Forecasting SEP Events based on Merged CME Catalogs using Machine Learning

by

Peter William Tarsoly

Bachelor of Science
Department of Computer Engineering and Sciences
Florida Institute of Technology
2020

A thesis
submitted to the College of Engineering and Science
at Florida Institute of Technology
in partial fulfillment of the requirements
for the degree of

Master of Science
in
Computer Engineering

Melbourne, Florida
December, 2021

We the undersigned committee
hereby approve the attached thesis

Forecasting SEP Events based on Merged CME Catalogs
using Machine Learning by Peter William Tarsoly

Philip Chan, Ph.D.
Associate Professor
Department of Computer Engineering and Sciences
Committee Chair

Ming Zhang, Ph.D.
Professor
Department of Aerospace, Physics and Space Sciences
Outside Committee Member

Georgios Anagnostopoulos, Ph.D.
Associate Professor
Department of Computer Engineering and Sciences
Committee Member

Philip Bernhard, Ph.D.
Associate Professor and Department Head
Department of Computer Engineering and Sciences

Abstract

Title:

Forecasting SEP Events based on Merged CME Catalogs

using Machine Learning

Author:

Peter William Tarsoly

Major Advisor:

Philip Chan, Ph.D.

The lack of preparation for a Solar Energetic Particle (SEP) event may be catastrophic for astronauts and aircraft passengers alike, along with their electronic equipment. It is widely theorized that SEP events are caused by Coronal Mass Ejections (CMEs), some occurring up to a full day beforehand, accompanied by additional space weather conditions. The only significant models for SEP forecasting are statistically or machine learning-based, often developed on imprecise data. We present an enhanced catalog of CMEs, along with other space weather phenomena, and their relationship with the occurrence of SEP events. Using the enhanced CME catalog, we combine machine learning techniques to create a model that achieves a TSS of 0.829, HSS of 0.712, and F1 Score of 0.714. Further, we analyze the model to determine the relative importance of each input measurement when making SEP occurrence predictions.

Table of Contents

List of Figures

List of Tables

Acknowledgements

First and foremost, I would like to express my utmost gratitude to my advisor, Dr. Philip Chan. Taking your course in Machine Learning opened my eyes and strengthened my foundational knowledge and interest in the field as a whole. Further, this work would have been much more difficult to complete without your constant support and guidance.

I would like to also thank my additional committee members. Almost a year ago, Dr. Ming Zhang introduced me to the fascinating subject of space weather. I appreciate his overall expertise and answering my constant questions about the subject, ultimately helping an engineer grasp how physicists understand the problem at-hand. Additionally, I thank Dr. Georgios Anagnostopoulos for his support throughout my undergraduate and graduate studies. His course in Pattern Recognition helped me further hone my knowledge in why certain algorithms and machine learning practices are quite useful in natural settings.

While completing this work, I received constructive advice and feedback from Peter Thomas, Jesse Torres, and Lulu Zhao. I would like to thank them for their cross-discipline input. Further, This work would not have been possible without the support of Karen Brown, Cheryl Mitravich, and Leslie Smith. Throughout my studies at Florida Tech, they put in a great amount of effort in assisting with all the necessary paperwork and other arrangements.

Finally, I want to thank my family and friends for helping make this journey an enjoyable and memorable one.

Chapter 1

Introduction

1.1 Motivation

Solar Energetic Particle (SEP) events result in potentially hazardous doses of radiation, both in Space and on Earth. As such, the consequences of not preparing for these events are grave [29]. Astronauts on or near the outside of Earth's magnetosphere, along with aircraft passengers on Earth flying over or near the magnetic poles, may be exposed to the same amount of radiation during a momentary SEP event that radiation workers are exposed to over a year. Further, electronic devices in similar conditions as these humans are prone to failure under radiation doses produced by SEP events. By providing an advanced warning that an SEP event may imminently occur, radiation exposure to humans and devices may be mitigated.

1.2 Problem

We define an SEP event as when the flux of >10 MeV protons becomes greater than 10 proton flux units (pfus). SEP events are understood to be driven by a solar phenomenon

called Coronal Mass Ejections (CMEs), often occurring hours before SEP events. The main problem we address is forecasting the occurrence of future SEP events using CME measurements.

1.3 Approach

We approach this problem through adding additional measurements to each CME event record, developing SEP forecasting models, and conducting model analyses. First, we enhance the CME catalog from the Space Weather Database Of Notifications, Knowledge, Information (DONKI) by adding measurements from another CME catalog, along with additional data from previous works addressing a similar problem. Next, we develop several machine learning models to forecast the occurrence of SEP events based on related CME measurements. These models are designed to take into account the rarity and importance of forecasting the occurrence of these events. Finally, we interpret high-performing models to determine the most important measurements and explain their most commonly made errors.

1.4 Contributions

In this work, we propose two main contributions: an improved dataset relating CMEs to SEP events, and a neural network model to forecast the occurrence of SEPs using CME measurements. For the dataset, we build on the work of Torres [32] who linked and used events from the CDAW CME catalog to address a similar problem. For the forecasting model, we handle the imbalance of SEP to non-SEP events through two main techniques. First, we employ two stages of training to separately perform representation and classifier learning. Second, to assist the representation learning

stage, we introduce an autoencoder branch to the network.

1.5 Organization

In chapter 2, we discuss work related to our approach to forecast SEP events. We additionally explain previous efforts used in handling imbalanced data, along with techniques to analyze these models. In chapter 3, we describe our process of enhancing the DONKI CME Catalog. In chapter 4, we introduce our forecasting models including our proposed representation learning technique. We further present the performance results of these models, analyze the importance of measurements (hereinafter features), and a discussion on common mistakes made by the model using our technique.

Chapter 2

Related Work

2.1 SEP Forecasting

The main line of work we are interested in within SEP forecasting is focused on the occurrence of an SEP event. Typically, these manifest in two forms: physics and machine learning based forecasting. However, these methods are not necessarily mutually exclusive; for instance, the physics-based methods are typically data-driven, whereas the machine learning methods often incorporate previous physics-based methods.

2.1.1 Physics-Based SEP Forecasting

Richardson *et al.* [22] present techniques for associating coronal mass ejections (CMEs) with solar energetic particle (SEP) events. Based on work originally meant to estimate the intensity of an SEP event based on its associated CME's characteristics, the authors dive further by exploring ways of mitigating the sharp imbalance between CMEs that are and are not associated with SEPs. First, they introduce a previously-derived gaussian fit formula to predict the peak proton intensity (in units of proton flux units

or pfus) of an SEP based on its related CME's linear speed, angular distance from the magnetic connection longitude between the Sun and the observing spacecraft (commonly known as connection angle), and the average width of all SEP-related CMEs. Further, they mention the formula was created using data from 334 CMEs within solar cycle 24, including observations from both Earth and Sun-orbiting spacecraft. Using the formula, the authors attempt to predict the occurrence of SEPs by thresholding its output, commonly using either 10^{-1} or 10^{-4} pfus. The event is classified as an SEP if its formula output is greater than the threshold value. However, they find that this immediately is not quite effective. By evaluating the formula on part of the data used to create the formula, the authors point out that, without any modifications, it routinely over-predicts the proton intensity during non-SEP related CMEs, leading to a high number of false alarms. In order to reduce the number of false alarms predicted, the authors explore a number of properties associated with the related CME to exclude certain events prior to predicting their peak proton intensity. These properties include creating thresholds for CME speed, width, their product, and Types II and III radio wave emissions. By excluding certain events below the threshold, they are able to reduce the false alarms from 66% of the events predicted to 19%, in the case of the product of CME speed and width. While the authors achieve significant results through their event exclusion method, they point out that the evaluation data was not exclusive of the data used to create the peak proton intensity formula. In order to test the robustness of their technique, they evaluate their method on data from solar cycle 23. However, they point out that this period only contained one Earth-orbiting spacecraft recording relevant data. Because of this, they claim many of the speed and width measurements are erroneous. They still find that using the formula, in conjunction with removing events using Type II and III emissions, is effective in predicting both whether or not a CME is associated with an SEP, and its respective peak proton

intensity.

Bruno and Richardson [18] present a model to predict the peak proton intensity (in proton flux units or pfus) of solar energetic particle (SEP) events based on a related CME's connection angle and the desired energy level. This contrasts with previous work that solely based peak proton intensity prediction on connection angle, and only considered the SEP event's energy spectrum between 14 and 34 MeV. By making particle energy a parameter in the model, the authors extend previous efforts to successfully predict proton intensity between 10 and 130 MeV. After analyzing the energy spectra of 32 SEP events between 2010 and 2014, they derived a multivariate gaussian-based formula for predicting peak particle intensity, based on energy level and connection angle. They find that their estimation matched each of the 32 peaks well, across their energy spectra. They additionally evaluated their model on an additional separate 20 SEP events from 2011 to 2017. Compared to the original predictions between 14 and 34 MeV, the authors accurately extrapolated peak particle intensity predictions between 10 and 130 MeV for each SEP event.

2.1.2 SEP Forecasting using Machine Learning

Boubrahimi *et al.* [15] present a decision tree-based method for forecasting SEPs using X-Ray data. Specifically, they employ time-series data from GOES to predict the occurrence of greater than 100 MeV SEP events. Interestingly, they used 94 X-Ray events from 1997 to 2013, with 47 both SEP and non-SEP related events. Though their method of combining multiple sources of data may improve model performance, the class-balanced ratio is not representative of the problem as observed in nature.

Kahler and Ling [25] explore various methods to forecast SEP events by using X-Ray measurements associated with solar flare events. They first provide an in-depth study of the relationship between SEP and flare events from both the perspective of peak flux

between different X-Ray bands and flare source location. They then create two models to forecast the occurrence of SEPs, based on flares from the western hemisphere of the sun, using multilayer perceptron network and k nearest neighbor techniques. Their data consisted of 261 solar flare events with a 4 to 1 non-SEP to SEP class imbalance ratio. They conclude that the multilayer perceptron network is the better choice, because they claim the decision boundary of the model using k nearest neighbors would be difficult to implement for operational use.

Inceoglu *et al.* [23] investigate modeling the likelihood of relations between solar flares, CMEs, and SEP events within magnetically-active regions of the sun. Specifically, they break their events into three classes: exclusively flares, flare+CME+SEP, and exclusively CMEs. Flare events were supplied from GOES, whereas CME events were supplied from DONKI between 2010 and 2018. Then, they construct separate classifier models using support vector machines and multilayer perceptron networks. They find that the support vector machine performs the best, specifically when tasked with determining if CMEs will not be associated with flares or SEPs.

Brea *et al.* [16] investigate the use of machine learning models with X-Ray, Type II and IV radio bursts, and CME events to predict the occurrence of SEPs. The models they explored included logistic regression, adaboost, and support vector machines. They find that their logistic regression and support vector machine model outperforms NOAA's Proton Prediction Model when compared using the Heidke Skill Score.

Torres presents machine learning-based approaches to predict the occurrence and intensity of SEP events [32]. In order to forecast the occurrence of an event, entries from the CDAW CME catalog are used to train a multi-layer perceptron (MLP) neural network. In addition to using measurements included in the catalog, Torres incorporates derived solar phenomena such as Type II radio bursts, Diffusive Shock Acceleration, and a non-machine learning based equation to predict peak proton inten-

sity. Additionally, a feature importance method for MLP neural networks is described. Torres's method determines the importance of each feature by multiplying together the network weights along the path of a feature, from input to output. The features are then ranked by importance. After using this method on the model, Torres found that sunspot number, Type II radio bursts, and the CME width were the most influential features. Torres then continues on to proposing a method to predict the intensity of an SEP event. The method involves time-series electron and X-ray intensity values as input, and seeks to predict the proton intensity up to an hour ahead in the future. Torres developed recurrent neural network-based approaches, including using rising, falling, and background intensity models to decrease prediction error in terms of both lag and intensity.

Lavasa *et al.* [28] studied the application of various machine learning algorithms to predict the occurrence of SEPs at 10 MeV using CME and solar flare properties. They treated the task as a classification problem, where each CME event is either related or not to an SEP event. Their dataset was created using CME speed and width entries from the CDAW CME Catalog from 1997-2013, along with solar flare data from GOES. CMEs were only included if they were associated with a C, M, or X class flare event. Their final dataset consisted of 33221 solar flares, 6218 CME events, and 257 SEP events. Further, they created separate datasets to study the impact of utilizing the properties of solar flares and CMEs, both together and separately, to predict the occurrence of SEPs. At best, their class imbalance ratio was 25 non-SEP to 1 SEP related entries; at worst, the ratio was 128 non-SEP to 1 SEP related entries. The machine learning algorithms compared included support vector machines, logistic regression, multi-layer perceptron classifier, and decision tree-based algorithms. They additionally assess different parameter and hyperparameter optimization methods, including undersampling the majority class in the training set. Nevertheless, they find

training the models on the data without modifications to class distribution typically performs the best, and that the logistic regression and random forest models proved to perform the best when minimizing the false alarm rate (FAR) or maximizing the probability of detection (POD) respectively. After running the feature permutation performance algorithm on all models, they determined that CME speed, width, and solar flare fluence were the most important features across all models.

2.2 Machine Learning Techniques

Due to the nature of the problem, two main machine learning techniques were explored: handling imbalanced data, and explaining both predictions and models. Considering SEP events are very rare events, the imbalanced dataset is handled with caution. Further, due to the physics background of the problem, it is important to understand the potential physical phenomena driving the occurrences of SEP events.

2.2.1 Handling Imbalanced Data

Kang et al [26] explore treating neural networks as a combination of two separate entities, as learned data representation and a classifier, in order to increase the accuracy of classifying long-tail distributed, underrepresented classes. Within a neural network-based classifier, the output layer is typically considered the classifier. Generally speaking, it need not even be a network layer, as the outputs of the representation can be treated as inputs to any other classifier of choice. To choose a classifier fit for the newly-learned representation, four main options are investigated. First, in a method called Classifier Re-Training (cRT), only the output layer is retrained using class-balanced sampling. Second, Nearest Class Mean (NCM) takes a departure from the output layer approach of a classifier. Instead, the mean vector for the output of

the representation for each class is computed. Each test sample is then classified based on the closest class mean vector. Third, the tau-normalized classifier normalizes every weight between the representation output and the classifier by the L2 norm raised to a hyperparameter, tau, of the weights connecting the former layer with the latter on a class-by-class basis. The authors claim that this method corrects imbalanced decision boundaries. Finally, Learnable Weight Scaling (LWS) is similar to the third option, but effectively allows for the learning of the tau hyperparameter. In order to compare the methods outlined for representation learning and classification, three image datasets with hundreds to thousands of categories, many following a long-tailed distribution, were introduced. For each dataset, at least one network architecture was explored, with each at minimum including a form of hidden and output layers. They first compare the sampling strategies presented for representation learning, revealing that instance-balanced sampling, combined with decoupled strategies for training, performs the best on classes with low representation. Second, they compare the weight norms of the different classifier options, finding that the tau-norm method is the most consistent amongst all classes. Finally, they compare their classifiers to others meant to deal with data imbalances, which shows the cRT, tau-norm, and LWS methods perform better than all previously proposed methods for the given datasets and network architectures.

Zhou *et al.* [36] present a framework called a Bilateral Branch Network (BBN) to separate learning representations from classifiers in order to increase classifier performance on "long-tailed", or underrepresented, classes. The authors created the framework after experimenting with different variations of fixing representation and classifier learning separately. The proposed framework consists of a neural network with three parts: a representation branch, a classifier branch, and a cumulative learning part that combines results from both together. The primary difference between the two branches of the neural network is their training regimen, leaving a desired effect during testing

and general inferencing. For each iteration of training, each branch is given a different sample in accordance with a specific sampling scheme. For the branch responsible for representation learning, samples are selected based off of the as-is data distribution. As there is no modification to how exactly samples are selected, the branch is also known as the conventional learning branch. Conversely, the branch responsible for learning the classifier, also known as the re-balancing branch, is given samples based off of a reversed sampler. For example, if class A makes up 90 percent and class B makes up the remaining 10 percent of the training data, samples from class A are picked 10 percent and those from class B are picked 90 percent of the time for the rebalancing branch. Additionally, weights within the hidden layers are shared amongst the two branches. Results from both branches are combined using a cumulative learning strategy. Before the output is generated, each branch output is multiplied by an iteration-varying coefficient, which is then considered the input for the output layer. These results are multiplied by their respective weight matrices, summed up, and finally go through a softmax activation function. The resulting error function is defined by a sum of the errors of each respective network, with each being multiplied by their same aforementioned iteration-varying coefficient. The coefficient acts in a way, such that at the beginning of training, the error function is solely the error from the conventional branch; at the end, the converse is true. Since the coefficient varies quadratically, the training focuses more on the representation learning first, then quickly emphasizes classifier learning towards the end. During test and inference in general, these coefficients are set to 0.5 to equally weight the branches.

Wang *et al.* [34] describe an additional approach to classification that seeks to increase the performance of classifying samples from long-tailed, underrepresented classes. Through a combination of model architecture and loss modifications, the authors show improved performance over other state of the art techniques when faced

with class-imbalanced problems. They introduce their approach as a framework that is a neural network model with two branches: one to learn features (or representation), and another to learn a classifier. During training, the feature learning branch is given samples with no modifications to sampling, whereas the classifier branch uses class-balanced sampling. Though this implies the branches receive different samples per training iteration, they share their hidden, or "backbone", network layers. Additionally, the two use different loss functions; while the classifier branch uses standard cross entropy, the feature branch uses supervised contrastive loss (SCL). SCL is minimized by both maximizing the similarity of the output with those of the same class, while minimizing the similarity with the output of other classes. Since this implies the need to compute the similarities for every sample, thereby drastically increasing the computational complexity, the authors introduce the use of one prototype output per class, and redefining the loss as prototypical supervised contrastive (PSC) loss. Finally, the importance of each branch within the overall loss function changes during training by multiplying the branch losses with an iteration-varying coefficient. Training starts off with no classifier loss, and gradually transitions to no feature loss instead by the end. During inference, only the classifier branch is used to determine model output.

2.2.2 Explaining Predictions and Models

In this section, we describe techniques to interpret machine learning techniques at two levels: on an individual sample basis, and a global model basis. The individual sample techniques attempt to explain why certain predictions are made; meanwhile, the global model techniques attempt to explain the function of the entire model.

2.2.2.1 Explaining Model Predictions

Ribeiro *et al.* [30] present a technique to address the trustworthiness of a model, both at the sample prediction and overall model levels. The authors claim and show that their method is interpretable, is faithful to the model both around a prediction and subsequently globally trusted, and is agnostic to model type. Through a number of experiments involving simulated and real human participants, the authors show that their technique outperforms numerous previous ones on key aspects of model and prediction explainability and trustworthiness. The authors first present their technique around the criteria they lay out for prediction explainers. To address explainer interpretability and trustworthiness around a certain prediction, they introduce a new classifier that uses a simple linear combination of important features as its decision boundary. They train the explainer model by attempting to match the predictor's output using a variant of mean squared error as the loss function. While discussing the loss function, they address the model agnostic criteria by training the explainer model with points generated within the vicinity of the specific sample regarding the specific features deemed important to the specific sample being explained. By addressing three of the aforementioned criteria, the authors call explanations generated from this first part of their technique Local Interpretable Model-Agnostic Explanations (LIME).

Jeyakumar *et al.* [24] compare a number of neural network classifier explanation methods, and present a new technique to explain model predictions by selecting similar training set samples. The authors start by presenting the most prominent explanation techniques to date, including LIME. They claim that most methods center around superimposing the explanation itself back onto a data sample, such as highlighting a part of an image. While these methods proved to be successful in the past, they typically are designed for specific data domains, presenting a nontrivial task to adapt each to other types of data. In order to overcome this obstacle, they present an

explanation method based on providing a given number of similar training samples. Further, cosine similarity across activation values per sample is employed to determine how close each training sample is with respect to the given sample being explained.

2.2.2.2 Explaining Entire Models

As previously mentioned, Torres [32] proposed a method to determine the importance of each feature in an MLP network. Torres's method determines the importance of each feature by multiplying together the network weights along the path of a feature, from input to output. This is achieved through multiplying the column-normalized weight matrices together. Specifically, they define a multilayer perceptron neural network with one hidden layer, and weight matrices W_1 and W_2. Then, they normalize the columns and take the absolute value of each weight, resulting in $\hat{W}_1$ and $\hat{W}_2$. In order to determine the importance of each feature, represented in vector I, they perform the following operation:

$$I = \hat{W}_1 \hat{W}_2 \tag{2.1}$$

Torres proved that by normalizing the weight matrix columns before multiplying them together, all values in I are also normalized between 0 and 1. Further, the feature with the largest value in S is considered most important. As such, the rest are ranked in descending order by each feature's importance.

Ribeiro *et al.* [30] present a technique using LIME to address the functionality of the global model by using LIME to explain a limited number of training samples. The proposed algorithm addresses the picking problem by seeking to attain the maximum coverage of globally-important features, all while constraining the number of explained samples to within a predefined budget. Since finding the exact solution to the picking

problem is computationally intractable, they introduce a greedy strategy to select the samples that add the most amount of coverage when added to the set of important samples. The authors coin this part of their technique the submodular pick (SP), referring to the entire method as SP-LIME. Unlike Torres's method, SP-LIME only implicitly describes the importance of certain features in a given model. Though not emphasized in their work, they define a feature importance calculation as a more explicit understanding of the model:

$$I_j = \sqrt{\sum_{i=1}^{n} |W_{i,j}|} \tag{2.2}$$

Where j is an input feature, $W_{i,j}$ is the linear coefficient obtained from LIME for the explanation of the ith event and jth feature, I_j is the importance of the jth feature, and n are the number of instances that are explained.

While introducing the Random Forest technique, Breiman [17] additionally introduced a method to determine the relative importance of each feature for not just Random Forests, but any classifier. In its simplest form, the values within each feature in the test set is individually permuted, and the performance drop is recorded. Features associated with the highest relative performance drop are considered most important; therefore, the features are sorted in descending order of performance drop as most to least important. While this method appears to produce meaningful results, we found it unreliable and decided not to use it further on in analyzing our results.

Another way to interpret a model is through finding the relations, formed through training, between variables themselves and how they contribute to the model output. Particularly for natural processes, it may be useful to derive a mathematical equation where features are treated as variables in an algebraic form. The process to find such models is commonly referred to as symbolic regression, and it is often approached using

a genetic algorithm-based technique.

Cranmer *et al.* [19] describe the creating algebraic equations, through symbolic regression, through neural network models of physical phenomena. In order to guide the eventual process of symbolic regression, they investigate using Graph Neural Networks (GNNs). GNNs are particularly useful for representing relationships between physical objects and their properties. Similar to an actual graph, a GNN captures object properties analogous to graph vertices, and interactions analogous to graph edges. A typical GNN is a collection of multilayer perceptron (MLP) neural networks, some of which are effectively tasked with working on subproblems. First layers typically deal with learning how objects, represented as graph nodes, interact with each other, giving them the name edge models. Outputs of the edge models are referred to as messages, which are effectively features created during training. The output layer consists of MLPs tasked with learning how interactions between objects affect individual objects, and take both messages and object properties as input. The authors claim that this network structure creates an inductive bias necessary to model the physical world while simplifying the symbolic regression process. In order to further generalize the results of the GNN, the authors experimented with different forms of regularizing the edge model results.

In their symbolic regression technique, a population of equations are created, and each individual is evaluated using a fitness measure that factors in accuracy and simplicity. In order to create relevant symbolic equations, they use the intermediate and final models within the GNN to determine the fitness of each equation. In this case, they assume that the output model from the GNN can be composed of results from intermediate models, including edge models. This assumption drastically reduces their search space, decreasing the amount of effort needed for fit results, because they divide up the overall model into smaller chunks.

The main problem they investigate is deriving an algebraic equation for determining the density of a dark matter halo, given surrounding halo properties. Prior to their work, there were no symbolic models that accurately represented the phenomenon. After ingesting the dataset of halos, and defining how they relate to each other, a GNN was created and symbolic regression was successfully performed, both for instances with and without the mass of a halo.

Although symbolic regression typically takes the form of a genetic algorithm, Sahoo *et al.* [31] describe a way of integrating it directly into a neural network model. The authors achieve this through a number of modifications to a typical network including architecture, training, and hyperparameter choices.

The first modification they propose is a change to the activation functions within the hidden and output layers. Within the hidden layers, every unit's activation function is either a unique unary mathematical operator, such as sine, cosine, and an identity operator of the dot product pre-activation, or a multiplicative operator that multiplies two input values together. Within the output layer, special division operators are introduced that ensure the denominators are never negative or approach zero.

The second modification regards the training process. In order to account for the division by zero problem, penalty terms are introduced into the loss function to avoid the problem on data both inside and outside of the observed range during training. Additionally, in order to realize meaningful coefficients as weights between hidden unit layers, a three-step, phased regularization scheme is utilized to remove meaningless coefficients.

The third change, tuning hyperparameters, selects the number of layers needed for the model's optimal performance. Instead of solely basing performance on error, they define it by model error on the test set, simplicity as a function of sparsity amongst the weights, and optionally error on data out of range of that seen within the test set.

When compared with the symbolic regression algorithm also used in Cranmer *et al.* [19], Sahoo *et al.*'s technique consistently outperformed its genetic algorithm-based counterpart. Additionally, when given problems representing physical systems, their proposed technique performed with lower error. Further, their technique generalized much better to data outside the range used within the test set; instead of increasing, the error decreased on the extrapolated inputs.

Further, Udrescu and Tegmark [33] present a deterministic algorithm that discovers scientific symbolic models from a given set of associated inputs and outputs known as mysteries. Their method differs from typical symbolic regression, as genetic algorithms are stochastic in nature. In short, through a divide-and-conquer technique, the algorithm "solves" mysteries by breaking them down into smaller parts which are computationally tractable to realize into symbolic forms. While their method relies on input and target output data, one could use their method to create a symbolic representation of a model by determining the model output for every training sample, and then use this technique to derive the expression.

Chapter 3

Enhancing DONKI CME Entries using CDAW CME Entries

3.1 Motivation

The CDAW CME dataset [1] was previously used to create a classifier model to predict whether a particular CME was the catalyst of a Solar Energetic Particle (SEP) event [32]. Considering CDAW gathers measurements for CME entries from Earth-centric instruments, it can only measure a projection of a given event's true attributes. On the other hand, the DONKI CME dataset [5] gathers measurements from Sun-centric instruments. Although these measurements are still projections of an event's true attributes, additional information is gained compared to solely using Earth-centric data. As an example, Figure 3.1 contains the event distribution of widths and half widths, for CDAW and DONKI respectively. They are both heavy-tailed distributions, with a secondary peak around 360 degrees for CDAW. These events are considered to have a "halo", an accurate description of the observation from Earth; however, most of

them are much smaller than 360 degrees. This problem does not exist within DONKI, because most events were measured using at least two instruments with at least one orbiting the Sun.

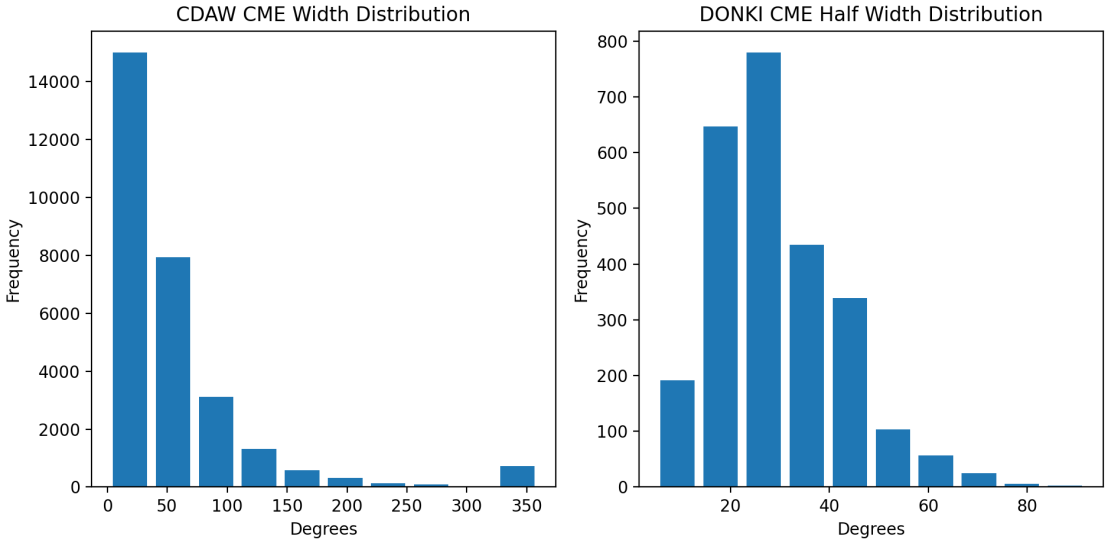

Figure 3.1: Event distributions of width and half width, for CDAW and DONKI entries respectively.

Additionally, due to the nature of instruments used amongst CDAW and DONKI databases, the recorded location of an event contains more information within DONKI compared to CDAW. Within CDAW entries, only positional angles of the event, from Earth's perspective, are recorded; meanwhile, DONKI records the latitude and longitude on the Sun of each event. This increases the accuracy of computing additionally useful values, such as the event's connection angle.

Further, every DONKI and CDAW CME entry contains a label of whether it is associated with an SEP. Within the CDAW dataset, for every CME associated with an SEP, there are 270 that are not. Using the DONKI dataset decreases this imbalance, as there are only 62 non-SEP CMEs for every CME associated with an SEP. This is driven by the fact that DONKI primarily records CMEs with fast speeds and are large in

width, whereas CDAW records all CMEs that can be observed from Earth's perspective. By decreasing the imbalance between events that are and are not associated with SEPs, all while focusing on significant CMEs, we are able to create more meaningful models to forecast SEPs.

While the DONKI CME catalog contains advantages over its CDAW counterpart, it does have some drawbacks. The primary problem with using the DONKI catalog is its current time span. The catalog, additionally labelled with SEP relations, only covers a 7 year span, which is less than a complete 11 year solar cycle. This constraint makes it difficult to leverage the cyclical nature of event frequencies to operationally evaluate forecasting models. Separately, for every actual CME event, the catalog may contain duplicate entries. Fortunately, most of these contain an indicator revealing the most accurate entry; however, there are additional rare cases where the catalog itself records one event as two separate events, and both may contain additional entries themselves. We discuss how we identify and resolve these duplicate entries in Section 3.3.

After comparing both DONKI and CDAW databases, DONKI is more suitable for building a model due to its accuracy for most event measurements. However, CDAW CMEs contain measurements that are not as affected by the hindrance introduced by taking measurements from Earth's perspective that DONKI does not contain, including event links to other space weather phenomena such as Type II radio waves completed in previous works [32]. Since DONKI events are more suitable for our purposes but are limited with information, we supplement them with information from CDAW CME entries, Type II radio waves, and sunspot number by merging their catalog entries together. Figure 3.2 depicts the previous and newly matched sources of data.

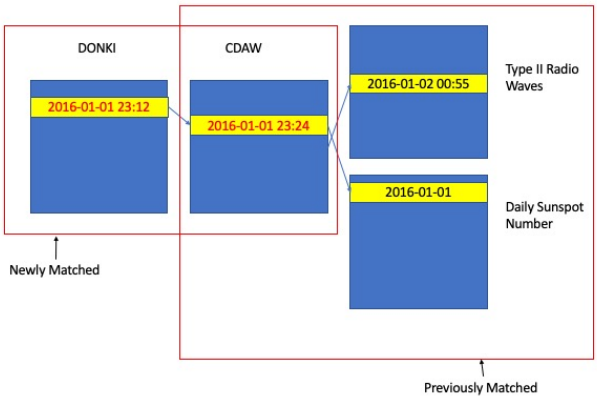

Figure 3.2: A comparison of previously [32] versus newly matched sources of data. This chapter addresses the process of linking DONKI to CDAW CME entries in order to gain additional information useful to forecasting future SEP events.

3.2 The Matching Process

We begin the match process by considering the potential matches for every DONKI entry. From this perspective, there are four types of scenarios:

1. There is a one-to-one match with a CDAW entry
2. There are no CDAW candidate entries
3. There are multiple CDAW candidate entries
4. There are multiple DONKI entries associated with a single CDAW entry

The first scenario is ideal; therefore, the matching process primarily addresses the other scenarios. For the second scenario, Without additional CDAW entries, there is no feasible way to address the given DONKI entry other than to not include the event within the combined data. We address the third scenario by attempting to find a matching CDAW entry using a number of criteria defined in Section 3.2.1. Finally, we describe how we perform a second pass on the DONKI entries, to resolve the fourth scenario, in Section 3.2.2.

3.2.1 Addressing Multiple CDAW Candidates

Originally, we attempted matching events solely based on a time window. However, this proved to create incorrect matches; for instance, two separate CMEs in CDAW and DONKI can exist where they occurred on opposite sides of the sun at the same time. To account for these events, matching criteria for DONKI to CDAW entries were created based on attributes of the entries in the following order:

1. Time
2. Approximate MPA with uncertainty
3. MPA quadrant match
4. Two entries are close in location
5. Two entries are close in speed

3.2.1.1 Time

The time criterion first filters out events outside of a $\pm$3-hour time window of the DONKI entry. After applying the filter, 127 DONKI entries are automatically removed with no CDAW candidate entries within their time window, removing the second type of match listed above. If there is only one candidate CDAW entry, the entries are matched. If there are multiple CDAW candidates left, the absolute elapsed entry start time is taken and sorted in ascending order between each and the DONKI entry. If the difference in elapsed time between the closest two are greater than 36 minutes, equivalently the amount of time that 3 frames occur during analysis on events input into CDAW, a one-to-one match is recorded between the DONKI and closest CDAW entry in time. The expression can be rewritten as follows:

$$difference_1 + 36\ minutes < difference_2 \tag{3.1}$$

If the condition is not met, the match is considered ambiguous using only time, and the candidate entries continue onto additional matching criteria.

3.2.1.2 Approximate MPA with uncertainty

The next criteria address the entries on an event location basis. The motivation to additionally consider location is that two unrelated CMEs may occur at or near the same time, creating a potential for mismatch between DONKI and CDAW event records. Figure 3.3 illustrates this exact dilemma. For a DONKI CME entry with a recorded start time of 00:50 on November 13th, there are multiple CDAW CME records within its time window. In CDAW, the event moving to the lower left has a start time of 00:54, and the one moving to the right has a start time of 1:12. While both events would not be filtered out by the first criterion, the location data of the DONKI entry reveal that the correct match is the entry on 00:54.

The most accurate location attribute associated with each CME entry in CDAW is the Measurement Position Angle (MPA). The MPA of a CME is obtained by finding the counterclockwise angle between the Sun's equivalent of a north pole (90 degrees latitude, 0 degrees longitude) and its leading edge as seen from Earth. For the second criterion, using each entry's latitude and longitude, we approximate the MPA and its uncertainty for each DONKI event and filter out additional candidate CDAW events. We start by approximating MPA for each DONKI entry:

$$MPA = atan2(-cos(latitude) * sin(longitude), sin(latitude)) \tag{3.2}$$

Since the range of arctangent is in between -180 and 180 degrees, we add 360 degrees

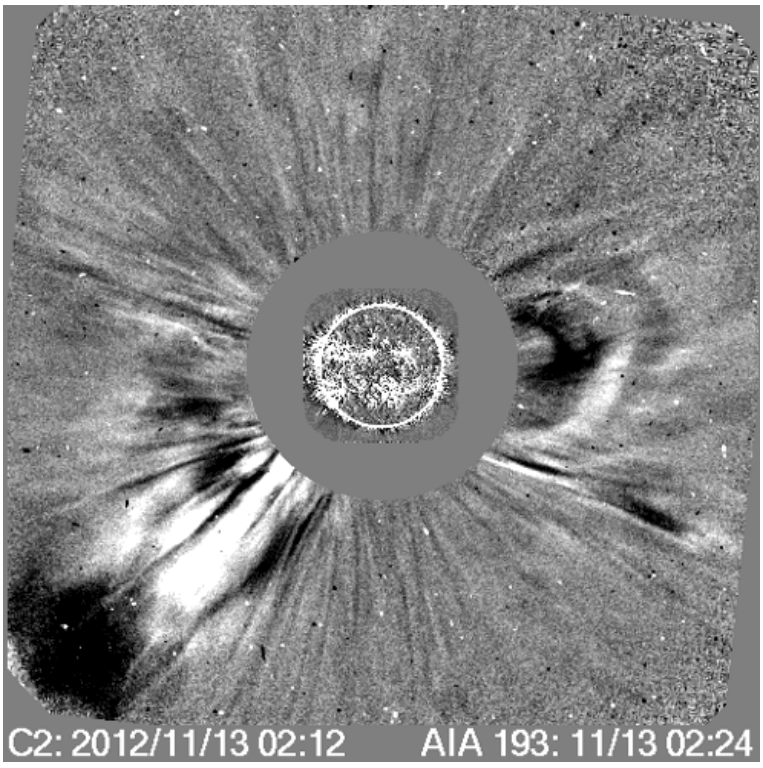

Figure 3.3: A coronagraph capture from the instrument used by CDAW to determine CME measurements showing at least two distinct CMEs (one at 00:54 on the lower left and 1:12 on the right) starting within 20 minutes of each other [4].

to the output if MPA is negative to constrain the range in between 0 and 360 degrees, consistent with how CDAW records MPA. We then calculate the MPA uncertainty:

$$\Delta MPA = \frac{|sin(\lambda)|}{sin^2(\lambda) + sin^2(\lambda) * cos^2(\lambda)}[|\Delta\varphi * cos(\varphi) * cos(\lambda)| + |\Delta\lambda * \frac{sin(\varphi)}{sin(\lambda)}|] \quad (3.3)$$

In the above equation, lambda and phi are latitude and longitude respectively. Delta lambda and phi represent the tolerance for each measurement; we currently fix these at 15 degrees. Once we determine the uncertainty, we construct bounds by adding and subtracting it from the approximate MPA. After accounting for wraparound in MPA, we filter out all candidate events outside of the range. If a one-to-one match does not exist, the candidate CDAW events are restored from the time criterion for additional

Table 3.1: The Quadrant-based approach to MPA matching.

Longitude	**Latitude**	**CDAW MPA**	**Quadrant**
Negative or ±Tolerance	Positive or ±Tolerance	0-90	1
Negative or ±Tolerance	Negative	90-180	2
Positive	Negative or ±Tolerance	180-270	3
Positive	Positive	270-360	4
Any	Any	Halo (large speed)	Any
Close to 0 or ±180	±Tolerance	Halo (small speed)	Any
Within ±Tolerance, 180 - Tolerance, or -180+Tolerance	$Magnitude < Tolerance$	Any	1 or 4
Within ±Tolerance, 180 - Tolerance, or -180+Tolerance	$Value > Tolerance$	Any	1 or 4
Within ±Tolerance, 180 - Tolerance, or -180+Tolerance	$Value < -Tolerance$	Any	2 or 3
$Value > Tolerance$	$Magnitude < Tolerance$	Any	3 or 4

checks.

3.2.1.3 MPA quadrant match

The second criterion has the potential to produce multiple candidate CDAW entries for a single DONKI entry, especially given the range of MPA uncertainty associated with a given event. When this case arises, the third criterion is applied on all candidate CDAW entries that satisfied the first criterion. Additionally, we take the latitude and longitude of each DONKI event, find acceptable ranges of MPA for the entry, and filter out CDAW entries that exist outside of these ranges. These ranges, shown in Table 3.1, are quadrant-based, and allow for ±15-degree tolerance near the edges of the quadrants.

3.2.1.4 Two entries are close in location

If multiple CDAW candidates still exist past the primary location-based methods, the fourth criterion considers matching the two closest CDAW entries with respect to angular distance in MPA. The angular distance between each CDAW entry and the DONKI entry are first taken and sorted in ascending order. If the difference of the distances of the two closest CDAW entries exceeds 30 degrees, we match the DONKI entry with the closest CDAW entry. This inequality can be represented as follows:

$$distance_1 + 30\ degrees < distance_2 \tag{3.4}$$

3.2.1.5 Two entries are close in speed

If the closest CDAW entries do not meet the previous criterion, the fifth criterion attempts to match entries based on speed. While linear speed is used for DONKI entries, the second order speed at 20 solar radii is used for CDAW entries. Like the fourth criterion, the absolute values of speed differences between each candidate CDAW entry against the DONKI entry are taken and sorted in ascending order. If the two closest CDAW entries in speed exceed 400 km/s, the closest entry in speed is matched. Similarly, the inequality for this final criterion can be represented in the following manner:

$$difference_1 + 400\ \frac{km}{s} < difference_2 \tag{3.5}$$

The remaining DONKI entries with multiple candidate CDAW entries are reviewed to see if the CDAW entry with the closest time also has the closest speed relative to the other candidates. In that case, a match is made; otherwise, the events are flagged for manual review.

3.2.2 Addressing Multiple DONKI Entries Matching a CDAW Entry

While the matching technique thus far can resolve the problem of multiple CDAW entries matching a single DONKI entry, it does not solve the opposite problem where multiple DONKI entries match a single CDAW entry. If multiple DONKI entries only match to one CDAW entry, all but one of the matches are kept by manual inspection. In our case, 86 of these pairs exist, thereby solving this problem by the deletion of 43 DONKI entries. In the case where only one DONKI entry matches to one CDAW entry, and other DONKI entries have multiple candidate CDAW events, that DONKI entry is matched with the CDAW entry. In the final case, when the DONKI entries with the same CDAW entry all have multiple CDAW candidate matches, all top unique CDAW candidate entries are ordered and matched with the DONKI entries.

3.3 Additional Data Modifications

Prior to starting the merging process, we remove certain entries within DONKI that are duplicate entries. If two events have a start time within an hour of each other and share the same source location, the event with the later start time is considered a duplicate and is removed. Additionally, after the merging process, we remove CMEs that occur during SEP events, but are not the cause of the SEPs themselves. These CMEs can hamper the objective of associating whether a CME is considered to cause an SEP event, as they occur during high proton intensity levels, yet do not contribute to the ongoing SEP itself.

3.4 Statistics of the Enhanced Dataset

A summary of the dataset statistics is presented in Table 3.2. The CDAW dataset contains 29,217 entries of CME events, spanning from January 1996 to March 2018. However, the DONKI dataset only contains 2,585 events, from April 2010 to September 2017. After removing DONKI entries with no CDAW candidate entry within their time window, entries occurring during SEP events, duplicate entries, and entries with only one CDAW candidate entry, sharing it with another DONKI entry, our final dataset consists of 2,394 CMEs. Of the 2,394 CMEs, only 39 are associated with SEPs; therefore, the imbalance ratio is one SEP-related event to 60 non SEP-related events.

Table 3.2: Statistics of the DONKI, CDAW, and Enhanced DONKI datasets.

Dataset	**Number of Events**	**Number of Non-SEPs**	**Number of SEPs**	**Non-SEP:SEP Ratio**	**Date Range**
CDAW	29,217	29,109	108	270:1	1/11/1996 - 3/31/2018
DONKI	2,585	2,544	41	62:1	4/3/2010 - 9/6/2017
Enhanced DONKI	2,394	2,355	39	60:1	4/3/2010 - 9/4/2017

Chapter 4

Forecasting SEP Events using the DONKI CME Catalog with Additional Features

4.1 Problem

The primary task is to predict whether a particular Coronal Mass Ejection (CME) event may be related to an upcoming Solar Energetic Particle (SEP) event. Considering CMEs can cause an elevation in proton flux, their event measurements can be used to forecast the occurrence of future SEP events. We treat the task as a classification problem with two classes: SEP and non-SEP related CMEs. Previously, Torres developed a Neural Network based classifier to address the problem [32]. The data used to train the classifier consisted of the CDAW CME catalog linked with other space weather phenomena, such as Type II radio bursts and sunspot numbers. In order to increase the classifier's performance, we explored two separate areas: the quality of

data and additional machine learning techniques. In order to improve the quality of data, we investigated merging event entries in both DONKI and CDAW CME catalogs in Chapter 3. In this chapter, we improve model performance by additionally looking beyond previously-investigated methods, such as oversampling and classifier score threshold adjustments, to include feature and representation learning.

4.2 Approach

4.2.1 Neural Network Classifier

Similar to the work of Torres [32], we first use a multi-layer perceptron neural network classifier using one hidden layer. Figure 4.1 depicts the architecture of the network using a common configuration of input features. Considering the categorical nature of the model's output, we look to minimize the difference between the observed and predicted distributions. Thus, the cross entropy loss function is used during training:

$$\mathcal{L}_{CE} = -ylog(\hat{y}) - (1-y)log(1-\hat{y}), \tag{4.1}$$

where y is the observed class, and $\hat{y}$ is the predicted score between [0,1]. In this particular problem, an observed value of 0 means the given CME event is not related to an SEP, and a value of 1 meaning the CME is related to an SEP. Figure 4.1 shows the configuration of layers for the multi-layer perceptron neural network classifier.

One key difference from the network used in Torres's work is the number of output units. Since we use the cross entropy loss function, the output of the network represents a probability distribution split amongst the two classes. While semantically the same as the output of Torres's network, this aids us in using additional techniques to increase network performance introduced later in this section.

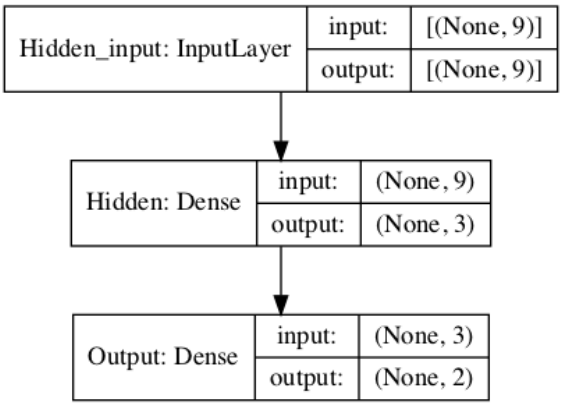

Figure 4.1: Network architecture for the multi-layer perceptron neural network classifier.

4.2.1.1 Feature Sets

Since merging the DONKI and CDAW CME catalogs as outlined in Chapter 3, we grouped together three sets of features. The first, or baseline, set is made of measurements directly from DONKI entries. The second set consists of selected features, both derived from catalog entries and taken from other sources of data. The third and final set of features are the remaining features from the CDAW CME catalog-based model. Specifically, the baseline features consist of the following:

- Linear Speed
- Half Angle
- Latitude
- Longitude

The selected features then consist of the following:

- Daily Sunspot number
- Type II visualization area

- Acceleration
- Peak proton intensity derived by Richardson *et al.* [22]
- Diffusive shock acceleration

Within the selected features, Acceleration and Type II visualization area are borrowed directly and from previous links with CDAW CMEs respectively. Meanwhile, Daily Sunspot number was directly linked to each DONKI entry. Finally, the Peak Proton Intensity and Diffusive Shock Acceleration were directly derived using DONKI CME entry measurements. The procedures used to calculate these two features can be found in Appendix A. The remaining features are broken down into four categories: CME speed, CME size, event location, and CME history. Further, these groups are used in section 4.3.5.1 to understand the importance of similar features relative to each other. These features are grouped as the following:

- Speed
  - 2nd order speed initial
  - 2nd order speed final
  - 2nd order speed at 20 solar radii
- Size
  - Halo
- Location
  - Central Position Angle (CPA)
  - Measurement Position Angle (MPA)

- CME History
  - Maximum speed in the past day
  - Number of CMEs in the past month
  - Number of CMEs in the past 9 hours
  - Number of CMEs with a Linear Speed >1000 km/s in the past 9 hours

The additional features in the Speed, Size, and Location groups come directly from linked CDAW entries, while the CME History features were recomputed using the merged DONKI-CDAW CME dataset.

4.2.2 Handling Imbalanced Data with Separate Feature and Classifier Learning

The imbalanced data solutions explored for this particular problem generally fall into one of two categories: data-based and model-based. The main data-based method employed is a technique called oversampling, where outlier class samples are duplicated in the training set. Typically, we oversample the data to a 3 to 1 negative to positive ratio. This technique has been used successfully in improving the performance of predicting the occurrence of SEP events [32].

For all models presented in this work, we employ some form of oversampling the minority class to hinder the bias imposed by the original distributions during training. This follows the work of Torres [32], who used a 3 to 1 non-SEP to SEP event oversampling ratio. Another technique to emphasize the minority class is to reweight the loss function. Equation 4.1 describes the cross entropy loss function for the classifier. In this case, the first term determines the loss of the case where an event is associated with an SEP, and the second term determines the loss of the case where an event is not

associated with an SEP. In this case, assuming the use of gradient descent with every training sample per training epoch, weighting the first term greater than the second is equivalent to oversampling. However, when using stochastic or mini batch gradient descent, particularly as described in Section 4.3.3.1, the aforementioned statement does not hold. For example, in the scenario where there are 10 samples per batch, the effective number of samples is different between the two techniques. For oversampling with replication, the effective number of samples is 10 for the weight update. However, for the reweighting technique, if one sample in the minority class is weighted as two, the effective number of samples in the batch is 11 for the weight update. Our initial results indicate that oversampling the minority class outperform the reweighting technique. Hence, we only report results using the oversampling technique.

While oversampling improves performance on minority classes, additional model-based methods were investigated to improve overall performance. To learn a better representation of the training data, a technique called classifier re-training (cRT) was first explored [26]. The technique presents two stages of training: first, it trains the entire model on the original training data, without oversampling. The first stage is used to learn better features output from the hidden layer units. Then, as a second stage, the weights between the hidden and input layers are frozen, and the weights between the hidden and output layers are reinitialized and retrained typically using class-balanced data. Given the newly-learned intermediary features output from the hidden layer, the second stage focuses on learning the classifier in the form of the output layer. Figure 4.2 depicts the weights modified during stages one and two.

The network shown in Figure 4.2 contains 9 features, a typical configuration seen in models using baseline and selected features. Additionally, in our case, class-balanced data is a special case of oversampling; for every SEP-related CME, there is exactly one non-SEP-related CME. However, after hyperparameter tuning, we determined that

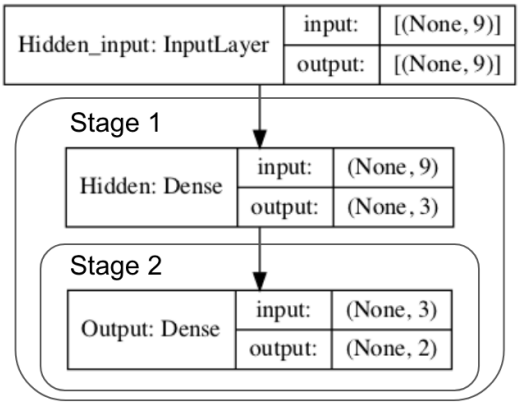

Figure 4.2: Network architecture for the cRT technique, showing the different weights updated during the two stages of cRT.

a 3 to 1 non-SEP to SEP event ratio performed better than typical class-balanced sampling.

Then, to learn better features than those used as inputs, an autoencoder model was investigated. Figure 4.3 depicts the autoencoder configured for the baseline and selected features. The autoencoder consists of two parts: the encoder and decoder. The encoder takes the input and outputs a transformed version, while the decoder takes the same transformed version and reconstructs the input into its original form. In order to train the model to accurately reconstruct the input, the mean square error (MSE) loss function is used:

$$\mathcal{L}_{MSE} = \frac{1}{N}\sum_{n=1}^{N}(\mathbf{x} - \hat{\mathbf{x}})^2 \tag{4.2}$$

Within the MSE loss function, N is the number of training samples, $\mathbf{x}$ is the input vector, and $\hat{\mathbf{x}}$ is the model-reconstructed input vector. The benefit of the autoencoder lies in using the output of the encoder, typically known as the z-layer, as newly-learned features based on the training data.

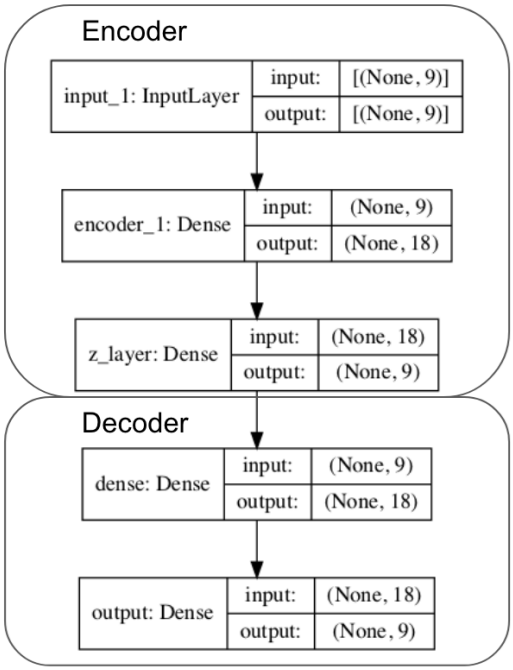

Figure 4.3: Network architecture for the Autoencoder model.

Leveraging the feature learning capabilities of an autoencoder, combined with the data representation and classifier learning ability of cRT, we propose a combination of the two techniques known as cRT+AE. The output of the middle z-layer of the autoencoder is additionally used as input into the output layer of the classifier. A joint loss function is employed, where MSE and cross entropy from the autoencoder and classifier are respectively used:

$$\mathcal{L}_{cRT+AE} = \alpha\mathcal{L}_{CE} + (1-\alpha)\mathcal{L}_{MSE} \tag{4.3}$$

In the joint loss function, α is a tunable parameter within the range of [0,1]. Figure 4.4 shows the resulting example network using the baseline and selected features model. Then, cRT was integrated in a similar fashion; however, instead of using class-balanced data in the second stage of training, a 4 to 1 non-SEP to SEP oversampling

ratio was used. Like regular cRT, all weights are updated using the original training dataset in the first stage. In the second stage, the decoder portion of the autoencoder is discarded, the encoder portion's weights are frozen, and the classifier layer weights are reinitialized and trained using the oversampled training data.

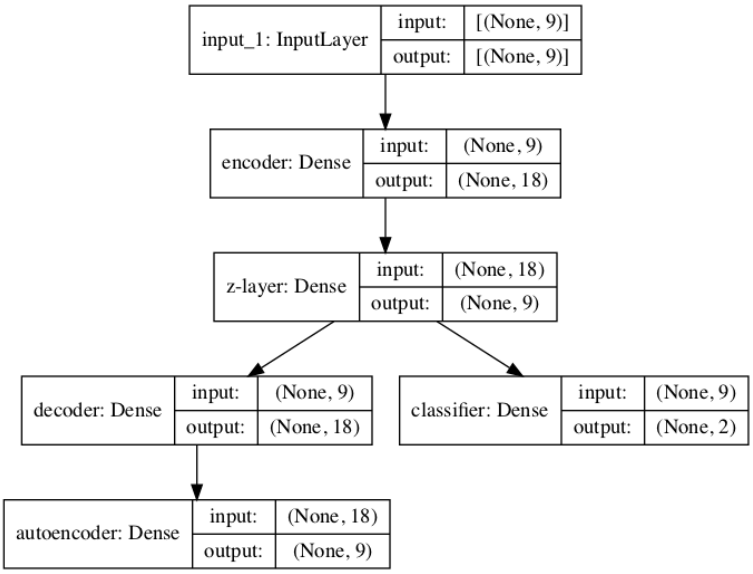

Figure 4.4: Network architecture for the cRT+AE technique.

4.3 Evaluation

4.3.1 Dataset Partitioning

We partition the dataset between train and test sets using two separate schemes: chronological and random. In order to test the operational capability of the model, we first split the data chronologically, where the first 70 percent of CME events are included in the training set, and the remaining 30 percent are used as the test set. This helps simulate "future" predictions using a model trained solely on "past" data. Table 4.1 describes the start and end dates for the train and test sets. While testing the op-

Table 4.1: Details of the chronological data partitioning.

Partition	**Event Count**	**Start Event**	**End Event**
Train	1673	4/3/10 9:54	4/26/15 12:36
Test	720	4/28/15 7:31	9/4/17 19:39

erational capacity of the model is important, the distribution of events associated with SEPs is highly underrepresented in the chronological test set. Of the 39 SEP-related events in the entite dataset, only 6 are in the chronological test set. In order to better distribute the SEP events, 30 percent of each class is randomly chosen for the test set, and the remaining data is assigned to the training set.

4.3.2 Evaluation Metrics

Before showing how we calculate evaluation metrics, Table 4.2 shares a brief overview of useful classification terminology, and how it relates to the main problem.

Table 4.2: The confusion matrix for our classification problem.

Is the CME Related to an SEP Event?	Actual No/Negative	Actual Yes/Positive
Predicted No/Negative	True Negative (TN)	False Negative (FN)
Predicted Yes/Positive	False Positive (FP)	True Positive (TP)

Three intermediary metrics useful for those commonly used to evaluate models include precision, recall, and False Positive Rate (FPR). Precision is defined as the likelihood that an event is actually positive given a positive prediction:

$$precision = \frac{TP}{TP + FP} \tag{4.4}$$

Recall can be described as the True Positive Rate (TPR). Given an event is actually positive, it describes the probability that it will also be predicted positive:

$$recall = TPR = \frac{TP}{TP + FN} \tag{4.5}$$

FPR is described as the likelihood that a negative sample will be predicted as positive:

$$FPR = \frac{FP}{FP + TN} \tag{4.6}$$

The first calculated metric we use is the F_1 score, computed as the equally-weighted harmonic mean of precision and recall:

$$F_1 = 2 * \frac{precision * recall}{precision + recall} \tag{4.7}$$

The True Skill Score (TSS), also known as Hanssen and Kuipers' Discriminant [35], is described as the difference between the true and false positive rates:

$$TSS = TPR - FPR \tag{4.8}$$

The Heidke Skill Score (HSS) is commonly used to determine a classifier's ability to perform better than random predictions [21]. The skill score has a range of $(-\infty, 1]$, where a score between (0,1] are better than randomly making predictions. Those between $(-\infty, 0)$, however, are considered worse than random.

$$HSS = 2 * \frac{TP * TN - FP * FN}{(TP + FP)(FP + TN) + (TP + FN)(FN + TN)} \tag{4.9}$$

Further, we report two additional metrics: Lowest Positive and Highest Negative average ranks (LPR and HNR respectively). Figure 4.5 depicts LPR and HNR. LPR and HNR are computed by ranking each test event's score in descending order, with

the event predicted most likely to be an SEP ranked first. In the perfect scenario, these two numbers are next to each other with an absolute difference of one; in the case of Figure 4.5, this would correspond to an LPR of 3 and HNR of 4. Further, LPR and HNR are not sensitive to the classifier threshold, allowing more general performance measures for the model than the aforementioned ones, whom rely on the threshold for defining classification errors.

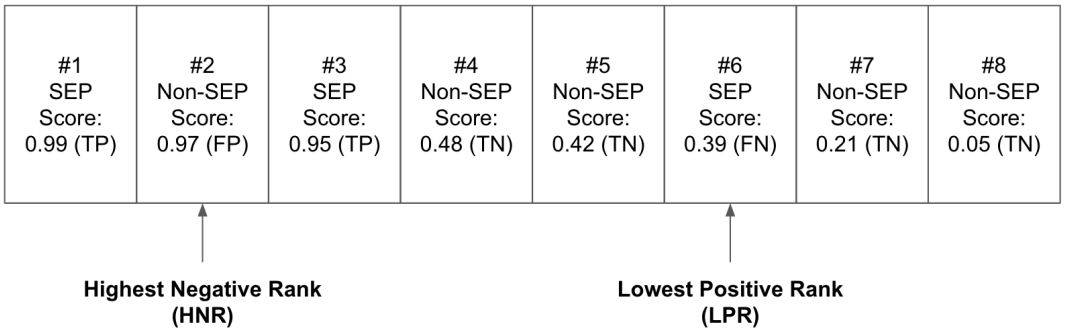

Figure 4.5: Visualization of LPR and HNR. Here, there are 8 events in the test set: 3 are related to SEPs and the other 5 are not. The scores correspond to an HNR of 2 and an LPR of 6.

The goal is to minimize the lowest positive rank, while also maximizing the highest negative rank. By increasing the highest negative rank, we rank more TPs above the worst FP; similarly, when we decrease the lowest positive rank, we increase the number of TNs below the worst FN. Further, as the ranks approach their ideal cases, it becomes more meaningful to adjust the score threshold to optimize the other metrics.

Another important attribute of HNR and LPR is their threshold-independent nature, which allows us to evaluate the model past its classifier capability. A more well-known threshold-independent metric is known as the area under the Receiver Operating Characteristic (ROC) curve, commonly known as AUC. Typically, better performance

corresponds to an increase in AUC. However, we do not use AUC in this work because the ROC curve covers FPR from 0 to 1; however, we only want to consider models that incur a small FPR.

4.3.3 Parameters and Procedures

4.3.3.1 Model Hyperparameters

All variations of the neural network classifier were implemented using Keras [7]. The Multilayer Perceptron Neural Network used the Sequential model, whereas cRT and cRT+AE were implemented by subclassing the Model class to cleanly transfer weights between training stages. Additionally, the sigmoid function was used as the activation function for all layers. The base model contains three units in the hidden layer and two in the output layer. Although Torres's model uses 30 hidden units, we found that three units maximized metrics in this case. Prior to training, weights are initialized using the Glorot uniform initializer. Weights are updated using mini batch gradient descent with 200 samples per mini batch. We additionally set momentum to 0.9 and use L2 weight regularization set to 0.0075. The model is then trained over 400 epochs.

The cRT model contains three units in the hidden layer and two in the output layer. Prior to training, weights are initialized using the Glorot uniform initializer. Weights are updated using mini batch gradient descent with 200 samples per mini batch. We set momentum to 0.9 and use L2 weight regularization set to 0.0075. During the first stage, the model is then trained over 400 epochs, without any oversampling in the training data. Then, during the second stage, the weights between the input and hidden layers are frozen, and the weights between the hidden and output layers are reinitialized using the Glorot uniform initializer. Finally, these weights are retrained over 400 epochs, using a 3 to 1 non-SEP to SEP oversampling ratio in the training set.

For the cRT+AE model, the autoencoder follows a n/2n/n/2n/n architecture, where n is the number of features. To ensure the autoencoder does not converge to the identity function [14], L2 weight regularization is utilized and is set to 0.0075. The weights connecting the z-layer of the autoencoder to the classifier use the same form of regularization. Prior to training, all weights are initialized using the Glorot uniform initializer. Mini batch gradient descent is once again used with 200 samples per mini batch. We additionally set momentum to 0.9. During the first stage, the model is then trained over 2000 epochs without any oversampling in the training data. The α in the loss function is set to 0.4, weighting the classifier more than the autoencoder. During the second stage, the decoder portion of the autoencoder is discarded, the encoder weights are frozen, and the weights between the z-layer and classifier layer are reinitialized using the Glorot uniform initializer. Finally, these weights are retrained over 400 epochs, using a 4 to 1 non-SEP to SEP oversampling ratio in the training set.

4.3.3.2 Data Procedures

Before model evaluation, the features are normalized between 0 and 1. This is performed by taking the difference between the individual values and the minimum value of the feature, divided by the difference between the maximum and minimum values of the feature. In cases where the difference between the minimum and maximum values of a feature are over three orders of magnitude, the log is taken before normalization. These features include Diffusive Shock and Type II Area Visualization.

There are a number of samples that contain missing data on a feature basis. For the baseline features, all missing values are assigned the feature's median value. Samples not related to a Type II event are initially given a 0 value for Type II Area Visualization. Considering all values of the feature are either 0 or a large positive integer greater than 1, all samples without Type II events retain their feature value of 0. There are

additional details on how missing data is handled for CDAW CME Entries detailed in Torres's work [32].

4.3.3.3 Evaluation Procedures

Each model configuration's results are created using the average of five independent experiments. Further, the same random seed is used each time for the random data partitioning to compare results.

4.3.4 Results

First, in Section 4.3.4.1 we incrementally add feature sets to the base multilayer perceptron neural network. Then, in Section 4.3.4.2, we use cRT and cRT+AE with the previously-added feature sets. Finally, in Section 4.3.4.3, we modify the classifier thresholds in order to maximize the F1 score using all features for both chronological and random data partitioning scenarios.

4.3.4.1 Varying Feature Sets

As described in Section 4.2.1.1, we first use only baseline features, then add in selected features, followed by all features. Table 4.3 displays the results using chronologically-partitioned data, whereas Table 4.4 shows results using randomly-split data. For each metric column, the best results are underlined.

Table 4.3: Results from adding sets of features to the base model using chronological data partitioning.

Features	**TSS**	**HSS**	**F1**	**LPR**	**HNR**	**TN**	**FP**	**FN**	**TP**
Baseline	0.6594	0.5215	0.5263	194.8	1.0	708.8	5.2	2.0	4.0
Baseline + Selected	<u>0.6613</u>	<u>0.5757</u>	<u>0.5797</u>	112.4	<u>2.0</u>	710.2	3.8	2.0	4.0
Baseline + All	0.6608	0.5591	0.5634	<u>89.6</u>	1.8	709.8	4.2	2.0	4.0

Table 4.4: Results from adding sets of features to the base model using random data partitioning.

Features	**TSS**	**HSS**	**F1**	**LPR**	**HNR**	**TN**	**FP**	**FN**	**TP**
Baseline	0.7123	0.3622	0.3782	69.4	4.2	679.4	26.6	3.0	9.0
Baseline + Selected	0.8544	0.4805	0.4930	59.8	8.6	685.6	20.4	1.4	10.6
Baseline + All	0.8327	0.5840	0.5930	64.0	9.0	693.8	12.2	1.8	10.2

As seen throughout both partitioning schemes, TSS, HSS, and F1 improve when additional features are added. Although the configuration with all features does not always improve metrics the best, this is expected, since the selected features are a subset that we believe help differentiate SEP-related from non-SEP-related CMEs. Considering Diffusive Shock Acceleration is derived using a CME event's speed, some form of Speed, Type II Radio Event occurrence, and Richardson's Peak Proton Intensity in the selected subset were found to differentiate these events by Richardson *et al.* [22]. In addition, LPR and HNR improve, suggesting that there may be benefit to adjusting the score threshold in order to further maximize the other metrics.

4.3.4.2 Using cRT and AE to Handle Imbalanced Data

In this section, we present the results of adding in additional techniques to handle the data imbalance, including classifier re-training (cRT) and the autoencoder (AE). For each table, we report results for models using baseline with both selected and all features, based on their positive results demonstrated in the previous section.

Table 4.5 shows the results of adding in cRT and AE to models using the baseline+selected (baseline+5) and baseline+all feature set configurations when the events are partitioned between training and test chronologically. For each metric column, the best results are underlined. In this case, although the models without cRT and AE produce the best LPR, adding in cRT and cRT+AE tend to increase F1 and the skill

scores.

Table 4.5: Results from adding in techniques to handle the data imbalance beyond oversampling to the model using chronological data partitioning.

Features/ Algorithm	**TSS**	**HSS**	**F1**	**LPR**	**HNR**	**TN**	**FP**	**FN**	**TP**
Baseline+5	0.6613	0.5757	0.5797	112.4	2.0	710.2	3.8	2.0	4.0
Baseline+5/ cRT	0.6619	0.5933	0.5970	125.0	2.0	710.6	3.4	2.0	4.0
Baseline+5/ cRT+AE	0.6611	0.5673	0.5714	286.2	2.8	710.0	4.0	2.0	4.0
Baseline+all	0.6608	0.5591	0.5634	89.6	1.8	709.8	4.2	2.0	4.0
Baseline+all /cRT	0.5936	0.5023	0.5070	170.0	1.6	709.4	4.6	2.4	3.6
Baseline+all /cRT+AE	0.8263	0.6211	0.6250	243.4	1.2	709.0	5.0	1.0	5.0

Table 4.6 shows the results of adding in cRT and AE to models using the baseline+selected (baseline+5) and baseline+all feature set configurations when the events are partitioned between training and test randomly. For each metric column, the best results are underlined. Similar to the results using chronological partitioning, adding in cRT and cRT+AE increase F1 and the skill scores under different feature configurations.

With the increased number of SEP-related events in the test set using random partitioning, it is more meaningful to compare the Lowest Positive and Highest Negative rank changes. Although using cRT and cRT+AE both increase TSS, HSS, and F1 in various configurations at a score threshold of 0.5, they typically vary in rank improvement much more than the original single-stage model.

4.3.4.3 Varying Classifier Thresholds to Maximize the F1 Score

To additionally optimize metrics, we vary the output score threshold that determines how an event is classified. Typically, the threshold is set to 0.5: if the output score of

Table 4.6: Results from adding in techniques to handle the data imbalance beyond oversampling to the model using random data partitioning.

Features/ Algorithm	**TSS**	**HSS**	**F1**	**LPR**	**HNR**	**TN**	**FP**	**FN**	**TP**
Baseline+5	0.8544	0.4805	0.4930	59.8	8.6	685.6	20.4	1.4	10.6
Baseline+5/ cRT	0.8095	0.5040	0.5155	73.8	8.6	689.2	16.8	2.0	10.0
Baseline+5/ cRT+AE	0.8748	0.5207	0.5320	111.8	5.0	688.2	17.8	1.2	10.8
Baseline+all	0.8327	0.5840	0.5930	64.0	9.0	693.8	12.2	1.8	10.2
Baseline+all /cRT	0.8680	0.6267	0.6347	49.8	8.6	695.2	10.8	1.4	10.6
Baseline+all /cRT+AE	0.8025	0.6082	0.6164	65.4	4.0	696.0	10.0	2.2	9.8

the network is above, then we predict it is related to an SEP; otherwise, we predict the event is not related to an SEP. Here, we optimized F1 through threshold adjustments. Using all features, the results improve, as seen in Tables 4.7 and 4.8, reporting results for chronological and random data partitioning respectively. For each metric column, the best results are underlined.

Table 4.7: Results of varying the classifier threshold to maximize the F1 score using all features and chronological data partitioning.

Model	**Threshold**	**TSS**	**HSS**	**F1**	**FP**	**FN**
Single Stage	0.7793	0.6625	0.6119	0.6154	3.0	2.0
cRT	0.2016	0.8235	0.5506	0.5556	7.0	1.0
cRT+AE	0.8495	0.8291	0.7115	0.7143	3.0	1.0

When we previously adjusted the threshold to increase a certain metric, it typically decreased other metrics. Opposite from previous results, when we now modify the threshold to maximize the F1 score, we see an additional increase in TSS and HSS. In the case of cRT+AE using chronological partitioning, the two were originally 0.8263 and 0.6211 with a threshold of 0.5.

Table 4.8: Results of varying the classifier threshold to maximize the F1 score using all features and random data partitioning.

Model	**Threshold**	**TSS**	**HSS**	**F1**	**FP**	**FN**
Single Stage	0.9783	0.6667	0.7973	0.8000	0.0	4.0
cRT	0.9652	0.6667	0.7973	0.8000	0.0	4.0
cRT+AE	0.5053	0.8206	0.6377	0.6452	9.0	2.0

Similar to the chronological data partitioning results, when using the cRT+AE model, TSS and HSS also increased from 0.8025 and 0.6082 respectively. The variability demonstrated amongst different configurations of cRT and cRT+AE models for Lowest Positive and Highest Negative rankings have clear effects on the optimized score thresholds. However, although cRT+AE has the worst F1 and HSS, it reduces the number of false negatives in half compared to the original single-stage and cRT models. Considering the severe consequences of false negatives in our problem, and the single digit number of false positives, the cRT+AE model remains the best choice.

4.3.5 Analysis

In this section, we take a closer look into the cRT+AE models using all features. In Section 4.3.5.1, we identify important features in two separate ways: by correlating feature value ranges with output scores, and by ranking features. In section 4.3.5.2, we discuss common errors made by the models.

4.3.5.1 Feature Importance

To better understand how the trained models perform given the ranges for different features, and to additionally observe any correlations between individual features and output score, we visualize each test set sample's predicted score against each individual feature. Figures 4.6 and 4.7 depict these scatter plots of the baseline and selected

features for the cRT+AE model using all features. Within the baseline features, there is a clear increase in score when events have a speed of 1000 km/s or greater or a half width of 40 degrees or higher. Considering CME speed, and half width to a lesser extent, have previously been used as a proxy to determine its association with an SEP event by Richardson *et al.* [22], the correlations shown through these plots validate the model's behavior. Further, within the selected features, an increase in both Type II area and Richardson's Formula output lead to an increase in score. Both have previously also been used to predict whether a CME was related to an SEP: we previously successfully used Type II area as a feature, and Richardson's formula previously was used to predict the peak proton intensity after a given CME event.

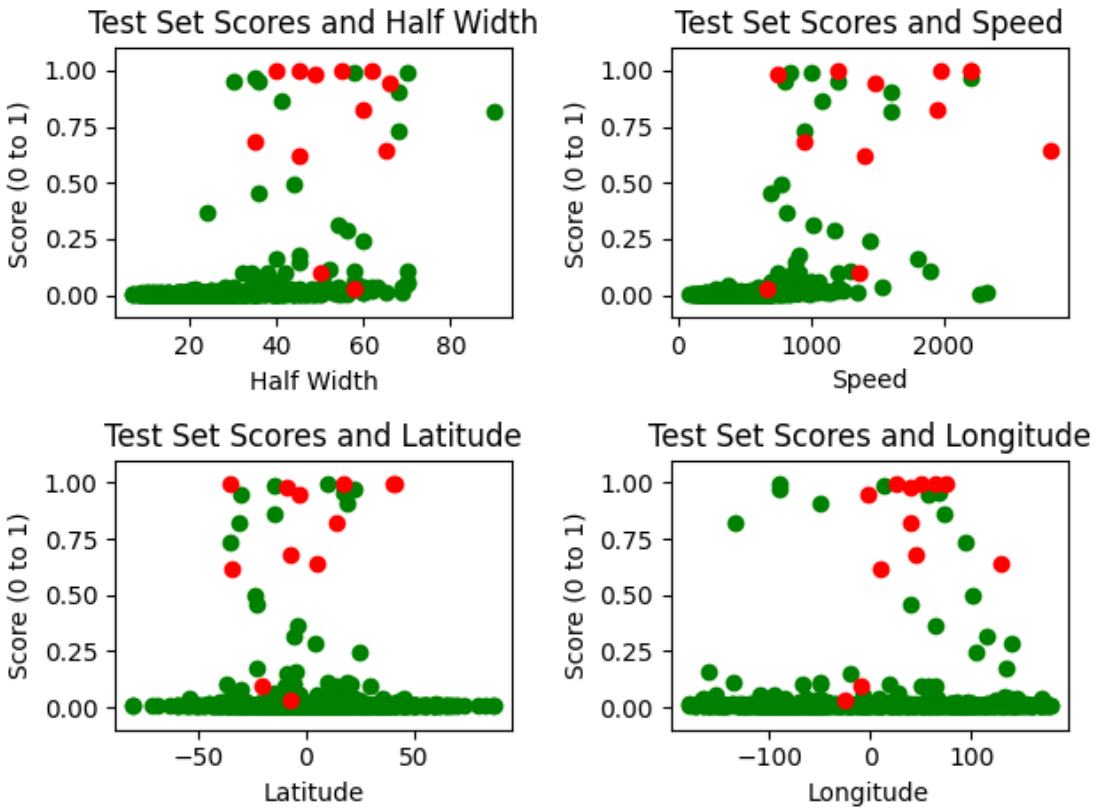

Figure 4.6: Individual scatter plots of each individual baseline feature. Red dots are SEP-related events, and green dots are non-SEP-related events.

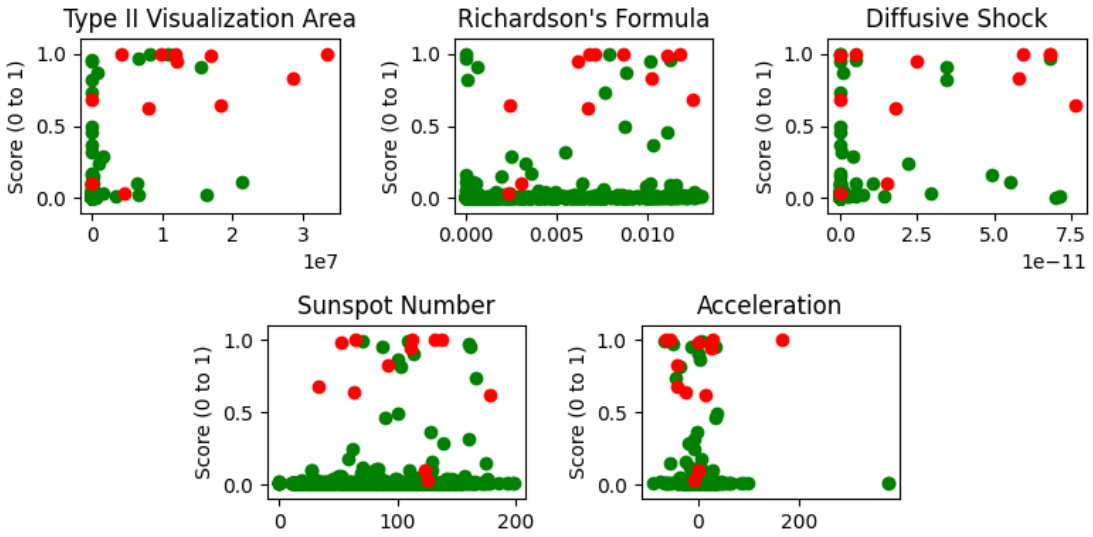

Figure 4.7: Individual scatter plots of individual selected features. Red dots are SEP-related events, and green dots are non-SEP-related events.

We additionally observe how certain features may impact model predictions by using techniques to rank all features from most to least important. First, we use the method proposed by Torres [32]. Table 4.9 shows the output of this method for the cRT+AE model using all features trained using the random partitioning scheme. Interestingly, sunspot number and acceleration are two of the top five features in Torres's work.

Another Feature Importance method we explored is called Local Interpretable Model-agnostic Explanations (LIME), developed by Ribeiro *et al.* [30]. Please refer to Chapter 2 for the details on how feature importance is calculated using this method. In addition to what Ribeiro *et al.* present, we normalize the feature importance vector between [0,1] to compare with Torres's method's results. The major difference between the two is that our method is model-based, whereas LIME is based on the collective explanations of every instance in the training set. Table 10 contains the results of this feature importance method for the cRT+AE model using all features trained using the

Table 4.9: All features ranked in descending order of importance using the method developed by Torres. This ranking was compiled after taking the average importance per feature over 5 independent runs.

Feature	**Importance**
1. Acceleration	0.0616
2. 2nd order speed final	0.0593
3. Diffusive Shock	0.0582
4. Latitude	0.0582
5. Sunspot Number	0.0578
6. DONKI Linear Speed	0.0566
7. CMEs in past 9 hours	0.0542
8. DONKI Half Width	0.0539
9. CPA	0.0535
10. CMEs in past month	0.0530
11. 2nd order speed at 20 solar radii	0.0523
12. Richardson's Formula	0.0518
13. MPA	0.0501
14. Longitude	0.0501
15. 2nd order speed initial	0.0488
16. Halo	0.0486
17. Max speed past day	0.0465
18. Type II Visualization Area	0.0436
19. CMEs over 1000 km/s past 9 hrs	0.0419

random partitioning scheme.

We compare the Feature Importance methods using two techniques: through comparing feature differences within ranges of rankings, and by ranking groups of features. First, we compare the methods by splitting the ranked features into three groups: 1-6 are high-ranking, 7-13 are mid-ranking, and 14-19 are low-ranking. Within high-ranking features, only Diffusive Shock, 2nd order speed final, and Sunspot Number remain across both methods. For mid-ranking features, only CPA and 2nd order speed at 20 solar radii are consistent. Finally, for low-ranking features, only Max speed past day, Type II Visualization Area, and the number of CMEs over 1000 km/s in the past 9 hours are similar. Although these rankings differ, it should be noted again that both

Table 4.10: Feature importance using LIME. This ranking was compiled after taking the average importance per feature over 5 independent runs.

Feature	**Importance**
1. Richardson's Formula	0.0924
2. Diffusive Shock	0.0910
3. 2nd order speed final	0.0740
4. Sunspot Number	0.0729
5. CMEs in past 9 hours	0.0718
6. DONKI Half Width	0.0645
7. Halo	0.0579
8. Latitude	0.0558
9. CPA	0.0556
10. Longitude	0.0427
11. 2nd order speed initial	0.0425
12. 2nd order speed at 20 solar radii	0.0405
13. Acceleration	0.0404
14. DONKI Linear Speed	0.0401
15. Type II Visualization Area	0.0398
16. MPA	0.0351
17. Max speed past day	0.0340
18. CMEs in past month	0.0255
19. CMEs over 1000 km/s past 9 hrs	0.0234

methods are designed differently: Torres's method is model-based, whereas LIME is instance-based. Further, simply comparing feature rankings in bins may additionally disregard how the features themselves are related. This gives motivation for introducing an additional technique to compare feature importance methods: grouping together all related features, sum their importances, and rank the groups of features accordingly. There are five primary groups: Speed, Location, Size, CME History, and Other. The Speed group contains the following features:

- DONKI Linear Speed
- Diffusive shock acceleration
- 2nd order speed final

- 2nd order speed initial
- 2nd order speed at 20 solar radii

The Location group contains features related to the physical origin of the CME event, and any derived features:

- Latitude
- Longitude
- Peak Particle Intensity formula (Richardson *et al.*)
- CPA (weighted by 0.5)
- MPA

It is important to note that CPA is weighted by 0.5, because both location and event size data is encoded in the measurement. As mentioned in the description of the CDAW catalog [1], an event's CPA is the counter clockwise angle between the effective north pole of the Sun and the center of the CME as viewed from Earth. However, CMEs large in size often appear as a "Halo" around the Sun from Earth; these events are given a CPA of 360. The Size group relates to features conveying the width of the CME event:

- CPA (weighted by 0.5)
- Halo
- DONKI Half Width

The CME History group is made up of features that are measured using previous CMEs. These features are related to detecting double CME events, where more than one CME cause an elevation in proton intensity:

- Maximum speed in the past day

- Number of CMEs in the past month
- Number of CMEs in the past 9 hours
- Number of CMEs with a Linear Speed >1000 km/s in the past 9 hours

Finally, the Other group consists primarily of other space weather phenomena:

- Acceleration
- Type II Visualization Area
- Sunspot Number

Tables 4.11 and 4.12 present the results of grouping features as a means of comparing feature importance methods. As expected, the rankings are the same, with speed at the top for both. Additionally, it is unsurprising that Location ranks second, considering the Peak Particle Intensity formula from Richardson *et al.* [22] is included in the group. Further, although the summed importance for each group differs between methods, this is expected because both are implemented differently.

Table 4.11: The Feature Group Importance for Torres's method, ranked in descending order.

Group	**Combined Importance**
1. Speed	0.2752
2. Location	0.2370
3. CME History	0.1956
4. Other	0.1630
5. Size	0.1293

4.3.5.2 Common Errors

To better understand classification errors, we plot test set errors on scatter plots where each axis is a feature and events are color-coded based on score. Figure 4.9 depicts

Table 4.12: The Feature Group Importance for the LIME method, ranked in descending order.

Group	Combined Importance
1. Speed	0.2881
2. Location	0.2538
3. CME History	0.1547
4. Other	0.1531
5. Size	0.1502

these plots for the cRT+AE model using all features for random data partitioning.

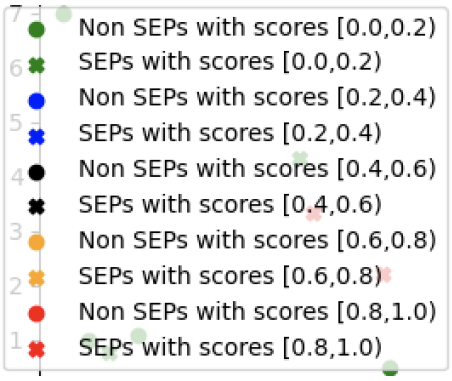

Figure 4.8: The legend for Figures 4.9 and 4.12.

The false negatives, depicted by green cross marks in Figure 4.9, are listed in Table 4.13.

Table 4.13: The false negatives typically seen when using the cRT+AE model with all features and the random data partitioning technique.

Event	Score	Notes
6/14/12 14:09	0.0981	Energetic Storm Particle (ESP) event
4/11/13 7:36	0.0318	Speed might be wrong (1150 instead of 675 km/s)

The event on June 14th, 2012 is considered an ESP event, where the source of the

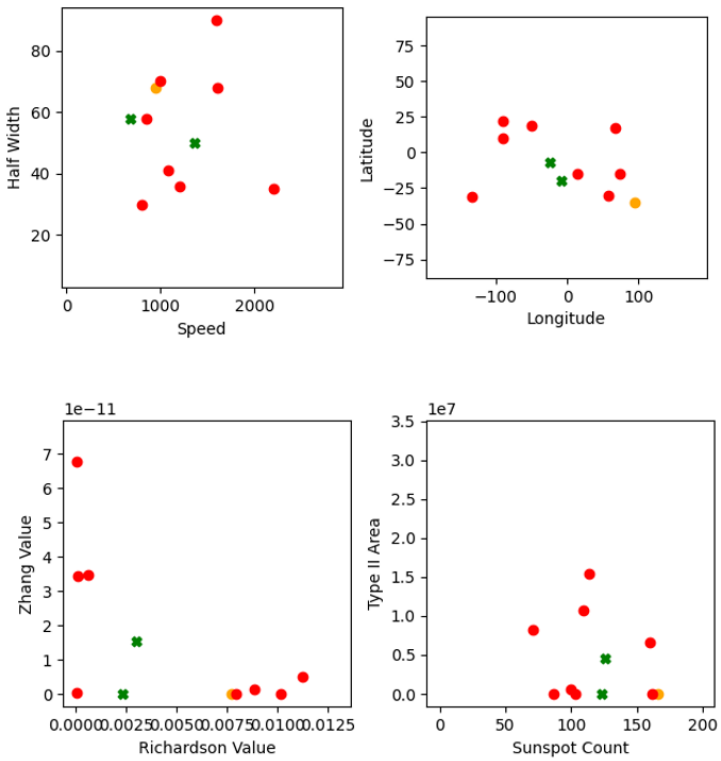

Figure 4.9: On the top left, linear speed and half width are paired; similarly, on the top right, longitude and latitude are paired. On the bottom left, the Richardson Formula and Diffusive Shock are paired; on the bottom right, the Sunspot Number and Type II Visualization Area features are paired. Refer to Figure 4.8 for the legend.

CME is close to a latitude and longitude of zero, and the elevation in proton flux occurs over a long period of time. Figure 4.10 depicts the PHTX file from CDAW for the time period of this event. In the case of this particular event, it takes almost a full 24 hours for the proton flux to cross the 10 pfu threshold.

The common false positives are shown as red and orange circles in Figure 4.9. The clearest observation from these plots can be seen on the top left, with CME half width and speed. All false positives have a speed greater than the lowest speed false negative. Considering the association between many high-speed CMEs and SEP events, at first glance one may presume that speed drove up their predicted scores. The false positives

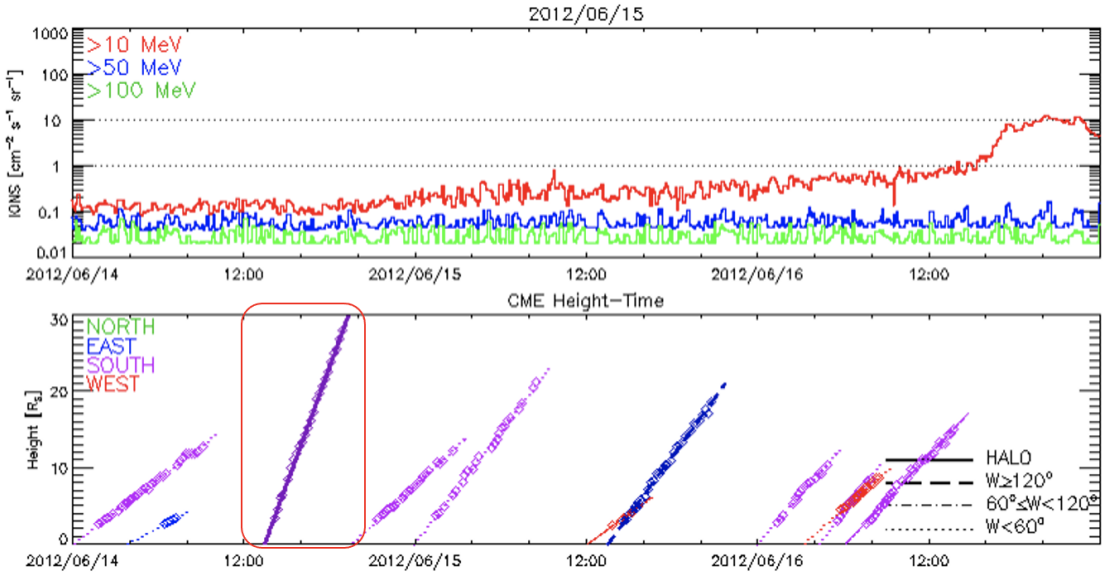

Figure 4.10: PHTX file from CDAW depicting proton flux at various energy levels (top plot), along with CMEs (middle plot) over June 14th-16th, 2012. The CME in question is circled in red, starting around 12:00 on June 14th [10].

also have a half width greater than 30 degrees; again, this is unsurprising given our previous analysis of the relationship between medium to large CME event half widths and SEP events. Additionally, Those nine false positives are listed in Table 4.14 with their scores and notes. An interesting observation is the high number of false positives that are associated with some form of elevated proton intensity. When we observe protons at 10 MeV, an SEP is defined when the proton intensity becomes 10 pfu or higher. Although these CMEs are associated with some form of elevation in proton intensity above background noise at 10 MeV, they do not meet or cross the 10 pfu threshold. For example, Figure 4.11 shows the PHTX file from December 28th, 2015, showing a peak proton intensity of around 3 pfu for 10 MeV. Additionally, the only false positives that do not generate an elevation in proton intensity have high values of important features, previously shown across both feature importance methods in Tables 4.11 and 4.12.

Table 4.14: The false positives typically seen when using the cRT+AE model with all features and the random data partitioning technique.

Event	Score	Notes
9/22/11 11:24	0.9947	Elevated Proton Flux
12/28/15 12:39	0.9913	Elevated Proton Flux
5/14/13 1:30	0.9696	Elevated Proton Flux
5/15/16 15:36	0.9548	Elevated Proton Flux
11/11/11 7:09	0.9512	High Predicted Peak Proton Flux Value
4/2/14 13:55	0.9071	High Speeds
8/17/13 19:24	0.8647	Elevated Proton Flux
12/26/13 3:40	0.8191	Elevated Proton Flux
11/7/13 0:00	0.7351	Elevated Proton Flux

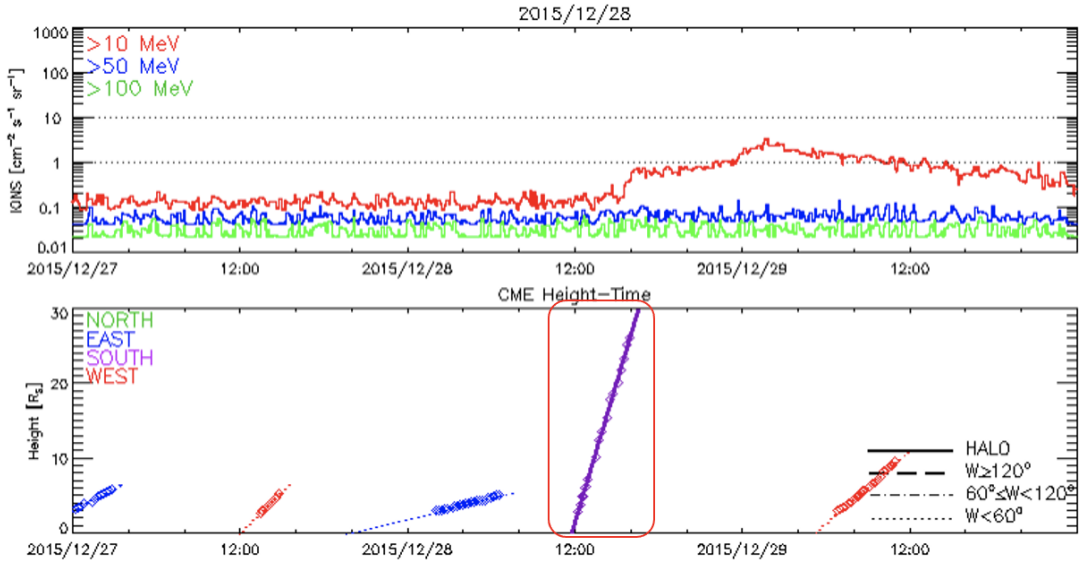

Figure 4.11: PHTX file from CDAW depicting proton flux at various energy levels (top plot), along with CMEs (middle plot) over December 27th-29th, 2015. The CME in question is circled in red, starting around 12:00 on December 28th [9].

We additionally present common mistakes made using the chronological data partitioning scheme. Figure 4.12 shows the 4 error plots with paired up features. Similar to the random partition common errors, all false positives have a high speed and large

half width. Further, there is a false negative with a speed of 300 km/s, an outlier for an SEP. All false positives additionally are associated with a Type II radio burst, and are predicted to have a high peak proton intensity (Richardson Value). The high Richardson Value stems from the location of the false positives, because the Richardson Value directly depends on the distance between the event and the 43 degree longitude line.

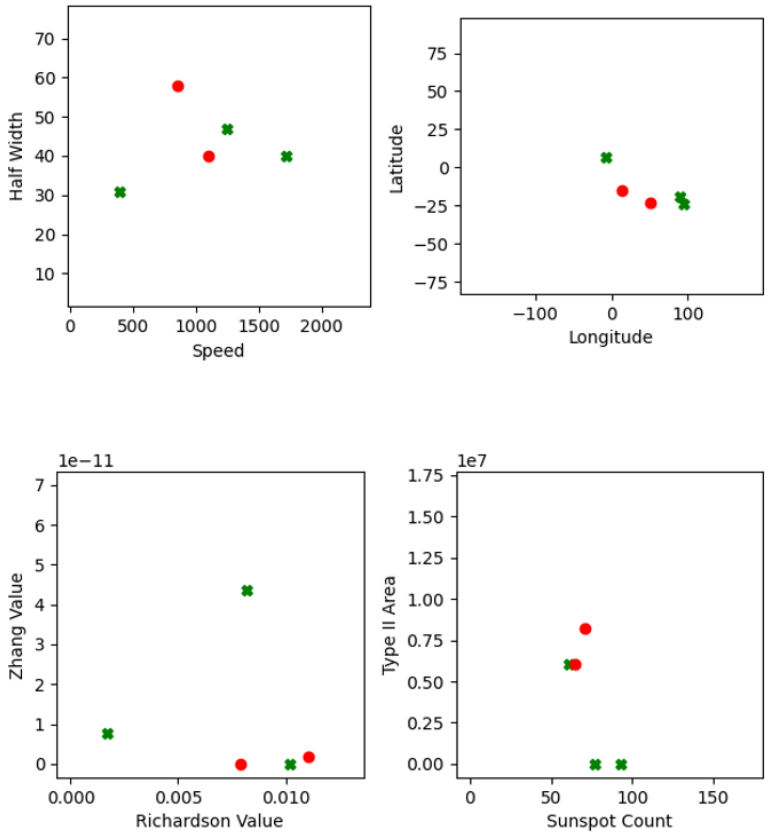

Figure 4.12: On the top left, linear speed and half width are paired; similarly, on the top right, longitude and latitude are paired. On the bottom left, the Richardson Formula and Diffusive Shock are paired; on the bottom right, the Sunspot Number and Type II Visualization Area features are paired. Refer to Figure 4.8 for the legend.

After looking over the common errors, similar to the random partition results, we make conclusions for each event. Table 4.15 shows the most common false negatives produced, along with some additional context for each. Unlike the random partition

results, there are three types of false negatives: ESP events, SEP events that barely crossed the 10 pfu threshold, and SEP events associated with two or more CMEs.

Table 4.15: The false negatives typically seen when using the cRT+AE model with all features and the chronological data partitioning technique.

Event	**Score**	**Notes**
6/18/15 1:25	0.3430	Flux Barely Crossed the Threshold
6/21/15 2:48	0.0758	ESP Event
10/29/15 2:48	0.0021	Double CME

Although the speed for the SEP-related CME on June 18th, 2015 was around 1720 km/s, numerous other factors likely contributed to its classification as a non-SEP event. The event had a recorded CPA of 279; in other words, it was not large enough to constitute as a halo event. Further, the half width of the event was 40 degrees, which we consider to be on the border of being considered a large event in size. Figure 4.13 shows the proton flux level for the event along with the CME associated with the SEP event. The proton flux barely crosses the 10 pfu threshold. By June 19th, the level was back below 10 pfu.

Another event of interest is the CME on October 29th, 2015. This event has a recorded linear speed of 390 km/s, and stops before approaching 20 solar radii. However, there was another CME event 2 hours prior, within the same geographic vicinity, that had a linear speed of 402 km/s. Figure 4.14 shows the two CME events in question; visually, they occur within the geographic vicinity of each other. Further, the remnants of the first event can be seen slightly below the second CME in the coronagraph on the right. This scenario is known as a Double CME event, which may have the potential to cause an SEP event from the output of several smaller CMEs within the vicinity of each other. In order to detect double CME events, we provide the models with context in the form of the CME History features. Figure 4.15 shows

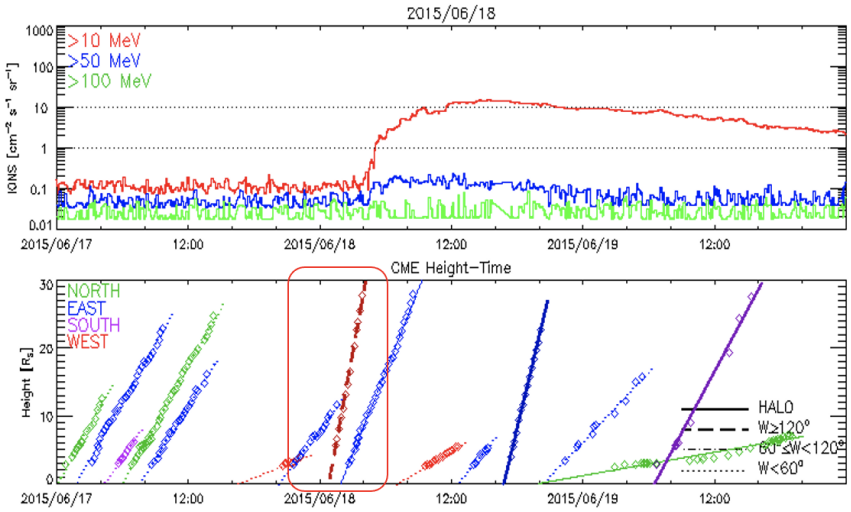

Figure 4.13: PHTX file from CDAW depicting proton flux at various energy levels (top plot), along with CMEs (middle plot) over June 17th-19th, 2015. The CME in question is circled in red, starting at the beginning of June 18th [11].

the CME event of interest, along with the corresponding proton flux levels.

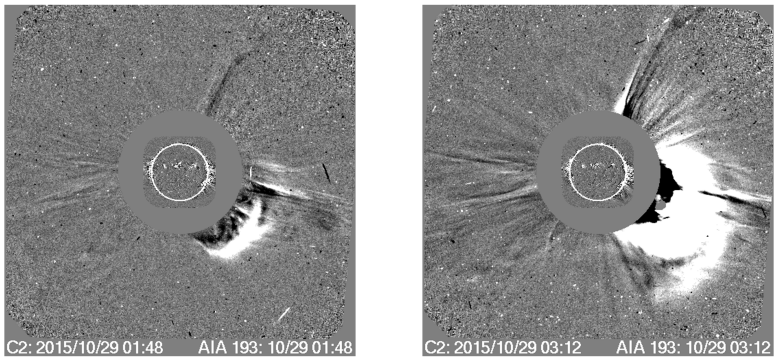

Figure 4.14: Coronagraph captures of a Double CME Event. The capture on the left was taken at 1:48 on October 29th [2], showing the first CME starting at 1:25. The capure on the right was taken at 3:12 on October 29th [3], showing the second CME starting at 2:48.

Table 4.16 shows the most common false positives produced, with additional com-

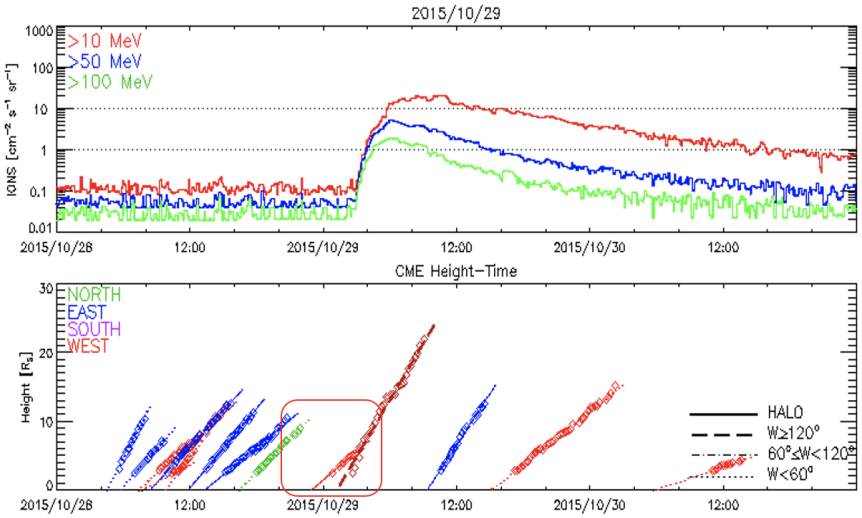

Figure 4.15: PHTX file from CDAW depicting proton flux at various energy levels (top plot), along with CMEs (middle plot) over October 28th-30th, 2015. The CMEs in question are circled in red, starting around the beginning of October 29th [8].

ments. Interestingly, all false positives within the chronological partitioning scheme experiments correspond to an elevation in proton flux. We suspect that the majority of false positives with elevated proton flux events may correspond to the addition of feature learning into our model. Previously, false positives generally were associated with large speeds. Although these two events had linear speeds of 850 and 1100 km/s, some false positive mistakes made by the single-stage model, using the same data partitions, included events with speeds over 2000 km/s. This suggests that the cRT+AE model may place more weight on non-speed based features compared to the single stage model.

Table 4.16: The false positives typically seen when using the cRT+AE model with all features and the chronological data partitioning technique.

Event	**Score**	**Notes**
12/28/15 12:39	0.9675	Elevated Proton Flux
9/20/15 18:12	0.9672	Elevated Proton Flux

Chapter 5

Conclusion

5.1 Summary of findings

In this work, we address the problem of forecasting the occurrence of upcoming SEP events through measurements of CMEs and related space weather phenomena. We introduce enhancements to the DONKI CME catalog through adding additional CME measurements from the CDAW CME catalog along with other space weather phenomena, such as Sunspot Number and Type II Radio Wave Bursts, previously linked together by Torres [32]. While smaller in number of total events compared to both DONKI and CDAW CME catalogs, the enhanced dataset includes measurements from instruments orbiting Earth and the Sun. Additionally, the imbalance between SEP and non-SEP related CMEs decreased compared to both original DONKI and CDAW CME catalogs. Using the enhanced CME dataset, we create models to predict the occurrence of future SEP events. We introduce the cRT+AE technique to learn better features, thereby mitigating the data imbalance, and compare it to using only oversampling SEP-related events. We show the advantage of using cRT+AE across different input feature sets and data partitioning schemes. Further, we analyze the model using

all input features and the cRT+AE method through two separate feature importance techniques. We find that, using both methods, the ranking of feature groups are the same. Finally, we discuss common types of errors, both false positives and negatives, made by the cRT+AE all-feature model. These errors include elevated proton flux levels seen within false positives, and associations with ESP and Double CME events within false negatives.

5.2 Limitations and possible improvements

Currently, we only address predicting the occurrence of SEPs using CME and other space weather phenomena measurements. Estimating future peak proton flux using these measurements is a separate, albeit related, problem. Using machine learning techniques, this may be modeled using regression. Data from GOES Energetic Particle Sensor [6] may be useful in replacing discrete target values used throughout this work with continuous-valued particle flux measurements to create the regression model. Further, a regression-based method may also double as an SEP event classifier by thresholding its output at 10 pfu.

As of this writing, the CDAW CME catalog has a near 5 month time delay in providing CME event measurements. Further, Type II Radio Bursts are seemingly delayed by a year, and are only officially linked to CDAW CME events [13]. In their current form, the models introduced in this work would not be able to operate under near real-time conditions due to these significant delays in publishing measurements. One potential improvement would be to add these borrowed measurements when recording CME events in the DONKI Catalog. This may be simple to implement, considering DONKI often already uses the same Earth-based instrument that CDAW uses to record CMEs. Another possibility to enable real-time use of these models would be to automate the

analysis of CMEs through computer vision techniques. Instead of having a monolithic model that uses coronagraph images as input and SEP occurrence as output, individual models could be developed to estimate measurements such as height-time and width, crucial for calculating the features used in this work. Therefore, one could still apply the models developed in this work in order to preserve consistency with the physical understanding of space weather phenomena, and their relationship to SEP events [27].

We find that our cRT+AE method results in a general increase in performance over only using oversampling to handle the class imbalance inherent to the natural occurrence of SEP events. However, the results are mixed when compared based on rank-based metrics, namely HNR and LPR. Although the cRT+AE method is used to learn better features than those provided as input, using a supervised contrastive learning based approach [34] may learn features that help discriminate between the classes.

The primary way we interpret our models is through ranking features, and subsequently feature groups, against each other. This assumes that certain features are more important than others. While this may be true, the associations between features, and how they produce predictions, may be equally important. Currently, we achieve this implicitly through plotting event scores against pairs of features. In order to provide a more explicit explanation, symbolic regression techniques should be explored [19] [31] [33].

As identified in Section 4.3.5.2, there are two main classes of False Negatives made by our proposed model: ESP-related events, and Double CME events. Typical characteristics of ESP events include an event location close to the geographical origin of the Sun and potentially taking days for the proton flux to rise above 10 pfu near Earth. Using our modification to the formula proposed by Richardson *et al.* [22] would likely produce a low proton flux estimate, as the origin is 43 degrees away from the

formula's peak output. Considering the output of this formula is grouped into the location feature group, and that the location group ranks second out of the five groups, one possible explanation for missing SEP-related ESP events is a mismatch between previously perceived versus additionally important locations. In order to mitigate this, additional features that separate SEP-related ESP events from normal CME events should be explored. For the second class of False Negatives, we already include the CME History feature group in an attempt to catch Double CME events. Part of the problem is the majority of Double CMEs, as we attempt to identify them through features, are not related to SEP events. Further, a dilemma arises regarding finding Double CMEs. For example, our proposed enhanced dataset may exclude certain DONKI CME entries when multiple DONKI entries match a single CDAW entry. If we only consider this enhanced dataset when computing these CME History features, we may lose Double CMEs that were recorded in DONKI, but only one CME was recorded in CDAW. Therefore, a two-step approach should be explored. First, an investigation into which dataset is best in identifying Double CMEs should be conducted. Then, using the optimal dataset, additional features in identifying Double CMEs that are related to SEP events should be explored.

Bibliography

[1] CDAW. https://cdaw.gsfc.nasa.gov/CME_list/.

[2] Coronagraph Capture from CDAW on 10/29/15 at 1:48. https://cdaw.gsfc.nasa.gov/images/soho/lasco/2015/10/29/20151029_014804_lasc2rdf_aia193rdf.png.

[3] Coronagraph Capture from CDAW on 10/29/15 at 3:12. https://cdaw.gsfc.nasa.gov/images/soho/lasco/2015/10/29/20151029_031208_lasc2rdf_aia193rdf.png.

[4] Coronagraph Capture from CDAW on 11/13/12 at 2:12. https://cdaw.gsfc.nasa.gov/images/soho/lasco/2012/11/13/20121113_021206_lasc2rdf_aia193rdf.png.

[5] DONKI. https://ccmc.gsfc.nasa.gov/donki/.

[6] GOES Space Environment Monitor. https://www.ngdc.noaa.gov/stp/satellite/goes/index.html.

[7] Keras. https://keras.io/api/.

[8] PHTX File for 10/29/2015. https://cdaw.gsfc.nasa.gov/CME_list/daily_plots/sephtx/2015_10/sephtx_20151029.png.

[9] PHTX File for 12/28/2015. https://cdaw.gsfc.nasa.gov/CME_list/daily_plots/sephtx/2015_12/sephtx_20151228.png.

[10] PHTX File for 6/15/2012. https://cdaw.gsfc.nasa.gov/CME_list/daily_plots/sephtx/2012_06/sephtx_20120615.png.

[11] PHTX File for 6/18/2015. https://cdaw.gsfc.nasa.gov/CME_list/daily_plots/sephtx/2015_06/sephtx_20150618.png.

[12] Real Time Solar Wind. https://www.swpc.noaa.gov/products/real-time-solar-wind.

[13] Type II Radio Burst List. https://cdaw.gsfc.nasa.gov/CME_list/radio/waves_type2.html.

[14] Pierre Baldi. Autoencoders, unsupervised learning, and deep architectures. In *Unsupervised and Transfer Learning - Workshop held at ICML 2011, Bellevue, Washington, USA, July 2, 2011*, volume 27 of *JMLR Proceedings*, pages 37–50. JMLR.org, 2012.

[15] Soukaina Filali Boubrahimi, Berkay Aydin, Petrus Martens, and Rafal Angryk. On the prediction of gt;100 mev solar energetic particle events using goes satellite data. In *2017 IEEE International Conference on Big Data (Big Data)*, pages 2533–2542, 2017.

[16] Pedro Brea, Hazel M Bain, and Eric T Adamson. Using machine learning techniques to forecast solar energetic particles. In *AGU Fall Meeting*, 2018.

[17] Leo Breiman. Random forests. *Machine learning*, 45(1):5–32, 2001.

[18] Alessandro Bruno and I. Richardson. Empirical model of 10 – 130 mev solar energetic particle spectra at 1 au based on coronal mass ejection speed and direction. *Solar Physics*, 296, 02 2021.

[19] Miles D. Cranmer, Alvaro Sanchez-Gonzalez, Peter W. Battaglia, Rui Xu, Kyle Cranmer, David N. Spergel, and Shirley Ho. Discovering symbolic models from deep learning with inductive biases. In *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual*, 2020.

[20] L. Oc. Drury. REVIEW ARTICLE: An introduction to the theory of diffusive shock acceleration of energetic particles in tenuous plasmas. *Reports on Progress in Physics*, 46(8):973–1027, August 1983.

[21] Otto Hyvärinen. A Probabilistic Derivation of Heidke Skill Score. *Weather and Forecasting*, 29(1):177 – 181, 2014.

[22] I.G. Richardson, M.L. Mays, B.J. Thompson. Prediction of Solar Energetic Particle Event Peak Proton Intensity Using a Simple Algorithm Based on CME Speed and Direction and Observations of Associated Solar Phenomena. *Space Weather*, 16:1862–1881, 2018.

[23] Fadil Inceoglu, Jacob H Jeppesen, Peter Kongstad, Néstor J Hernández Marcano, Rune H Jacobsen, and Christoffer Karoff. Using machine learning methods to forecast if solar flares will be associated with cmes and seps. *The Astrophysical Journal*, 861(2):128, 2018.

[24] Jeya Vikranth Jeyakumar, Joseph Noor, Yu-Hsi Cheng, Luis Garcia, and Mani Srivastava. How can i explain this to you? an empirical study of deep neural network explanation methods. *Advances in Neural Information Processing Systems*, 2020.

[25] Kahler, Stephen W. and Ling, Alan. G. Forecasting solar energetic particle (sep) events with flare x-ray peak ratios. *J. Space Weather Space Clim.*, 8:A47, 2018.

[26] Bingyi Kang, Saining Xie, Marcus Rohrbach, Zhicheng Yan, Albert Gordo, Jiashi Feng, and Yannis Kalantidis. Decoupling representation and classifier for long-tailed recognition. In *8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020*. OpenReview.net, 2020.

[27] Anuj Karpatne, Gowtham Atluri, James H Faghmous, Michael Steinbach, Arindam Banerjee, Auroop Ganguly, Shashi Shekhar, Nagiza Samatova, and Vipin Kumar. Theory-guided data science: A new paradigm for scientific discovery from data. *IEEE Transactions on knowledge and data engineering*, 29(10):2318–2331, 2017.

[28] E Lavasa, Georgios Giannopoulos, Aikaterini Papaioannou, A Anastasiadis, IA Daglis, Angels Aran, David Pacheco, and B Sanahuja. Assessing the predictability of solar energetic particles with the use of machine learning techniques. *Solar Physics*, 296(7):1–47, 2021.

[29] Donald V Reames. The two sources of solar energetic particles. *Space Science Reviews*, 175(1-4):53–92, 2013.

[30] Marco Túlio Ribeiro, Sameer Singh, and Carlos Guestrin. "Why Should I Trust You?": Explaining the Predictions of Any Classifier. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, San Francisco, CA, USA, August 13-17, 2016*, pages 1135–1144. ACM, 2016.

[31] Subham Sahoo, Christoph Lampert, and Georg Martius. Learning equations for extrapolation and control. In *International Conference on Machine Learning*, pages 4442–4450. PMLR, 2018.

[32] Jesse Scott Torres. A Machine Learning Approach to Forecasting SEP Events with Solar Activities. Master's thesis, Florida Institute of Technology, https://repository.lib.fit.edu/handle/11141/3215, December 2020.

[33] Silviu-Marian Udrescu and Max Tegmark. Ai feynman: A physics-inspired method for symbolic regression. *Science Advances*, 6(16):eaay2631, 2020.

[34] Peng Wang, Kai Han, Xiu-Shen Wei, Lei Zhang, and Lei Wang. Contrastive learning based hybrid networks for long-tailed image classification. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 943–952, 2021.

[35] Frank Woodcock. The Evaluation of Yes/No Forecasts for Scientific and Administrative Purposes. *Monthly Weather Review*, 104(10):1209 – 1214, 1976.

[36] Boyan Zhou, Quan Cui, Xiu-Shen Wei, and Zhao-Min Chen. Bbn: Bilateral-branch network with cumulative learning for long-tailed visual recognition. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9719–9728, 2020.

Appendix A

Calculating Features Derived from Physics

Here, we explain how we calculate the values for two separate features: peak proton intensity from Richardson et al [22] (commonly referred to as Richardson's Formula or Value), and Diffusive Shock Acceleration from Drury [20] (commonly referred to as Diffusive Shock or V^{V^2}).

A.1 Peak Proton Intensity

Richardson *et al.* [22] derived a formula to predict the peak proton intensity for 14-34 MeV using characteristics from the most recently-occurred CME event. Torres used this for predicting the occurrence of an SEP event [32]. In this work, we modify the formula slightly. In the works of Richardson *et al.* and Torres, they include a speed term as input into Equation A.1. We decide to omit the speed input, as we include various types of speed features elsewhere in our proposed models. Therefore, we use the formula as follows:

$$I\phi(MeVs * cm^2 * sr)^{-1} \approx 0.013exp(-\frac{\phi^2}{2\sigma^2}), \tag{A.1}$$

where σ is set to 43 degrees and ϕ is the connection angle between the CME and the magnetic field line connecting the Sun and Earth. The connection angle is calculated as follows:

$$\phi = arccos[sin(\theta_1) * sin(\theta_2) + cos(\theta_1) * cos(\theta_2) * cos(\theta_1 - \theta_2)], \tag{A.2}$$

where θ_1 is the event latitude from DONKI, ϕ_1 is the event longitude from DONKI, ϕ_2 is fixed at 0 degrees similar to Torres's work. However, dissimilar to Torres's work, θ_2 is calculated as follows:

$$\theta_2 = \frac{27.2 * 1.5 * 10^8}{V_{SWS} * 360}, \tag{A.3}$$

where V_{SWS} is the solar wind speed in km/day [12]. Further, θ_2 is converted to degrees in order to calculate the connection angle.

A.2 Diffusive Shock Acceleration

Torres additionally used Diffusive Shock Acceleration, referred to as V^{V^2}, as a feature to predict SEP occurrence. Though there are many adjustable parameters, we only vary the CME Linear Speed. In our case, we use the linear speed from each event's DONKI entry. We begin by showing the entire expression to calculate Diffusive Shock:

$$\eta v \frac{1}{\gamma - 1} \frac{1}{(1 + \frac{v_{inj}^2}{\kappa v_{inj}^2})^{\kappa+1}} (\frac{v_{inj}}{v})^{\gamma+1}, \tag{A.4}$$

where ν is shock efficiency set to 0.1, v is particle speed set to 44,000 km/s for 10 MeV protons, and κ is a distribution parameter set to 2. Then, v_{inj} is calculated as follows:

$$v_{inj} = 2.5V, \tag{A.5}$$

where V is the linear speed of the particular CME. Finally, before we compute γ, we compute the Mach number of the CME event M as follows:

$$M = \frac{V}{V_A}, \tag{A.6}$$

where V_A is the Alfven Speed set to 600 km/s. If $M > 1.1$, we calculate γ as follows:

$$\gamma = \frac{4M^2}{M^2 - 1} \tag{A.7}$$

If $M \leq 1.1$, we substitute 1.1 in for M and compute γ using the above equation. In this case, this leads us to fix γ as 23.