. Moussas, Advances in Space Research **62**, 2701 (2018) doi:10.1016/j.asr.2018.07.024.

[7] A. O. Benz, Living Reviews in Solar Physics **14**, 2 (2017) doi:10.1007/s41116-016-0004-3.

[8] T. N. Woods, T. Eden, F. G. Eparvier, A. R. Jones, D. L. Woodraska, P. C. Chamberlin and J. L. Machol, Journal of Geophysical Research: Space Physics **129**, (2024) doi:10.1029/2024JA032925.

[9] SOHO-Gallery: Best Of SOHO, https://soho.nascom.nasa.gov/gallery/images/las02.html.

[10] D. F. Webb and T. A. Howard, Living Reviews in Solar Physics **9**, (2012) doi:10.12942/lrsp-2012-3.

[11] X. H. Zhao, X. S. Feng, H. Q. Feng and Z. Li, The Astrophysical Journal **849**, 79 (2017) doi:10.3847/1538-4357/aa8e49.

[12] N. Crosby, D. Heynderickx, P. Jiggens, A. Aran, B. Sanahuja, P. Truscott, F. Lei, C. Jacobs, S. Poedts, S. Gabriel, I. Sandberg, A. Glover and A. Hilgers, Space Weather **13**, 406 (2015) doi:10.1002/2013SW001008.

[13] J. Gieseler, N. Dresing, C. Palmroos, J. L. Freiherr Von Forstner, D. J. Price, R. Vainio, A. Kouloumvakos, L. Rodríguez-García, D. Trotta, V. Génot, A. Masson, M. Roth and A. Veronig, Frontiers in Astronomy and Space Sciences **9**, (2023) doi:10.3389/fspas.2022.1058810.

[14] S. J. Russell and P. Norvig, *Artificial intelligence: a modern approach*, *Prentice Hall series in artificial intelligence*, third edition, global edition ed. (Pearson, New Jersey, USA, 2016).

[15] G. James, D. Witten, T. Hastie, R. Tibshirani and J. Taylor, *An Introduction to Statistical Learning: with Applications in Python*, *Springer Texts in Statistics* (Springer International Publishing, Cham, Switzerland, 2023) doi:10.1007/978-3-031-38747-0.

[16] T. Hastie, R. Tibshirani and J. Friedman, *The Elements of Statistical Learning*, *Springer Series in Statistics* (Springer, New York, USA, 2009) doi:10.1007/978-0-387-84858-7.

[17] M. Belkin, Acta Numerica **30**, 203 (2021) doi:10.1017/S0962492921000039.

[18] G. Brown and R. Ali, Trans. Mach. Learn. Res. **2024**, (2024).

[19] N. Gopalswamy, G. Michałek, S. Yashiro, P. Mäkelä, S. Akiyama, H. Xie and A. Vourlidas, The SOHO LASCO CME Catalog – Version 2, 2025, arXiv:2407.04165 [astro-ph].

[20] A. Papaioannou, I. Sandberg, A. Anastasiadis, A. Kouloumvakos, M. K. Georgoulis, K. Tziotziou, G. Tsiropoula, P. Jiggens and A. Hilgers, Journal of Space Weather and Space Climate **6**, A42 (2016) doi:10.1051/swsc/2016035.

[21] S. Rotti, B. Aydin, M. K. Georgoulis and P. C. Martens, The Astrophysical Journal Supplement Series **262**, 29 (2022) doi:10.3847/1538-4365/ac87ac.

[22] Y. Wang and J. Guo, Astronomy & Astrophysics **691**, A54 (2024) doi:10.1051/0004-6361/202450046.

[23] V. Domingo, B. Fleck and A. I. Poland, Solar Physics **162**, 1 (1995) doi:10.1007/BF00733425.

Appendix

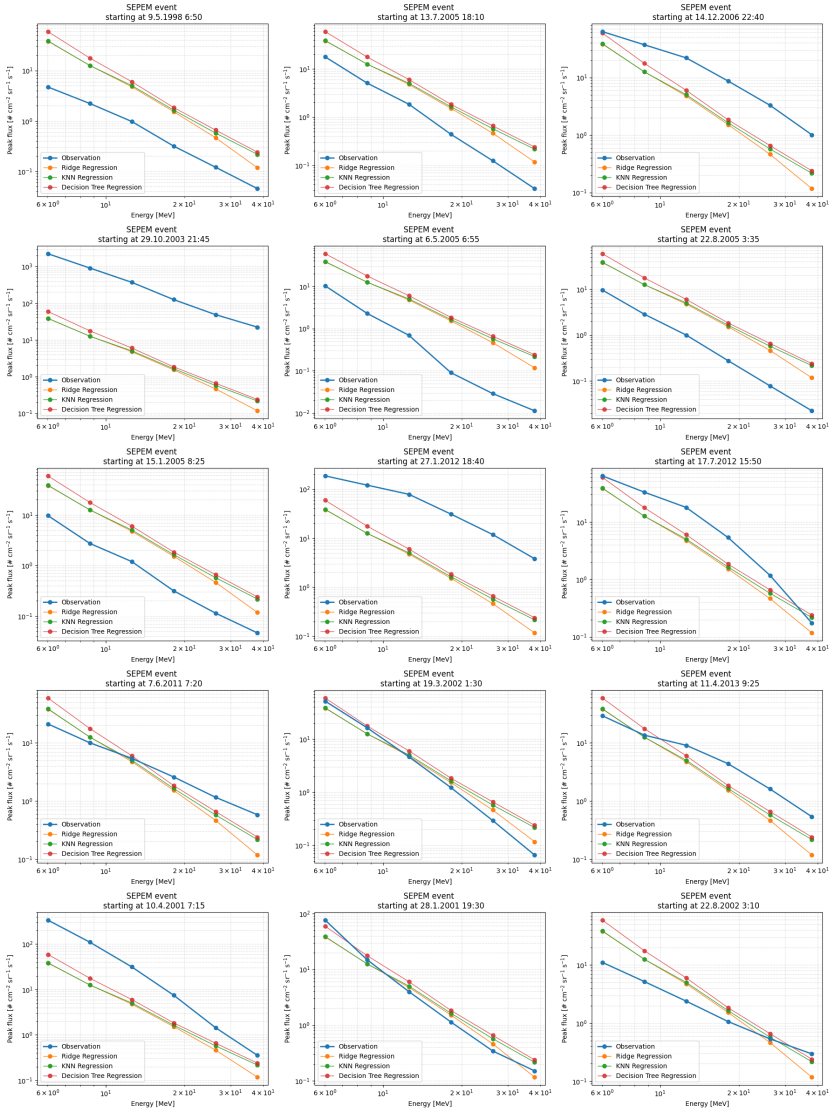

Figure 12. SEP events 1-15 of the test set of a randomly split data set with observed and predicted spectra of the ridge, KNN and decision tree regression models. Each model has been trained with the input features $F_{SXR_{peak}}$, $F_{SXR_{int}}$ and V_{CME}

.

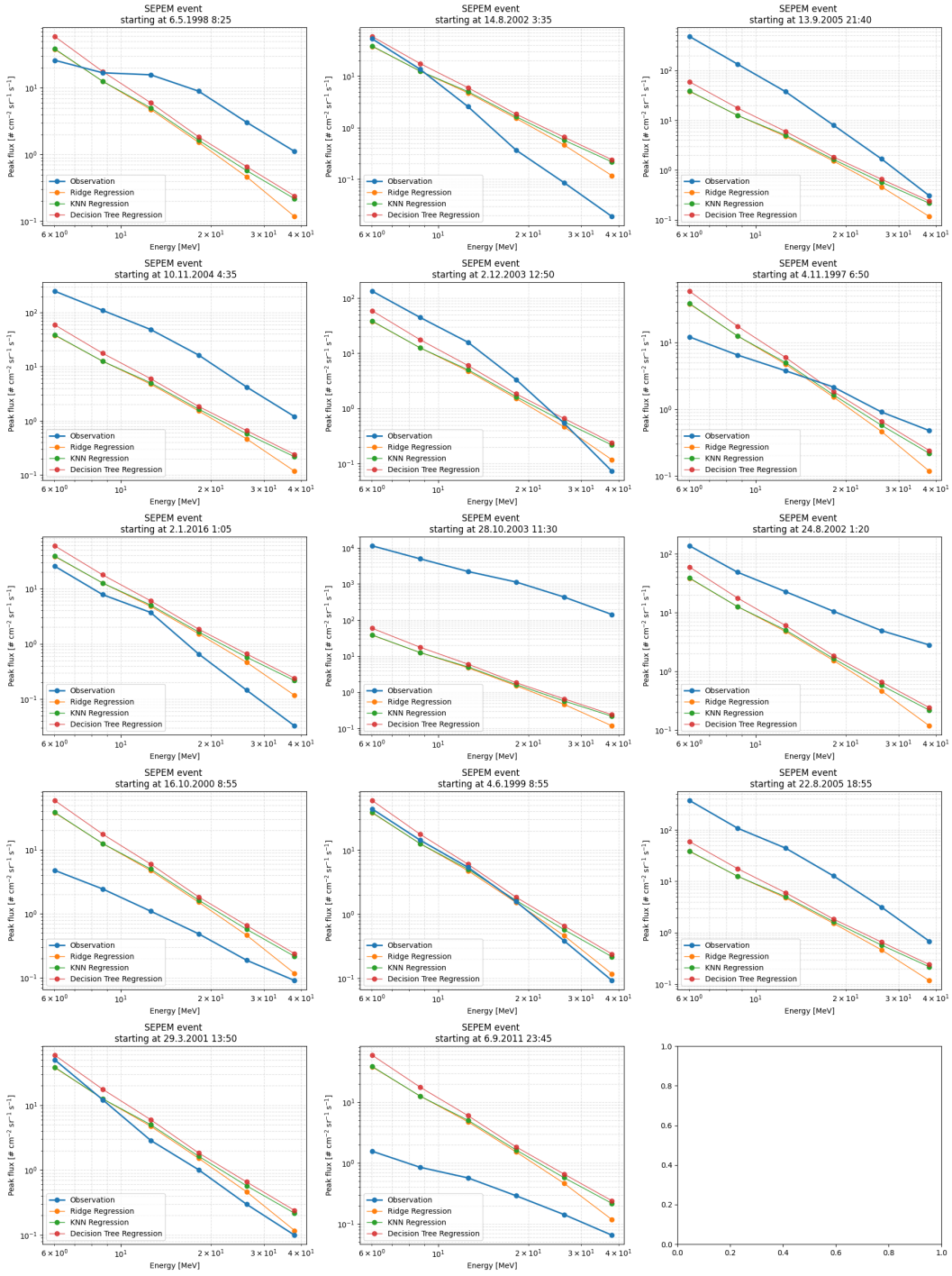

Figure 13. SEP events 16-29 of the test set of a randomly split data set with observed and predicted spectra of the ridge, KNN and decision tree regression models. Each model has been trained with the input features $F_{SXR_{peak}}$, $F_{SXR_{int}}$ and V_{CME}

.