Forecasting >100 MeV SEP Events and Intensity based on CME and other Solar Activities using Machine Learning

by

Daniel Lee Griessler

Bachelor of Science
Computer Science
Florida Institute of Technology
2020

Bachelor of Science
Mathematical Sciences
Florida Institute of Technology
2020

A thesis
submitted to the College 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, 2023

We the undersigned committee
hereby approve the attached thesis

Forecasting >100 MeV SEP Events and Intensity based on CME and other Solar Activities using Machine Learning by Daniel Lee Griessler

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

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

Marius C. Silaghi, Ph.D.
Professor
Electrical Engineering and Computer Science

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

Abstract

Title:

Forecasting >100 MeV SEP Events and Intensity based on CME and other Solar Activities using Machine Learning

Author:

Daniel Lee Griessler

Major Advisor:

Philip Chan, Ph.D.

There is a severe risk for astronauts and machinery from high intensity Solar Energetic Particle (SEP) events which can be mitigated through accurate forecast of their presence and peak intensity. By using characteristics of CME and other space weather phenomena, machine learning techniques have the potential to classify and predict the peak intensity of SEP events. The extreme scarcity of SEP events in current datasets poses a challenge to traditional machine learning techniques. In this work, we first demonstrate classifier machine learning techniques that can achieve an F1 score of 0.800 in forecasting SEP events. We then propose techniques for forecasting SEP peak intensity including Combining Richardson forecast (RC), learning Richardson Error (RE), and integrating retraining with DenseLoss (DL+rRT+AE). Finally, we demonstrate through DL+rRT+AE that we can achieve the same F1 score of 0.800 for forecasting SEP peak intensity.

Table of Contents

Abstract iii

List of Figures ix

List of Tables xi

Acknowledgments xvii

1 Introduction 1

1.1 Motivation 1

1.2 Problem Statement 2

1.3 Approaches 2

1.4 Contributions 3

1.5 Organization 5

2 Related Work 6

2.1 SEP Forecasting 6

2.1.1 Using Characteristics of Solar Flares and X-rays 6

2.1.2 Using Characteristics of CMEs 9

2.1.3 Using Characteristics of Other Space Weather Phenomena . . . 13

2.2 Machine Learning with Imbalanced Data 14

2.2.1 Imbalanced Classification 14

2.2.2 Imbalanced Regression 19

2.3 Explaining Predictions and Models 22

3 Description of SEP and CME Data 24

3.1 Features from DONKI and CDAW CME Catalogs 25

3.2 Features derived from DONKI and CDAW CME Catalogs 26

3.3 Features from outside DONKI and CDAW CME Catalogs 27

3.4 Feature value adjustments 28

3.5 Normalization of features 28

4 Forecasting SEP Events 29

4.1 Approaches 29

4.1.1 Regular Neural Network with Oversampling (cRegNN) 29

4.1.2 Classifier Re-training (cRT) 31

4.1.3 Classifier Re-training with Autoencoder (cRT+AE) 31

4.2 Experimental Evaluation 34

4.2.1 Evaluation Criteria 34

4.2.2 Evaluation Procedures 36

4.2.2.1 Procedures for Training Regular Neural Network with Oversampling 38

4.2.2.2 Procedures for Training cRT 39

4.2.2.3 Procedures for Training cRT+AE 39

4.2.3 Results 40

4.2.3.1 Regular NN with Oversampling (cRegNN) 41

4.2.3.2 Classifier Re-Training (cRT) 42

4.2.3.3 Classifier Re-Training with Autoencoder (cRT+AE) . 43

4.2.3.4 Comparing the Main Approaches 44

4.2.4 Analysis . . . 45

4.2.4.1 Regular NN with Oversampling (cRegNN) . . . 45

4.2.4.2 Classifier Re-Training (cRT) . . . 48

4.2.4.3 Classifier Re-Training with Autoencoder (cRT+AE) . 51

4.2.4.3.1 Feature Importance . . . 54

5 Forecasting SEP Intensities . . . 64

5.1 Approaches . . . 65

5.1.1 Regular Neural Network with Oversampling (rRegNN) . . . 65

5.1.2 Regression Re-training (rRT) . . . 66

5.1.3 Regression Re-training with Autoencoder (rRT+AE) . . . 66

5.1.4 Incorporating Richardson's model . . . 68

5.1.4.1 Richardson Network . . . 69

5.1.4.2 Incorporating Richardson Network . . . 70

5.1.4.2.1 Combining Richardson Forecast (RC) . . . 70

5.1.4.2.2 Learning Richardson Error (RE) . . . 71

5.1.5 DenseLoss (DL) . . . 72

5.2 Experimental Evaluation . . . 75

5.2.1 Evaluation Criteria . . . 75

5.2.2 Evaluation Procedures . . . 76

5.2.2.1 Procedures for Training Regular Neural Network with Oversampling . . . 77

5.2.2.2 Procedures for Training rRT . . . 77

5.2.2.3 Procedures for Training rRT+AE . . . 78

5.2.2.4 Procedures for Learning Richardson Coefficients . . . 78

5.2.2.5 Procedures for Training RC . . . 78

5.2.2.6 Procedures for Training RE 79

5.2.2.7 Procedures for Training DL 79

5.2.3 Results 81

5.2.3.1 Regular Neural Network with Oversampling (rRegNN) 84

5.2.3.2 Regression Re-training (rRT) 85

5.2.3.3 Regression Re-training with Autoencoder (rRT+AE) . 86

5.2.3.4 Learning Richardson Coefficients 86

5.2.3.5 Combining Richardson Forecast (RC) 89

5.2.3.6 Learning Richardson Error (RE) 91

5.2.3.7 DenseLoss (DL) 94

5.2.3.8 Comparing the Main Approaches 98

5.2.4 Analysis 100

5.2.4.1 Regression Re-training with Autoencoder (rRT+AE) . 101

5.2.4.2 Combining Richardson Forecast (RC) 104

5.2.4.3 Learning Richardson Error (RE) 106

5.2.4.4 DenseLoss (DL) 109

5.2.4.4.1 Feature Importance 110

6 Conclusions 115

6.1 Summary of Findings 115

6.2 Limitations and Possible Improvements 116

References 118

A Richardson's Model 122

B Diffusive Shock 124

C Changes to CME Data 126

List of Figures

4.1 Network Architecture for cRegNN. 30

4.2 Network architecture for cRT. 31

4.3 Network architecture for the Autoencoder model. 32

4.4 Network architecture for cRT+AE. 33

4.5 F1 score vs oversampling rate for classifier techniques. 41

4.6 Predicted classification score vs Actual classification for cRegNN, cRT, and cRT+AE. Events above dotted threshold line are predicted SEP events. FPs and FNs are in the upper left and lower right respectively. 45

4.7 Predicted classifier score vs 5 features for cRegNN with 20% oversampling. 47

4.8 Predicted classifier score vs 5 features for cRT with 60% oversampling. 50

4.9 Predicted classifier score vs 5 features for cRT+AE with 70% oversampling. 52

4.10 Predicted classifier score vs 5 features for cRT+AE with 70% oversampling. 60

5.1 Network architecture for rRT+AE. 67

5.2 Network architecture for learning Richardson's formula coefficients. . . 69

5.3 Network architecture for RC. 71

5.4 Network architecture for RE. 73

5.5 F1 score vs oversampling rate for rRegNN, rRT, and rRT+AE techniques. 82

5.6 F1 score vs oversampling rate for RC+rRegNN, RC+rRT, and RC+rRT+AE techniques. 91

5.7 F1 score vs oversampling rate for RE+rRegNN, RE+rRT, and RE+rRT+AE techniques. 94

5.8 F1 score vs α_{DW} for DL+rRegNN, DL+rRT, and DL+rRT+AE techniques. 94

5.9 Predicted vs Actual peak intensity for rRT+AE, RC+rRT+AE, RE+rRT+AE, and DL+rRT+AE. Perfect intensity predictions are on the diagonal dotted line. The horizontal and vertical dotted lines indicate the threshold for classification. FPs and FNs are in the upper left and lower right quadrants respectively. 100

5.10 Predicted ln peak intensity vs 5 features for rRT+AE with 10% oversampling. 102

5.11 Predicted ln peak intensity vs 5 features for RC+rRT+AE with 50% oversampling. 105

5.12 Predicted ln peak intensity vs 5 features for RE+rRT+AE with 20% oversampling. 107

5.13 Predicted ln peak intensity vs 5 features for DL+rRT+AE with 0.6 α_{DW}.110

5.14 Predicted classifier score vs 4 features for DL+rRT+AE 0.6. 113

List of Tables

3.1 Distribution of 100 MeV SEP, Elevated, and Background CME events. 25

4.1 Confusion matrix for classification tasks. 34

4.2 Distribution of events in our 3-fold data sets. 37

4.3 Epoch counts for cRegNN, cRT, and cRT+AE techniques used per oversampling percent. 39

4.4 Summarized metric results averaged over 5 runs for the cRegNN technique with the test set in 3-fold Dataset 1. Each row in a technique has an oversampled dataset by some percent indicated by the trailing number after the row technique name, e.g. cRegNN 10 is the cRegNN technique with 10% oversampling. The best value for each metric in cRegNN is underlined. The value of the metric is in bold if it is the best across all tested techniques. 41

4.5 Summarized metric results averaged over 5 runs for the cRT technique with the test set in 3-fold Dataset 1. Formatting is the same as in Table 4.4. 42

4.6 Summarized metric results averaged over 5 runs for the cRT+AE technique with the test set in 3-fold Dataset 1. Formatting is the same as in Table 4.4. 43

4.7 Summarized metric results averaged over 5 runs for the best F1 rows of the cRegNN, cRT, and cRT+AE techniques with the test set in 3-fold Dataset 1. Row formatting is the same as in Table 4.4. The best metric value is in bold. 44

4.8 The important features for the False Positive (FP) and False Negative (FN) events for cRegNN 20 from Figure 4.6a. The features are given across the top in abbreviated format: DONKI Date (DD), Latitude (Lat), Longitude (Lon), Linear Speed (Spd), Half Width (Hw), Acceleration (Acc), Actual 100 MeV Peak Intensity ln (Peak ln), Classifier Score (Score), and the Classifier Prediction (FP/FN). 46

4.9 The important features for the False Positive (FP) and False Negative (FN) events for cRT 60 from Figure 4.6b. Acronyms are the same as in Table 4.8. 49

4.10 The important features for the False Positive (FP) and False Negative (FN) events for cRT+AE 70 from Figure 4.6c. Acronyms are the same as in Table 4.8. 52

4.11 Overall Feature importance values ($\hat{I}_j$) for cRT+AE 70. 58

4.12 Feature importance values from LIME for FP and FN in Table 4.10 with cRT+AE 70. Includes feature importance values, W_{ij}, and contribution values, C_{ij}, with their respective rankings from largest to smallest feature importance value, $R_{W_{ij}}$ and $R_{C_{ij}}$. The C_{ij} shown are scaled by 1000. The top 3 are made bold, italic, and underlined, respectively (positive for FP, negative for FN) for the W_{ij} and C_{ij} values. 59

4.13 Feature Group Importance for cRT+AE 70 for the overall dataset ($\hat{I}_J$), the FP event ($\hat{W}_{iJ}$), and the FN event ($\hat{W}_{iJ}$) and their respective ranks ($R_{\hat{I}_J}$, $R_{\hat{W}_{iJ}}$ for FP, $R_{\hat{W}_{iJ}}$ for FN). 62

5.1 Confusion matrix for regression tasks. 76

5.2 Epoch counts for rRegNN, rRT, and rRT+AE approaches used per oversampling percent. 77

5.3 Epoch counts for 3-fold learning Richardson coefficients. 78

5.4 Epoch counts for RC+rRegNN, RC+rRT, and RC+rRT+AE approaches used per oversampling percent. 79

5.5 Epoch counts for RE+rRegNN, RE+rRT, and RE+rRT+AE approaches used per oversampling percent. 80

5.6 Pareto estimates for α parameter for 3-fold datasets. 80

5.7 Epoch counts for DenseLoss DL+rRegNN, DL+rRT, and DL+rRT+AE approaches used per oversampling percent. 82

5.8 Summarized regression metric results averaged over 5 runs for rRegNN, rRT, and rRT+AE techniques with 3-fold Dataset 1. Each row in a technique has an oversampled dataset by some percent indicated by the trailing number after the row technique name, e.g. rRegNN 10 is the rRegNN technique with 10% oversampling. The best value for each metric in each technique is underlined with the best overall value for each metric in bold. 83

5.9 Summarized classifier adapted metric results averaged over 5 runs for rRegNN, rRT, and rRT+AE techniques with 3-fold Dataset 1. Formatting is the same as in Table 4.4. 84

5.10 Original vs learned coefficients with their MAE of log intensity and intensity (MAE ln, MAE), MAE of log intensity and intensity for SEP events (MAE SEP ln, MAE SEP) calculated for SEP+Elevated dataset. Bold indicates the better metric value. 87

5.11 Classification metrics using ln Richardson's Equation with original vs learned coefficients calculated for SEP+Elevated+Background dataset. Bold indicates the better metric value. 87

5.12 Summarized regression metric results averaged over 5 runs for RC+rRegNN, RC+rRT, and RC+rRT+AE techniques with 3-fold Dataset 1. Formatting is the same as in Table 4.4. 89

5.13 Summarized classifier adapted metric results averaged over 5 runs for RC+rRegNN, RC+rRT, and RC+rRT+AE techniques with 3-fold Dataset 1. Formatting is the same as in Table 4.4. 90

5.14 Summarized regression metric results averaged over 5 runs for RE+rRegNN, RE+rRT, and RE+rRT+AE techniques with 3-fold Dataset 1. Formatting is the same as in Table 4.4. 92

5.15 Summarized classifier adapted metric results averaged over 5 runs for RE+rRegNN, RE+rRT, and RE+rRT+AE techniques with 3-fold Dataset 1. Formatting is the same as in Table 4.4. 93

5.16 Summarized regression metric results averaged over 5 runs for DL+rRegNN, DL+rRT, DL+rRT+AE techniques with 3-fold Dataset 1. Formatting is the same as in Table 4.4 except the trailing number is the α_{DW} instead of the oversampling percent. 95

5.17 Summarized classifier adapted metric results averaged over 5 runs for DL+rRegNN, DL+rRT, and DL+rRT+AE techniques with 3-fold Dataset 1. Formatting is the same as in Table 5.16. 96

5.18 Summarized regression metric results averaged over 5 runs for the best F1 across all techniques with 3-fold Dataset 1. Each row in a technique with either an oversampling rate or an α_{DW} indicated by the trailing number after the row technique name. The best value for each metric across all techniques is in bold. 98

5.19 Summarized classifier adapted metric results averaged over 5 runs for the best F1 across all techniques with 3-fold Dataset 1. Formatting is the same as in Table 5.18. 99

5.20 The important features for the False Positive (FP) and False Negative (FN) events in rRT+AE 10 visualized in Figure 5.9a. The features are given across the top in abbreviated format: DONKI Date (DD), Latitude (Lat), Longitude (Lon), Linear Speed (Spd), Half Width (Hw), Acceleration (Acc), Actual 100 MeV Peak Intensity ln (Peak ln), Predicted 100 MeV Peak Intensity ln (Pred), and the classification (FP/FN).101

5.21 The important features for the False Positive (FP) and False Negative (FN) events in RC+rRT+AE 50 visualized in Figure 5.9b. Acronyms are the same as in Table 5.20. 104

5.22 The important features for the False Positive (FP) and False Negative (FN) events in RE+rRT+AE 20 visualized in Figure 5.9c. Acronyms are the same as in Table 5.20. 106

5.23 The important features for the False Positive (FP) and False Negative (FN) events in DL+rRT+AE 0.6 visualized in Figure 5.9d. Acronyms are the same as in Table 5.20. 109

5.24 Overall Feature importance values ($\hat{I}_j$) for DL+rRT+AE 0.6. 111

5.25 Feature importance values from LIME for FP and FN in Table 5.23 with DL+rRT+AE 0.6. This Table has the same format as in Table 4.12. . . 112

5.26 Feature Group Importance for DL+rRT+AE 0.6 with the same format as Table 4.13. 113

C.1 Adjustments made to specific events in the dataset. 126

Acknowledgements

Firstly, I would like to thank my advisor, Dr. Philip Chan. His insights and deep understanding of the problems and machine learning techniques immensely continued to improve my comprehension from the beginning of my thesis to the end. The quality of this work would have much suffered without his guidance.

Secondly, I would like to thank Dr. Ming Zhang. Dr. Zhang provided thorough knowledge, comprehension, and guidance on the domain space. His expert knowledge and insights helped me better comprehend the physics throughout the research project. I would also like to thank my other committee member, Dr. Marius Silaghi. I have taken courses from him from Assembly and Operating Systems in my undergraduate years to Robotics and AI in my graduate studies. I gained a much deeper understanding of the low level mechanics and machinations of computers through his lectures and projects. I also gained a much a better insight and appreciation for AI in relation to robotics and its applications.

I would like to thank Peter Thomas and Peter Tarsoly who helped me get started on the project and offered feedback. I would also like to thank Cheryl Mitravich and my faculty advisor Dr. Terrance O'Connor for helping me navigate the paper and administrative aspects of my graduate studies.

Finally, I would like to thank my family and friends for their support. They were great sources of encouragement.

Chapter 1

Introduction

1.1 Motivation

Solar energetic particles (SEPs) are composed of mostly highly charged electrons and protons and are accelerated by various activity such as in conjunction with a solar flare on the Sun or at the front of a coronal mass ejection in space [11]. A solar particle storm, or SEP event, occurs when the SEPs have such high speed that they reach Earth travelling 93 million miles in less than an hour. [22]. An SEP event can cause severe damage to spacecraft and expose astronauts to dangerous radiation. For spacecraft, SEPs can fry electronics, corrupt computer programming, damage solar panels, and disorient a spacecraft's navigational star tracker. For astronauts as they are unprotected by Earth's magnetic field and atmosphere, SEPs can damage cells or DNA increasing future risk for cancer or cause acute radiation sickness as they pass through spacecraft or astronaut's skin. This makes it crucial to forecast their occurrence and intensity with high-fidelity.

Coronal mass ejections (CMEs) form after a solar eruption and are large clouds of solar plasma and embedded magnetic fields [11]. They expand, often measuring millions of miles across, as they travel through space and can collide with planetary magnetic fields. CMEs are a large contributor to the production of SEPs; CMEs often drive magnetic shocks that can accelerate SEPs to high energy levels [9]. CME features have been gathered through advanced Earth and Space based instruments. Through analysis and interrelation of the patterns in these CME features, it may be possible to forecast SEP event occurrence and forecast SEP event intensity.

1.2 Problem Statement

The first problem that we study is classifying a CME event to be associated with a SEP or a Non-SEP event using its set of features. The second problem that we study is predicting the ln peak intensity of 100 MeV protons associated with a CME event using its set of features. A 100 MeV SEP event is defined as an event in which the intensity of 100 MeV protons meets or surpasses a threshold of 1 proton flux unit (pfu). The dataset of CMEs that we were able to form is very imbalanced. The percentage of 100 MeV SEP events in the entire dataset is 0.57%. The imbalanced dataset presents a challenge both to the first problem of classification and to the second problem of regression.

1.3 Approaches

To form our dataset, we adapt the dataset from Thomas [19] whose dataset was created for 10 MeV SEP events to include only the 100 MeV SEP events. We can adapt his already imbalanced dataset because it contained events for >10 MeV, and this study is for a subset with >100 MeV.

In the first problem, we use three machine learning classifier architectures to predict if a CME event is an SEP event or not. Performance on the original dataset will be poor because of its heavy imbalance towards Non-SEP events. To compensate, we use oversampling to increase the performance on the otherwise rare occurring SEP events. We also experiment with improving performance by first learning the representation and then secondly learning the classification of a CME event in two stages. We use two-stage learning because it has been proven to improve performance in imbalanced classification problems in other applications.

In the second problem, we adapt the three approaches from the first problem to regression and additionally experiment with several other machine learning regression architectures to predict the peak intensity of >100 MeV protons associated with a CME event. We experiment with improving upon one of the features we use for training, the Richardson formula [15], in two of the additional architectures. We learn new coefficients for their equation and adapt it into a network in two new techniques, firstly to train a weighted combination between its output and our own prediction and secondly to train our network to model the error left over in its output. Finally, we apply an adapted technique, DenseLoss [17], which applies weights to individual instances based on the density distribution of the peak intensity values. The weights applied by DenseLoss offers customized weights for larger SEP events replacing the need for oversampling in this technique.

1.4 Contributions

In this work, we propose three main contributions, namely

- For forecasting SEP events, Classifier Re-Training with Autoencoder (cRT+AE) can achieve 0.800 in F1.
- For forecasting SEP peak intensity, we propose Combining Richardson forecast

(RC), learning Richardson Error (RE), and integrating retraining with DenseLoss (DL+rRT+AE).

- For forecasting SEP peak intensity, DenseLoss with Regression Re-Training and Autoencoder (DL+rRT+AE) can achieve 0.800 in F1.

The proposed techniques in our contribution for forecasting SEP peak intensity can be generalized to other regression related tasks. The key idea of the Combining Richardson forecast is to learn the weights for combining two forecasts together. We use Richardson's equation as the first forecast, and we output the second forecast with a neural network. In other problem domains, works may be completed to forecast in their respective domain, and they can combine their forecast with another such as an equation or model from their own previous works. The idea of learning Richardson Error is for our neural network to model the error remaining in the Richardson equation's forecast. This can be generalized to other problems as again they might have an equation or model from prior work that is fairly accurate, and they can apply their network to learn the error remaining in that forecast. The DL+rRT+AE is the proposed combination of techniques from several separate papers combined together. DenseLoss [17] was developed for general use in imbalanced regression problems. It was a new loss function that used a new weight, DenseWeight, applied per sample based on the target value's distribution to help overcome the imbalanced dataset. Regression re-training and autoencoder (rRT+AE) was found in previous works including Thomas [19] as a two stage training technique to separately train the feature representation and the regressor along with an autoencoder to discover new features. These techniques both apply to imbalanced regression problems in general, so their combination also applies to imbalanced regression problems. We apply it specifically to forecasting SEP peak intensity.

1.5 Organization

In Chapter 2, we discuss related work in SEP forecasting and Machine learning techniques to deal with imbalanced datasets. In Chapter 3, we discuss the features of CME events and details of our dataset. In Chapter 4, we describe our approaches for predicting the classification of SEPs. We then present the results and analyze both their performance and errors. In Chapter 5, we describe our approaches for predicting the peak intensity of >100 MeV protons followed by analyzing the results of their application.

Chapter 2

Related Work

2.1 SEP Forecasting

These works were focused on forecasting SEP events and other information. They varied in the intensity of SEP events that they studied with many studying >10 MeV events. We group them by their input features. There is a set that used solar flares and X-ray features, a set that used CME features, and a set that used other space weather phenomena.

2.1.1 Using Characteristics of Solar Flares and X-rays

Several of the papers in current literature used features of solar flares and X-rays to perform prediction for SEP events. Solar flares contribute to the acceleration of SEP particles making them more dangerous. Various characteristics of solar flares have been observed and used as input when predicting SEPs including X-ray readings. X-ray features can provide advanced warning and forecasting capabilities in real-world application since the information we can obtain from these features travels faster than other space weather effects.

Boubrahimi et al. [1] predicted SEP events >100 MeV using decision trees on proton and X-ray time series data. Their dataset was collected from measurements by Geostationary Operational Earth Satellites (GOES) including short and long X-ray channel data and proton channels that covered various ranges of MeV. They used undersampling to ensure their dataset was balanced selecting only enough negative events to equal the positive events. To perform classification, they generated features based on a 10-hour span window using a Vector Autoregression Model (VAR). A VAR allowed them to express each proton time series window as a linear function of past values of itself and other proton and x-ray time series. A feature vector was formed from the coefficients of the proton equations for a data sample. Missing values in the generated features were filled with the 3-nearest neighbors class-level imputation technique. Through this method, a missing value was filled by weighting the 3 closest neighbors in the same class based on the mean squared difference of the features that were not missing. The fully fleshed out feature vectors were fed into the decision tree model.

Brea et al. [2] improved on the Proton Prediction Model (PPM) cited from Balch (2008) to predict SEP events through two techniques. The first technique was logistic regression, a linear model used for classification. Weights for features were learned through gradient descent, and the output layer applied a sigmoid activation function to predict if the event was an SEP event. They used a fixed threshold of 0.5 to change the probability into a classification. The second technique was boosted decision trees through AdaBoost. AdaBoost applied classification through the sum of the predictions from an ensemble of decisions trees each with a single split. They used GridSearchCV to tune hyperparameters such as the size of the ensemble and the learning rate. They experimented with several sets of features. The first set was chosen to match the feature set of the PPM model they were improving on which included flare X-ray peak flux, integrated X-ray flux, presence of a type II radio burst, and presence of a type IV radio burst. In the second set, additional features were added including flare location, flare integrated flux, and flare X-ray temperature and emission measure.

Kahler and Ling [5] revisited prior studies of X-ray measurements in 0.05-0.4nm and 0.1-0.8nm bands and their application to >10 MeV SEP forecasting. The authors noted that X-ray flare events typically preceded CME events which can drive SEP events and have shorter time scales making them standard for SEP forecasting. After making observations about the relations between the ratio of the different X-ray bands and the location of the flare source to SEP occurrence, they used observed peak flux ratios from events in the western hemisphere of the sun as input features into their classification techniques. They applied multi-layer perceptron and k-nearest-neighbor classification techniques to classify events as either SEP or No-SEP.

Kasapis et al. [7] evaluated the potential use of Space-Weather MDI Active Region Patches (SMARP) related to solar flares to predict > 10 MeV SEP events. They compared their predictive potential with the baseline use of two other features: solar flare peak intensity and flare location. They applied two groups of ML algorithms. The first group was variations of Support Vector Machines (SVMs) originally designed to solve binary classification problems. SVMs map an input feature vector to a higher dimensional space forming a trained weight vector using a user-defined kernel. They used four different kernels: linear, second order polynomial, third order polynomial, and Gaussian Radial Basis Function (RBF). They used cross-validation to mitigate overfitting. A regularization parameter, C, controlled the scale of the SVM loss function which they varied. The second group was linear models i.e. regression methods where the target value is the linear combination of input features. A threshold value was used to classify the regression output. One linear model they applied was the ridge regression algorithm which minimized the Least Squares loss function with an additional hyperparameter controlled penalize term applied to the size of the coefficients. For this method, they did not apply cross-validation due to non-significant selection bias in their training data random picking process. Another linear model they applied was Logistic Regression which included ℓ_2 regularization as a penalty and the Limited-memory Broyden-Fletcher-Goldfarb-Shanno (L-BFGS) optimization algorithm as a solver. This model had a constant hyperparameter that controlled its regularization strength which they varied. For data construction, they utilized undersampling to form balanced training and test sets. They repeated constructing balanced training and testing pairs 100 times with negative samples randomly sampled without replacement and positive samples randomly sampled with replacement with different training/test splits. This was to guarantee with probability almost 1 that they covered the best and worst cases for their metrics.

2.1.2 Using Characteristics of CMEs

The main catalogs of data for CMEs included the SOHO LASCO CME Catalog at CDAW and Space Weather Database Of Notifications, Knowledge, and Information (DONKI) at CCMC. These works pulled raw features from either one or both catalogs and created derived feature values to use alongside.

Richardson et al. [15] assessed whether a formula created by Richardson previously that predicted peak intensity of 14- to 24-MeV protons in an SEP event at 1 AU to the solar event location could be used to predict the SEP intensity at any location at 1 AU. Richardson cited many patterns that had been identified in the properties of CMEs that were related to SEPs. For example, there was a widely reported correlation between the peak intensity of an SEP event and the expansion speed of its related CME. Conversely, the intensity tended to decrease as the connection angle increased between the related solar event and the magnetic field line that linked the observing spacecraft to the sun.

Other influences included, but were not limited to, occurrence of preceding CMEs and variations in the seed particle population for acceleration remaining from previous solar events. None of these patterns could perfectly predict the peak intensity of SEP events. The equation created by Richardson previously used the connection angle and speed of a CME to estimate a predicted peak intensity value. They calculated the peak intensity using Richardson's formula on CMEs collected from the DONKI catalog in a carefully chosen timeframe when the spacecraft observers were approximately equally spaced in longitude around the sun. They attempted to reduce the number of false positives they observed through filtering events based on speed and width. They experimented with including the type II radio emissions accompanying the CMEs which were believed to be evidence for particle acceleration at CME-driven shocks. Then, they experimented with including type III radio emissions usually associated with SEP events since large SEP events were almost always associated with bright, long-lasting type III emissions. Type III emissions were characterized in two ways. The first way was by its visual effect which split the CMEs into four groups: no type III, weak, moderate, and bright. The second way was by the duration for which the type III emission intensity was >6 dB accompanying each CME. Some of the events included in their dataset were used during the creation of the Richardson equation in the first place which made their results not completely independent. Therefore, they also experimented with applying the formula to an independent sample of CMEs from the CDAW catalog with filters for a minimum width and speed. They filled in missing longitude values through an automated procedure that compared CDAW CME parameters to NOAA flare reports with various considerations.

Bruno and Richardson [3] developed an empirical model using a 2-dimensional Gaussian to predict 10-130 MeV protons at 1 AU. They formulated that the intensity can be represented by the maximum intensity of the event E, $\Phi_0(E)$ and the 2D

Gaussian, $G(E, \delta)$:

$$\Phi(E, \delta) = \Phi_0(E)G(E, \delta) \tag{2.1}$$

where δ is the connection angle. Connection angle was calculated from the longitude and latitude of the event and the latitude and longitude of the SEP distribution peak.

Tarsoly [18] created a merged CME dataset and studied forecasting 10 MeV SEP events. To form the merged CME dataset, they matched CME events between the DONKI CME catalog containing measurements gathered with Sun-centric instruments and the CDAW CME catalog containing measurements gathered with Earth-centric instruments. In comparison, the DONKI catalog was found to be more suitable to build a model while the CDAW catalog had measurements taken from a better perspective which was part of the motivation to match up the two catalogs. They developed a matching algorithm handling cases from the ideal case of a one-to-one match to the less ideal cases such as multiple entries in either catalog or missing entries in CDAW. The final dataset contained a ratio of 60:1 non-SEP to SEP events. They then leveraged this catalog to forecast 10 MeV SEP events. They built off the work of Torres [21] by applying additional machine learning techniques. They explored classifier re-training (cRT) to separate learning the representation and classification of SEP events in two stages instead of jointly. They augmented cRT with the autoencoder technique they called cRT+AE. The autoencoder technique was supposed to help by learning new features from the data through an encoder and decoder pair. In the joined technique, there were two stages. In the first stage, they had two branches, the autoencoder branch and the classifier branch, whose loss functions were combined with a weighted sum controlled with a hyperparameter α. In the second stage, the autoencoder branch was discarded, the representation was frozen, and they reinitialized and relearned the classifier. To overcome the imbalanced dataset, they used oversampling throughout their techniques.

Thomas [19] studied predicting the intensity and times of 10 MeV SEP events. They used the imbalanced dataset composed by Tarsoly[18]. Their initial approach used random oversampling of SEPs to overcome the imbalance in the dataset. They further experimented with additionally randomly oversampling high-speed and large width Non-SEPs after observing that higher predicted events had similar features. By oversampling the Non-SEP instances in particular, they hoped to reduce their erroneous prediction since there would be more examples of Non-SEPs with those features. For their next approach, they adapted cRT to regression tasks and augmented it with an autoencoder which they called rRT+AE. The rRT+AE method had two stages. In the first stage, there were two branches: one branch was the autoencoder branch, and the other branch was the regressor branch trying to predict the peak intensity. In the second stage, the autoencoder branch was discarded, the representation was frozen, and they reinitialized and relearned the regressor. Their final approach was called adaptive calibration which was also adapted to regression from a classifier technique. This technique had three stages. The first stage was the same as the rRT+AE method, and, from it, they extracted the representation frozen for use in the next stage. In the second stage, they added two regression layers forming two branches. One branch was trained on a uniform training distribution to learn a regressor that performs well on most of the instances in the uniform distribution. The other branch was trained on an oversampled distribution to perform well on the minority instances. The weights from these layers at the end of training were frozen. In the third stage, they added a third branch which learns a calibration value from the internal representation form of the input. The calibration value is then used in the weighted sum of the frozen two branches from the second stage. They applied these same techniques to predict the time it takes for an event to reach threshold and peak intensity.

2.1.3 Using Characteristics of Other Space Weather Phenomena

These other works either used multiple of the previous categories or used other space weather phenomena to perform SEP predictions. The features were derived from a variety of sources such as flux of protons and radio waves and combinations of the previous features in solar flares, x-ray, and CMEs.

Kim et al. [8] applied an artificial neural network (ANN) and a genetic algorithm (GA) to predict >10 MeV SPE events using solar radio flux (SRF) at 2800 MHz, 1415 MHz, and 610 MHz from 1976-1994. SPE is an older term for SEP. Their ANN design had 3 hidden layers fed by two input features consisting of overall rate of increase and daily total SRF. The first hidden layer used a tan-sigmoid activation function, the second hidden layer used a log-sigmoid activation function, and the third layer used a linear transfer activation function. The output was the number of SEPs predicted. For the GA input, they normalized the values of daily totals SRF at 2800, 1415, and 610 MHz between 0 and 1. They used the overall equation from the ANN approach, from the input to output layers with successive weight and activation function applications, as the fitness function to be minimized in GA. They applied the GA to find optimum parameters for predicting SPEs from SRF.

Torres et al. [20, 21] accomplished two tasks. The first task was a classification problem forecasting SEP events. To perform this task, they aggregated features from the CDAW CME catalog, a set of derived features, and features occurring at the time of the CME from other sources. They submitted this aggregated feature set as one of their main contributions. The features were fed into a multilayer perceptron neural network. The ratio of SEP events, 1 to 300, revealed an imbalanced dataset towards non-SEP instances. To compensate, they used oversampling to increase the relative importance of the SEP events. To perform analysis on the features that contributed to the errors in their model, they formulated and used a feature importance algorithm based on the learned model weights. This feature importance calculation was another contribution of their work. The second task was forecasting SEP intensity using time series data of electron intensities for >0.25 and >0.67 MeV channels, proton intensities from the >10 MeV channel, and program generated phases. Their time series input data included data from the past two hours, and their output was predicting the output proton flux either at the next hour or next half hour. Firstly, they compared the multilayer perceptron algorithm to a recurrent neural network (RNN) which were designed for better use with time-series data. However, there was still an imbalance in the data which could not be addressed through methods such as oversampling. Instead, they separated their data in two approaches into different intensity ranges with separate models per range. In the first approach, they set high, medium, and low thresholds to create the three groups selected through manual inspection. In the second approach, they split labels into rising (between onset and peak), falling (between peak and end), and background (everywhere else) for the related event timestamps using their designed program. In another set of experiments using the three prior approaches, they added x-ray features interpolating missing data as needed.

2.2 Machine Learning with Imbalanced Data

2.2.1 Imbalanced Classification

These works focused on machine learning techniques when performing classification tasks on imbalanced data. When training a machine learning model, an input is encoded into an internal representation inside the model before a classification is output based on that internal representation. We group these works based on how they handle training the feature representation and the classifier parts of a machine learning network.

We first discuss the first category focused on training the two parts jointly. Zhou et al. [28] proposed a method of learning the representation and classifier at the same time. Their solution was offered to alleviate issues they found in the general methods of re-balancing and re-weighting used to handle long-tailed problems. They stated that re-balancing through over-sampling can cause over-fitting of the minority events while under-fitting the overall data distribution. They also stated that re-weighting will disfigure the original data distribution. Their proposed method was Bilateral-Branch Network (BBN). It is Bilateral because there are two branches: the ”conventional learning branch” to train the representation and the ”re-balancing branch” to train the classifier. The conventional learning branch used a uniform sampler i.e. the original data distribution while the re-balancing branch used a reversed sampler i.e. the inverse of the original data distribution. The outputs of the two branches were combined through a softmax classification layer with an automatically generated parameter α based on the epoch number. The α parameter started with more weight for the representation learning and then gradually shifted over epochs to provide more weight to the classifier learning. Their model used this α value in their loss function to weight the cross-entropy loss from each branch in a linear sum. After training, α was set to 0.5 during testing since both branches were equally important.

Wang et al. [23] questioned whether the typical cross-entropy loss function was ideal for learning features from an imbalanced dataset for classification. They proposed a custom network structure with two branches. The first branch learned the representation, i.e. the features, using a contrastive loss function. The goal was that instances in the feature space should be close to other instances of the same class while far from instances of other classes. The input to the first branch was an anchor point with positive samples from the same class and negative samples from other classes. The second branch learned the classifier using a cross-entropy loss function. The input to the second branch was image and label pairs class-sampled. The two branches shared an initial backbone network to learn the representation vector, r, of an image input, x. In the first branch for feature learning, the representation vector, r, was mapped to a vector representation, z, through a nonlinear multiple-layer perceptron with one hidden layer. The ℓ_2 normalization was applied to z before being fed to a supervised contrastive loss function. They identified a memory consumption issue when applying the supervised constrastive (SC) loss function linear to the product of the number of positive and negative samples included. To resolve this issue, they proposed a loss function called prototypical supervised contrastive (PSC) loss. In PSC, their goal was to learn a prototype for each class and force the representation of samples in that class to be close to their class prototype and far away from the prototypes of the other classes. They also generalized PSC to multiple prototype supervised contrastive (MPSC) loss to support multiple prototypes per class. In the second branch for classifier learning, the representation vector, r, was fed to a linear layer to predict the likelihood that the representation belonged to each class. These likelihoods were fed to a cross entropy loss function. The two loss functions were combined through a parameter α calculated based on the epoch number.

The second category focused on decoupling the training of the feature representation and the classifier training parts. Kang et al. [6] explored several techniques to decouple the representation learning from the classifier learning for long-tailed classification including Classifier Re-training (cRT), Nearest Class Mean classifier (NCM), τ-normalized classifier (τ-normalized), and Learnable weight scaling (LWS). They justified this decoupled approach through experimentation finding better performance in decoupling into two stages instead of training both the representation and classifier jointly for long-tailed classification. With the ImageNet-LT dataset, cRT had the highest long-tail recognition accuracy using most of the backbone architectures. The cRT technique decoupled the representation learning from the classifier learning through two stage training. In the first stage, they sampled uniformly from the dataset to train the entire network. The outcome of the first stage was a model that had learned the representation of the data through the encoded trained weights. The representation weights were then frozen for use in the second stage. Since these weights were frozen, during the second stage training they were not updated through gradient backpropagation. In the second stage, the classifier weights were randomly re-initialized, and the model was provided a class-balanced dataset to optimize the classifier. The class-balanced dataset was generated through over or undersampling to equalize the representation of all classes. This equal representation helped equalize each class's influence on the backpropagated gradient during classifier training.

Zhang et al. [26] explored two-stage learning starting with an ablative study into the two stages. For the first stage when learning the representation, they found instance-based sampling produced better results and that learning the features can lead to large performance gain. For the second stage when learning the classifier, they explored methods such as cRT and found a large performance gap from the ideal classifier. They posited this performance gap could be shortened by adjusting the distribution of the dataset in the second stage. They proposed their own two-stage learning scheme. Their first stage learned the features of the data using the imbalanced dataset with instance-balanced sampling. In the second stage, they froze the representation and introduced two techniques. In the first, an adaptive calibration function calculated the weighted sum of the original class score from the first stage and a transformed class score to provide the final output classification vector. The transformed class score was the original class score adjusted by a linear equation with calibration parameters learned for its class. The weight for the linear sum of these scores was a defined confidence score function $\sigma(z)$ implemented in a linear layer followed by a non-linear activation function trained during the second stage. The confidence score was thus tuned to the feature representation z provided by the frozen representation from the first stage for its associated input x. The calibrated class scores were combined to form a predicted distribution for the model. The second technique they termed "General-ized Re-weighting" was a method for re-weighting the classes in the loss function. The predicted distribution from their first technique is aligned with a reference distribution chosen by the model trainer which they suggested should favor a class-balanced distribution. The alignment is done through minimizing the expected KL-divergence between the predicted distribution and the reference distribution.

Zhong et al. [27] discovered that models trained on long-tailed datasets and two-stage models were over-confident, and classification models tended to inaccurately label minority classes. They measured overconfidence through the widely used expected calibration error (ECE), the difference between the accuracy of the model and the model's confidence in its predictions. They first applied mixup with two-stage models such as cRT which helped lower the over-confidence with representation learning, but it had no effect or, if there was an effect, it was negative with classifier learning. Although they do not specify, mixup comes from Zhang et al. [25] who proposed a method of data augmentation by combining the input vectors and labels of pairs of samples through weighted linear sums. The samples generated by mixup encourage the model to perform linearly in-between actual training samples. Zhong et al. proposed a Mixup Shifted Label-Aware Smoothing model (MiSLAS) to address the issues of over-confidence and limited improvement in the classifier learning from mixup. This technique introduced a scaling term in the cross-entropy loss function per class. They required that the scaling term would be inversely proportional to the number instances in its class to resolve the disparity of much larger weight norms for majority classes vs minority classes. They also introduced a shift learning strategy for two-stage model training after showing a unreasonable comparison of statistics such as mean and variance between different amounts of class-based oversampling.

2.2.2 Imbalanced Regression

These works focused on machine learning techniques for regression tasks with imbalanced datasets. Their implemented approaches corrected the imbalance in some way.

We first discuss the first category focused on balancing the loss function. Steininger et al. [17] introduced a sample weighting approach called DenseWeight which they included into a cost-sensitive learning approach called DenseLoss. These were meant to be used with imbalanced regression datasets. DenseWeight used the distribution of the target values to weight rarer samples higher in the loss function than the more common samples. The weights were applied per sample, so samples within the same class such as outlier rare events had even higher weights than other rare samples with lower target values. To calculate the DenseWeight, they started by estimating the target value distribution using kernel density estimation. Then, they normalized the density value into the range $[0, 1]$. Next, they ensured that the weights did not get smaller than a small positive constant ϵ. Finally, they made the mean weight 1. In DenseLoss, the loss per sample was calculated using the user-selected original metric weighted by its DenseWeight. They used a parameter α to manipulate how much DenseWeight was applied.

Ren et al. [13] performed a statistical analysis of Mean Squared Error (MSE), the standard loss function in regression, and found it will cause regressors to underestimate rare labels. They proposed Balanced MSE which used the training label distribution to create a balanced prediction across labels. They defined Balanced MSE as the sum between standard MSE and a balancing term which contains an integral weighted by the training distribution. They showed through proof a unified statistical view of imbalanced classification and regression through their definition of Balanced MSE. They also showed closed-form calculations of the integral in their Balanced MSE definition. The first they called GMM-based Analytical Integration (GAI) where GMM stands for Gaussian Mixture Model. This approach required expressing the distribution of the training labels as a Gaussian. To remove this constraint, they also offered closed-form solutions based on the Monte Carlo Method (CMC). The first was called Batch-based Monte-Carlo (BMC). BMC employed a tunable parameter with random batches to calculate a closed-form solution within each training batch. The second was called Bin-based Numerical Integration (BNI) which divided the label space into evenly distributed bins before applying KDE to estimate the label distribution per bin to form the closed-form solution. There was a hyperparameter in their integral, σ_{noise}, that they optimized by making it learnable.

The second category involved balancing the feature space. Yang et al. [24] identified three major challenges in imbalanced regression tasks. The first was with the continuous nature of target values there is a lack of hard boundaries separating them into classes making it difficult to adapt techniques such as re-weighting or re-sampling from imbalanced classification tasks. The second was that metrics calculated with continuous target values such as distance have impact on how imbalance is measured. The third was possible missing data motivated the application of strategies such as interpolation or extrapolation. To address these challenges, they proposed two techniques. The first technique they termed label distribution smoothing (LDS). They applied a symmetric kernel function to the label density distribution. The produced smoothed label distribution was closer aligned to the similarity of the continuous target values whose data samples likely had some overlap e.g. images of close ages. This allowed them to apply techniques such as re-weighting for addressing the remaining issue of the imbalanced dataset directly as LDS did not alone fix it. The second technique they termed feature distribution smoothing (FDS). FDS was motivated by the assumption that the output was continuous, so the feature space should also be continuous. Thus, by collecting the feature mean and variance and applying a smoothing kernel function, a new smoothed representation could be generated. The FDS technique sat as an extra layer between the last representation layer and the output layer calculating the evolving smoothed statistics to adjust the internal representation before sending it to the output regression function.

Gong et al. [4] proposed a RankSim regularizier on the intuition that, for regression problems, the natural order of the target values should be reflected in the feature representation. They gave the example of age as target value, and they state that the feature representation of a 21-year-old should be closer to that of a 25-year old rather than a 70-year old. They started outlining their RankSim method by defining a ranking function through a minimizer of a linear combinatorial objective to ensure that backpropagation could be applied using a tunable gradient of continuous interpolation. They obtained a pairwise similarity matrix in the label space by applying a similarity function on the labels in a subset of all samples, and they obtained a pairwise similarity matrix in the feature space by applying a similarity function to the feature representation of the elements in the subset. They ranked each similarity matrix using their ranking function. They defined their loss function as the sum of the loss between the ranks in label and feature space. This effectively encouraged the sorted list of labels to match the sorted list of features as closely as possible. This method does not fix the imbalanced dataset issue, so they applied it to existing methods such as rRT, LDS, and FDS.

The third category focused on balancing the input data. Moniz et al. [10] adapted the SMOTEBoost approach originally designed for classification tasks to imbalanced regression. They first defined a relevance function that mapped target values into the range $[0, 1]$ with the value 1 indicating the most relevant and 0 indicating the least relevant. They then augmented four different boosting algorithms with SMOTE. With some modifications per algorithm, in each iteration, the current distribution of samples was augmented with new synthetically highly relevant examples generated using SMOTE. A new model was generated from this distribution. The error was then calculated as the sum of the weights in samples with high relative error between their predicted and actual target values. This error becomes part of the update parameter applied to the current distribution for use in the next iteration. The output is the weighted sum of all the model predictions, one per epoch, with weights proportional to a function of their error rates on the training set.

2.3 Explaining Predictions and Models

Ribeiro et al. [14] outlined solutions for ”trusting a prediction” and ”trusting a model”. For the former solution, they introduced Local Interpretable Model-agnostic Explanations (LIME) to identify an interpretable model over a interpretable representation that is faithful to the classifier. An interpretable model could be understood by humans. Similarly, an interpretable representation was an encoding that can be understood by humans while the representation of the actual model may be more complex and incomprehensible. The model-agnostic part meant they made no assumptions about the model they were explaining. To learn the local behavior of the model around an instance, they generated samples around the instance to form a linear fit that was locally faithful. The output was a set of local feature importance weights for the given sample. For the latter solution, they looked for a global view of a model by explaining a set of instances. The pick algorithm they initially defined was NP-hard, so they introduced submodular pick (SP-LIME) with a greedy algorithm that iteratively added the instance with the highest marginal gain to the solution. Once they had a set of instances, they used their former solution to generate the local importance of the features of each instance. They let I_j denote the global importance for feature j and W_{ij} denote the local importance for instance i of feature j. They wanted the global importance of features present in many instances to be be higher than other features not present in many instances. They also did not want to select instances with similar explanations to avoid redundancy. To calculate global importance I_j for some feature j, they used the equation:

$$I_j \leftarrow \sqrt{\sum_{i=1}^{N} |W_{ij}|} \tag{2.2}$$

where an instance i had local importance for feature j represented by the value W_{ij} and where N was the number instances.

Chapter 3

Description of SEP and CME Data

Our dataset includes CME events from two sources: SOHO LASCO CME Catalog at CDAW and Space Weather Database Of Notifications, Knowledge, and Information (DONKI) at CCMC that span from 4/3/2010 to 9/6/2017. Previously, Tarsoly matched up the CME events from the CDAW and DONKI catalogs to form a cohesive dataset [18]. Tarsoly's dataset for CMEs associated with SEPs with >10 MeV protons was already an imbalanced dataset with SEP events occurring very rarely compared to Non-SEP events. From Tarsoly's dataset, our dataset for CMEs associated with 100 MeV SEP events was derived and formed a more imbalanced dataset. Tarsoly's dataset included 44 SEP events, 39 Elevated events, and 2309 Background events vs our dataset with 13 SEP events, 16 Elevated events, and 2256 Background events as summarized in Table 3.1. The reduced number of SEP events (from 44 down to 13) makes the dataset more imbalanced and the forecasting of 100 MeV SEP events more difficult. To augment these sparse SEP events whose peak intensity of 100 MeV protons are ≥1 pfu, we identified Elevated proton events as those whose 100 MeV protons peak intensities are between $1/e^2$ and 1 pfu and added them to the dataset.

Within the problem of classification, we distinguish between two classes: SEP and

Table 3.1: Distribution of 100 MeV SEP, Elevated, and Background CME events.

Event Category	Peak Intensity (pfu)	Number of Instances	Percentage
SEP	≥ 1	13	0.57
Elevated	$> \frac{1}{e^2}$ and < 1	16	0.70
Background	$\frac{1}{e^2}$	2256	98.73

Non-SEP. The Elevated and Background event categories are combined together to form the Non-SEP events. Therefore, if a CME event is an SEP event, then it has a value of 1, but if a CME event is a Non-SEP event then it has a value of 0.

Within the problem of regression, we are predicting the ln peak intensity of 100 MeV protons of each event. We have the peak intensity values for each CME, so we take the ln to form the ln peak intensity per CME.

In both the classification and regression problems, we have a set of input features. We used similar features to Torres [21], Tarsoly [18], and Thomas [19]. There are features that come from the DONKI and CDAW catalogs directly. There are a set of features derived using the raw DONKI and CDAW values. Finally, there are features from outside the DONKI and CDAW catalogs.

3.1 Features from DONKI and CDAW CME Catalogs

From the DONKI catalog, we included the following features:

- Latitude
- Longitude
- Half Width
- Linear Speed

From the CDAW catalog, we included the following features:

- Acceleration
- 2nd order speed final
- 2nd order speed at 20 solar radii
- Central Position Angle (CPA)
- Measurement Position Angle (MPA)
- Whether or not the CME is a Halo (CPA = 360°)

3.2 Features derived from DONKI and CDAW CME Catalogs

Tarsoly [18] and Thomas [19] calculated a set of features calculated from the past CMEs they called the CME History features. We calculated the same history features for our dataset. Specifically, the history features include:

- Number of CMEs in the Past Month
- Number of CMEs in the Past 9 Hours
- Number of CMEs with Speed over 1000 km/s in the past 9 Hours
- Maximum Speed for a CME in the Past Day

The speed referenced in the speed history features is the DONKI speed.

We calculated additional features following along with Thomas [19].

- V Log V

- Richardson's equation
- Diffusive Shock

The V Log V feature is exactly what it sounds like and is derived from the DONKI speed:

$$feature_{VLogV} = V * ln(V) \tag{3.1}$$

Richardson et al. modeled peak intensity of CMEs from the CDAW catalog with proton intensities from 14- to 24-MeV [15]. We used part of Richardson's equation as one of the input features. For more details of our Richardson feature see Appendix A.

Torres outlines the calculation for Diffusive Shock in their Appendix A [21]. We adapted the Diffusive Shock equation by replacing some of the constants with the appropriate values for our dataset. For more details of our Diffusive Shock feature see our Appendix B.

3.3 Features from outside DONKI and CDAW CME Catalogs

We have additional features from outside the DONKI and CDAW catalogs:

- Daily Sunspot count
- Type II visualization area

The Daily Sunspot count was directly linked by DONKI entry and was provided by the Solar Influences Data Analysis Center [16].

Type II visualization area was calculated from the CDAW catalog of Type II bursts associated with timestamps of CME events. The Type II visualization area is the difference in start and end time for the Type II bursts in minutes multiplied by the difference in the starting and ending frequency of Type II burst in kHz.

3.4 Feature value adjustments

Almost all the raw feature values drawn directly from the DONKI and CDAW catalogs were the original, unchanged values reflected in their respective catalogs. However, under expert guidance, we made a few adjustments to specific events based on careful, individual analysis. The motivation and summary of these adjustments is summarized in Appendix C.

3.5 Normalization of features

Before being passed into a neural network, each feature, f, is scaled into the range $[0, 1]$ to avoid biasing the network with features that have large values using the following equation:

$$f_i = \frac{f_i - f_{min}}{f_{max} - f_{min}}, \tag{3.2}$$

where the feature f has value f_i in CME event i, $f_{min} = min(f_i)$, and $f_{max} = max(f_i)$. For the Diffusive Shock and Type II Area Visualization features, we took the log before normalization, that is using $ln(f_i)$ instead of f_i in Equation 3.2 and when calculating f_{min} and f_{max}. The difference between the maximum and minimum Diffusive Shock was on the order of 10^{-13}, and the difference between the maximum and minimum Type II Area Visualization was on the order of 10^7. Without the log before normalizing, the values for the events on the lower end of the ranges for these two features would become insignificant which is not desired during normalizing.

Chapter 4

Forecasting SEP Events

The first problem that was studied in this work was classifying a CME event as an SEP or a Non-SEP event using its set of input features as outlined in Chapter 3. An SEP event is a CME with 100 MeV protons whose peak intensity has a value $\geq$1. These events are of particular importance because they cause the most damage if undetected.

4.1 Approaches

All neural networks were implemented using Keras on top of TensorFlow. All neural networks use a sigmoid activation function to output a prediction, p, in the range $[0, 1]$ predicting if the event is considered an SEP event. For each technique, we select the threshold, t, that maximizes the performance. An event, i, is predicted to be an SEP event if $p_i \geq t$.

4.1.1 Regular Neural Network with Oversampling (cRegNN)

The regular neural network is a multi-layer Leaky ReLU classifier using 2 hidden layers. Figure 4.1 depicts the architecture of the network. We use the binary cross entropy

(CE) loss function during training:

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

where y is the observed class, and $\hat{y}$ is the predicted score between $[0, 1]$. We find a threshold, t, during evaluation that transforms the predicted score, p, into 1 if $p \geq t$ or 0 otherwise. In our case, a prediction of 1 indicates a CME is an SEP event and a prediction of 0 indicates a CME is not an SEP event. The first instance was run with the original, imbalanced dataset.

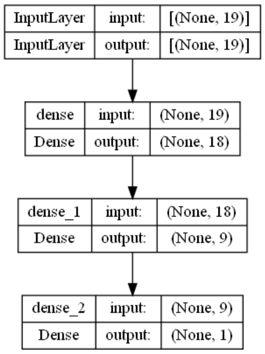

InputLayer | input: | [(None, 19)]
InputLayer | output: | [(None, 19)]
dense | input: | (None, 19)
Dense | output: | (None, 18)
dense_1 | input: | (None, 18)
Dense | output: | (None, 9)
dense_2 | input: | (None, 9)
Dense | output: | (None, 1)

Figure 4.1: Network Architecture for cRegNN.

We next explored oversampling the minority SEP and Elevated events to improve performance. Events are duplicated by replicating them until a certain percentage of duplication is achieved. For example in 10% oversampling, 5% of the samples are SEP events, 5% of the samples are Elevated events, and 80% of the samples are Background events. We abuse the notation to say 0% oversampling is the original, imbalanced dataset. The network architecture is the same as in Figure 4.1, but the training data is oversampled to increase the importance of the SEP and Elevated events.

4.1.2 Classifier Re-training (cRT)

We also explored a technique called classifier re-training (cRT) [6]. This technique separates training into two stages. In the first stage, the NN model is trained on the original, imbalanced dataset to learn the features in the hidden units. Then, all but the output layer is frozen and reused along with an extra hidden layer and a reinitialized and retrained output layer in the second stage which is trained using a class-balanced dataset to learn the classifier. Figure 4.2 illustrates the two stages. The green section is frozen and reused from stage 1 to stage 2. The orange section is retrained and reinitialized in stage 2.

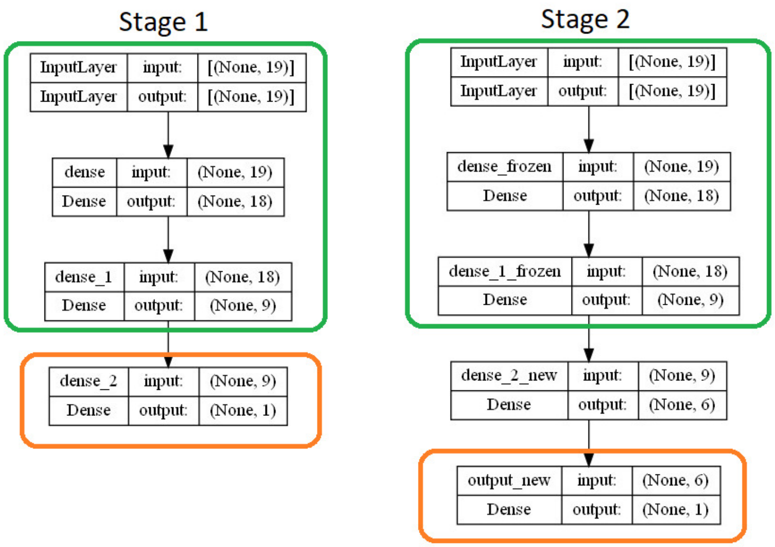

Stage 1
InputLayer | input: | [(None, 19)]
InputLayer | output: | [(None, 19)]
dense | input: | (None, 19)
Dense | output: | (None, 18)
dense_1 | input: | (None, 18)
Dense | output: | (None, 9)
dense_2 | input: | (None, 9)
Dense | output: | (None, 1)
Stage 2
InputLayer | input: | [(None, 19)]
InputLayer | output: | [(None, 19)]
dense_frozen | input: | (None, 19)
Dense | output: | (None, 18)
dense_1_frozen | input: | (None, 18)
Dense | output: | (None, 9)
dense_2_new | input: | (None, 9)
Dense | output: | (None, 6)
output_new | input: | (None, 6)
Dense | output: | (None, 1)

Figure 4.2: Network architecture for cRT.

4.1.3 Classifier Re-training with Autoencoder (cRT+AE)

First, an overview of the autoencoder. The autoencoder model consists of the encoder and decoder parts. The encoder transforms the input into an intermediate representation in the model. The decoder transforms the intermediate representation to the original input. Figure 4.3 illustrates the autoencoder architecture. To train the autoencoder, we use the mean square error (MSE) loss function:

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

where N is the number of training samples, x is the input vector, and $\hat{x}$ is the output of the autoencoder. The autoencoder's encoded intermediate representation, here the z-layer, holds new features based on the training data.

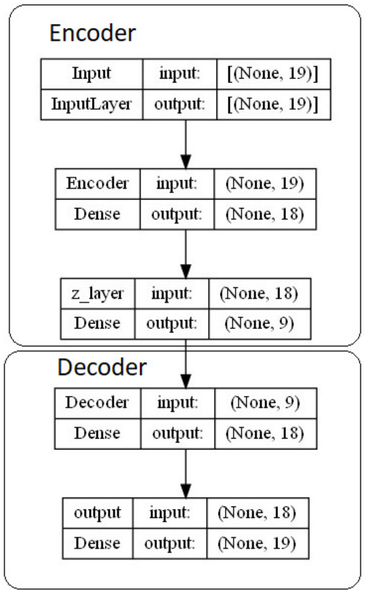

Encoder
Input | input: | [(None, 19)]
InputLayer | output: | [(None, 19)]
Encoder | input: | (None, 19)
Dense | output: | (None, 18)
z_layer | input: | (None, 18)
Dense | output: | (None, 9)
Decoder
Decoder | input: | (None, 9)
Dense | output: | (None, 18)
output | input: | (None, 18)
Dense | output: | (None, 19)

Figure 4.3: Network architecture for the Autoencoder model.

Tarsoly proposed a combination of cRT and autoencoder called cRT+AE [18]. We adapt his technique to our model with a few modifications. We use a joint loss function with the α tunable parameter, but only with one of the functions:

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

The goal is to estimate an α value to roughly equalize the influence of both the autoencoder and classifier branches. Our network architecture of the cRT+AE approach is very similar structurally to Tarsoly's as illustrated in Figure 4.4. To learn the α parameter, we train the autoencoder and classifier branches illustrated in Figure 4.4 separately and then combine their errors:

$$\alpha = \frac{1}{N} \sum_{n=1}^{N} \frac{\mathcal{L}_{CE_n}}{\mathcal{L}_{MSE_n}}, \tag{4.4}$$

where N is the number of epochs, $\mathcal{L}_{CE_n}$ is the binary cross entropy loss for epoch n, and $\mathcal{L}_{MSE_n}$ is the mean squared error loss for epoch n.

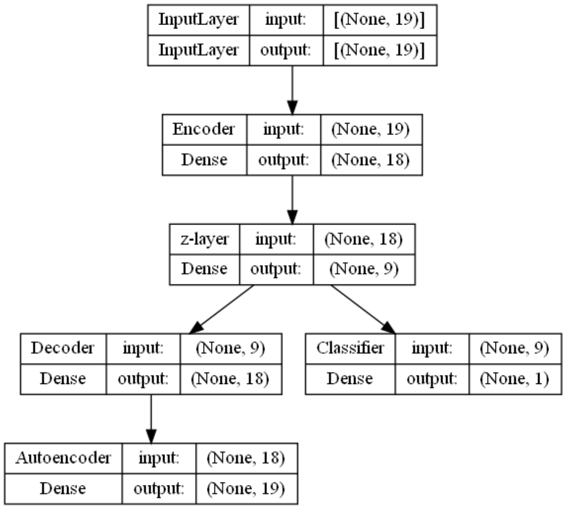

InputLayer | input: | [(None, 19)]
InputLayer | output: | [(None, 19)]
Encoder | input: | (None, 19)
Dense | output: | (None, 18)
z-layer | input: | (None, 18)
Dense | output: | (None, 9)
Decoder | input: | (None, 9)
Dense | output: | (None, 18)
Classifier | input: | (None, 9)
Dense | output: | (None, 1)
Autoencoder | input: | (None, 18)
Dense | output: | (None, 19)

Figure 4.4: Network architecture for cRT+AE.

With the α value estimated, we train the cRT+AE network in two steps. In the first step, we train the network shown in Figure 4.4 using the original, imbalanced dataset. The goal of the first step is to learn the features of the input data which will be preserved in the z-layer like the cRT method. In the second step, the Decoder/Autoencoder branch is discarded, the layers from the InputLayer up to the z-layer are frozen and reused, a new hidden layer is inserted between the z-layer and the Classifier, and the Classifier layer is reinitialized and retrained. The resulting network architecture is the same as the second stage of cRT, and we use an oversampled dataset for training.

4.2 Experimental Evaluation

4.2.1 Evaluation Criteria

We split data into two classes, SEP and non-SEP, based on their actual peak intensity. If a CME event has a peak intensity ≥1 pfu, then we classify it as an SEP event. All other events are non-SEP events. The classification models use a sigmoid activation function which outputs a prediction, p, that a CME event is an SEP event. With a threshold, t, we can classify the output as SEP if $p \geq t$. Therefore, the CME events that are classified as SEP events and have a prediction $p \geq t$ are ”true positives”. The same CME event classified as an SEP event would be a ”false positive” if its prediction $p < t$. The confusion matrix is displayed in Table 4.1.

Table 4.1: Confusion matrix for classification tasks.

	Prediction $p \geq t$	Prediction $p < t$
Actual Peak Intensity ≥1	True Positive (TP)	False Negative (FN)
Actual Peak Intensity <1	False Positive (FP)	True Negative (TN)

From the confusion matrix categories, we define a set of metrics beginning with precision and recall. Let 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 then:

$$precision = \frac{TP}{TP + FP} \tag{4.5}$$

$$recall = \frac{TP}{TP + FN} \tag{4.6}$$

Precision measures the fraction of correctly identified SEP events over all SEP event classifications. Recall measures the fraction of correctly identified SEP events over all actual SEP events. A combined metric, F1-Score, gives an insight into a measure of the algorithm's performance on both measurements.

$$F1 = 2 \times \frac{precision \times recall}{precision + recall} = \frac{TP}{TP + \frac{1}{2}(FP + FN)} \tag{4.7}$$

We add an additional metric used in the astrophysics community for binary classification problems: the true skill statistic (TSS). TSS is defined as the difference between the true positive rate (TPR) and the false positive rate (FPR). TSS measures the trade-off when more events are classified as true positives which is typically accompanied by an increase in the number of false positives.

$$TSS = \frac{TP}{TP + FN} - \frac{FP}{FP + TN} = TPR - FPR \tag{4.8}$$

The final metric we use to measure performance is the Heidke-Skill Score which measures how well the model performs relative to a random selection. A score of 0 indicates the model is not better than random. A score <0 indicates worse than random while a score >0 indicates better than random.

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

These are the same metrics as used in Thomas [19]. Thomas proved that HSS reduces to the F1-score when the number of true negatives is much greater than the other values in the confusion matrix.

4.2.2 Evaluation Procedures

Our dataset is composed of 13 SEP events, 16 Elevated Events, and 2256 Background events. Due to the small number of SEP samples, 3-fold cross validation was applied. To form the 3-folds, we used stratified sampling based on the ln Peak Intensity to ensure an equal distribution of events between the training, validation, and test sets. From the sorted list of events by ln Peak Intensity, we used buckets of 6 samples which were then split randomly with 3 samples going to the training set, 1 sample going to the validation set, and 2 samples going to the test set. The 3-fold datasets are summarized in Table 4.2.

In Table 4.2, the number of elevated training, validation, and test events and SEP training, validation, and test events are not the same across the 3 fold groups, but they do still add up to the correct total. This is because of how we split up the SEP and Elevated events. There are 29 SEP and Elevated events combined which means there are 4 groups of 3/1/2 (training/validation/test) with 5 events left over. The concept of 3-fold cross validation is every sample is in exactly one of the test sets. Therefore, the last 5 events are split 2/1/2, 2/1/2, and 3/1/1 between the first, second, and third fold respectively. Since we apply stratified sampling based on ln Peak Intensity across SEP and Elevated events combined, 1 of the 5 events is an SEP event and the rest are Elevated events. Based on which group, training/validation/test, the SEP randomly ends up, we generate different combinations of training, validation, and test sets across the 3-fold cross validation.

Each fold is composed of three sets: the training set, the validation set, and the test set. The training set is used as input into a machine learning model during training. The output of the model is compared to the actual target value using a loss function such as binary cross entropy. The loss is then back-propagated through the layers of

Table 4.2: Distribution of events in our 3-fold data sets.

Fold	Type	Training Set	Validation Set	Test Set	Total
1	Background	1240	340	676	2256
1	Elevated	8	3	5	16
1	SEP	6	2	5	13
2	Background	1240	340	676	2256
2	Elevated	7	3	6	16
2	SEP	7	2	4	13
3	Background	1240	340	676	2256
3	Elevated	8	3	5	16
3	SEP	7	2	4	13

the network using batch stochastic gradient descent using Equation 4.10.

$$\Delta w = -\eta \nabla L \tag{4.10}$$

where w are the weights of the last layer of the model, L is the loss calculated for a batch of inputs, and η is the learning rate. The learning rate is a hyperparameter. Selecting a large learning rate increases the rate at which the model converges, but, if too large, can prevent convergence. Selecting too small of a learning rate can result in a model getting stuck in local minimums, and it takes longer to converge. The validation set is used to evaluate the model's performance during initial training. During initial training, the model does not see the validation set. Every epoch, the model is trained using the training set, and the loss is calculated from the validation set. The training error will continue to decrease, but the validation error usually forms a V shape where it decreases to a minimum before increasing. Continuing to train the model after the minimum validation error can cause overfitting. Using the validation set, we find an optimal epoch number for actual training. During actual training, we combine the training and validation sets together into one training set and train up to the found epoch number.

All neural networks are run 5 times with different random initializations of the neural network weights for an optimal number of epochs. The optimal number of epochs was found from an initial run with a validation set. Metrics are calculated in each of the 5 runs. The final metrics supplied in the results are the average over the 5 runs. This approach helps justify the results as not caused by the lucky or unlucky initial random initialization of weights and to help avoid finding local minimum during training.

After a technique is run 5 times to the optimal epoch number, we tested threshold values from 0.1 to 0.9 in 0.1 increments to find the best performing threshold in the average F1 score performance over the 5 runs. The threshold value, t, is used to classify the model's output, p, in the range $[0, 1]$ to a classification SEP if $p \geq t$ or to a classification non-SEP for $p < t$.

Unless otherwise specified, all unmentioned parameters were the default values in the keras Tensorflow library.

4.2.2.1 Procedures for Training Regular Neural Network with Oversampling

This approach included varying an oversampling percentage from 0%, i.e. the original, imabalanced dataset, to 90% in 10% increments forming 10 different networks. To initialize each model, we used a random uniform initializer between -0.05 and 0.05. We used an Adam optimizer with a learning rate of 0.001 and Adam epsilon 1.0. Hidden layers used the LeakyReLU activation function with an alpha of 0.3. The optimal epoch number varied per oversampling rate. We summarize the epoch numbers in Table 4.3.

Table 4.3: Epoch counts for cRegNN, cRT, and cRT+AE techniques used per oversampling percent.

Oversampling Percent	cRegNN	cRT	cRT+AE
0	7683	N/A	N/A
10	2422	913	1029
20	2042	671	1262
30	1612	704	1828
40	1284	524	1070
50	1029	470	785
60	800	400	617
70	602	342	429
80	408	238	263
90	215	115	122

4.2.2.2 Procedures for Training cRT

Classifier retraining included two steps. The first step was completed by recalling the trained model from the Regular Neural Network with 0% oversampling. We took that model and froze and reused the layers up to the z-layer, added a new hidden layer, and then reinitialized and reused the output layer in the second step. Like the Regular Neural Network with Oversampling, this approach used an oversampled dataset in the second step which varied from 10% to 90% in 10% increments forming 9 different networks. The weights were initialized using a random uniform initializer between -0.05 and 0.05. We used an Adam optimizer with a learning rate of 0.001 and Adam epsilon 1.0. Hidden layers used the LeakyReLU activation function with an alpha of 0.3. The optimal epoch number in the second step varied per oversampling rate. We summarize the epoch numbers in Table 4.3.

4.2.2.3 Procedures for Training cRT+AE

Classifier retraining with autoencoder began with estimating a suitable α value. We trained the autoencoder and classifier branches of the cRT+AE method separately for

50000 epochs. They both used a random uniform weight initializer between -0.05 and 0.05 and a LeakyReLU activation function on the hidden layers with alpha as 0.3. The autoencoder branch used a Mean Squared Error loss function while the classifier branch used a Binary Cross Entropy loss function. We combined the errors using Equation 4.4 and found an alpha value. With the α value estimated, we trained the cRT+AE network in two steps. For the first step, we used a random uniform weight initializer between -0.05 and 0.05 and a LeakyReLU activation function on the hidden layers with alpha as 0.3. The first step used the original, imbalanced dataset. We found 20788 epochs to be optimal for the first step. The first step was fully trained only once (not 5 times), and it was used as the basis for the second step which was run 5 times per oversampled dataset. Like the Regular Network with Oversampling and cRT, we varied oversampling in the second step from 10% to 90% in 10% increments forming 9 networks. In the second step, we used another random uniform weight initializer between -0.05 and 0.05 to initialize the additional hidden layer and to reinitialize the classifier layer, and we used LeakyReLU activation functions on the hidden layers with alpha as 0.3. The output layer used a Binary Cross Entropy loss function. The optimal epoch number for the second step varied per oversampling rate. We summarize the epoch numbers in Table 4.3.

4.2.3 Results

In this section, we present the results for the first fold of the 3-fold dataset. The same training and validation sets are used across the respective techniques e.g. cRegNN with 10% oversampling is the same training and validation sets as cRT with 10% oversampling. The same test set was used for all approaches. Therefore, the metrics presented below are comparable across techniques.

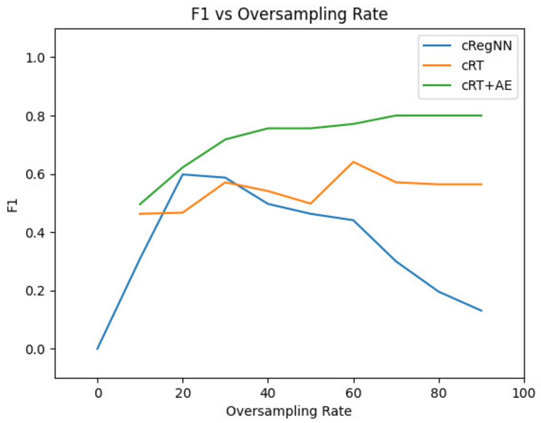


Figure 4.5: F1 score vs oversampling rate for classifier techniques.

4.2.3.1 Regular NN with Oversampling (cRegNN)

Table 4.4: Summarized metric results averaged over 5 runs for the cRegNN technique with the test set in 3-fold Dataset 1. Each row in a technique has an oversampled dataset by some percent indicated by the trailing number after the row technique name, e.g. cRegNN 10 is the cRegNN technique with 10% oversampling. The best value for each metric in cRegNN is underlined. The value of the metric is in bold if it is the best across all tested techniques.

Technique	Threshold	FP	FN	TP	TN	F1	HSS	TSS
cRegNN 0	0.5	**0.0**	5.0	0.0	**681.0**	0.000	0.000	0.000
cRegNN 10	0.2	7.2	2.2	2.8	673.8	0.309	0.305	0.550
cRegNN 20	0.4	5.8	0.4	4.6	675.2	0.598	0.594	0.911
cRegNN 30	0.4	6.8	**0.2**	**4.8**	674.2	0.587	0.583	**0.950**
cRegNN 40	0.4	10.2	**0.2**	**4.8**	670.8	0.497	0.491	0.945
cRegNN 50	0.5	8.8	1.2	3.8	672.2	0.463	0.457	0.747
cRegNN 60	0.5	11.6	1.2	3.8	669.4	0.441	0.435	0.743
cRegNN 70	0.4	24.6	0.6	4.4	656.4	0.300	0.291	0.844
cRegNN 80	0.4	41.8	0.4	4.6	639.2	0.196	0.185	0.859
cRegNN 90	0.5	28.6	2.4	2.6	652.4	0.131	0.125	0.478

Table 4.4 shows the classification metrics averaged over 5 runs for the cRegNN technique with the test set in 3-fold Dataset 1. Each row used a training set with different amounts of oversampling. It is easy to have 0.0 FP and all 681.0 TN as seen

in cRegNN 0 by under classifying all events. This also results in having all 5.0 FN events as well. cRegNN 30 has the best TSS score due to the almost perfect count of TP events. The best F1 score is cRegNN 20 which has just 0.2 more FN and 1.0 less FP. The fractional difference in FN count is because the results are averaged over 5 runs: 2 of the 5 runs had a FN in cRegNN 20 vs of 1 of the 5 runs in cRegNN 30. Figure 4.5 helps illustrate the effect of oversampling on the average F1 in cRegNN. This technique has a peak at 20% oversampling with rapid decline at 10% and 20% oversampling but a more gradual decline with high oversampling rates. Since 20% and 30% oversampling are about the same amount, it suggests that there might be an even higher peak with an oversampling between 20-30%. We leave this for future study.

4.2.3.2 Classifier Re-Training (cRT)

Table 4.5: Summarized metric results averaged over 5 runs for the cRT technique with the test set in 3-fold Dataset 1. Formatting is the same as in Table 4.4.

Technique	Threshold	FP	FN	TP	TN	F1	HSS	TSS
cRT 10	0.2	11.4	**0.2**	**4.8**	669.6	0.463	0.457	0.943
cRT 20	0.3	10.4	0.4	4.6	670.6	0.467	0.461	0.905
cRT 30	0.5	5.0	1.0	4.0	676.0	0.571	0.567	0.793
cRT 40	0.5	5.8	1.0	4.0	675.2	0.541	0.536	0.791
cRT 50	0.5	7.2	1.0	4.0	673.8	0.498	0.492	0.789
cRT 60	0.6	3.2	1.2	3.8	677.8	0.641	0.638	0.755
cRT 70	0.6	5.0	1.0	4.0	676.0	0.571	0.567	0.793
cRT 80	0.6	5.2	1.0	4.0	675.8	0.564	0.560	0.792
cRT 90	0.6	5.2	1.0	4.0	675.8	0.564	0.560	0.792

Table 4.5 shows the classification metrics averaged over 5 runs for the cRT technique with the test set in 3-fold Dataset 1. Each row used a training set with different amounts of oversampling. The best F1 score in cRT is an improvement at 0.641 over the best F1 score in cRegNN at 0.598, however the TP count fell by 0.8. The increase in F1 score is due to the reduced number of FP now 3.2 instead of 5.8. The tradeoff of more

FP corresponding to more TP is well known since usually to include more TP events you must decrease the threshold which includes more FP events. We can see that the threshold for cRegNN 30 was lower at 0.4 vs 0.6 in cRT 60, and cRegNN 30 had more TP and FP events. Figure 4.5 helps illustrate the effect of oversampling on the average F1 metric in cRT. This technique has a much different distribution shape for the average F1 score. Figure 4.5 for cRT has two peaks with the peak at 30% oversampling below the overall peak at 60% oversampling. The 30% oversampling improved performance is consistent with the cRegNN technique, but the higher peak at 60% oversampling is more unique. It could be argued that the cRegNN has a secondary peak around 60% since the F1 score does not decline as significantly for cRegNN from 50-60 as 40-50.

4.2.3.3 Classifier Re-Training with Autoencoder (cRT+AE)

Table 4.6: Summarized metric results averaged over 5 runs for the cRT+AE technique with the test set in 3-fold Dataset 1. Formatting is the same as in Table 4.4.

Technique	Threshold	FP	FN	TP	TN	F1	HSS	TSS
cRT+AE 10	0.1	1.4	2.4	2.6	679.6	0.496	0.494	0.518
cRT+AE 20	0.2	5.4	0.4	4.6	675.6	0.622	0.619	0.912
cRT+AE 30	0.6	2.2	1.0	4.0	678.8	0.718	0.715	0.797
cRT+AE 40	0.6	1.6	1.0	4.0	679.4	0.756	0.754	0.798
cRT+AE 50	0.6	1.6	1.0	4.0	679.4	0.756	0.754	0.798
cRT+AE 60	0.6	1.4	1.0	4.0	679.6	0.771	0.769	0.798
cRT+AE 70	0.6	1.0	1.0	4.0	680.0	**0.800**	**0.799**	0.799
cRT+AE 80	0.6	1.0	1.0	4.0	680.0	**0.800**	**0.799**	0.799
cRT+AE 90	0.6	1.0	1.0	4.0	680.0	**0.800**	**0.799**	0.799

Table 4.6 shows the classification metrics averaged over 5 runs for the cRT+AE technique with the test set in 3-fold Dataset 1. Each row used a training set with different amounts of oversampling. We achieve the highest F1 score in cRT+AE 70, 80, and 90 at 0.800 vs the prior highest F1 in cRT 60 at 0.641. The threshold value remains the same at 0.6, but we have dropped to 1.0 FP while increasing to 1.0 FN. The

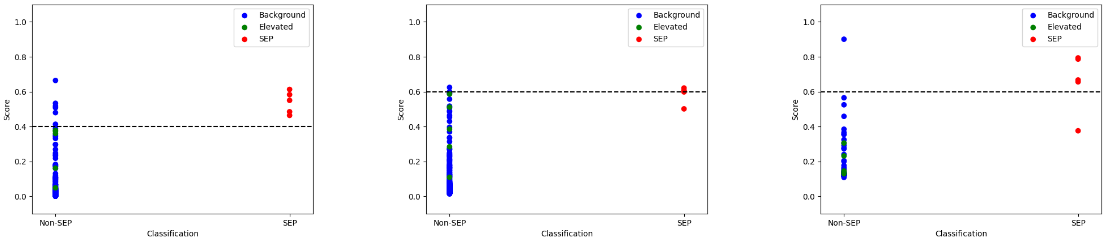
whole number of FP and FN events indicates that the results were the same across all 5 runs. Figure 4.5 helps illustrate the effect of oversampling on the average F1 metric in cRT+AE. In contrast to the previous two techniques, the cRT+AE approach shows a positive correlation between F1 score and the oversampling rate. This shape was expected from all techniques since the goal of increasing the oversampling rate is to make the minority SEP and Elevated events more and more emphasized thereby increasing performance, but it only occurred in this technique.

4.2.3.4 Comparing the Main Approaches

Table 4.7: Summarized metric results averaged over 5 runs for the best F1 rows of the cRegNN, cRT, and cRT+AE techniques with the test set in 3-fold Dataset 1. Row formatting is the same as in Table 4.4. The best metric value is in bold.

Technique	Threshold	FP	FN	TP	TN	F1	HSS	TSS
cRegNN 20	0.4	5.8	**0.4**	**4.6**	675.2	0.598	0.594	**0.911**
cRT 60	0.6	3.2	1.2	3.8	677.8	0.641	0.638	0.755
cRT+AE 70	0.6	**1.0**	1.0	4.0	**680.0**	**0.800**	**0.799**	0.799

In analyzing the highest F1 scores across all techniques, we see an improvement from cRegNN to cRT to cRT+AE with a maximum of 0.800 F1 and 0.799 HSS in cRT+AE with 70, 80, and 90 oversampling. The average metrics for cRT+AE 70/80/90 were identical, so Table 4.7 shows only the cRT+AE 70 row. However, the highest TSS of the best F1 scores is cRegNN 20. There, the TP count indicates almost a perfect SEP prediction across all 5 runs at 4.6, however this comes at the cost of about 6.0 FP events. In comparison, the cRT+AE 70 row has only 1.0 FP, but it also has a solid 1.0 FN. This means we consistently missed an SEP event classification across all 5 runs. Due to the severity of the protons, it could be more advantageous to use the cRegNN 20 model which predicts almost perfectly the SEP events, however this



(a) cRegNN with 20% oversampling (b) cRT with 60% oversampling (c) cRT+AE with 70% oversampling

Figure 4.6: Predicted classification score vs Actual classification for cRegNN, cRT, and cRT+AE. Events above dotted threshold line are predicted SEP events. FPs and FNs are in the upper left and lower right respectively.

would depend on the acceptance of FPs. We do not have a threshold for how many FPs are allowed, so we instead aim to optimize the F1 score. Therefore, we would say that there is an improvement from cRegNN to cRT to cRT+AE culminating with the cRT+AE 70/80/90 models performing the best.

4.2.4 Analysis

4.2.4.1 Regular NN with Oversampling (cRegNN)

The highest average F1 score for the cRegNN technique was 0.598 using 20% oversampling. The average FN count was very close to but not quite 0.0. In 2 of the 5 runs, 1 of the SEP events was predicted as a FN. We illustrate the median run in Figure 4.6a. Here, we can see there are no FN events and 6 FP events.

Table 4.8 lists a set of important features for the FP events in the cRegNN technique with 20% oversampling. These features include the same features presented in the DONKI catalog answering how fast, what direction, and how wide a CME. They also include the acceleration feature which had some strong patterns across the three techniques possibly contributing to high classifier score. To illustrate the distribution of each feature, we include feature plots in Figure 4.7. Each feature plot shows the

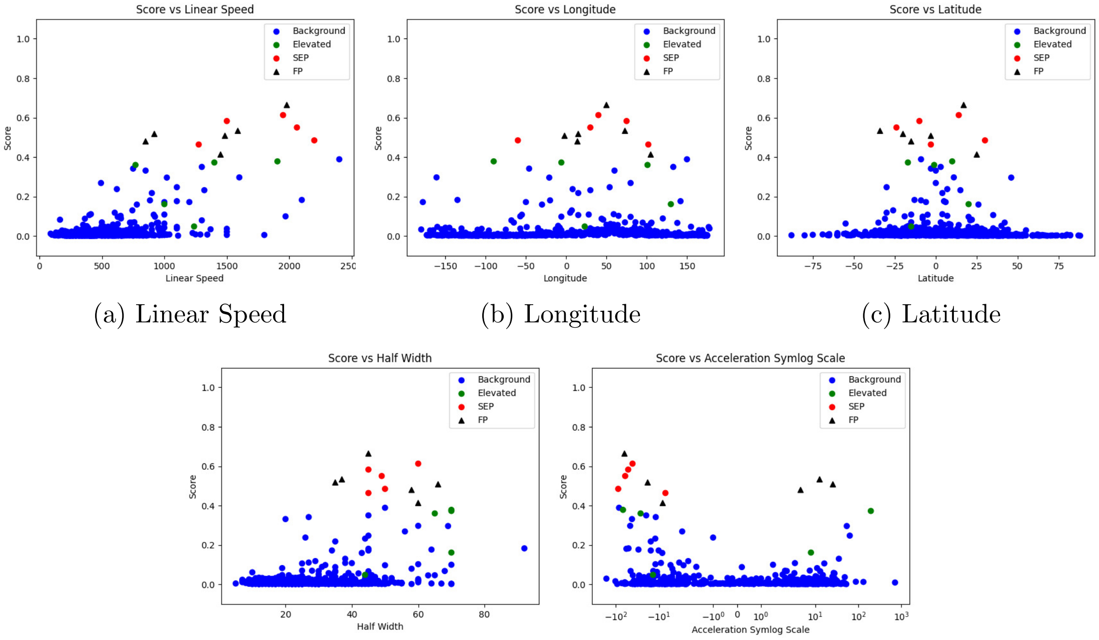
Table 4.8: The important features for the False Positive (FP) and False Negative (FN) events for cRegNN 20 from Figure 4.6a. The features are given across the top in abbreviated format: DONKI Date (DD), Latitude (Lat), Longitude (Lon), Linear Speed (Spd), Half Width (Hw), Acceleration (Acc), Actual 100 MeV Peak Intensity ln (Peak ln), Classifier Score (Score), and the Classifier Prediction (FP/FN).

DD	Lat	Lon	Spd	Hw	Acc	Peak ln	Score	FP/FN
3/7/2011 20:12	17	50	1980	45	-63.1	-2.000	0.664	FP
1/1/2016 23:12	-34	73	1588	37	12.7	-2.000	0.533	FP
2/15/2011 2:25	-20	15	920	35	-18.3	-2.000	0.518	FP
3/15/2013 6:54	-3	-2	1485	66	25.8	-2.000	0.508	FP
12/28/2015 12:39	-15	14	850	58	4.6	-2.000	0.480	FP
3/18/2012 0:39	25	105	1450	60	-8.2	-2.000	0.413	FP

feature values on the x-axis and the classifier score on the y-axis. While the rest of the features have raw values on the x-axis, the acceleration feature uses the symlog scale to better emphasize the distribution of the feature otherwise dominated by outlier acceleration value. Many of the acceleration values are negative meaning they are actually decelerating upon observation. The classifier scores for the FP events vary from just above the threshold value to the highest classifier score among all the other events. By studying these events, we can better understand what feature or features may contribute to their erroneously high scores.

From analyzing the Linear Speed feature in Figure 4.7a, we see a weak positive correlation between speed and classification score. The TP events all have relatively high speed and high classifier score, but there are several FP background events with speeds below 1000 yet relatively higher classifier scores compared to other background events. A higher DONKI speed may not be the sole cause of a higher classifier score.

The longitude feature in Figure 4.7b shows a peak around 57 degrees. We expected a peak around 57 degrees because it is approximately the longitudinal direction of the inter-planetary magnetic field (IMF) line with the Earth, although the value varies. Most of the FP and TP events have longitude values close to 57. As we move away



(a) Linear Speed

(b) Longitude

(c) Latitude

(d) Half Width

(e) Acceleration with symlog scale

Figure 4.7: Predicted classifier score vs 5 features for cRegNN with 20% oversampling.

from 57 degrees in either direction, the score in the FP and TP events drops. All but 1 FP event is within the range of the other SEP longitude values which may make the FP events difficult to distinguish from the TP events. Therefore, the longitude feature may be contributing to the higher scores for the FP events.

We also see a peak in the latitude feature in Figure 4.7c around 0. We expected a peak around 0 degrees because it is approximately the latitudinal direction of the IMF line with the Earth. The FP events are mixed with some very favorable latitude values of 0 while others deviate by as much as -34 degrees. However, the highest predicted FP event has a less favorable latitude value along with the highest score among the TP events. These events likely have other features that are contributing to their higher scores. One of the FP events even has the same latitude value, -3, as a TP event but a higher score. Clearly, the latitude feature is not the only contributor to the FP events elevated scores.

Another weaker peak can be seen in the half width feature in Figure 4.7d around 45. We classify as a weaker peak because of some of the other FP events with higher scores but less half width. The FP events with the higher prediction scores have half widths close to 45. The half width values may be contributing to their increased score. Meanwhile, the FP with the lowest score has the same half width as the TP with the highest score. For this event, there must be other features contributing to either its lower score or the TP event's higher score. Since several other FP events with similar half widths have lower scores, it is likely that the TP event has other features contributing to its much higher score.

The average Acceleration feature value of the TP events is about -60 before symlog scale shown in Figure 4.7e as a peak close to -10^2 in symlog scale. The FP events are mixed with some that have acceleration values close to that peak and others farther away. There are two FP events with acceleration close to -60 with values -63.1 and -42.3 with another FP event a farther away at -8.2. We can see that there is a linear drop in score for the three FP events as we move away from the peak at 10^{-2}. Conversely, three of the FP events have positive values and appear to be forming a second smaller peak at close to symlog 10^2. The similarity in acceleration values between the FP events and the TP events could be contributing to their higher classifier score. The peak on the positive side is likely caused by other features since most of the TN events with similar acceleration values have very low scores.

4.2.4.2 Classifier Re-Training (cRT)

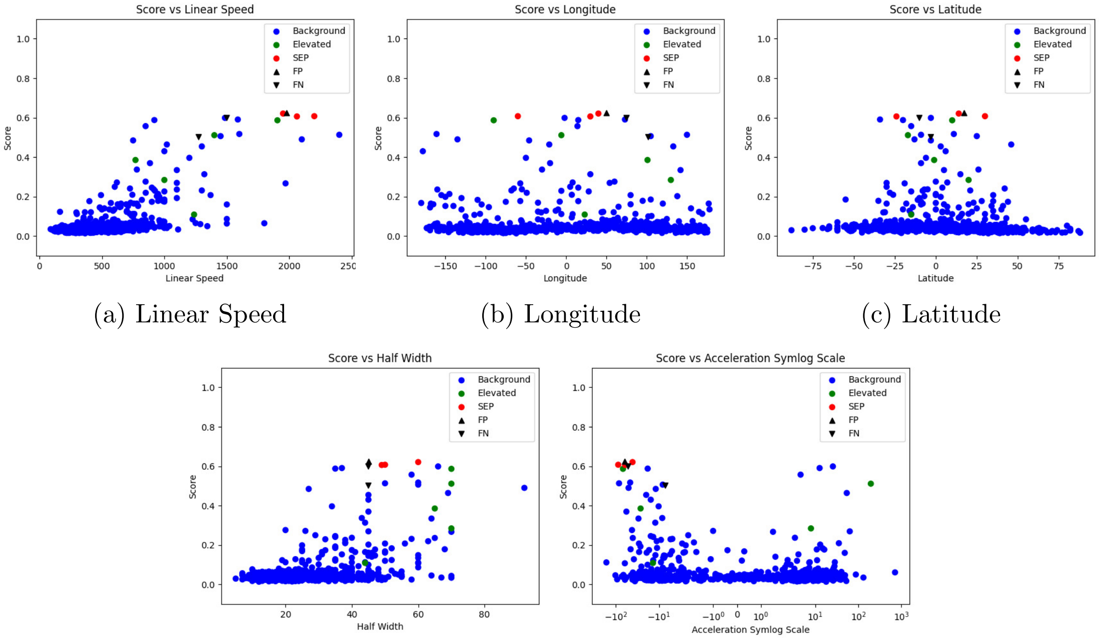
The highest average F1 score for the cRT technique was 0.641 using 60% oversampling. This is a higher average F1 score than the highest average F1 in the cRegNN technique due primarily to the fewer average FP events, 3.2 down from 5.8, despite slightly more average FN events, 1.2 up from 0.4. The increase in FN events is due to 1 run when a second SEP event fell just barely below the threshold value of 0.6. This run also had only 1 FP event which is why its F1 score was higher than the run with the worst F1 score. We illustrate the median run in Figure 4.6b. We use the same red for SEP, green for Elevated, blue for Background, and dotted threshold indicator as the cRegNN analysis.

Table 4.9: The important features for the False Positive (FP) and False Negative (FN) events for cRT 60 from Figure 4.6b. Acronyms are the same as in Table 4.8.

DD	Lat	Lon	Spd	Hw	Acc	Peak ln	Score	FP/FN
3/7/2011 20:12	17	50	1980	45	-63.1	-2.000	0.625	FP
5/17/2012 1:48	-10	75	1500	45	-51.8	2.708	0.599	FN
1/6/2014 8:09	-3	102	1275	45	-7.1	1.061	0.501	FN

Table 4.9 lists a set of important features for the FP and FN events in the cRT technique with 60% oversampling. We tabulate the same features as the cRegNN for comparison. We can see that one of the FN events is just barely a FN with a prediction of 0.599 and a threshold of 0.6. We could remove the FN by shifting the threshold down just below the FN, but we would actually add another FP event that is not listed in that table which has just slightly more prediction than the FN event. A more thorough search of an optimal threshold value could improve the average performance, but we leave that for future study.

We saw the FP event in the cRegNN, but the FN events are new. We include feature plots in Figure 4.8 to show the score vs feature distribution where the feature values are on the x-axis and the classifier scores on the y-axis. In terms of classifier score, the FP event on 3/7/2011 again has the highest score amongst all events including the TP events. The FN event on 5/17/2012 is very close to being a TP event, but the other FN event on 1/6/2014 has a much lower score. The cRT technique does seem to be an improvement over the cRegNN technique with reducing the FP events but at the cost



(a) Linear Speed

(b) Longitude

(c) Latitude

(d) Half Width

(e) Acceleration with symlog scale

Figure 4.8: Predicted classifier score vs 5 features for cRT with 60% oversampling.

of 2 FN events.

In terms of Linear Speed in Figure 4.8a, we see a stronger positive correlation between Linear Speed and classifier score. There's a sharp decrease in score just below 1000 Linear speed, and the scores of events with higher speeds mostly have high scores. The FN event with a lower speed also has a lower score than the FN event with higher speed. The positive correlation suggests that the higher speeds could be contributing to higher classifier scores.

The longitude feature does not have as distinctive of a peak in Figure 4.8b as it did in the cRegNN analysis. Around the 50 degree area, there is a gentle sloping curve to the right and left with a decreasing score. However, some of the TN events have higher scores than the FN events despite their varied longitude values, especially events such as the elevated event around -100. These higher predictions in the TN events may be due to the outlier TP event with longitude around -60. It is still likely that the longitude values are contributing to the scores though the pattern seems less clear than in the previous analysis.

Similar to longitude, the latitude peak has spread out more in Figure 4.8c than we saw previously in the cRegNN analysis. The effect does not seem as extreme as the longitude comparison as latitude values have higher predictions between about -25 and 25 vs the entire range of the longitude values. The FP elevated event has a somewhat favorable latitude value of 10 which may have contributed to its incorrect classification.

A similar trend to longitude and latitude can be seen in the spread out peak of the half width feature in Figure 4.8d. The peak looks spread further because of some of the higher predicted TN events to the right and left sides. The FP event and both FN events share the same half width. Clearly, other features are contributing to their varied predictions.

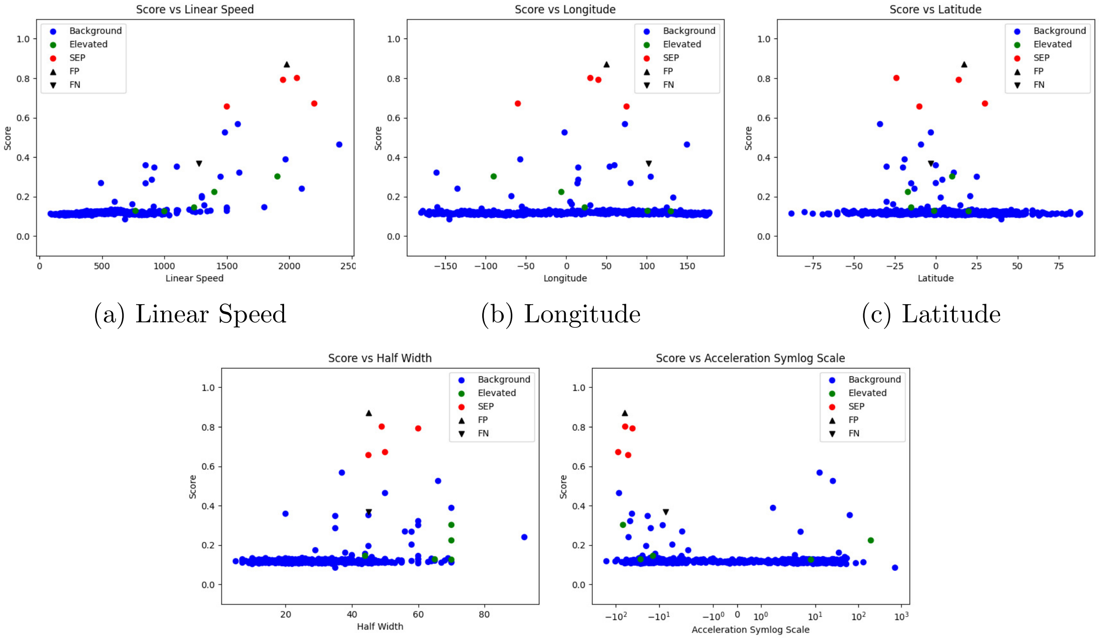
The acceleration feature still appears to have a peak around -60 in Figure 4.8e, however the peak prediction values are lower than in the cRegNN analysis. There are also higher scores close to the positive peak, but they are now all TN events. The FP elevated event has an acceleration of -68.3 which is very favorable. This favorable acceleration may be increasing its classifier score. The FN event that is very close to being a TP event also has very favorable acceleration. The other FN event has a higher acceleration, and this is likely contributing to its lower score.

4.2.4.3 Classifier Re-Training with Autoencoder (cRT+AE)

The highest F1 score across all techniques was 0.800 with the cRT+AE technique when using 70%/80%/90% oversampling. We will focus on analyzing the 70% oversampling due to their similarity. We have seen the FP throughout all three approaches and the FN since the rRT. Figure 4.6c is a plot of the median run although they all had identical F1 scores showing the classifier score vs the actual classification.

Table 4.10: The important features for the False Positive (FP) and False Negative (FN) events for cRT+AE 70 from Figure 4.6c. Acronyms are the same as in Table 4.8.

DD	Lat	Lon	Spd	Hw	Acc	Peak ln	Score	FP/FN
3/7/2011 20:12	17	50	1980	45	-63.1	-2.000	0.872	FP
1/6/2014 8:09	-3	102	1275	45	-7.1	1.061	0.370	FN



(a) Linear Speed

(b) Longitude

(c) Latitude

(d) Half Width

(e) Acceleration with symlog scale

Figure 4.9: Predicted classifier score vs 5 features for cRT+AE with 70% oversampling.

Table 4.10 lists features for the FP and FN event in cRT+AE similar to the previous two approaches. We added feature plots in Figure 4.9 to illustrate the classifier score vs feature distributions. The classifier score for the FP event is still the highest compared to all the other events. The FN event has a lower classifier score than in the cRT technique.

The Linear Speed in Figure 4.9a has the strongest positive correlation yet. There is a distinct increase in classifier score as the speed increases. The speed of the FN event is only 1250 which is not much larger than 1000 speed, so it could be that its lower speed compared to the TP and FP events is contributing to its lower score.

When analyzing the longitude feature in Figure 4.9b, we have recaptured the peak at 57 degrees though not as distinctly as the original cRegNN technique. The FP longitude is more favorable, and the FN longitude is less favorable to a peak around 57. The fact that these events have persisted across the top techniques makes it more likely that longitude is contributing more significantly to their respective scores.

The shape of the latitude feature in Figure 4.9c does not have as distinctive of a peak at 0 degrees similar to cRT. The FN event has a fairly favorable latitude score for a peak around 0 degrees. Therefore, the latitude feature may be helping increase the FN score above most of the other TN events, but there are other features pulling its score down in comparison to some of the higher predicted TN events with similar latitude values. In contrast, the FP event has a much higher score but less favorable latitude. Other features are likely contributing to its increased score.

Similar to the longitude feature, we have recaptured the peak around 45 in the half width feature in analyzing Figure 4.9d. The FP, one of the TP, and the FN have a half width of 45 with two of the other TP events close by with values of 49 and 50. When considering the FP and TP events, the half width feature appears to be contributing to their increased scores. However, the FN event has a lower classifier score than several other TN events with varying half width values. Therefore, there are other features reducing the FN score even with its favorable half width value.

We still seem to have a peak in the acceleration feature in Figure 4.9e. The peak seems more distinctive than cRegNN due to the fewer FP events and the more precise classifier scores for the TP events. In the cRegNN, the TP events had a much higher spread of classifier scores than in this approach. The acceleration feature appears to contribute to the increased classifier score of the FP event with the favorable -63.1 acceleration. It also appears to contribute to the decreased classifier score of the FN event with the less favorable -7.1 acceleration. There still seems to be a peak on the positive side as well, but it has shifted closer to 10^1 in symlog scale. Despite high classifier scores, they remain correctly classified.

4.2.4.3.1 Feature Importance

To further analyze the cRT+AE 70 model, we calculate the feature importance values using Local Interpretable Model-agnostic Explanations (LIME) developed by Ribeiro et al. [14] both to help explain the overall model and to help explain the remaining FP and FN events present in the model. Please refer to Section 3.5 for the details on how feature importance values are calculated using this method with the following additions.

For a given instance i, we calculate the local feature importance values, W_{ij}, using LIME. LIME utilizes a random seed to generate samples during the process of calculating the local feature importance values. To compensate, we generate local feature importance values using LIME with 5 different seeds, so instance i has 5 values W_{ijk} for $k \in [1, 2, 3, 4, 5]$. We then combine them together to calculate the average W_{ij}:

$$W_{ij} = \frac{1}{R} \sum_{k=1}^{R} W_{ijk} \tag{4.11}$$

where $R = 5$.

To calculate the global model feature importance values, we use all instances in the entire dataset. The calculation of the global importance values is based on the local importance calculations, so more samples should provide a more accurate estimate of global importance. To calculate the global importance for feature j denoted as I_j:

$$I_j = \sqrt{\sum_{i=1}^{N} |W_{ij}|} \tag{4.12}$$

where N is the number of samples. The importance value are then normalized:

$$\hat{I}_j = \frac{I_j}{\sum_{i=1}^{M} I_i} \tag{4.13}$$

where M is the number of features.

We take a step further than LIME for analyzing the specific FP and FN instances. In analyzing the W_{ij} for the FP and FN event and the $\hat{I}_j$ values, we found that the W_{ij} values were not completely representative of how important a feature was to the resulting classification. The W_{ij} value is a weight that can be interpreted as the slope of a linear fit around the instance, i. For a change in any feature value by the same amount, say 1, the W_{ij} quantifies the impact on the prediction positive/negative and larger/smaller. Therefore, for a single instance i, a W_{ij} value that is higher means that the feature j is more important than another whose value is lower since it assumes the features both change by the same amount. However, we have some extreme cases such as the feature CMEs over 1000 km/s in the past 9 hours which has a value of 1 for a handful of background events, 1 elevated event, and 0 SEP events in the entire dataset and the rest have a value of 0. This results in a high W_{ij} value for the CMEs over 1000 km/s in the past 9 hours feature which we will see later in Table 4.11 despite the feature having no actual impact on the prediction when the feature value for an instance is 0. To compensate, we also calculate the contribution of a feature C_{ij}.

$$C_{ij} = W_{ij} \cdot X_{ij} \tag{4.14}$$

where X_{ij} is the normalized value we feed into the model as from Chapter 3 for feature j and for instance i.

Simply comparing feature rankings in bins may disregard how the features themselves are related. This gives the motivation for introducing an additional technique originally introduced by Tarsoly [18] to group the feature importance methods summing over related features before ranking. They defined five primary groups which we also use: Speed, Location, Size, CME History, and Other. The Speed group contains the following features:

- Linear Speed
- Diffusive Shock
- 2nd order speed final
- 2nd order speed at 20 solar radii
- V Log V

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

- Latitude
- Longitude
- Richardson's equation
- CPA (weighted by 0.5)
- MPA

CPA is weighted by 0.5 because it encodes both location and event size information. The Size group related to features conveying the width of the CME event:

- CPA (weighted by 0.5)
- Halo

- Half Width

The CME History group is made up of features that are measured using previous CMEs.

- Number of CMEs in the Past Month
- Number of CMEs in the Past 9 Hours
- Number of CMEs with Speed over 1000 km/s in the past 9 Hours
- Maximum Speed for a CME in the Past Day

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

- Acceleration
- Type II visualization area
- Daily Sunspot count

We calculate group feature importance for the FP and FN events specifically and for the overall model. For an individual event, i, we calculate a quantity W_{iJ} which represents the feature importance for a group of features, J, which is a subset of feature indexes. We calculate W_{iJ} using the W_{ij} values from Equation 4.11.

$$W_{iJ} = \sum_{j \in J} |W_{ij}| \tag{4.15}$$

In the special case of the feature CPA, it is weighted by 0.5 in Equation 4.15. We then normalize these values:

$$\hat{W}_{iJ} = \frac{W_{iJ}}{\sum_{k=1}^{G} W_{ik}} \tag{4.16}$$

where G is the number of feature groups.

For the overall dataset, we calculate a quantity $\hat{I}_J$ which represents the feature importance for a group of features, J, which is a subset of feature indexes. We calculate $\hat{I}_J$ using the global model importance values, $\hat{I}_j$, obtained from Equation 4.13.

$$\hat{I}_J = \sum_{j \in J} \hat{I}_j \tag{4.17}$$

In the special case of the feature CPA, it is weighted by 0.5 in Equation 4.17.

Table 4.11: Overall Feature importance values ($\hat{I}_j$) for cRT+AE 70.

Feature (j)	Importance ($\hat{I}_j$)
1. V Log V	0.117
2. Diffusive shock	0.111
3. Linear Speed	0.086
4. Richardson's equation	0.084
5. 2nd order speed final	0.071
6. 2nd order speed at 20 solar radii	0.067
7. Type II Visualization Area	0.056
8. CMEs over 1000 km/s past 9 hrs	0.052
9. Max speed past day	0.049
10. Halo	0.047
11. CMEs in past month	0.043
12. Longitude	0.040
13. MPA	0.037
14. Daily Sunspot Count	0.029
15. Half Width	0.029
16. Latitude	0.028
17. Acceleration	0.026
18. CPA	0.014
19. CMEs in past 9 hours	0.014

Table 4.11 lists the $\hat{I}_j$ values calculated over the entire dataset. Several members of the speed group are highly important including V Log V, Diffusive Shock, and Linear Speed in the top 3 and the other two speed members are not much farther below at ranks 5 and 6. By the time we get to rank 7, the normalized importance value has already dropped by half. Richardson's equation, from the Location group

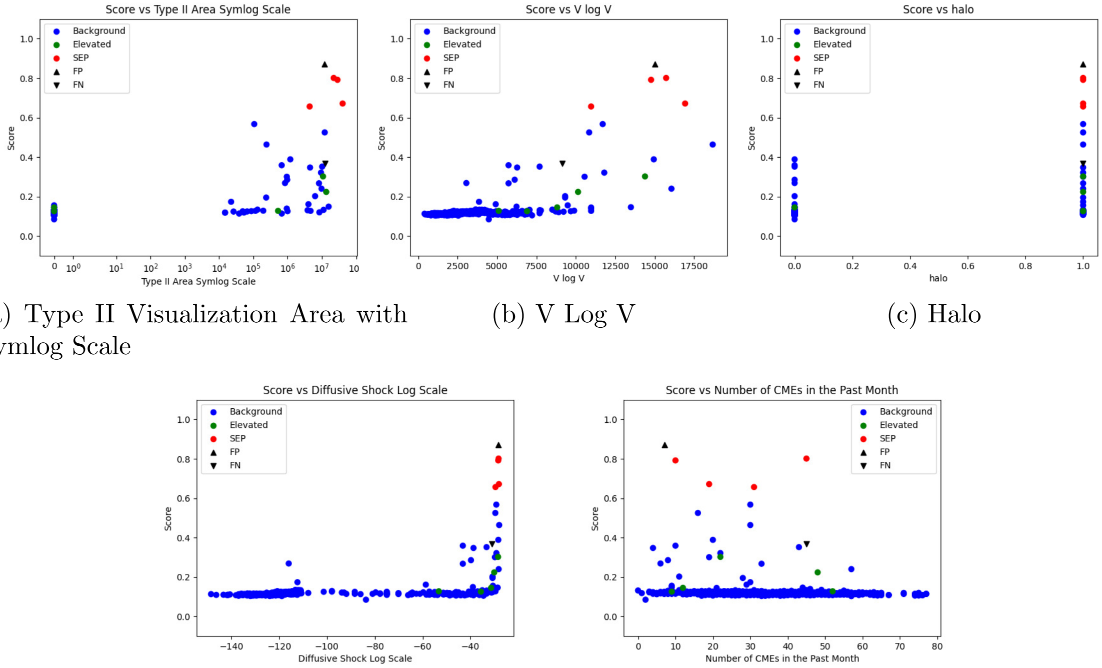
since we use only the connection angle part of his equation in the feature, is rank 4 with relatively high importance. The high importance of speed and location supports Richardson's entire equation [15] which models peak intensity based on the speed and location features of a CME.

Table 4.12: Feature importance values from LIME for FP and FN in Table 4.10 with cRT+AE 70. Includes feature importance values, W_{ij}, and contribution values, C_{ij}, with their respective rankings from largest to smallest feature importance value, $R_{W_{ij}}$ and $R_{C_{ij}}$. The C_{ij} shown are scaled by 1000. The top 3 are made bold, italic, and underlined, respectively (positive for FP, negative for FN) for the W_{ij} and C_{ij} values.

Feature (j)	FP				FN			
(ordered by $\hat{I}_j$ as in Table 4.11)	$R_{W_{ij}}$	W_{ij}	$R_{C_{ij}}$	C_{ij}	$R_{W_{ij}}$	W_{ij}	$R_{C_{ij}}$	C_{ij}
V Log V	**1**	**0.033**	*2*	*22.296*	1	0.012	3	4.788
Diffusive shock	13	0.003	12	2.896	**19**	**-0.006**	**19**	**-5.892**
Linear Speed	*2*	*0.028*	4	19.547	3	0.008	5	3.436
Richardson's equation	7	0.015	6	14.067	6	0.006	4	3.908
2nd order speed final	$\underline{3}$	$\underline{0.026}$	5	15.863	5	0.006	6	2.774
2nd order speed at 20 solar radii	6	0.019	7	10.422	13	-0.000	14	-0.159
Type II Visualization Area	4	0.026	**1**	**23.491**	4	0.007	2	6.343
CMEs over 1000 km/s past 9 hrs	16	0.001	17	0.000	9	0.002	12	0.000
Max speed past day	14	0.002	15	0.550	*18*	*-0.002*	16	-0.260
Halo	5	0.022	$\underline{3}$	$\underline{21.609}$	2	0.010	1	10.077
CMEs in past month	18	-0.007	18	-0.674	14	-0.001	$\underline{17}$	$\underline{-0.319}$
Longitude	17	0.000	16	0.080	12	-0.000	13	-0.105
MPA	9	0.007	9	6.496	8	0.002	9	1.484
Daily Sunspot Count	11	0.006	10	4.031	7	0.002	7	2.021
Half Width	10	0.007	11	3.358	11	0.001	10	0.409
Latitude	15	0.002	14	1.118	$\underline{17}$	$\underline{-0.001}$	*18*	*-0.358*
Acceleration	19	-0.013	19	-2.391	16	-0.001	15	-0.170
CPA	8	0.008	8	7.605	10	0.002	8	1.640
CMEs in past 9 hours	12	0.006	13	1.387	15	-0.001	11	-0.000

Table 4.12 focuses on the FP and FN instances from Table 4.10. Here, the features are listed in the same order as the overall feature importance for comparison, but the ranks are calculated on the W_{ij} and C_{ij} values in each event. We analyze the top positive feature contributors to the FP and the top negative feature contributors to the FN to gain further insight into their respective incorrect classifications. We then analyze the features in more detail referring to Figure 4.10.

In first analyzing the FP event, we see it shares the same top W_{ij} rank as the overall feature importance and the second, Linear Speed, and third, 2nd order speed final, rankings are not much further down. This is not too surprising since this event



(a) Type II Visualization Area with Symlog Scale

(b) V Log V

(c) Halo

(d) Diffusive Shock with Log Scale

(e) Number of CMEs in the Past Month

Figure 4.10: Predicted classifier score vs 5 features for cRT+AE with 70% oversampling.

has one of the higher Linear speed values. When examining the contributions, the highest contributor is shown to be Type II Visualization Area followed closely by V Log V and Halo.

In the next analysis of the FN event, the top negative contributors are Diffusive Shock, Latitude, and CMEs in the past month. The Diffusive Shock feature was one of the features we took the log of before normalizing as discussed in Chapter 3 due to the magnitudes of difference between the highest and lowest feature values. After the log scale, the value is very close to the other SEP events as we see in Figure 4.10d which makes its high negative contribution less expected. The latitude feature we have analyzed previously. The FN event has a latitude of 0, and we had expected a peak around 0. In reexamining Figure 4.9c, the background events do seem to suggest a negative trend as latitude increases while the FP events seem to vary around a peak of 0. The behavior of the background events may be determining this negative local importance value. When considered globally, latitude has a normalized feature importance of 0.028 about 20% of the top importance value. The discrepancy between the prediction behaviors between the background events negative trend and variance in the SEP events may be why the overall feature importance is so low for latitude. The CMEs in the past month is a new feature we did not previously analyze and has a global importance just below the middle ranking.

We illustrate the feature plots for the top positive FP contributors and top negative FN contributors features that we have not previously analyzed in Figure 4.10. There does seem to be a positive correlation in Type II Visualization Area shown in Figure 4.10a as the Type II Area increases generally so does the prediction. In both the FP and FN, it is considered a positive W_{ij} contributing to classification as a SEP. Its higher positive contribution value in both events seems due to the very high normalized feature value after the log scaling was applied despite a lower W_{ij} than other features. Analysis of the V Log V plot in Figure 4.10b should be very similar to the analysis of the Linear Speed feature, and we can see the same positive trend we saw in the speed feature. The higher positive contribution in the FP event seems due to its very high speed. The halo feature is categorical, either true or false. As we can see in Figure 4.10c, most of events have a halo value of 1 including the FP, FN, and the rest of the SEP events. In addition, the predictions on the events without a halo are much lower forming a positive trend although if the feature value is 0 then it has no contribution to their prediction value.

Figure 4.10d shows the Diffusive Shock feature with a log scale. The FN's feature value seems very close to the other SEP and the FP feature values. In analyzing the background events around the FN, it could be argued there is a small negative trend as quantified by the small negative W_{ij} which when combined with such a high normalized feature value results in a high negative contribution. Diffusive Shock has a very high overall feature importance likely due to the compactness of the SEP feature values after the log was taken. The feature number of CMEs in the past month has a much lower rank. Figure 4.10e illustrates the negative trend quantified in the W_{ij} values for the FP and FN. An overall negative trend seems less evident especially considering the TP event with a very similar feature value as the FN event, but, local to the FN event and TP event, there does seem to be a negative trend for this feature.

Table 4.13: Feature Group Importance for cRT+AE 70 for the overall dataset ($\hat{I}_J$), the FP event ($\hat{W}_{iJ}$), and the FN event ($\hat{W}_{iJ}$) and their respective ranks ($R_{\hat{I}_J}$, $R_{\hat{W}_{iJ}}$ for FP, $R_{\hat{W}_{iJ}}$ for FN).

	Overall		FP		FN	
Group (J)	$R_{\hat{I}_J}$	$\hat{I}_J$	$R_{\hat{W}_{iJ}}$	$\hat{W}_{iJ}$	$R_{\hat{W}_{iJ}}$	$\hat{W}_{iJ}$
Speed	1	0.452	1	0.474	1	0.467
Location	2	0.196	4	0.125	4	0.142
CME History	3	0.158	5	0.067	5	0.076
Other	4	0.111	2	0.192	3	0.144
Size	5	0.083	3	0.142	2	0.170

Table 4.13 shows the group importance values for the overall dataset, the FP, and the FN. In all three, the Speed group seems to dominate the group importance values. From there, the FP and FN both have a second and third ranked group with similar values but different groups. The very high Type II visualization area seems to drive the Other group importance up for the FP, and the Halo feature seems to drive the Size group importance for the FN event although it is a positive trend. The overall group has Location and CME History as the second and third groups. We can see a higher positive local feature importance for the Richardson's equation in both events, and we analyzed latitude as a top contributor to the FN local importance. While we had seen some negative correlation in the Number of CMEs in the Past Day, several

of the other history features seem to have a modest contribution in the FP and FN events.

Chapter 5

Forecasting SEP Intensities

The second problem that was studied in this work was predicting the ln peak intensity of 100 MeV protons associated with a CME event using its set of input features as defined in Chapter 3. In the dataset, there were a few outlier peak intensity values that caused issues with predicting the peak intensity directly. Across the entire dataset, the highest and lowest peak intensity values differed by two orders of magnitude from the highest peak intensity of 56.311 to the lowest $\frac{1}{e^2} = 0.135$, the fixed constant we used for background events. In Richardson's equation [15], the predicted intensity grows exponentially as speed increases. In addition, there were very few SEP events in comparison to the number of background events. Training an accurate exponential function to predict intensity from these few SEP events was expected to be difficult. By applying the ln to Richardson's equation [15], the equation predicts that the ln peak intensity grows linearly to the speed. A linear equation was expected to be much easier to learn on this dataset. By exponentiating our ln peak intensity prediction, we can recover the predicted peak intensity of the protons.

Accurately predicting the peak intensity of the 100 MeV protons with a CME can help categorize that event based on how severe its damage is likely to be. The threshold for determining an SEP event is not standardized among astrophysicists. Our approach is easily adapted to any selected threshold since the model is trained on and predicts the ln peak intensity. The threshold value is only used when evaluating a model's performance using classifier adapted metrics. In our case, we classify that CME event as an SEP event when their 100 MeV protons have a peak intensity value ≥1 i.e. that their ln peak intensity value is ≥0.

5.1 Approaches

All neural networks were implemented using Keras on top of TensorFlow. All neural networks had no activation function in their output layer. Their single unit output was the predicted log peak intensity of the 100 MeV protons of the associated CME event input.

5.1.1 Regular Neural Network with Oversampling (rRegNN)

The regular neural network is a multi-layer Leaky ReLU regressor using 2 hidden layers. The architecture is the same as the classification Regular Neural Network with Oversampling and is illustrated in Figure 4.1. We use the mean square error (MSE) loss function during training:

$$\mathcal{L}_{MSE} = \frac{1}{N}\sum_{n=1}^{N}(y - \hat{y})^2, \tag{5.1}$$

where N is the number of training samples, y is the actual log peak intensity, and $\hat{y}$ is the predicted log peak intensity. The dataset used during training is the original, imbalanced dataset.

As in the classification regular Neural Network, we also explored oversampling the minority SEP and Elevated events to improve performance. The oversampled training sets e.g. 10% oversampling, 20% oversampling etc. were the same between the regression and classification tasks with the exception of the regressor output instead of the classifier output. We also abuse the notation to say 0% oversampling is the original, imbalanced dataset as we did in the classifier.

5.1.2 Regression Re-training (rRT)

Similar to Thomas [19], we modified the cRT approach from the classifier task in 4.1.2 for regression learning by replacing the output layer with a linear regression output layer. The rRT method is broken into the same two stages as the cRT method. The first stage is used to learn new features from the input. Then, all but the output layer is frozen and reused with an extra hidden layer and a reinitialized and retrained output layer in the second stage. The second stage is trained on a dataset with the SEP and Elevated events oversampled. The oversampling increases the importance of the minority SEP and Elevated events which helps train the regressor on the otherwise rare events. The architecture is the same illustration as the architecture of the cRT method in Figure 4.2.

5.1.3 Regression Re-training with Autoencoder (rRT+AE)

The rRT+AE technique is almost identical to the cRT+AE method. One difference is the joint loss function is defined as:

$$\mathcal{L}_{rRT+AE} = \mathcal{L}_{MSE_{REG}} + \alpha \mathcal{L}_{MSE_{AUT}}, \tag{5.2}$$

since the regression task uses the MSE loss function, here MSE_{REG}, instead of the CE loss function. The autoencoder still uses the MSE loss function, here MSE_{AUT}. The network architecture is illustrated in Figure 5.1. The α term is estimated by separately training the branches shown in Figure 5.1 and combining their errors:

$$\alpha = \frac{1}{N}\sum_{n=1}^{N}\frac{\mathcal{L}_{MSE_{REG,n}}}{\mathcal{L}_{MSE_{AUT,n}}}, \tag{5.3}$$

where N is the number of epoch, $L_{MSE_{REG,n}}$ is the loss for the regression branch for epoch n, and $L_{MSE_{AUT,n}}$ is the loss for the autoencoder branch for epoch n.

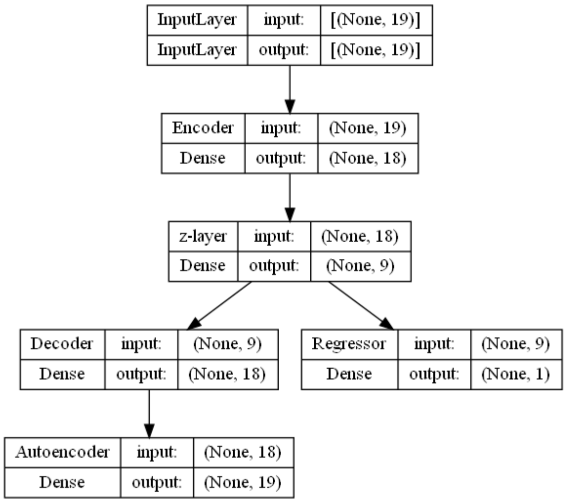

InputLayer | input: | [(None, 19)]
InputLayer | output: | [(None, 19)]
Encoder | input: | (None, 19)
Dense | output: | (None, 18)
z-layer | input: | (None, 18)
Dense | output: | (None, 9)
Decoder | input: | (None, 9)
Dense | output: | (None, 18)
Regressor | input: | (None, 9)
Dense | output: | (None, 1)
Autoencoder | input: | (None, 18)
Dense | output: | (None, 19)

Figure 5.1: Network architecture for rRT+AE.

With the α value estimated, we train the network in two steps. In the first step, we train the network shown in Figure 5.1. The goal of the first step is to learn the features of the input data which will be preserved in the z-layer like the rRT method. In the second step, the Decoder/Autoencoder branch is discarded, the InputLayer up to the z-layer are frozen and reused, a new hidden layer is inserted between the z-layer and the Regressor, and the Regressor layer is reinitialized and retrained. The resulting network architecture is the same as the second stage of rRT, and we use an oversampled dataset for training.

5.1.4 Incorporating Richardson's model

One of the input features was part of an entire formula derived by Richardson et al. [15] to model the peak intensity of a CME event with connection angle ϕ and speed V shown in Equation 5.4.

$$I(\phi) \approx 0.013exp(0.0036V - \frac{\phi^2}{2\sigma^2}), \sigma = 43^\circ \tag{5.4}$$

Richardson et al. used CMEs from the CDAW catalog with proton intensities from 14- to 24-MeV. Our dataset was composed of CMEs with proton intensities of >100 MeV. We theorized we could improve Richardson's formula by learning new coefficients in the Richardson equation and replacing the Richardson input feature in our network with a better approximation estimated from our dataset. First, we rewrote Richardson's equation into a form that was easier to translate into a neural network. We introduced new notation, w_{exp} and w_v, to form the equation:

$$I(\phi) \approx w_{exp}exp(w_vV - \frac{\phi^2}{2\sigma^2}), \sigma = 43^\circ \tag{5.5}$$

where $w_{exp} = 0.013$ and $w_v = 0.0036$ in Richardson's equation. To predict the log of the intensity, we took the log of both sides of the equation:

$$ln(I(\phi)) \approx ln(w_{exp}) + w_vV - \frac{\phi^2}{2\sigma^2}, \sigma = 43^\circ \tag{5.6}$$

If we considered $x_1 = V$, $x_2 = -\frac{\phi^2}{2\sigma^2}, \sigma = 43^\circ$, and $w_0 = ln(w_{exp})$ then we arrived at our final form of Richardson's log equation:

$$ln(I(\phi)) \approx w_0 + w_vx_1 + 1x_2 \tag{5.7}$$

5.1.4.1 Richardson Network

We form a simple neural network to learn Equation 5.7 where we learn a new w_0 and w_v while providing x_1 and x_2 input features from the SEP and Elevated CME events. We specifically do not train on the Background CME events because we synthetically fixed the peak intensity values of the background events to be a small constant, and it would be detrimental to include these filler values during training. The network architecture is shown in Figure 5.2. There are two inputs: x_1 which is the speed as marked in the DONKI catalog and x_2 which is the connection angle term defined above calculated using the DONKI catalog features. The output feature is the log of the peak intensity. There is no activation function in the dense layer, and the dense layer's construction replicates Equation 5.7. The bias term for the dense layer is w_0. From w_0, we can recover w_{exp} through the equation: $w_{exp} = exp(w_0)$. The w_v is the weight of the x_1 term. To avoid training the weight of the connection angle term, we fix it at 1 by placing a constraint on the dense layer making that weight always 1.

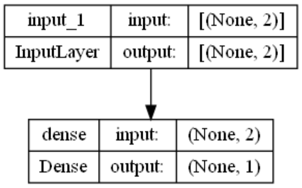


Figure 5.2: Network architecture for learning Richardson's formula coefficients.

When training this network, we use the raw DONKI speed and connection angle values i.e. they are not normalized in the range $[0, 1]$ as other input features in our other networks. Richardson's equation was based on the raw CME speed and connection angle, not normalized values. We want to learn new coefficients to Richardson's equation, so we do not want to change the input format to the model. In addition, we want the differences in the DONKI speed to be emphasized such as 2200 vs 1000 which would not be as extreme with normalized values.

5.1.4.2 Incorporating Richardson Network

5.1.4.2.1 Combining Richardson Forecast (RC)

To incorporate Richardson Network, our first approach (denoted as RC), combines the forecast from Richardson Network with the forecast from our model for each input instance. The technique is broken down into two steps. In step 1, we use the method described in the previous section to learn new w_{exp} and w_v coefficients for the Richardson network. In step 2, we train our regular NN alongside the frozen Richardson network to predict the log peak intensity of the CME event. In this approach, we do not use any Richardson feature as input. The Richardson network frozen alongside our training network provides the input from the Richardson equation. The output from our regular NN and the frozen Richardson network are combined together in a hidden layer, so the weights used in combining the frozen Richardson network and our regular NN are learned during training. To formalize step 2, we first notate our feature vector as $\overrightarrow{x}$ which includes the raw DONKI speed and connection angle term defined above when formulating the Richardson network and all but the Richardson network feature from the normalized features described in the Description of SEP Data feature list. We notate $\overrightarrow{x_R}$ as the isolated raw DONKI speed and connection angle term features from $\overrightarrow{x}$. We notate $\overrightarrow{x_N}$ as the isolated normalized features from $\overrightarrow{x}$. Then, the output from our neural network is the combination of the output from the Richardson Network, RN, and the output of our Neural Network, NN, using the equation:

$$ln(Intensity(\overrightarrow{x})) = w_0 + w_1 * RN(\overrightarrow{x_R}) + w_2 * NN(\overrightarrow{x_N}) \tag{5.8}$$

The network architecture is illustrated in Figure 5.3. The combine layer introduces three trainable weights in the output dense_3 layer: the bias term, w_0, the weight for the frozen Richardson network, w_1, and the weight for our regular NN, w_2. In step 2 of approach RC, we apply the previous approaches including Regular Neural Network with Oversampling which we call RC+rRegNN, rRT which we call RC+rRT, and rRT+AE which we call RC+rRT+AE. The Richardson network remains fixed and frozen after formation in step 1.

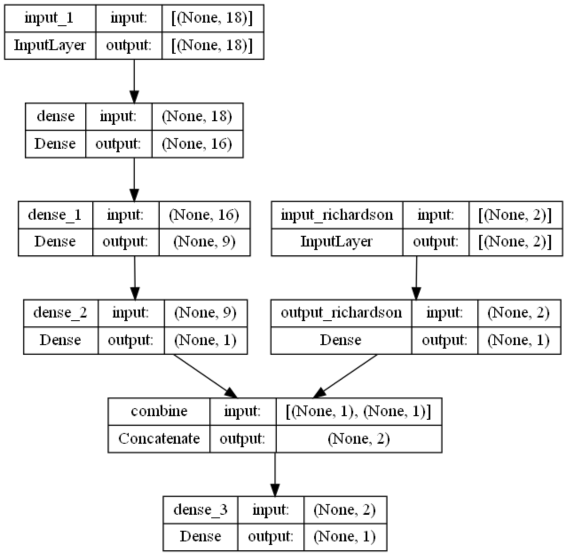

input_1 | input: | [(None, 18)]
InputLayer | output: | [(None, 18)]
dense | input: | (None, 18)
Dense | output: | (None, 16)
dense_1 | input: | (None, 16)
Dense | output: | (None, 9)
input_richardson | input: | [(None, 2)]
InputLayer | output: | [(None, 2)]
dense_2 | input: | (None, 9)
Dense | output: | (None, 1)
output_richardson | input: | (None, 2)
Dense | output: | (None, 1)
combine | input: | [(None, 1), (None, 1)]
Concatenate | output: | (None, 2)
dense_3 | input: | (None, 2)
Dense | output: | (None, 1)

Figure 5.3: Network architecture for RC.

5.1.4.2.2 Learning Richardson Error (RE)

To incorporate Richardson Network, our second approach (denoted as RE), estimates the error of Richardson forecast for each input instance. By taking the ln of both sides, the formula produced by Richardson is an approximation for the ln peak intensity of proton intensities associated with a CME. Instead of training a neural network to predict the ln peak intensity of the CME events as in our other techniques, we experiment with training a neural network to model the error remaining in the prediction provided by the Richardson ln equation using our trained coefficients. The technique is broken down into two steps. In step 1, we use the method described in the previous section to learn new w_{exp} and w_v coefficients for the Richardson network. In step 2, we freeze the model used to train the coefficients and train our regular NN model alongside combining in 1:1 the output from the frozen Richardson network and our NN that is being trained. In this approach, we do not use any Richardson feature as input. The Richardson network frozen alongside our training network provides the input from the Richardson equation. To formalize step 2, we use the same notation as in 5.1.4.2.1 with the overall feature vector $\overrightarrow{x}$ and isolated feature vectors $\overrightarrow{x_R}$ and $\overrightarrow{x_N}$. The output from our neural network is the prediction from the Richardson Network, RN, with some error, $\epsilon(\overrightarrow{x_N})$, modeled by our Neural Network (NN) summarized in the equation:

$$ln(Intensity(\overrightarrow{x})) = RN(\overrightarrow{x_R}) + \epsilon(\overrightarrow{x_N}) = RN(\overrightarrow{x_R}) + NN(\overrightarrow{x_N}) \tag{5.9}$$

The network architecture is illustrated in Figure 5.4. The dense_3 layer has no bias term, and the weights from both incoming branches are constrained to 1. In step 2 of approach RE, we apply the previous approaches Regular Neural Network with Oversampling which we call RE+rRegNN, rRT which we call RE+rRT, and rRT+AE which we call RE+rRT+AE. The Richardson network remains fixed and frozen after formation in step 1.

5.1.5 DenseLoss (DL)

Steininger et al. proposed a sample weighting approach called DenseWeight which they included into a cost-sensitive learning approach called DenseLoss [17]. The idea of DenseWeight is to use the distribution of the training examples to weight lower-probability training samples higher (SEP and Elevated events in our case) in the loss function than the common training samples (background events in our case). To apply

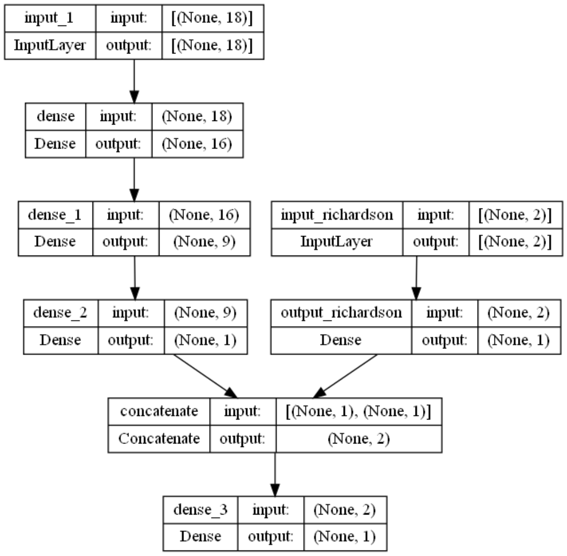

input_1 | input: | [(None, 18)]
InputLayer | output: | [(None, 18)]
dense | input: | (None, 18)
Dense | output: | (None, 16)
dense_1 | input: | (None, 16)
Dense | output: | (None, 9)
input_richardson | input: | [(None, 2)]
InputLayer | output: | [(None, 2)]
dense_2 | input: | (None, 9)
Dense | output: | (None, 1)
output_richardson | input: | (None, 2)
Dense | output: | (None, 1)
concatenate | input: | [(None, 1), (None, 1)]
Concatenate | output: | (None, 2)
dense_3 | input: | (None, 2)
Dense | output: | (None, 1)

Figure 5.4: Network architecture for RE.

it, we need to calculate a DenseWeight for the SEP events, the Elevated events, and the Background events. We define our datasets, d_f, to include all of the SEP events, all of the Elevated events, but only one Background event where d_f is a list of the included peak intensity values for events in 3-fold dataset f for $f \in \{1, 2, 3\}$. The SEP and Elevated peak intensity values came from real data, but the Background peak intensity value was a set constant value. To avoid biasing the weight computation, we only include a single instance of a background event in each d_f. We still need the one instance because we need to calculate its DenseWeight. The first step to calculate the DenseWeight is to approximate the target value distribution. In our case, our event distribution matches a Pareto Distribution. The Probability Density Function (PDF) of a Pareto distribution is given by:

$$f(x;\alpha) = \alpha \frac{x_{min}^{\alpha}}{x^{\alpha+1}}; x \geq x_{min}, \alpha > 0 \tag{5.10}$$

where α is a shape parameter. We use the Maximum Likelihood Estimation method to estimate the α values for each of the 3-fold datasets as defined in Equation 5.11 where N is the number of elements in the dataset d_f, x_i are elements in d_f, and $x_{min} = min(x_i)$ for all x_i in d_f.

$$\alpha = \frac{N}{\sum_{i=1}^{N} ln(x_i) - N ln(x_{min})} \tag{5.11}$$

With pareto's α calculated, we resume the notation of Steininger and notate the pareto density function as p(y):

$$p(y) = \alpha \frac{y_{min}^{\alpha}}{y^{\alpha+1}} \tag{5.12}$$

From there, we can arrive at the final form of the DenseWeight's weight function:

$$f_w(\alpha_{DW}, y) = \frac{max(1 - \alpha_{DW} p'(y), \epsilon)}{\frac{1}{N} \sum_{i=1}^{N} \left(max(1 - \alpha_{DW} p'(y_i), \epsilon)\right)} \tag{5.13}$$

where we notate the DenseWeight parameter as α_{DW} since we already use α in the pareto distribution and ϵ is a small constant such as 10^{-5}. The DenseWeight parameter α_{DW} tunes the amount of weight that is applied. At 0, DenseWeight is disabled. At 1, the common data point weights reach epsilon. All weights are positive for $\alpha_{DW} < 1$, and common data point weights are negative for $\alpha_{DW} > 1$.

The DenseWeight function is used directly in the Dense Loss function:

$$\mathcal{L}_{DenseLoss}(\alpha_{DW}) = \frac{1}{N} \sum_{i=1}^{N} f_w(\alpha_{DW}, y_i) \cdot M(\hat{y_i}, y_i) \tag{5.14}$$

where we use the MSE as the metric M.

The DenseWeight function applies a weight to each individual sample increasing the importance of rare events such as SEP events. This replaces the role of using an oversampled dataset but weights samples differently. In an oversampled dataset, rare events are duplicated but weighted equally amongst their class in terms of importance. DenseWeight applies varied weights within each class and across the entire dataset. Concretely, this increases the importance of SEP events with much higher ln peak intensity values more than other SEP events with smaller ln peak intensity values. For this reason, we do not use oversampled datasets in this approach. Instead, we perform a hyperparameter search for α_{DW} that maximizes the performance in each DenseLoss technique discussed below. We test α_{DW} values in the list: [0, 0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9, 1.0, 1.5, 2.0]. We compare their performances over an average of 5 runs to find the best performer.

We experiment with using the DenseLoss technique with the previous rRegNN, rRT, and rRT+AE techniques. Within these techniques, we replace oversampling and MSE with DenseLoss. We denote the techniques as DL+rRegNN, DL+rRT, and DL+rRT+AE.

5.2 Experimental Evaluation

5.2.1 Evaluation Criteria

In the regression task, we want to quantify the difference between the predicted ln peak intensity and the actual ln peak intensity. We leverage several metrics to perform this task. One metric is the mean absolute error (MAE). We are particularly interested in SEP events, so we calculate the MAE over SEP events only as well as the MAE over the entire test set. We also calculate the pearson correlation (PCC) for the SEP events only and the SEP+Elevated events. The background events were arbitrarily set to a constant small intensity value, so they are not added to the PCC metrics. The pearson correlation measures the linear correlation between two sets of values. For a vector of values $\overrightarrow{a} = \{a_0, a_1, ..., a_N\}$ and $\overrightarrow{b} = \{b_0, b_1, ..., b_N\}$, the PCC is defined as

$$PCC = \frac{\sum_{i=0}^{N}(a_i - \bar{a})(b_i - \bar{b})}{\sqrt{\sum_{i=0}^{N}(a_i - \bar{a})^2}\sqrt{\sum_{i=0}^{N}(b_i - \bar{b})^2}} \tag{5.15}$$

where $\bar{a} = \frac{1}{N}\sum_{i=0}^{N} a_i$ and $\bar{b} = \frac{1}{N}\sum_{i=0}^{N} b_i$ are the mean values for their respective sets. In our case, $\overrightarrow{a}$ corresponds to the ln peak intensities predicted by the model, $\hat{y}_{predicted}$, and $\overrightarrow{b}$ corresponds to the observed ln peak intensities $\hat{y}_{observed}$.

We also adapt the classification metrics to the regression task. We classify events as either SEP or non-SEP (including Elevated and Background events) based on the peak intensity. If the peak intensity is ≥1 pfu, an instance is classified as an SEP event. Since we predict the ln peak intensity, it follows that an event is predicted to be an SEP event if its predicted ln peak intensity is ≥0, otherwise it is predicted to be a non-SEP event. This allows us to form a similar confusion matrix to the classification confusion matrix as illustrated in Table 5.1. We calculate the same additional metrics: F1, TSS, and HSS.

Table 5.1: Confusion matrix for regression tasks.

	Predicted Log Peak ≥ 0	Prediction < 0
Actual Log Peak Intensity ≥ 0	True Positive (TP)	False Negative (FN)
Actual Log Peak Intensity < 0	False Positive (FP)	True Negative (TN)

5.2.2 Evaluation Procedures

We use the same training and test sets as the classification task. When learning the Richardson coefficients, we also apply 4-fold cross validation to the training and validation sets in each fold. The 4-fold of the training and validation sets help give a more accurate epoch count to use during actual training. We also train each model 5 times with an initial run to find the optimal epoch number as in the classification evaluation procedures. The resulting metrics are reported as an average over the 5 runs.

5.2.2.1 Procedures for Training Regular Neural Network with Oversampling

The procedures for this approach were identical to the Regular Neural Network with Oversampling defined for the classifier in 4.2.2.1 with the exception of using the mean squared error loss function in the output. The optimal epoch number varied per oversampling rate. We summarize the epoch numbers in Table 5.2.

Table 5.2: Epoch counts for rRegNN, rRT, and rRT+AE approaches used per oversampling percent.

Oversampling Percent	rRegNN	rRT	rRT+AE
0	11594	N/A	N/A
10	208	414	427
20	155	268	295
30	5862	1335	406
40	6906	11994	1514
50	5800	10645	9695
60	6400	10502	7707
70	4923	7141	24561
80	3577	13613	24598
90	4897	5362	18818

5.2.2.2 Procedures for Training rRT

The procedures for this approach were identical to the cRT defined for the classifier in 4.2.2.2 with the exception of using the mean squared error loss function in the output. The optimal epoch number for the second step varied per oversampling rate. We summarize the epoch numbers in Table 5.2.

5.2.2.3 Procedures for Training rRT+AE

The procedures for this approach were identical to the cRT+AE defined for the classifier in 4.2.2.3 with the exception of using the mean squared error loss function in the regressor branch which replaced the classifier branch. The optimal epoch count for the first step was 11784. The optimal epoch number for the second step varied per oversampling rate. We summarize the epoch numbers in Table 5.2.

5.2.2.4 Procedures for Learning Richardson Coefficients

To learn the Richardson coefficients, we used 4-fold cross validation to split up the training and validation data sets. All models were initialized initially with a random uniform initializer between -0.05 and 0.05. All models also used an Adam optimizer with a learning rate of 0.0001 and Adam epsilon 1.0. The average optimal epoch number varied between the 3-fold test sets. We summarize the epoch numbers in Table 5.3.

Table 5.3: Epoch counts for 3-fold learning Richardson coefficients.

Fold	Epochs
1	272,974
2	327,482
3	224,278

5.2.2.5 Procedures for Training RC

All RC approaches began with learning the new Richardson coefficients. The network used to train the Richardson coefficients was then frozen and placed alongside the typical network of the respective techniques e.g. the rRegNN network for the RC+rRegNN approach, the rRT network for the RC+rRT approach, etc. The procedure for the respective technique was directly applied to the left branch of the RC architecture illustrated in Figure 5.3. The optimal epoch number for RC+rRegNN, the second step of RC+rRT, and the second step of RC+rRT+AE varied per oversampling rate. We summarize the epoch numbers in Table 5.4.

Table 5.4: Epoch counts for RC+rRegNN, RC+rRT, and RC+rRT+AE approaches used per oversampling percent.

Oversampling Percent	RC+rRegNN	RC+rRT	RC+rRT+AE
0	39069	N/A	N/A
10	19091	6830	1669
20	14588	2672	3688
30	5225	3815	3301
40	6181	2478	3134
50	4675	819	11080
60	6837	640	8118
70	3998	413	7386
80	17025	11350	3998
90	4727	40563	2979

5.2.2.6 Procedures for Training RE

All RE approaches began with learning the new Richardson coefficients. The network used to train the Richardson coefficients was then frozen and placed alongside the typical network of the respective techniques e.g. the rRegNN network for the RE+rRegNN approach, the rRT network for the RE+rRT approach, etc. The procedure for the respective technique was directly applied to the left branch of the RE architecture illustrated in Figure 5.4. The optimal epoch number for RE+rRegNN, the second step of RE+rRT, and the second step of RE+rRT+AE varied per oversampling rate. We summarize the epoch numbers in Table 5.5.

5.2.2.7 Procedures for Training DL

To form the Pareto PDF of the peak intensity, we estimated the α shape parameter using the Maximum Likelihood Estimation Method. There are 3-folds, so we have

Table 5.5: Epoch counts for RE+rRegNN, RE+rRT, and RE+rRT+AE approaches used per oversampling percent.

Oversampling Percent	RC+rRegNN	RC+rRT	RC+rRT+AE
0	6497	N/A	N/A
10	12393	888	4105
20	11254	12744	2900
30	12546	1350	1899
40	12026	730	13420
50	5557	749	12314
60	35600	580	6436
70	20234	522	26623
80	46344	404	29446
90	4536	253	19900

three datasets, d_f, for $f \in \{1, 2, 3\}$. We summarize the α values in Table 5.6.

Table 5.6: Pareto estimates for α parameter for 3-fold datasets.

Fold	α Estimate
1	0.5338
2	0.4833
3	0.4937

To implement the DenseLoss function, we created a lookup table for each α_{DW} to return the DenseWeight of a CME event from its ln peak intensity value. The lookup table used integers as keys, but the ln peak intensity values were floating point values. To form the keys of the table, and every time we provided a ln peak intensity value as a query, we converted the floating point to an integer by multiplying by 10^5 and cutting off any leftover fractional component. We created a custom DenseLoss loss function implementation for each α_{DW} using their respective DenseWeight lookup table to lookup the DenseWeight and apply it to the MSE loss function metric. For each DL technique, we tested α_{DW} values in the list: $[0, 0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9, 1.0, 1.5, 2.0]$. Each α_{DW} was tested in its own model for a total of 13 models per technique: DL+rRegNN, DL+rRT, and DL+rRT+AE. To initialize each model, we used a random uniform initializer between -0.05 and 0.05. We used an Adam optimizer with a learning rate of 0.001 and Adam epsilon 1.0. Hidden layers used the LeakyReLU activation function with an alpha of 0.3. In addition, we used a batch size of 179 when training with a validation set and 168 when training with the combined training/validation set. We do not use oversampling in these approaches which means a batch during training might not have any SEP or Elevated events at all. Using these batch sizes, we aimed to get 2 SEP or Elevated events per batch.

The DL+rRegNN procedure was the same as the rRegNN procedure except using the values above, and we used the DenseLoss function using the DenseWeight corresponding to the tested α_{DW} value. The DL+rRT procedure began with following the first step of the rRT procedures except using the values above. We then followed the second step of the rRT procedures, but we replaced the MSE loss function with the DenseLoss function using the DenseWeight corresponding to the tested α_{DW} value. The DL+rRT+AE procedure began with following the first step of the rRT+AE procedures except using the values above. We then followed the second step of the rRT+AE procedures, but we replaced the MSE loss function with the DenseLoss function using the DenseWeight corresponding to the tested α_{DW} value. In both the DL+rRT second step and the DL+rRT+AE second step, we continued to use the original, imbalanced dataset.

The optimal epoch number varied in each technique per α_{DW}. We summarize the epoch numbers in Table 5.7.

5.2.3 Results

In this section, we present the results for the first fold of the 3-fold dataset. The same training and validation sets are used across the respective techniques i.e. cRegNN

Table 5.7: Epoch counts for DenseLoss DL+rRegNN, DL+rRT, and DL+rRT+AE approaches used per oversampling percent.

α_{DW}	DL+rRegNN	DL+rRT	DL+rRT+AE
0.0	61684	145302	137272
0.1	59977	134839	136239
0.2	64299	143775	126619
0.3	62210	149600	108119
0.4	55000	146476	66763
0.5	50731	45532	39361
0.6	47385	41910	33870
0.7	42900	48648	93996
0.8	41850	130969	39055
0.9	55758	124152	31672
1.0	6638	110394	107301
1.5	98885	98779	86567
2.0	13959	99596	87271

with 10% oversampling is the same training and validation sets as cRT with 10% oversampling. The same test set was used for all approaches. Therefore, the metrics presented below are comparable across techniques. Metrics are discussed per method for ease of presentation.

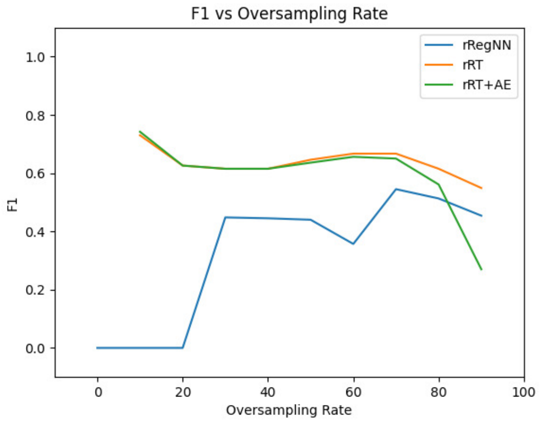


Figure 5.5: F1 score vs oversampling rate for rRegNN, rRT, and rRT+AE techniques.

Table 5.8: Summarized regression metric results averaged over 5 runs for rRegNN, rRT, and rRT+AE techniques with 3-fold Dataset 1. Each row in a technique has an oversampled dataset by some percent indicated by the trailing number after the row technique name, e.g. rRegNN 10 is the rRegNN technique with 10% oversampling. The best value for each metric in each technique is underlined with the best overall value for each metric in bold.

Technique	PCC SEP	PCC SEP+Elevated	MAE SEP	MAE
rRegNN 0	-0.042	0.693	2.753	**0.068**
rRegNN 10	**0.175**	0.390	3.074	0.170
rRegNN 20	0.153	0.334	2.711	0.221
rRegNN 30	-0.083	0.698	2.134	0.131
rRegNN 40	-0.269	0.647	2.151	0.118
rRegNN 50	-0.145	0.660	2.260	0.118
rRegNN 60	-0.450	0.519	2.512	0.117
rRegNN 70	-0.258	0.632	2.161	0.126
rRegNN 80	-0.155	0.686	2.125	0.142
rRegNN 90	-0.307	0.623	2.235	0.139
rRT 10	0.143	0.751	1.857	0.104
rRT 20	0.105	0.742	1.653	0.147
rRT 30	0.072	0.735	1.537	0.188
rRT 40	0.010	0.719	1.554	0.236
rRT 50	0.003	0.720	1.550	0.311
rRT 60	-0.025	0.718	1.557	0.397
rRT 70	-0.033	0.711	1.557	0.539
rRT 80	-0.033	0.667	1.495	0.627
rRT 90	0.065	0.660	1.716	1.108
rRT+AE 10	0.137	**0.753**	1.821	0.103
rRT+AE 20	0.094	0.742	1.613	0.144
rRT+AE 30	0.067	0.737	1.542	0.189
rRT+AE 40	0.061	0.735	1.530	0.246
rRT+AE 50	0.002	0.725	1.548	0.312
rRT+AE 60	-0.016	0.719	1.559	0.392
rRT+AE 70	0.013	0.678	1.458	0.480
rRT+AE 80	-0.156	0.601	**1.443**	0.658
rRT+AE 90	-0.273	0.320	2.109	0.866

Table 5.9: Summarized classifier adapted metric results averaged over 5 runs for rRegNN, rRT, and rRT+AE techniques with 3-fold Dataset 1. Formatting is the same as in Table 4.4.

Technique	FP	FN	TP	TN	F1	HSS	TSS
rRegNN 0	0.4	5.0	0.0	680.6	0.000	-0.001	0.000
rRegNN 10	**0.0**	5.0	0.0	**681.0**	0.000	0.000	0.000
rRegNN 20	**0.0**	5.0	0.0	**681.0**	0.000	0.000	0.000
rRegNN 30	3.6	2.4	2.6	677.4	0.448	0.443	0.515
rRegNN 40	3.4	2.6	2.4	677.6	0.445	0.441	0.475
rRegNN 50	3.0	2.6	2.4	678.0	0.440	0.436	0.476
rRegNN 60	2.8	3.2	1.8	678.2	0.357	0.353	0.356
rRegNN 70	2.4	2.2	2.8	678.6	0.545	0.542	0.556
rRegNN 80	2.2	2.4	2.6	678.8	0.513	0.510	0.517
rRegNN 90	3.0	2.6	2.4	678.0	0.454	0.450	0.476
rRT 10	1.6	1.2	3.8	679.4	0.730	0.728	0.758
rRT 20	3.8	1.0	4.0	677.2	0.626	0.622	0.794
rRT 30	4.0	1.0	4.0	677.0	0.615	0.612	0.794
rRT 40	4.0	1.0	4.0	677.0	0.615	0.612	0.794
rRT 50	3.4	1.0	4.0	677.6	0.646	0.643	0.795
rRT 60	3.0	1.0	4.0	678.0	0.667	0.664	0.796
rRT 70	3.0	1.0	4.0	678.0	0.667	0.664	0.796
rRT 80	5.4	**0.4**	**4.6**	675.6	0.615	0.612	**0.912**
rRT 90	5.0	1.4	3.6	676.0	0.549	0.545	0.713
rRT+AE 10	1.8	1.0	4.0	679.2	**0.742**	**0.740**	0.797
rRT+AE 20	3.8	1.0	4.0	677.2	0.626	0.622	0.794
rRT+AE 30	4.0	1.0	4.0	677.0	0.615	0.612	0.794
rRT+AE 40	4.0	1.0	4.0	677.0	0.615	0.612	0.794
rRT+AE 50	3.6	1.0	4.0	677.4	0.636	0.633	0.795
rRT+AE 60	3.2	1.0	4.0	677.8	0.656	0.653	0.795
rRT+AE 70	4.2	0.6	4.4	676.8	0.650	0.647	0.874
rRT+AE 80	6.6	0.6	4.4	674.4	0.561	0.557	0.870
rRT+AE 90	15.2	1.8	3.2	665.8	0.270	0.262	0.618

5.2.3.1 Regular Neural Network with Oversampling (rRegNN)

Table 5.8 lists the regression metrics averaged over 5 runs for pearson correlation (PCC) on the SEP events only, PCC on the SEP+Elevated events, the mean absolute error (MAE) on the SEP events, and the MAE for all events. Table 5.9 lists the classification metrics adapted to the regression problem. The PCC metric for SEP events is only maximized with the 10% oversampled training set across all three techniques with the maximum value in rRegNN 10. This is also the worst performing technique because it under-predicts all events with all 5.0 FN and 0.0 FP. Like the classifier task, we can achieve all TN easily by under-predicting all events. The best performing F1 technique is rRegNN 70 where our FN falls to 2.2, but we gain 2.4 FP events. The MAE of the SEP events while one of the lower values at 2.161 is still significant for rRegNN 70. There are further improvements by applying the other techniques.

Figure 5.5 illustrates how the F1 score changes as the rRegNN oversampling amount varies. The plot slightly resembles a step function with a sudden jump at 30% oversampling and then later at 70% oversampling. The F1 begins to fall after the beginning of each step which is a bit unexpected. We did see similar peak values around 30% and 70% oversampling in the classifier for cRegNN, although, in this case, the 70% oversampling is the higher peak.

5.2.3.2 Regression Re-training (rRT)

The worst F1 score in rRT is still better than the best F1 score in rRegNN. The best F1 in rRT is 0.730 in rRT 10. The increase in performance is due to reducing the FP by 1.0 and the FP by 0.8. This technique also has the highest HSS, PCC SEP, PC SEP+Elevated, and MAE than the other rRT techniques. Like the classifier analysis in cRT, the rRT approach, specifically rRT 80, has the lowest FN across all three basic techniques at 0.4 FN events. While almost a perfect predictor, the FP have increased up to 5.4 illustrating the same trade-off that more TP mean more FP. The rRT 10 technique does have the largest error in the MAE for SEP events only which we reduce by using the autoencoder.

Figure 5.5 illustrates how the F1 score changes as the rRT oversampling amount varies. In rRT, we are back to a secondary peak around 70% instead of the primary peak from rRegNN. If not for the sudden peak at 10% oversampling, the peak F1 score would still be at 70%. Since the F1 score for 70 and 80 are the same, it could be there is an event higher F1 score somewhere in-between, but we leave that exploration for future work.

5.2.3.3 Regression Re-training with Autoencoder (rRT+AE)

The rRT+AE 10 technique has the highest F1 score of 0.742. Clearly, the rRT component makes most of the improvement over the original rRegNN approach, but the autoencoder increases our performance further. The FN has fallen by 0.2 to 1.0 and the FP has risen by 0.2 to 1.8 over the rRT 10 technique. The rRT+AE 10 technique has the highest PCC of SEP+Elevated across all three techniques. The high pearson score suggests a strong linear correlation between the rRT+AE prediction and the actual ln peak intensity values. There is room to improve however with the remaining 1.821 MAE in the SEP events.

Figure 5.5 illustrates how the F1 score changes as the rRT+AE oversampling amount varies. The plot for the rRT+AE looks very similar to the rRT F1 score plot. The difference is that the 90% oversampling has a sharp decline in F1 score performance. It is not a large surprise that the plots are similar since both techniques have two stages of training focused on learning the representation and classifier separately.

5.2.3.4 Learning Richardson Coefficients

We learned very different coefficients for Richardson's equation as seen in Table 5.10. Our w_v coefficient is just less than half of the original. Typically, SEP events with higher speed have higher peak intensity. Yet, according to our learned w_v coefficient, the speed is unexpectedly not emphasized very much. The w_{exp} coefficient meanwhile

Table 5.10: Original vs learned coefficients with their MAE of log intensity and intensity (MAE ln, MAE), MAE of log intensity and intensity for SEP events (MAE SEP ln, MAE SEP) calculated for SEP+Elevated dataset. Bold indicates the better metric value.

Approach	w_{exp}	w_v	MAE ln	MAE	MAE SEP ln	MAE SEP
Original	0.013	0.0036	**1.405**	9.146	2.073	18.109
Learned	0.383	0.0015	1.559	**7.727**	**1.806**	**14.907**

Table 5.11: Classification metrics using ln Richardson's Equation with original vs learned coefficients calculated for SEP+Elevated+Background dataset. Bold indicates the better metric value.

Approach	FP	FN	TP	TN	F1	HSS	TSS
Original	**5.0**	**1.0**	**4.0**	**676.0**	**0.571**	**0.567**	**0.793**
Learned	13.0	**1.0**	**4.0**	668.0	0.364	0.356	0.781

increased by about 30x. In Richardson's ln equation, this coefficient is a constant applied to all events, so it is not as important as the w_v coefficient based on velocity.

These learned coefficients are our network's best fit for the SEP and Elevated events in the training/validation sets combined from 3-fold dataset 1. To evaluate our coefficients against Richardson's coefficients, we compared several MAE values. Our coefficient network was trained to learn the Richardson's ln equation, so the first MAE is calculated between the predicted ln Peak Intensity (from our model or from Richardson's original ln equation) and the actual ln peak intensity value. We also wanted to compare the MAE on the actual peak intensity value. To calculate this MAE of intensity, we exponentiated each prediction value before calculating the MAE against the actual peak intensity value (instead of the actual ln peak intensity value in the previous calculation). These MAE values were calculated for the entire training set including SEP and Elevated events. We repeated these two MAE calculations on just the SEP events and report those statistics as well in Table 5.10.

For the MAE of Intensity, MAE of SEP ln Intensity, and MAE of SEP Intensity, we perform better than Richardson's original formula. We perform slightly worse when comparing the MAE on the ln intensity across SEP and Elevated events increasing from 1.405 to 1.559. When we examine the main contributors to the increased error in our learned model, we find the largest error in an outlier SEP event. The output of our learned model for that event is much less than Richardson's prediction at -1.254 vs Richardson's 0.036 though we both under-predict the actual ln peak intensity of 4.031. However, when examining the other events, in 4 of them we have on average 0.656 less error. It is our decreased error on these 4 events that dominate our increased performance on the MAE of Intensity. In addition, we reduced the gap between our prediction and Richardson's prediction after exponentiating in most of the cases where we had larger errors in ln Intensity.

We also calculated the classification metrics of the ln Richardson equation using the original Richardson coefficients and our learned coefficients shown in Table 5.11. For these metrics, we used all events including SEP, Elevated, and Background events unlike the MSE calculations which only used the SEP+Elevated dataset. When comparing the classification metrics, we see worse classification metrics in our learned model. Using the original approach, it correctly classifies all but 1.0 TP event in the test set. Our learned approach also generates only 1.0 FN event, but it also generates 13.0 FP events versus the original 5.0 FP events. The FN event is not the same between the two approaches, we missed the outlier SEP event in the learned approach. However, on the remaining TP events, we reduced their error between the prediction and actual ln peak intensity considerably as evidenced by the MAE results. The FP errors in the learned approach seem to be due to overprediction as most had higher predictions than Richardson's original equation. In particular, all but 1 of the Elevated events were predicted higher than their actual 100 MeV Peak Intensity ln.

5.2.3.5 Combining Richardson Forecast (RC)

Table 5.12: Summarized regression metric results averaged over 5 runs for RC+rRegNN, RC+rRT, and RC+rRT+AE techniques with 3-fold Dataset 1. Formatting is the same as in Table 4.4.

Technique	PCC SEP	PCC SEP+Elevated	MAE SEP	MAE
RC+rRegNN 0	-0.629	0.313	3.092	**0.062**
RC+rRegNN 10	-0.382	0.424	3.061	0.114
RC+rRegNN 20	0.109	0.525	3.255	0.124
RC+rRegNN 30	**0.333**	**0.813**	2.589	0.208
RC+rRegNN 40	0.259	0.692	2.890	0.188
RC+rRegNN 50	0.185	0.731	2.584	0.229
RC+rRegNN 60	-0.113	0.656	2.717	0.152
RC+rRegNN 70	0.104	0.778	2.459	0.210
RC+rRegNN 80	-0.146	0.505	3.010	0.126
RC+rRegNN 90	-0.384	0.537	2.761	0.176
RC+rRT 10	-0.642	0.139	3.128	0.096
RC+rRT 20	-0.627	0.165	3.015	0.145
RC+rRT 30	-0.666	0.188	2.847	0.189
RC+rRT 40	-0.657	0.201	2.781	0.246
RC+rRT 50	-0.641	0.276	2.532	0.424
RC+rRT 60	-0.704	0.253	2.423	0.598
RC+rRT 70	-0.692	0.264	2.312	0.677
RC+rRT 80	0.216	0.295	2.238	0.560
RC+rRT 90	0.295	-0.012	3.321	0.470
RC+rRT+AE 10	-0.052	0.665	2.310	0.158
RC+rRT+AE 20	0.055	0.733	1.660	0.176
RC+rRT+AE 30	-0.050	0.731	1.588	0.234
RC+rRT+AE 40	-0.217	0.722	1.559	0.289
RC+rRT+AE 50	-0.183	0.686	1.571	0.354
RC+rRT+AE 60	-0.238	0.543	1.723	0.426
RC+rRT+AE 70	-0.076	0.663	1.477	0.507
RC+rRT+AE 80	-0.183	0.676	1.419	0.592
RC+rRT+AE 90	0.283	0.809	**1.251**	0.887

According to our MAE analysis, our Richardson coefficients perform better than the original Richardson equation. It is true the MAE and MAE of SEP is lower when using the RC techniques than over the original three techniques as seen in Table 5.12.

Table 5.13: Summarized classifier adapted metric results averaged over 5 runs for RC+rRegNN, RC+rRT, and RC+rRT+AE techniques with 3-fold Dataset 1. Formatting is the same as in Table 4.4.

Technique	FP	FN	TP	TN	F1	HSS	TSS
RC+rRegNN 0	2.2	4.8	0.2	678.8	0.050	0.046	0.039
RC+rRegNN 10	2.6	4.6	0.4	678.4	0.102	0.097	0.079
RC+rRegNN 20	2.8	4.8	0.2	678.2	0.040	0.035	0.039
RC+rRegNN 30	3.6	4.0	1.0	677.4	0.203	0.197	0.196
RC+rRegNN 40	3.4	4.4	0.6	677.6	0.129	0.123	0.117
RC+rRegNN 50	3.8	3.8	1.2	677.2	0.217	0.211	0.235
RC+rRegNN 60	3.8	4.2	0.8	677.2	0.153	0.147	0.156
RC+rRegNN 70	4.2	4.2	0.8	676.8	0.146	0.140	0.156
RC+rRegNN 80	3.8	4.2	0.8	677.2	0.163	0.157	0.157
RC+rRegNN 90	3.0	4.2	0.8	678.0	0.179	0.174	0.156
RC+rRT 10	3.0	4.0	1.0	678.0	0.222	0.217	0.196
RC+rRT 20	3.0	4.0	1.0	678.0	0.222	0.217	0.196
RC+rRT 30	3.0	4.0	1.0	678.0	0.222	0.217	0.196
RC+rRT 40	3.0	4.0	1.0	678.0	0.222	0.217	0.196
RC+rRT 50	3.2	3.8	1.2	677.8	0.265	0.260	0.235
RC+rRT 60	2.6	3.2	1.8	678.4	0.382	0.378	0.356
RC+rRT 70	3.0	3.2	1.8	678.0	0.365	0.361	0.356
RC+rRT 80	7.4	3.2	1.8	673.6	0.252	0.245	0.349
RC+rRT 90	10.0	3.6	1.4	671.0	0.174	0.166	0.265
RC+rRT+AE 10	**1.6**	2.6	2.4	**679.4**	0.480	0.478	0.478
RC+rRT+AE 20	4.8	1.0	4.0	676.2	0.583	0.579	0.793
RC+rRT+AE 30	6.4	0.2	4.8	674.6	0.593	0.589	0.951
RC+rRT+AE 40	7.0	**0.0**	**5.0**	674.0	0.588	0.584	**0.990**
RC+rRT+AE 50	5.2	0.2	4.8	675.8	**0.644**	**0.641**	0.952
RC+rRT+AE 60	4.4	1.2	3.8	676.6	0.553	0.549	0.754
RC+rRT+AE 70	6.0	0.6	4.4	675.0	0.571	0.567	0.871
RC+rRT+AE 80	7.0	0.6	4.4	674.0	0.539	0.534	0.870
RC+rRT+AE 90	39.6	**0.0**	**5.0**	641.4	0.243	0.234	0.942

However, the highest F1 score we achieve in RC is lower at 0.644 as seen in Table 5.13. We do have the same pattern of increased performance from RC+rRegNN to RC+rRT to RC+rRT+AE, but the increase from RC+rRegNN to RC+rRT is not as dramatic as in previous results. Using RC+rRT+AE 40, we are able to achieve a perfect 5.0 TP

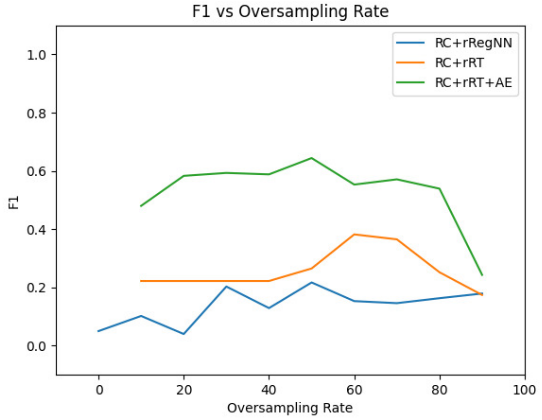


Figure 5.6: F1 score vs oversampling rate for RC+rRegNN, RC+rRT, and RC+rRT+AE techniques.

events which means we predicted them in all 5 runs, however the FP amount is even higher at 7.0 compared to rRT 80 when we last almost had a perfect predictor. The best F1 technique of RC+rRT+AE 50 is not much worse in terms of FN at 0.2 and drops to 5.2 FP events. This is still a lot more FP events than our best results so far in the rRT+AE 10 technique.

Figure 5.6 illustrates how the F1 score changes for the different RC techniques as the oversampling amount varies. Although the RC+rRT technique has a similar peak at 70% oversampling like the previous results, a new peak at 50% is present in the RC+rRegNN and RC+rRT+AE figures. The RC+rRT+AE also shows the same steep decline in F1 score at 90% oversampling as the rRT+AE 90 technique.

5.2.3.6 Learning Richardson Error (RE)

Using the RE technique performs even worse in terms of F1 than RC as seen in Table 5.15. Although we see the pattern of improvement from RE+rRegNN to RE+rRT to RE+rRT+AE, the highest F1 score is 0.545 with RE+rRT+AE 20. The RE+rRT techniques did learn very strong linear correlations to the actual ln peak intensity with

Table 5.14: Summarized regression metric results averaged over 5 runs for RE+rRegNN, RE+rRT, and RE+rRT+AE techniques with 3-fold Dataset 1. Formatting is the same as in Table 4.4.

Technique	PCC SEP	PCC SEP+Elevated	MAE SEP	MAE
RE+rRegNN 0	-0.512	0.464	2.454	**0.400**
RE+rRegNN 10	-0.559	0.514	2.405	0.442
RE+rRegNN 20	-0.409	0.538	2.591	0.456
RE+rRegNN 30	-0.498	0.428	2.735	0.466
RE+rRegNN 40	-0.587	0.510	2.596	0.475
RE+rRegNN 50	-0.397	0.438	2.533	0.493
RE+rRegNN 60	-0.157	0.636	2.307	0.481
RE+rRegNN 70	-0.539	0.153	2.864	0.473
RE+rRegNN 80	-0.364	0.427	2.939	0.472
RE+rRegNN 90	-0.123	0.654	2.278	0.494
RE+rRT 10	0.736	**0.866**	2.145	0.417
RE+rRT 20	0.130	0.760	1.912	0.443
RE+rRT 30	0.496	0.822	1.679	0.463
RE+rRT 40	0.653	0.832	1.584	0.504
RE+rRT 50	0.663	0.825	1.529	0.534
RE+rRT 60	0.694	0.829	1.480	0.570
RE+rRT 70	0.665	0.829	1.450	0.601
RE+rRT 80	0.729	0.841	**1.415**	0.657
RE+rRT 90	**0.892**	0.863	1.445	0.735
RE+rRT+AE 10	-0.582	0.579	2.166	0.413
RE+rRT+AE 20	-0.626	0.547	2.018	0.439
RE+rRT+AE 30	-0.609	0.557	1.946	0.460
RE+rRT+AE 40	-0.451	0.619	1.989	0.488
RE+rRT+AE 50	-0.385	0.643	1.938	0.525
RE+rRT+AE 60	-0.433	0.594	1.917	0.568
RE+rRT+AE 70	-0.386	0.637	2.144	0.530
RE+rRT+AE 80	-0.571	0.511	2.274	0.577
RE+rRT+AE 90	-0.153	0.572	2.005	0.566

peak values in RE+rRT 10 and RE+rRT 90. However, their F1 scores are much worse due to large amounts of FN or FP events. Like RC+rRT+AE 40, there are multiple oversample rates in RE+rRT that have all 5.0 TP events, but the FP events rise higher to 13.0 over the 7.0 achieved previously. Overall, the rRT+AE 10 still appears to be

Table 5.15: Summarized classifier adapted metric results averaged over 5 runs for RE+rRegNN, RE+rRT, and RE+rRT+AE techniques with 3-fold Dataset 1. Formatting is the same as in Table 4.4.

Technique	FP	FN	TP	TN	F1	HSS	TSS
RE+rRegNN 0	**2.4**	4.0	1.0	**678.6**	0.194	0.190	0.197
RE+rRegNN 10	4.2	3.6	1.4	676.8	0.257	0.252	0.275
RE+rRegNN 20	4.4	3.8	1.2	676.6	0.238	0.232	0.234
RE+rRegNN 30	4.8	4.0	1.0	676.2	0.193	0.187	0.195
RE+rRegNN 40	4.6	4.0	1.0	676.4	0.193	0.186	0.195
RE+rRegNN 50	4.8	3.6	1.4	676.2	0.249	0.243	0.273
RE+rRegNN 60	6.2	2.4	2.6	674.8	0.369	0.363	0.511
RE+rRegNN 70	6.2	3.6	1.4	674.8	0.231	0.224	0.271
RE+rRegNN 80	4.8	4.2	0.8	676.2	0.144	0.138	0.154
RE+rRegNN 90	5.0	3.4	1.6	676.0	0.283	0.277	0.313
RE+rRT 10	3.4	4.0	1.0	677.6	0.213	0.208	0.195
RE+rRT 20	5.0	1.8	3.2	676.0	0.479	0.475	0.633
RE+rRT 30	6.4	1.0	4.0	674.6	0.521	0.516	0.791
RE+rRT 40	8.6	1.0	4.0	672.4	0.456	0.450	0.787
RE+rRT 50	11.2	0.8	4.2	669.8	0.415	0.409	0.824
RE+rRT 60	13.0	**0.0**	**5.0**	668.0	0.435	0.429	**0.981**
RE+rRT 70	13.6	**0.0**	**5.0**	667.4	0.425	0.418	0.980
RE+rRT 80	21.2	**0.0**	**5.0**	659.8	0.321	0.312	0.969
RE+rRT 90	22.8	**0.0**	**5.0**	658.2	0.306	0.297	0.967
RE+rRT+AE 10	2.8	3.0	2.0	678.2	0.409	0.405	0.396
RE+rRT+AE 20	3.0	2.0	3.0	678.0	**0.545**	**0.542**	0.596
RE+rRT+AE 30	3.2	2.0	3.0	677.8	0.536	0.533	0.595
RE+rRT+AE 40	4.6	2.4	2.6	676.4	0.427	0.422	0.513
RE+rRT+AE 50	6.2	2.2	2.8	674.8	0.404	0.398	0.551
RE+rRT+AE 60	6.0	2.2	2.8	675.0	0.406	0.401	0.551
RE+rRT+AE 70	6.6	3.2	1.8	674.4	0.277	0.270	0.350
RE+rRT+AE 80	7.4	2.8	2.2	673.6	0.308	0.301	0.429
RE+rRT+AE 90	11.4	2.6	2.4	669.6	0.277	0.269	0.463

the best performer.

Figure 5.7 illustrates how the F1 score changes for the different RE techniques as the oversampling amount varies. In RE+rRegNN, there is a peak close to 70% as in the previous results. In the RE+rRT and RE+rRT+AE techniques, the peak has shifted

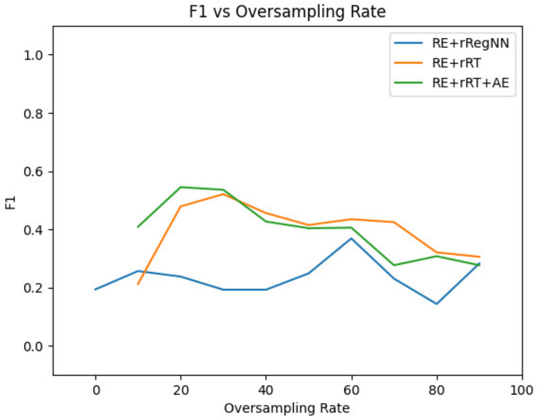


Figure 5.7: F1 score vs oversampling rate for RE+rRegNN, RE+rRT, and RE+rRT+AE techniques.

lower to 20% or 30%. The decline in F1 score as oversampling increases is similar to the results we saw in the rRT and rRT+AE techniques alone, but the F1 score around 70% oversampling is a minor peak if only because of the drop in F1 score at 80% oversampling.

5.2.3.7 DenseLoss (DL)

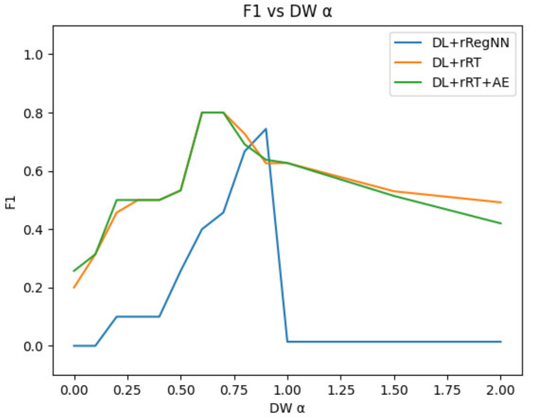


Figure 5.8: F1 score vs α_{DW} for DL+rRegNN, DL+rRT, and DL+rRT+AE techniques.

Table 5.16: Summarized regression metric results averaged over 5 runs for DL+rRegNN, DL+rRT, DL+rRT+AE techniques with 3-fold Dataset 1. Formatting is the same as in Table 4.4 except the trailing number is the α_{DW} instead of the oversampling percent.

Technique	PCC SEP	PCC SEP+Elevated	MAE SEP	MAE
DL+rRegNN 0.0	-0.083	0.701	2.670	0.067
DL+rRegNN 0.1	-0.054	0.700	2.645	0.069
DL+rRegNN 0.2	-0.064	0.712	2.494	0.069
DL+rRegNN 0.3	-0.053	0.712	2.441	0.072
DL+rRegNN 0.4	0.018	0.702	2.441	0.077
DL+rRegNN 0.5	0.024	0.696	2.352	0.080
DL+rRegNN 0.6	0.025	0.707	2.187	0.082
DL+rRegNN 0.7	0.053	0.701	2.083	0.093
DL+rRegNN 0.8	0.098	0.689	1.968	0.117
DL+rRegNN 0.9	0.083	0.708	1.726	0.150
DL+rRegNN 1.0	-0.104	0.097	1.770	2.130
DL+rRegNN 1.5	-0.054	0.420	1.614	2.281
DL+rRegNN 2.0	-0.160	0.127	1.574	2.324
DL+rRT 0.0	-0.062	0.705	2.371	**0.065**
DL+rRT 0.1	-0.062	0.707	2.326	0.066
DL+rRT 0.2	-0.062	0.707	2.269	0.067
DL+rRT 0.3	-0.061	0.707	2.209	0.068
DL+rRT 0.4	-0.060	0.709	2.144	0.070
DL+rRT 0.5	0.032	0.734	2.015	0.077
DL+rRT 0.6	0.121	0.752	1.905	0.081
DL+rRT 0.7	**0.153**	0.757	1.793	0.086
DL+rRT 0.8	0.142	0.755	1.710	0.090
DL+rRT 0.9	0.061	0.734	1.649	0.107
DL+rRT 1.0	-0.065	0.708	1.544	1.512
DL+rRT 1.5	-0.065	0.709	1.501	1.701
DL+rRT 2.0	-0.065	0.708	**1.487**	1.794
DL+rRT+AE 0.0	-0.066	0.706	2.359	**0.065**
DL+rRT+AE 0.1	-0.066	0.707	2.312	0.066
DL+rRT+AE 0.2	-0.066	0.708	2.261	0.067
DL+rRT+AE 0.3	-0.062	0.711	2.209	0.069
DL+rRT+AE 0.4	-0.043	0.718	2.143	0.072
DL+rRT+AE 0.5	0.064	0.743	2.015	0.077
DL+rRT+AE 0.6	0.152	**0.760**	1.912	0.081
DL+rRT+AE 0.7	0.099	0.750	1.801	0.084
DL+rRT+AE 0.8	0.131	0.753	1.730	0.093
DL+rRT+AE 0.9	0.073	0.739	1.658	0.106
DL+rRT+AE 1.0	-0.068	0.708	1.543	1.512
DL+rRT+AE 1.5	-0.068	0.710	1.503	1.741
DL+rRT+AE 2.0	-0.068	0.710	1.489	1.838

Table 5.17: Summarized classifier adapted metric results averaged over 5 runs for DL+rRegNN, DL+rRT, and DL+rRT+AE techniques with 3-fold Dataset 1. Formatting is the same as in Table 5.16.

Technique	FP	FN	TP	TN	F1	HSS	TSS
DL+rRegNN 0.0	**0.4**	5.0	0.0	**680.6**	0.000	-0.001	0.000
DL+rRegNN 0.1	0.6	5.0	0.0	680.4	0.000	-0.001	0.000
DL+rRegNN 0.2	0.8	4.6	0.4	680.2	0.100	0.098	0.080
DL+rRegNN 0.3	0.8	4.6	0.4	680.2	0.100	0.098	0.080
DL+rRegNN 0.4	0.8	4.6	0.4	680.2	0.100	0.098	0.080
DL+rRegNN 0.5	0.8	4.0	1.0	680.2	0.257	0.255	0.199
DL+rRegNN 0.6	1.0	3.4	1.6	680.0	0.400	0.397	0.319
DL+rRegNN 0.7	1.0	3.2	1.8	680.0	0.457	0.454	0.359
DL+rRegNN 0.8	1.0	2.0	3.0	680.0	0.667	0.664	0.599
DL+rRegNN 0.9	1.8	1.0	4.0	679.2	0.744	0.742	0.797
DL+rRegNN 1.0	681.0	**0.0**	**5.0**	0.0	0.014	0.000	0.000
DL+rRegNN 1.5	681.0	**0.0**	**5.0**	0.0	0.014	0.000	0.000
DL+rRegNN 2.0	681.0	**0.0**	**5.0**	0.0	0.014	0.000	0.000
DL+rRT 0.0	1.0	4.2	0.8	680.0	0.200	0.197	0.159
DL+rRT 0.1	1.0	3.8	1.2	680.0	0.314	0.312	0.239
DL+rRT 0.2	1.0	3.2	1.8	680.0	0.457	0.454	0.359
DL+rRT 0.3	1.0	3.0	2.0	680.0	0.500	0.497	0.399
DL+rRT 0.4	1.0	3.0	2.0	680.0	0.500	0.497	0.399
DL+rRT 0.5	1.0	2.8	2.2	680.0	0.533	0.531	0.439
DL+rRT 0.6	1.0	1.0	4.0	680.0	**0.800**	**0.799**	0.799
DL+rRT 0.7	1.0	1.0	4.0	680.0	**0.800**	**0.799**	0.799
DL+rRT 0.8	2.0	1.0	4.0	679.0	0.727	0.725	0.797
DL+rRT 0.9	3.8	1.0	4.0	677.2	0.626	0.622	0.794
DL+rRT 1.0	4.6	0.6	4.4	676.4	0.627	0.624	0.873
DL+rRT 1.5	9.0	**0.0**	**5.0**	672.0	0.530	0.525	**0.987**
DL+rRT 2.0	10.4	**0.0**	**5.0**	670.6	0.492	0.486	0.985
DL+rRT+AE 0.0	1.0	4.0	1.0	680.0	0.257	0.254	0.199
DL+rRT+AE 0.1	1.0	3.8	1.2	680.0	0.314	0.312	0.239
DL+rRT+AE 0.2	1.0	3.0	2.0	680.0	0.500	0.497	0.399
DL+rRT+AE 0.3	1.0	3.0	2.0	680.0	0.500	0.497	0.399
DL+rRT+AE 0.4	1.0	3.0	2.0	680.0	0.500	0.497	0.399
DL+rRT+AE 0.5	1.0	2.8	2.2	680.0	0.533	0.531	0.439
DL+rRT+AE 0.6	1.0	1.0	4.0	680.0	**0.800**	**0.799**	0.799
DL+rRT+AE 0.7	1.0	1.0	4.0	680.0	**0.800**	**0.799**	0.799
DL+rRT+AE 0.8	2.6	1.0	4.0	678.4	0.691	0.688	0.796
DL+rRT+AE 0.9	3.6	1.0	4.0	677.4	0.638	0.635	0.795
DL+rRT+AE 1.0	4.6	0.6	4.4	676.4	0.627	0.624	0.873
DL+rRT+AE 1.5	9.6	**0.0**	**5.0**	671.4	0.514	0.508	0.986
DL+rRT+AE 2.0	14.0	**0.0**	**5.0**	667.0	0.420	0.414	0.979

The best F1 score in the DenseLoss+rRegNN technique outperforms the rRT+AE 10, and it just gets better through the other techniques as seen in Table 5.17. By applying DL+rRT with α_{DW} 0.6 or 0.7, we reduce the FP count down to 1.0 keeping the FN at 1.0. These are the same FP and FN count in the best classifier technique. In the analysis, we will examine the events to see if they are the same FP and FN events we had in the classifier analysis. Adding the autoencoder in DL+rRT+AE does not gain much improvement. The DL+rRT 1.5 with the 5.0 TP events has more FP events than RC+rRT+AE 40, therefore RC+rRT+AE 40 remains the best F1 score for 5.0 TP events.

There are suspicious looking results when examining DL+rRegNN 1.0, 1.5, and 2.0. The F1 score was steadily increasing as the α_{DW} was increasing before it suddenly fell. These results are caused by the DenseWeight construction though the effect is not as drastic in the other DL techniques. When α_{DW} has a value of 1.0, the weight of the background events falls to a very small constant. In our batches, only about 2 events were SEP or Elevated events, and their weights were higher. However, they were drowned out by the many background events whose values were very small because their weights were very small. The DenseLoss function adds all the values up and divides them by the number of elements. With many of those values close to 0, the resulting loss was much smaller than it was for smaller α_{DW} values. We ran an adjusted version of DenseLoss where we divided by the sum of the weights instead of the number of elements in the dataset. While this did improve the performance of the DL techniques with higher α_{DW} values, they were still less performant than the lower α_{DW} values, and the remaining α_{DW} values showed similar performance to the original implementation. Therefore, we continue to present the original DL implementation and results.

Figure 5.8 illustrates how the F1 score changes for the different DL techniques as the α_{DW} increases. Like the oversampling, the higher α_{DW} represents higher emphasis on SEP events, so we expected increasing performance as α_{DW} increased. The DL+rRegNN technique had positive correlation of α_{DW} and F1 until 1.0 when there is a steep decrease to a flat 0.0 in F1. The best performing DL+rRT and DL+rRT+AE have peak F1 at α_{DW} 0.6 and 0.7. Due to the decreasing F1 on either side of these α_{DW} values, it could be there is a higher peak when using an α_{DW} in-between 0.6 and 0.7, but we leave that for future work.

5.2.3.8 Comparing the Main Approaches

Table 5.18: Summarized regression metric results averaged over 5 runs for the best F1 across all techniques with 3-fold Dataset 1. Each row in a technique with either an oversampling rate or an α_{DW} indicated by the trailing number after the row technique name. The best value for each metric across all techniques is in bold.

Technique	PCC SEP	PCC SEP+Elevated	MAE SEP	MAE
rRegNN 70	-0.258	0.632	2.161	0.126
rRT 10	0.143	0.751	1.857	0.104
rRT+AE 10	0.137	0.753	1.821	0.103
RC+rRegNN 50	0.185	0.731	2.584	0.229
RC+rRT 60	-0.704	0.253	2.423	0.598
RC+rRT+AE 50	-0.183	0.686	**1.571**	0.354
RE+rRegNN 60	-0.157	0.636	2.307	0.481
RE+rRT 30	**0.496**	**0.822**	1.679	0.463
RE+rRT+AE 20	-0.626	0.547	2.018	0.439
DL+rRegNN 0.9	0.083	0.708	1.726	0.150
DL+rRT 0.7	0.153	0.757	1.793	0.086
DL+rRT+AE 0.6	0.152	0.760	1.912	**0.081**

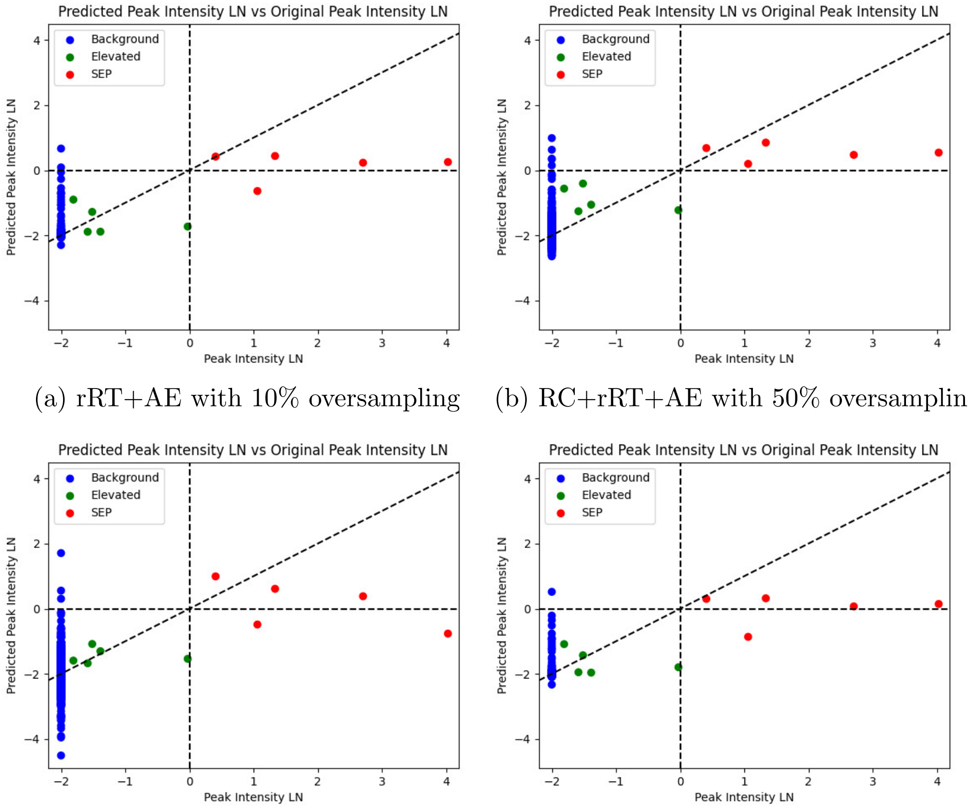
In comparing the rows of Table 5.18, RE+rRT 30 has the best PCC in both the SEP only and SEP+Elevated datasets. In terms of the SEP events alone, the PCC SEP in this technique is much higher than others with only a small increase in MAE SEP over the optimal RC+rRT+AE 50. Comparatively when we add the Elevated events, the PCC SEP+Elevated improvement is less impressive with the next highest value in DL+rRT+AE 0.6. The improved values of PCC in RE+rRT 30 do not guarantee low MAE error as its MAE error is one of the highest, 0.463, compared to the minimum in

Table 5.19: Summarized classifier adapted metric results averaged over 5 runs for the best F1 across all techniques with 3-fold Dataset 1. Formatting is the same as in Table 5.18.

Technique	FP	FN	TP	TN	F1	HSS	TSS
rRegNN 70	2.4	2.2	2.8	678.6	0.545	0.542	0.556
rRT 10	1.6	1.2	3.8	679.4	0.730	0.728	0.758
rRT+AE 10	1.8	1.0	4.0	679.2	0.742	0.740	0.797
RC+rRegNN 50	3.8	3.8	1.2	677.2	0.217	0.211	0.235
RC+rRT 60	2.6	3.2	1.8	678.4	0.382	0.378	0.356
RC+rRT+AE 50	5.2	**0.2**	**4.8**	675.8	0.644	0.641	**0.952**
RE+rRegNN 60	6.2	2.4	2.6	674.8	0.369	0.363	0.511
RE+rRT 30	6.4	1.0	4.0	674.6	0.521	0.516	0.791
RE+rRT+AE 20	3.0	2.0	3.0	678.0	0.545	0.542	0.596
DL+rRegNN 0.9	1.8	1.0	4.0	679.2	0.744	0.742	0.797
DL+rRT 0.7	**1.0**	1.0	4.0	**680.0**	**0.800**	**0.799**	0.799
DL+rRT+AE 0.6	**1.0**	1.0	4.0	**680.0**	**0.800**	**0.799**	0.799

DL+rRT+AE 0.6 of 0.081. We expect that RC+rRT+AE 50 and DL+rRT+AE 0.6 should have higher classifier statistics as their MAE errors are lower.

When comparing the best F1 scores across the various techniques in Table 5.19, the DL+rRT 0.7 and DL+rRT+AE 0.6 almost completely dominate the highest metrics. The one exception is the TSS high score in RC+rRT+AE 50 which almost manages to perfectly predict the TP events at the cost of 4.2 more FP events. This metric is unsurprising considering it had the lowest MAE SEP. The RE+rRT 30 has the most FP events across all techniques which reduces its performance considerably. It does do fairly well on the TP events, but its high value of PCC SEP may not be the cause. As evidence, the RC+rRT+AE 50 had an even higher TP count but much worse PCC SEP. Across all four groups, F1 generally increases from the respective rRegNN to rRT culminating with rRT+AE. Therefore, two stage training is outperforming joint training with the addition of the autoencoder usually further improving performance.



(a) rRT+AE with 10% oversampling (b) RC+rRT+AE with 50% oversampling

(c) RE+rRT+AE with 20% oversampling (d) DL+rRT+AE with 0.6 α_{DW}

Figure 5.9: Predicted vs Actual peak intensity for rRT+AE, RC+rRT+AE, RE+rRT+AE, and DL+rRT+AE. Perfect intensity predictions are on the diagonal dotted line. The horizontal and vertical dotted lines indicate the threshold for classification. FPs and FNs are in the upper left and lower right quadrants respectively.

5.2.4 Analysis

Figure 5.9 allows for easy comparison of the top F1 performing models with plots comparing their predicted output vs the actual ln peak intensity. The figures also help illustrate the results we found when comparing these main approaches. Threshold lines separate FPs in the upper left quadrant and FNs in the lower right quadrant. In comparison to the other plots, the distribution in rRT+AE 10 is unsurprisingly fairly like the DL+rRT+AE 0.6 when you compare the rRT+AE 10 model's slightly lower F1 score of 0.742 to its highest 0.800 F1 score. The rRT+AE 10 run shows an extra

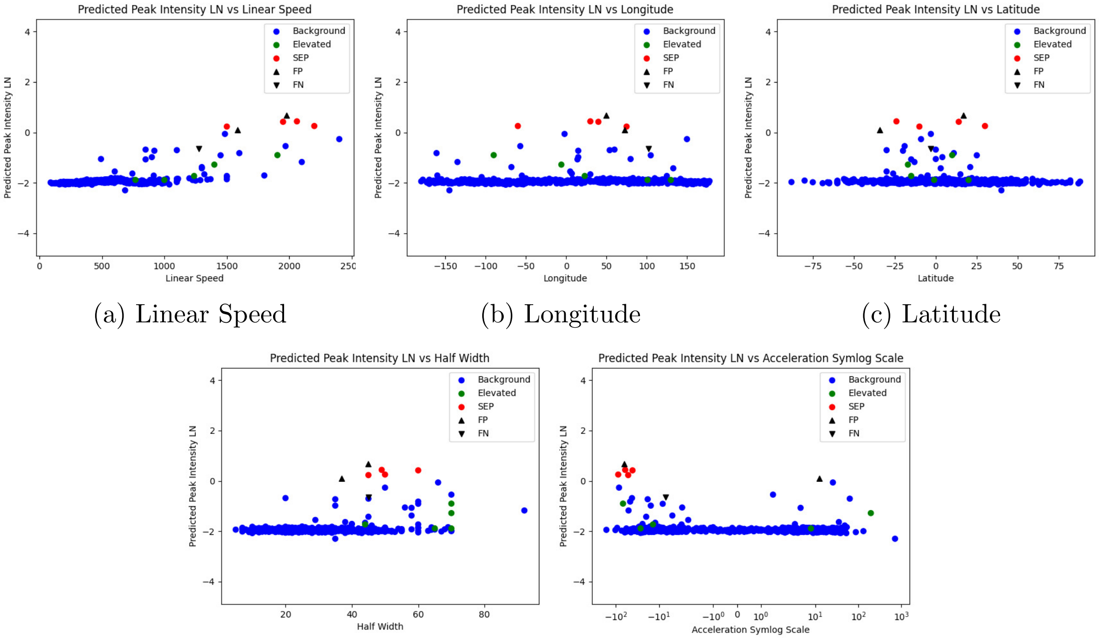
FP just above the threshold line which causes the reduced F1 score. The MAE on SEP events in rRT+AE 10 is just a bit smaller than it is in DL+rRT+AE 0.6 which we can spot in the very slightly increased predictions of the SEP events, but it is not very significant. The lowest MAE error on the SEP events was in RC+rRT+AE 50. All of the SEP events were predicted higher in comparison to rRT+AE 10 and DL+rRT+AE 0.6 moving them closer to the perfect prediction diagonal line. The most improvement was made in the lower intensity SEP events with only minor movements in the higher peak intensity events. The DL+rRT+AE 0.6 does not get all the TP events as the RC+rRT+AE 50 model does, but it seems clearer from the RC+rRT+AE plot that it is over-predicting most samples which allows it to capture the extra TP with the trade-off of more FP events. Not only does it have the extra TP and FP events, but the elevated events also have a higher prediction although they are still correctly classified. The RE+rRT+AE 20 has comparable performance on the elevated events as DL+rRT+AE 0.6, but the 2 extra FP and 1 extra FN event reduce its F1 performance considerably.

5.2.4.1 Regression Re-training with Autoencoder (rRT+AE)

Table 5.20: The important features for the False Positive (FP) and False Negative (FN) events in rRT+AE 10 visualized in Figure 5.9a. The features are given across the top in abbreviated format: DONKI Date (DD), Latitude (Lat), Longitude (Lon), Linear Speed (Spd), Half Width (Hw), Acceleration (Acc), Actual 100 MeV Peak Intensity ln (Peak ln), Predicted 100 MeV Peak Intensity ln (Pred), and the classification (FP/FN).

DD	Lat	Lon	Spd	Hw	Acc	Peak ln	Pred	FP/FN
3/7/2011 20:12	17	50	1980	45	-63.1	-2.000	0.665	FP
1/1/2016 23:12	-34	73	1588	37	12.7	-2.000	0.087	FP
1/6/2014 8:09	-3	102	1275	45	-7.1	1.061	-0.639	FN

Table 5.20 lists the important features for the rRT+AE 10 technique. We tabulate the same features as we did in the classifier analysis namely: latitude, longitude, linear speed, half width, and acceleration. To provide context to these values, we plot each



(a) Linear Speed

(b) Longitude

(c) Latitude

(d) Half Width

(e) Acceleration with symlog scale

Figure 5.10: Predicted ln peak intensity vs 5 features for rRT+AE with 10% oversampling.

feature against the predicted ln peak intensity in Figure 5.10.

Similar to our prior analysis of Linear speed vs classifier score, there appears to be a positive correlation between linear speed and predicted ln peak intensity in Figure 5.10a. However, the slope of the correlation line seems shallow due to the slightly elevated ln peak intensity of TN events around 1000 and 1250. The speed does seem to be contributing to higher predictions since the TP and FP events all have higher predicted values and higher speeds vs the FN event with a lower speed and lower prediction. Other TN background and elevated events also seem to have higher predictions with higher speeds which supports the positive correlation between linear speed and prediction.

There is a less definitive shape in the longitude feature portrayed in Figure 5.10b. We previously saw a peak around 57 degrees, and most of the TP and FP events which have higher predictions are close to that longitude. However, there is an outlier TP event that has a longitude around -50 with a high predicted value. This outlier event may be the cause of the variable increased predictions of some of the TN events around 0 and -50 over other TN events. The FN event has a larger longitude and lower prediction like most other TN events with larger longitudes. Its higher longitude value may be causing its prediction to be lower.

The peak we previously saw in the latitude feature was around 0. In Figure 5.10c, the peak seems to have flattened out to include the range of latitude values from -25 to 25. The cause of this spread may be due to the TP SEP events whose latitude values vary across this range. The FP event on the lower end of the range past the smallest TP event's latitude value has a lower predicted value than the other FP event whose latitude value is more favorable. The FN event has a more favorable latitude which may be increasing its prediction above most of the other background events, but there are exceptions. Other features are contributing to its lower prediction since there are TN events with a higher prediction and similar latitude values.

The half width feature exhibits a similar transition from fixed peak to a more general range of values that correspond to higher predictions illustrated in Figure 5.10d. In this case, the range seems to be from just under 40 to about 75 with steep looking drops in prediction on each side. The steeper looking drops are due to events sharing the same half width such as the two elevated and one background event around 75. Clearly, other features are contributing to their varied predictions. The half width of the FP and TP event are also equal, supporting the conclusion that other features are playing a major role in determining their prediction values given their disparity. Like the latitude event, the high predicted FP event is within the range of the other TP events while the other lower predicted FP event is outside. However, that outside FP is within the range of half widths we identified corresponded to increased predicted value which may have increased its prediction above the other TN events.

We see familiar twin peaks in the acceleration feature exhibited in Figure 5.10e as we have identified previously in the classifier analysis. Again, the highly predicted FP event is right within the range of the other TP events, likely making it difficult to distinguish. The other FP event along with some higher predicted TN events form the secondary symmetrically opposite positive peak around 10^2 in symlog scale. In comparison to the TP events, its acceleration is not very favorable which may be contributing to its lower relative prediction to the other FP event. The FN event is much more favorable in comparison, but its almost obscured by TN events with higher predictions. Some of the TN events between the FN and TP events have higher predictions which may have been influenced by the distance between them in their acceleration feature despite both having a high actual ln peak intensity.

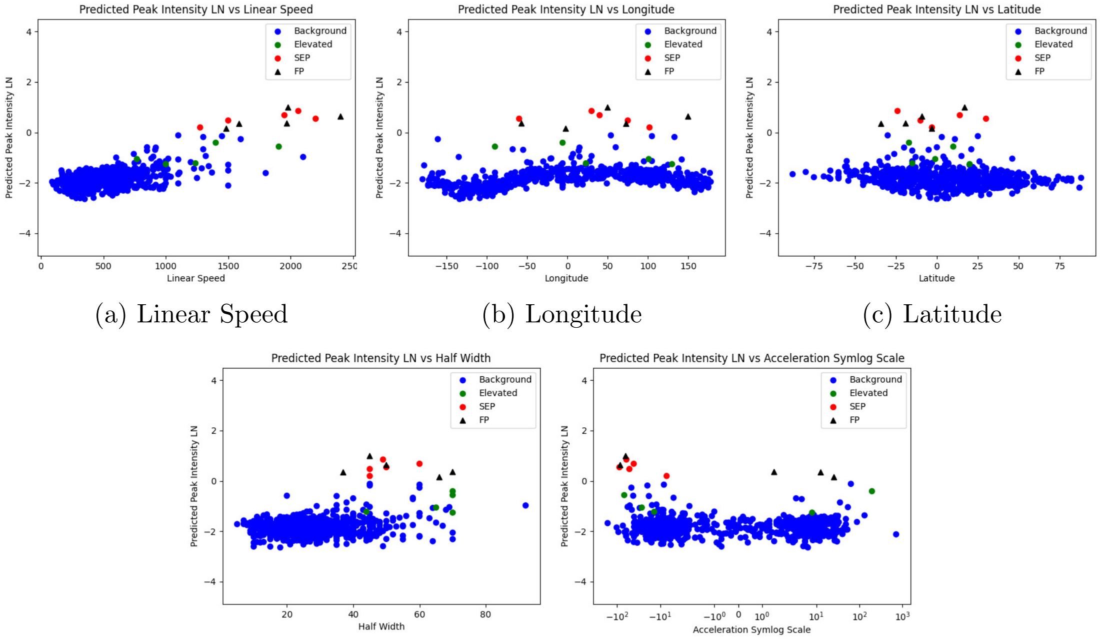
5.2.4.2 Combining Richardson Forecast (RC)

Table 5.21: The important features for the False Positive (FP) and False Negative (FN) events in RC+rRT+AE 50 visualized in Figure 5.9b. Acronyms are the same as in Table 5.20.

DD	Lat	Lon	Spd	Hw	Acc	Peak ln	Pred	FP/FN
3/7/2011 20:12	17	50	1980	45	-63.1	-2.000	0.987	FP
12/13/2014 14:24	-9	150	2400	50	-84.4	-2.000	0.628	FP
6/21/2013 3:24	-19	-57	1970	70	1.5	-2.000	0.354	FP
1/1/2016 23:12	-34	73	1588	37	12.7	-2.000	0.339	FP
3/15/2013 6:54	-3	-2	1485	66	25.8	-2.000	0.147	FP

Table 5.21 and Figure 5.11 capture the feature details for the FP and FN events for RC+rRT+AE 50. Although this technique had a lower F1 score than rRT+AE 10, this run perfectly classified all SEP events as TP. This came at the cost of more FP events. Analyzing the cause of the FP events may help eliminate them in the future.

There is a strong correlation between linear speed and prediction in Figure 5.11a.



(a) Linear Speed (b) Longitude (c) Latitude

(d) Half Width (e) Acceleration with symlog scale

Figure 5.11: Predicted ln peak intensity vs 5 features for RC+rRT+AE with 50% oversampling.

The prediction on all events generally increases as the speed increases. Some of the top predictions are with the events with top speeds though there does seem to be a drop in prediction with the few largest speed value.

The spread in the longitude feature we saw previously seems more pronounced in Figure 5.11b due to the higher predictions in the extra FP events. While the prediction of the previously FN event in rRT+AE 10 has increased, several of the TN predictions around it have also increased. Increasing its performance may have required putting more weight for higher longitude values. This seems to be the case for one of the higher predicted FP events which has a much larger longitude value. In analyzing the TP events, there still seems to be a peak around 50 degrees.

Figure 5.11c shows higher predictions for latitudes in the range of -25 to 25 degrees as we saw in rRT+AE 10. Four of the five FP events are within that range of values with three of them with almost the same latitude and prediction as one of the TP events. Their similarity in latitude values and subsequent prediction values suggests that their latitude feature contributed to their incorrect classification.

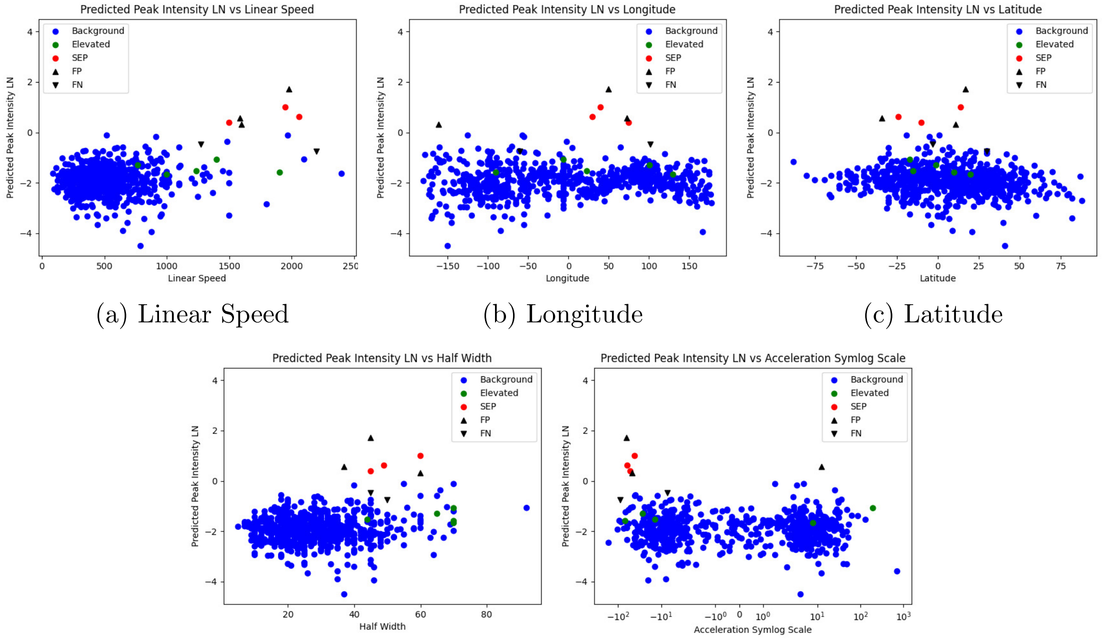
As we saw in rRT+AE 10, several events share the same half width but have varied prediction values. This forms vertical lines in Figure 5.11d around 50, 60, and 70. Clearly, there are other features contributing to their prediction because if the half width dominated their prediction they would be on top of each other. The higher predicted FP events share half widths with the TP events likely making them harder to distinguish. There doesn't seem to be a strong peak or correlation due to the varied half widths in the TP and FP events.

Figure 5.11e continues to have twin peaks around 10^{-2} and 10^{2}. The additional FP events are generally close to one of the two peaks. The FP on the left peak have close to the same value and prediction in the symlog scale likely increasing their prediction because they cannot be distinguished by acceleration alone. The other FP events at the right peak make it more pronounced along with some higher predicted TN background and elevated events.

5.2.4.3 Learning Richardson Error (RE)

Table 5.22: The important features for the False Positive (FP) and False Negative (FN) events in RE+rRT+AE 20 visualized in Figure 5.9c. Acronyms are the same as in Table 5.20.

DD	Lat	Lon	Spd	Hw	Acc	Peak ln	Pred	FP/FN
3/7/2011 20:12	17	50	1980	45	-63.1	-2.000	1.708	FP
1/1/2016 23:12	-34	73	1588	37	12.7	-2.000	0.556	FP
3/24/2012 0:39	11	-161	1600	60	-46.6	-2.000	0.306	FP
1/6/2014 8:09	-3	102	1275	45	-7.1	1.061	-0.485	FN
3/7/2012 0:36	30	-60	2200	50	-88.2	4.031	-0.764	FN



(a) Linear Speed (b) Longitude (c) Latitude

(d) Half Width (e) Acceleration with symlog scale

Figure 5.12: Predicted ln peak intensity vs 5 features for RE+rRT+AE with 20% oversampling.

Table 5.22 and Figure 5.12 present the important features for RE+rRT+AE 20. This technique has the lowest F1 compared to the top F1 scores from the other techniques. Its main differential is an additional FN event not found in the other top techniques.

There still appears to be a linear correlation in Figure 5.12a between speed and prediction, but it is much weaker than observed previously. The predictions are more varied in the TN events with lower speed values which weakens the correlation. The TP and FP events with higher speeds continue to have higher predictions generally, but there is a high-speed FN exception which has low prediction. This is a more emphasized case of the trend we saw in RC+rRT+AE 50 where the prediction was higher as the speed got higher until it dropped off for the highest speed events. This suggests there might be another feature that these very high-speed events share that decrease their prediction.

There is also variance in the predictions across the range of longitude values in Figure 5.12b. The previous TP SEP event at longitude -60 blurs into the background events as its prediction has fallen and the TN prediction has risen. Although the three TP and 2 FP events have high predictions and longitude values around 57, the background TN events have more and more variance in their prediction as the longitude decreases with both the highest and lowest predictions for the TN events close to the limit of the Longitude plot. This makes it difficult to assess if there is still a peak around 57 degrees. With the higher predictions of the TN events and the outlier FP event with longitude -161, the presence of the peak is less emphasized. If 57 is favorable, then the outlier FP event mentioned must have other features increasing its prediction. Its longitude value may be contributing to the fact that it's the lowest predicted FP event.

The variance of the TN events in the latitude on the other hand seems to emphasize the peak around 0 in Figure 5.12c. Several TN events have increased predictions around 0 degrees and predictions decrease on either side. All three FP events have latitude values close to 0 which is likely increasing their prediction. The previous TP close to 25 latitude has again dropped into the background events like the longitude feature. Its prediction is lower than the other FN we have seen before which has a more favorable latitude close to 0.

The linear lines seen previously in the half width feature seem less obvious in Figure 5.12d. This is seems caused by the variation in the TN events, the extra FP events, and the new FN event. The new FN event previously had a prediction very similar to the TP with a latitude close to 50. Since it has a lower prediction value and the other SEP event with a similar latitude is a TP, it seems less likely that the half width feature is contributing to the FN low prediction.

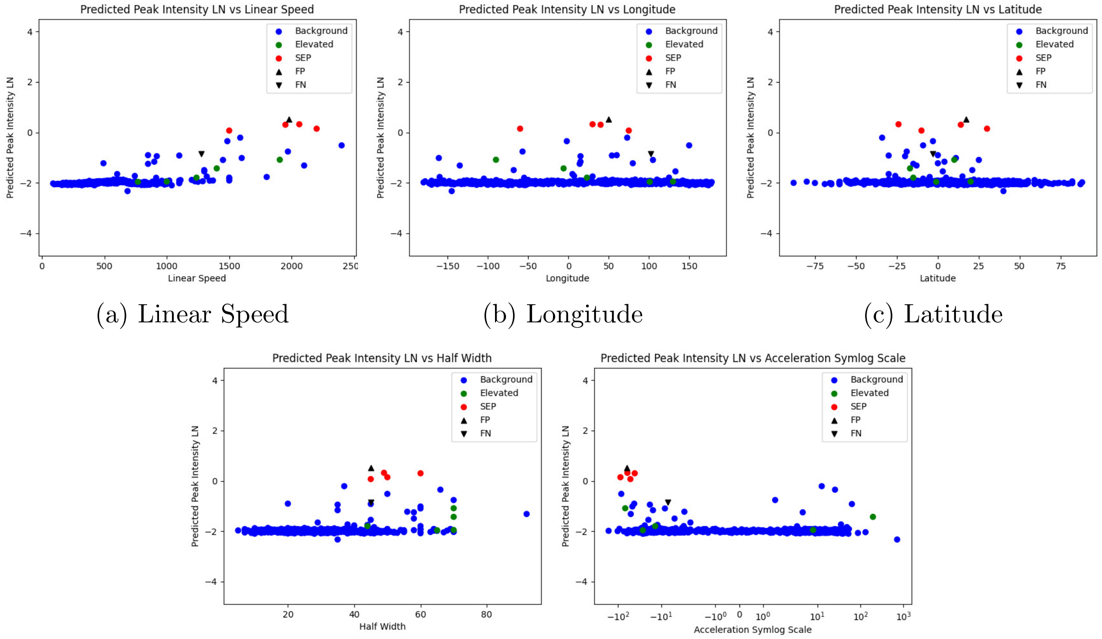
Figure 5.12e reveals the high concentration of events with either a quite positive or quite negative acceleration feature with fewer events in between. There still seems to be a peak around -10^2 with the secondary peak around 10^2. Several high predictions in the TN and 1 of the FP events make the secondary peak on the right more distinct. The other two FP events have acceleration values similar to the other TP events which is likely contributing to their higher predictions. The two FN fall on either side at -7.1 and -88.2. We might expect the FN to the left of the peak to have a higher prediction since it has a closer acceleration to the other SEP events, but its prediction is a little less than the other FN's prediction. This suggests that other features are contributing to its lower prediction.

5.2.4.4 DenseLoss (DL)

Table 5.23: The important features for the False Positive (FP) and False Negative (FN) events in DL+rRT+AE 0.6 visualized in Figure 5.9d. Acronyms are the same as in Table 5.20.

DD	Lat	Lon	Spd	Hw	Acc	Peak ln	Pred	FP/FN
3/7/2011 20:12	17	50	1980	45	-63.1	-2.000	0.519	FP
1/6/2014 8:09	-3	102	1275	45	-7.1	1.061	-0.863	FN

Table 5.23 and Figure 5.13 list and illustrate the features for DL+rRT+AE 0.6, respectively. Upon inspection, we can see that not only did these events persist as errors in all the regression approaches, but these are the same two events as we had in our best classifier technique. The feature plots for DL+rRT+AE 0.6 almost exactly match with the feature plots of rRT+AE 10 illustrated in Figure 5.10. The only marked difference is correctly classifying its extra FP event as a TN. Comparing these 5 feature plots between the two techniques does not reveal the cause of this improvement. The plots of all the other features also show almost identical distributions for each particular



(a) Linear Speed (b) Longitude (c) Latitude

(d) Half Width (e) Acceleration with symlog scale

Figure 5.13: Predicted ln peak intensity vs 5 features for DL+rRT+AE with 0.6 α_{DW}.

feature with very minor variations in a few choice events.

5.2.4.4.1 Feature Importance

Table 5.24 shows the overall feature importance values for the DL+rRT+AE 0.6 model. The feature importance values and ranks are very similar to our previous analysis of the cRT+AE 70 model. Some features have moved especially in the some of the mid to lower ranked features such as Halo falling to rank 18. The top features are almost the same with the same top speed and location features.

Table 5.25 shows the FP and FN local feature importance and contribution values. As we saw in the cRT+AE 70 FP event, the V Log V, Type II Visualization Area, and Linear Speed have high contributions. In the FN event, the Diffusive Shock, 2nd order speed at 20 solar radii, and Latitude have high negative contributions. Compared to the cRT+AE 70 analysis, we have seen the Diffusive Shock and Latitude features but

Table 5.24: Overall Feature importance values ($\hat{I}_j$) for DL+rRT+AE 0.6.

Feature (j)	Importance ($\hat{I}_j$)
1. V Log V	0.128
2. Diffusive shock	0.125
3. Linear Speed	0.093
4. Richardson's equation	0.082
5. 2nd order speed at 20 solar radii	0.075
6. 2nd order speed final	0.075
7. CMEs over 1000 km/s past 9 hrs	0.053
8. Max speed past day	0.051
9. Type II Visualization Area	0.049
10. Longitude	0.036
11. CMEs in past month	0.034
12. Half Width	0.031
13. Daily Sunspot Count	0.030
14. MPA	0.028
15. Acceleration	0.027
16. Latitude	0.027
17. CMEs in past 9 hours	0.023
18. Halo	0.020
19. CPA	0.012

the 2nd order speed at 20 solar radii is new although also negative in the previous feature importance analysis.

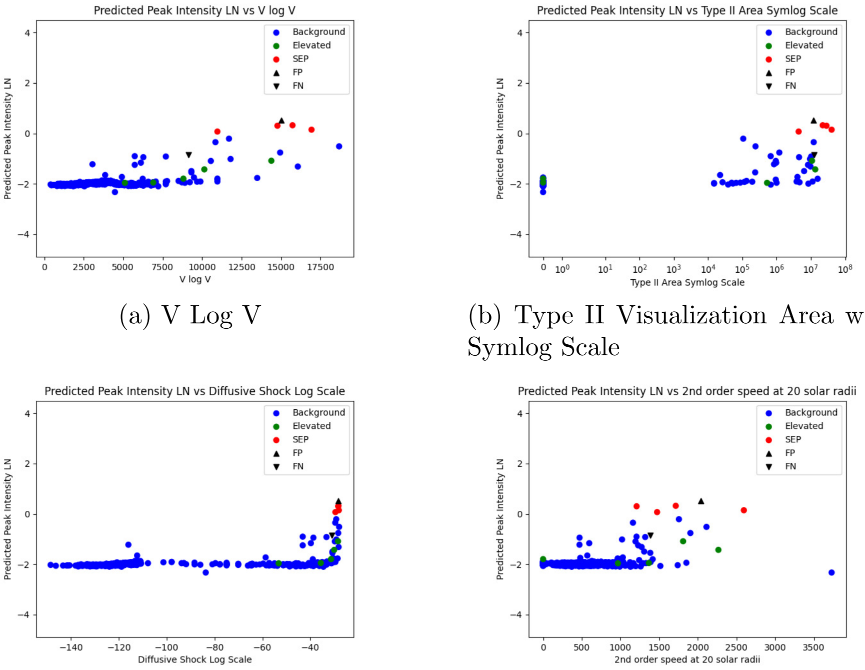
Figure 5.14 illustrates the features we have not previously analyzed for the DL+rRT+AE 0.6 model contributing to the FP and FN local importance values. Analysis of the V Log V feature in Figure 5.14a reveals the same positive correlation we saw for the Linear Speed feature already analyzed. There is a positive correlation overall and locally around the FP and FN feature values. The Type II Visualization Area in Figure 5.14b looks similar to its counterpart in the cRT+AE 70 analysis in Figure 4.10a. The predictions are more compact than the scores were, but there still seems to be a small positive correlation. We already saw the strong positive correlation in Linear Speed previously in Figure 5.13a.

Table 5.25: Feature importance values from LIME for FP and FN in Table 5.23 with DL+rRT+AE 0.6. This Table has the same format as in Table 4.12.

Feature (j)	FP				FN			
(ordered by $\hat{I}_j$ as in Table 5.24)	$R_{W_{ij}}$	W_{ij}	$R_{C_{ij}}$	C_{ij}	$R_{W_{ij}}$	W_{ij}	$R_{C_{ij}}$	C_{ij}
V Log V	**1**	**0.140**	**1**	**93.895**	1	0.088	2	35.526
Diffusive shock	18	-0.032	19	-32.086	**19**	**-0.064**	**19**	**-62.507**
Linear Speed	*2*	*0.105*	3	73.245	2	0.053	5	23.387
Richardson's equation	6	0.063	5	57.069	5	0.038	4	23.738
2nd order speed at 20 solar radii	7	0.041	8	22.524	*18*	*-0.016*	*18*	*-5.869*
2nd order speed final	3	0.091	6	55.096	3	0.039	6	16.957
CMEs over 1000 km/s past 9 hrs	13	0.009	14	0.000	7	0.014	12	0.000
Max speed past day	15	-0.001	15	-0.284	17	-0.015	15	-1.761
Type II Visualization Area	4	0.080	*2*	*73.882*	6	0.031	3	28.790
Longitude	16	-0.003	16	-1.802	13	-0.002	16	-1.841
CMEs in past month	17	-0.025	17	-2.228	14	-0.003	14	-1.670
Half Width	11	0.019	11	8.766	11	0.001	10	0.561
Daily Sunspot Count	10	0.020	10	13.613	8	0.012	7	10.040
MPA	8	0.023	9	19.943	9	0.008	9	6.151
Acceleration	19	-0.040	18	-7.369	15	-0.005	13	-1.120
Latitude	14	0.004	13	2.213	16	-0.005	17	-2.214
CMEs in past 9 hours	12	0.018	12	4.406	12	-0.001	11	-0.000
Halo	5	0.064	4	63.944	4	0.038	1	38.141
CPA	9	0.023	7	22.555	10	0.007	8	7.075

Figure 5.14c depicts the prediction of events against the Diffusive Shock feature. The global trend for this feature is positive. The Diffusive Shock feature of the FN event is high, but it is less than the other SEP events. Locally to the FN event, most of the events are background events which also have high feature values (about 0.92 normalized) but their actual ln intensity is about -2. Their proximity might explain why the local linear model in LIME has a negative importance for Diffusive Shock. The relatively large negative contribution C_{ij} of this feature to the prediction is caused by the high feature value X_{ij} and negative importance W_{ij}. Figure 5.14d depicts the 2nd order speed at 20 solar radii which is the new feature we have not previously analyzed. We see a similar overall pattern in this feature as it is related to speed with again an overall positive correlation. Local to the FN event however, there does seem to be a negative correlation due to the close background events and their higher predictions. Revisiting the Latitude feature in Figure 5.13c, we also can see a negative correlation local to the FN despite what we had considered a more favorable latitude value.

Table 5.26 shows the group feature importance for the overall dataset, the FP, and



(a) V Log V

(b) Type II Visualization Area with Symlog Scale

(c) Diffusive Shock with Log Scale

(d) 2nd order speed at 20 solar radii

Figure 5.14: Predicted classifier score vs 4 features for DL+rRT+AE 0.6.

Table 5.26: Feature Group Importance for DL+rRT+AE 0.6 with the same format as Table 4.13.

	Overall		FP		FN	
Group (J)	$R_{\hat{I}_J}$	$\hat{I}_J$	$R_{\hat{W}_{iJ}}$	$\hat{W}_{iJ}$	$R_{\hat{W}_{iJ}}$	$\hat{W}_{iJ}$
Speed	1	0.496	1	0.512	1	0.592
Location	2	0.180	3	0.130	2	0.128
CME History	3	0.161	5	0.065	5	0.073
Other	4	0.106	2	0.175	3	0.109
Size	5	0.058	4	0.118	4	0.098

the FN. Same as the group analysis in the cRT+AE 70, we have the same overall group ranking order, and speed dominates in all three columns. The FP and FN again differ in the second and third highest group. The FP again has Other as its second highest group supported by the Type II Visualization Area importance. The FN has Location as its second highest group supported by the high local importance values of Richardson's equation, CPA, and MPA. We also have analyzed how the latitude feature has a high contribution to the FN prediction. The overall group again has Location and CME History as the second and third groups. The FP and FN show some agreement on the group ranking for Location, but both of their CME History rankings are 5.

Chapter 6

Conclusions

6.1 Summary of Findings

In this work, we first addressed the problem of classifying SEP events using CME measurements and derived values. We leveraged oversampling to overcome the large imbalance of the very few SEP events vs the prolific number of Non-SEP events with success as the best performing models across the three techniques used more and more oversampling. Using cRegNN 20, we achieved the highest TSS score of 0.911 almost perfectly classifying all of the SEP events, however we saw a trade-off because of its 5.8 FP events. There was an improvement when applying cRT 60 with the FP count dropping to 3.2 but at the cost of almost another entire FN event. Our greatest performance was in the inclusion of the autoencoder in cRT+AE 70 with an F1 score of 0.800 and only 1.0 FP and 1.0 FN.

We secondly addressed the problem of predicting the ln peak intensity for SEP events using the same dataset. Across all four main approaches, we found a similar trend of increasing performance by applying re-training, rRT, and further the autoencoder, rRT+AE. The best performer in the first main approach was rRT+AE 10.

With this technique, we had a fairly high F1 score of 0.742. Our experimentation into learning new Richardson coefficients showed improvement over the original Richardson equation in our MAE comparison, and the approaches of RC and RE showed that in improved PCC and TSS. The RE+rRT 30 technique had the highest PCC over the SEP and SEP+Elevated datasets, but its poor F1 performance was caused by its 6.4 FP. The RC+rRT+AE 50 had the top TSS score of 0.952 even closer to a perfect TP count at 4.8 than the classifier's top results, but this came with the trade-off of 5.2 FP events. The last main approach of DL, specifically the DL+rRT+AE 0.6 technique, maximized our F1 score at 0.800. The remaining 1.0 FP and 1.0 FN events were the same across the top classification and top regression models.

6.2 Limitations and Possible Improvements

In the results and analysis, we only presented the evaluation on 1 of the 3-fold cross validation (CV) datasets. Since we have so few SEP events, we may have gotten lucky or unlucky in the random partitioning that produced the training/test pair that we used. A better assessment of our approaches would be to present the average over all 3 folds so that each SEP event ends up in one of the test sets. We can also take the CV analysis a step further by running multiple rounds when generating the CV datasets. Generating a CV dataset involves randomly partitioning a bucket into training, validation, and test set for each fold. By repeating this random partitioning, we can generate multiple CV datasets. The averaged and analyzed results over all the CV datasets will be even more reliable than only including the remaining folds of the CV dataset into additional results and analysis.

As mentioned in our contribution, the RC, RE, and DL+rRT+AE techniques can generalize. The general idea of the RC technique is to combine the Richardson forecast and another forecast together in a learned weighted sum. The idea of the RE technique was to learn the error remaining in the Richardson forecast. We can generalize that to say its an Estimating Error (EE) technique which involves learning a model that estimates the error remaining in another forecast. We stated that the RC+rRT+AE 50 model had the highest TSS almost perfect predicting all the SEP events (only 0.2 FN remaining). The DL+rRT+AE 0.6 technique had the highest F1 score, more FN 1.0 vs 0.2 but less FP 1.0 vs 5.2. We could apply the key idea of the RC technique to combine these models together. That is, we could experiment with combining the Richardson forecast with the DL+rRT+AE technique, RC+DL+rRT+AE, performing a hyperparameter search for its best α_{DW} value. The RE technique did worse on its own, but we could also experiment further with applying the generalized form of the EE technique to the RC+DL+rRT+AE technique, EE+RC+DL+rRT+AE. Thereby, we would estimate the error remaining in the RC+DL+rRT+AE model's forecast.

There are untested techniques that may have improved performance in either the classification or regressions tasks if applied. For example, we discussed the RankSim approach by Gong et al. [4] in the Related Work that was applied alongside other techniques such as rRT to improve performance on imbalanced datasets by affecting the feature space. Unlike the autoencoder technique which we used to help find new features from the input data, the RankSim technique uses the target values to help align the ranked feature representation with the ranked target values.

Looking to other tasks, this work does not give a time estimate either for onset or when the predicted peak of the intensity will occur. The techniques outlined in this work could be applied to predicting when the harmful protons will arrive or when they will be at their highest level for better practical application.

Bibliography

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

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

[3] Alessandro Bruno and Ian G Richardson. Empirical model of 10–130 MeV solar energetic particle spectra at 1 AU based on coronal mass ejection speed and direction. *Solar Physics*, 296(2):36, 2021.

[4] Yu Gong, Greg Mori, and Fred Tung. RankSim: Ranking similarity regularization for deep imbalanced regression. In *International Conference on Machine Learning*, pages 7634–7649. PMLR, 2022.

[5] Stephen W Kahler and Alan G Ling. Forecasting solar energetic particle (SEP) events with flare x-ray peak ratios. *Journal of Space Weather and Space Climate*, 8:A47, 2018.

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

[7] Spiridon Kasapis, Lulu Zhao, Yang Chen, Xiantong Wang, Monica Bobra, and Tamas Gombosi. Interpretable machine learning to forecast SEP events for solar cycle 23. *Space Weather*, 20(2):e2021SW002842, 2022.

[8] Kyong Nam Kim, Sun Ae Sin, Kum Ae Song, and Jin Hyok Kong. A technique for prediction of SPEs from solar radio flux by statistical analysis, ANN and GA. *Astrophysics and Space Science*, 363(8):170, 2018.

[9] Kamen Kozarev, Mohamed Nedal, Rositsa Miteva, Momchil Dechev, and Pietro Zucca. A multi-event study of early-stage SEP acceleration by CME-driven shocks—sun to 1 AU. *Frontiers in Astronomy and Space Sciences*, 9:801429, 2022.

[10] Nuno Moniz, Rita Ribeiro, Vitor Cerqueira, and Nitesh Chawla. Smoteboost for regression: Improving the prediction of extreme values. In *2018 IEEE 5th international conference on data science and advanced analytics (DSAA)*, pages 150–159. IEEE, 2018.

[11] The heliopedia. `https://www.nasa.gov/mission_pages/sunearth/the-heliopedia`.

[12] Omniweb. `https://omniweb.gsfc.nasa.gov/form/dx1.html`.

[13] Jiawei Ren, Mingyuan Zhang, Cunjun Yu, and Ziwei Liu. Balanced MSE for imbalanced visual regression. In *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 7916–7925. IEEE, 2022.

[14] Marco Tulio Ribeiro, Sameer Singh, and Carlos Guestrin. "why should i trust you?" explaining the predictions of any classifier. In *Proceedings of the 22nd ACM SIGKDD international conference on knowledge discovery and data mining*, pages 1135–1144, 2016.

[15] IG Richardson, ML Markus, and BJ Thompson. Prediction of solar energetic particle event peak proton intensity using a simple algorithm based on CME speed and direction and observations of associated solar phenomena. *Space Weather*, 16(GSFC-E-DAA-TN65896), 2018.

[16] Daily total sunspot number. `https://www.sidc.be/SILSO/infosndtot`.

[17] Michael Steininger, Konstantin Kobs, Padraig Davidson, Anna Krause, and Andreas Hotho. Density-based weighting for imbalanced regression. *Machine Learning*, 110:2187–2211, 2021.

[18] Peter William Tarsoly. Forecasting SEP events based on merged CME catalogs using machine learning. Master's thesis, Florida Institute of Technology, 2021.

[19] Peter John Thomas. A machine learning approach to forecasting SEP intensity and times based on CME and other solar activities. Master's thesis, Florida Institute of Technology, 2022.

[20] Jesse Torres, Lulu Zhao, Philip K Chan, and Ming Zhang. A machine learning approach to predicting SEP events using properties of coronal mass ejections. *Space Weather*, 20(7):e2021SW002797, 2022.

[21] Jesse Scott Torres. A machine learning approach to forecasting SEP events with solar activities. Master's thesis, Florida Institute of Technology, 2020.

[22] Lina Tran. Pathfinding experiment to study origins of solar energetic particles. `https://www.nasa.gov/feature/goddard/2021/uvsc-pathfinding-experiment-study-origins-of-solar-energetic-particles`, 10 2021.

[23] Peng Wang, Kai Han, Xiu-Shen Wei, Lei Zhang, and Lei Wang. Contrastive learning based hybrid networks for long-tailed image classification. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 943–952, 2021.

[24] Yuzhe Yang, Kaiwen Zha, Yingcong Chen, Hao Wang, and Dina Katabi. Delving into deep imbalanced regression. In *International Conference on Machine Learning*, pages 11842–11851. PMLR, 2021.

[25] H Zhang, M Cisse, Y Dauphin, and D Lopez-Paz. mixup: Beyond empirical risk minimization. *ICLR*, 2, 2018.

[26] Songyang Zhang, Zeming Li, Shipeng Yan, Xuming He, and Jian Sun. Distribution alignment: A unified framework for long-tail visual recognition. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 2361–2370, 2021.

[27] Zhisheng Zhong, Jiequan Cui, Shu Liu, and Jiaya Jia. Improving calibration for long-tailed recognition. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 16489–16498, 2021.

[28] Boyan Zhou, Quan Cui, Xiu-Shen Wei, and Zhao-Min Chen. BBN: Bilateral-branch network with cumulative learning for long-tailed visual recognition. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 9719–9728, 2020.

Appendix A

Richardson's Model

Richardson et al. [15] modeled peak intensity using the following equation:

$$I(\phi)(MeVs \cdot cm^2 \cdot sr)^{-1} \approx 0.013exp(0.0036V - \frac{\phi^2}{2\sigma^2}), \sigma = 43^\circ \tag{A.1}$$

for a CME with connection angle ϕ and speed V. Richardson used CMEs from the CDAW catalog with proton intensities from 14- to 24-MeV to create Equation A.1. We include only the connection angle term in our input feature derived from his equation as

$$Feature_{Richardson} = 0.013exp(-\frac{\phi^2}{2\sigma^2}), \sigma = 43^\circ \tag{A.2}$$

for two reasons. Firstly, we already have linear speed as an input feature. Secondly, we want the neural network models to find any possible relationship among connection angle, speed, and other features without biasing it through the use of Richardson's full equation.

Connection angle is calculated from the DONKI features using the following equation:

$$\phi = arccos(sin(\theta_1) * sin(\theta_2) + cos(\theta_1) * cos(\theta_2) * cos(\phi_1 - \phi_2)) \tag{A.3}$$

where $\theta_1 = latitude$, $\theta_2 = 0$, $\phi_1 = longitude$, and $\phi_2 = \frac{AngularSpeedOfSun*1AU}{SolarWindSpeed} = \frac{\frac{360}{27.27*86400}*1.5*10^8}{SolarWindSpeed}$. Solar wind speed was pulled from the OMNIWeb website [12] as the Plasma Flow Speed averaged over the hour when the CME started according to the DONKI timestamp. According to Appendix A in Torres [21], the value for θ_2 varies between -7° and 7° based on the seasons. We fix it at 0 for ease of implementation.

Appendix B

Diffusive Shock

We adapted the Diffusive Shock Equation specified by Torres replacing the constants used with 10 MeV data with constants for 100 MeV protons [21]. First, they define the constants involved in the Diffusive Shock equation.

- v is particle speed; for 100 MeV protons, v = 128,474.629 km/s (see Equation B.1)
- V_A is Alven speed, which is typically between 500 and 2000 km/s; we fix this value at 600 km/s
- v_{th} is proton thermal speed, which is around 150 km/s
- η is shock efficiency, which is around 0.1
- κ is the distribution parameter, which is between 1.5 and 3; we fix this value at 2
- V_{sh} is shock speed, or the list of Linear speed values for each CME

To explain the 100 MeV constant replacing the 10 MeV constant, v can be calculated from:

$$v = c\sqrt{1 - (\frac{1}{\gamma})^2} = 128,474.629 \tag{B.1}$$

where $c = 3 \cdot 10^5 km/s$ and $\gamma = \frac{100MeV+938MeV}{938MeV}$. The other constants are pulled directly from Torres.

Torres next calculates a quantity M:

$$M = \frac{V_{sh}}{V_A} \tag{B.2}$$

They use a threshold of 1.1 to compute a quantity γ to be used in the Diffusive Shock equation.

If M >1.1, then:

$$\gamma = \frac{4M^2}{M^2 - 1} \tag{B.3}$$

Otherwise:

$$\gamma = \frac{4 \cdot 1.1^2}{1.1^2 - 1} \approx 23 \tag{B.4}$$

Torres calculates another quantity v_{inj} for the final result:

$$v_{inj} = 2.5V_{sh} \tag{B.5}$$

Finally, the overall Diffusive Shock equation is defined:

$$DiffusiveShock = \eta v \frac{1}{\gamma - 1} \frac{1}{(1 + \frac{v_{inj}^2}{\kappa v_{th}^2})^{\kappa+1}} (\frac{v_{inj}}{v})^{\gamma+1} \tag{B.6}$$

Appendix C

Changes to CME Data

With expert knowledge from Dr. Ming Zhang, we made some adjustments to specific samples in the dataset gathered from CDAW and DONKI catalogs. These adjustments were made after careful analysis by Dr. Zhang of the specific events across both catalogs. Modifications were considered because there is some discrepancy between the CDAW and DONKI catalog entries for some feature values. Discrepancies can exist because of the manual, human involved component when estimating their values from relevant measurements. In Table C.1, we summarize the advised adjustments we made to the original dataset.

Table C.1: Adjustments made to specific events in the dataset.

DONKI Date	Feature Name	Old Value	New Value
2/25/2014 1:25	Linear Speed	1670	2147
9/22/2011 11:24	Linear Speed	1000	1905
4/11/2013 7:36	Linear Speed	675	1000
7/23/2012 02:36	Type II Area	18377000	3355800