A Machine Learning Approach to Forecasting SEP Intensity and Times based on CME and other Solar Activities

by

Peter John Thomas

Bachelor of Science
Nuclear Engineering
UC Berkeley
2016

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

Master of Science
in
Computer Science

Melbourne, Florida
July, 2022

We the undersigned committee
hereby approve the attached thesis

A Machine Learning Approach to Forecasting SEP Intensity and Times based on CME and other Solar Activities by Peter John Thomas

---

Philip Chan, Ph.D.
Associate Professor
Computer Engineering and Sciences
Major Advisor

Ming Zhang, Ph.D.
Professor
Aerospace, Physics, and Space Sciences

Marius C. Silaghi, Ph.D.
Professor
Computer Engineering and Sciences

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

Abstract

Title:

A Machine Learning Approach to Forecasting SEP Intensity and Times based on CME and other Solar Activities

Author:

Peter John Thomas

Major Advisor:

Philip Chan, Ph.D.

High intensity Solar Energetic Particle (SEP) events pose severe risks for astronauts and critical infrastructure. The ability to accurately forecast the peak intensity and times of these events would enable preparatory measures to mitigate much of this risk. Machine learning approaches have the potential to use characteristics of CMEs and other space weather phenomena to predict SEP intensities and times. However, the severe sparsity of SEP events in current datasets poses a problem to traditional machine learning techniques. In this work, we present a dataset of proton event intensities and times, as well as features for corresponding CMEs and space weather events. We then demonstrate machine learning techniques for imbalanced data that are able to achieve an MAE of 1.50, a TSS of 0.74, an HSS of 0.73, and an F1-Score of 0.74 for intensity prediction. Additionally, we demonstrate our models' ability to forecast SEP event times, achieving an MAE of 0.74 for threshold and an MAE of 0.69 for peak times.

Table of Contents

List of Figures

List of Tables

Acknowledgements

Firstly, I would like to thank my advisor, Dr. Philip Chan, for advising me throughout my graduate research and during the writing of this thesis. He imparted a firm understanding of the underlying principles of machine learning in his foundational course on the subject as well as during our research that I hope to carry with me throughout my professional career. I would not have been able to effectively apply machine learning techniques and interpret their results without his guidance.

I would also like to thank Dr. Ming Zhang for providing me the opportunity to work on this fascinating problem. His expertise on space weather was invaluable in helping me understand the hard physics behind our research, as well as to interpret our results based on the actual science driving the formation of CMEs and SEPs. Additionally, I would like to thank my other committee member, Dr. Marius Silaghi. His course on computer architecture was one of the first prequisites I completed to qualify for graduate studies in computer science, and bridged the abstraction of computing in my mind to the actual machine under the hood. I can say with certainty that my grasp on the fundamentals of computer science would be much weaker without having taken his course.

During my work, I recieved feedback and technical aid from Jesse Torres and Peter Tarsoly, and I would like to extend my gratitude to them for their help during our research. I would also like to thank to Karen Brown, Cheryl Mitravich, and Leslie

Smith for the immense support they have provided helping me navigate the paperwork and administrative aspects of my graduate studies.

Finally, I would like to thank my family and friends for supporting me on this journey. In particular, my former coworkers and current friends, Justin Fletcher and Ian McQuaid, were instrumental in setting me down a path to pursue an advanced education in computer science. Additionally, my brother Philip Thomas was a constant source of encouragement while continuing my graduate studies.

Chapter 1

Introduction

1.1 Motivation

Developing high-fidelity methods for forecasting occurence and intensity of Solar Energetic Particle (SEP) events is crucial for protecting astronauts and vulnerable infrastructure in space as well as closer to the Earth's surface. Solar energetic particles consist of protons, electrons, and other higher mass ions. Of these particles, the most crucial to predict accurately are protons, as they can cause immense structural and bodily damage by ionizing consituent materials. Though protons are constantly emitted from the sun, coronal mass ejections (CMEs) can cause protons to be emitted at higher intensity. Being able to provide indications for times as well as intensity of events would enable satellite operators, astronauts, and custodians for vulnerable infrastructure on Earth to mitigate the worst effects of SEP events.

1.2 Problem

We study the prediction of intensity and times for 10 MeV Proton SEP Events. A 10 MeV SEP event is defined as an event where the intensity of 10 MeV protons surpasses a threshold of 10 proton flux units (pfus). A key driving force for SEP events are interplanetary shocks, events where charged particles transfer substantial amounts of energy via interaction with electromagnetic fields rather than through kinetic collision. Interplanetary shocks are, in turn, caused by CMEs, which are events where large masses of plasma and magnetic flux are ejected by the sun. CME events often occur hours before associated SEP events, and can be detected with optical sensors, providing advanced warning. In addition, the features of CMEs such as their speed and direction are strongly correlated with the strength of the resultant SEP event. These features can then be used to inform an algorithm to predict intensity and as well as times for SEP events.

1.3 Approach

We gather intensity and temporal data for proton events by determining the start, threshold, peak, and ending intensities, as well as times for all of these intensity thresholds, by using data gathered by GOES proton detectors. These instruments measure proton intensity using > 10, > 50, and > 100 MeV channels. There are substantially more 10 MeV proton events recorded above the cosmic ray background than for the other channels, which is why we chose to demonstrate our techniques on these events first. We associated 10 MeV SEP events to corresponding CME events by matching the proton event start time to the CME with the closest event time. After this matching process was performed, further refinement of the matching process was performed

through visual inspection using CME information listed in the CDAW and DONKI catalogs. We then demonstrated several machine learning regression architectures to predict event intensity as well as times. Different machine learning techniques are used to address the grievous imbalance of SEP events in the overall dataset. We then provide analysis of the results and evaluate events that cause our models to under or over predict the observed target values.

1.4 Contributions

In this work, we propose two main contributions: an improved dataset that relates CME and other solar weather features to corresponding SEP intensity as well as threshold and peak time, and several neural network architectures to predict SEP peak intensity, threshold time and peak times.

For the dataset, we build on the work of Torres [19] who linked events from the CDAW CME catalog to SEP events to forecast occurence and predict intensity of these events. This dataset was then expanded by Tarsoly [18], who linked events in the DONKI catalog to those in the CDAW catalog. Tarsoly [18] used this new dataset, enhanced with additional CME features such as speed and width of event, to forecast SEP occurence. In this work, we associate flux data gathered from GOES proton sensors to CME events in the previous dataset to provide proton event intensity as well as times.

For the regression model, we employ two stages of training to seperate representation and regression learning for the model. To aid the representation stage of model learning, we utilize an autoencoding network branch to encode accurate data representations in the model's feature extractor. We then demonstrate a third stage of training which calibrates regression scores learned by training on uniform and oversampled

distributions of data.

For the peak intensity prediction problem, we find that by using the two-stage approach we are able to achieve an F1-score of 0.74 with 2.6 false-negatives out of 10 SEP events in the test set. With the additional calibration stage, we are able to achieve an F1-score of 0.76 with 3.2 false-negatives. For threshold and peak time prediction, we find that we are able to achieve a 0.74 and 0.69 SEP-MAE respectively using the two-stage approach.

1.5 Organization

We discuss prior work related to forecasting SEP events as well as machine learning with imbalanced data in Chapter 2. In Chapter 3, we introduce and apply our models to the task of predicting SEP peak intensity. We then analyze the results to evaluate the predictive performance of our models. Finally, in chapter 4, we describe the problem of predicting threshold and peak time for SEP events and analyze the performance of our models on this problem.

Chapter 2

Related Work

2.1 SEP Forecasting and Intensity Prediction

We are interested in predicting the peak intensity as well as times for SEP events. Relating to the first task, physics-based models have been used in previous work to predict the peak intensity of SEP events. Additionally, Machine learning techniques have been used in several works to forecast occurence and characteristics of SEP events. We present a synopsis of these works in the following sections.

2.1.1 Forecasting and Intensity Prediction with Empirical Models

Multiple physics-based models have been developed to forecast SEP events and predict their peak intensities. In Richardson's work [15], a model for predicting intensities of SEP events for 14 - 24 MeV protons at a distance of 1 Astronomical Unit (AU) was developed using 25 SEP events observed by the Large Angle and Spectrometric Coronagraph (LASCO) and Solar Terrestrial Relations Observatory (STEREO) coro-

nagraph. This formula uses the connection angle and speed of the associated CME as input parameters, and is given by equation 2.1.

$$I(\psi)(\text{MeV}s \cdot \text{cm}^2)^{-1} \approx 0.013\exp(0.0036V - \frac{\psi^2}{2\sigma^2}), \sigma = 43^\circ \tag{2.1}$$

To evaluate the performance of their model, Richardson et al. predicted the intensity of proton events associated to CME events spanning a time period from October 2011-July 2012 in the Donki catalog. In addition, because observations of some of these SEP events were used to develop the empirical model, a seperate test set composed of events from the CDAW catalog was utilized to evaluate the model's performance. Richardson et al. determined that their empirical model was effective at predicting proton intensities for higher intensity events, and that there was a fairly linear correlation between predicted and observed intensities for these events. However, at lower observed intensities, the model began to fail and produce a larger number of "false alarms", i.e. CME events whose associated proton intensity was incorrectly predicted to surpass the SEP intensity threshold.

Bruno and Richardson [6] improved on the previous empirical model by using a two-dimensional Gaussian using longitude and connection-angle as input parameters. The formula that was proposed in this work also contained terms dependent on the spatial distribution and velocity of the associated CME, and was developed to predict the peak intensity of SEP events between 10 MeV and 130 MeV. This model is given by the formula,

$$\phi(E) = \phi_0(E, V)G(E, \delta) \tag{2.2}$$

where ϕ is the intensity of the event, ϕ_0 is the maximum intensity calculated with CME velocity, and $G(E, \delta)$ is the gaussian of latitude and longitude of the CME event. Bruno and Richardson then used a set of 20 SEP events that were not used to formulate

the parameters of the model to test its predictive performance. Their results showed that their model was able to achieve reasonable accuracy predicting peak intensities for events between 14-24 MeV. However, model performance tended to drop off at higher and lower energies.

2.1.2 Forecasting SEP Events with Machine Learning Techniques

Recently, machine learning techniques have been used to provide forecasts and predict information about SEP events. Numerous classes of input features derived from various spaceweather phenomena have been used to inform machine learning approaches to SEP prediction, including X-ray readings, characteristics of the solar surface during flare events, information about CMES, as well as some combination thereof.

2.1.2.1 Using Characteristics of X-rays

Many of the papers in current literature use features of X-rays to perform prediction for SEP events. The advantage of using X-rays to inform machine learning approaches is that information about these events travels faster than for other spaceweather effects, providing advanced warning and forecasting capabilities when deployed in the real world.

In Kahler and Ling [8], the flux of X-rays during solar flare events was used to predict the occurence of > 10 MeV proton SEP events. In this case, X-ray features were chosen over CME features, as information about X-rays during solar flare events precedes that for corresponding CME events, providing earlier warning about SEP events that follow. In particular, the relation between the ratio between different X-ray bands and the location of the flare source was used to inform a model to forecast SEP

events. Using events generated from the western hemisphere of the sun, Kahler and Ling trained both a multi-layer perceptron (MLP) and a K-nearest neighbors (KNN) classifier to flag inputs as SEPs or non-SEPs based on observed peak flux ratios for the event. The results of their work suggested that the decision boundary that the MLP algorithm learned was smoother than that of the KNN classifier, making it more suitable as a forecasting tool.

Giamini et al. also used machine learning techniques to predict the occurence of SEP events [3] using readings of soft X-rays (SXR) produced by solar flares collected by NOAA GOES spacecraft from 1988-2013. Giamini et. al composed time-series of X-ray flux data to serve as the input for their model, a multi-layer neural network. Like in many of the other works we discuss, Giamini et al. encountered problems with the class-imbalance between the SEP and non-SEP instances in the dataset. To improve model performance on the SEP minority class, Giamini et al. investigated a series of sampling techniques, including random oversampling and undersampling as well as the Synthetic Minority Oversampling Technique (SMOTE). SMOTE works by constructing new, synthetic examples for the minority class by randomly selecting an instance from the minority class and identifying the k-nearest neighbors of that instance. Once these neighbors are identified, an instance from this set of neighbors is selected at random. The vector between the original instance and the neighbor is identified in feature space and multiplied by some factor between 0 and 1, and then added to the features of the original instance to construct a new example. In addition to this oversampling technique, a weight schema was used in the model loss function to weigh SEP examples more highly than non-SEP instances, biasing the algorithm to classify SEP instances more accurately. The model that Giamini et al. used consisted of five hidden layers with batch normalization applied between layers. To evaluate their model, Giamini et al. used a k-fold cross-validation procedure which used all but one

of the SEP events in the training set and a randomly sampled subset of 300 examples from the oversall set of 15000 non-SEP instances. After model training, the model was used to predict scores for the single the SEP instance and the rest of the non-SEP instances in the test set. A correct prediction on the SEP instance was defined as a "hit" while a wrong prediction was defined as a "miss". The accuracy statistics for this single SEP instance in the test set were compiled over ten training runs. This process is then repeated until all SEP instances have been included in the test set once. Using this evaluation procedure, the authors of the paper recorded that 191 out of 220 SEP instances were always predicted correctly, while 19 SEP instances were never predicted correctly and 10 SEP instances were only predicted correctly for some of the trial runs.

2.1.2.2 Using Characteristics of Solar Flares

In addition to the aforementioned X-rays readings, other characteristics of solar flares have been used to predict information about SEPs, including various readings of the magnetic field on the sun's surface during these events. Occurence of these magnetic field features is highly correlated to both SEP as well as CME events, and so can provide information about both types of events.

Inceoglu et al. [7] used features of solar flares to predict the occurrence of both CMEs as well as SEPs. This led from observations that the speeds and energies of CMEs could be correlated to whether or not there was an accompanying bright flare event. The authors of the paper evaluated the predictive performance of two classes of algorithm, SVMs and MLPs, using features derived from vector magnetic field data of active regions of the solar surface recorded by the Helioseismic and Magnetic Imager (HMI) on board the Solar Dynamics Observatory (SDO). A number of different forecasting windows were evaluated. For the 24-hour time window, the authors of the paper were able to achieve a TSS and an HSS of 0.79 ± 0.11 for forecasting whether a

flare event would be unassociated with any other observable solar event, while achieving a TSS $= 0.74 \pm 0.13$ and an HSS $= 0.77 \pm 0.05$ for predicting whether a flare would be associated a CME and/or SEP.

Kasapis et. al [10] used characteristics of active regions of the solar surface during solar flare events contained in the Space-Weather MDI Active Region Patches Database (SMARPs) as well as the connection angle between the geometric center of the active region and the magnetic-foot point of the Earth. This database was compiled with readings taken by the Michelson Doppler Imager (MDI) on the SOHO Spacecraft of the Solar surface between the years 1996 and 2010. They then trained Support Vector Machines (SVM) and linear regression models to predict which flare eruption events would lead to SEP events. To compensate for the relative lack of SEP events in the dataset, the authors of the paper assigned 90% of the SEP events to the training set and used undersampling to make the number of positive SEP events equal to the number of negative non-SEP events. The authors then tested their algorithm on a test set that was composed of the remaining SEP events and consisted of equal numbers of SEP events and non-SEP events. This method for partitioning the dataset was used to create a new split for each of the 100 runs conducted with each model. For each split, the SEP events were selected with replacement (as there were few of these events to select from) while the non-SEP events are selected without replacement. To quantify the predictive power of their models, the authors of this paper used accuracy, the True Skill Statistics (TSS), and the Heidke Skill Score (HSS) as evaluation metrics. Using a third degree support vector machine model, the authors of this paper were able to achieve an accuracy of 70% of correct predictions and an HSS of 0.43 on the balanced test-set.

2.1.2.3 Using Characteristics of Type II Radio Bursts

In addition to to previous events, radio wave readings for different types of radio events such as Type II radio busts have also been used to inform model predictions. Stumpo et al. [17] used the averaged time of M2 SXR and 1 MHz radio fluxes integrated over a period of 5 minutes. The dataset that they composed consisted of 989 M2 flare events. Of those events, 92 were labeled as SEP instances. This feature set was then used to inform a logistic regression model to perform prediction for SEP events. As in other two papers, the authors of this paper implemented methods for improving performance on the SEP instances, which, again, composed an extreme minority of the entire dataset. In particular, the authors of Stumpo et al. used a combination of a weighting schema in the loss function that weighted SEP instances more heavily than non-SEP instances as well as SMOTE oversampling. Unlike other papers, the performance metric that was tracked in this paper was the F1-score, which is a harmonic mean of the precision and recall, and will be defined in section 3.3.2. At optimal threshold, the logistic regression model that was demonstrated in their paper was able to achieve an F1-Score of 0.65.

2.1.2.4 Using Characteristics of Multiple Space Weather Phenomena

Finally, features from several different solar activities have been combined to provide models information from several domains of space weather phenomena. This has the advantage of increasing the predictive power of the model, as the model now has more information to extract to form its predictions. These features are derived from a variety of sources, such as aforementioned X-rays and solar flares as well as CMEs and the flux of protons in some temporal window before the event occurs.

Like Giamini et. al, Boubrahimi et. al [4] sought to exploit the features of X-ray and proton flux readings conducted by GOES satellites during flare events. Additionally,

they also used features of CMEs to predict the occurence of > 100 MeV SEP events. During flare events and CMEs, flashes of increased brightness can be observed near the solar surface due to X-ray and UV radiation, which is one of the first indications that a high-intensity charged particle event is about to occur. These observations can be conducted up to 30 minutes before the correlated SEP takes place, giving some protential for providing early forecast and warning capabilities. In total, Boubrahimi et. al's dataset consisted of X-ray and proton fluxes for $47 > 100$ MeV SEP events, consisting of the model's positive examples. An additional 47 instances were constructed using X-ray and proton fluxes that were not associated with a > 100 MeV SEP, composing the negative examples in the dataset. In their work, Boubrahimi et. al used decision tree models to model the correlation between measured X-ray and proton data and whether or not an SEP event would result from the observations. To obtain the attributes for the decision tree, the correlations between the measured X-ray and proton fluxes over time were determined using a vector autoregression model (VAR). VAR models estimate the value of a time-series equation by determining the dependence of the equation on previous values in the time series as well as quantities in other, related time series. The coefficients for the equations for proton flux determined using the VAR models were used as the input feature vectors for each instance in the dataset. The targets for the prediction model were binary classification labels that divided the instances of the dataset into two classes: SEP and non-SEP. A decision tree was then constructed using the VAR coefficients as the input attributes. Using this model, Boubrahimi et. al were able to achieve an accuracy of 0.78, with an F1-score of 0.82 and a recall of 0.73.

Predictions for 10 MeV SEP events were performed by Brea [5] using X-ray fluxes measured by the detectors on GOES and Type II and type IV radio bursts as input. In addition, CME features such as speed, width, and direction were introduced as inputs

to the model, forming different sets of features whose predictive performance was compared in the paper. Brea then trained a variety of machine learning architectures such as logistic regression models, decision trees using the boosting algorithm AdaBoost, and SVMs to predict a score between '0' and '1', with '0.5' taken as the threshold value to classify the event as either an SEP or non-SEP (i.e., events with predicted scores above '0.5' are forecasted as SEP events). Brea determined that the addition of CME features helped improve the predictive performance of the model. In addition, SVMs attained the highest HSS score on the problem, scoring an HSS of 0.53 on their test set.

Torres [19] sourced inputs from a variety of data sources including features of CME events retrieved from the CDAW catalog and Type II radio bursts to compose his dataset. Additionally, he used a set of features derived from equations describing the behavior of CMEs and SEPs, such as the diffusive shock acceleration formula which describes the behavior of charged particles in a strong magnetic field. He used these features to train a multi-layer perceptron (MLP) model to forecast occurence of SEP events. Additionally, he used proton and electron flux readings taken over time to train a recurrent neural network (RNN) to predict intensity of SEP events. Importantly for our work, he ranked the importance of various features used to train his forecasting model and determined that the features of most importance were Type II area of radio burst events, the number of sunspots that had been spotted on the sun's surface at the time of a CME, and the characteristics of the CMEs listed in the CDAW catalog such as width and acceleration.

Lavasa et. al [11] used features of solar flares such as soft X-ray readings from GOES satellites and CMEs such as speeds and half-angles gathered from the CDAW catalog to perform forecasting of SEP events. In addition, Lavasa et. al attempted to formalize the results of their work, taking into account the generalizability of their

algorithm to real world usage. They took into consideration aspects such as how well their problem was formulized, how effective the hyperparameter studies for their machine learning algorithms were conducted, and how well the results of the models could be explained by the set of input features to ensure that the results in their paper were reproducible. To that end, the authors of this paper made the code that they used to perform model training open source and available to the larger public. A variety of machine learning architectures were prototyped to perform the forecasting task, and the authors of the paper achieved the highest performance using random forest methods, achieving a probability of detection (POD) of 0.76 ± 0.06, a TSS of 0.75 ± 5, and an HSS of 0.69 ± 0.04.

Finally, Tarsoly [18] expanded on Torres's work by linking CME events listed in the DONKI catalog to events in the CDAW catalog and introducing a set of features from DONKI including speed, half-angle, longitude, and latitude of the CME event. This set of DONKI features, as well as the set used in Torres's work, comprised his input features for model training. Tarsoly then used several machine learning techniques, such as classifier retraining (cRT), which is described in further detail in the next section, as well as cRT with an autoencoder branch (cRT+AE) to forecast whether a CME would lead to a corresponding SEP event. As in Torres's work, Tarsoly employed techniques for determining the ranking of importance for different subsets of feature inputs, and determined that the most important set of input features were those relating to CME speed, location, and history. Tarsoly then provided an analysis of common errors made by his models, particularly the false-negatives, and noted that there was a high correlation with false-negatives and Energetic Storm Particle (ESP) events and double CME events, i.e. events where multiple CMEs serve as the progenitors of the same SEP event.

2.2 Machine Learning with Imbalanced Datasets

2.2.1 Imbalanced Classification

Imbalanced classification problems are relatively well-studied compared to other domains in machine learning dealing with imbalanced data. For classification problems, the output target is a vector of length 'n', where 'n' is the number of classes in the problem. Each element of the vector therefore describes the predicted probability that an instance belongs to that class in the dataset. The label associated with each instance is its class ID. The goal of model learning is to output a class probability distribution with the correct ID assigned the highest probability. Imbalanced classification problems are, therefore, a set of problems where the number of instances belonging to one or multiple class IDs are heavily outnumbered by instances from other classes. Imbalanced classification tasks pose a problem for conventional machine learning techniques, as the features that machine learning models learn are representative of the overall distribution of training data. If a minority class is not well represented in this distribution, the model may not be able to learn an effective representation for that minority class, lowering its prediction accuracy on that class. The following works deal with improving representation learning for minority classes in long-tail training distributions as well as improving model performance on imbalanced classification tasks in general.

Kang et al. [9] describe several techniques for improving model performance on imbalanced classification problems. These techniques include Classifier ReTraining (cRT), nearest class mean classifier (NCM), and tau-normalized classification. Of these techniques, cRT achieved the highest classification accuracy on the ImageNet-LT benchmark dataset using most model backbones. This technique improves model perfor-

mance on imbalanced data by separating the representation and classification learning for the model into two distinct stages. The "representation" learning portion of model training involves learning appropriate representations of the input data. These representations are used as templates describing what feature values are associated with instances in different classes. The classification learning portion of model training aims to produce prediction scores that are as close to the true target probability vector as possible, using the representations that are learned by the model. In traditional learning algorithms, these two learning tasks are performed in tandem. However, Kang et. al posited that separating the representation and classification learning would improve model performance on imbalanced classification problems by ensuring that the representations the model learned were consistent with the actual distribution of data. Instead of single stage training, Kang et. al proposed a training strategy where the representation and classification learning were performed in two separate stages. The first stage performed the representation training for the model on a uniform distribution of input data. In a uniform distribution, each instance in the original data distribution is sampled with equal probability, with no variation in sampling rate for instances in different classes. After the first stage of training, the weights of the feature extractor layers for the network, which are responsible for learning feature representations of the input data, are frozen. This means that, when subsequent training is performed, the weights for these layers of the network are not updated via gradient backpropagation. The weights of the classification head for the network are then reinitialized, and the model is trained on a class-balanced distribution of input data. This class-balanced distribution can refer to making the number of instances per class equal in the input dataset or ensuring that each class has the same influence on the gradient that is back-propagated through the network after each iteration of training. This is achieved using some over or under-sampling strategy on the input dataset or with some class-based

reweighting schema in the loss function. This two-stage method for model training yielded significantly better performance on imbalanced datasets compared to when the representation and classification learning were performed simultaneously.

Zhong et al. [22] expanded on the two-stage model learning technique by investigating the tendency for classification models to predict not only inaccurate labels for minority classes in imbalanced datasets, but also how overconfident the model was when making those predictions. They measured this overconfidence by measuring the expected calibration error (ECE), which is the difference between the model's confidence on predictions and the actual predictive accuracy for that class. The paper posited that using methods to lower the ECE would improve model performance on minority examples in the dataset. The paper then detailed three techniques for lowering the ECE. One of these techniques is mixup, which linearly combines the features and target labels of examples in the training dataset. In essence, this extrapolates known examples in feature space and label space, providing the model more examples in the feature space between known instances. The second technique introduces a class-wise scaling term in the cross-entropy loss function used for classification tasks that is inversely proportional to the number of instances in that class. This is to remedy the problem of weight norms for majority classes being much larger than those for minority classes in the prediction layer, resulting in overconfidence for majority class predictions. Thus, the scaling factor is larger with minority classes and smaller for majority classes, which makes the weight norms of the different classes in the prediction layer more equal. Finally, the paper introduced a shift learning strategy for batch normalization for two-stage model training. This technique was designed specifically to address the difference in running statistics for batch normalization layers when training at different oversampling rates using two stages of training. Using the long-tailed versions of CIFAR-10 and CIFAR-100 as benchmark datasets, the authors of the pa-

per determined that the ECE for their trained model was minimized when all three techniques were employed.

Zhou et. al [23] posited that common techniques for improving model performance on long-tail classification datasets, such as reweighting the loss function and oversampling minority classes, ended up hurting the representations in the deep features learned by the model. Zhou et al. argued that, due to the sparsity of unqiue instances in the minority class, oversampling and reweighting strategies could move the decision boundary in feature space to more accurately classify minority examples. However, the representations in the deep features of the model could result in overfitting of minority instances in the training distribution, resulting in decreased performance on minority class instances falling outside that distribution. In addition, the representations the model learned could cause it to underfit the overall data distribution of the model, hurting model performance on majority classes. To address the problem of learning effective representations for the minority and overall data distribution, Zhou et al. proposed a two-branch network architecture, one of which would be trained on the uniform data distribution while the other would trained on a reversed sampled distribution. The features for each of these branches are then linearly combined before the final softmax classification layer with a parameter α whose value is dependent on the current epoch of training. This training schema is coupled with a cross entropy loss that uses the same alpha parameter to linearly combine the error calculated for the uniform distribution and the class-balanced distribution. After training has completed, the alpha parameter is set to a constant value of 0.5, equally weighting the output of the uniform and reverse sampled branches of the model. Using the CIFAR-10 and CIFAR-100 benchmark datasets and introducting varying imbalance ratios for the long-tail minority classes, the authors of the paper determined that their BBN architecture had lower top-1 error than conventional imbalanced classification techniques

such as oversampling and reweighting strategies.

Finally, Zhang et. al (2021) [21] proposed two techniques tp address the distribution misalignment between uniform and class-balanced datasets. As in Kang et. al, the technique was first described in the context of imbalanced classification problems. Zhang et. al posits that one of the problems with training on imbalanced data is that there is a distribution misalignment between the uniform-balanced dataset and the class-balanced dataset. The paper proposed two techniques to address this misalignment. The first, adaptive calibration uses the weighted sum of two scores to provide the final output classification vector. This first score is the classification vector output by the model trained on the uniform-balanced distribution. The second score is a class calibrated score that adjusts the score value for each class in the classification vector by another linear equation. A linear sum of these two scores weighted by a function $\sigma(z)$ is then used to predict the final classification score for the test-set distribution. The function $\sigma(z)$ forms a distinct layer whose weights are learned during the second-stage of model training. The input z of $\sigma(z)$ are the features learned by the feature extractor for each instance in the first-stage of training. This approach demonstrated higher classification accuracy when using imbalanced training data than using the just original, instance-balanced score. The second technique described by the paper, termed "generalized-reweighting", was a method for reweighting different classes in the loss function. The purpose of this reweighting function was to minimize the K-L divergence between the output probability distribution of the model and a reference distribution that is chosen by the model trainer. In the paper, the authors suggested that this reference distribution should favor the class-balanced distribution. Using both techniques and a two-stage cRT model, the paper authors were able to achieve a higher classification accuracy on their benchmark dataset than cRT alone.

2.2.2 Imbalanced Regression

Imbalanced regression problems are analogous to imbalanced classification tasks, but pose some additional hurdles. One of these hurdles is due to the fact that the output target for the model is no longer a discrete classification vector but a continuous regression value. This makes defining what instances compose the majority and minority examples in the dataset more difficult than in classification tasks, where instances with the same label can be grouped into the same set. In particular, it can be difficult to define which examples should be oversampled to improve model performance on important minority instances. Another problem is that because the target label across minority examples is not the same, the learning task for the model may be more difficult. The model must not only learn effective representations of features for instances in the minority categories, but must also learn to generate varying regression targets rather than a constant classification vector for each of those instances.

A technique called SMOTEBoost was developed to improve model performance on regression tasks by augmenting minority instances with synthetic examples. SMOTEBoost was introduced in Moniz et al. [12] adapted the SMOTE method for augmenting imbalanced classification data for regression. SMOTEBoost extends this augmentation technique for imbalanced regression problems by first defining a relevance function that maps instances in the dataset to a value in the range [0, 1] based on their target value. Different models are then trained over datasets where the highly relevant examples in the dataset are oversampled using SMOTE. After training has completed for each model, the error of the model on the highly relevant examples in the dataset is calculated. Using this distribution, a new ratio of highly relevant samples in the input dataset is generated by creating synthetic examples with SMOTE. These models are collected in an ensemble, and the output regression target is taken to be a sum of the

scores produced all of the models in the ensemble weighted by a parameter β_t that is a function of its error on the training set.

Another paper by Yang et al. [20] made the observation that the test error for imbalanced regression problems tended to be smooth even if the distribution of samples was not. Yang et al. reasoned that performance on imbalanced regression tasks could be improved by smoothing the target label distribution to match the inverse of the test error distribution across examples. They implemented this by using a symmetric convolutional kernel to smooth the label distribution, sampling examples in the training set to match the smoothed distribution. This technique was called Label Distribution Smoothing (LDS). Additionally, they made the observation that feature statistics for minority instances in the train set bore an unexpected similarity to examples that composed the majority of the dataset, even if the target values for those majority instances were quite different. Yang et al. reasoned that the feature statistics learned for minority instances should be similar to other instances with similar target values. A convolutional kernel was employed again to smooth the feature statistics between data bins with similar target values, forcing the statistics for bins with close target values to be similar to each other. This technique for tracking and smoothing feature statistics for each epoch of training was termed Feature Distribution Smoothing (FDS). Using LDS and FDS, the authors of this paper demonstrated improved mean absolute error on minority instances versus other common techniques for imbalanced regression using the IMDB-WIKI dataset that was sampled to create an artificial imbalance of data as a benchmark.

Ribeiro et al. [14] brought up another issue that was not dealt with in any of the previously mentioned papers, namely, that the metrics used to evaluate typical regression models were not adequate for determine performance on imbalanced regression tasks, especially if the imbalanced instances in the dataset were of particular impor-

tance. In typical regression tasks, different target labels are assumed to be of uniform importance. However, in most cases, and especially in the case of imbalanced regression, certain target labels will be more important to predict accurately than others. Ribeiro et al.s work details methods for generating relevance functions from imbalanced datasets, which expresses the application specific bias we would like to impart to the model. The relevance function should give high weight to data with outlier target labels. Their method uses a list of control points and interpolates between these points to generate a continuous relevance function over a set of provided target labels. Ribeiro et al. then details a set of metrics to use for imbalanced regression tasks that give a better indication on the models performance on highly relevant examples as well as the overall distribution of data than typical regression metrics such as mean-squared-error (MSE) and mean-absolute-error (MAE). One of these is the Squared Error-Relevance Area (SERA). SERA is calculated by generating a plot of squared error values for instances above some relevance threshold t. This squared error is generated over the domain of values the relevance threshold can take, [0, 1]. The area under the curve generated by this function is taken to be the SERA value. The goal of model learning on an imbalanced regression dataset is then defined as optimizing this SERA metric. SERA has the useful property of associating error of predictions to the relevance of the associated target label. If SERA is large, then that implies that the model performs worse against highly relevant examples that are included in all relevance threshold cutoffs. However, SERA also accounts for error in examples with low relevance, giving an indication of the models performance over the entire distribution of target values, rather than just the set of extreme target labels. This gives an indication of the models performance over the entire distribution of possible target values, rather than just majority or minority instances.

Chapter 3

Predicting SEP Peak Intensity

3.1 Problem

The first problem that was studied in this work was predicting the peak intensity of 10 MeV proton events using the characteristics of associated CMEs. In particular, we would like to accurately predict the peak intensity of SEP events, which are defined as 10 MeV proton events with a peak intensity above 10.0 pfu. These events are of particular importance in our research, as these extreme events can cause the most damage if they go undetected.

3.1.1 Input Features

To perform the prediction task, a set of suitable input features needed to be determined. One set of inputs consisted of speed, width, and direction of CME events. In addition, contextual information about the frequency of other CMEs as well as occurrence of space weather events on the solar surface were also used as inputs. These inputs were compiled from data gathered from the DONKI and CDAW CME catalogs as well as

the CDAW Type II radio bursts catalog. This required matching CME events in the DONKI database to their corresponding CME event in the CDAW database, which had been performed in previous works [18].

The four features gathered from the DONKI catalog were used as a baseline feature set.

- Speed ($\frac{km}{s}$)
- Half-Angle ($^\circ$)
- Longitude ($^\circ$)
- Latitude ($^\circ$)

This set of DONKI features are especially important for our prediction task, as they compose the primary set of features describing the attributes of CME events that we associate with each SEP. These feature values are, therefore, highly correlated with the predictions made by the model. In general, high speeds as well as half-angles ($> 1000\frac{km}{s}$ and $> 45^\circ$, respectively) are highly correlated with increasing peak intensity. For longitude and latitude, the relation between the feature value and intensity is not positively correlated as it is with speed and width. However, there is still a strong correlation between the peak intensity of the proton event and the direction of the corresponding CME. In particular, the observed intensities of proton events peak when the direction of the associated CME is at approximately 57° longitude and 0° latitude, which roughly corresponds to the direction of the IMF line between the sun and the Earth. In addition to these features, the fitted second order speed of CME events determined from coronograph readings were also used as inputs. These features, as well as the area of Type 2 Radio Events, which are also highly correlated with the intensity of the observed SEP, were sourced from the CDAW catalog.

- Acceleration
- Type 2 Area
- Initial Second Order Speed ($\frac{km}{s}$)
- Final Second Order Speed ($\frac{km}{s}$)
- Second Order Speed at 20 Solar Radii ($\frac{km}{s}$)
- Central Position Angle (CPA) ($^\circ$)
- Measurement Position Angle (MPA) ($^\circ$)

Additionally, a set of additional features were compiled that provide a snapshot of space weather patterns on the sun during the time of the CME event.

- Number of Sunspots
- Number of CMEs in the past month
- Number of CMES in the past 9 hours
- Number of CMEs with speed over 1000 $\frac{km}{s}$ in the past 9 hours
- Maximum speed for a CME in the past day

Finally, we used a set of features derived from values of other features previously mentioned. While the machine learning algorithm is expected to learn these relations from the input data if they are helpful in performing the intensity prediction task, these features were provided to help the algorithm converge to an optimal solution. These derived features consist of formulas mapping input features to values that strongly correlate with the intensity value of the proton event. Notably, the output of the Richardson Formula for each event, which uses the empirical model presented

in Richardson et al. [15] to predict the peak intensity of SEP instances, is also used as input to the model. In addition, a formula for diffusive shock is derived based on diffusive shock acceleration theory from plasma physics. This feature helps describe the growth of the proton front of the CME over time. The formula for this feature was presented in Torres's thesis [19], and details for its formulation are described in that work.

- Richardson Formula
- V log V
- HALO ("1" when the CPA is 360°, "0" otherwise)
- Diffusive Shock

These twenty features comprised the entire set of inputs for our machine learning model for the peak intensity prediction task.

3.1.2 Output Target Values

To determine the prediction target for each instance in the problem, the time of the DONKI CME needed to be matched to the start time of a proton event. This was done by matching the DONKI CME to the SEP event whose start time was closest to the CME time, within a three day padding window. Only proton events whose start time occurred after the CME were considered. This matching algorithm ensured positive matches most of the time, but some CME instances needed to be matched to proton events manually. Some examples of these cases were instances when a higher-speed CME or a CME with a more favorable direction occurred earlier than other CMEs before the proton event, producing a wrong match. In these instances, inspection of the PHTX file on the CDAW website as well as the characteristics of CME events in

the DONKI catalog allowed us to manually match the proton event to the appropriate CME event.

When a match between a CME and a corresponding proton event was made, the peak intensity of the proton event was used as the target label for the problem. Two major categories of data instances composed using this method. The first category were SEP events, whose peak intensity was measured above 10.0 pfu. The second category of event were proton events that did not exceed the 10.0 pfu intensity threshold needed to be categorized as SEP events, but were still strong enough to be detected above the cosmic proton background, which is at approximately $\frac{10.0}{e^2}$ pfu. In cases where a CME event was not matched to an SEP event or an elevated proton intensity event, the aforementioned estimate of the cosmic proton background, or $\frac{10.0}{e^2}$ pfu, was used. It should be noted that this target was arbitrarily chosen, and could be changed in the future to some value below the elevated event threshold, provided more accurate readings become available. The result of this label matching task were three main categories of "proton event": SEP events, elevated intensity proton events, and events assigned the constant background intensity. The number of instances per event category is provided in table 3.1.

Table 3.1: Number of instance per event category

Event Category	**Peak Intensity (pfu)**	**Number of Instances**
1. SEP Event	≥ 10.0	44 (1.8%)
2. Elevated Intensity Proton Event	< 10.0 and $> \frac{10.0}{e^2}$	39 (1.6%)
3. Background Proton Events	$\frac{10.0}{e^2}$	2309 (96.6 %)

The range of peak intensity values for SEP events in the compiled dataset differed by multiple orders of magnitude, ranging from just above the 10 pfu threshold to > 1000 pfu. Because of this, the log of peak intensity was taken as the prediction target for the algorithm so that the range of weight values that needed to be learned

by the regression head would not have to vary as widely or be as large as they would with the original, untransformed prediction target.

3.2 Approach

3.2.1 Random-Oversampling of SEP Events

SEP peak intensity prediction forms an extreme example of an imbalanced regression problem. In particular, the events of interest (the SEP events) comprise $< 2\%$ of the entire dataset. Batch stochastic gradient descent, the mechanism by which the model performs weight updates from the calculated loss after each iteration of training, is performed a per batch basis. With a large enough batch size, each batch can be reasonably expected to reflect the overall distribution of input data. Therefore, the contribution of SEP events to each iteration of weight updates will be negligible for the uniform distribution. This will cause the model to learn a set of weights that will cause it to underpredict the peak intensities for most SEP instances. To mitigate this issue, random oversampling techniques were used to increase the fraction of high-importance SEP events in the training dataset. This technique oversamples the SEP events to some fraction of the original training set by selecting events at random to duplicate. Different oversampling rates were used to find an optimal sampling fraction for the SEP events.

We considered other oversampling techniques, such as SMOTE and other techniques that oversample the data by creating synthetic examples. However, as the generation of synthetic examples could not be tied to some physics-based process, the new events could not be guaranteed to be representative of the underlying physics of the real events. Hence, we decided not to apply these synthetic oversampling techniques to our

problem in order to ensure that the predictions of our model were explainable using the physics associated with each input event.

3.2.2 Random-Oversampling of High Speed and Large Width Non-SEP Events

While running experiments on the initial dataset, we observed that a number of false positives were being generated by the models trained on data where only SEP events were oversampled. We can see this problem in figure 3.1, where we plot the intensity predicted by the trained model vs the values of the speed and half-angle input features for this event. In this figure, the red dots are SEP events, the green dots are elevated intensity proton events, and the blue dots are background proton events.

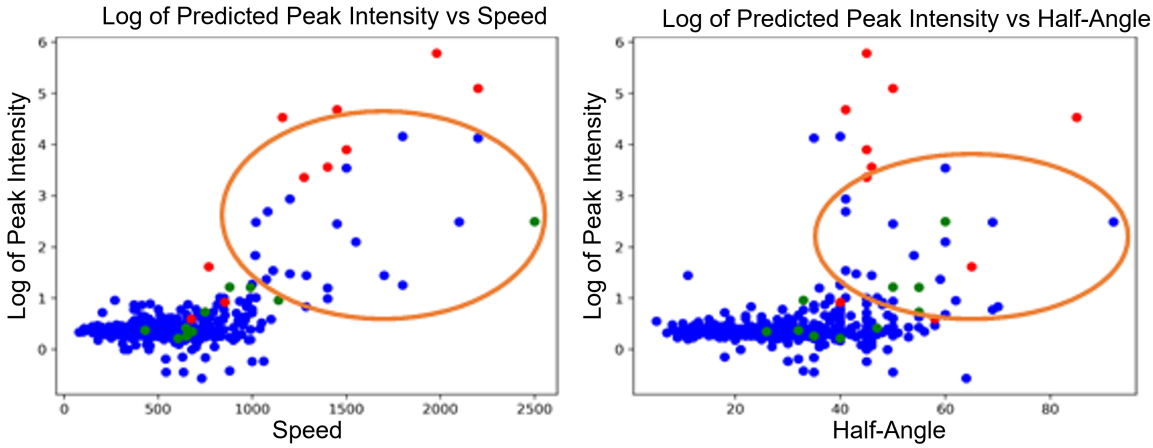


Figure 3.1: Log of predicted intensity vs event speed (left) and half-angle (right)

We can see from this figure that there is a strong, positive correlation between high speed and/or large width events and the intensity predicted by the model. Consequently, many non-SEP events with higher speeds and larger widths caused the model to predict higher intensities than their actual target value, generating a number of "false positives", or events with predicted intensity over the 10.0 pfu SEP threshold.

We were able to mitigate some of this behavior by going through the CME events that are correlated to elevated proton events and confirming that they were correlated to the right event manually. In some instances, the wrong CME was being matched to event, which may have caused the overprediction. However, for most of the events in the dataset, we were unable to find a corresponding match with lower speeds. Therefore, to help mitigate the generation of false positives, we introduced some degree of oversampling for high-speed and large-width non-SEP events to provide the learning model more examples of events with those characteristics whose targets fell below the threshold intensity. Because we oversample using two sets of criterion, we will refer to the oversampling rates used during training as $p-q\%$, where p is the oversampling rate for SEP events and q is the oversampling rate for high-speed and large-width non-SEP events.

3.2.3 Regression-Retraining with Autoencoder

We modified the cRT technique described in 2.2.1 for regression learning by replacing the classification softmax layer with a linear regression output layer. This technique has been used for imbalanced regression problems before. However, this would be the first time it would be applied to SEP peak intensity prediction. As in cRT, training for the regression variant is conducted in two seperate stages. The first stage of training is conducted on a uniform distribution of training data. The purpose of this stage of training is to learn effective representations of the input data. To aid in this task, a decoder branch was also added to the model. The goal of this branch is to reconstruct the input based on the learned representations that are generated by the feature extractor for that instance. This forces the feature extractor to learn more effective representations of the input dataset so that the input features can be reconstructed.

Figure 3.2 shows the model architecture used during the first stage of training. The

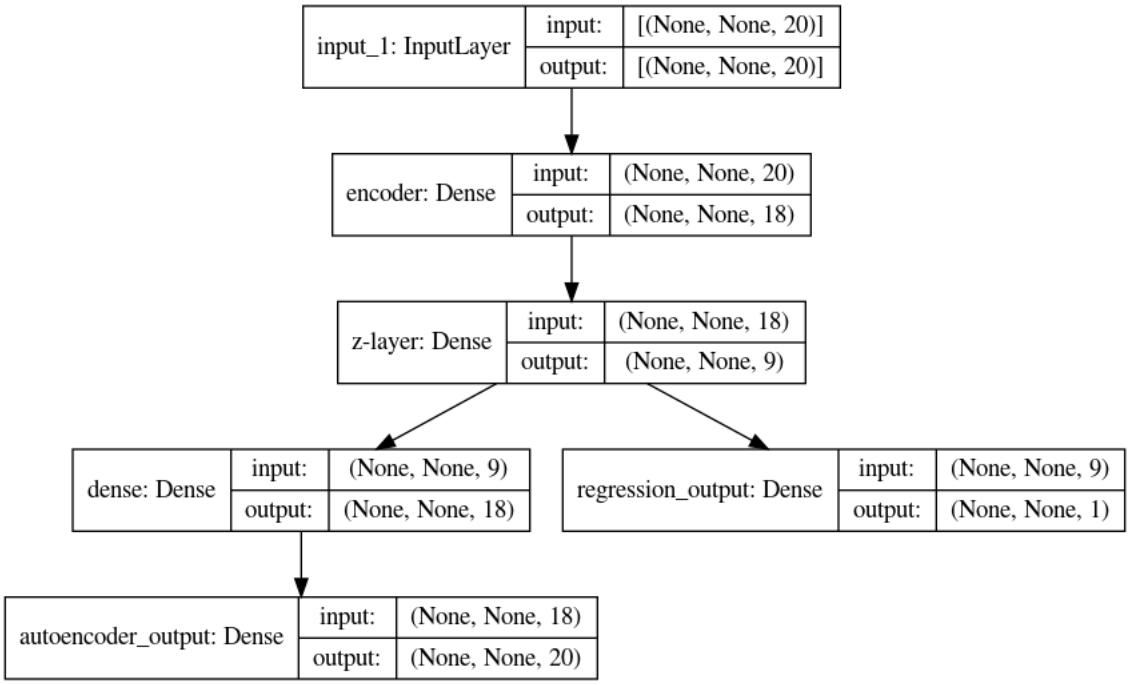


Figure 3.2: Diagram of rRT+AE architecture

feature extractor consists of the "encoder" layer and the "z-layer". From the feature extractor, the model splits into two branches. The left-branch of the model is the "decoder" branch, and is responsible for reconstructing the input of the model. This branch is only present during the first "representation" stage of model training, and is removed for the second stage. The right output branch of the model, the regression head, is responsible for accurately predicting intensity values that match the actual target peak intensity for each input. This branch is kept in both stages of training, though the weights for the regression head are reinitialized after the first-stage of training has been completed.

After the first stage of training, the weights of the feature extractor layers are frozen. This means that the values for these weights in the model will not be adjusted in subsequent epochs of training. As was mentioned previously, the weights in the regression head layers are reinitialized, while the decoder branch is removed from the

model. The purpose of weight reinitialization for the regression head is so that it can be trained specifically for the regression task on oversampled data using the representations learned from the first-stage. The architecture during the second-stage of model training, therefore, looks like figure 3.3.

Typically, the second stage of training for the cRT model is performed on the class-balanced dataset. Because our problem is a regression problem, we do not have classes to divide the inputs into. Instead, we train the second-stage architecture on an input training distribution where the SEP events and possibly high-width and large-speed non-SEP events are oversampled. We refer to this distribution, henceforth, as the "oversampled distribution".

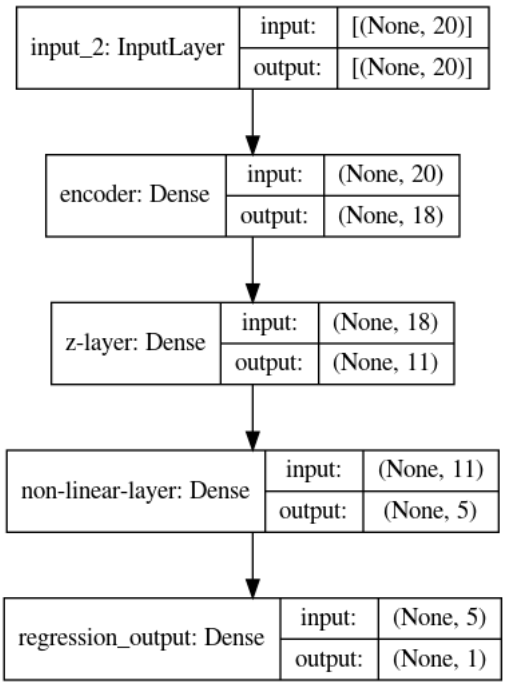


Figure 3.3: Diagram of rRT architecture with decoder branch removed

3.2.4 Adaptive Calibration

In addition to the rRT+AE architecture described in the previous section, we also adapt the adaptive-calibration technique that was also described in 2.2.1 for our regression task. Because our task is a regression problem, we do not have scores for each class. Instead, our model outputs a single regression value. To modify the adaptive calibration network for our own problem, we reasoned that we could combine the scores from a second-stage model trained on the oversampled distribution as well as one trained on the uniform distribution.

To obtain these scores, we train the first stage of an rRT+AE model on the uniform training distribution, as was described in the previous section. The architecture for this stage is shown in figure 3.2. As was the goal previously, the purpose of this stage of model training is for the feature extractor to learn effective representations of the entire data distribution before training specifically to optimize regression performance on the SEP instances. After first-stage training has completed, the weights in the feature extractor are frozen and the regression head is removed. The architecture of the model at this point is given by 3.4.

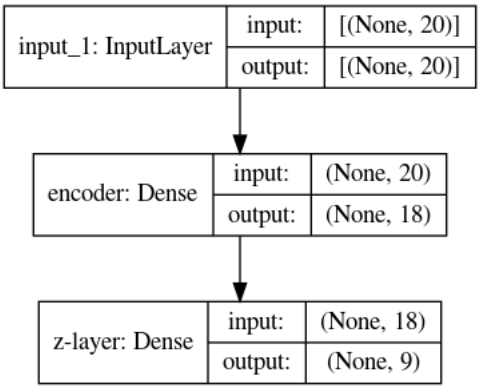


Figure 3.4: Diagram of feature extractor after first stage of training

The feature extractor trained in the first-stage is then used as the basis for two

second-stage regression models. These models are formed by adding two regression layers to the end of the feature extractor from the first model (the "non-linear" and "regression output" layers that are displayed in 3.5). One of these branches is trained again on the uniform training distribution, while the other is trained on an oversampled distribution. The goal of this stage of training is to train one regressor that performs well on the majority of instances in the uniform distribution, and another that performs well on important, minority instances that are overrepresented in the oversampled distribution. As mentioned previously, the weights of the feature extractor are frozen so that only weights in the regression heads of the two models are adjusted during training. After these two models are trained, the weights for the regression layers on the two models are themselves frozen, freezing the regression "score" on both the uniform and oversampled distributions.

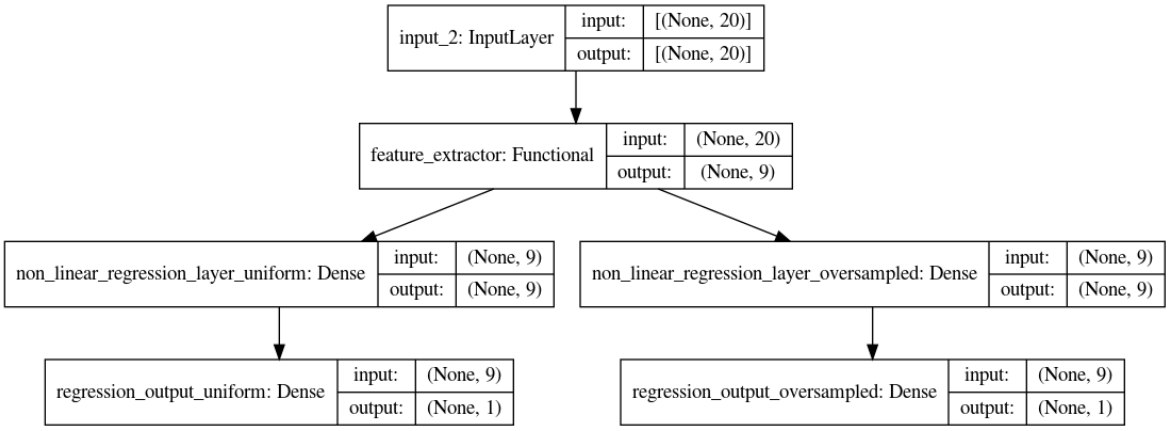


Figure 3.5: Diagram of regression model for second stage of training

In the final stage of training, a third branch is attached to the feature extractor, as shown in figure 3.6. This third layer learns to output a single $\sigma(z)$ value, which is a function of the representation vector z generated by the feature extractor for an individual instance. The purpose of this value is to weigh the regression outputs of the two models that were trained in second stage. This is intended to "calibrate" the

score of the model to the uniform or the oversampled score based on the features of the an individual instance. Thus, for a specific instance with a representation vector z, $\sigma(z)$ linearly combines the uniform regression output $s_{\text{Uniform}}(z)$ and the oversampled regression output $s_{\text{Oversampled}}(z)$ as follows.

$$\hat{y} = (1 - \sigma(z)) \cdot s_{\text{Uniform}}(z) + \sigma(z) \cdot s_{\text{Oversampled}}(z) \tag{3.1}$$

An implication of this equation is that the adaptive-calibration model is never able to predict scores that are above the maximum predicted value for the branch trained on the oversampled distribution or below those generated by the branch trained on the uniform distribution. Thus, the model is never be able to predict the right scores for events that are above the predicted intensity for the oversampled branch as well as below the predicted intensity for the uniform branch. We will discuss the implications of this formulation and their effects on model predictions in the results.

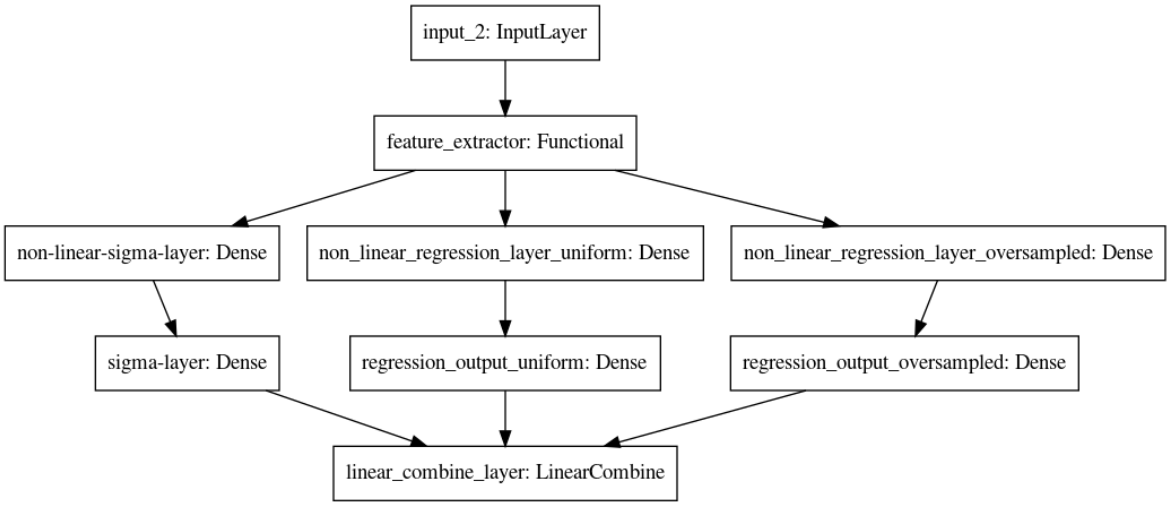


Figure 3.6: Diagram of adaptive calibration architecture for third-stage of training

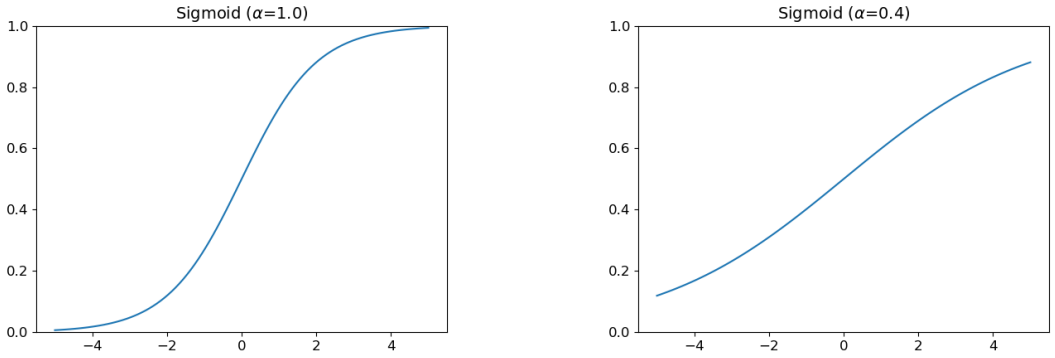
There are some properties that are desired for the σ function. Notably, we desire the values of σ to vary in a range of [0, 1]. In addition, we want to ensure that the values that the model learns to output vary greatly within this range, depending on

whether the features of the input instance match those of the majority of instances in the uniform distribution or the oversampled distribution. The learned σ values for instances in the dataset should not tend to drift closely to "0" or "1", but should vary in that range depending on the features of the specific instance. The typical sigmoid activation function, as shown in figure 3.7, tends to produce scores that are very close to '0' or '1' to be useful for this task. Consequently, most outputs from the feature extractor will generate scores either too close to '0' or '1', behavior we wish to avoid if we wanted the predicted score for each instance to be different. We therefore modified the sigmoid function to lower the slope of the function, thereby extending the domain where values that were not overly close to either '0' and '1' could be generated. Equation 3.2 shows the standard sigmoid function for some scalar input 'x', as well as our modified sigma function. We lowered the slope of this function by adding a coefficient 'α' to the exponential term in the denominator of the function. Hyperparameter tuning showed that using a coefficient of $\alpha = 0.4$ produced the best performance out of the adaptive calibration model. This produces a sigmoid with a gentler slope and qualities we desire for our σ function, as seen in figure 3.7.

$$\sigma(x) = \frac{1}{1 - e^{x}} \tag{3.2}$$

$$\sigma_{gentle}(x) = \frac{1}{1 - e^{\alpha x}} \tag{3.3}$$

For our problem, we expect that the sigma value should be more closely weighted to the uniform distribution when predicting intensity values for non-SEP instances (i.e., $\sigma(z_0)$ should be close to 0 when z_0 consists of features of a non-SEP instance), as those instances compose the vast-majority of instances in the uniform distribution. Conversely, we expect the sigma value to be weighted more closely towards the class-balanced distribution for SEP instances, as SEP events make up a greater proportion of the class-balanced distribution (i.e., $\sigma(z_1)$ should be close to 1 when z_1 consists of



(a) Standard-Sigmoid

(b) Gentle-Sigmoid

Figure 3.7: Standard sigmoid activation (left) and gentle sigmoid with reduced slope (right)

features for an SEP instance). Initially, we observed while training the model that the output σ values did not vary greatly over the test dataset, regardless of whether the input instance was a non-SEP event or an SEP event. In particular, we observed that the sigma values for each instance in the test set tended to coalesce towards either a '1' or '0', favoring one branch of the calibrated model over the other (usually, the favored branch was the uniform trained model, as that is the distribution of data the calibration stage is performed on). This behavior was undesirable, as we wished the model to output a variety of sigma values to generate different linear combinations of the uniform and class-balanced branch of the model. In particular, we wanted the sigma values of non-SEP and SEP instances to be different, and for the non-SEP sigma values to favor the uniform branch while those for the SEP instances should factor the oversampled branch. For this to work, the feature extractor for the model needs to produce representations that allow the sigma layer to differentiate SEP and non-SEP instances. We experimented with decoupling the regression feature extractor from the sigma branch and, instead, training a model to classify SEP and non-SEP instances in conjunction with the regression model trained during the first stage. The feature

extractor for this classification model was then frozen and used as the input for the sigma layers in our adaptive-calibration model. The reasoning for this was that features learned by the classification model may help σ layer of the adaptive-calibration model distinguish between the features of non-SEP and SEP events more effectively and, thus, place more weight on the score of the oversampled branch for SEP events and vice-versa. This schema seemed to generate a greater variety of sigma values and improve the predictive performance of the adaptive calibration model, and is the schema we use when presenting our results.

3.3 Evaluation Criterion

3.3.1 Regression Metrics

For regression tasks, we would like to measure how close the value predicted by the algorithm is to the true prediction target. There are a set of metrics that are suitable for this task. For this task, we use mean absolute error (MAE). Because the events that we are interested in, the SEP events, compose less than 2% of the entire dataset, calculating mean absolute error over all prediction targets does not give us a good indication of how well our algorithm performs on these instances. For this reason, we calculate the MAE of the SEP instances separately in addition to the MAE of combined isntances. We also calculate the pearson correlation (PCC) between the observed and predicted values of peak intensity for just SEP instances as well as all instances in the dataset that are not assigned a constant target of $\frac{10.0}{e^2}$ (i.e., SEP and elevated proton events). The pearson correlation is calculated between two sets of values, and is a measure of the linear correlation between the two sets. For a vector of values $\overrightarrow{x} = \{x_0, x_1, ..., x_n\}$ and $\overrightarrow{g} = \{g_0, g_1, ..., g_n\}$, the pearson correlation (PCC) is defined

as

$$PCC = \frac{\sum_{i=0}^{n}(x_i - \overline{x})(g_i - \overline{g})}{\sqrt{\sum_{i=0}^{n}(x_i - \overline{x})^2}\sqrt{\sum_{i=0}^{n}(g_i - \overline{g})^2}} \tag{3.4}$$

where $\overline{x} = \frac{1}{n}\sum_{i=0}^{n} x_i$ is the mean for set x. In our case, the vector $\overrightarrow{x}$ corresponds to the peak intensities predicted by our model, $\overrightarrow{y}_{\text{predicted}}$, and the vector $\overrightarrow{g}$ corresponds to the observed peak intensities $\overrightarrow{y}_{\text{observed}}$.

3.3.2 Classification Metrics

In addition to the set of regression metrics that were mentioned in the previous section, a set of classification metrics were also compiled for the peak intensity prediction task. While peak intensity prediction is a regression problem, the events in the training set can be divided into SEP events and non-SEP events based on their target peak intensity. Indeed, being able to determine whether or not an event is an SEP based on the peak intensity predicted by our model is of critical importance to its use in the real-world. Therefore, we track a set of classification metrics that provide an indication on how effectively our model is able to delineate SEP and non-SEP events based on their predicted peak-intensity. Events that have a peak intensity of greater than or equal to 10.0 pfu are labeled as positive SEP events, while those that are below that threshold are labeled as negatives (or non-SEP events). Thus, cases where both the observed and predicted peak intensities are above 10.0 are considered to be "true positives", while instances where the observed intensity is below but the predicted value is above 10.0 pfu are considered to be false positives. This set of metrics is important because our task consists of not only predicting the peak intensity of proton events accurately, but also in providing an indication of whether or not an event should be classified as an SEP. The confusion matrix that is used for our classification metrics is provided in figure 3.2.

Table 3.2: Confusion matrix with two classes

	Predicted Peak Intensity ≥ 10.0	Predicted Peak Intensity < 10.0
True Peak Intensity ≥ 10.0	True Positive	False Negative
True Peak Intensity < 10.0	False Positive	True Negative

With these confusion matrix categories, we can define a set of classification metrics for our problem. In particular, we use precision and recall to provide a measure on how effectively our algorithm is able to correctly predict SEP events. Letting TP be the number of true positives, FP be the number of false positives, TN be the number of true negatives, and FN be the number of false negatives, we define the precision and recall as,

$$precision = \frac{TP}{TP + FP} \tag{3.5}$$

$$recall = \frac{TP}{TP + FN} \tag{3.6}$$

Precision is a measure of the fraction of predicted positives that are correct, while recall is a measure of how many true positives were predicted correctly. We can get a measure of the algorithm's performance on both measurements and, thus, its overall classification performance using the harmonic mean of recall and precision, dubbed the F1-Score.

$$\text{F1} = 2 \times \frac{precision \times recall}{precision + recall} = \frac{TP}{TP + \frac{1}{2}(FP + FN)} \tag{3.7}$$

Another metric that is commonly used in the astrophysics community for binary classification problems is the true skill statistic (TSS), which is the difference between the true positive rate (TPR) and the false positive rate (FPR), and is used to determine the tradeoff of increasing the number of true positives, which typically results in an increase in false positives as well.

$$TSS = \frac{TP}{TP + FN} - \frac{FP}{FP + TN} = TPR - FPR \tag{3.8}$$

In our case, the TSS closely mirrors the value of the recall for each experiment, as the number of Negatives (non-SEPs) greatly outweighs the number of Positives (SEPs) in our problem. Therefore, the number of true negatives will usually be much larger than the other quantities in the confusion matrix, and the measured FPR will be relatively small compared to the TPR.

The final classification metric that was used to measure the performance of our model was the Heidke-Skill Score, which measures the model's ability to correctly forecast with respect to some reference value (in this case, the probability of a random correct prediction).

$$HSS = \frac{2(TP \cdot TN - TP \cdot FN)}{(TP + FP)(FP + TN) + (TP + FN)(FN + TN)} \tag{3.9}$$

The value of HSS can be any value in the range $(-\infty, 1]$. Scores that are > 0 indicate that the model produces predictions perform better than random chance, while scores ≤ 0 indicate that the predictions made by the model are worse than random. In our case, HSS typically does not differ much from the F1-score. This is because the HSS reduces to F1-score when the number of true negatives is much greater than other quantities in the confusion matrix, as proven in the appendix.

3.4 Evaluation Procedures

3.4.1 Partitioning the Dataset

An important component of machine learning tasks is determining how the accumulated data should be partitioned into training, validation, and test datasets. In the context of machine learning problems, the train partition is used as input to the machine learning model during training. The outputs of the model are then compared with the

actual target value of the input training data using some form of cost or loss criterion (e.g., mean-squared-error, mean-absolute-error, etc.). The value calculated by this loss criterion is then back-propagated through the different layers of the machine learning model. The weights of each of the models' layers are updated using batch stochastic gradient descent using equation 3.10.

$$\Delta w = -\eta \nabla L \tag{3.10}$$

Where w are the weights for the last layer of the model, L is the loss calculated for a batch of inputs, and η is the learning rate, a hyperparameter that is set before model training. Using larger learning rates means that the model may reach convergence more quickly, but it is also possible for the learning rate to be so large that the optimal solution cannot be reached using stochastic gradient descent, resulting in lower model performance. The validation set is used to evaluate the performance of the model during training. Usually, this is done by compiling a set of performance statistics for the validation partition after each epoch of model training. For our model, the regression metrics and classification metrics that were described in the two previous sections were kept track of for each epoch of model training. The most important use of the validation set is to prevent model 'overfitting' on the training data. Because the training data is all the input the model will see during training, weight adjustment will optimize model performance against training instances. This optimization runs the risk of lowering generalization performance on instances outside of the training set, a problem that is known as 'overfitting'. To prevent overfitting, model performance on the validation set is tracked during training. When performance on the validation set begins to drop due to overfitting on the training set, model training is halted. Finally, the test set is used to evaluate the final model performance after training. No additional training steps or adjustments to the model are made after performance is evaluated on

the test set, and the model's performance on instances in the test set is taken as an indication of its performance on real-world data. In our work, we used a 50-20-30% split to partition our data into the train, validation, and test sets respectively. An additional nuance with our problem was that, because SEP events made up such a small population of the entire dataset, they made up relatively few training instances as well. To ensure that the model could train on as many distinct SEP examples as possible, we combined the training and validation sets of data after the first round of training had completed and an optimal set of hyperparameters and number of training epochs had been determined.

3.4.1.1 Random-Stratified Partitioning

For the tasks presented in this work, two partitioning schemes were used to evaluate model performance. The first partitioning scheme was a variation on the random partition, which assigns data to the training, validation, and test sets using a uniform distribution to sample events (i.e., if the training set consists of 50% of the entire dataset, then each sample has an 50% chance of being selected for the training set, etc.). We soon realized that, due to the extreme scarcity of SEP events in the overall dataset, the naive random partition could assign greatly differing distributions of SEP events to the train, valid, and test partitions. For example, if there are three SEP events with extremely high intensity ($>$ 1000 pfu) that are all assigned to the test set, the model has no examples representative of these extreme events to learn from, causing the model to greatly underpredict the intensity of these important events. To ensure that the distribution of SEP events was relatively similar in the train, validation, and test sets, SEP events were first split into 10 bins based on their peak intensity. These bins were selected such that each bin contained a similar number of SEP events (4 per bin in our case). Events from these bins were then uniformly assigned to either the

train set, validation set, or test set based on the desired split (i.e., because we wanted to assign 50% of events to the initial train partition, we assigned two events per bin to this partition). This ensured that the distribution of peak intensities for SEP events was similar across partitions.

Another problem that occurred in the early stages of model training, related to the distribution of SEP events, but not entirely solved by making the distribution of SEP events similar between the different partitions, was the issue of false positives generated by the model. As defined in the previous section 3.3.2, false positives are non-SEP events that are predicted to have a peak intensity above the SEP-event threshold. These false positives are largely generated with non-SEP events that have similar features to SEP events. Because event width and speeds are highly correlated with peak intensity, non-SEP events that have high speeds or large-widths are more likely to generate false positives for our model, as was discussed in section 3.2.2. To mitigate this issue, we needed to ensure that the different partitions for our model also contained a similar distribution of high speed and large width non-SEP events. Ensuring that the distribution of these events was roughly consistent across partitions was more difficult than with SEP events, as we needed to ensure that the two-feature dimensions of speed and half-angle were kept consistent across each partition.

We delineate high-speed, large-width non-SEP events using 1000 $\frac{km}{s}$ and 45.0° as cutoffs, respectively. Using these cutoffs, we could divide the non-SEP events into four cases. Case A. consisted of events above the 1000 $\frac{km}{s}$ speed threshold and above the 45.0° width threshold. Case B consisted of high-speed non-SEP events that were above the high-speed threshold, but below the large-width threshold. Case C consisted of large-width non-SEP events that were above the large-width threshold, but below the high-speed threshold. Finally, case D consisted of events that were below the high-speed and large-width threshold. The number of non-SEP events in each category is

provided in table 3.3.

Table 3.3: Number of non-SEP instances per case

Case	Attributes	Number of Instances
Case A	Speed $\geq 1000\frac{km}{s}$ and Half-Angle $\geq 45°$	53
Case B	Speed $\geq 1000\frac{km}{s}$ and Half-Angle $< 45°$	47
Case C	Speed $< 1000\frac{km}{s}$ and Half-Angle $\geq 45°$	319
Case D	Speed $< 1000\frac{km}{s}$ and Half-Angle $< 45°$	1929

Events from case B and case C were sampled such that the distribution of events across the range of speed and width feature values were relatively equal. For case A, because the distribution of non-SEP events needed to be the same over two-dimensions, we implemented a two-dimensional stratified sampling procedure. This procedure consisted of splitting high-speed, large-width non-SEP events into bins with an equivalent number of events based on their speeds and widths. These bins were then split into regions consisting of a roughly equal number of bins in the two-dimensional feature space formed by plotting the high-speed, large-width non-SEP events by their speed and width. We can then use the regional-stratified sampling algorithm 1 to ensure that the distribution of high-speed and large-width non-SEPs is the same across the training, validation, and testing partitions.

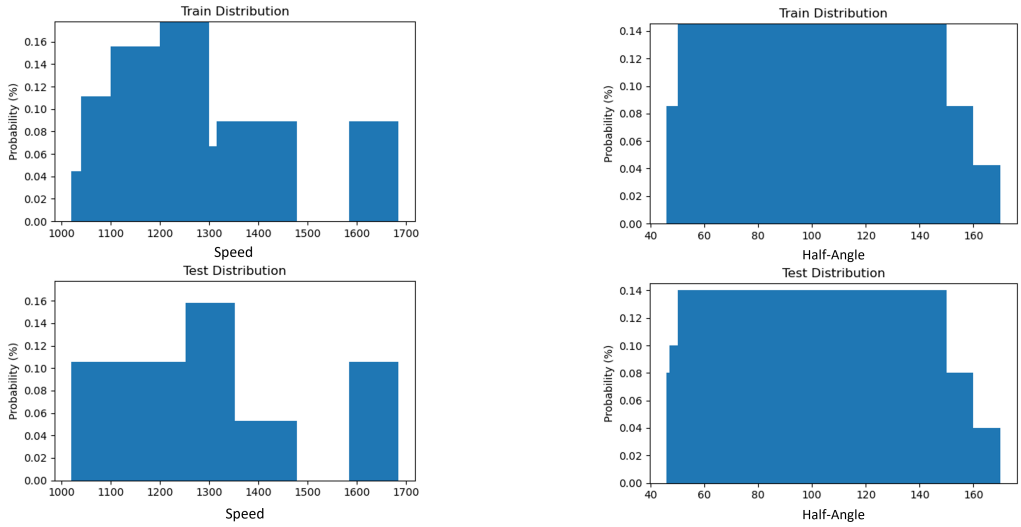
Algorithm 1 Regional Stratified Sampling

1: **procedure** RegionalStratifiedSampling
2: Allocate events into a number of bins so that each bin has approximately the same number of events
3: Calculate the number of total events we want to assign to the training partition with train_fraction · num(events)
4: Get the number of events we want to assign to the train partition with train_fraction · num_events_per_bin.
5: Round the number of events we want to assign to each bin to the nearest whole number.
6: Get the difference between the desired total number of train events minus the number of events in the rounded bins.
7: Split bins into regions so that an equal number of bins are assigend to each region.

```
    Get the fractional and rounded number of train events to assign each region.
    while the difference between desired number of events in the train distribution
        ↪ and number of assigned events is not zero: do
        Determine the binwise error between the number of events assigned and the
            ↪ fractional number of events desired.
        if only one bin has the largest error. then
            assign an event from this bin to the training distribution.
        else
            Calculate the region-wise error between the number of events assigned
            ↪ and the fractional number of events desired.
            Choose the region with the largest error and, from this region, choose
            ↪ the bin with the largest error to assign to the training distribution.
        end if
    end while
end procedure
```

The random, stratified sampled distribution, therefore, consists of a roughly equal distribution of SEP events across the three partitions according to intensity, as well as a similar distribution of non-SEP events according to width and speed. The partition scheme represents the "perfect" partition, where the distributions of minority SEP and non-SEP events are equal across the train, valid, and test partitions. We therefore expect optimal model performance on the test set instances after training.

The distribution of events after the 1-D regional stratified sampling algorithm is applied to case B and case C events are presented in figure 3.8. The left figure displays the distribution of Case B events in the training and test sets by their speed while the right figure displays the distribution of Case C events in the training and test sets by their half-angle. We see that the distribution of events is relatively similar in the training and test partitions for both categories of event. There is a slight imbalance in the partitions with low-speed events case B events, where proportionally more low-speed events are assigned to the train set than in the test-set. Conversely, a slightly larger proportion of high speed events are assigned to the test set compared to the training set. This imbalance should not affect model performance on the test set too



(a) Case B Distribution

(b) Case C Distribution

Figure 3.8: Percent of non-SEP events assigned to the training (top) and test (bottom) sets vs their feature values for case B (left) and case C (right) events

much, so long as high-speed case B events are still relatively well represented in the train set, which is the case.

Table 3.4: Number of case A events assigned to the training set by speed and half-angle using 2-D regional stratified sampling

		Speed $(\frac{km}{s})$			
		1015-1100	1100-1300	1300-1507	1507-2400
Half-Angle (°)	60.0-92.0	3	1	1	4
	50.0-60.0	2	3	2	3
	45.0-50.0	1	1	3	2
	41.0-45.0	3	6	2	0

Using 2-D stratified sampling, the number of case A events we assign per speed and half-angle bin to each partition are given in tables 3.4 and 3.5 for the train and test set, respectively. We see that the 2-D stratified regional assignment algorithm assigns, on a bin-to-bin basis, approximately one case A event to the test set for every 3 in the train set. In cases where there are fewer than three events in a bin, however, this ratio

Table 3.5: Number of case A events assigned to the test set by speed and half-angle using 2-D regional stratified sampling

		Speed ($\frac{km}{s}$)			
		1015-1100	1100-1300	1300-1507	1507-2400
Half-Angle (°)	60.0-92.0	1	0	0	2
	50.0-60.0	1	1	1	1
	45.0-50.0	1	0	2	1
	41.0-45.0	2	2	1	0

cannot be maintained. Therefore, in cases where there are only two events per bin, the algorithm assigns one to each partition . When there is only one event per bin, the event is assigned to the train set. In this way, we ensure that the distribution of important high-speed, large-width non-SEP events is relatively similar over the training and test sets and, in cases where we cannot assign a representative event to each set (i.e., cases with only one event per bin), the event is assigned to the training set so that the algorithm is still trained with these outlier events.

3.4.1.2 Chronological Partitioning

Naturally, in the real-world, this sort of "perfect" partition of events can not be expected. The random stratified sampling partition assumes that the entire distribution of SEP intensities as well as non-SEP speeds and widths are known. In reality, it is likely that our algorithm will encounter SEP events with higher intensity than those in the current dataset, as well as non-SEP events that have speeds and widths that are not currently contained within the dataset. Because the cycle of CME and SEP events is highly dependent on the 11-year solar cycle, the variation of features and peak intensities for events across this period will need to be modeled. In addition, we would like to partition the data using a method that mimics the the use case for the real-world deployment of our model, namely predicting the peak intensity of future events

using training instances from the past. To determine our model performance in this use-case, we need to use a partition scheme that is more likely to be representative of the real-world, where we cannot count that all of the events that our model will need to accurately predict will be well represented in the training set, and where the intensity, speed, and width of events encountered is expected to vary greatly over the solar cycle. To represent this real-world distribution, we also implement a chronological partitioning scheme. With this scheme, instances are sorted chronologically by DONKI CME time. The 50% earliest events are then assigned to the train partition. Then, the next 20% events that occur after the last event in the train set are assigned to the validation set. Finally, the last 30% events that occur chronologically in the DONKI catalog are assigned to the test set. This partition scheme is designed to better mimic our model's performance on events that may fall outside our initial training distribution.

3.4.2 Procedures for Training rRT+AE

To train the rRT+AE model, we first trained the first stage of the model (the "representation" learning portion of the training pipeline) for 1000 epochs with a learning rate of 1e-4 on the initial, uniform data distribution. The loss function that was used for both the regression and decoder branches of the model was mean-squared error. After the first stage of training, the weights of the feature extractor were frozen and the weights of the regression head were reinitialized. The model was then trained for an additional 500 epochs with a learning rate of 1e-3 on a data distribution where the SEP events were oversampled. During both stages of model training, a batch size of 128 was used. In addition, Adam optimization was used to determine the weight adjustments after each epoch. For each experiment configuration (e.g., changes in the oversampling rate in the second stage of training or any hyperparameter adjustment), five different models were trained using this same configuration. The metrics displayed in the results

tables for each partition are the averages of those five runs, while the prediction scatter plots shown in the analyses sections are taken from the median model in each training configuration as determined by F1-score.

3.4.3 Procedures for Training Adaptive Calibration

For the adaptive calibration models, the training proceeded similarly to the rRT+AE model in the first stage. A model with two branches, a regression head and an autoencoder branch, was trained on the uniform distribution of data. In addition to this regression model, however, a model with a classification head and an autoencoder branch were also trained to classify SEP and non-SEP instances by outputting a single score ('1' for SEP instances, '0' for non-SEP instances). Both of these first-stage models are trained for 1000 epochs using a learning rate of 1e-4 and a batch size of 128. For the second stage of training, the feature extractors for both the first-stage regression and classification models were frozen. Two different regression heads were initialized and attached to the frozen regression feature extractor to form two, separate regression models. These models were trained on the uniform distribution of data and the class-balanced distribution of data, respectively, for 500 epochs using a learning rate of 1e-3. Finally, the weights of the two regression models were frozen and the models were combined using the sigma equation explained in section 3.2.4. The layers that are responsible for learning the sigma value used the feature output by the frozen classification feature-extractor as input. A final, calibration stage of training was conducted for 1000 epochs using a learning rate of 1e-3 on the uniform distribution of data. As with the rRT+AE model, each experiment using the adaptive-calibration model consisted of five runs. The averaged metrics of the five runs are presented in the following tables of results, while the prediction scatter plots shown in the analyses sections are, again, taken from the median model in each configuration as determined

by F1-score.

3.5 Randomized Partition Results

3.5.1 rRT+AE Results

Table 3.6: Summary of regression results using rRT+AE and random-partitioning. An oversampling rate of p-q denotes p% SEP and q% high-speed and large-width non-SEP events

Oversampling Rate	SEP MAE	Combined MAE	SEP PCC	Non-Constant PCC
None	1.70	**0.08**	0.58	0.75
10-0%	1.35	0.16	0.61	**0.78**
10-10%	1.63	0.15	0.62	0.74
20-0%	1.30	0.23	0.62	0.77
20-10%	1.51	0.26	0.58	0.72
20-20%	1.50	0.21	0.53	0.76
30-0%	1.21	0.34	0.58	**0.78**
30-10%	1.48	0.31	0.59	0.72
30-20%	1.47	0.29	0.57	0.67
40-0%	**1.09**	0.41	0.62	0.77
40-10%	1.50	0.38	0.56	0.70
40-20%	1.59	0.35	0.58	0.72
40-30%	1.41	0.39	**0.69**	0.75
50-0%	1.26	0.39	0.59	0.73
50-10%	1.41	0.45	0.56	0.73
50-20%	1.39	0.51	0.56	0.72
50-30%	1.58	0.45	0.57	0.72

We see from the regression metrics posted in table 3.6 that oversampling SEP events, in general, improves the resulting SEP-MAE. We optimize performance on this metric by oversampling at a 40-0% oversampling rate. Looking at the classification metrics in 3.7, we see that this oversampling rate also optimizes the number of false-negatives and the TSS-score reported by the model and, with an average of 0.6 false-negatives, this implies that almost all SEP events are correctly classified at this oversampling rate. However, we also notice that a high number of false positives are generated at

Table 3.7: Summary of classification results using rRT+AE and random-partitioning. An oversampling rate of p-q denotes p% SEP and q% high-speed and large-width non-SEP events

Oversampling Rate	False Positives	False Negatives	F1-Score	TSS	HSS
None	**1.2**	4.2	0.68	0.58	0.68
10-0%	4.2	3.0	0.66	0.69	0.65
10-10%	3.6	3.2	0.70	0.68	0.70
20-0%	5.6	2.4	0.65	0.75	0.64
20-10%	5.0	2.8	0.65	0.71	0.64
20-20%	2.6	2.6	**0.74**	0.74	**0.73**
30-0%	8.8	1.4	0.63	0.85	0.62
30-10%	5.6	2.6	0.64	0.73	0.64
30-20%	4.0	3.2	0.66	0.67	0.65
40-0%	14.8	**0.6**	0.57	**0.92**	0.56
40-10%	10.0	2.6	0.54	0.73	0.53
40-20%	6.6	3.6	0.56	0.63	0.55
40-30%	5.2	3.8	0.58	0.61	0.58
50-0%	15.2	1.2	0.52	0.86	0.51
50-10%	9.2	1.8	0.60	0.81	0.59
50-20%	10.6	2.0	0.56	0.78	0.55
50-30%	8.4	2.4	0.59	0.75	0.58

this oversampling rate. Indeed, we see that, with an average of 14.8 false-positives, this oversampling rate generates 1.57 false alarms for every true detection, making it an unreliable SEP detector. Conversely, if we use a sampling rate of 20-20%, we see that the we can lower the average number of false negatives and false positives to 2.6, lending credence to the idea that some oversampling of high-speed and large-width non-SEP events tempers the algorithms tendency to predict high peak intensities for all events with those features. This oversampling rate optimizes the F1-score for the model, which is why we choose it as the optimal oversampling rate using rRT+AE.

3.5.2 Adaptive-Calibration Results

Table 3.8: Summary of regression results using adaptive-calibration and random-partitioning. An oversampling rate of p-q denotes p% SEP and q% high-speed and large-width non-SEP events for the oversampled branch of the network

Oversampling Rate	SEP MAE	Combined MAE	SEP PCC	Non-Constant PCC
10-0%	1.64	0.09	0.63	0.77
10-10%	1.62	0.09	0.60	0.79
20-0%	1.55	0.09	**0.67**	**0.81**
20-10%	1.60	0.10	0.61	0.77
20-20%	1.60	0.09	0.61	0.77
30-0%	1.61	**0.08**	0.59	0.76
30-10%	1.59	**0.08**	0.61	0.79
30-20%	1.59	0.09	0.61	0.79
30-30%	1.50	0.09	0.65	0.79
40-0%	1.59	**0.08**	0.63	0.79
40-10%	1.56	**0.08**	0.63	0.79
40-20%	1.49	0.09	**0.67**	0.80
40-30%	**1.46**	0.10	0.65	0.79
50-0%	1.67	**0.08**	0.63	0.79
50-10%	1.63	**0.08**	0.63	0.79
50-20%	1.63	0.09	0.61	0.78
50-30%	1.61	0.09	0.61	0.79

For the adaptive calibration model, regression results are presented in table 3.8, while classification results are presented in 3.9. Experiments with the adaptive-calibration model show is unable to achieve the lowest SEP-MAE that is achieved using rRT+AE. Indeed, the lowest SEP-MAE achieved by the adaptive-calibration model, 1.46 at an oversampling rate of 40-30%, is considerably higher than the lowest SEP-MAE of 1.09 achieved by the rRT+AE model. This makes sense, as the adaptive-calibration model is a linear combination of scores from the oversampled branch and the instance-balanced branches of the model, so the final regression prediction will be somewhere in between the low scores predicted by the uniform model and the high scores predicted by the class-balanced model. However, it is notable that the adaptive-calibration model is able to achieve the highest F1-score of all architectures displayed, 0.76, and is able to achieve

Table 3.9: Summary of classification results using adaptive-calibration and random-partitioning. An oversampling rate of p-q denotes p% SEP and q% high-speed and large-width non-SEP events

Oversampling Rate	False Positives	False Negatives	F1-Score	TSS	HSS
10-0%	2.2	4.4	0.63	0.56	0.62
10-10%	0.8	3.6	0.74	0.64	0.74
20-0%	**0.6**	3.6	0.75	0.64	0.75
20-10%	1.6	4.2	0.67	0.58	0.67
20-20%	1.0	3.8	0.72	0.62	0.72
30-0%	1.0	3.8	0.72	0.62	0.72
30-10%	1.6	3.6	0.71	0.62	0.70
30-20%	1.4	3.8	0.71	0.64	0.70
30-30%	2.0	3.8	0.68	0.62	0.68
40-0%	1.2	3.8	0.71	0.62	0.71
40-10%	**0.6**	3.8	0.74	0.62	0.74
40-20%	1.0	3.4	0.75	0.66	0.75
40-30%	0.8	3.4	**0.76**	0.66	**0.76**
50-0%	0.8	4.2	0.70	0.58	0.70
50-10%	0.6	3.6	0.75	0.64	0.75
50-20%	1.2	3.8	0.71	0.62	0.71
50-30%	1.0	**3.2**	**0.76**	**0.68**	**0.76**

this score at the oversampling rate that produced the lowest adaptive-calibration SEP-MAE, 40-30%. Furthermore, the model at this oversampling rate generates a lower average combined MAE of 0.10 than the rRT+AE models trained at 40-0% (combined MAE of 0.60) and 20-20% (combined MAE of 0.21), which contributes to the adaptive-calibration model's lower average number of false-positives. However, this oversampling rate with adaptive-calibration still produces a considerable number of false-negatives, 3.4 on average. Using an oversampling rate of 50-30%, we are able to achieve an identical F1-score, and reduce the average number of false-negatives down to 3.2. However, this is still higher than the average number of false negatives produced by the rRT+AE model with the highest F1-score, which, given the importance of detecting SEP events correctly, should be kept in mind when making a decision on the final model architecture to deploy.

3.5.3 Comparison with Single-Stage Neural Network

To determine whether there was any additional benefit to using additional stages to split the representation, regression (and calibration) learning of the model, we also evaluated a "standard" neural network that was trained with only a single-stage stage. To keep the comparison between the techniques that have been previously discussed and single-stage neural network training free from other variables such as architecture and hyperparameters used, we train the model using the same architecture as was presented in figure 3.5. Additionally, all hyperparameters such as learning rate and number of epochs used are kept consistent with the second-stage of training for the rRT+AE model. Various rates of oversampling for SEP and high-speed and large-width non-SEP events were performed to determine the optimal oversampling rate to use with the single-stage model architecture. The regression metrics for the neural network, rRT+AE model, and adaptive-calibration model that produced the highest F1-scores in each model category are presented in table 3.10, while the classification metrics are presented in table 3.11.

Table 3.10: Summary of regression results for model comparison with random-partitioning. An oversampling rate of p-q denotes p% SEP and q% high-speed and large-width non-SEPs

Model Architecture	Oversampling Rate	SEP MAE	Combined MAE	SEP PCC	Non-Constant PCC
single-stage network	10-10%	1.72	0.20	0.60	0.75
rRT+AE	20-20%	**1.50**	0.21	0.53	0.76
adaptive-calibration	50-30%	1.61	**0.09**	**0.61**	**0.79**

Table 3.11: Summary of classification results for model comparison with random-partitioning. An oversampling rate of p-q denotes p% SEP and q% high-speed and large-width non-SEP events

Model Architecture	Oversampling Rate	False Positives	False Negatives	F1-Score	TSS	HSS
single-stage network	10-10%	3.0	4.0	0.64	0.60	0.63
rRT+AE	20-20%	2.6	**2.6**	0.74	**0.74**	0.73
adaptive-calibration	50-30%	**1.0**	3.2	**0.76**	0.68	**0.76**

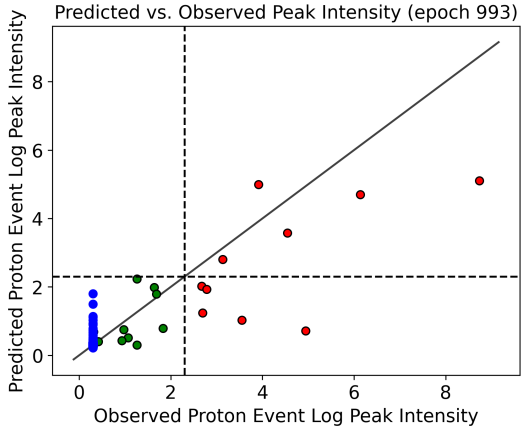
We see that the single-stage neural network fails to achieve the highest score in any of the metrics that were presented. In particular, the F1-score for the standard neural network is substantially lower than the best rRT+AE and adaptive-calibration models. Additionally, the standard neural network generates more false-positives and more false-negatives than either the rRT+AE model and the adaptive-calibration model, showing that there is some benefit to the additional stages of training that are performed, at least with with the random partitioned dataset.

3.6 Analysis of Random Partition Results

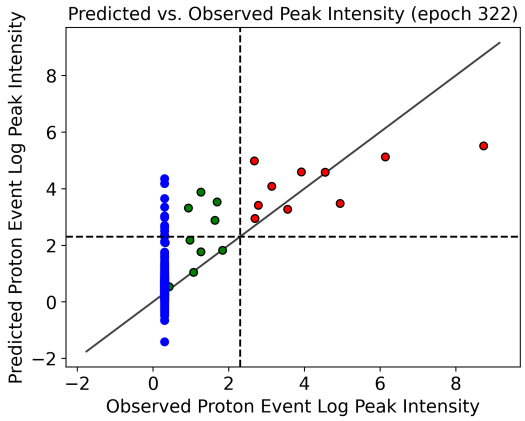
3.6.1 Analysis of rRT+AE Results

The peak intensity predictions for the median model trained with no oversampling in the second stage and the best rRT+AE models in terms of mean absolute error and F1-score are presented in figure 3.9. The y-axis of this figure is the natural log of the predicted value for peak intensity, while the x-axis is the natural log of the observed value for that instance. The red points in the plot are SEP events, the green points are elevated proton events, and the blue points are background proton events. The solid diagonal line represents the line of equality, where the predicted value matches the true value. The dashed horizontal line represents the SEP prediction threshold at an intensity of 10.0 pfu, while the dashed vertical line shows the SEP threshold for observed peak intensity values.

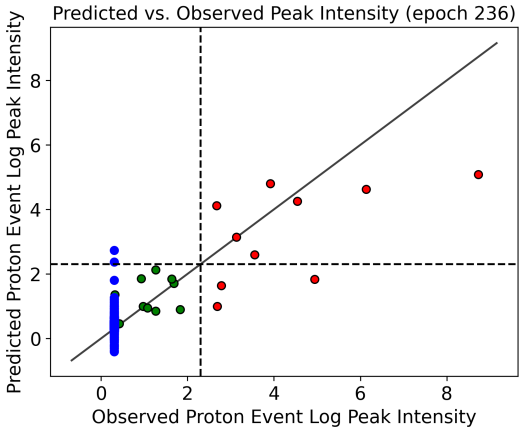
Comparing the prediction plots for no-oversampling and oversampling at a rate of 40-0%, we see that far more events cross the horizontal SEP prediction threshold at higher levels of SEP oversampling, corraborating our findings in section 3.5.1. This goes for SEP and non-SEP events, as we witness a far greater number of true positives



(a) No Oversampling



(b) 40-0% Oversampling



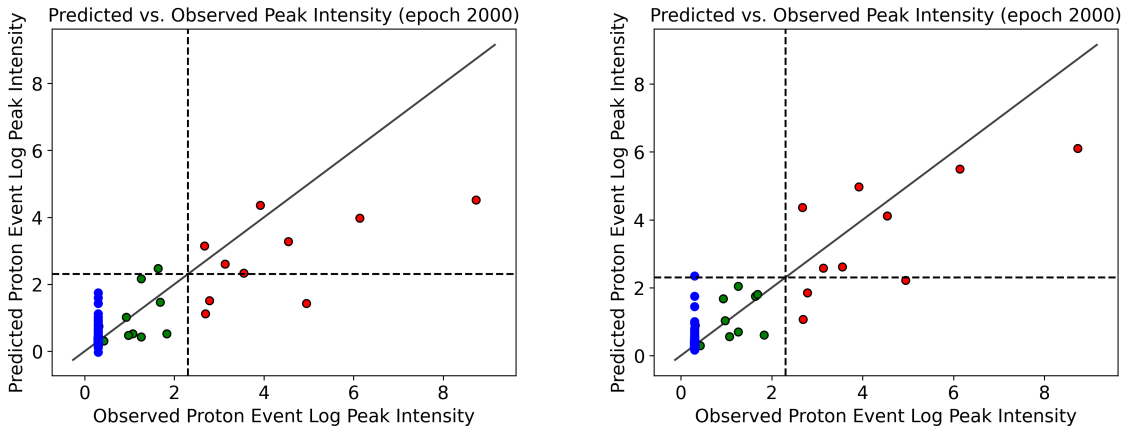
(c) 20-20% Oversampling

Figure 3.9: Log of peak intensity prediction scatter plots for varying oversampling rates using rRT+AE (random partition).

in the upper right quadrant as well as false-positives in the upper left quadrant. We can conclude that the model at an oversampling rate of 40-0% will a higher rate of true detections than with no oversampling, but will also introduce considerable more false alarms, which can make the predictions produced by the model unreliable. Introducing some degree of oversampling of high-speed and large-width non-SEP events can help mitigate the model's tendency to produce false-positives at higher oversampling rates, while still reducing the number of false-negatives produced by the model. We can see this trend by looking at the prediction plot for the median model trained with 20-20%

oversampling, which contains fewer false-negatives in the bottom right quadrant than the model trained with no-oversampling.

3.6.2 Analysis of Adaptive-Calibration Results



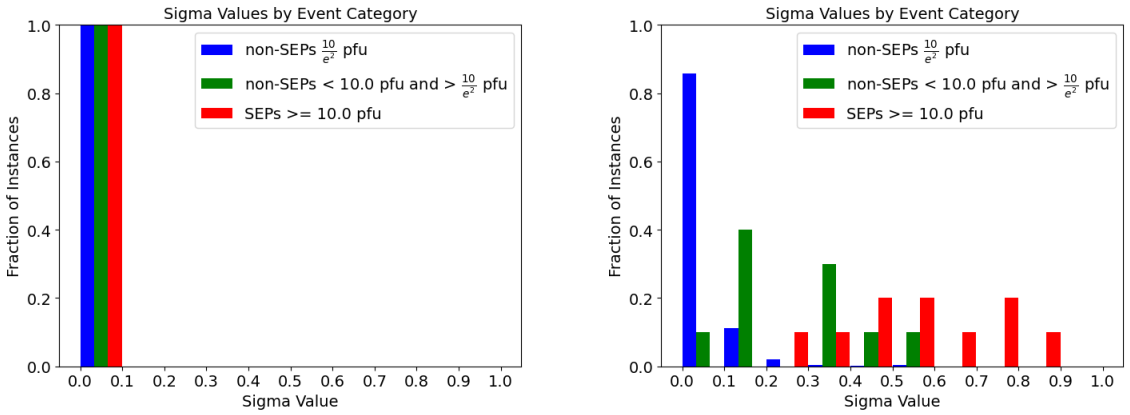
(a) 40-30%

(b) 50-30% Oversampling

Figure 3.10: Peak intensity prediction scatter plots for adaptive-calibration models with best F1-scores on random partition

The prediction plots for the natural log of the peak intensity for the median adaptive-calibration models at the oversampling rates producing the best F1-scores are presented in figure 3.10. We note that, the model trained at an oversampling rate of 50-30% tends to produce higher scores for all events than the model trained at an oversampling rate of 40-30%, which correlates with the greater number of SEP events used in the first oversampled distribution. We also notice that, relative to the prediction plots for the rRT+AE architectures, higher predictions for the SEP events can be generated without subsequently raising the predictions for non-SEP events as much, contributing to fewer false-positives.

To gain a better perspective on the inner-workings of the adaptive-calibration model, we plot the sigma values generated for instances in the test set for the models



(a) 40-30%

(b) 50-30% Oversampling

Figure 3.11: Fraction of random-partition test-set instances in each event category with with their associated sigma values for model sampled at 40-30% (left) and 50-30% (right). The x-axis is the sigma-value assigned to the event while the y-axis is the proportion of events from each category with that sigma value. As in the scatter plots, red is for SEP events, green is for non-SEP elevated events, and blue are constant-target events.

trained at 40-30% and 50-30% rates of oversampling. This plot is presented in figure 3.11. For the model trained at 50-30% oversampling, we see that the sigma values that are learned for non-SEP instances are all close to 0.0, weighting the output of the model towards the uniform distribution for these instances. For SEP instances, we see that the sigma-value is more varied, weighting some towards the uniform distribution. However, the majority of SEP instances have at least some sigma weighting towards the class-balanced distribution, which is expected. In contrast, the sigma scores learned for the 40-30% are all weighted towards the uniform distribution, which may be contributing to the lower scores generated by this model. In addition, the greater proportion of non-SEP data in the oversampled training distribution may be causing the model to learn to output sigma values that are, in general, more closely weighted to the uniform distribution. It should be noted that this is the distribution of sigma scores for only the median model trained in this configuration. The averaged sigma scores for all five

models trained with this configuration are presented in figure 3.12.

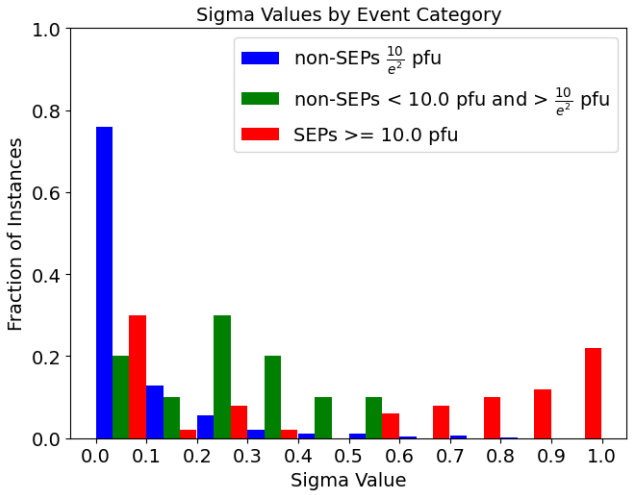


Figure 3.12: Average distribution of sigma values by event for models trained at 40-30% oversampling rate

We see that the distribution of sigma-scores for SEP events is far more favorable averaged over the five models than with the median model. However, the fact that the adaptive-calibration model can learn sigma functions that assign all events in the test set a score of '0' will need to be investigated in more detail. Additional hyperparameter tuning or even modifications to the network architecture itself may be needed to ensure that we always get a relatively favorable distribution of sigma-scores, which should help lower the number of false-negatives generated by our model.

3.6.3 Analysis of Common False Negatives and False Positives

Analysis showed that three SEP events in the test set were consistently predicted below the SEP threshold, and were subsequently labeled as false-negatives in the results. These events and the values for a set of important CME features obtained from the DONKI catalog for each event are presented in table 3.12, while a set of important features from the CDAW catalog are provided in 3.13. In addition, the events and their

predicted peak intensities using the median rRT+AE model oversampled at a rate of 20-20% as well as observed intensities are presented in table 3.14.

Table 3.12: DONKI feature values of common false negatives

	CME Start Time	SEP Start Time	Latitude (°)	Longitude (°)	Half-Angle (°)	Speed ($\frac{km}{s}$)
1	2011-03-21 02:54:00	2011-03-21 06:24:00	20	130	70	1000
2	2015-06-18 01:25:00	2015-06-18 04:30:00	-19	90	40	1720
3	2012-07-17 14:24:00	2012-07-17 15:25:00	-30	54	45	1100

Table 3.13: CDAW feature values of common false negatives

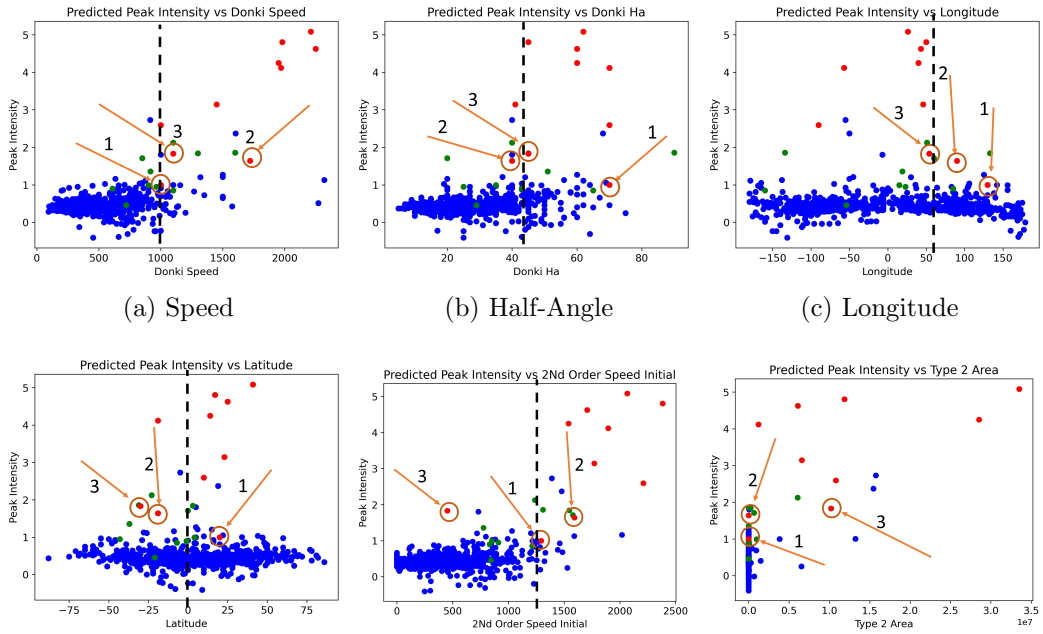
	CME Start Time	SEP Start Time	Initial 2nd Order Speed ($\frac{km}{s}$)	Type 2 Area
1	2011-03-21 02:54:00	2011-03-21 06:24:00	1294	0
2	2015-06-18 01:25:00	2015-06-18 04:30:00	1529	0
3	2012-07-17 14:24:00	2012-07-17 15:25:00	452	10,191,000

Table 3.14: Predicted and observed peak intensities for common false negatives

	CME Start Time	SEP Start Time	Predicted Peak Intensity (pfu)	Observed Peak Intensity (pfu)
1	2011-03-21 02:54:00	2011-03-21 06:24:00	2.71	14.75
2	2015-06-18 01:25:00	2015-06-18 04:30:00	5.17	16.18
3	2012-07-17 14:24:00	2012-07-17 15:25:00	6.24	140.51

The events at 2015-06-18 04:30:00 and the 2011-03-21 06:24:00 are relatively low intensity SEPs (< 20.0 pfu). We expect that the features composing these instances reflect that, causing the model to potentially report lower intensities than what were observed. The last event, or the 2012-07-17 15:25:00 event, has a fairly high observed peak intensity. It is therefore especially crucial to identify reasons for why our algorithm is underpredicting intensity values for this event. To better understand why our algorithms may be underpredicting intensities for these events, we present plots of predicted intensity vs feature value for the test set in figure 3.13 and plots of observed intensity vs feature value for the train set in figure 3.14.

The false-negatives in tables 3.12, 3.13, and 3.14 are outlined in orange in the feature plots provided by figure 3.13. We see that the 2011-03-21 06:24:00 event (labeled '1' in the plots) is an outlier in terms of longitude, having a longitude that is much higher



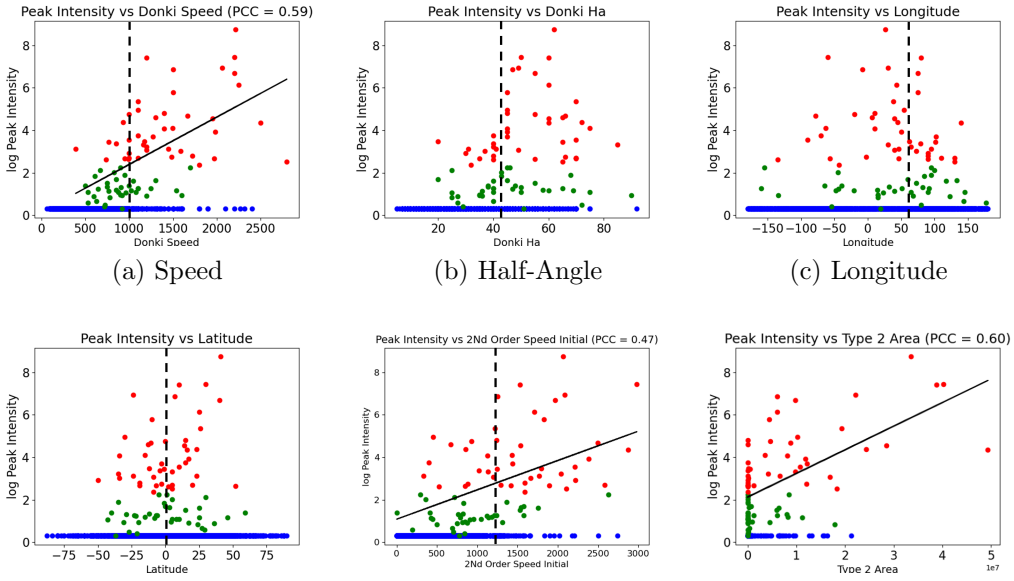
(a) Speed (b) Half-Angle (c) Longitude

(d) Latitude (e) Initial 2nd Order Speed (f) Type 2 Area

Figure 3.13: Log of predicted intensity vs feature values for instances in the test set using rRT+AE oversampled at 20-20%. For half-angle, speed, and initial 2nd order speed, the threshold for which values above are highly correlated with elevated intensities is marked with a dashed black lione. For longitude and latitude, the intensity peaks are denoted with a dashed black line as well. The false-negatives are outlined in orange and are labeled according to the row number of the event in tables 3.12, 3.13, and 3.14

.

than ideal (greater than 57°, the longitudinal direction of the IMF line with the Earth). Additionally, both the speed and the Type 2 Area for this event are relatively small compared to other elevated events. From the plot of observed intensities vs feature values given in figure 3.14, we see that higher values for speed and Type 2 Area are correlated with increasing observed intensity. Therefore, the low scores for the 2011-03-21 06:24:00 event in both feature categories could be contributing to the lower predicted intensity. For the 2015-06-18 04:30:00 (labeled '2' in the plots), we see that, in contrast to the previous event, the speed and longitude are favorable, but the Type 2 Area for the event is low. This is potentially why the predicted intensity for this event is



(a) Speed (b) Half-Angle (c) Longitude

(d) Latitude (e) Initial 2nd Order Speed (f) Type 2 Area

Figure 3.14: Log of observed intensity vs feature values for instances in the training set. For half-angle, speed, and initial 2nd order speed, the threshold for which values above are highly correlated with elevated intensities is marked with a dashed black lione. For longitude and latitude, the intensity peaks are denoted with a dashed black line as well. The solid line represents the best fit line for plotting observed intensity vs feature value, with pearson correlation between these two sets of values given as 'PCC' in the title of each plot.

.

elevated, but still below the SEP threshold. Finally, the 2012-07-17 15:25:00 event (labeled '3' in the plots) has favorable feature values for all of the features obtained from the DONKI catalog and has a high Type 2 Area. However, the second order initial speed for the event is much lower than for the other two events and for all other SEP events in the test set. Given that there is a fairly high correlation between increasing second order speed and observed intensity, as can be seen in figure 3.14 as well as from the positive pearson correlation of 0.49 between observed intensity and values for this feature, the low value for this feature might be the reason the prediction the 2012-07-17 15:25:00 event is so low.

In addition to the three false negatives that were just discussed, several false posi-

tives were generated by our models as well. It is important to examine the features of these events as well, to get a better understanding of what feature values are likely to generate false alarms in the future. The set of false positives was not consistent over all models that we've analyzed thus far, so we will restrict our analysis to the rRT+AE model trained with 20-20% oversampling. These false positives and the values for a select group of important features sourced from the DONKI catalog are provided in table 3.15, while a set of important CDAW features are provided in table 3.16. The predicted and observed intensities for these false positives are given in table 3.17.

Table 3.15: False positives generated by rRT+AE with 20-20% oversampling and their associated DONKI feature values

CME Start Time	Latitude (°)	Longitude (°)	Half-Angle (°)	Speed ($\frac{km}{s}$)
2014-04-02 13:55:00	19	-50	68	1604
2011-05-29 21:30:00	-5	-55	40	915

Table 3.16: False positives generated by rRT+AE with 20-20% oversampling and their associated CDAW feature values

CME Start Time	Initial Second Order Speed ($\frac{km}{s}$)	Type II Radio Area
2014-04-02 13:55:00	1478	15,445,520
2011-05-29 21:30:00	1390	15,750,900

Table 3.17: False positives generated by rRT+AE with 20-20% oversampling as well as their predicted and observed intensities

CME Start Time	Predicted Peak Intensity (pfu)	Observed Peak Intensity (pfu)
2014-04-02 13:55:00	10.71	$\frac{10.0}{e^2}$
2011-05-29 21:30:00	15.31	$\frac{10.0}{e^2}$

For the event on 2014-04-02 13:55:00, we note that the speed and width of the event are high, but the direction of the event is badly correlated with peak intensity. However, the event also has a high type II radio area, which is also highly correlated with elevated peak intensities. We also note that the intensity predicted for this event,

10.71 pfu, is barely over the SEP threshold, so this would be considered a weaker SEP event. It's likely that the values for half-width, speed, and type II area for this instance are driving up the predicted intensity for the event, even if the other important features such as direction are less correlated with elevated peak intensities.

For the event on 2011-05-20 21:30:00, all of the values for the DONKI features, with the exception of half-angle, are badly correlated with elevated peak intensity. The speed of the event is relatively low, and the direction is unfavorable. However, the second order speeds for the event are fairly high (including the initial second order speed, the fitted speed presented in the table, which has a value of 1390 $\frac{km}{s}$). In addition, the Type II radio area for this event is also relatively high. It's possible that the set of fitted speeds as well as the type II radio area are driving up the predicted intensity for this event. Like the previous event, the predicted intensity for this event is relatively low, 15.31 pfu, which means that this would be considered a weaker SEP event.

3.7 Chronological Partition Results

3.7.1 rRT+AE Results

Tables 3.18 and 3.19 show the test set results for training an rRT+AE model on the chronological partition using varying rates of oversampling. We see from the results that, in-general, increasing the oversampling of SEP events in the second-stage training distribution decreases the SEP-MAE, as was seen with random partitioning. In addition, we notice by looking at similar regions of SEP oversampling that the number of average false-positives goes down when some oversampling of high-speed, large-width non-SEP events is introduced (though the models trained with 30% SEP oversampling

Table 3.18: Summary of regression results for rRT+AE using chronological partitioning. An oversampling rate of p-q denotes p% SEP and q% high-speed and large-width non-SEP events

Oversampling Rate	SEP MAE	Combined MAE	SEP PCC	Non-Constant PCC
None	1.83	**0.08**	**-0.05**	**0.14**
10-0%	1.62	0.12	-0.26	-0.02
10-10%	1.65	0.14	-0.21	0.04
20-0%	1.66	0.19	-0.09	0.00
20-10%	**1.59**	0.18	-0.29	-0.07
30-0%	1.69	0.25	-0.17	-0.02
30-10%	1.66	0.20	-0.30	-0.11
30-20%	1.74	0.28	-0.40	-0.27
40-0%	1.67	0.28	-0.16	-0.07
40-10%	1.93	0.30	-0.33	-0.10
40-20%	1.72	0.25	-0.24	-0.11
40-30%	1.60	0.36	-0.09	0.02

Table 3.19: Summary of classification results for rRT+AE using chronological partitioning. An oversampling rate of p-q denotes p% SEP and q% high-speed and large-width non-SEP events

Oversampling Rate	False Positives	False Negatives	F1-Score	TSS	HSS
None	**1.20**	3.20	0.53	0.46	0.53
10-0%	3.80	2.00	0.58	0.66	0.57
10-10%	1.80	3.20	0.53	0.46	0.53
20-0%	5.20	1.80	0.55	0.69	0.54
20-10%	3.80	1.80	**0.60**	0.69	**0.60**
30-0%	4.80	1.60	0.58	0.73	0.58
30-10%	6.20	1.80	0.51	0.69	0.51
30-20%	4.60	2.40	0.51	0.59	0.51
40-0%	5.20	**1.40**	**0.60**	**0.76**	0.59
40-10%	5.20	2.20	0.51	0.63	0.50
40-20%	4.80	2.20	0.53	0.63	0.53
40-30%	4.40	2.20	0.54	0.63	0.53

seem to contradict this trend). Unlike with random partitioning, the best average SEP-MAE and F1-scores are exhibited at the same oversampling rate with chronological partitioning, at an oversampling rate of 20-10%. An equivalent F1-score is achieved at 40-0% oversampling, and with a lower average number of false-negatives (1.40 vs

1.80), but with a corresponding increase in average false-positives compared to the models trained at an oversampling rate of 20-10%.

We also observe that, in general, the scores for with chronological partitioning are poorer than those exhibited for the random partition. In fact, the pearson coefficient for SEP predictions is always negative, which indicates that the predictions and observed peak intensities for SEPs in the test-set are not linearly correlated. This is in-line our earlier observation for the random-stratified partitioned dataset, which is that this partitioning scheme represented a "best-case" partition scheme where the distribution of events in the train, validation, and test sets are as similar as possible. In contrast, chronological partitioning represents a more "realistic" scheme, where the distribution of events in the training and validation sets (or "past" events) are not guaranteed to represent the distribution in the test set (or "future" events, i.e. events that our model has not seen yet that we would need to generate accurate predictions for).

3.7.2 Adaptive-Calibration Results

Table 3.20: Summary of regression results for adaptive-calibration using chronological partitioning. An oversampling rate of p-q denotes p% SEP and q% high-speed and large-width non-SEP events for the oversampled distribution

Oversampling Rate	SEP MAE	Combined MAE	SEP PCC	Non-Constant PCC
10-0%	**1.75**	0.08	-0.24	0.09
10-10%	1.99	0.08	-0.17	0.06
20-0%	1.78	0.09	-0.10	0.15
20-10%	1.78	0.08	**-0.07**	**0.22**
30-0%	1.80	0.08	-0.08	0.15
30-10%	1.86	0.07	-0.42	-0.04
30-20%	2.08	0.08	-0.15	0.07
40-0%	1.76	0.07	-0.16	0.12
40-10%	1.88	**0.06**	-0.32	0.00
40-20%	1.79	0.08	-0.42	-0.01
40-30%	1.92	0.07	-0.24	0.03

Table 3.21: Summary of classification results for adaptive-calibration using chronological partitioning. An oversampling rate of p-q denotes p% SEP and q% high-speed and large-width non-SEP events for the oversampled distribution

Oversampling Rate	False Positives	False Negatives	F1-Score	TSS	HSS
10-0%	0.6	2.8	**0.65**	0.53	**0.65**
10-10%	1.2	3.6	0.49	0.40	0.49
20-0%	0.6	3.4	0.56	0.43	0.56
20-10%	**0.4**	3.6	0.54	0.40	0.54
30-0%	0.6	3.0	0.62	0.50	0.62
30-10%	0.6	3.0	0.62	0.50	0.62
30-20%	**0.4**	4.6	0.34	0.23	0.34
40-0%	1.0	2.8	0.63	0.53	0.63
40-10%	1.0	3.4	0.54	0.43	0.54
40-20%	1.4	**2.6**	0.64	**0.56**	0.63
40-30%	1.2	3.0	0.58	0.50	0.58

In-line with our previous observations with random partitioning (section 3.5.2), we observe that adaptive-calibration is unable to produce as low of an SEP-MAE as rRT+AE. Correspondingly, it is unable to produce as few false-negatives as the best rRT+AE models as well. However, adaptive-calibration produces the highest F1-score (0.65 versus 0.60), though this might be due to the high tradeoff in false-positives for a commisserate reduction in false-negatives, which will be discussed further in section 3.8.2.

3.7.3 Comparison with Single-Stage Neural Network

As with the randomly-partitioned data in section 3.5.3, we trained a single-stage neural network at varying oversampling rates to determine performance relative to the multi-stage techniques. The regression results from the best single-stage, rRT+AE, and adaptive-calibration models, as determined by F1-score, are presented in tables 3.22 and 3.23

We see that, unlike with the randomly partitioned data, the results on the chrono-

Table 3.22: Summary of regression results for model comparison with chronological partitioning. An oversampling rate of p-q denotes p% SEP and q% high-speed and large-width non-SEPs

Model Architecture	Oversampling Rate	SEP MAE	Combined MAE	SEP PCC	Non-Constant PCC
single-stage network	20-0%	1.70	0.13	-0.57	-0.26
rRT+AE	20-10%	**1.59**	0.18	-0.29	-0.07
adaptive-calibration	10-0%	1.75	**0.08**	**-0.24**	**0.09**

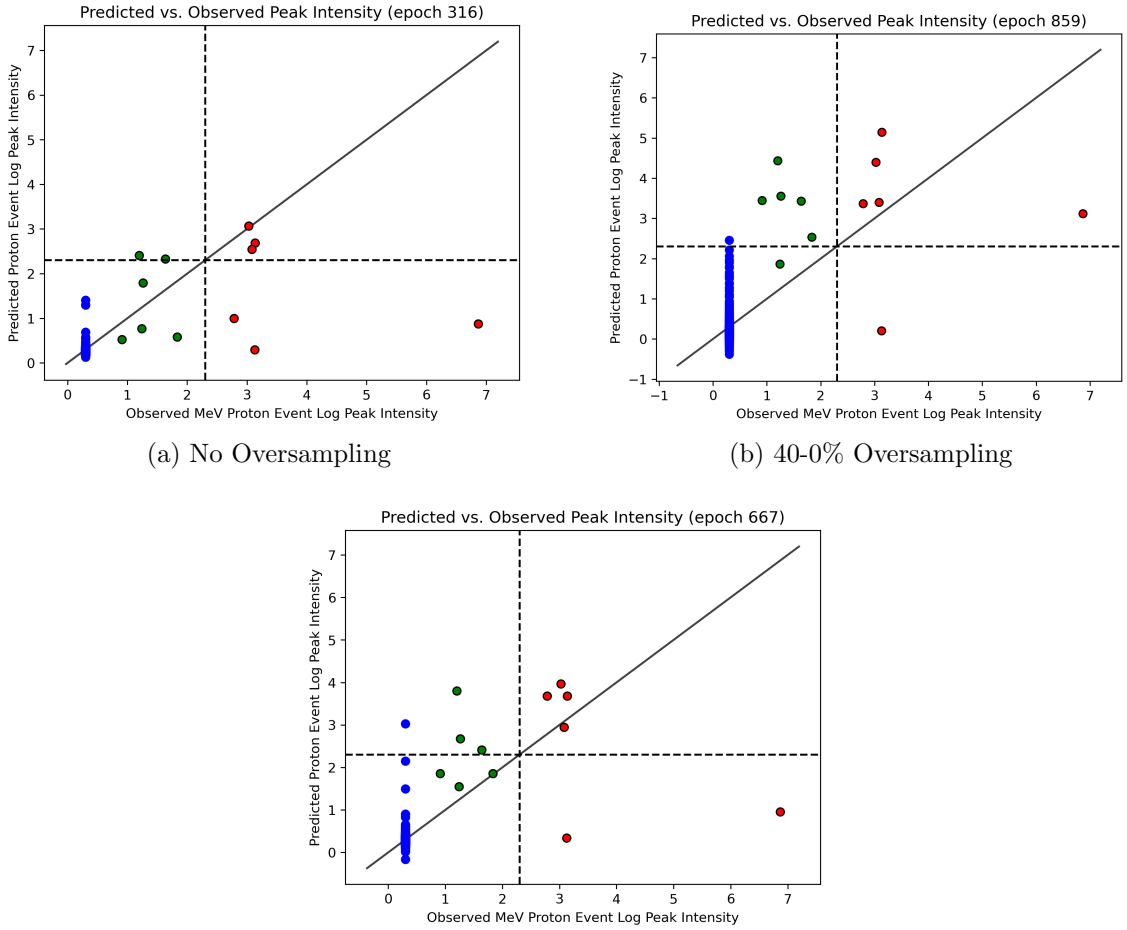
Table 3.23: Summary of classification results for model comparison with chronological partitioning. An oversampling rate of p-q denotes p% SEP and q% high-speed and large-width non-SEP events

Model Architecture	Oversampling Rate	False Positives	False Negatives	F1-Score	TSS	HSS
single-stage network	20-0%	2.4	2.0	**0.65**	0.66	0.64
rRT+AE	20-10%	3.8	**1.8**	0.60	**0.69**	0.60
adaptive-calibration	10-0%	**1.0**	3.2	**0.65**	0.63	**0.65**

logically partitioned data are mixed. In particular, the single-stage network is able to tie with the adaptive-calibration model in terms of F1-score, and is able to achieve that score with fewer false-negatives (with a corresponding rise in number of false-positives). In addition, it performs better than the rRT+AE model in terms of F1, though perfoms worse in terms of SEP-MAE. A potential reason for why the multi-stage models are not outperforming the single-stage model on the chronological partition is that the first-stage of training for both the rRT+AE and adaptive-calibration techniques assumes that the representations learned are consistent with the distribution of data in the test set. However, with chronological partitioning, the distribution of SEP events in particular is not guaranteed to be consistent over the training and test sets. Therefore, separating representation and regression learning into seperate stages may be less effective when the partitions contain dissimilar distributions of data. Further research will need to be conducted to determine how to measure the similarity of the distributions across the partitions and how it affects training of multi-stage networks.

3.8 Analysis of Chronological Partition Results

3.8.1 Analysis of rRT+AE Results



(a) No Oversampling

(b) 40-0% Oversampling

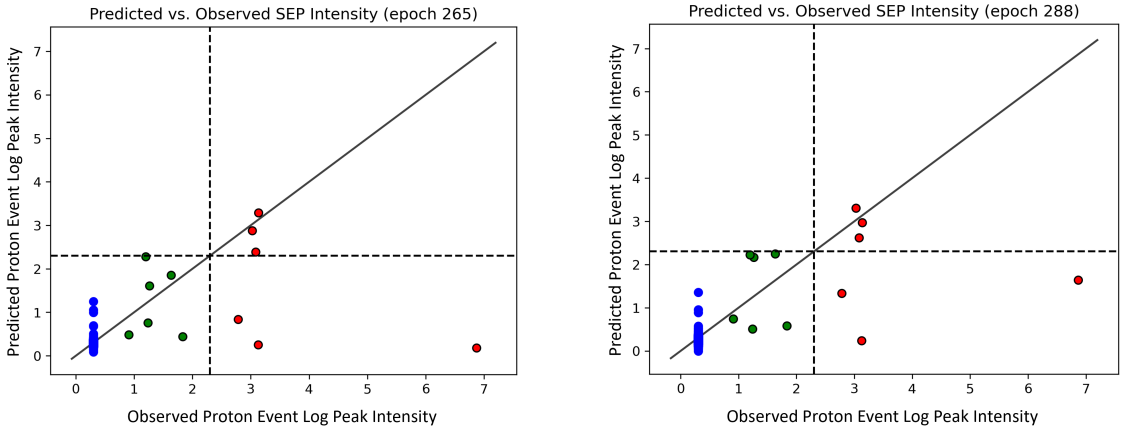
(c) 20-10% Oversampling

Figure 3.15: Peak intensity prediction scatter plots for rRT+AE with no oversampling (top left), an oversampling rate of 40-0% (top right), and an oversampling rate of 20-10% (bottom)

Figure 3.15 shows prediction scatter plots with the median model at no oversampling and the best sampling rate with respect to F1-Scores (40-0% and 20-10%) and SEP-MAE (20-10%). The model oversampling at 40-0% is included, as it minimizes

the false-negatives generated by the model. We see at high rates of SEP oversampling that the model is able to correctly detect most of the SEP events in the test set, as can be seen in the top-right quadrant of the scatter plot for 40-0% oversampling. However, the false positives (top-left quadrant) generated at this rate of oversampling are higher than with the other two oversampling rates displayed. Furthermore, a trend we see at all three exhibited oversampling rates is that the predicted intensities for the false-positives are almost the same as those predicted for the SEP events, a worrying trend. We further note that the false-negatives that are generated by each model (bottom right quadrant) are predicted far below the SEP intensity threshold. We will examine the features of these false-negatives in section 3.8.3 and provide analysis on why their predicted intensities might be so low.

3.8.2 Analysis of Adaptive-Calibration Results



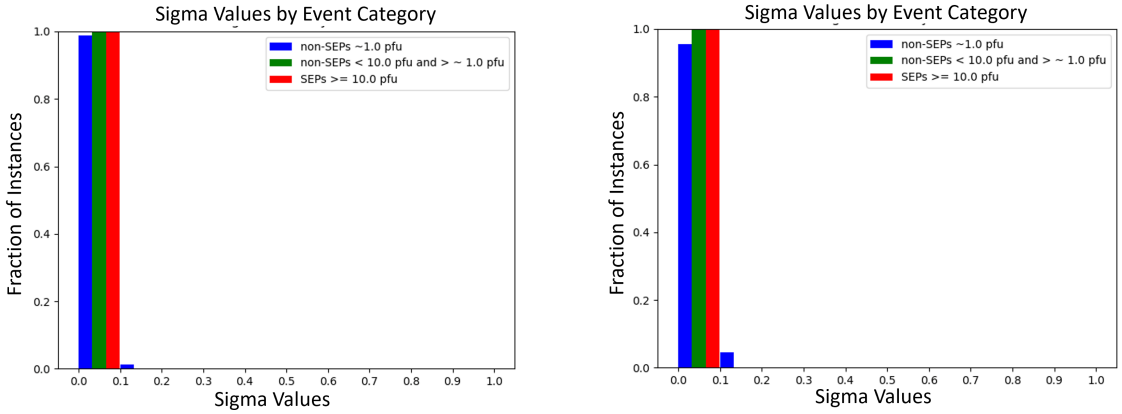
(a) 10-0% Oversampling

(b) 40-20% Oversampling

Figure 3.16: Peak intensity prediction scatter plots for adaptive-calibration models with best F1-scores on chronological partition

Figure 3.16 shows the plots with best runs based on F1-Score (10-0% sampling rate for the oversampled distribution) and false negatives (40-20% sampling rate for

the oversampled distribution). From the scatter plot, the lower rate of false negatives does not seem to be readily apparent, and we note that the adaptive-calibration model predicts the same events under threshold as the rRT+AE model with no oversampling. Indeed, performance for both oversampling rates seems to most closely mimic the rRT+AE model with no oversampling. Looking at the plotted sigma values for the adaptive-calibration models in figure 3.17, we see that the model is learning scores for sigma that favor the uniform distribution, which is why the adaptive-calibration models prediction scatter plots look so similar. Thus, the reason that this model may be achieving the highest F1 is that too many false-positives are being generated when reducing false-negatives, thereby reducing precision at a faster rate than recall is increasing. As was noted in the analysis for adaptive-calibration for the random-partition in section 3.6.2, further research will have to be conducted to determine how to bias the model to learn sigma values closer to 1.0 for SEP events.



(a) 10-0% Oversampling

(b) 40-20% Oversampling

Figure 3.17: Fraction of chronological partition test-set instances in each event category with with their associated sigma values for model sampled at 10-0% (left) and 40-20% (right) rates. The x-axis is the sigma-value assigned to the event while the y-axis is the proportion of events from each category with that sigma value. As in the scatter plots, red is for SEP events, green is for non-SEP elevated events, and blue are constant-target events.

3.8.3 Analysis of Common False Negatives and False Positives

We analyze the false negatives that were common to all models shown in the previous results section. Specifically, we analyze the predictions for the rRT+AE model that was oversampled at a 20-10% rate, as it had one of the highest F1-scores for that architecture as well as a decent balance of false-negatives and false-positives. The features of the false-negatives generated by this model are presented in table 3.24, while their predicted and observed peak intensities are presented in table 3.25.

Table 3.24: False negatives in the chronological partition and their feature values

CME Start Time	SEP Start Time	Latitude (°)	Longitude (°)	Half-Angle (°)	Speed ($\frac{km}{s}$)
2015-06-21 02:48:00	2015-06-21 19:54:00	7	-8	47	1501
2015-10-29 02:48:00	2015-10-29 03:00:00	-24	95	31	390

Table 3.25: False negatives in the chronological partition and their predicted / observed intensities

CME Start Time	SEP Start Time	Predicted Peak Intensity (pfu)	Observed Peak Intensity (pfu)
2015-06-21 02:48:00	2015-06-21 19:54:00	2.60	961.13
2015-10-29 02:48:00	2015-10-29 03:00:00	1.40	22.818

The event at 2015-06-21 02:48:00 has a relatively high speed of 1501 $\frac{km}{s}$, but has a highly unfavorable longitude, which could be contributing to its low predicted intensity. This event is particularly problematic, as the directional features of the event are not favorable, but the actual intensity of 961.13 pfu is high, meaning that it is an important event to identify. The PHTX File for this event is provided in figure 3.18.

We note that there seem to be a sequence of weak CMEs preceding the event, followed by a very strong eastward facing CME that the event is currently associated with (outlined in purple box) in bottom plot, including CMEs whose directions are similar to the associated event (the blue CMEs, which face eastward). It is possible that multiple of these CME events contributed to the final intensity of the SEP event. In that case, the solution to the problem may require further analysis on how to

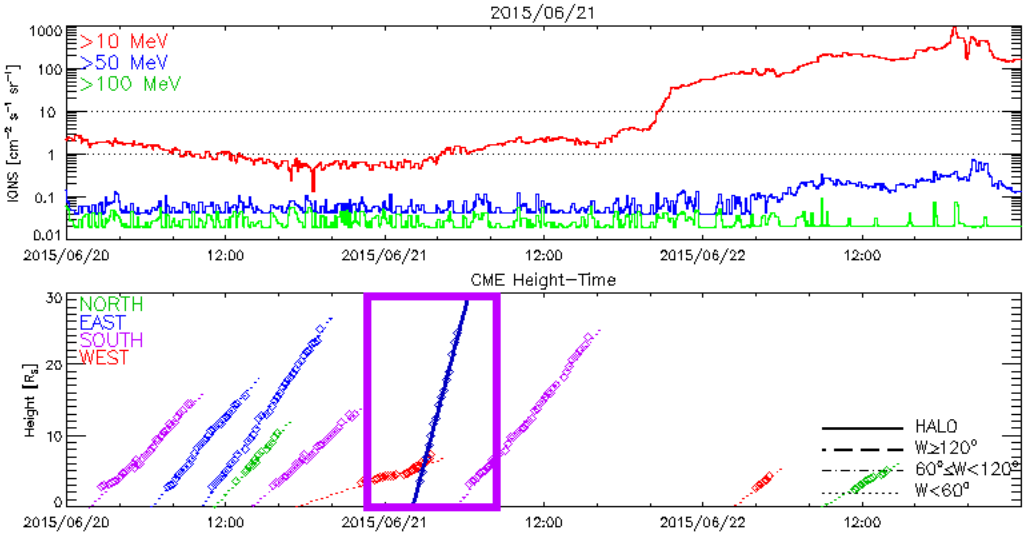


Figure 3.18: PHTX Plot for 2015-06-21 Event. The red graph in the top plot indicates the flux of 10 MeV protons over time, while CMEs are shown in the bottom plot. The CME that is associated to this SEP event is outlined with a purple box [1]

incorporate features of multiple contributing CMEs in the model's input to produce a single target value.

This problem applies to the last the event at 2015-10-29 02:48:00. The CME associated with this event has a speed of 390 $\frac{km}{s}$, which is very low for an SEP event (indeed, it is the lowest speed CME associated with an SEP event in our dataset). Analysis of the PHTX file for the event on the CDAW website indicates that this event was most likely caused by multiple, cumulative weak CME events rather than a single strong CME event. This PHTX plot is shown in figure 3.19.

Multiple CMEs occurring at the beginning of the day on 2015-10-29 and the previous day likely contributed to the rise in 10 MeV proton flux shown in the top plot, rather than the single CME that it is currently associated to (contained within the purple box in the bottom plot). Therefore, limiting the input features of this event to only one of those CMEs will result in the model heavily under-predicting the event's peak intensity. Future work can address the poor predictive performance of the model

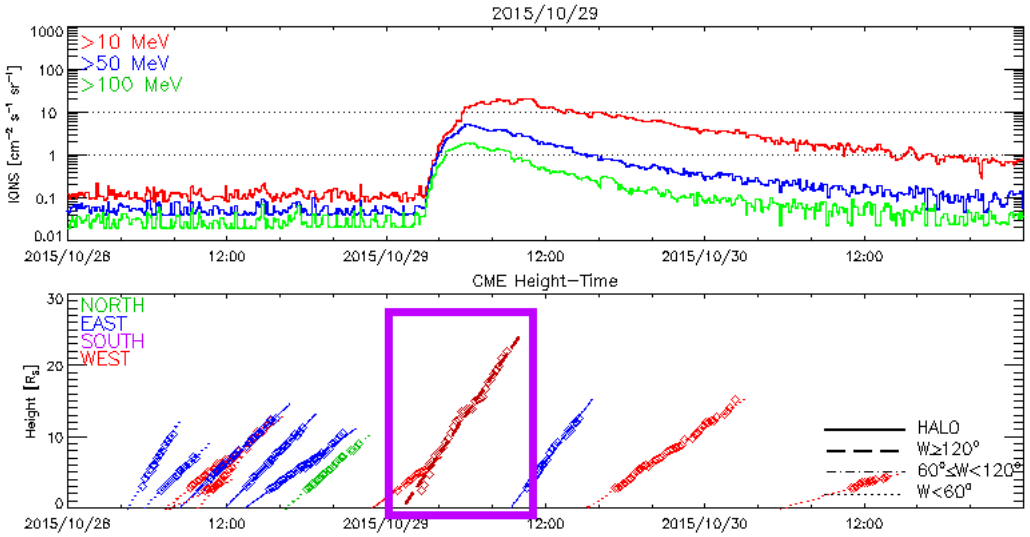


Figure 3.19: PHTX Plot for 2015-10-29 Event. The red graph in the top plot indicates the flux of 10 MeV protons over time. Multiple weak CMEs in the bottom plot may have contributed to the SEP event in the top plot, including the one that it is currently associated to (outlined with purple box) [2]

for multiple CME events by introducing new features that take into account the cumulative effect of multiple CME events, or by changing the inputs of the model to a temporal sequence that incorporates information about all of the cumulative CME events in the preceding day, such as a sequence of coronagraph frames.

In addition to the false negatives generated, we also examine the false positives that were generated by the rRT+AE model with 20-10% oversampling. The features of these false positives are presented in table 3.26, while the predicted and observed peak intensities for these false positives are presented in table 3.27

Table 3.26: False positives generated by rRT+AE with 20-10% oversampling and their feature values

CME Start Time	Latitude (°)	Longitude (°)	Half-Angle (°)	Speed ($\frac{km}{s}$)	Initial Second Order Speed ($\frac{km}{s}$)
2015-12-28 12:39:00	-15	14	58	850	1182
2015-09-20 18:12:00	-23	51	40	1100	1238
2015-06-14 04:24:00	-23	40	36	701	1005
2015-07-01 14:36:00	3	133	45	1300	1546

Three out of four of these events are proton events with elevated intensities, which

Table 3.27: False positives generated by rRT+AE with 20-10% oversampling and their predicted / observed intensities

CME Start Time	Predicted Peak Intensity (pfu)	Observed Peak Intensity (pfu)
2015-12-28 12:39:00	44.72	3.31
2015-09-20 18:12:00	14.52	3.53
2015-06-14 04:24:00	20.64	$\frac{10.0}{e^2}$
2015-07-01 14:36:00	11.20	5.13

are expected to have feature values that are more favorable for higher intensity predictions. Looking at the DONKI features for these events, we see that the directions of the 2015-09-20 18:12:00 and the 2015-06-14 04:24:00 events, while not direcly on the IMF line between the Sun and the Earth, do not fall too far from the optimal peak values of 57° and 0° for longitude and latitude, respectively. In addition, the speeds of the 2015-09-20 18:12:00 and 2015-07-01 14:36:00 are relatively high ($> 1000\frac{km}{s}$), while the half-angles for the 2015-12-28 12:39:00, 2015-09-20 18:12:00, and 2015-07-01 14:46:00 events are also fairly high ($\geq 40°$), which could be contributing to the higher intensity predicted by the model. Perhaps more revealing are the values for the second order speed fits provided by the CDAW catalog, which are relatively high for all of the listed events. As was noted in 3.6.3, values for the fitted speeds provided by the CDAW catalog have a high, positive correlation with observed intensity, which makes this feature value a driving factor for the generation of these false positives. These combinations of feature values are likely what's contributing to the model's overpredictions for these events. In the future, we could explore oversampling of non-SEP events based on these fitted speeds rather than DONKI speed or half-angle, or some combination thereof, to reduce the rate of false-positives across the random and chronological partition.

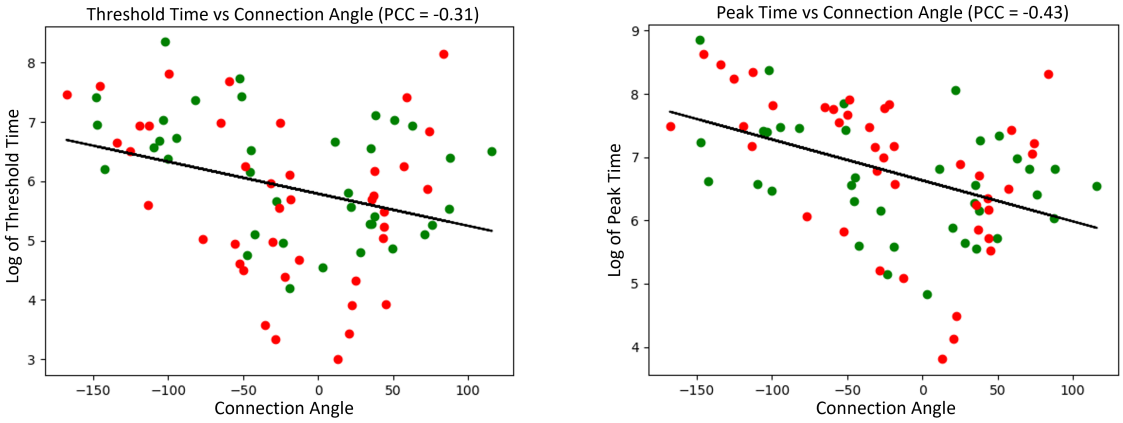
Chapter 4

Predicting Threshold and Peak Time

4.1 Problem

The next problem that was studied was predicting the time it took for an event to reach threshold and peak intensity, respectively. In addition to intensity prediction, being able to provide an estimate for event arrival is important for decision-makers, as it gives an indication on when mitigating measures need to be put into effect. Since these mitigating measures can affect the intended operation of important equipment (e.g., the shielding measures that are needed to protect satellite electronics prevents use of sensors onboard), we do not want to underpredict the time of SEP events by too much, as that increases the time that those expensive mitigating measures need to be maintained. Conversely, and with even more devastating consequences, we do not want to overpredict the times of SEP events, as that could result in the destruction of equipment and harm to human life.

4.1.1 Input Features



(a) Threshold Time vs Connection Angle

(b) Peak Time vs Connection Angle

Figure 4.1: Threshold and peak time vs connection angle

As with predicting peak intensity, we needed to choose a set of input features well correlated to the peak and threshold time prediction tasks. To this end, all of the features that were used in the previous problem were kept for threshold and peak time prediction. In addition to this set of features, an additional feature was derived to account for the strong correlation between the times of events and the orientation of the event with respect to the IMF line between the Sun and the Earth. The connection angle is defined as the angular distance between the direction of propagation for the CME and the IMF line between the Sun and the Earth, and therefore expect lower observed threshold and peak times to be highly correlated with a connection angle of 0°. Figure 4.1 shows the correlation between connection angle and both threshold and peak time. The pearson correlation calculated for both the threshold time and the peak time with respect to the connection angle are -0.31 and -0.43 respectively, which indicates that there is a slightly negative linear correlation between the connection angle and both the threshold and peak times, as can be seen with the best-fit line in each plot.

We can see that the lowest values for peak and threshold time culminate at around 0°, and then tend to increase for larger values of connection angle. To take advantage of this strong correlation between connection angle and observed threshold and peak times, we use it as an additional input feature for our model for the time prediction tasks.

4.1.2 Output Target Values

To perform the threshold and peak time prediction tasks, a continous regression value for threshold and peak times needed to be assigned to each CME instance in the input dataset. To obtain continuous values for the model to predict, the time of the event could not be predicted directly, as the output value for the model had to take some representation that could be represented by a floating point value. A convenient way to represent time in a floating point representation is as a time delta with respect to some reference time. This reference time could be defined at any point with respect to the event times, but different reference points could prove less meaningful for model training. For example, making the reference point a set date, such as the date of the first DONKI entry, means that the target value would increase chronologically no matter what the input features are, which means that any associations between the target value and the input features would be lost. Therefore, the reference point used to calculate the time values for the targets was set at the CME time, as recorded in the DONKI catalog. Our output target values were, therfore, defined as the number of minutes between the time of the CME and times where the intensity of the SEP event surpassed the threshold and reached their climax.

For peak time prediction, associating target values to CME instances associated to recorded proton events is straight forward. The Julian Day for the time that the peak intensity was recorded for these events was provided with the proton data. Converting

this value to UCT datetime and then calculating number of minutes between the CME time and this UCT peak datetime gives us our target value for this task. For CME instances that were not associated to a proton event, the concept of a "peak time" is nebulous, as there is no peak intensity to link these events to. We therefore assign these events an arbitrary, constant value. We was surmised that because these unassociated events are, in general, lower speed and have a weaker connection angle, the time for any weak proton event associated with these instances would be higher than for stronger proton events. Setting the timedelta for the unassociated events to this high value should allow the model to learn useful trends for the rest of the dataset. We therefore set this constant value by taking the mean time of all of the target values for the associated events and then adding five standard deviation to the mean, a timedelta far above what could be reasonably expected for an SEP or elevated proton event. This value, after rounding, is 8500.0 minutes. It should be noted that this arbitrary value could be reviewed in the future, so that we can choose a value is more realistic or conducive to model learning for SEP and non-SEP events that have "real" peak times. Indeed, we could even drop these events from the input dataset, as it can be assumed that, in general, these events will be detected below threshold by the peak intensity prediction model. These events would subsequently be removed from the set of inputs for the time prediction models.

For threshold time, calculating target values was similar to peak times. However, there is an additional nuance that needs to be considered when calculating threshold times for associated proton events, namely that the threshold that is used to distinguish between SEP and non-SEP elevated proton events will be different. For SEP events, the threshold time is set to the time that the proton intensity surpasses the 10.0 pfu threshold. Non-SEP events with elevated intensity never surpass this threshold by definition, so some other marker had to be chosen to signify the threshold time for

these events. In our case, we set the threshold time for these events to be the number of minutes after the CME event where the intensity for the associated proton event surpasses $\frac{10.0}{e^2}$ pfu. This target value should be analogous to the threshold time for SEP events, though, as with the arbitrary value assigned to unassociated CME events, this could be reviewed in the future to provide a more realistic target. For the unassociated non-SEP events, we perform the same calculation as with peak time. After rounding, this arbitrary constant value is around 4500.0 minutes.

4.2 Approach

4.2.1 Random Oversampling

As in the previous chapter, various rates of random-oversampling for SEP events were experimented with to improve model performance on these critical events. The correlation between SEP events and the time of event is less strong than with peak intensity, however. SEP events are guaranteed to have the strongest peak intensities by definition. In general, SEP events also tend to have the lowest threshold and peak times. This trend makes sense, as intensity of the event is highly correlated with its speed and direction, meaning that the time between the associated CME event and when the threshold and peak intensity at the sensors are measured will be faster. However, this is not always the case. As mentioned previously, the threshold for non-SEP elevated proton events is lower than for SEP events, meaning that the prerequisite intensity to reach this threshold can be met sooner. Therefore, SEP events do not necessarily have lower threshold and peak times than other elevated proton events, which means that we need to vary the oversampling rate for both SEP and elevated proton events to learn features to improve performance for events with low and high threshold and

peak times.

While there is some correlation between the speed and half-angle of the CME event and the resulting threshold and peak time (in particular, higher speeds generally correlate to lower threshold and peak times), the trend is not as strong as with peak intensity. Therefore, the high-speed and large-width non-SEP events in the dataset are not as relevant to the predictions made by a model trained to produce values for threshold and peak times. For this reason, there is little reason to perform oversampling of these types of events. Instead, we conduct varying oversampling rates for the non-SEP elevated proton events.

4.2.2 rRT+AE

The same regression techniques and model hyperparameters that were discussed in 3.2.3 are used for threshold and peak time prediction, without alteration.

4.2.3 Adaptive Calibration

The same regression techniques and model hyperparameters that were discussed in 3.2.4 were used, without alteration. It should be noted that the expected distribution of sigma values for the calibration stage of model training is different than for the peak intensity prediction task, as the delineation between target values for SEP events and non-SEP elevated events is less pronounced. Therefore, we expect that the distribution of SEP events with sigma values that favor the uniform distribution should be larger for time prediction than with peak intensity prediction. Likewise, we expect to see a greater proportion of non-SEP elevated events that have sigma values favoring the oversampled distribution.

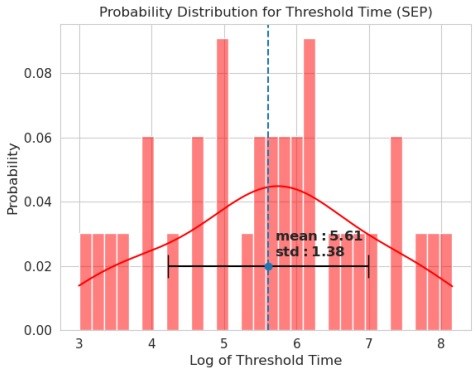
4.3 Evaluation Criterion

For predicting threshold and peak times, we used the same set of regression metrics as with peak intensity prediction. In particular, the mean absolute error for SEP events and over the entire dataset were calculated evaluate model performance on minority instances and over the entire distribution, respectively. Unlike with peak intensity prediction, classification metrics were not used for the time prediction tasks. This is because there is no delineation between SEP events and non-SEP events using threshold or peak time alone. In addition, we did not identify any use case for classifying proton events based on their threshold and peak time.

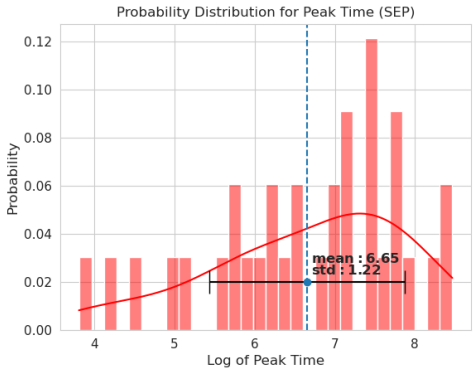
4.3.1 Comparing Regression Metrics for Peak and Threshold Time Prediction

It should be stated that the primary regression metric we use for evaluating performance of our algorithms on the time prediction tasks, SEP and combined mean absolute error, cannot be directly compared between different problem domains. I.e., the mean absolute error that is reported for threshold time cannot be directly compared to that of peak time to determine which problem domain our algorithms perform better at. We can see why this is the case by plotting the probability distribution of observable target values in the training set for each domain, as seen in figure 4.2.

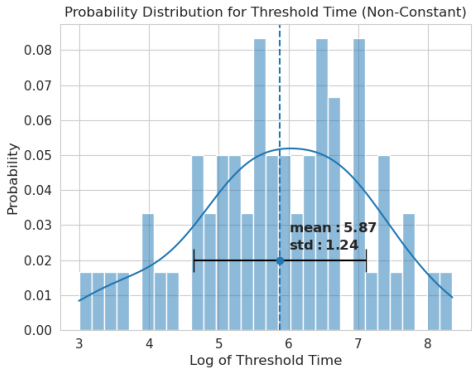
Looking at the probability distribution of observed SEP target values in the training set, provided in figure 4.2, we see that the distribution of values for peak time is shifted right relative to threshold values. More importantly, the standard deviation of prediction targets for threshold time is 1.38, while the standard deviation of prediction targets for peak time is only 1.19. This means that the mean absolute error for peak



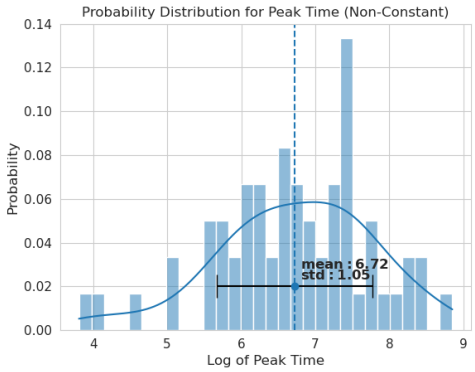
(a) Threshold Time (SEP)



(b) Peak Time (SEP)



(c) Threshold Time (Non-Constant)



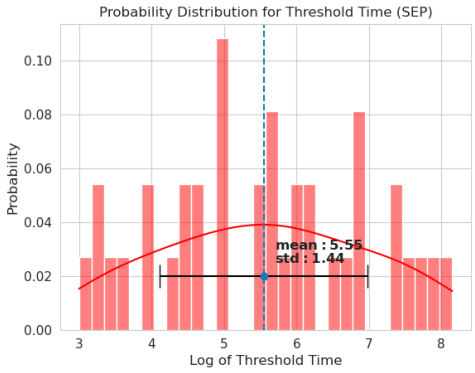
(d) Peak Time (Non-Constant)

Figure 4.2: Probability distribution of target values for threshold time (left) and peak time (right), split between SEP events (top) and all non-constant events (bottom) in the training set using random-stratified partitioning. The blue point in each plot signifies the mean of the distribution, while the black error bars signify the standard deviation.

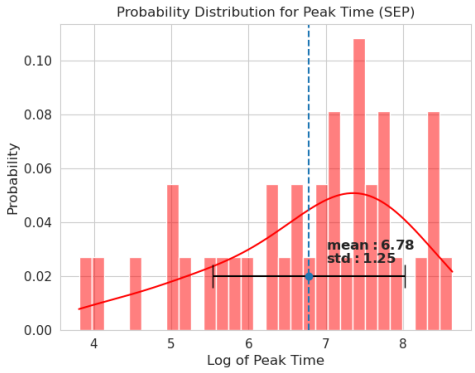
time prediction needs to be lower for the predicted value to still be within range of the most probable prediction values for the problem (i.e., a mean absolute error of > 1.19, when added or subtracted with the mean, 6.80, places the average prediction value in the 31.8% of outlier values in either tail of the distribution, while the same mean absolute error in threshold time prediction puts the prediction within the 68.2% most probable values in the observed distribution). This analysis assumes that the distribution of observed values is gaussian (which is not the case, as the distributions

of observed values in figure are asymmetric), but holds well enough to illustrate why the values for mean absolute error in both problem domains are not interchangeable. We can use the pearson correlation between observed and predicted values to provide some indication of how well our model is performing in one problem domain relative to the other (i.e., a problem domain where the pearson correlation between observed and predicted values is smaller is an indication that our model's performance is worse than in a domain where the pearson correlation is larger). Another metric that could be used to compare model performance across domains is the ratio of the mean absolute error with the standard deviation of the observed value, which gives an indication of the probability of the predicted values given the underlying probability distribution for observed target values. Low values for this ratio indicate that the mean predicted value is highly probable, while higher values for this ratio (> 1.0) indicate that the mean predicted value falls within the $< 31.8\%$ probability tails of the observed distribution, which is less desirable. This provides some means to compare the results from the threshold and peak time prediction tasks, and is preferable to the pearson correlation because it allows for direct comparisons for listed MAEs in both problem domains. Therefore, we present the SEP MAE normalized with the standard deviation of observed target values as a metric for comparison between both problem domains. We will refer to this value henceforth as R_{SEP}.

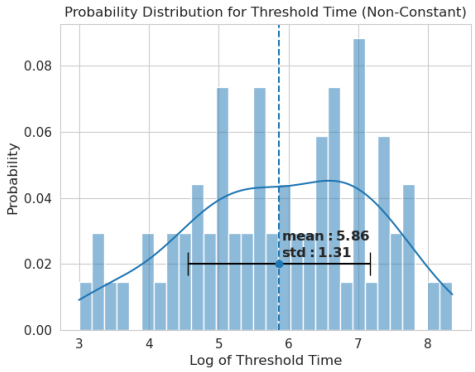
As with random partitioning, we present the distribution of target values in the training set using chronological partitioning for SEPs (top) and all non-constant events (bottom) for threshold (left) and peak time prediction (right). We note that the standard deviation for the distributions of target values for SEPs and for all non-constant events is slightly larger than with the training set in the random partition. Potentially, this could mean that extreme events (i.e., those with low or high values for time) are better represented in the chronological partition though, of course, we still expect the



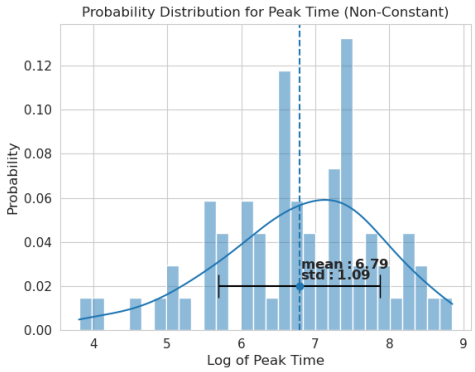
(a) Threshold Time (SEP)



(b) Peak Time (SEP)



(c) Threshold Time (non-Constant)



(d) Peak Time (non-Constant)

Figure 4.3: Probability distribution of target values for threshold time (left) and peak time (right), split between SEP events (top) and all non-constant events (bottom) in the training set using chronological partitioning. The blue point in each plot signifies the mean of the distribution, while the black error bars signify the standard deviation.

distribution of target values between the training and the test set to be fairly dissimilar. Model performance on the chronologically partitioned test should, therefore, still be lower than with random partitioning. Furthermore, the standard deviation for threshold time is 1.44 compared to 1.38 in the random partition, while the standard deviation is 1.25 compared to 1.22 for peak time, so the actual difference in standard deviations between the two training sets is quite small. As with the random partitioned data, we keep track of the ratio of SEP-MAE to the standard deviation of observed target values (R_{SEP}) to determine our models' relative predictive performance on the

threshold and peak time prediction tasks.

4.4 Randomized Partition Results

4.4.1 rRT+AE Results

Table 4.1: Threshold time: summary of regression results for rRT+AE using random partitioning. An oversampling rate of p-q% denotes p% SEP and q% non-SEP Elevated Proton Events

Oversampling Rate	SEP MAE	Combined MAE	SEP PCC	Non-Constant PCC	R_{SEP}
None	1.44	**0.23**	0.55	0.64	1.05
10-0%	0.93	**0.23**	0.62	0.63	0.68
10-10%	0.86	0.30	0.64	0.64	0.63
20-0%	0.80	0.29	0.67	0.65	0.58
20-10%	**0.74**	0.32	0.66	0.65	**0.54**
30-0%	0.95	0.33	0.62	0.60	0.69
30-10%	0.81	0.39	0.66	0.65	0.59
30-20%	0.79	0.45	0.65	0.67	0.58
40-0%	0.79	0.39	0.66	0.64	0.58
40-10%	0.80	0.46	0.62	0.65	0.58
40-20%	0.82	0.50	0.66	0.65	0.60
40-30%	0.76	0.66	**0.68**	**0.68**	0.55

We see that oversampling SEP events has a large effect on the reported SEP-MAE, as can be seen with the much lower SEP-MAE going from no oversampling (1.44) to $10 - 0\%$ (0.93) oversampling. In addition, we have some evidence that increasing the oversampling rate of elevated proton events improves the resulting SEP-MAE score, as the lowest SEP-MAE is seen at a sampling rate of $20 - 10\%$ (0.74). We can also see this effect when fixing SEP oversampling rates, as all rates of SEP oversampling benefit from some degree of oversampling of elevated events.

It should also be noted that the pearson coefficient for SEP and non-constant events for most runs in the table are 0.60, showing that there is a strong positive correlation between predicted and observed threshold times. Hence, instances with lower observed

threshold times tend to produce lower predictions, which is desired behavior.

Table 4.2: Peak time: summary of regression results for rRT+AE using random partitioning. An oversampling rate of p-q% denotes p% SEP and q% non-SEP Elevated Proton Events

Oversampling Rate	SEP MAE	Combined MAE	SEP PCC	Non-Constant PCC	R_{SEP}
None	0.86	**0.23**	0.58	0.45	0.70
10-0%	**0.69**	**0.23**	0.59	0.38	**0.57**
10-10%	0.93	0.30	0.57	0.37	0.77
20-0%	0.72	0.27	0.61	0.38	0.59
20-10%	0.96	0.34	0.58	0.37	0.79
30-0%	0.80	0.34	0.59	0.38	0.66
30-10%	0.89	0.38	0.56	0.39	0.72
30-20%	1.02	0.46	0.57	0.37	0.84
40-0%	0.87	0.34	**0.63**	0.41	0.71
40-10%	0.92	0.44	0.57	0.41	0.75
40-20%	0.96	0.51	0.59	0.40	0.79
40-30%	0.83	0.63	0.62	**0.47**	0.68

As with predicting threshold time, optimal performance for predicting peak time is achieved with some degree of oversampling for SEP events, as the best SEP MAE is achieved at a sampling rate of 10-0% (0.69). However, there is less evidence that sampling of elevated proton events improves model predictive performance, at least when considering SEP-MAE. At all sampling rates for SEP events, introducing oversampling for the elevated proton events increases the reported SEP-MAE (with the exception of sampling at 40-30% vs sampling at 40-0%, where the respective SEP MAEs are 0.83 and 0.87).

The pearson coefficients for SEP events for peak time prediction are not as high as they were with threshold time predictions, as listed in table 4.1. Moreover, the non-constant pearson coefficients are significantly lower than those for SEP events, indicating that the predicted values for elevated proton events are, in general, less correlated with the observed values than those for SEPs are. This might provide some indication for why oversampling of elevated proton events appears to raise the SEP-MAE, as the algorithm appears to struggle at learning effective correlations between

input features for elevated events and the observed target value, at least compared to SEP events. Referring back to the plot of peak time distributions with SEPs and all non-constant events in figure 4.2, we note that, while the mean of observed peak times is not changed too much with the addition of elevated events (6.65), the standard deviation is reduced from 1.22 from 1.05, a larger change than was seen with threshold time, where the standard deviation dropped from 1.38 to 1.24. This indicates that the change in the distribution of target values when adding elevated proton events is more pronounced with peak time than with threshold time. This misalignment in the distribution of peak times for SEPs and elevated proton events might be why the algorithm performs worse with respect to SEP-MAE when elevated events are oversampled. In addition to the previous observation, we note that the lowest ratio between SEP-MAE and the standard deviation for observed SEP values, R_{SEP}, is 0.57. Comparatively, the lowest R_{SEP} ratio we achieved for threshold time prediction was 0.54. The values for R_{SEP} are, in general, higher for peak time prediction across oversampling rates than they were with threshold time, further indicating that our model is slightly less performant at peak time prediction than threshold time prediction.

4.4.2 Adaptive-Calibration Results

The results for using adaptive-calibration at varying oversampling rates for the oversampled distribution are shown in table 4.3. For predicting event threshold time, the best run for adaptive-calibration occurs at an oversampling rate of 30-0%. However, there does not seem to be any advantage conferred using adaptive-calibration over rRT+AE, as the best SEP-MAE for rRT+AE was 0.74 while the lowest SEP-MAE adaptive-calibration is able to achieve is 0.96. An observation we make is that adaptive-calibration is able to keep the combined MAE low while optimizing SEP-MAE. However, it is not as important to optimize non-SEP MAE because these events

Table 4.3: Threshold time: summary of regression results for adaptive-calibration using random-partitioning. An oversampling rate of p-q% denotes p% SEP and q% non-SEP Elevated Proton Events

Oversampling Rate	SEP MAE	Combined MAE	SEP PCC	Non-Constant PCC	R_{SEP}
10-0%	1.07	0.18	0.62	0.62	0.78
10-10%	1.07	0.18	0.61	0.62	0.78
20-0%	1.01	0.18	**0.64**	0.62	0.74
20-10%	1.04	0.19	0.62	0.63	0.76
30-0%	**0.96**	**0.17**	**0.64**	**0.64**	**0.70**
30-10%	1.05	0.18	0.63	0.63	0.76
30-20%	1.03	0.17	0.61	0.61	0.75
40-0%	1.05	0.17	0.61	0.60	0.76
40-10%	1.05	0.21	0.60	0.60	0.76
40-20%	1.10	0.17	0.61	0.60	0.80
40-30%	1.12	0.18	0.60	0.63	0.81

would theoretically be filtered out by the intensity prediction algorithm.

Table 4.4: Peak time: summary of regression results for adaptive-calibration using random partitioning. An oversampling rate of p-q% denotes p% SEP and q% Elevated Proton Events

Oversampling Rate	SEP MAE	Combined MAE	SEP PCC	Non-Constant PCC	R_{SEP}
10-0%	1.04	**0.16**	0.41	0.19	0.85
10-10%	**0.95**	**0.16**	0.39	0.20	**0.78**
20-0%	1.04	0.17	0.45	0.18	0.85
20-10%	1.19	0.17	**0.46**	**0.26**	0.98
30-0%	1.08	0.17	0.45	0.23	0.89
40-0%	1.11	0.17	0.44	0.22	0.91
40-10%	1.09	0.17	**0.46**	0.23	0.89
40-20%	0.98	0.17	0.45	0.23	0.80
40-30%	1.03	0.17	0.43	0.20	0.84

The results of performing peak-time prediction with the adaptive-calibration model at various oversampling rates for the oversampled distribution are presented in table 4.4. As with threshold time prediction, using adaptive calibration seems to show a decrease in performance over rRT+AE. The best SEP-MAE that the adaptive-calibration model was able to achieve was 0.95 at an oversampling rate of 10-10%, while the best SEP-MAE for rRT+AE was 0.69, showing that there is no benefit gained by adding a calibration stage to this task.

4.4.3 Comparison with Single-Stage Neural Network

As with peak intensity prediction, we compare the performance of our techniques with a "standard" neural network using a single stage of training. The architecture and hyperparameters for the standard neural network that is used for comparison are the same as they were in the peak intensity prediction task, described in section 3.5.3. Various oversampling rates were tested, and the results from the single-stage neural network with the lowest SEP-MAE score for threshold and peak time prediction are presented in tables 4.5 and 4.6, respectively.

Table 4.5: Threshold time: summary of regression results for model comparison using random-partitioning. An oversampling rate of p-q denotes p% SEP and q% elevated proton events

Model Architecture	Oversampling Rate	SEP MAE	Combined MAE	SEP PCC	Non-Constant PCC	R_{SEP}
single-stage network	10-10%	0.86	0.29	0.65	0.61	0.62
rRT+AE	20-10%	**0.74**	0.32	**0.66**	**0.65**	**0.54**
adaptive-calibration	50-30%	0.96	**0.17**	0.64	0.64	0.70

Table 4.6: Peak time: summary of regression results for model comparison using random-partitioning. An oversampling rate of p-q denotes p% SEP and q% elevated proton events

Model Architecture	Oversampling Rate	SEP MAE	Combined MAE	SEP PCC	Non-Constant PCC	R_{SEP}
single-stage network	10-0%	0.75	0.23	0.50	0.31	0.61
rRT+AE	10-0%	**0.69**	0.23	**0.59**	**0.38**	**0.57**
adaptive-calibration	50-30%	0.95	**0.16**	0.39	0.20	0.78

Using SEP-MAE as the governing metric, we see that, similar to peak intensity prediction, the predictive performance of our model improves using two-stage rRT+AE over a single-stage approach in both time prediction tasks. Unlike with peak intensity prediction, the performance of the adaptive-calibration model is worse than that of both the two-stage rRT+AE and single-stage model. As was noted in the analysis for peak-intensity prediction, adaptive-calibration will attempt to optimize predictions over both SEP and non-SEP instances in the dataset. In cases where this is important, such as

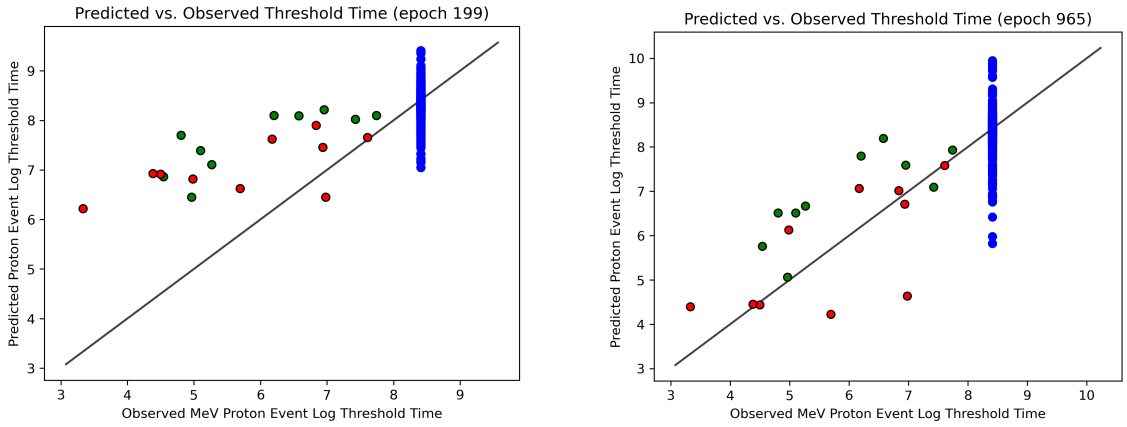
peak-intensity prediction, where we want to limit the number of false-positives while also limiting false-negatives, the adaptive-calibration model offers some advantages over the other models. We can see evidence of that dual-optimization here, as the combined MAE for the adaptive-calibration model for both threshold and peak time prediction are lower than the other two machine learning techniques. In the time prediction tasks, however, we are only seeking to optimize the performance of our algorithm on the SEP instances. The use of adaptive-calibration is therefore limited for these two prediction problems.

4.5 Analysis of Randomized Partition Results

4.5.1 Analysis of rRT+AE Results

As was the case with analysis for peak intensity, the scatter plots provided below show the predicted regression values made by the model for threshold and peak times vs the actual target value. Red markers in the plot are SEP events while green are elevated proton events and blue are constant-target events. The solid diagonal line in each plot is the line where the actual and predicted values are equivalent. The log of both the predicted and observed threshold and peak times are taken before they are plotted to better show the trend between predicted and true values.

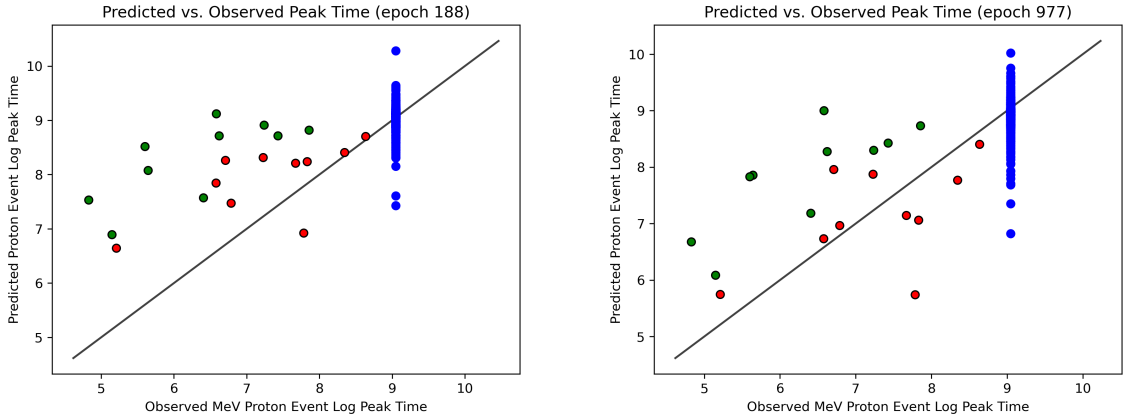
The threshold time prediction scatter plots in figure 4.4 display rRT+AE with no oversampling vs the model with the best sampling rate (20-10% according to SEP-MAE). We see that the effect of oversampling the SEP events (red in the scatter plot) is to drive the predicted times for all events in the test dataset down. This is expected, as SEP events will tend to have a lower threshold time than other events due to the high speed of the CMEs and favorable longitude and latitude needed to create those events



(a) No Oversampling (b) 20-10% Oversampling

Figure 4.4: Threshold time prediction scatter plots for no oversampling (left) and 20-10% oversampling (right) using rRT+AE

in general. We also see that the predicted threshold times for the SEP and elevated proton events are highly correlated with the line of equality, which corraborates the high Pearson correlation between predicted and observed values seen in table 4.1.



(a) No Oversampling (b) 10-0% Oversampling

Figure 4.5: Peak Time Prediction scatter plots for no oversampling (left) and 10-0% oversampling (right) using rRT+AE

The scatter plots for peak time prediction in figure 4.5 show model predictions for rRT+AE with no oversampling and with the best sampling rate of 10-0% according to

SEP-MAE. We observe similar trends with increasing SEP oversampling rates as with the threshold time prediction model. In particular, oversampling of SEP events drives scores for all events in the scatter plot down. In addition, the scores of red, SEP events in the plot are driven down further than other events, and are considerably closer to the line of equality than when no oversampling is used, confirming the results in table 4.2. However, unlike with threshold time prediction, we observe that the predictions for "green", elevated proton events are not as well correlated with the observed values as SEP events are, which corresponds to the lower pearson coefficients listed in table 4.2 for non-constant events versus those for only SEP events. In particular, the peak times for elevated proton events seem to be universally overpredicted. In addition, we note that, though the predictions for peak time are still well correlated with the observed values, the threshold time predictions are more accurate for many of the instances in the test set, as can be seen by the number of SEP instances that are close to line of equality for the threshold time prediction scatter plots vs the peak time prediction scatter plots. This observation is consistent with the higher R_{SEP} scores reported for the peak time prediction model, indicating that our model is slighly less performant at peak time prediction than threshold time prediction.

4.5.2 Analysis of Adaptive-Calibration Results

The threshold time prediction scatter plot in figure 4.6 for the adaptive calibration sampled at a 30-0% rate shows a good correlation between predicted and observed values for threshold time prediction. However, adaptive calibration is not able to accurately predict the threshold times for SEPs as effectively as the rRT+AE with an oversampling rate of 10-0%, as can be seen by the greater absolute distance between SEP events in figure 4.6 and the line of equality. Looking at the sigma bar plot in figure 4.7, we get an indication for why this might be the case.

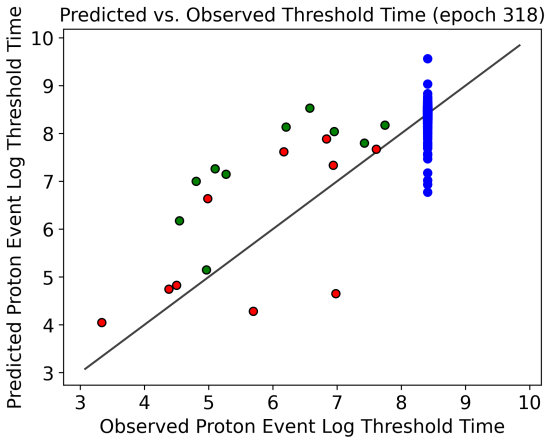


Figure 4.6: Threshold time prediction scatter plots for adaptive-calibration run using 30-0% oversampling for oversampled distribution

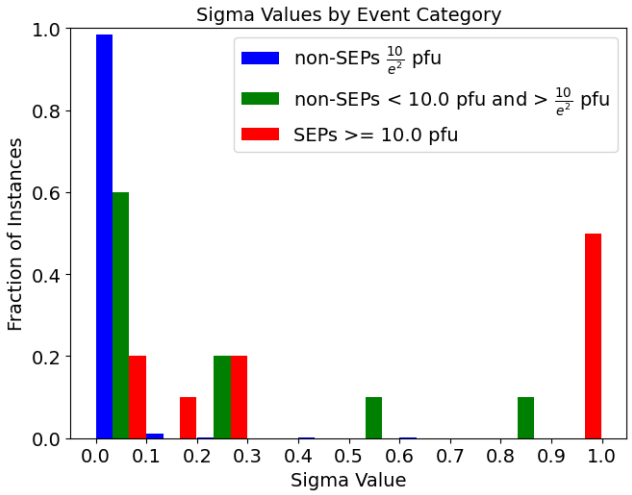


Figure 4.7: Fraction of random partition test-set instances in each event category with with their associated sigma values for threshold time prediction model sampled at 30-0% rate. The x-axis is the sigma-value assigned to the event while the y-axis is the proportion of events from each category with that sigma value. As in the scatter plots, red is for SEP events, green is for non-SEP elevated events, and blue are constant-target events.

While most SEP events favor the oversampled distribution, there are clusters with lower sigma scores, presumably populated by SEP events whose features more closely resemble the non-SEP events that were assigned the constant-target value. As noted before, this should be expected, as the SEP target values for threshold time are not

cleanly delineated from other events like they were with peak intensity prediction, meaning that the distribution of sigma values for SEPs should be more evenly spread. This could, however, be impacting the model's ability to accurately predict times for SEP events. We see some indications that adaptive-calibration algorithm is more effective at assigning non-SEP events low sigma values, so there might be some value using the algorithm to differentiate between majority non-SEP events and minority SEP instances. However, as explained in section 4.4.2, because we could expect most of these non-SEP events to be filtered by the peak intensity prediction algorithm in real-world deployment, the utility of separating the non-SEP and SEP events for threshold time prediction is limited.

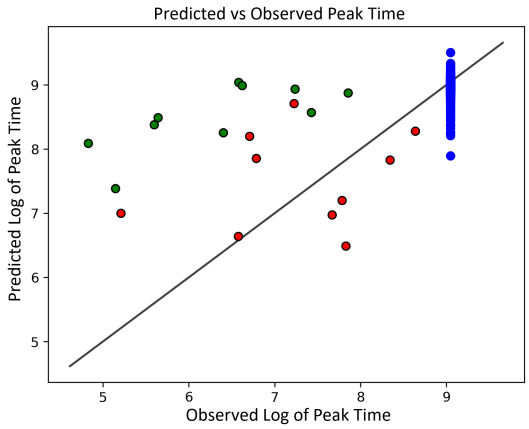


Figure 4.8: Peak time prediction scatter plot for best adaptive-calibration run for random partition

The peak time prediction scatter plot in figure 4.8 shows predictions for the adaptive-calibration model sampled at a 10-10% rate. Again, as with threshold time prediction, adaptive-calibration does not seem to garner any improvement over oversampling with rRT+AE. Moreover, the separation of predicted values for elevated and SEP events seems to be even more pronounced with this model.

Like with threshold time, we plot the distribution of sigma values generated for each event in the test set. We note that the distribution of sigma values is almost a

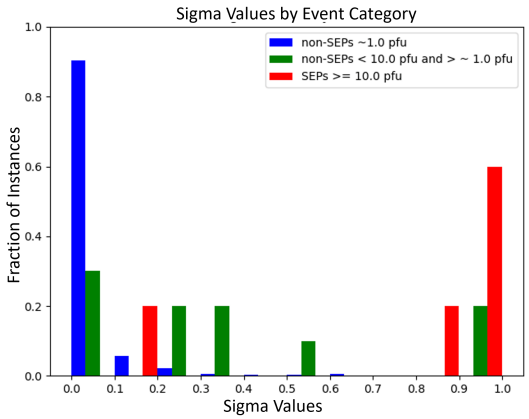


Figure 4.9: Fraction of random partition test-set instances in each event category with with their associated sigma values for peak time prediction model sampled at 10-10% rate. The x-axis is the sigma-value assigned to the event while the y-axis is the proportion of events from each category with that sigma value. As in the scatter plots, red is for SEP events, green is for non-SEP elevated events, and blue are constant-target events.

perfect example of the expected behavior for the sigma calibration layer. Notably, the distribution of sigma values for red SEP events is shifted far to the right, favoring the oversampled score, though there are still some SEP instances that are being reported with low sigma-values, which more correspond to the overpredicted events in the scatter plot. In constrast, the blue constant-target non-SEP events are shifted far to the left, favoring the uniform score. Green non-SEP events seem to be separable from the distribution of events for SEP and constant-target events, though there is a bias to low sigma values, which might be contributing to why elevated proton events are being overpredicted by the model.

4.6 Chronological Partition Results

4.6.1 rRT+AE Results

Table 4.7: Threshold time: summary of regression results for rRT+AE using chronological partitioning. An oversampling rate of p-q% denotes p% SEP and q% Elevated non-SEP Proton Events

Oversampling Rate	SEP MAE	Combined MAE	SEP PCC	Non-Constant PCC	R_{SEP}
None	1.29	0.23	**-0.32**	-0.08	0.90
10-0%	**1.20**	0.21	-0.54	-0.15	**0.83**
10-10%	1.27	0.24	-0.43	**-0.07**	0.88
20-0%	1.30	**0.18**	-0.44	-0.08	0.90
20-10%	1.29	0.29	-0.47	-0.15	0.90
30-0%	1.48	0.30	-0.50	-0.14	1.03
40-0%	1.59	0.29	-0.46	-0.13	1.10
40-10%	1.54	0.30	-0.49	-0.10	1.07
40-20%	1.54	0.40	-0.41	-0.11	1.07
40-30%	1.42	0.44	-0.42	-0.10	0.99
50-0%	1.51	0.37	-0.48	-0.14	1.05
50-10%	1.44	0.41	-0.53	-0.17	1.00
50-20%	1.42	0.46	-0.43	-0.11	0.99
50-30%	1.39	0.55	-0.41	-0.08	0.97

Results for predicting threshold time using rRT+AE using chronological partitioning are presented in table 4.7. As with random partitioning, we perform varying rates of oversampling for SEP events and elevated non-SEP proton events to improve our model's ability to accurately predict times for these events. In general, when fixing SEP oversampling rate, introducing some level of elevated proton event oversampling does reduce the average SEP-MAE, as was noted in the results for random partitioning. However, SEP-MAE is optimized using an oversampling rate of 10-0%, so this trend does not hold over all SEP oversampling rates. Indeed, at higher oversampling rates, we actually see a sharp increase in SEP-MAE. Another aspect to note is that, similar to the peak intensity prediction problem, the SEP and non-constant pearson coefficients for the chronological partition are negative, which indicates that there is a negative

correlation between the predicted values and the observed values for events. Indeed, we that there are several events in the test set whose predictions diverge sharply from the observed values, which will be seen in the analysis presented in section 4.7.

Table 4.8: Peak time: summary of regression results for rRT+AE using chronological partitioning. An oversampling rate of p-q% denotes p% SEP and q% Elevated non-SEP Proton Events

Oversampling Rate	SEP MAE	Combined MAE	SEP PCC	Non-Constant PCC	R_{SEP}
None	1.07	0.22	**0.08**	0.13	0.86
10-0%	1.02	**0.15**	-0.09	0.12	0.82
10-10%	0.97	0.19	-0.17	0.04	0.78
20-0%	1.04	0.17	-0.15	0.11	0.83
20-10%	1.02	0.23	-0.16	0.10	0.82
30-0%	0.98	0.21	-0.09	0.11	0.78
40-0%	1.00	0.21	-0.14	0.09	0.80
40-10%	1.00	0.25	-0.15	0.12	0.80
40-20%	0.94	0.34	-0.09	0.16	0.75
40-30%	0.99	0.40	-0.16	0.10	0.80
50-0%	0.97	0.26	-0.12	0.14	0.78
50-10%	1.02	0.29	-0.08	0.13	0.82
50-20%	0.91	0.37	-0.09	0.16	0.73
50-30%	**0.89**	0.48	-0.02	**0.18**	**0.72**

We present the results of training with rRT+AE for peak time prediction using chronological partitioning in table 4.8. In general, oversampling of SEP events predictably lowers the SEP-MAE, as can be seen when going from 20-0% (1.04) to 30-0% (0.98), and from 40-0% (1.00) to 50-0%, though this trend does not hold across all SEP oversampling threshold (for example, the SEP-MAE increases from 0.98 to 1.00 going from 30-0% to 40-0% oversampling). We observe the best average SEP-MAE is reported with an oversampling rate of 50-30%. As with threshold time and peak intensity prediction for the chronological partition, the observed pearson coefficient is negative. However, these values are not as strongly negative as with threshold time, indicating that the predicted and observed values for peak time prediction are better correlated. In addition, the overall non-constant pearson-coefficient is positive at all oversampling rates, which might be why the optimal SEP-MAE is achieved with some

degree of elevated proton event oversampling. We also find, in contrast to the random partition, the lowest SEP-MAE normalized over the standard deviation of observed values in the training distribution is lower for peak time prediction (0.72) than threshold time prediction (0.83), meaning that the predictive performance for our model on the peak time prediction task is higher than for threshold time prediction using the chronological partition of data.

4.6.2 Adaptive-Calibration Results

Table 4.9: Threshold time: summary of regression results for adaptive-calibration using chronological partitioning. An oversampling rate of p-q% denotes p% SEP and q% Elevated non-SEP Proton Events

Oversampling Rate	SEP MAE	Combined MAE	SEP PCC	Non-Constant PCC	R_{SEP}
10-0%	**1.20**	**0.14**	-0.48	-0.27	**0.83**
10-10%	1.21	0.15	-0.47	-0.27	0.84
20-0%	1.26	0.15	-0.45	-0.27	0.88
20-10%	1.31	0.15	**-0.41**	-0.26	0.91
30-0%	1.38	0.15	-0.44	-0.28	0.96
40-0%	1.28	0.16	-0.43	-0.25	0.89
40-10%	1.35	0.15	-0.42	-0.24	0.94
40-20%	1.27	0.16	**-0.41**	**-0.23**	0.88
40-30%	1.27	0.15	-0.46	-0.25	0.88
50-0%	1.30	0.15	-0.45	-0.27	0.90
50-10%	1.23	0.15	-0.45	-0.27	0.85
50-20%	1.27	0.15	-0.44	-0.25	0.88
50-30%	1.22	0.16	-0.48	-0.28	0.85

The regression metrics for predicting threshold time with the adaptive-calibration model are presented in table 4.9. Interestingly, adaptive-calibration has the same SEP-MAE as the best performing rRT+AE model (1.20). Additionally, this SEP-MAE was achieved at the same oversampling rate (10-0%). This might indicate that this oversampling rate provides the best training distribution to optimize model performance on SEP threshold time prediction irrespective of the number of stages of training used. This oversampling rate may be useful to keep in mind when training a model for real-

world deployment. However, even though the adaptive-calibration model's performance on this metric was just as good as the best rRT+AE model, rRT+AE is the preferred architecture for this problem, as it requires fewer stages and less hyperparameter tuning than the adaptive-calibration model.

Table 4.10: Peak time: summary of regression results for adaptive-calibration using chronological partitioning. An oversampling rate of p-q means p% SEP and q% Elevated non-SEP Proton Events

Oversampling Rate	SEP MAE	Combined MAE	SEP PCC	Non-Constant PCC	R_{SEP}
10-0%	1.09	0.14	-0.33	-0.23	0.87
10-10%	1.08	**0.13**	-0.32	-0.23	0.86
20-0%	1.16	0.14	-0.32	-0.22	0.93
20-10%	**1.07**	0.14	-0.32	-0.22	**0.86**
30-0%	1.13	**0.13**	-0.38	-0.24	0.90
40-0%	1.16	0.15	**-0.31**	-0.20	0.93
40-10%	1.12	**0.13**	-0.34	-0.22	0.90
40-20%	1.13	**0.13**	-0.35	-0.25	0.90
40-30%	1.17	**0.13**	**-0.31**	**-0.19**	0.94
50-0%	1.17	**0.13**	**-0.31**	-0.25	0.94
50-10%	1.17	**0.13**	**-0.31**	**-0.19**	0.94
50-20%	1.11	0.14	-0.34	-0.22	0.89
50-30%	1.13	**0.13**	-0.32	-0.21	0.90

For peak time prediction using the chronological partition, again, adaptive calibration does not seem to confer any advantage over using rRT+AE. Indeed, we see an increase in SEP-MAE using adaptive-calibration (1.07) vs rRT+AE (0.97).

4.6.3 Comparison with Single-Stage Neural Network

Similar to the random partition, we trained a single-stage neural network to compare with the multi-stage rRT+AE and adaptive-calibration techniques. We present the best single-stage neural network as determined by SEP-MAE score in tables 4.11 and 4.12.

For threshold time prediction, using SEP-MAE as the governing metric, we note that rRT+AE and adaptive-calibration outperforms single-stage training, with the

Table 4.11: Threshold time: summary of regression results for model comparison using chronological partitioning. An oversampling rate of p-q denotes p% SEP and q% elevated proton events

Model Architecture	Oversampling Rate	SEP MAE	Combined MAE	SEP PCC	Non-Constant PCC	R_{SEP}
single-stage network	10-0%	1.26	0.20	**-0.41**	**-0.09**	0.88
rRT+AE	10-0%	**1.20**	0.21	-0.54	-0.15	**0.83**
adaptive-calibration	10-0%	**1.20**	**0.14**	-0.48	-0.27	**0.83**

Table 4.12: Peak time: summary of regression results for model comparison using chronological partitioning. An oversampling rate of p-q denotes p% SEP and q% elevated proton events

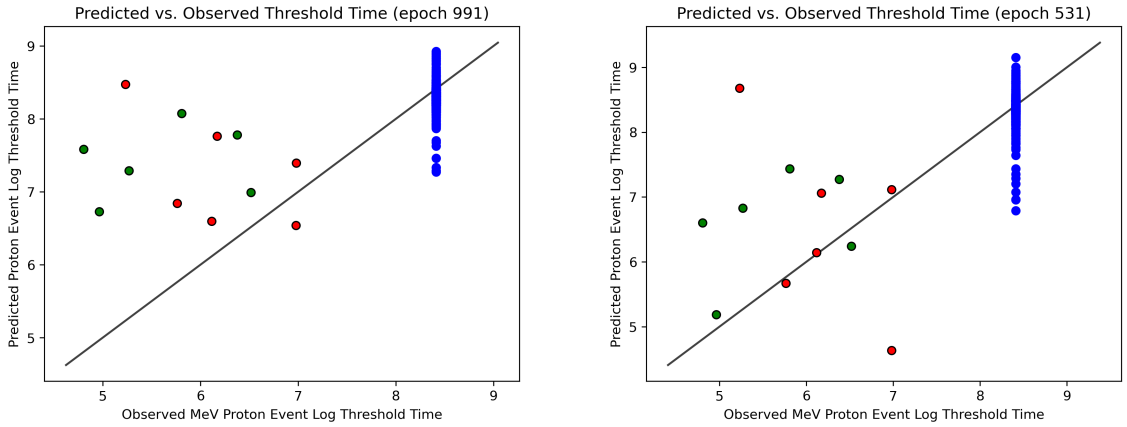
Model Architecture	Oversampling Rate	SEP MAE	Combined MAE	SEP PCC	Non-Constant PCC	R_{SEP}
single-stage network	10-10%	1.02	0.18	-0.06	**0.12**	0.82
rRT+AE	50-30%	**0.89**	0.48	**-0.02**	0.18	**0.72**
adaptive-calibration	20-10%	1.07	**0.14**	-0.32	-0.22	0.86

multi-stage techniques achieving an SEP-MAE of 1.20 while the single-stage model only achieving 1.26. For peak time prediction, however, while the two-stage rRT+AE model outperforms the single-stage model (netting an SEP-MAE of 0.89 vs 1.02), the adaptive-calibration model is outperformed by the single-stage model, as was the case in the random-partition. We can therefore conclude that two-stage rRT+AE is the most performant model for SEP threshold and peak time prediction. Another interesting point is that the oversampling rate for all the best performing models for threshold time prediction is 10-0%, as was noted in 4.6.2, providing further evidence that this might be a universally optimal oversampling rate to use in real-world deployment.

4.7 Analysis of Chronological Partition Results

4.7.1 Analysis of rRT+AE Results

The scatter plots presented here are for the median rRT+AE model for experiments conducted with no oversampling and those conducted at the best oversampling rate for the second-stage, as determined by SEP-MAE.



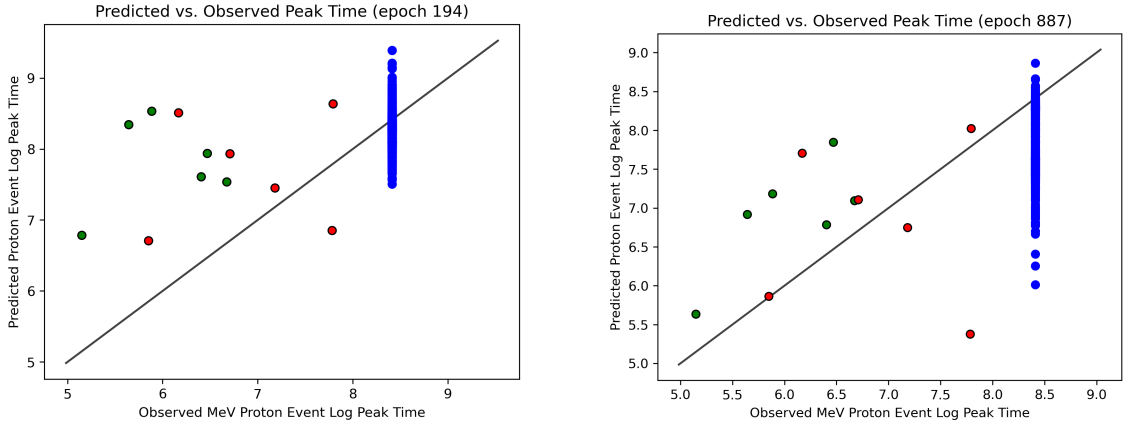
(a) No Oversampling

(b) 10-0% Oversampling

Figure 4.10: Threshold time prediction scatter plot for rRT+AE with no oversampling and best oversampling rate for chronological partition

Threshold time predictions are presented in the scatter plot in figure 4.10. The scatter plots show the predictions with no oversampling and with a 10-0% sampling rate, which was the best sampling rate according to SEP-MAE metric. The effect of oversampling is to drive scores for all events down, closer to the lower observed threshold times for the SEP events. In addition, when using oversampling, the SEP events have predicted scores closer to the line of equality than without. There are two main exceptions to this rule. One event in the test set was predicted with a threshold time far over the observed time, and one was severely underpredicted. We will examine these events in closer detail in 4.7.3.

For peak time prediction using rRT+AE with no oversampling and at the best oversampling rate of 50-30% (according to SEP-MAE) in figure 4.11, we see similar trends as were seen with threshold time prediction. Oversampling has the dual effect of making the predictions for SEP events (and elevated proton events, in this case) more accurate, as well as driving the predicted scores for all events lower. In general, the correlation between the predicted and observed values for SEP events is strong, which is why the pearson coefficient for SEP events is higher for peak time than with threshold



(a) No Oversampling (b) 50-30% Oversampling

Figure 4.11: Peak time prediction scatter plot for rRT+AE with no oversampling and best oversampling rate for chronological partition

time prediction. However, there is still one outlier event that is heavily underpredicted by the model, which will be discussed in section 4.7.3.

4.7.2 Analysis of Adaptive-Calibration Results

As with the rRT+AE model, the scatter plots presented are for the median model at the best oversampling rate according to SEP-MAE.

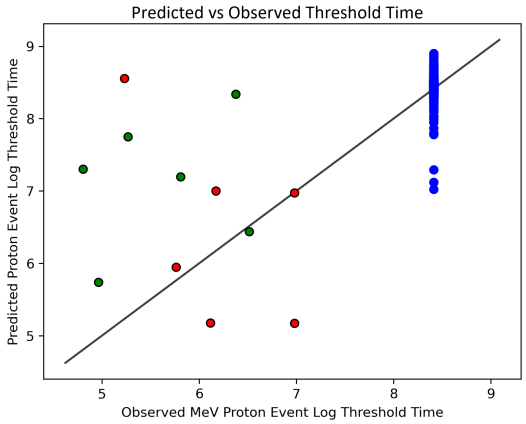


Figure 4.12: Threshold time prediction scatter plot for 10-0% oversampled adaptive-calibration run for chronological partition

The threshold time prediction plot for the best run with an oversampling rate of 10-0% is shown in figure 4.12, and shows some promising trends. The model is able to predict the threshold times for some proton events with reasonable accuracy. However, the trend is less strong than with the oversampled rRT+AE model. In particular, there is a large group of proton events that the model is over-predicting threshold times for. Seeing as, in a deployed setting, we would want advanced warning of SEP events, we want to minimize overpredictions made by our models. Therefore, this behavior from the adaptive-calibration model is undesirable.

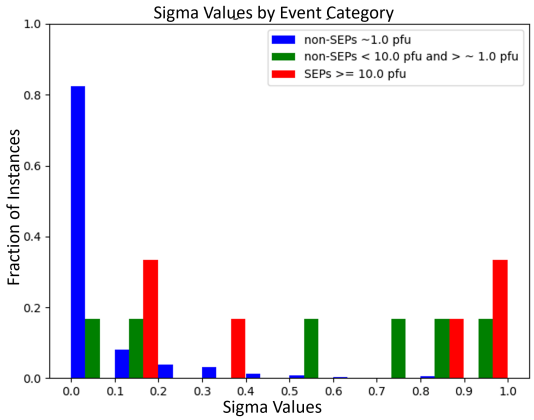


Figure 4.13: Fraction of chronological partition test-set instances in each event category with with their associated sigma values for threshold time prediction model sampled at 10-0% rate. The x-axis is the sigma-value assigned to the event while the y-axis is the proportion of events from each category with that sigma value. As in the scatter plots, red is for SEP events, green is for non-SEP elevated events, and blue are constant-target events.

For the distribution of sigma values in the test set shown in figure 4.13, we see that the SEP events are approximately evenly split between high values and low values for sigma. The extreme events with sigma values near 1.0 correspond to most under-predicted events for the model, while the events closest to 0.0 most likely represent the overpresented events for the model. In addition, we notice that the model favors a fairly uniform distribution of sigma values for elevated events, which explains why

they have predicted threshold times in between the constant-target non-SEPs and the SEP events themselves. Notably, while there is still a shift in the distribution for each category of event, the shift is less evident than that seen for the random-partition in figure 4.7, which may provide some indication for why the model performs poorly on the chronological-partition relative to the the random-partition. Analysis of feature values that generate these different sigma distributions may help identify ways to better delineate SEP, elevated non-SEP events, and unassociated non-SEP events in the future.

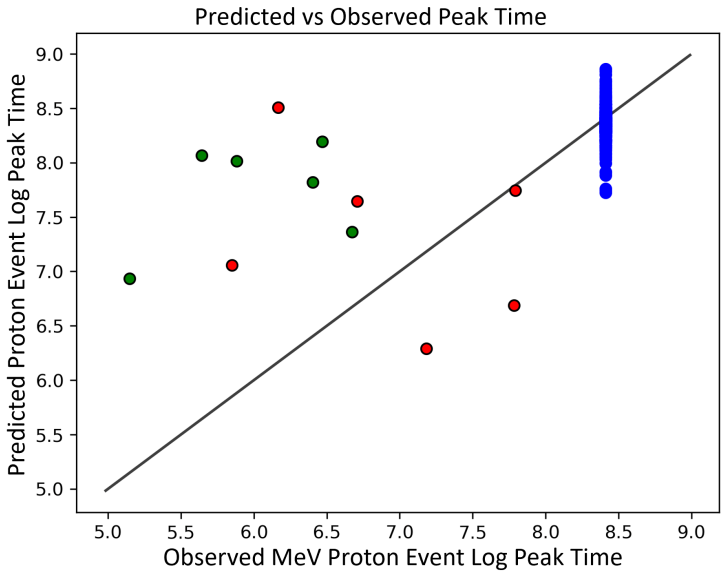


Figure 4.14: Peak time prediction scatter plots for 20-10% oversampled adaptive-calibration run for chronological partition

The peak time prediction scatter plot for the adaptive-calibration model trained using an oversampled distribution of 20-10% is shown in figure 4.14. As with threshold time prediction, we note that the correlation between predictions made by the model and observed values is weaker than what was displayed with the best rRT+AE model.

Looking at the distribution of sigma values for this model, provided in figure 4.15, we see that the distribution of SEP and elevated proton events is clustered around lower

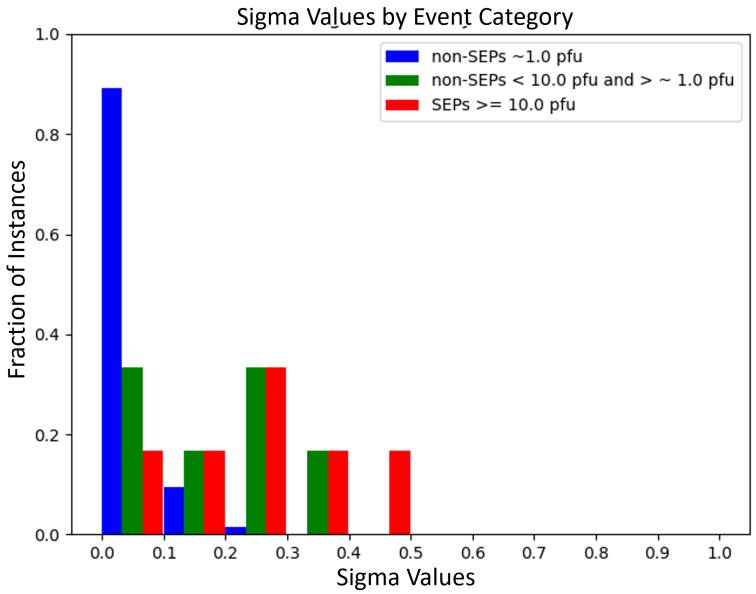


Figure 4.15: Fraction of chronological partition test-set instances in each event category with with their associated sigma values for model sampled at 20-10% rate. The x-axis is the sigma-value assigned to the event while the y-axis is the proportion of events from each category with that sigma value. As in the scatter plots, red is for SEP events, green is for non-SEP elevated events, and blue are constant-target events

values of sigma, favoring the uniform distribution branch of the model. This contrasts with the sigma value distribution for threshold-time prediction on the chronological partition, which shows a slight shift towards higher values with SEP events. Though the events still seem to be separable from constant-target non-SEPs, the clustering around low values of sigma could explain why the model has a greater tendency to overpredict values for peak times than for threshold times.

4.7.3 Analysis of Poorly-Predicted SEP Events

Table 4.13: Feature values of SEP events with poorly predicted threshold times

CME Start Time	Latitude (°)	Longitude (○)	Half-Angle (°)	Speed ($\frac{km}{s}$)	Connection Angle (°)
2015-10-29 02:48:00	-24	95	31	390	37.54
2015-06-25 08:36:00	23	46	41	1450	43.95

Table 4.14: Predicted and observed threshold times for poorly predicted SEP events

CME Start Time	Predicted Threshold Time (min)	Observed Threshold Time (min)
2015-10-29 02:48:00	5,878	187
2015-06-25 08:36:00	102	1074

We limit analysis of the poorly-predicted events to SEP events, as these events are most important to predict correctly. The two poorly-predicted SEP events we will perform analysis for are identified in figure 4.10. The times and features of these two events are provided in table 4.13, while the predicted times given by the median rRT+AE model at an oversampling rate of 10-0% and observed threshold times for the events are contained in table 4.14.

For the event with corresponding CME on 2015-10-29 02:48:00, we note that the speed as listed by the DONKI catalog is only 390 $\frac{km}{s}$. The extremely low value for this feature is probably the predominant reason the model is heavily overpredicting the target value for this event, as threshold time is heavily correlated with event speed. Indeed, this event was also listed as a false-negative in 3.8.3, and the reasons why the model was underpredicting the peak intensity of this event are likely the same (notably, that the SEP event associated with this CME is most likely the product of multiple CMEs, not just the one listed). Conversely, for the event with corresponding CME at 2015-06-25 that is being heavily underpredicted by the model, we note that the speed, 1450 $\frac{km}{s}$, is fairly high. In addition, the direction of this event (46° longitude and 23° latitude) is favorable, and the connection angle, while not 0°, is still well correlated with low observed threshold times. This combination of feature values is the most likely explanation for why the model is heavily underpredicting the threshold time for this event. While it is still important to understand why the model is underpredicting values for this event, we note that the error made by the model in this case is less critical than the heavy overprediction made for the 2015-10-29 02:48:00 event. In general,

we would prefer that our model underpredict than overpredict threshold times, as underpredictions will cause a warning to be sent prematurely, while an overprediction could cause the warning to be issued after the event has actually occurred and damage has already been caused.

Table 4.15: Feature values of SEP events with poorly predicted peak times

CME Start Time	Predicted Peak Time (min)	Observed Peak Time (min)
2015-06-25 08:36:00	217	2399

We also analyzed an event that was severely underpredicted for the peak time prediction problem. Similarly,the predicted time as produced by the median rRT+AE model at an oversampling rate of 50-20% and observed times for the event are contained in table 4.15. Notably, this event caused the severe underprediction for threshold time prediction as well. The reasons for the this low prediction are, therefore, likely to be the same.

Chapter 5

Conclusion

5.1 Summary of Algorithms Used

This thesis presents two new machine learning algorithms for imbalanced regression problems. The first algorithm, rRT+AE, modifies the two-stage cRT model that was described in Kang et. al [9]. Like in Kang et. al, our model separates the representation learning for the training distribution and learning for the actual prediction task into two distinct training phases. Similarly, the representation learning for our model is performed on the uniform distribution of data, and the representations learned by training on this distribution are frozen in the feature extractor. The weights in the prediction head are then reinitialized for the second stage of training, which uses an oversampled distribution of data. Unlike in Kang et. al, we perform regression by mapping the features of each instance to a real number instead of predicting a probability vector for classification. The function that our algorithm learns is therefore fundamentally different. Whereas classifier learning attempts to learn boundaries seperating instances from different classes in feature space, regression learning attempts to fit each of the instances in the training distribution using their features as the independent variables

to the function. To implement this regression function, we replaced the softmax activation function on the prediction head of the cRT model with a linear activation function. In addition, our algorithm differs from Kang et. al's implementation by utilizing an autoencoder branch in addition to the regression branch during the first-stage of training. The autoencoder reconstructs the input features of an instance based on the representations generated by the feature extractor. This forces the feature extractor to learn more effective representations of the input instances so that their features can be reconstructed by the decoder branch of the network. These modifications allow our rRT+AE model to be used effectively for our imbalanced regression task, showing marked improvement over single-stage training with the exception of peak intensity prediction with chronological partitioning.

The second algorithm that was described in this work, adaptive-calibration, was a modification of the adaptive-calibration for classification technique described in Zhang et. al. To adapt the technique for regression tasks, a larger degree of modification was required than with cRT. To start with, the classwise calibration parameters that are trained in the second stage of classification adaptive-calibration have no relevance for our regression task, so we needed to develop another method for generating the calibrated score for the oversampled distribution. Our solution was to conduct regression training on the uniform and oversampled distribution in the same stage of training. The weights for the two regression heads were then frozen. We then introduced a third, "calibration" stage of training where we learned a function $\sigma(z)$ to linearly combine the uniform score $s_{uniform}(z)$ and the oversampled score $s_{oversampled}(z)$ by training on the uniform distribution of data, using the representations z learned by the feature extractor during the first-stage of training. The goal of this stage of training was to learn a function $\sigma(z)$ that favored $s_{uniform}(z)$ for the majority of non-SEP instances and $s_{oversampled}(z)$ for instances that were over-represented in the oversampled distribution,

namely the SEP instances. This contrasts from the classification adaptive-calibration network, which uses just two-stages of training, and learns the class calibration parameters composing the calibrated score in-tandem with the weighting function $\sigma(z)$. A current limitation of this algorithm is that the output score of the model in the calibration stage cannot be higher than $s_{oversampled}(z)$ or lower than $s_{uniform}(z)$. Figure 5.1 gives the distribution of SEP, elevated proton events, and constant-target events whose observed peak intensities are below $s_{uniform}(z)$, above $s_{oversampled}(z)$, and in range of the two scores. From the figure, we see that we cannot assume that the observed target for all SEP instances will be lower than the predicted score for $s_{oversampled}(z)$, nor that the observed target for non-SEP instances will be higher than the prediction score from $s_{uniform}(z)$. This is consistent with observations from our results, where we saw that the number of false-negatives generated by adaptive-calibration is higher than rRT+AE models with similar F1-scores, implying that the values being predicted by the adaptive-calibration model for SEP instances are not as high as with oversampled rRT+AE. This shortcoming will need to be addressed in future work with this algorithm.

We note from the results that the adaptive-calibration model seems to performs well for imbalanced tasks where the regression predictions for the minority instances need to be noticeably delineated from the predictions of the majority instances. This is the case with the peak intensity prediction task, as it is critically important that SEP instances are predicted with intensities above threshold, while non-SEP instances are predicted with intensities below threshold. We note that the adaptive-calibration model is able to produce some modest improvement in F1-scores for intensity prediction, implying that this model is able to delineate between SEP and non-SEP instances fairly effectively. However, as noted previously, adaptive-calibration generates more false-negatives than oversampled rRT+AE at a similar F1-score and, due to the additional training time and

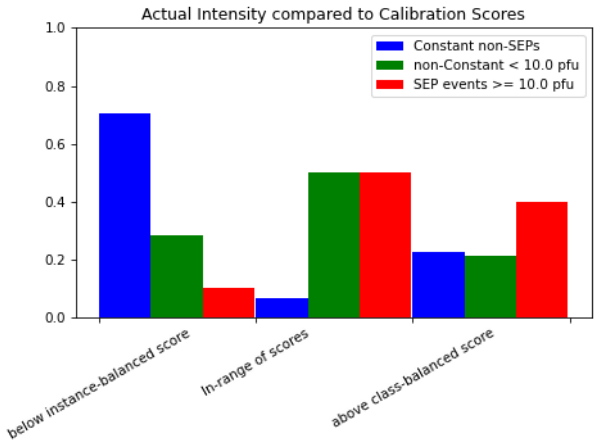


Figure 5.1: Fraction of instances per event category whose actual intensities are below the "instance-balanced", or uniform, regression score (left bin), are above the "class-balanced", or oversampled, regression score (right bin), and whose actual intensities are in-between the predicted scores made by the uniform and oversampled regression heads (center bin).

hyperparameter fine-tuning that is required with the adaptive-calibration architecture, rRT+AE may still be the architecture of choice for this problem.

Generalizing to other imbalanced regression problems, adaptive-calibration may serve as a useful heuristic to determine if the features of the minority instances can be distinguished from those of majority instances for regression tasks. This can be seen in the distribution of sigma-scores learned by the model for instances in each event category. If the model is capable of learning representations that allow it to distinguish between instances in different event categories for the regression task, it should be capable of learning a distribution of sigma values that favors the oversampled branch for minority instances and the uniform branch for majority instances, as was demonstrated for SEP peak intensity and time prediction. This could help with feature selection as well as hyperparameter tuning of models used in other imbalanced regression tasks.

5.2 Summary of Findings

In this work, we addressed the problem of predicting intensity and times for SEP events using CME measurements and features derived from other sources such as Type II radio wave bursts. We extend the CME feature database by correlating SEP events to provide target values for peak intensity as well as threshold and peak times for SEP events. Using this dataset with the regression labels, we developed models to predict regression targets for peak intensity and event times. In particular, we introduced the rRT+AE technique to perform predictions for SEP event peak intensity and times. We then augmented this technique with adaptive calibration. The results on whether or not rRT+AE or adaptive-calibration proved superior were mixed.

For SEP peak intensity prediction with random partitioning, we achieve a best SEP-MAE of 1.09 and TSS of 0.92 using an oversampling rate of 40-0% using rRT+AE. Conversely, the best F1-score of 0.76 was achieved using adaptive-calibration using oversampling rates of 40-30% and 50-0%, resulting in 3.4 and 3.2 false-negatives, respectively. However, for our use case, we wish to maximize the number of true detections as possible. We could therefore make the argument that the rRT+AE model with an oversampling rate of 20-20% was our best performing model. Though its F1-score (0.74) is not quite as high as the best performing adaptive-calibration models, we are able to attain this relatively high F1-score while reducing the number of false-negatives to 2.6. This tradeoff will need to factor into decisions on which model architecture to deploy for intensity prediction in the real-world.

For chronological partitioning, we achieved a best SEP-MAE of 1.60 using rRT+AE at an oversampling rate of 40-30%, while the highest TSS of 0.76 and lowest number of false-negatives of 1.4 was achieved using rRT+AE at an oversampling rate of 40-0%. However, the highest F1-score achieved on this partition was 0.65, and was achieved

using adaptive-calibration with 2.8 false-negatives.

For time prediction, rRT+AE consistently produced the best results, as determined using SEP-MAE. The best result with random-partitioning for threshold time prediction was an SEP-MAE of 0.74 using rRT+AE at an oversampling rate of 20-10%, while the best result on the random partition for peak time prediction was an SEP-MAE of 0.69 using rRT+AE at an oversampling rate of 10-0%. For chronological partitioning, the best score for the threshold time prediction problem was an SEP-MAE of 1.20, which was achieved using rRT+AE with 10-0% oversampling. For peak time prediction, the best result was an SEP-MAE of 0.89 using rRT+AE at an oversampling rate of 50-30%.

5.3 Limitations and Possible Improvements

This work is currently limited to 10 MeV SEP events. We could expand the peak intensity prediction task to higher energy proton events, such as 50 Mev and 100 MeV. While the problem of predicting intensity and times for proton events for other energies is related, the data imbalance problem for those energy regimes are even more extreme than with 10 MeV SEP events, posing a problem for future work. Additional techniques in combination with those discussed in this thesis may need to be explored to improve performance on these even more imbalanced domains.

Another limitation of our current work is the inability of our adaptive-calibration model to generate improved scores on instances with observed targets above the over-sampled score and those with observed targets below the uniform score. A series of techniques are being investigated to address this fundamental limitation of the the adaptive-calibration model, including scaling the scores for both branches of the model using factors obtained from the training data, as well as score offsets that are learned

from feature representations during model training. Some combination of these techniques may help improve our model's ability to lower false-negatives while keeping the number of subsequent false-alarms to a minimum.

In addition, the model proposed in this work would not be able to operate under real-time conditions with current data constraints due to delays in publishing measurements to the CDAW CME catalog. One possibility to enable real-time deployment of this model would be to generate features of CME events using computer vision techniques on raw coronagraph frames. This model could predict useful features for the peak intensity as well as threshold and peak time prediction models, such as the height-time and width of CME plumes in coronagraph frames. As the measurement of CME attributes takes place over time and a sequence of frames, there is a temporal component to this prediction task that would need to be investigated as well.

Finally, we could perform further research on events that are difficult for our model to predict accurately. In particular, out model struggles to accurately predict intensity and times in scenarios where double or multiple CMEs are correlated to the event. Usually, the speed and width of any single CME in these double or multi-CME scenarios are lower than when only one CME causes the SEP event. As a result, the model has difficulty predicting elevated intensities when only one of these CMEs is used as input. Our feature set already includes features that attempt to provide contextual information such as the number of CMEs that have occurred over varying periods of time before the input event, but these features are not sufficient to fully eliminate the false-negatives caused by double-CMEs. One approach that could be explored is incorporating the features of all events over these time-periods as an input sequence to a recurrent neural-network. This would allow the model to learn relations not only from the number of events that are contained within a certain period of time, but how the features of each of these events contribute to the resulting SEP as well.

Bibliography

[1] Phtx file for 2015-06-21 sep event.

[2] Phtx file for 2015-10-29 sep event.

[3] S. Aminalragia-Giamini, S. Raptis, A. Anastasiadis, A. Tsigkanos, I. Sandberg, A. Papaioannou, C. Papadimitriou, P. Jiggens, A. Aran, and I. A. Daglis. Solar Energetic Particle Event Occurrence Prediction using Solar Flare Soft X-Ray Measurements and Machine Learning. *Space Weather*, 11, 2021.

[4] S. Boubrahimi, B. Aydin, P. Martens, and R. Angryk. On the Prediction of > 100 MeV Solar Energetic Particle Events Using GOES Satellite Data. In *IEEE International Conference on Big Data*, pages 2533–2542, 2017.

[5] P. Brea. Using Machine Learning Techniques to Forecast Solar Energetic Particles. In *SOARS*, 2019.

[6] A. 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. volume 296, 2021.

[7] F. Inceoglu, J. Jeppesen, P. Kongstad, N. J. Hernández, R. H. Jacobsen, and C. Karoff. Using Machine Learning Methods to Forecast if Solar Flares will be Associated with CMEs and SEPs. *The Astrophysical Journal*, 861, 2018.

[8] S. W. Kahler and A. G. Ling. Forecasting Solar Energetic Particle (SEP) events with Flare X-ray peak ratios. *Journal of Space Weather and Space Climate*, 8, 2018.

[9] B. Kang, S. Xie, M. Rohrbach, Z. Yan, Z. Gordo, J. Feng, and Y. Kalantidis. Decoupling Representation and Classifier for Long-Tailed Recognition. In *2019 Conference on Learning Representations*, New Orleans, LA, USA, 2019 [Online].

[10] S. Kasapis, L. Zhao, Y. Chen, X. Wang, M. Bobra, and T. Gombosi. Interpretable Machine Learning to Forecast SEP Events for Solar Cycle 23. *Space Weather*, 20, 2022.

[11] E. Lavasa, G. Giannopoulos, A. Papaioannou, A. Anastasiadis, I.A. Daglis, A. Aran, D. Pacheco, and B. Sanahuja. Assessing the Predictability of Solar Energetic Particles with the use of Machine Learning Techniques. *Solar Physics*, 296, 2021.

[12] N. Moniz, R. Ribeiro, V. Cerqueira, and N. Chawla. SMOTEBoost for Regression: Improving the Prediction of Extreme Values. In *IEEE International Conference on Data Science and Advanced Analytics*, pages 150–159, 2018.

[13] M. Núñez. Predicting Solar Energetic Proton Events (E $>$ 10 MeV). *Space Weather*, 9, 2011.

[14] R. Ribeiro and N. Moniz. Imbalanced Regression and Extreme Value Prediction. In *European Conference on Machine Learning and Principles and Practice of Knowledge Discovery in Databases.*, 2020.

[15] I. G. Richardson. 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.

[16] I. G. Richardson, T. T von, H. V. Cane, E. R. Christian, C. M. S. Cohen, A. W. Labrador, R. A. Leske, R. A. Mewaldt, M. E. Wiedenbeck, and E. C. Stone. $>$ 25 MeV Proton Events Observed by the High Energy Telescopes on the STEREO A and B Spacecraft and/or at Earth during the First Seven Years of the STEREO Mission. *Solar Physics*, 298:3059–3107.

[17] M. Stumpo, S. Benella, M. Laurenza, T. Alberti, G. Consolini, and M. Marcucci. Open Issues in Statistical Forecasting of Solar Proton Events: A Machine Learning Perspective. *Space Weather*, 19, 2021.

[18] P. Tarsoly. Forecasting SEP Events based on Merged CME Catalogs using Machine Learning. Master's thesis, Florida Institute of Technology, Melbourne, FL, 2021.

[19] J. Torres. A Machine Learning Approach to Forecasting SEP Events with Solar Activities. Master's thesis, Florida Institute of Technology, Melbourne, FL, 2020.

[20] Y. Yang, K. Zha, Y. Chen, H. Wang, and D. Karabi. Delving into Deep Imbalanced Regression. In *Computer Vision and Pattern Recognition.*, 2021.

[21] S. Zhang, Z. Li, S. Yan, X. He, and J. Sun. Distribution Alignment: A Unified Framework for Long-Tail Visual Recognition. In *Computer Vision and Pattern Recognition*, Nashville, RN, USA, 2021 [Online].

[22] Z. Zhong, J. Cui, S. Liu, and J. Jia. Improving Calibration for Long-Tailed Recognition. In *Computer Vision and Pattern Recognition*, 2021.

[23] B. Zhou, Q. Cui, X. Wei, and Z. Chen. BBN: Bilateral-Branch Network with Cumulative Learning for Long-Tailed Visual Recognition. In *Computer Vision and Pattern Recognition*, 2020 [Online].

Appendix

Proof: HSS Reduces to F1-Score when TN is much greater than TP, FP, and FN

Let TP be true positives, FP be false positives, TN be true negatives, and FN be false negatives. The definitions for HSS and F1-Score are:

$$F1 = \frac{TP}{TP + \frac{1}{2}(FP + FN)} \tag{1}$$

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

Factoring two across all terms in the numerator for HSS gives the following.

$$HSS = \frac{(2 \cdot TP \cdot TN - 2 \cdot TP \cdot FN)}{((TP + FP)(FP + TN) + (TP + FN)(FN + TN))} \tag{3}$$

We then multiply the terms in the denominator to get

$$HSS = \frac{(2 \cdot TP \cdot TN - 2 \cdot TP \cdot FN)}{(TP \cdot FP + TP \cdot TN + FP \cdot FP + FP \cdot TN) + (TP \cdot FN + TP \cdot TN + FN \cdot FN + FN \cdot TN)} \tag{4}$$

Assume that $TN \gg TP$, FP, and FP. Let α be a factor, and let β be TP, FP, or FN. When $TN \gg \beta$, then $\alpha \cdot (TN \pm \beta) \approx \alpha \cdot TN$, as TN dominates β, and β is the negligible term in $TN \pm \beta$. Equation 4 can be rewritten as

$$HSS = \frac{2 \cdot TP(TN - FN)}{TP(TN + FP) + FP(FP + TN) + TP(FN + TN) + FN(FN + TN)} \tag{5}$$

Using the above assumption, equation 5 reduces to

$$HSS \approx \frac{2 \cdot TP \cdot TN}{2(TP \cdot TN) + (FP \cdot TN) + (FN \cdot TN)} \tag{6}$$

Factoring out TN gives us

$$HSS \approx \frac{TN(2TP)}{TN(2TP + FP + FN)} = \frac{2TP}{2TP + FP + FN} \tag{7}$$

Dividing both the numerator and the denominator by 2 gets us

$$HSS \approx \frac{TP}{TP + \frac{1}{2}(FP + FN)} \tag{8}$$

Which is equivalent to the F1-Score, proving that HSS reduces to F1-Score when $TN \gg TP$, FN, and FP.