

Technical University of Košice
Faculty of Electrical Engineering and Informatics

**Applicability of machine learning for ESA
mission Vigil**

Diploma thesis

2025

Bc. Adam Majirský

Technical University of Košice
Faculty of Electrical Engineering and Informatics

Applicability of machine learning for ESA mission Vigil

Diploma thesis

Study programme: Business informatics
Field of study: Computer Science
Department: Department of Cybernetics and Artificial Intelligence (KKUI)
Supervisor: doc. Ing. Peter Butka, CSc.
Consultant: RNDr. Šimon Mackovjak, PhD.

Košice 2025

Bc. Adam Majirský

Abstrakt

Extrémne javy vesmírneho počasia predstavujú významné ohrozenie pre modernú infraštruktúru. Pripravovaná misia Vigil má za úlohu poskytovať monitorovanie a predpovedanie slnečnej činnosti. V tejto práci sme vytvorili dataset historicky extrémnych prejavov vesmírneho počasia s využitím údajov z prístrojov najpodobnejších misii Vigil, čo umožňuje štúdie dolovania v dátach pre túto pripravovanú misiu. Vyvinuli sme softvér umožňujúci prístup k týmto dátam, a poskytli sme ho vedeckej komunite. Natrénovali sme modely hlbokého učenia na predpovede geomagnetických búrok. Naša štúdia poskytuje pohľad na zdroje šírenie extrémnych javov vesmírneho počasia, ale navrhuje aj rámec na porovnávanie týchto javov navzájom.

Kľúčové slová

Vesmírne počasia, geomagnetické búrky, hlboké učenie, predikcia

Abstract

Extreme space weather events pose a significant threat to modern infrastructure. Upcoming Vigil mission aims to provide monitoring and forecasting solar activity. In this work, we created a dataset of historically extreme space weather events using data from the most similar instruments to Vigil, enabling data-driven studies for this upcoming mission. We developed a pipeline to obtain to these observations, and provided it to the scientific community. We trained deep learning models for geomagnetic storm forecasting. Our study provides insights into the sources and propagation extreme space weather events, and also proposes a framework for comparing these events to each other.

Keywords

Space weather, geomagnetic storms, deep learning, prediction

TECHNICAL UNIVERSITY OF KOŠICE
FACULTY OF ELECTRICAL ENGINEERING AND INFORMATICS
Department of Cybernetics and Artificial Intelligence

DIPLOMA THESIS
ASSIGNMENT

Field of study: **Computer Science**
Study programme: **Business Informatics**

Thesis title:

Applicability of machine learning for ESA mission Vigil
Využitelnosť strojového učenia pre plánovanú ESA misiu Vigil

Student: **Bc. Adam Majirský**


Supervisor: **doc. Ing. Peter Butka, PhD.**
Supervising department: **Department of Cybernetics and Artificial Intelligence**
Consultant: **RNDr. Šimon Mackovjak, PhD.**
Consultant's affiliation: **Ústav experimentálnej fyziky SAV**

Thesis preparation instructions:

1. Analyze missions related to the focus of the planned ESA Vigil space weather monitoring mission, with an emphasis on the expected data produced by the mission.
2. Prepare datasets simulating future data from the Vigil mission based on related missions and observed phenomena in the past.
3. Analyze the possibilities of selected aspects of space weather with the aim of modeling them using machine learning and appropriate visualization.
4. Design, implement and evaluate experiments for selected cases of analysis of the prepared data.
5. Prepare documentation according to the instructions of the department and the supervisor.

Language of the thesis: English
Thesis submission deadline: 17.04.2025
Assigned on: 31.10.2024




.....
prof. Ing. Liberios Vokorokos, PhD.
Dean of the Faculty

Declaration

I hereby declare that this thesis is my own work and effort. Where other sources of information have been used, they have been acknowledged.

Košice 17. 4. 2025

.....

Signature

Acknowledgement

I would like to thank my supervisor, doc. Ing. Peter Butka, PhD., for the knowledge, comments, and valuable advice he shared with me. Sincere thanks also go to RNDr. Šimon Mackovjak, PhD., who showed me how science is done. The knowledge gained in this work will surely shape my career. I always looked forward to our joint meetings and left them with a broader perspective on the given challenges, and certainly on life as well.

Contents

Introduction	1
1 Machine Learning	3
1.1 Neural Networks	5
1.2 Activation Functions	7
1.3 Loss Functions	10
1.4 Backpropagation and Optimization	11
1.5 Deep Learning Training Principles	13
1.6 Convolutional Neural Networks	17
1.7 Recurrent Neural Networks	20
2 Space Weather	22
2.1 Solar Activity Monitoring Missions	23
2.2 Machine Learning in Space Weather	27
2.3 State of The Art for Dst Prediction	28
3 Dst Prediction	31
3.1 Scientific Understanding	31
3.1.1 Constraints	34
3.1.2 Concerns	36
3.1.3 Plan	37
3.1.4 Goals	37
3.2 Data Understanding	38
3.3 Data Preparation	43
3.3.1 Processing Multi-modality	43
3.3.2 Data Integration for Modeling	48
3.4 Modeling	49
3.4.1 Data Split	49
3.4.2 Model Implementation	52

3.4.3	Model Evaluation Methods	55
3.4.4	Model Performance	60
3.5	Project Evaluation	63
4	Conclusion	65
	Appendices	78

List of Figures

1–1	(a) Shows Perceptron. (b) Shows Multilayer Perceptron with input, hidden and output layers of Perceptrons interconnected. (c) Shows Multilayer Perceptron in vector notation according to Aggarwal (2023). \bar{x} and \bar{h}_l are vectors and W_l are matrices with size depending on size of layers. This notation corresponds to equations 1.3, 1.4 and 1.5	7
1–2	Approximation of a function with multiple linear pieces. A neural network creates one extra linear piece per hidden unit. With enough hidden units, we should be able to approximate any function. Courtesy of Prince (2023).	8
1–3	Activation functions commonly used in neural networks	9
1–4	Demonstration of model capacity and its fit on training data points. (A) Underfitting: linear model is too simple to capture sinusoidal pattern. It has poor performance on training data and will have poor performance if new, unseen points are added. (B) Overfitting: High-Degree polynomial is too complex for sinusoidal pattern and fits training data very closely. It perfectly meets all training data, but misses sinusoidal pattern. New unseen data point will be probably missed by model prediction badly. (C) Optimal Fit: Model captures sinusoidal pattern, although it is not perfect on training data, it maintains good generalization when new data will be added.	14

1–5	Figure (A) shows bias-variance trade-off to which authors refer as 'Classical' regime. Figure (B) shows that given large capacity of a model, training loss becomes zero-close and test loss starts to descent again to even lower value than optimal point in Classical regime. This second regime author refer as 'Modern' or 'Overparametrized' regime. Courtesy of Belkin et al. (2019).	15
1–6	A Convolutional Neural Network consisting of two sequential convolutional and pooling layers, followed by a flattening operation that transforms the 2D feature map into a 1D representation, which is then processed by a fully connected classifier. Taken from Sermanet et al. (2012).	19
1–7	Structure of GRU cell. Taken from Abboush et al. (2023).	21
2–1	Lagrange points in the two-body Sun-Earth system, according to Lo et al. (2010).	26
2–2	Annual number of publications dedicated to solving machine learning problems in the space weather domain, according to KA et al. (2023).	28
3–1	Data count for each extreme SWE event by different instruments measuring space weather activity. Events 2, 3, and 39 (highlighted in red) lacks crucial data. Events 16, 18, 20, 24, 26 and 27 (highlighted in green) are merged with preceding events, as they occurred within very close interval.	40
3–2	Showcase of data from different instruments used in dataset.	41

3–3	<i>Left:</i> Overlap of 35 images of moments when the most extreme SWE events were produced, as captured by SOHO/EIT 195 Å and SDO/AIA 193 Å. Visible flares are numbered, corresponding to events listed in Table 3–1. It is noted that science-ready FITS files were used to generate source images. However, the cross-calibration was not considered between the instruments as we are here interested only in the locations of flares and not in their absolute intensity. <i>Right:</i> The distribution of time scales for events listed in Table 3–1. Here, the duration is calculated from the time when the event was visible on SOHO/EIT 195 Å, SDO/AIA 193 Å, and/or SOHO/LASCO C2 images until the time when the Dst index reached its extreme value. The four missing time scales (bars) are caused by unavailable images that are needed for the estimation of the event initialization time.	42
3–4	MESWE Dataset created by aligning remote-sensing image data with in-situ measurements for extreme SWE events. The dataset consists of 13 attributes, capturing information on coronal hole coverage across regions of the solar disc, running differences calculated in the four quadrants of coronagraph images, in-situ measurements consisting of solar wind speed, solar wind density, South-North component of magnetic field and Dst index values.	45
3–5	Wasserstein distances between one event to each other. We can observe large distance of events 19,21,30, and 31 with respect to other extreme SWE events.	50
3–6	Median values of solar wind speed during each extreme SWE event.	50

3–7	Different examples of categorized events. Left y-axis display Coronagraph Visual Features values. Right y-axis display Dst value in nanotesla. We can observe that right before Dst decrease, each example has different type of activity in Coronagraph Visual Features values. (a) Event 31 (2006-12-14) is categorized as Spiking. There is visible coronagraph activity before Dst decrease for all 4 Q-attributes with higher percentage value. (b) Event 38 (2023-04-24) is categorized as Non-spiking. Although there is visible coronagraph activity before Dst decrease, only Q4 is active and at very low value, only around 1%. (c) Event 11 (2000-09-17) is categorized as Unclassified. Although there is visible coronagraph activity before Dst decrease, not all attributes from coronagraphs are active.	51
3–8	Different models used to forecast Dst index.	56
3–9	Changes in the warping path and diagonal values for different time series alignments. (a–c) show different simulated time series, while (d–f) depict their corresponding DTW cost matrices with warping paths. In an ideal alignment, the warping path follows the diagonal, and diagonal values remain zero. For time series with differences in magnitude, the warping path deviates slightly from the diagonal, and diagonal values are no longer zero. When sequences are misaligned in the time axis, only the first and last steps of the warping path remain on the diagonal, with a significant increase in diagonal values.	59
3–10	Detailed test set predictions 10 steps ahead of event that occurred on March 2015 using different approaches.	62

List of Tables

1–1	Comparison between Machine Learning and Deep Learning	4
2–1	Table of planned Vigil instruments.	27
2–2	The matrix of space missions instruments that are similar to the category of Vigil’s instruments.	27
3–1	Table of identified great and severe geomagnetic storms based on the Dst and ap indices. The event start time represents the first moment when one of the parameters is below (above) the threshold for determining an extreme geomagnetic storm. The event end time represents the time when all of the parameters did not exceed the threshold in the following at least two hours. For each time interval, the most extreme parameter values are displayed.	32
3–2	Attributes in MESWE Dataset	44
3–3	Selected events for each k-fold validation and test sets, ensuring sets similarity. Remaining events were used as training data in each k-fold, so each k-fold is trained, validated and tested on different combination of extreme SWE events. Each validation or test set in each k-fold contains one representative event from each of the S, NS, and U event categories.	52
3–4	$RMSE$ and D_{on} with D_{sum} of cost DTW matrix for $GRU -$ $CONV - IEC$ approaches and for prediction 10 steps and 20 steps ahead of Dst prediction on test set.	60
3–5	$RMSE$ and D_{on} with D_{sum} of cost DTW matrix for $GRU - IE$ and $GRU - Attn - IEC$ approaches and for prediction 10 steps and 20 steps ahead of Dst prediction on test set.	61

List of Symbols and Abbreviation

Å Angstrom

AIA Atmospheric Imaging Assembly

Adam Adaptive Moment Estimation

CNN Convolutional Neural Networks

CME Coronal Mass Ejection

CRISP-DM Cross-Industry Standard Process for Data Mining

CIR Corotating Interaction Regio

DL Deep Learning

DST Disturbances Storm Time

DTW Dynamic Time Wrapping

EIT Extreme Ultraviolet Imaging Telescope

EUV Extreme Ultraviolet

GELU Gaussian Error Linear Unit

GRU Gated Recurrent Unit

GOES Geostationary Operational Environmental Satellite

HEK Heliophysics Events Knowledgebase

KPI Key Performance Indicators

LSTM Long Short-Term Memory

ML Machine Learning

MLP Multilayer Perceptron

MESWE Most Extreme Space Weather Events

nm nanometer

ReLU Rectified Linear Unit

SWE Space Weather

SDO Solar Dynamics Observatory

SOHO Solar and Heliospheric Observatory

STEREO Solar TERrestrial RELations Observatory

UV Ultraviolet

Introduction

Extreme Space Weather events can negatively affect ground-based infrastructure and satellite communications. European Space Agency plans to launch a new operational mission, Vigil, to monitor space weather activity and provide timely warnings about immediate danger. Prediction of solar events and their effect on Earth has been longstanding challenge. To tackle this, research community utilize both numerical models based on laws of physics and machine learning models to extract patterns in large amounts of data.

Machine Learning gained popularity in the field of space weather in recent years and it does not seem to stop soon. We discuss machine learning theory in chapter 1, space weather in 2 and finally history and state of the art of the machine learning in space weather in 2.2 and 2.3. There is also line of work within the research group at Department of Cybernetic and Artificial Intelligence (KKUI) devoted to this exploration of machine learning use. Motúzová (2023) predicted Earth's geomagnetic state using deep learning and historic parameters of solar wind. Zboray (2024) used explainable methods for deep learning models that predicted Earth's geomagnetic state.

This work serves as pathfinder study for data-driven project for upcoming Vigil mission. Since machine learning expects historic data, it creates first interesting challenge - what data can be used for Vigil in order to use machine learning for space weather prediction? To tackle this challenge, we propose new dataset from periods of extreme solar activity, MESWE, which we describe in chapter 3.3. This enables data-driven studies for Vigil. To our knowledge, this is the first study dedicated to this. Our research findings on this topic has been published in peer reviewed journal Majirský et al. (2025).

However, main benefit of proposed MESWE dataset is integration of in situ measurements and images of solar activity. This provides integrated information from different sources and is worth of exploring. We use deep learning methods to

predict geomagnetic state of the Earth, which is framed as Dst index prediction in time series. Implementation is described in chapters 3.4.2. Results of experiments are reported and discussed in 3.4.4.

This work was conducted in collaboration with RNDr. Šimon Mackovjak, PhD., affiliated with the Institute of Experimental Physics, Slovak Academy of Sciences. Outcomes were part of deliverables for the RPA SKR1-23 project "Study toward enhancing reliability and timeliness of Vigil mission predictions through Machine Learning", funded by the European Space Agency.

1 Machine Learning

The exponential growth in data generation and collection has created unprecedented opportunities across various domains, extending far beyond traditional business applications. However, manually analyzing these vast datasets to identify patterns, relationships, and underlying rules would be a daunting task for human analysts.

According to LeCun et al. (2015) Machine Learning (ML) offers a framework of techniques and mathematical approaches to automatically discover these patterns and relationships, enabling systematic knowledge extraction from data. Deep Learning (DL), a specialized subfield of ML, employs artificial neural networks (ANNs) as powerful tools to map input data to meaningful outputs tailored for specific objectives. The conceptual distinctions between ML and DL are summarized in Table 1 – 1. Both ML and DL are current philosophies or ways to achieve Artificial Intelligence (AI).

Since the introduction of AlexNet Krizhevsky et al. (2012), research has consistently demonstrated that DL outperforms traditional ML approaches in tasks involving high-dimensional and large-scale data, such as Computer Vision (CV) and Natural Language Processing (NLP). Nonetheless, ML remains valuable in scenarios characterized by limited training samples or when model interpretability is of critical importance.

The most common ways to create ML/DL models are supervised learning and unsupervised learning. In supervised learning, datasets consist of paired inputs and corresponding observed or measured outputs. For example, a dataset may include housing attributes paired with their respective prices or images paired with labels indicating the presence of a dog or a cat. In this paradigm, a ML or DL model approximates a function that maps inputs to expected outputs based on these provided pairs. The objective is for the supervised model to generalize and accurately predict outputs for new inputs for which the observed outputs are not yet known.

Common tasks adapted for ML and DL methodologies are regression and classi-

Table 1 – 1: Comparison between Machine Learning and Deep Learning

Aspect	Machine Learning	Deep Learning
Data Requirements	Suitable for small datasets	Requires large datasets for best performance
Feature Engineering	Requires manual feature extraction and selection	Automatically learns and extracts relevant features from raw data
Interpretability	More transparent and easier to understand how decisions are made	Often considered a "black box" with less interpretable decision-making process
Problem Types	Better for structured data and traditional problems with clear feature definitions	Excels at complex patterns in unstructured data (images, text, audio)
Scalability	Limited by feature engineering	Scales great with more data and compute
Model Examples	SVM, Random Forests, Linear Regression	MLP, CNN, RNN

fication Prince (2023). Regression involves predicting a continuous numerical output based on a set of input features. For instance, estimating the price of a house based on attributes such as the number of bedrooms, location, and available utilities exemplifies a regression task.

In contrast, classification tasks involve assigning input data to predefined discrete categories. An example of a classification task is identifying whether an input image contains a dog or a cat.

Unsupervised learning algorithms operate only on input data without corresponding output labels. The primary goal in this context is to uncover the underlying structure or distribution of the input data without explicit feedback regarding correctness. An example is a clustering task, where the model groups data points based

on their proximity or similarity in the feature space.

1.1 Neural Networks

The most simplest unit of a neural network is thought to be Perceptron proposed by Rosenblatt (1958). Perceptron is simple mathematical construct that takes input represented as vector of n attributes, $\bar{X} = [x_1, \dots, x_n]$. The inputs are multiplied by their corresponding weights, $\bar{W} = [w_1, \dots, w_n]$ and summed. These weights represent the importance of each input x_i in determining the computed output \hat{y} . An activation function σ then maps the weighted sum to either 1 or -1, representing a binary 'yes/no' classification of the output \hat{y} . We can describe this as follows:

$$\sigma(x) = \begin{cases} -1 & \text{if } x < 0 \\ +1 & \text{if } x \geq 0 \end{cases} \quad (1.1)$$

$$\hat{y} = f(\bar{X}) = \sigma\left(\sum_{i=1}^n w_i x_i\right) \quad (1.2)$$

Weights are unknown and are initialized randomly. These random weights will provide output \hat{y} with large loss L when compared to true values y , where $L = |y - \hat{y}|$. The goal of function $f(\cdot)$, in this case perceptron, is to update the weights to minimize the loss between the true values and the predicted values. Details on how weights are modified are outlined in chapter 1.4.

Minsky and Papert (1969) argued that Perceptron is not general solution, as it can not solve tasks, where data are not linearly separable. The primary focus on the solution to this problem was to provide more layers of perceptrons and use non-linear activation functions σ . However, the scientific community struggled to develop an algorithm to modify the weights effectively until the 1980s, when the backpropagation algorithm was independently discovered by four research groups Werbos (1974); Parker (1985); Lecun (1985); Rumelhart et al. (1986).

A Multilayer Perceptron (MLP), often referred to as a Feed Forward Network (FFN) or Artificial Neural Network (ANN), is an extension of the original perceptron

concept. It consists of multiple layers of perceptrons, where the outputs of one layer serve as the inputs to the next. This architecture typically includes an input layer, one or more hidden layers, and an output layer, with each layer fully connected to the subsequent layer. According to Aggarwal (2023) we can express neural network with $n - 1$ hidden layers as following:

$$\bar{h}_1 = \sigma(W_1 \bar{x}), \quad (1.3)$$

$$\bar{h}_{l+1} = \sigma(W_{l+1} \bar{h}_l) \quad \forall l \in \{1, \dots, n-1\}, \quad (1.4)$$

$$\bar{o} = \sigma(W_{n+1} \bar{h}_n), \quad (1.5)$$

where \bar{x} is the input vector to the MLP, \bar{h}_l is activations of the l -th hidden layer, W_l are weight matrix for the l -th layer, where each row of the matrix corresponds to a single neuron in the l -th layer, and each column corresponds to an input from the next layer. \bar{o} is the output vector of this MLP construct. Perceptron and MLP are shown in figure 1–1.

In MLPs, the structure of the layers plays a crucial role which is called capacity: the number of neurons in a layer, known as the width of the network, and the number of hidden layers, referred to as the depth. These, along with the choice of activation function σ , are adjustable.

Why might one choose different activation functions? As noted by Aggarwal (2023), a neural network with many hidden layers and only linear activation functions is fundamentally no different from a neural network with just one hidden layer and linear activation functions. Curious readers are directed to the proof of Theorem 1.5.1 for further details. This observation highlights the importance of nonlinear activation functions, which allow the model to capture complex, non-linear relationships in the data.

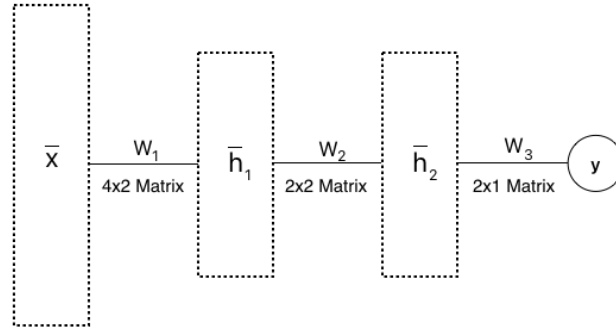
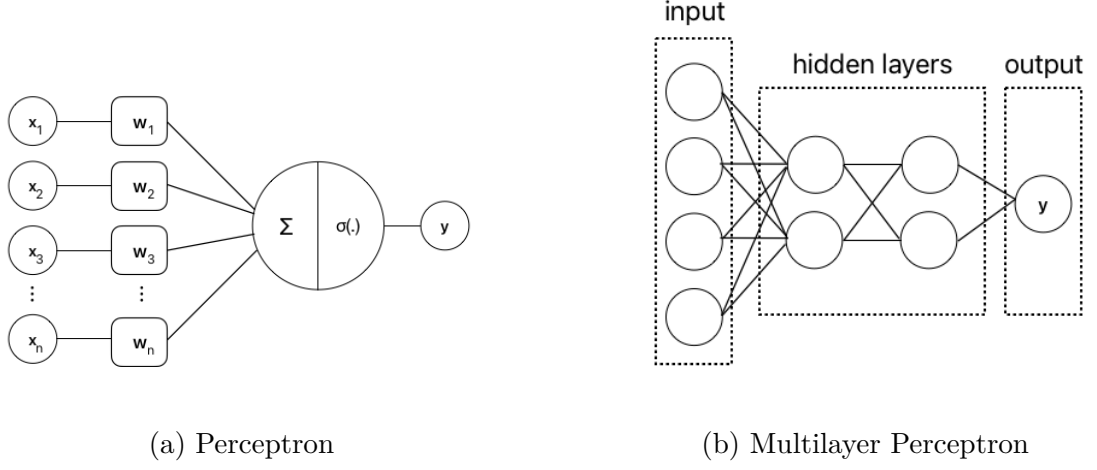


Figure 1 – 1: (a) Shows Perceptron. (b) Shows Multilayer Perceptron with input, hidden and output layers of Perceptrons interconnected. (c) Shows Multilayer Perceptron in vector notation according to Aggarwal (2023). \bar{x} and \bar{h}_l are vectors and W_l are matrices with size depending on size of layers. This notation corresponds to equations 1.3, 1.4 and 1.5

1.2 Activation Functions

Non-linear activation functions introduce the necessary non-linearity, allowing the network to learn and represent complex patterns. They enable outputs of one layer to interact in non-linear ways with subsequent layers, which is crucial for deep networks

that capture hierarchical features in data. Cybenko (1989) and Hornik (1991) shown that neural network with at least one hidden layer with non-linear activation function can approximate any function well if number of neurons in hidden layer is large enough. This is show in figure 1–2.

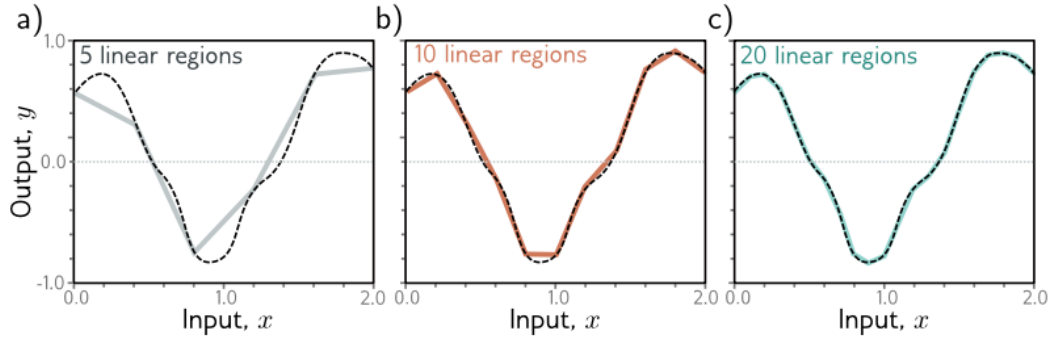


Figure 1–2: Approximation of a function with multiple linear pieces. A neural network creates one extra linear piece per hidden unit. With enough hidden units, we should be able to approximate any function. Courtesy of Prince (2023).

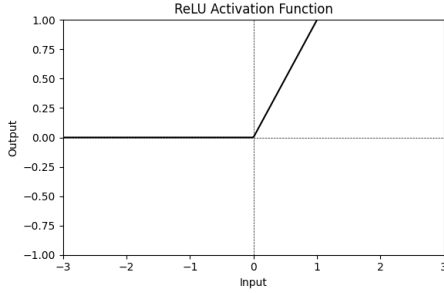
Sigmoid, hyperbolic tangents and ReLU are the most commonly used activation functions. Figure 1–3 shows their output. They can be denoted as following:

$$\text{sigmoid}(x) = \frac{1}{1 + e^{-x}}, \quad (1.6)$$

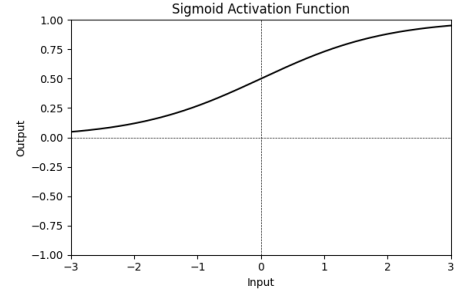
$$\tanh(x) = \frac{e^x - e^{-x}}{e^x + e^{-x}}, \quad (1.7)$$

$$\text{ReLU}(x) = \max(0, x). \quad (1.8)$$

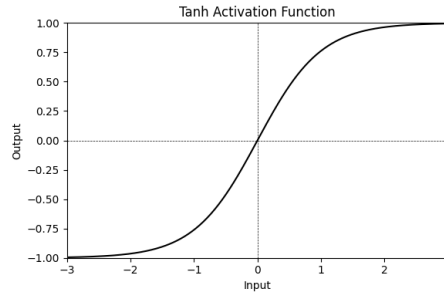
ReLU is notably most used function for hidden layers in MLPs, since Glorot et al. (2011) demonstrated that ReLU activation functions offer significant advantages over sigmoid and tanh functions. The authors established that both sigmoid and tanh functions can lead to saturation during training, where the gradients become



(a) ReLU



(b) Sigmoid



(c) Tanh

Figure 1 – 3: Activation functions commonly used in neural networks

exceedingly small, thereby impeding the neural network's convergence rate. This saturation effect occurs because sigmoid and tanh functions compress their inputs into a finite range, causing the gradients to vanish when the activation values approach either end of their range. In contrast, ReLU maintains a constant gradient for all positive inputs, allowing for more efficient training and faster convergence. This was supported by their empirical evidence of ReLU's superior performance compared to sigmoid and tanh functions.

Despite these advantages, ReLU activation functions present their own significant limitations. ReLU exhibits 'dying ReLU' problem, where neurons can become permanently inactive Lu et al. (2019). This occurs because ReLU produces zero gradient for all negative inputs, effectively preventing weight updates during backpropagation. This issue becomes especially problematic in neural networks with more hidden layers, as the cumulative effect of large number of inactive neurons can se-

verely hinder the model's ability to train.

Research community have proposed several modified activation functions to address this problem. Maas et al. (2013) introduces Leaky ReLU a small positive slope for negative inputs, ensuring that neurons remain partially active even for negative values, thereby maintaining gradient flow throughout the network. More recently, Gaussian Error Linear Unit (GELU) proposed by Hendrycks and Gimpel (2016) has gained prominence. GELU provides a smoother transition between active and inactive states while incorporating probabilistic properties that can enhance model regularization. Nevertheless, the efficacy of alternative activation functions to ReLU exhibits substantial variability across different tasks and datasets. ReLU maintains its position as a widely adopted activation function due to its straightforward implementation and consistently reliable performance across diverse applications.

1.3 Loss Functions

The loss function quantifies the discrepancy between the predicted outputs of a model, \hat{y} , and the corresponding true values, y , and providing this comparison in a single scalar value. The process of training a neural network involves optimizing its parameters (or weights, W_l) to minimize this loss function.

A critical property of a loss function is its differentiability, as this enables the use of backpropagation and gradient-based optimization methods, to iteratively update the network's parameters as we shall see in chapter 1.4.

Choice of loss function depends on task. For regression tasks, the Mean Squared Error is most common and is defined as follows:

$$MSE = \frac{1}{N} \sum_{i=1}^N (y_i - \hat{y}_i)^2, \quad (1.9)$$

where N is the number of samples, y_i are true values and \hat{y}_i denotes outputs of neural network.

For classification tasks, the Cross-Entropy Loss is used and is defined as:

$$\text{Cross-Entropy} = -\frac{1}{N} \sum_{i=1}^N \sum_{j=1}^C y_{i,j} \log(\hat{y}_{i,j}), \quad (1.10)$$

where K is the number of classes, $y_{i,j}$ is a binary indicator denoting the true class of sample i and $\hat{y}_{i,j}$ is the output probability of the class k .

1.4 Backpropagation and Optimization

We have already mentioned that weights in neural networks are adjusted to minimize the loss function. Natural question arises, how are these weights updated? We can start to approach this with an example. Consider finding minimum of this simple function $f : y = x^2$. One can call back that calculus offers an approach for this task. At the minimum, the derivative, the rate of change, equals zero, as the slope is flat. Differentiating f , setting the derivative to zero, and solving for x yields the point where $f(y)$ achieves its minimum.

However, the result of the loss unction depends on every weight of the neural network. Consider replacing \hat{y} in equation 1.9 by \bar{o} from equation 1.5 which represents output of MLP with numerous weights as W matrices. Unlike the simple case discussed above, finding the minimum of this loss function cannot follow the same direct approach.

Backpropagation Rumelhart et al. (1986) calculates partial differentiation of the loss function with respect to every weight W_{ij} in neural network. Nowadays, backpropagation is calculated effectively with frameworks as PyTorch Paszke et al. (2017) or Tensorflow Abadi et al. (2016), reducing complexity of implementation deep learning model. However it is useful to know how automatic differentiation is done inside these frameworks.

Consider equations 1.3, 1.4 and 1.5 as forward pass of MLP. Then, according to Aggarwal (2023), backward loss propagation and gradient calculation for weights can be noted as following:

$$\delta_o = \frac{\partial L}{\partial \bar{o}} \cdot \sigma'(\bar{o}), \quad (1.11)$$

$$\delta_l = (W_{l+1}^T \delta_l) \odot \sigma'(\bar{h}_l), \quad (1.12)$$

where δ is error term, or also known as local gradient. This combines backpropagated error from following layer and the derivative of activation function in current layer $\sigma'(\cdot)$. Different activation functions have different derivatives, which can be found for example in Lederer (2021). \odot is element-wise product. Then, to calculate weight gradients at layer l , we multiply error term δ_l with a values of previous hidden layer $l - 1$:

$$\frac{\partial L}{\partial W_l} = \delta_l \bar{h}_{l-1}, \quad (1.13)$$

and for the first layer of weights we replace \bar{h}_{l-1} for inputs x^T :

$$\frac{\partial L}{\partial W_1} = \delta_1 \bar{x}^T. \quad (1.14)$$

Gradients of all weights are also commonly denoted as $\frac{\partial L}{\partial W} = \nabla L$ in literature.

Gradient optimization methods update every weight by subtracting a fraction of the gradient at step t , from the previous weights W_{t-1} . Gradients are scaled by the small constant, learning rate α . This moves the value of the loss function towards the minimum:

$$W_t \Leftarrow W_{t-1} - \alpha \nabla L_t. \quad (1.15)$$

This is optimization method is known as Gradient Descent and it is calculated on entire dataset until convergence of a function to a minimum. This is meant by training deep learning model. Stochastic Gradient Descent uses single data sample or small batch of data.

There are known other optimization methods that incorporate momentum to help avoid getting stuck in saddle points of the loss function, where gradient are

small and can slow down convergence. Including Momentum-based Gradient Descent, RMSprop, AdaGrad. Most used optimizer method is however Adam, proposed by Kingma (2014). It is designed to combine advantages of RMSprop and AdaGrad. Adam estimates first m_t and second moment v_t , that are moving averages of past gradients. These moments are tracked as follows:

$$m_t = \beta_1 m_{t-1} + (1 - \beta_1) \nabla L_t \quad (1.16)$$

$$v_t = \beta_2 v_{t-1} + (1 - \beta_2) \nabla L_t^2 \quad (1.17)$$

$$W_t \Leftarrow W_{t-1} - \alpha \frac{m_t}{\sqrt{v_t} + \epsilon} \quad (1.18)$$

where betas β_1 and β_2 are hyperparameters that determinate contribution of past gradients, α is learning rate and ϵ is small constant to prevent division by zero.

1.5 Deep Learning Training Principles

ML and DL models fundamentally aim to achieve great generalization performance on previously unseen examples, commonly referred to as test data. During the training process, weights of the model are optimized using a training dataset, which is assumed to be drawn from the same statistical distribution as the future data model will encounter. Model performance is quantified through two key metrics: the training loss, which measures performance on the training data, and the test loss, which estimates generalization capability. Additionally, a validation dataset is frequently employed as an intermediate evaluation step to select best hyperparameters or to utilize early stopping.

The training objective can be formally defined as the optimization of model parameters to minimize the training loss while simultaneously maintaining a minimal gap between training and test loss. Practitioners commonly encounter two phenomena: underfitting and overfitting. Underfitting occurs when the model fails to achieve

low training error, indicating insufficient learning capacity. Overfitting occurs when there exists an excessive gap between training and test errors, indicating that the model has learned exact solutions or specific details on the training data, which are not true for test data. To handle underfitting and overfitting correctly, it is important to select a model with a capacity that appropriately aligns with the complexity of the task. Overfitting and underfitting are shown in figure 1 – 4.

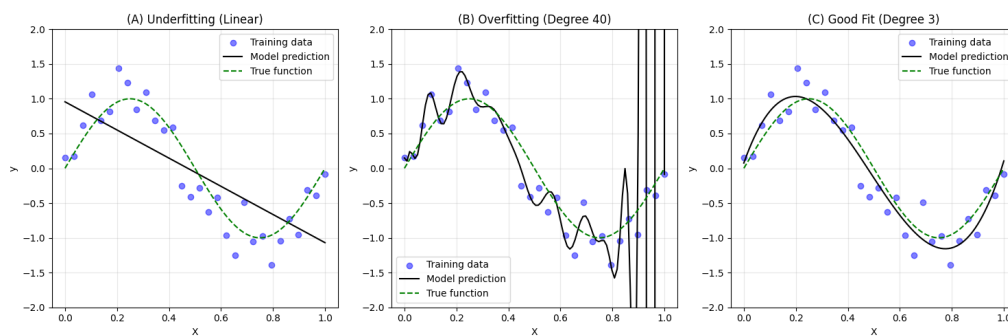


Figure 1 – 4: Demonstration of model capacity and its fit on training data points.

(A) Underfitting: linear model is too simple to capture sinusoidal pattern. It has poor performance on training data and will have poor performance if new, unseen points are added. (B) Overfitting: High-Degree polynomial is too complex for sinusoidal pattern and fits training data very closely. It perfectly meets all training data, but misses sinusoidal pattern. New unseen data point will be probably missed by model prediction badly. (C) Optimal Fit: Model captures sinusoidal pattern, although it is not perfect on training data, it maintains good generalization when new data will be added.

To prevent model overfitting, we can constrain the model's complexity by using techniques of regularization. Most used are:

1. **L1 regularization:** encourages sparsity in model parameters by driving weights to zero, making effective feature selection. It is useful when dealing with many input attributes, but only few are relevant to target variable and others contain noise Ng (2004).

2. **L2 regularization:** penalizes large parameter values that encourages model to use smaller weights across all features Ng (2004).
3. **Dropout:** randomly deactivates neurons during training with p probability, while during testing all neurons are present, but outgoing weights of layer are multiplied by $\frac{1}{1-p}$. It was shown that this improves generalization performance Srivastava et al. (2014).

However, most simple and most used technique to prevent overfitting is early stopping. This halts training process when model stops improving on validation set and saves parameters of the model in optimal performance.

Recently, Belkin et al. (2019) showed that way-over parametrized Machine Learning models can in fact later achieve lower generalization error instead of keep overfitting. This surprising phenomenon is known as double descent, hence descent of test loss first, then as model overfits loss increases but after interpolation threshold of parameters count, test loss starts to decrease again. Figure 1–5 shows double descent in over-parametrized model regime. For interested readers, many literature is devoted to this research Adlam and Pennington (2020), Nakkiran et al. (2021), Lafon and Thomas (2024).

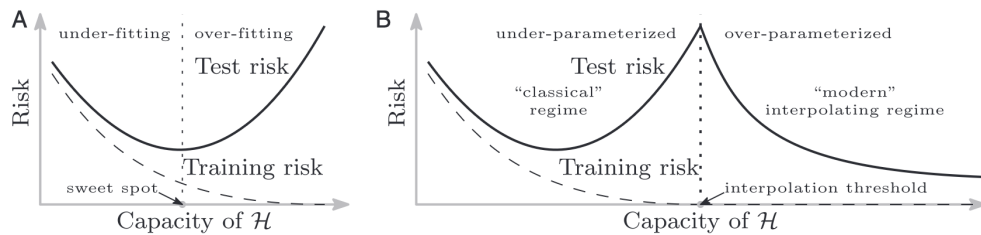


Figure 1–5: Figure (A) shows bias-variance trade-off to which authors refer as 'Classical' regime. Figure (B) shows that given large capacity of a model, training loss becomes zero-close and test loss starts to descent again to even lower value than optimal point in Classical regime. This second regime author refer as 'Modern' or 'Overparametrized' regime. Courtesy of Belkin et al. (2019).

Hyperparameters, such as the capacity of a neural network, learning rate, or amount of regularization are configuration settings of the model that have influence on training and performance. The process of selecting optimal hyperparameters is referred to as neural architecture search Elsken et al. (2019).

Widely used approach for neural architecture search is grid search, in which all possible combinations of specified hyperparameters values are systematically explored. However, the computational cost of grid search grows exponentially with the number of hyperparameters. Moreover, Bergstra and Bengio (2012) demonstrated that grid search often wastes computational resources by tuning less important hyperparameters. Authors proposed random search, which samples hyperparameter values from predefined probability distributions instead of exhaustively testing all combinations. This approach enables the exploration of a broader search space and has been shown to identify better-performing hyperparameters with fewer trials, thereby reducing computational resources.

We have showed how we can calculate gradient and update weights in neural network over entire training dataset or each training data point. However, individual points of the data might be mislabeled or be outliers, thus making training unstable. Furthermore, computing gradients for each training data point is time consuming. Updating weights on loss over entire dataset at once makes slower updates and is memory demanding. To balance those approaches, gradient computation is often performed on mini-batches of data points from the training set. This approach has several advantages: it stabilizes training by averaging over multiple samples and reduces computation time by leveraging data parallelism.

Batch size is frequently included in hyperparameter optimization. However Shal-lue et al. (2019), argues that this practice is wrong, and provides evidence that model can achieve comparable performance across different batch sizes, as long as other hyperparameters are appropriately tuned for each batch size.

A recommended practice for deep learning practitioners is to first determine the largest batch size that fits within the available memory constraints. Using the

largest feasible batch size allows the model to train faster by requiring fewer gradient updates to converge. Once an optimal batch size is identified, other hyperparameters should be tuned accordingly to maximize model performance.

Initialization of the weights is also important. A simple approach is to initialize weights using random values drawn from a Normal distribution with zero mean and a small standard deviation, to have small weights values and therefore small gradients that provide stable convergence. However, this method does not take into consideration the number of input and output connections of neurons.

Glorot and Bengio (2010) argues that neurons with fewer input connections are more sensitive to changes in weights than neurons with a larger number of inputs. This difference can lead to imbalances in gradient magnitudes during backpropagation and might saturate neurons that use tanh or sigmoid activation function, which results in near-zero gradients and slows training. Therefore, they propose initialization of weights drawn from normal distribution with zero mean and standard deviation as $std = \sqrt{2/r_{in} + r_{out}}$, where r_{in} is number of input connections and r_{out} denotes number of output connections.

1.6 Convolutional Neural Networks

Convolutional neural networks (CNNs) were designed to process grid-like data structures, such as images. Their effectiveness was first demonstrated by LeCun et al. (1989) in the task of handwritten postal zip codes recognition. CNNs gained further recognition with its application to document recognition LeCun et al. (1998). However, they achieved widespread dominance after Krizhevsky et al. (2012) efficiently used more compute from GPUs and reported unprecedented performance on the large-scale ImageNet dataset.

CNNs consist of layers applying $n \times m$ filter ¹ sliding over whole input data in vertical and horizontal dimension using convolution operation. Convolution in this

¹In literature also mentioned as kernel

setting basically means that $n \times m$ input data values are multiplied with filter element wise and summed up to output one scalar value. After sliding over entire input data and getting scalar values, they form feature map. Size of resulting feature map can be adjusted with size of filter's stride and padding around input data. Values in filter are what are parameters to be learned by backpropagation. Resulting feature maps are than capable of recognizing specific edges, corners or entire objects on input data (image) Zeiler (2014). Convolution layer can be described as:

$$O(i, j) = \sum_{m=1}^M \sum_{n=1}^N F(m, n) I(i + m, j + n) \quad (1.19)$$

where, I is input image and F is filter, (i, j) are indices of the output feature map O and (m, n) are indices over the filter.

Feature maps are then passed through a non-linear activation function. To reduce the computational complexity and dimensionality of the feature maps, max pooling operation is used. Max pooling divides input into small regions (e.g. 2x2) and outputs maximum value from each region. This effectively summarizes the dominant features in small areas of the image. In this layer, no trainable parameters are present. After max pooling, the reduced feature maps are then flattened into a one-dimensional vector and passed to the fully connected layer, which is basically MLP described in 1.1. CNN is shown in figure 1–6.

Feature maps can be normalized across batch. Batch Normalization, proposed by Ioffe (2015), ensures that the inputs to each hidden layer maintain a consistent distribution throughout training. It was shown that it enables larger learning rates and increases generalization accuracy. The formula for Batch normalization is as follows:

$$\mu_B = \frac{1}{b} \sum_{i=1}^b (x_i) \quad (1.20)$$

$$\sigma_B^2 = \frac{1}{b} \sum_{i=1}^b (x_i - \mu_B)^2 \quad (1.21)$$

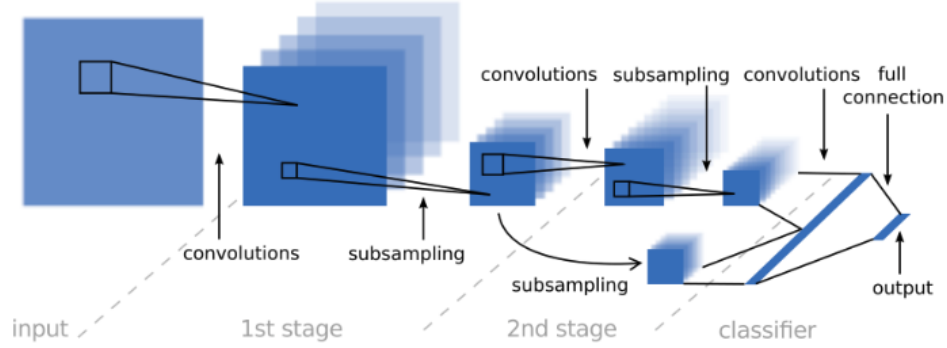


Figure 1 – 6: A Convolutional Neural Network consisting of two sequential convolutional and pooling layers, followed by a flattening operation that transforms the 2D feature map into a 1D representation, which is then processed by a fully connected classifier. Taken from Sermanet et al. (2012).

$$y_i = \gamma \frac{x_i - \mu_B}{\sqrt{\sigma_B^2 + \varepsilon}} + \beta \quad (1.22)$$

where x_i is image in batch represented as tensor with three dimensions: channels, width, height, b is the number of samples in the batch, μ_B is mean of input features across the batch, σ_B^2 is variance, y_i is result of BN where γ and β are learnable parameters that allow model to scale and shift normalized values to learn optimal distribution. Batch normalization is applied independently to each channel of x_i .

Convolution is commonly used only for one image input (e.g. image classification). In cases of sequences images, to capture these not only spatial but temporal dependencies in sequence, Time Distributed Layer approach is used. This layer applies convolutional operations to each image in the sequence and produces feature maps for each image in sequence, maintaining the temporal structure of the image sequence. Once the feature maps are obtained, they are typically flattened and passed to a recurrent layer such as an LSTM, or GRU.

1.7 Recurrent Neural Networks

Feed forward networks, as discussed earlier, identify patterns in large amount of data. However, certain types of data exhibit inherent temporal structures or sequential dependencies. Examples include time-series data and natural language. Consider the following two sentences:

- The student finished the thesis.
- The thesis finished the student.

Although these sentences contain the same words, their differing word order results in distinct meanings.

Recurrent neural networks (RNNs) are designed to process sequential data, where the current state of parameters depends on the previous one. Unlike feedforward networks, RNNs share weights across different time steps, allowing information to flow through the input data rather than assigning separate hidden parameters to each item in inputted sequence. Training RNNs has been proven to be difficult due to emerging issues such as vanishing and exploding gradients Bengio et al. (1994). Long Short-Term Memory (LSTM) Hochreiter (1997) and Gated Recurrent Unit (GRU) Cho (2014) are RNN architectures address this problem. GRUs, in particular, use fewer parameters than LSTMs, making them faster to train Chung et al. (2014), which is why we used them in this work. GRU cell can be noted as following:

$$z_t = \sigma(W_z x_t + U_z h_{t-1} + b_z) \quad (1.23)$$

$$r_t = \sigma(W_r x_t + U_r h_{t-1} + b_r) \quad (1.24)$$

$$\tilde{h}_t = \tanh(W_h x_t + U_h (h_{t-1} \odot r_t) + b_h) \quad (1.25)$$

$$h_t = z_t \odot \tilde{h}_t + (1 - z_t) \odot h_{t-1}, \quad (1.26)$$

where z_t is update gate that controls how much of the previous h_{t-1} state of parameters should be passed through a sigmoid function σ to the next state. Reset gate r_t determines how much of previous state should be ignored in candidate hidden state \tilde{h}_t . In the final hidden state computation h_t is decided, how much of the previous state should be retain for current time step. W_z, W_r, W_h are parameters of inputs at time step x_t and U_z, U_r, U_h are parameters of previous state h_{t-1} . These parameters are optimized. For the initial step, hidden state h_0 is set to zeros. GRU cell is shown in figure 1–7.

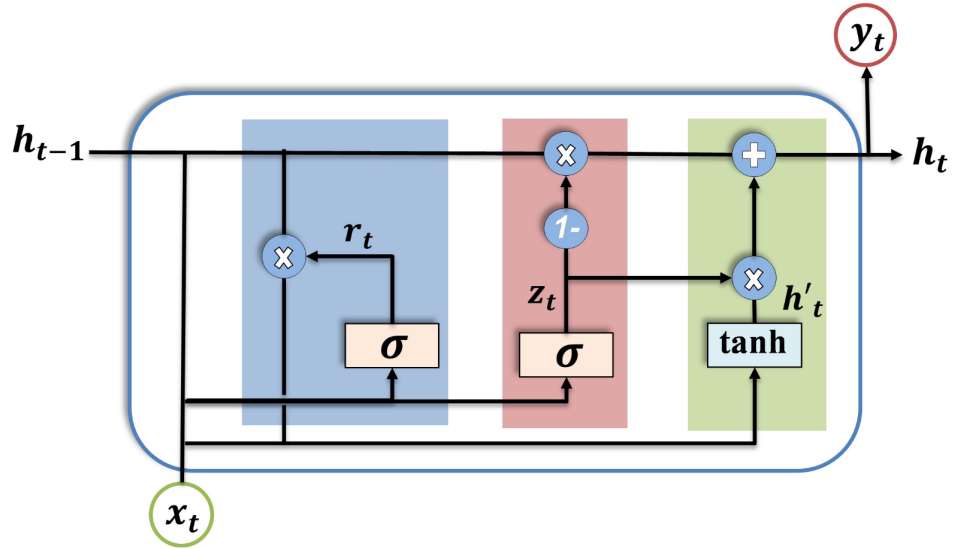


Figure 1 – 7: Structure of GRU cell. Taken from Abboush et al. (2023).

2 Space Weather

Modern society depends on technologies for its critical infrastructure such as power grids, communication systems and global transportation. Space weather events (SWE) occurs in interplanetary space between Sun and Earth and can have negative impacts on Earth and Space infrastructure (Hapgood et al., 2021).

Numerous SWE phenomena are known, but for this work, it is sufficient to be familiar with these basic ones:

- **Active region** is an area in Sun's corona with intense magnetic fields and they appear as bright areas on images of solar corona. These areas are known to be source of Solar Flares.
- **Solar flare** is sudden intense emission of energy in Sun's atmosphere. They can be recognized as bright flash visible in image sequence. In the form of electromagnetic radiation and high-energy particles, a strong eruption can sometimes cause rapid and significant changes in the environment near Earth.
- **Solar wind** is a continuous stream of charged particles ejected from the Sun. Its velocity can range from 400 to 1000 km/s or above. When interacting with Earth's magnetosphere, it can lead to geomagnetic storms.
- **Coronal mass ejection (CME)** is explosion of coronal mass into the surrounding space, consisting primarily of electrons, protons, and heavier ions. It is associated with a solar flare, which can produce a CME, but this is not always the case. CMEs propagate within the solar wind and shockwave accelerates its particles.
- **Corotating Interaction Region (CIR)** are structures in the heliosphere that rotate with the Sun. They emerge when high speed solar wind streams, overtake preceding slow speed streams. This interaction increases plasma density and magnetic field strength. CIRs can persist for multiple solar rotations.,

and when CIRs reach Earth, their southward-directed interplanetary magnetic field component can reconnect with the geomagnetic field, facilitating the transfer of energy into the magnetosphere causing geomagnetic storms.

- **Geomagnetic storms** are major disturbances in Earth’s magnetosphere caused by the interaction of solar wind’s southward magnetic field component with Earth’s geomagnetic field. Reconnection then transfers energy from the magnetosphere into the Earth’s atmosphere.

The most well-known case where space weather affected our technology is the Carrington Event of 1859 (Giegengack, 2015). A geomagnetic storm caused by a CME disrupted the emerging telegraph network. In 1989, a geomagnetic storm caused a power plant outage in Quebec, leaving up to 6 million people without electricity for 9 hours (Allen et al., 1989). In 2003, a geomagnetic storm caused 50,000 people to lose power in Sweden and damaged 12 transformers in Africa (Cid et al., 2014). According to Dang et al. (2022) in February 2022, two geomagnetic storms caused by solar activity and the passage of CMEs increased atmospheric drag, leading to the loss of 38 Starlink satellites.

Although the upcoming solar cycle 25 was not expected to be stronger than the previous 24th cycle, as highlighted by Penza et al. (2023), there is still a risk of solar flares causing CMEs directed toward Earth that can emerge at any time. Recently, the intense ‘Mother’s Day’ geomagnetic storm in May 2024 Spogli et al. (2024), one of the strongest in the past 30 years, highlighted this ongoing risk. Therefore, monitoring and predicting space weather is already important for protecting existing infrastructure, and its importance will continue to grow for future human-crewed space missions.

2.1 Solar Activity Monitoring Missions

Over the past decades, Space Agencies have spent much effort for monitoring solar activity, due to its critical importance for modern society. Data obtained from these

activities can help scientists understand the specific mechanics and triggers of solar weather. These data have been collected through a series of missions, which can be conceptualized as successive generations of solar activity monitoring efforts.

1. Generation: **The Solar and Heliospheric Observatory (SOHO)**, launched in 1995. SOHO is equipped with 12 instruments that capture solar activity. They can be divided into 3 categories: instruments for helioseismology, solar corona investigation instruments and instruments for in-situ space wind measurements (Domingo et al., 1995). Positioned at Lagrange Point 1 (L1), SOHO maintains an uninterrupted view of the Sun.
2. Generation: **Solar TERrestrial RELations Observatory (STEREO)** launched in 2006, consisted of two spacecraft: STEREO A (Ahead) and STEREO B (Behind). These spacecraft orbited the Sun, with one positioned ahead of Earth and the other trailing behind (Kaiser et al., 2008). This unique trajectories enabled scientists to obtain a 3D view of the Sun from two angles. However, STEREO B was lost in 2014 and contact could not be reestablished. In August 2023, STEREO A completed a full orbit around the Sun and once again crossed Earth's orbital trajectory.
3. Generation: **Solar Dynamics Observatory (SDO)** launched in 2010. The SDO satellite is equipped with 3 instruments: Atmospheric Imaging Assembly (AIA), Extreme Ultraviolet Variability Experiment (EVE) and Helioseismic and Magnetic Imager (HMI) (Pesnell et al., 2012). These telescopes capture solar activity in better resolution than SOHO, therefore it is considered the successor to the SOHO satellite. Since its instruments have a high cadence and better quality, they generate a lot of data suitable for machine learning tasks.
4. Generation: **Geostationary Operational Environmental Satellite (GOES) Series**. GOES satellites are placed in geosynchronous orbit, which keeps them above a specific location on the Earth's surface. The first GOES satellite was

launched in 1975 and monitored Earth’s weather. The latest GOES-R series consists of four satellites that provide imagery and atmospheric measurements of Earth’s weather (Goodman et al., 2019). In addition to these objectives, GOES satellites also have the Solar Ultraviolet Imager (SUVI), which is used to observe the Sun’s corona. R-series satellite launch dates:

- GOES-16: November 2016
- GOES-17: March 2018
- GOES-18: March 2022
- GOES-19: June 2024

The satellites are scheduled to operate until 2040.

5. Generation (Planned): **Vigil** scheduled for launch in 2031. The Vigil mission will monitor solar activity and provide warnings of space weather events that could threaten critical infrastructure on Earth. Located at Lagrange point 5 (L_5), Vigil will orbit the Sun at an angular offset of 60° from Earth’s position. This will allow Vigil to image an additional 60° of the solar disc that is not visible from Earth. This additional view can detect potentially dangerous solar activity 4 days before it is visible from Earth (Vourlidas, 2015). Another advantage of the L_5 point is that, thanks to its heliospheric imager, Vigil can observe the propagation of CMEs in space between the Sun and Earth. The L_5 point is shown in the figure 2–1. Vigil will produce a huge amount of data on solar activity. Therefore, automated knowledge extraction will be key to identifying potential trends, anomalies, and even discoveries within this data. Machine learning seems to be a useful tool for this task.

The main payload of the Vigil mission will consist of 6 instruments which are described in Table 2–1. To illustrate the similarity of the categories of Vigil’s instruments and already operated missions’ instruments, we have created a matrix

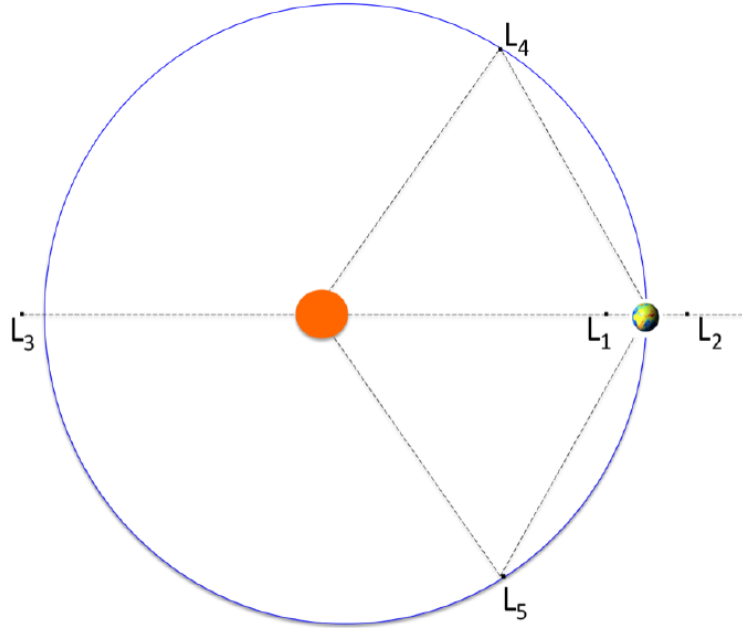


Figure 2 – 1: Lagrange points in the two-body Sun-Earth system, according to Lo et al. (2010).

presented in Table 2–2. Notably, the Vigil instruments are currently in development. The exact characteristics will be provided once the construction and testing are finalized. However, for now, it is expected that the Vigil will have the ability to monitor SWE phenomena within their pathway from Sun to Earth. The solar magnetic field in the photosphere will be measured by a Photospheric Magnetic field Imager (PMI) Staub et al. (2020). Structures in the solar corona will be imaged by a Joint EUV coronal Diagnostic Investigation (JEDI). CMEs will be captured by a Compact Coronagraph (CCOR) and their propagation in space will be tracked by a Heliospheric Imager (HI) Tappin et al. (2023). The interplanetary magnetic field at L5 point will be measured by Magnetometer (MAG) Eastwood et al. (2024) and solar wind features like Stream and Corotating Interaction Regions (SIRs and CIRs) will be detected in-situ by Plasma Analyser (PLA) Zhang et al. (2024). It is important to consider that all Vigil outcomes will be orchestrated with outcomes from missions on the Sun-Earth line. It is expected, that multi-perspective coronagraph observations for modeling the 3D structure of CMEs, extended magnetogram

coverage of the solar surface, and in-situ measurements of the ambient solar wind before its rotating streams become geoeffective, would be essential outputs.

Table 2 – 1: Table of planned Vigil instruments.

Instrument	Purpose
Photospheric Magnetic field Imager (PMI)	Is designed to provide vector magnetic field mapping of the solar photosphere.
Joint EUV coronal Diagnostic Investigation (JEDI)	Is designed to image solar corona.
Compact Coronagraph (CCOR)	Is designed to perform solar coronagraphy needed for identification of CME evolution.
Heliospheric Imager (HI)	Is designed to provide the tracking of the propagation of CME in space by heliospheric imaging.
Magnetometer (MAG)	Is designed to measure in-situ vector of Interplanetary Magnetic Field.
Plasma Analyser (PLA)	Is designed to measure in-situ solar wind density, temperature, and velocity.

Table 2 – 2: The matrix of space missions instruments that are similar to the category of Vigil’s instruments.

Vigil	SOHO	SDO	SoI/O	STEREO	GOES	WIND	ACE	DSCOVR
PMI	MDI	HMI	PHI					
JEDI	EIT	AIA	EUI	EUVI	SUVI			
CCOR	LASCO			COR	CCOR			
HI			HI	HI				
MAG			MAG	IMPACT		MFI	MAG	MAG
PLA	CELIAS		SWA	PLASTIC		SWE	SWEPAM	FC

2.2 Machine Learning in Space Weather

The idea of applying neural networks to data from the space weather domain appeared as early as 1988 (Fozzard et al., 1988). At that time, a three-layer neural network

was able to match a rule-based expert system in predicting the classification class of a solar flare.

With the increase in computer power and the amount of data available, along with new algorithms that are easier to use thanks to abstract tools and libraries, the solar weather community has become more interested in and exploring the benefits that machine learning can bring to understanding and predicting solar phenomena and their impacts (Camporeale, 2019), (Lilensten et al., 2021). According to (KA et al., 2023), this growing interest is also visible in the number of total publications on the use of machine learning in this domain, especially after 2018 and still growing. Figure 2–2 shows the number of annual publications.

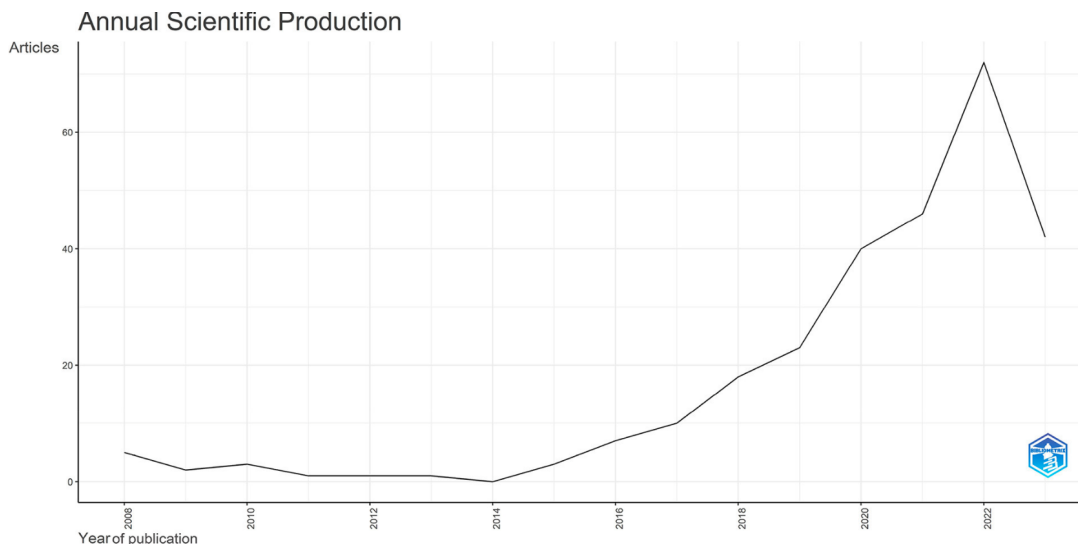


Figure 2 – 2: Annual number of publications dedicated to solving machine learning problems in the space weather domain, according to KA et al. (2023).

2.3 State of The Art for Dst Prediction

A common parameter to measure geomagnetic disturbances is the Disturbance storm time index (Dst index) (Sugiura and Wilson, 1964). Dst index represents a measure of the average intensity of the Earth’s external geomagnetic field, derived from data on north-south magnetic field fluctuations recorded by four strategically positioned

ground-based magnetometers along the geomagnetic equator: Hermanus, Honolulu, San Juan, and Kakioka (Sugiura and Kamei, 1991). Although some researchers argue that the Dst index alone is insufficient to fully characterize extreme space weather events (Borovsky and Shprits, 2017), Dst index prediction is an extensive area of research within the Space Weather community with many publications dedicated to it. Two main approaches are Physics-based models and Machine Learning techniques. In our work, we focus on the latter-mentioned approach.

Lundstedt (1991) was one of the first to predict the Dst index using feed-forward neural networks with one one-hour lead time. Jankovičová et al. (2002) used Principal Component Analysis to reduce input parameters for NN. Bala and Reiff (2012) proposed an improved operational forecast of Dst running in real-time. Lazzús et al. (2017) proposed NN, where weights were optimized by the Particle Swarm Optimization algorithm. The Bagging Ensemble learning method was proposed by Xu et al. (2020), combining NN, SVR, and LSTM network. Park et al. (2021) presented an operational model to predict the Dst index combining NN and Empirical model. This model was trained on geomagnetic storms data, with the distinction between storms driven by CME and CIR. Recently, Hu et al. (2022) presented CNN trained on 51 geomagnetic storms to predict the probability of $Dst < -100$ nT one day ahead using remote-sensing data from SOHO mission. Later, Hu et al. (2023) demonstrated a model trained on 66 geomagnetic storms, employing a Gated Recurrent Unit (GRU) to predict the Dst index up to 6 hours in advance. The model incorporated uncertainty estimation and a boosting method to improve accuracy.

We see two possible ways to engage in the current discussion on Dst prediction within this work. First, a community already knows what in-situ measurements are useful as input parameters to predict the Dst index (Lethy et al., 2018), but only a handful of works e.g. Hu et al. (2022) incorporated remote-sensing data for Dst prediction. Second, it is important to address delayed forecasts, which commonly occur in Dst index prediction, as it is a time-series forecasting task. While the Dst index generally remains stable, on rare occasions caused by geomagnetic storms,

the Dst index can change by a large margin in a very short time. Those rare rapid fluctuations are indeed difficult to predict accurately and what seems to be reported as acceptable conventional metric value (R^2 or RMSE) may not fully reflect true prediction quality or usefulness, since prediction is not on time. This latency often appears as a shift between predicted and actual values when plotted together, indicating a delay in the prediction's responsiveness to sudden changes.

3 Dst Prediction

Work of this project was done using Cross-Industry Standard Process for Data Mining (CRISP-DM) methodology (Paralič, 2003). It provides an organized approach to uncover insights and discoveries from raw data. CRISP-DM follows six main phases: Business understanding, data understanding, data preparation, modeling, evaluation, and deployment. Since this study was conducted with scientific exploration objectives, we refer to the initial CRISP-DM phase as Scientific Understanding.

3.1 Scientific Understanding

The accurate and timely prediction of the Dst index is critical, as it serves as a measure of Earth’s geomagnetic state. During geomagnetic storms, disruptions to the Dst index have been linked to reduced GPS accuracy and the potential for damage to critical infrastructure as discussed in detail in chapter 2. Dst index is a continuous numerical variable and deep learning provides a promising approach to create deep learning model, that can predict numerical value y several timesteps T ahead. This is referred to as a time-series regression task and it is used in many domains.

Dst prediction was addressed by numerous studies (see chapter 2.3), highlighting the significance of this topic within the space weather research community. Many prior works have visualized Dst prediction over extended time periods spanning months or years, during which Dst is dominantly stable near zero, with occasional fluctuations. While such predictions visualizations provide an overview of model performance, it often overlook the critical need for timely and precise forecasting of sharp Dst decreases, which are the primary markers of geomagnetic storms. Focus on these extended sequences can lack of granularity needed to evaluate accuracy and timeliness of Dst prediction during sudden and sharp changes. As a result, validating models performance on such sequences may report seemingly acceptable performance metrics while failing to detect the model’s inability to predict sharp

Dst decreases in a timely manner, a crucial capability for practical applications. This has been observed and reported in numerous works (Kugblenu et al., 1999; Stepanova and Pérez, 2000; Wintoft and Wik, 2018; Zhang et al., 2023), where it often appears as a shift between predicted and actual values when plotted together.

Significant decreases in the Dst index represents geomagnetic storms and are relatively rare, typically occurring during extreme SWE events. According to classification thresholds established by Loewe and Prölss (1997), we define severe geomagnetic storms as $-200nT > Dst > -350nT$ or $179 \leq Ap \leq 300$, while great storms correspond to $-350nT > Dst$ or $300 < Ap$. Over the past 30 years, only 39 such severe and great storms have been recorded. These identified storms are outlined in the Table 3–1.

Table 3–1: Table of identified great and severe geomagnetic storms based on the Dst and ap indices. The event start time represents the first moment when one of the parameters is below (above) the threshold for determining an extreme geomagnetic storm. The event end time represents the time when all of the parameters did not exceed the threshold in the following at least two hours. For each time interval, the most extreme parameter values are displayed.

No.	Start	End	Dst	ap	Class	Class
					Dst	ap
1	1998-05-04 03:00	1998-05-04 08:00	-205	300	severe	severe
2	1998-08-26 22:00	1998-08-27 08:00	-148	207		severe
3	1998-09-25 00:00	1998-09-25 09:00	-207	236	severe	severe
4	1998-08-11 03:00	1998-08-11 05:00	-140	179		severe
5	1999-09-22 21:00	1999-09-22 23:00	-173	207		severe
6	1999-10-22 03:00	1999-10-22 08:00	-237	207	severe	severe
7	2000-04-06 18:00	2000-04-07 04:00	-292	300	severe	severe
8	2000-05-24 00:00	2000-05-24 05:00	-129	207		severe
9	2000-07-15 12:00	2000-07-16 05:00	-300	400	severe	great

Continued on next page

Table 3 – 1 – continued from previous page

No.	Start	End	Dst	ap	Class	ap
					Dst	
10	2000-08-12 03:00	2000-08-12 11:00	-234	179	severe	severe
11	2000-09-17 21:00	2000-09-18 02:00	-201	236	severe	severe
12	2001-03-31 03:00	2001-04-01 02:00	-387	300	great	severe
13	2001-04-11 15:00	2001-04-12 02:00	-271	236	severe	severe
14	2001-11-05 23:00	2001-11-06 16:00	-292	300	severe	severe
15	2001-11-24 06:00	2001-11-24 08:00	-92	236		severe
16	2001-11-24 12:00	2001-11-24 18:00	-221	179	severe	severe
17	2002-05-23 11:00	2002-05-23 17:00	-109	236		severe
18	2003-05-29 15:00	2003-05-29 23:00	-144	236		severe
19	2003-10-29 06:00	2003-10-30 07:00	-353	400	great	great
20	2003-10-30 17:00	2003-10-31 05:00	-383	400	great	great
21	2003-11-20 12:00	2003-11-20 02:00	-422	300	great	severe
22	2004-07-27 00:00	2004-07-27 17:00	-170	300		severe
23	2004-11-07 21:00	2004-11-08 11:00	-374	300	great	severe
24	2004-11-09 18:00	2004-11-10 13:00	-263	300	severe	severe
25	2005-01-07 21:00	2005-01-07 23:00	-71	179		severe
26	2005-01-18 01:00	2005-01-18 08:00	-103	179		severe
27	2005-01-21 17:00	2005-01-21 20:00	-25	207		severe
28	2005-05-15 06:00	2005-05-15 11:00	-247	236	severe	severe
29	2005-08-24 07:00	2005-08-24 11:00	-184	300		severe
30	2005-09-11 06:00	2005-09-11 08:00	-119	179		severe
31	2006-12-14 21:00	2006-12-14 05:00	-159	236		severe
32	2011-08-05 21:00	2011-08-05 23:00	-96	179		severe
33	2012-03-09 03:00	2012-03-09 12:00	-145	207		severe
34	2013-10-02 02:00	2013-10-02 05:00	-39	179		severe
35	2015-03-17 21:00	2015-03-17 23:00	-234	179	severe	severe
36	2015-06-22 18:00	2015-06-22 20:00	-114	236		severe

Continued on next page

Table 3 – 1 – continued from previous page

No.	Start	End	Dst	ap	Class	ap
Dst						
37	2017-09-07 21:00	2017-09-08 02:00	-122	207		severe
38	2023-04-24 04:00	2023-04-24 06:00	-213	20	severe	
39	2024-05-10 21:00	2024-05-11 19:00	-412	40	great	

The average duration of a geomagnetic storm and the associated sharp decrease of Dst values typically spans one to two days. This limited duration and frequency result in a relatively small dataset of high-impact events, that might be a challenge for deep learning model. However, studies Hu et al. (2022, 2023) have trained models exclusively on data from extreme events and reported interesting results despite the constrained dataset size.

3.1.1 Constraints

To develop a effective Dst prediction model, it is essential to obtain and process data from the most extreme space weather events possible. Each mission observing solar activity discussed in chapter 2 is a potential data source. However, the data sources must satisfy three critical constraints:

- **Similarity constraint:** Vigil is designed to warn Earth of hazardous solar activity, particularly during extreme events. Developing Dst prediction model with data that resemble expected Vigil observations is significant step toward facilitating the rapid deployment of machine learning models once Vigil becomes operational. It is impossible to guarantee that future data from Vigil will perfectly match, heritage instruments from prior solar missions can serve as valuable a-like data. We outlined heritage instruments in table 2–2. Once Vigil will be operational, transfer learning can be applied to make use of its data.

-
- **Time constraint:** most of the extreme SWE events occurred around the years 1998–2006, see Table 3–1. This introduces a time constraint that must be considered when selecting data sources. We should avoid using data from instruments that were not operational during this period, as such data would miss relevant information to the prediction of Dst index during geomagnetic storms.
 - **Data Content constraint:** current state-of-the-art models for predicting Dst primarily rely on in-situ data, such as measurements of Z-component of magnetic field (B_z), solar wind speed (V_p) and density (N_p). It is essential to incorporate instruments that provide these data. However, Hu et al. (2022) demonstrated that using only remote sensing data, specifically images from the SOHO mission, yielded promising results. This raises an question: Can remote sensing data, such as Extreme Ultraviolet (EUV) and Coronagraph images, provide sufficient information to predict the Dst index? Furthermore, could integrating these image-based data with in-situ measurements improve model's performance? Investigation of combination these two data types may offer a novel approach to improving prediction accuracy.

To address all identified constraints, we should prioritize data from the WIND and SOHO missions, as both were operational at the time when most extreme SWE events occurred. These missions provide needed in-situ and remote sensing observations that align with the similarity, time, and data content requirements. We will use data from WIND's instrument Magnetic Field Investigation (MFI), that measures Magnetic field Z-component (B_z) at L1 point. SOHO will provide measurements of instrument Charge, Element, and Isotope Analysis System (CELIAS), which provide solar wind speed (V_p) and solar wind density (N_p).

Furthermore, SOHO will provide images from Extreme ultraviolet Imaging Telescope (EIT) instrument, that captures images of the solar corona in four different wavelengths: 171 Å, 195 Å, 284 Å, 304 Å. Wavelength 195 Å is useful to observe coro-

nal holes. According to Rotter et al. (2012) coronal holes have impact on solar wind parameters.

We will also incorporate data from SOHO’s Large Angle and Spectrometric Coronagraph (LASCO) instrument, which provides comprehensive visual coverage of surrounding solar atmosphere. This instrument can capture coronal mass ejections (CMEs), and we believe this can enhance the predictive accuracy of the model predicting Dst index.

Following the launch of the Solar Dynamics Observatory (SDO) mission in 2010, Atmospheric Imaging Assembly (AIA), was recognized as an enhanced successor to the EIT onboard the Solar and Heliospheric Observatory (SOHO). Therefore, the operational cadence of SOHO’s EIT instrument was reduced. When obtaining data post-2010, we should primarily focus on utilizing the SDO AIA instrument for EUV images.

3.1.2 Concerns

The primary concern is whether enough data of geomagnetic storms exist for training deep learning models. For instance, if each extreme SWE event that triggered a severe or great geomagnetic storm included only one rapid Dst descent, there would be merely 39 short time windows containing these sharp Dst oscillations, which is only handful. However, mentioned studies Hu et al. (2022, 2023) successfully trained models with datasets containing 51 and 66 solar storms, respectively, and achieved notable results. Therefore, this limitation may not be as critical as initially anticipated.

Geomagnetic storms can be caused by CMEs or CIRs (see chapter 2), which are characterized by distinct solar wind parameters. CMEs may not be observed in coronagraph images. Thus, a second concern arises: the limited number of geomagnetic storms data available for analysis may exhibit significant variability by nature, making it challenging for models to learn consistent patterns on training data. Investigation of the data should include an analysis of the distributions associated

with each geomagnetic storm to determine whether they are similar for meaningful model interpretation.

To better capture the evolution of geomagnetic storms, data could include several days prior to the storm’s peak as well as the several days after, capturing recovery period. While the mentioned concern about limited important data remains, consider that remote-sensing instruments with a cadence of approximately 20 minutes over a span of a few days can generate thousands of images for all of the 39 events. However, only a handful fraction of these images may capture CMEs related with geomagnetic storms. This sparsity of relevant images creates a challenge for analysis and handling thousands images. Therefore, methodology should be proposed for an easy incorporation of multi-modal data to the model.

3.1.3 Plan

The work will follow a structured approach to investigate whether multimodal data enhances deep learning models for Dst index prediction, prioritizing novel predictive strategies over incremental improvements to the state of the art. We identify the most comparable existing instruments to those planned for the future Vigil mission and collect data from these sources and integrate them into joint dataset. To ensure a robust evaluation, we develop a framework for correctly partitioning extreme space weather events. We implement various deep learning models that integrate time series and image data to capture complex dependencies. Finally, we evaluate the models using predefined metrics to assess their predictive performance and report outcomes.

3.1.4 Goals

The project is considered successful upon achieving the following objectives:

- Develop and integrate a comprehensive dataset combining time series and image-based representations of solar activity, thereby capturing the multiface-

ted nature of space weather phenomena.

- Rigorously compare the performance of our novel deep learning architectures against established models in the space weather community, demonstrating superior predictive accuracy both quantitatively (via standard metrics) and qualitatively (through visual assessment of forecast outputs).
- Contribute our insights and novelty to the field of space weather in form of scientific publication in peer-reviewed journal.

3.2 Data Understanding

We have developed a lightweight pipeline in Python to download images and in-situ measurements from these instruments. Specifically, the pipeline supports the acquisition of EUV images (195 Å) from SOHO/EIT, coronagraph images from SOHO/LASCO C2 and C3, proton speed and proton number density from SOHO/CELIAS, and interplanetary magnetic field measurements from WIND/MFI.

This developed pipeline is available online on our GitHub repository SPACE::LAB et al. (2024). The Python scripts are organized into classes and methods, each dedicated to managing data downloads for a specific instrument. Therefore, users only interested in pipeline functionality for data from a single instrument can focus on the relevant part of the code without having to read the entire codebase. The script uses a COSPAR-recommended standard for time series SWE data delivery - the Heliophysics Application Programmer's Interface (HAPI) Weigel, Vandegriff, Faden, King, Roberts, Harris, Candey, Lal, Boardsen, Lindholm et al. (2021). It also utilizes the *hapiclient* Python package Weigel, Batta, Faden and jvandegriff (2021). Additionally, the script employs the *humpy* Python package, which provides a high-level interface to the Helioviewer API ². However, we observed that EIT (195 Å) images obtained via *humpy* are occasionally blurred, indicating a reduction in image quality.

²<https://humpy.readthedocs.io/en/latest/>

To ensure higher fidelity, we retrieved EIT images from the Virtual Solar Observatory (VSO) platform³, which provides same images without defects. The images were acquired in the Flexible Image Transport System (FITS) file format, a widely adopted standard in the astronomical community for data storage and analysis⁴.

Additionally, we have prepared Python script to obtain SDO/AIA (193 Å) images also from *hupy* Python package.

To make yet unprocessed dataset complete, we collected hourly values of Dst from the World Data Center for Geomagnetism, Kyoto website⁵.

Mentioned pipeline with supporting scripts was used to download data for the most extreme SWE events classified as severe and great storms outlined as in 3–1. Each event’s data range spans 10 days before and after the Dst index minimum. Measurements were acquired at the highest available temporal resolution for each instrument: LASCO provided data at 20-30 minute intervals, while the EIT operated at a 12-minute cadence and AIA provided images every 4 minutes. Solar wind parameters were recorded by the WIND spacecraft’s magnetometer at 1-minute resolution, and plasma measurements from the SOHO’s CELIAS were obtained at 30-second intervals. Target variable, Dst index is measured hourly.

From this point onward, we will refer to data from each extreme SWE event according to its numbering in 3–1. The dataset excludes events 2 and 3 due to the absence of crucial coronagraph images, which are essential for our study. Each event’s data range generally spans 10 days before and after the Dst index minimum. On occasion, when data were missing for entire days, they had been removed, resulting in some events having a shorter time span. Events 16, 18, 20, 24, 26, and 27 were merged with preceding events, as they occurred within a 10-day range of each other. Amount of data for each event from different instruments is shown in figure 3–1. Coronagraph and EUV images take up to 82,3 GB of disk space. Samples of data

³<https://sdac.virtualsolar.org/cgi/search>

⁴<https://docs.astropy.org/en/latest/io/fits/index.html>

⁵<https://wdc.kugi.kyoto-u.ac.jp/dst/dir/>

from each instrument are shown in figure 3–2.

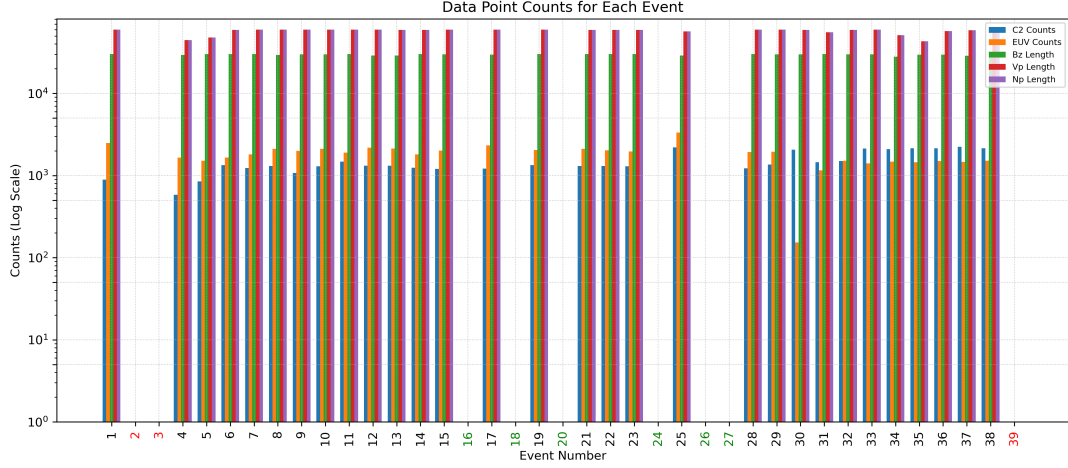
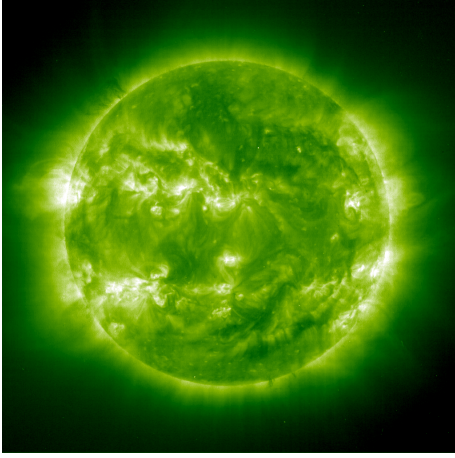


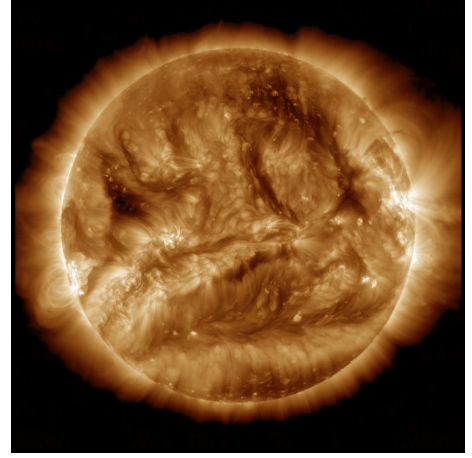
Figure 3–1: Data count for each extreme SWE event by different instruments measuring space weather activity. Events 2, 3, and 39 (highlighted in red) lacks crucial data. Events 16, 18, 20, 24, 26 and 27 (highlighted in green) are merged with preceding events, as they occurred within very close interval.

As the initial analysis of this data, we have investigated locations of solar flares that were, in many cases, the origin of CMEs and then extreme geomagnetic storms. Figure 3–3 (*left*) displays an overlay of EUV images from SOHO/EIT and SDO/AIA, taken during the occurrence of 19 flares out of the 39 events listed in 3–1. In each instance, the flares were observed near the center of the solar disk. However, the distribution of these events is broad, showing no particular location or longitude that can be pinpointed as a primary area for the initiation of extreme SWE events. The absence of nearly half of the extreme SWE events in Figure 3–3 (*left*) can be attributed to missing EUV images for certain dates and times when these flares occurred. Even more, there might be a lack of association between these events and a visible solar flare on the overlay EUV image. For example, solar filament eruption is capable to generate a massive CME but it is not well visible in EUV 195 Å filter.

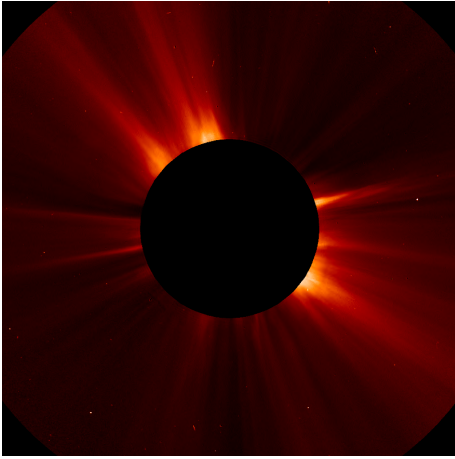
We analyzed the time scales between the time when the SWE event was initiated



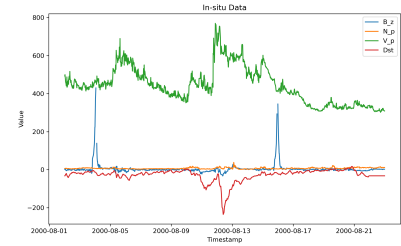
(a) EUV image from EIT 195 Å instrument on board SOHO. Taken on 02.08.2000 at 01:13.



(b) EUV image from AIA 193 Å instrument on board SDO. Taken on 03.07.2015 at 00:59.



(c) coronagraph image from LASCO C2 instrument on board SOHO. Taken on 02.08.2000 at 01:31.



(d) In-situ data of solar wind speed V_p and density N_p from SOHO/CELIAS. Z-component of magnetic field B_z is provided by WIND/MFI. Sample covers event 15. It is visible that in-situ parameters operate in different scales.

Figure 3 – 2: Showcase of data from different instruments used in dataset.

(i.e. visible on SOHO/EIT 195 Å, SDO/AIA 193 Å EUV, and/or SOHO/LASCO C2 images) and the time when the Dst index reached its extreme value. The event initiation observations align with those in work by Zhang et al. (2007). The distribution of time scales for all events from 3–1 is presented in Figure 3–3 (*right*). The mean value for the considered time scales is ≈ 57 hours. The minimal time scale

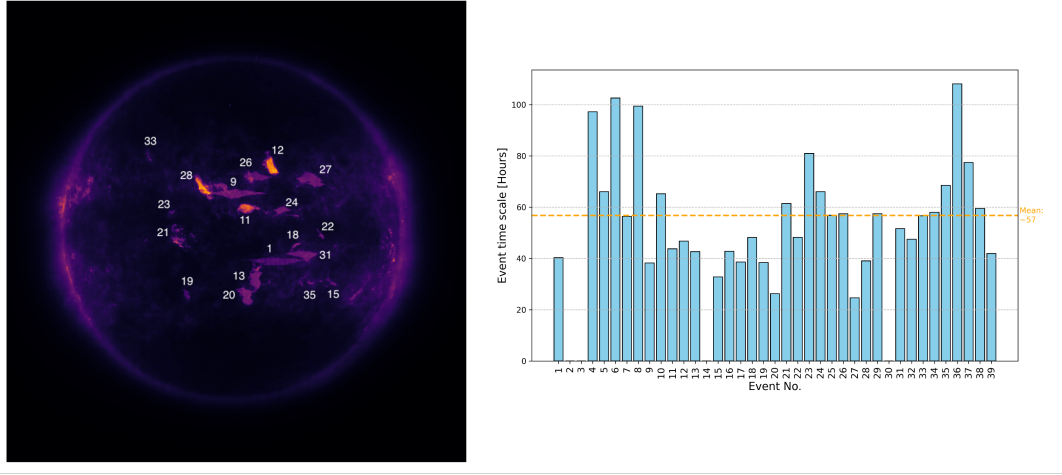


Figure 3–3: *Left:* Overlap of 35 images of moments when the most extreme SWE events were produced, as captured by SOHO/EIT 195 Å and SDO/AIA 193 Å. Visible flares are numbered, corresponding to events listed in Table 3–1. It is noted that science-ready FITS files were used to generate source images. However, the cross-calibration was not considered between the instruments as we are here interested only in the locations of flares and not in their absolute intensity. *Right:* The distribution of time scales for events listed in Table 3–1. Here, the duration is calculated from the time when the event was visible on SOHO/EIT 195 Å, SDO/AIA 193 Å, and/or SOHO/LASCO C2 images until the time when the Dst index reached its extreme value. The four missing time scales (bars) are caused by unavailable images that are needed for the estimation of the event initialization time.

is ≈ 25 hours. This is an interesting result as the time scale is the most important parameter for mitigating the negative consequences of geomagnetic storms. Thus, once the extreme SWE event is initiated, there are only 1 or 2 days to perform preventing actions. On the other hand, the minimal value of ≈ 25 hours is still enough to make proper decisions and appropriate actions.

3.3 Data Preparation

Instruments identified as data sources in this study operate at different temporal cadences. Given our hypothesis that coronagraph images might be valuable supplementary data, we synchronized all data to the temporal sampling rate of the LASCO coronagraph images. Specifically, we used LASCO image timestamps as the primary temporal reference points and resampled data from other instruments to the nearest corresponding timestamps. This ensures that all other measured data are paired with the timestamps of the coronagraph images. The average time interval between consecutive time steps is approximately 23 minutes.

We propose this dataset as Most Extreme Space Weather Events (MESWE) dataset, which covers data from 38 extreme SWE events and includes processed EUV images, processed coronagraph images, and in-situ measurements, totaling 13 attributes, described in Table 3–2. It consist of 30 files in .csv format for each applicable extreme SWE event. Steps for dataset creation are outlined in Figure 3–4.

3.3.1 Processing Multi-modality

Here we present framework to process EIT and Coronagraph images so they can be incorporated in MESWE dataset together with other in-situ measurements.

EUV Images Data

To capture the state of coronal holes during extreme SWE events, we used SOHO/EIT 195 Å images for events before August 2010. After this date, we have used SDO/AIA 193 Å images.

Idea of incorporating information about coronal holes is tied with study by Rotter et al. (2012), where authors shows that coronal holes area are highly correlated with solar wind speed (V_p). Thus, we can investigate if adding information about coronal holes can be helpful to predict Dst index through solar wind speed.

Table 3 – 2: Attributes in MESWE Dataset

Attribute	Description
Q1	Pixels changed (%) in Quadrant 1 of Coronagraph image [%]
Q2	Pixels changed in Quadrant 2 of Coronagraph image [%]
Q3	Pixels changed in Quadrant 3 of Coronagraph image [%]
Q4	Pixels changed in Quadrant 4 of Coronagraph image [%]
N_CH	Coronal holes coverage in the North region of solar disc [%]
R_CH	Coronal holes coverage in the Right region of solar disc [%]
S_CH	Coronal holes coverage in the South region of solar disc [%]
L_CH	Coronal holes coverage in the Left region of solar disc [%]
C_CH	Coronal holes coverage in the Center region of solar disc [%]
V_p	Solar wind speed [km^{-1}]
N_p	Solar wind density [cm^{-3}]
B_z	Southward component of interplanetary magnetic field [nT]
Dst	Dst index values [nT]

To segment coronal holes in EUV images, we applied the SCSS-Net convolutional neural network Mackovjak et al. (2021). We used two separate model weights trained individually on EIT and AIA images as input data and SPoCA CH masks Verbeeck et al. (2014) as ground-truth labels. SCSS-Net takes as a input grayscale solar corona of size 256x256 pixels. Outputs are binary masks, where pixels with a value of 255 represent a coronal hole (CH), and pixels with a value of 0 indicate no coronal hole.

We then divided the solar disc into five regions: North, Right, South, Left, and Center. For each region, we calculated the percentage of coronal hole coverage from the binary mask returned by SCSS-Net with respect to non coronal hole pixels in that region. The resulting percentages of CH pixels in each region are five attributes of MESWE Dataset. They indicate if there are sources of high-speed streams, that could impact the L1 and Earth’s geomagnetic conditions through CIRs. This is commonly observed phenomenon when large CH area is located in the center of solar disc.

For interested readers, in following paragraphs we provide detailed engineering

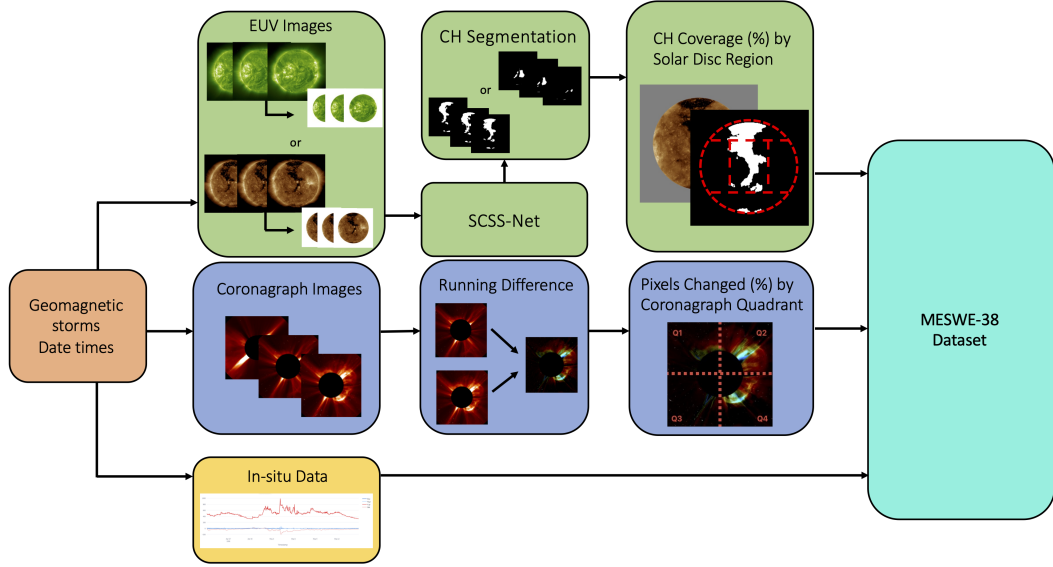


Figure 3–4: MESWE Dataset created by aligning remote-sensing image data with in-situ measurements for extreme SWE events. The dataset consists of 13 attributes, capturing information on coronal hole coverage across regions of the solar disc, running differences calculated in the four quadrants of coronagraph images, in-situ measurements consisting of solar wind speed, solar wind density, South-North component of magnetic field and Dst index values.

approach to this estimation of CH area coverage in each solar disc region. Binary masks (BMs) returned by SCSS-net lack explicit solar disc boundaries, preventing direct computation of CH coverage within specific disc regions. We also could not use hardcoded coordinates of solar disc areas, as the solar disc is not consistently centered in the original EUV images. We used pairs of BMs and original EUV images, but with cropped backgrounds, ensuring that only the solar disc remained visible against a uniform gray background without outer atmosphere.

First, we have defined edges of solar disc. The boundaries of the solar disc were

identified by detecting the transition from the gray background to non-gray pixels. By checking when pixel color was not gray, we determined the first and last rows and columns of disc's edges and stored them. The five regions were then delineated as follows (image coordinates (0,0) starting at upper-left corner):

- **North Region:** Pixels that were not gray and at row position within interval of $[\text{first row}, \text{first row} + (\text{last row} - \text{first row}) * 0.2]$
- **South Region:** Pixels that were not gray and at row position within interval of $[\text{last row}, \text{last row} - (\text{last row} - \text{first row}) * 0.2]$
- **Left Region:** Pixels that were not gray and not in North nor South Region at column position within interval of $[\text{first column}, \text{first column} + (\text{last column} - \text{first column}) * 0.25]$
- **Right Region:** Pixels that were not gray and not in North nor South Region at column position within interval of $[\text{last column}, \text{last column} - (\text{last column} - \text{first column}) * 0.25]$
- **Center Region:** Pixels that were not gray and not in any regions above.

After defining these regions, the binary mask was compared to the solar disc images with regions boundaries. The CH coverage ratio for each region was calculated as the proportion of white pixels on BMs (representing coronal holes) to the total number of non-gray pixels within the corresponding region of solar disc image with cropped background. We are aware that this is not physically correct area of CH due to the Sun's curvature, but it provides a sufficient approximation for our purposes.

We could not use original black background because coronal holes on solar disc are black and we also could not use white background because active regions on solar disc are of white color. Therefore ratios would be smaller proportional to coronal holes or active regions.

These processing steps are shown as green group of actions in figure 3–4.

Coronagraph Images Data

Coronagraph images can capture Coronal Mass Ejections as they erupt from the Sun. These images might contain valuable information for ML models: time of CME occurred and its directional propagation. By processing these images, our objective was to keep both the timing and directional information of CMEs, ensuring that the extracted features remain valuable for subsequent analysis and ML models.

To determine when CME happened, we subtracted consecutive coronagraph images in sequence, creating running difference. Pixels with values close to zero in running difference images indicate no change between consecutive images, while non-zero pixel values suggest a sudden ejection of mass into interplanetary space, signaling a CME.

To determine the direction of a CME, we divided the running difference image into four equal-sized quadrants and calculated the percentage of non-zero pixels to measure the amount of changed pixels in each quadrant.

The idea behind this processing step is that if all four quadrants show a significant change in pixels, the CME should be directed towards Earth. This data might provide ML model information, that can be used for timely prediction of the Dst index. The percentage of changed pixels in four equal-sized quadrants of running difference coronagraph images are four attributes of the MESWE Dataset. This pre-processing procedure might be considered as transformation from 2D Image signal into 1D signal. These steps are shown as purple group of actions in figure 3–4.

For each extreme SWE event, we have stored coronagraph 32×32 -pixel grayscale images for efficient processing in modeling phase. These images are saved in both PNG format and NumPy .npy files, containing tensor representing multiple image sequences prepared for direct input into the model alongside in-situ data.

3.3.2 Data Integration for Modeling

Since the dataset consists of time series data, our objective is to predict the target variable, the Dst index, T time steps ahead. This prediction follows a sliding window approach. The input data are structured as tensors with dimensions (B, TS, F) , where B represents the batch size, TS the time steps provided as temporal history, and F the number of features. However, temporal histories in inputs are not shared across different extreme events to prevent cases where a sequence contains partial histories from multiple events. For example, a situation where half of the history originates from Event 5 and the other half from Event 6 would introduce inconsistencies and degrade model performance. In-situ and coronagraph data can be provided as separate tensors to the model.

The coronagraph imagery is represented as five-dimensional tensors with shape (B, S, C, W, H) , where B denotes batch size, S represents the images sequence length, C indicates the number of channels ($C = 1$ because grayscale images), and W and H correspond to the width and height of images, respectively. These image tensors can be incorporated alongside the corresponding in-situ data tensor to the model, matched by same timestamp.

We apply Savitzky–Golay filter (Schafer, 2011) to the Q-Data as a preprocessing step before modeling. This step eliminates noise that downgrades performance of the model by few metric points. Next preprocessing step was discretizing continuous numerical values of Q-Data attributes into ordinal categories through a binning process. Finally, we standardize the data using $x' = \frac{x-\mu}{\sigma}$, where μ and σ represent the mean and standard deviation, respectively. These transformations are not stored in the final MESWE dataset but are rather performed dynamically during data preparation.

3.4 Modeling

3.4.1 Data Split

There is no intuitive approach on how to split data from extreme SWE events in order to be validation set and test set similar to each other. If validation set is easier for the model to converge to some result, it might lead to poor performance on test set and vice-versa. We wanted to deliver strong experimental results, therefore we adopted cross-validation method Bates et al. (2024).

While all extreme SWE events are distinct, some can be vastly different compared to others. To quantify these differences, we opted for Wasserstein distance. Wasserstein distance calculates the minimum amount of 'work' required to transform one probability distribution into the other (Panaretos and Zemel, 2019). These distances from one event to each other is shown in Figure 3–5. We can see that events 19, 21, 30, 31 and partially 25 are very different to other events, particularly to event 14 and 34. We suspect that solar wind speed attribute is responsible for this vast difference as median for each of the events on opposite sides differs by lot (see Figure 3–6). To minimize selection bias and ensure that validation and test sets are as similar as possible, we created five different k-fold sets of train, validation, and test data (i.e. 5-fold cross-validation).

We assume that events caused by CMEs differ from those caused by CIRs. Events caused by CME can be identified by high numerical values in the first four attributes of MESWE dataset, specifically the Q1-Q4 attributes listed in Table 3–2. Based on this argument, we classify all extreme SWE events into three groups:

1. **Quadrants spiking event (S):** All four attributes Q1-Q4 from Table 3–2 have higher values before the Dst decrease. Events: {9, 14, 15, 17, 19, 21, 30, 31, 32, 33}
2. **Quadrants non-spiking event (NS):** Not all four attributes Q1-Q4 from Table 3–2 show higher values before Dst. Events: {4, 5, 6, 7, 8, 10, 22, 25, 34,

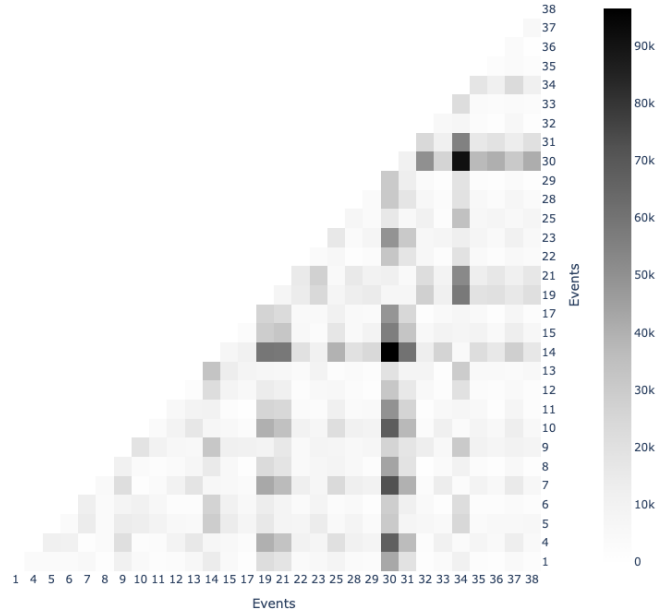


Figure 3 – 5: Wasserstein distances between one event to each other. We can observe large distance of events 19,21,30, and 31 with respect to other extreme SWE events.

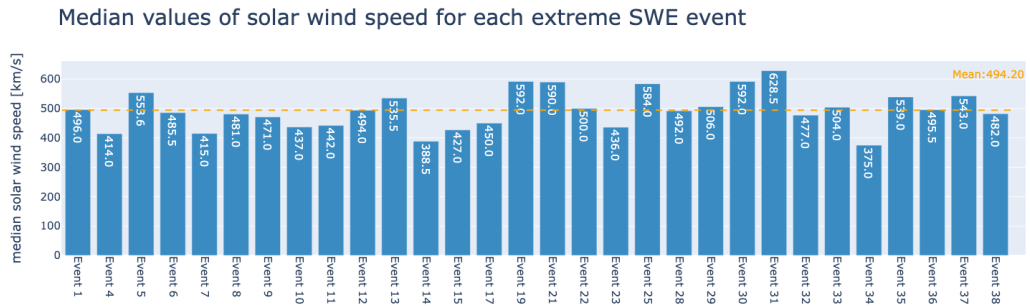
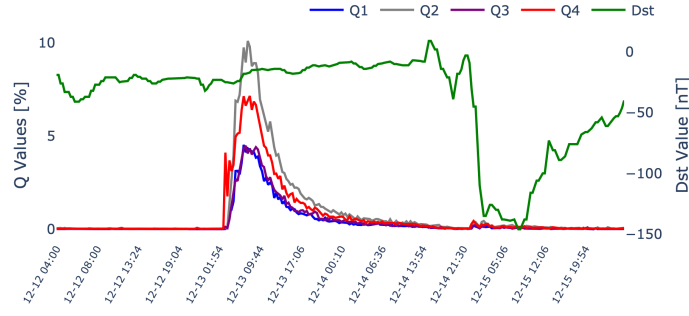


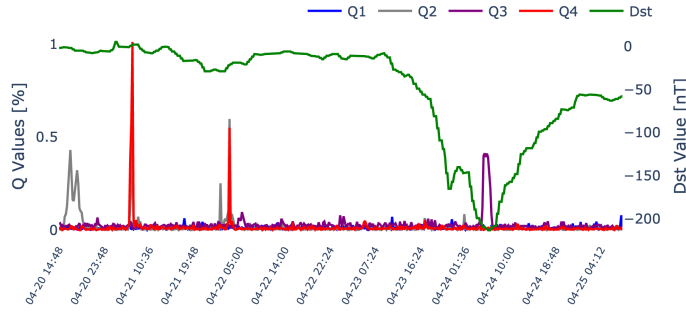
Figure 3 – 6: Median values of solar wind speed during each extreme SWE event.

35, 36, 37, 38}

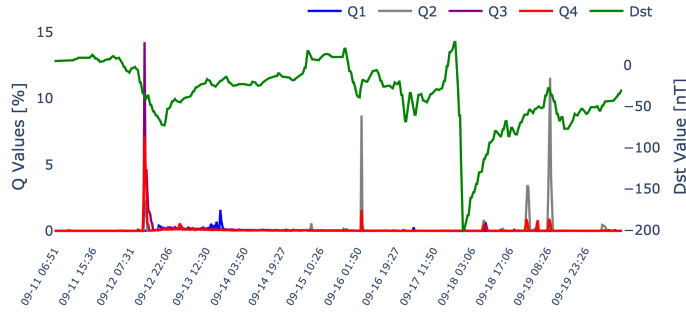
3. **Unclassified (U):** Unclear or uncertain cases. Events: {1, 11, 12, 13, 23, 28, 29}



(a) Spiking



(b) Non-spiking



(c) Unclassified

Figure 3 – 7: Different examples of categorized events. Left y-axis display Coronagraph Visual Features values. Right y-axis display Dst value in nanotesla. We can observe that right before Dst decrease, each example has different type of activity in Coronagraph Visual Features values. **(a)** Event 31 (2006-12-14) is categorized as Spiking. There is visible coronagraph activity before Dst decrease for all 4 Q-attributes with higher percentage value. **(b)** Event 38 (2023-04-24) is categorized as Non-spiking. Although there is visible coronagraph activity before Dst decrease, only Q4 is active and at very low value, only around 1%. **(c)** Event 11 (2000-09-17) is categorized as Unclassified. Although there is visible coronagraph activity before Dst decrease, not all attributes from coronagraphs are active.

Each validation and test set in every k-fold contains one event from each of these three categories, totaling three events in each set, while the remaining events are used for training. To maximize similarity between validation and test sets, we calculated an Event Sum Distance (ESD) for each event, defined as the sum of its Wasserstein distances to other events. The Set Distance (SD) is the sum of ESDs for the three events chosen for each validation or test set. We then searched all possible validation and test set combinations and selected the top five instances where the SD difference between validation and test sets was smallest. Selected k-folds are presented in Table 3–3.

Table 3–3: Selected events for each k-fold validation and test sets, ensuring sets similarity. Remaining events were used as training data in each k-fold, so each k-fold is trained, validated and tested on different combination of extreme SWE events. Each validation or test set in each k-fold contains one representative event from each of the S, NS, and U event categories.

k-fold	Validation Set Events	Test Set Events
1	4, 21, 11	8, 14, 28
2	22, 19, 28	25, 21, 1
3	4, 31, 1	34, 15, 28
4	7, 15, 13	35, 21, 1
5	7, 21, 29	35, 19, 23

3.4.2 Model Implementation

We have implemented three different deep learning models: GRU-IE, GRU-CONV-IEC, GRU-Attn-IEC. The primary differences between these models lie in their input data and their processing, which evolve iteratively across versions. These iterative modifications allow for a comparison of model performance on the Dst index prediction task, enabling an evaluation of whether each change improves or degrades

predictive accuracy. The details of these variations will be discussed in the following sections.

Model GRU-IE

This model takes as input in-situ data, combined with EUV processed images. This model is a recurrent neural network (RNN) with a final linear transformation. The overall architecture is depicted in Figure 3–8.

We conducted systematic search over hyperparameters to identify optimal model configuration. Best performing model has Gated Recurrent Unit (GRU) as the recurrent layer with 64 hidden neurons and 2 stacked layers. A final linear transformation projects the GRU’s hidden representation to a scalar output representing the Dst index prediction. However, as predictions are processed in batches, the output takes the matrix shape [B,P].

Weight optimization was performed using AdamW optimizer with learning rate $1 * 10^{-5}$. To reduce overfitting, weight decay with value 0.1 along with dropout in the GRU layer set to 0.2. Empirical results reveal that applying a moving average smoothing with window of size 5 to 10% of the input batch improves model performance.

Model GRU-CONV-IEC

This model takes two distinct inputs. The first input consists of in-situ and EUV data, identical to the input used in model GRU-IE. The second input consist of sequences of original coronagraph images, which are downsampled to 32×32 pixels in grayscale to reduce computational and memory needs. Further details regarding the tensor shapes of these inputs are outlined in Section 3.3.2 on Data Integration.

Each input is processed through a dedicated branch within the deep learning model architecture. The in-situ data branch follows the same processing steps as GRU-IE. The coronagraph image branch utilizes a Time-Distributed Layer to capture both spatial and temporal dependencies within the image sequences. This layer

applies convolutional operations to each image independently while preserving temporal structure, generating feature maps for each timestep. The outputs of the convolutional layers, feature maps, are batch-normalized and passed through a ReLU activation function. Subsequently, the spatial dimensions (width and height) of the feature maps are flattened, ensuring a compatible tensor shape for input into a recurrent GRU layer.

The outputs from both processing branches are fused at a later stage by concatenating tensors along the feature dimension. The resulting representation is then passed through a linear transformation to produce a scalar output for each batch. The complete model architecture is illustrated in Figure 3–8.

The best-performing model configuration was obtained with the following hyperparameters: convolutional layers with 2 and 8 filters, GRU layers with 152 hidden neurons and 2 stacked layers in both branches. The model was optimized using the AdamW optimizer with a learning rate of $1 * 10^{-4}$.

Model GRU-Attn-IEC

This model follows a similar structure to GRU-IE and GRU-CONV-IEC, taking two distinct inputs. The first input remains unchanged from GRU-IE and GRU-CONV-IEC, consisting of in-situ and EUV data. The second input also consists of coronagraph data; however, instead of raw image sequences, it is represented as four processed 1D signals, as discussed in Section 3.3.1.

We hypothesize that these processed signals retain the essential information present in the original coronagraph image sequences while being more computationally efficient and easier to manipulate. The first processing branch, remains identical to its counterpart in previous models. However, the second branch, responsible for processing the coronagraph-derived signals, requires a specialized attention. By that we mean the Self-Attention mechanism, as introduced by Vaswani (2017), which has become a SOTA technique in deep learning community. This mechanism allows the model to effectively capture long-range dependencies within the input sequences,

enhancing the representation of temporal patterns in the coronagraph data.

The processed signals are first passed through a linear transformation followed by a non-linear activation function, GELU (Hendrycks and Gimpel, 2016). The transformed representations are then fed into a GRU layer, where the final hidden state is used as the query input for a multi-headed self-attention mechanism. The key and value inputs to the self-attention layer consist of all hidden states produced by the GRU.

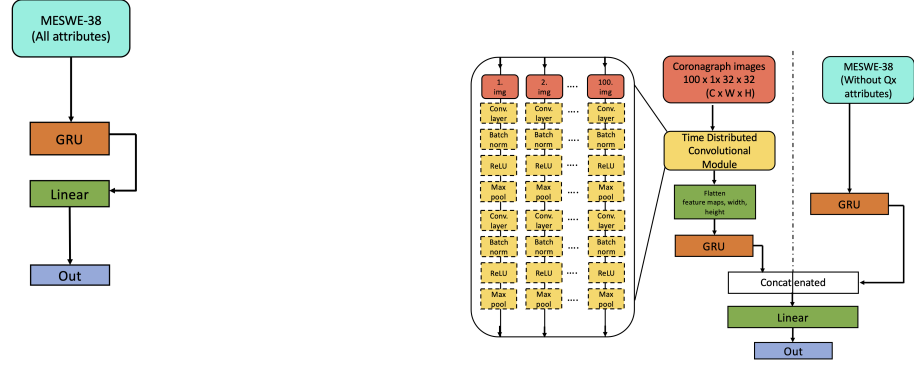
We agree that this design differs from the conventional Transformer architecture, as it retains RNN components while incorporating self-attention. However, empirical results indicate that this hybrid approach yields the best performance. While a variant eliminating the GRU altogether may be an alternative, we have not yet identified a configuration that outperforms our current model. Recent developments in state-space models, such as RWKV (Peng et al., 2023) and MAMBA (Gu and Dao, 2023), offer promising approaches for reducing complexity while maintaining expressive temporal dependencies. While these methods have been predominantly applied to language modeling, their adaptation to time series tasks appears straightforward. To our knowledge, their application in space weather studies remains unexplored.

Finally, the outputs from both processing branches are concatenated and passed through a linear transformation to obtain a scalar prediction for each batch, representing the Dst index.

The best-performing model was configured with GRU layers containing 256 hidden neurons and 2 stacked layers in both branches. The model was optimized using the AdamW optimizer with a learning rate of $3 * 10^{-5}$, with minimal to no regularization applied.

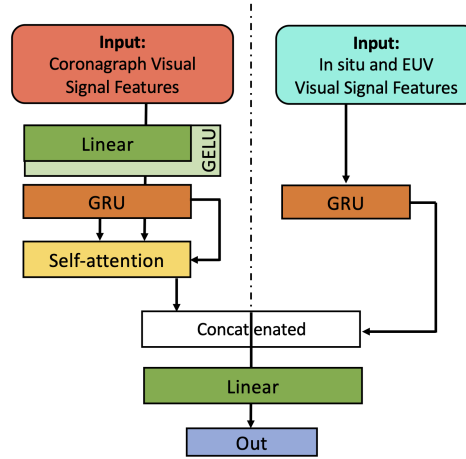
3.4.3 Model Evaluation Methods

We evaluated the performance of models using 5-fold cross-validation, as described in Section 3.4.1. For each fold, the models were trained and tested across five distinct random seeds. Seeds were chosen once at random and were consistently applied



(a) Model GRU-IE.

(b) Model GRU-CONV-IEC



(c) Model GRU-Attn-IEC

Figure 3 – 8: Different models used to forecast Dst index.

across all models to ensure fair comparison. The performance metrics are reported as mean and standard deviation across these seeds. Root Mean Square Error (RMSE) was employed as the primary evaluation metric, since it is commonly used in time series analysis and its adoption within studies on Dst index prediction. RMSE is given by:

$$RMSE = \sqrt{\frac{1}{n} \sum_{i=1}^n (\hat{y}_i - y_i)^2}. \quad (3.1)$$

Recently, Laperre et al. (2020) criticized the use of classical metrics such as RMSE in the evaluation of Dst index predictions, arguing that these metrics fail to capture behavior, when magnitudes of a decrease in Dst are predicted correctly but with a temporal shift, predicting the decrease after it has already occurred in the observed data. This phenomenon, referred to as persistence behavior, can be visually identified when one looks closely on predicted and actual time series data. To address this issue, authors proposed the use of Dynamic Time Wrapping (DTW) measure.

DTW similarity measure between for time series data (Berndt and Clifford, 1994; Senin, 2008). It seeks optimal alignment between two sequences with variations in the time axis. Given two time series $\hat{Y} = (\hat{y}_1, \hat{y}_2, \dots, \hat{y}_i)$ and $Y = (y_1, y_2, \dots, y_j)$ of length of i and j , cost matrix M of size $i \times j$ is constructed and filled as follows:

$$M(i, j) = d(\hat{y}_i, y_j) + \min\{M(i-1, j-1), M(i-1, j), M(i, j-1)\}, \quad (3.2)$$

where $d()$ is distance measure, in our case Euclidean distance.

Optimal alignment is then determined using dynamic programming to find the minimum-cost warping path in cost matrix M by backtracking. This process starts at bottom-right corner of M , $M(i, j)$ and follows the smallest neighboring value at each step until reaches the top-left corner, $M(1, 1)$. Steps taken in backtracking forms the optimal warping path which aligns \hat{Y} and Y with the minimal accumulated cost.

DTW can also align very distant parts of sequences. Thus warping window w is constraint applied in DTW, ensuring that the warping path satisfies $|i - j| \leq w$, restricting to seeking alignment only within w steps.

While our focus is on Dst index prediction, DTW has also been used to analyze solar wind time series data in (Samara et al., 2022), highlighting its applicability in

space weather research community. In this study, we use same implementation of DTW as provided in (Laperre et al., 2020).

From DTW cost matrix M , one can derive valuable insights, such as: I) steps of warping path off diagonal line and II) sum of values on diagonal line. In figure 3–9 are shown 3 different pairs of sequences Y and \hat{Y} and corresponding DTW cost matrices. Figure 3–9(a) shows ideal alignment, figure 3–9(b) shows temporal alignment, but with few magnitude differences, figure 3–9(c) shows temporal misalignment with magnitude changes kept same as in 3–9(b). The situation shown in figure 3–9(c) is the most common when predicting the Dst index T steps ahead. This is referred to as persistence behavior of predictions \hat{Y} .

Notice how warping path changes and how values on diagonal line changes in figures 3–9(d), 3–9(e) and 3–9(f). In an ideal alignment, the path follows the diagonal, and all diagonal values are zero. When time series exhibit differences in magnitude, the warping path deviates from the diagonal for a few steps, and the values along the diagonal are no longer zero. For sequences with temporal misalignment, only the first and last steps of the path remain on the diagonal line, and the values along the diagonal increase significantly.

In addition to RMSE, we compare models using the percentage of steps on the diagonal in the cost matrix D_{on} (higher is better) and the sum of values along the diagonal line D_{sum} (lower is better).

Community Benchmark

We also compared our approach with common methodologies used in the Dst forecast community. These approaches construct long time series datasets containing solar wind and interplanetary magnetic field measurements spanning across many years. This long sequence is then partitioned into training, validation and test set. The methodology is reported in e.g. (Efitorov et al., 2018; Lethy et al., 2018; Purnomo et al., 2021; Zhang et al., 2023). We evaluated performance of this methodology on the solar wind and IMF data from MESWE test set, which comprises data from

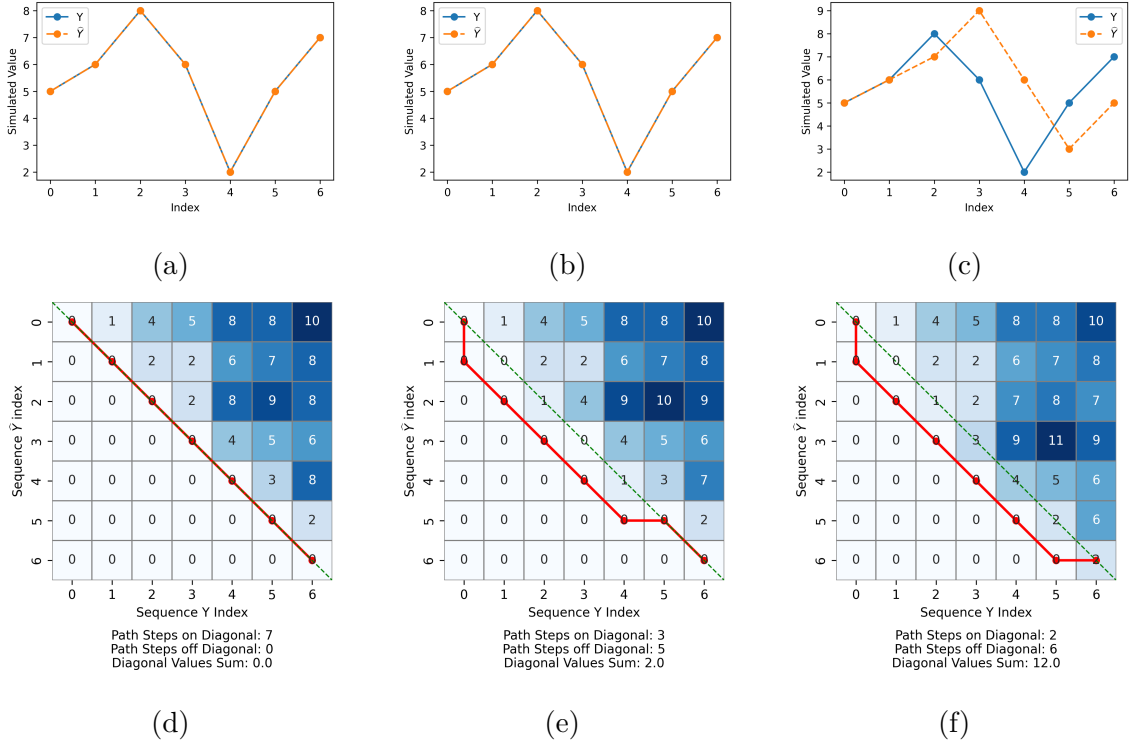


Figure 3–9: Changes in the warping path and diagonal values for different time series alignments. (a–c) show different simulated time series, while (d–f) depict their corresponding DTW cost matrices with warping paths. In an ideal alignment, the warping path follows the diagonal, and diagonal values remain zero. For time series with differences in magnitude, the warping path deviates slightly from the diagonal, and diagonal values are no longer zero. When sequences are misaligned in the time axis, only the first and last steps of the warping path remain on the diagonal, with a significant increase in diagonal values.

extreme solar activity periods. For this comparison, we obtained measurements of V_p , N_p , B_z and Dst from 1998 to 2016 in 1h cadence. Since MESWE dataset follows the cadence of the SOHO/LASCO coronagraph (20-30 min.), we interpolated the data to 30 min. cadence as well. The final 20% of time steps, period from April 2013 to December 2015, were classified as test set. This test time span includes same extreme event from MESWE dataset used in test set, specifically under number 35

from the 3rd fold in Table 3–3. We have trained GRU recurrent neural network and evaluated its performance on Event 35. The results were compared against those obtained using community benchmark dataset as described above, as well as the $GRU - IE$ and $GRU - Attn - IEC$ approaches on the MESWE dataset for the same event. This way, we can compare visually same event using not only different deep learning approaches, but also different methodologies.

3.4.4 Model Performance

$GRU - CONV - IEC$ reported low RMSE values for both prediction tasks. However, the D_{on} and D_{sum} measures were significantly lower than those of other approaches. This might indicate that the shift between predicted and target Dst values is more significant across all test sets in different k-folds.

Table 3–4: $RMSE$ and D_{on} with D_{sum} of cost DTW matrix for $GRU - CONV - IEC$ approaches and for prediction 10 steps and 20 steps ahead of Dst prediction on test set.

		GRU-CONV-IEC		
	k-fold	RMSE [nT]	D_{on} [%]	D_{sum}
\hat{y}_{t+10}	1	14.0	3.8	4.29×10^7
	2	12.9	3.3	6.33×10^7
	3	21.6	5.7	6.06×10^7
	4	12.4	5.5	4.63×10^7
	5	18.3	4.7	3.15×10^7
	Mean	15.8	4.7	4.89×10^7
\hat{y}_{t+20}	1	22.0	2.3	3.69×10^7
	2	16.6	3.0	4.43×10^7
	3	28.2	2.1	6.10×10^7
	4	15.1	2.9	3.65×10^7
	5	24.5	2.7	2.61×10^7
	Mean	21.3	2.6	4.10×10^7

Performance of models $GRU - IE$ and $GRU - Attn - IEC$ on different k-folds as

shown in Table 3–3 from MESWE dataset are shown in Table 3–5 for predictions 10 and 20 time steps ahead. The mean RMSE varies across k-folds for both forecasting horizons. For the 10-step prediction, $GRU - IE$ achieves a lower test RMSE of 17.0 compared to 17.4 for $GRU - Attn - IEC$. However, for the 20-step prediction, $GRU - Attn - IEC$ slightly outperforms $GRU - IE$, with RMSE values of 24.1 and 24.3, respectively.

Table 3–5: $RMSE$ and D_{on} with D_{sum} of cost DTW matrix for $GRU - IE$ and $GRU - Attn - IEC$ approaches and for prediction 10 steps and 20 steps ahead of Dst prediction on test set.

		GRU-IE			GRU-Attn-IEC		
	k-fold	RMSE [nT]	D_{on} [%]	D_{sum}	RMSE [nT]	D_{on} [%]	D_{sum}
\hat{y}_{t+10}	1	20.4 ± 0.3	6.8 ± 0.1	2.15×10^7	21.0 ± 0.4	6.79 ± 0.6	2.22×10^7
	2	17.5 ± 0.2	5.2 ± 0.4	2.39×10^7	18.4 ± 0.7	4.44 ± 0.5	2.89×10^7
	3	12.5 ± 0.1	6.2 ± 0.5	2.52×10^7	13.0 ± 0.2	6.51 ± 0.8	2.75×10^7
	4	14.2 ± 0.3	4.9 ± 0.5	2.87×10^7	14.6 ± 0.2	4.98 ± 0.4	3.56×10^7
	5	20.5 ± 0.2	3.9 ± 0.3	4.58×10^7	19.9 ± 0.5	3.73 ± 0.3	4.65×10^7
	Mean	17.0	<i>5.3</i>	2.9×10^7	17.4	<i>5.3</i>	3.21×10^7
\hat{y}_{t+20}	1	28.1 ± 0.4	4.93 ± 1.0	3.46×10^7	29.1 ± 0.6	5.17 ± 0.1	3.15×10^7
	2	24.6 ± 0.6	2.38 ± 0.6	3.35×10^7	24.9 ± 0.5	2.43 ± 0.5	3.91×10^7
	3	17.3 ± 0.2	3.75 ± 0.7	3.54×10^7	17.3 ± 0.5	4.41 ± 0.3	3.43×10^7
	4	21.6 ± 0.3	2.48 ± 0.1	4.11×10^7	21.5 ± 0.4	2.47 ± 0.1	4.75×10^7
	5	29.7 ± 0.1	2.11 ± 0.2	6.66×10^7	27.7 ± 0.9	2.14 ± 0.3	6.45×10^7
	Mean	24.3	3.1	4.22×10^7	24.1	3.33	4.34×10^7

In contrast, the mean D_{sum} across k-folds is lower for $GRU - IE$ approach for both predictions horizons, meaning that it aligns predictions slightly better with observed Dst values. However, the performance of the models according to D_{on} varies with the prediction horizon length. For 10-step-ahead predictions, both models achieves same mean of D_{on} across k-folds. However, when extending the prediction horizon to 20 steps, $GRU - Attn - IEC$ model achieves a superior mean D_{on} across k-folds, exceeding $GRU - IE$ by 0.13%.

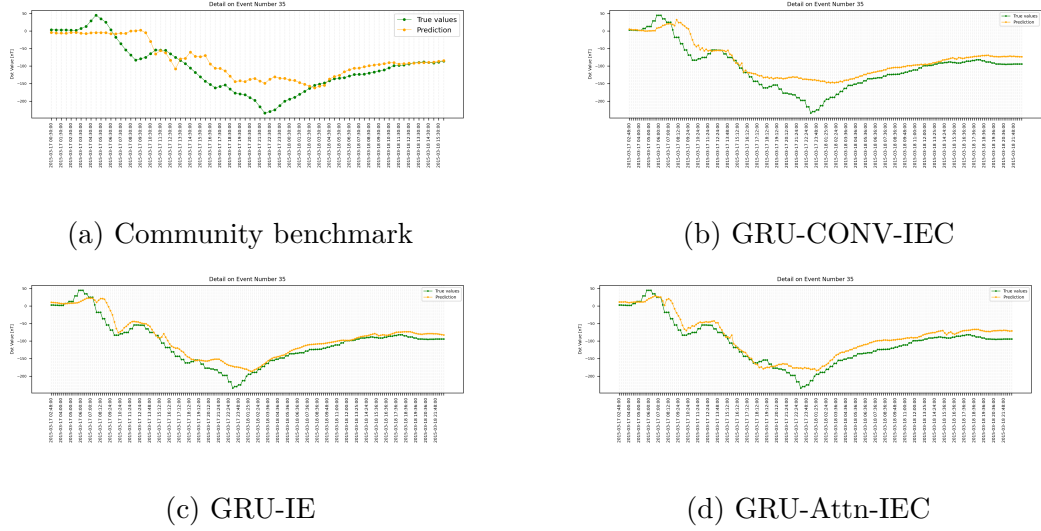


Figure 3–10: Detailed test set predictions 10 steps ahead of event that occurred on March 2015 using different approaches.

From this comparative analysis, it can be concluded that *GRU – IE* achieves better mean values for two out of three comparative scores for 10 time steps ahead forecast, but as we extend prediction to 20 steps, it turns around into *GRU – Attn – IEC* favor. The difference in performance between the two models remains relatively small, raising the question of whether the added complexity of *GRU – Attn – IEC* justifies the marginal improvement in predictive accuracy.

Across different k-folds, both models demonstrate the poorest performance on the fifth fold across all comparable measures. This suggests that this fold presents greater challenges for the models in learning underlying patterns and generalizing it on the test set. This might lead to important methodological question: “How do we know if a upcoming geomagnetic storm is not from fifth fold?”.

Figure 3–10 presents predictions 10 steps ahead for Event 35 (March 2015) created by different approaches. 3–10(a) shows predictions based on the community benchmark dataset as described in Section 3.4.3, while 3–10(b) and 3–10(c) shows predictions on MESWE dataset with *GRU – IE* and *GRU – Attn – IEC* approaches, respectively, as detailed in Section 3.4.3. Predictions in 3–10(b) and 3–10(c) align

more closely with the target Dst values and outline a reduced persistence effect. In particular, in 3–10(c) during Dst decline between 9:00 and 10:00, predictions are closer to target values compared to 3–10(b) at the same time, where Dst prediction is slightly overestimated. However, neither approach 3–10(b) nor 3–10(c) accurately predicted magnitude of the decrease, underestimating it by ≈ 60 nT.

3.5 Project Evaluation

The project has successfully achieved all three originally defined objectives as stated in 3.1.4.

First, we constructed the MESWE dataset by integrating in situ measurements of solar wind parameters, preprocessed EUV and coronagraph imagery from remote sensing instruments, and the geomagnetic activity index (Dst). By transforming these raw data into a unified structure, the dataset provides usable inputs for heterogeneous machine learning models, particularly for applications involving time series regression. MESWE dataset and codes are available online at github ⁶. They are also available in other locations such as: ⁷⁸ and also in workspace at KKUI datalab ⁹.

Second, our comparative analysis demonstrated that the deep learning architecture with preprocessed remote sens outperformed established models in the space weather forecasting domain. Quantitative evaluation using standard metrics confirmed superior predictive accuracy, particularly during extreme events. Qualitative assessment of forecast outputs further validated the model’s enhanced capability to capture the timing and magnitude of geomagnetic disturbances.

Third, we successfully reported our findings through a peer-reviewed publication Majirský et al. (2025). These contributions advance the state of machine learning

⁶<https://github.com/kkuichi/am367df>

⁷<https://github.com/space-lab-sk/dst-images-insitu>

⁸<https://github.com/space-lab-sk/vigil-like-data>

⁹<https://datalab.kkui.fei.tuke.sk/gpu/>

applications in space weather forecasting and fulfill our scientific contribution objective as it was primary criterion for the project's success.

Based on the successful achievement of project objectives, we can see few avenues for future work. The SOHO/LASCO coronagraph images were employed as the primary temporal reference, with a typical cadence of approximately 20 to 30 minutes. In contrast, the in situ solar wind data were available at a significantly higher temporal resolution of 30-60 second intervals. In the current approach, we utilized only the mean values of solar wind parameters within each 20-30 minute interval corresponding to SOHO/LASCO images. However, this may not fully capture the temporal variability and dynamic features of the solar wind. As a follow-up improvement, we propose incorporating additional statistical aggregations, including maximum, minimum, standard deviation and rate of change values of in situ measurements within each 20-30 minute interval. We are uncertain if this will provide model more noise or valuable signal.

Another potential follow-up for further research work involves incorporating magnetograms into the input data for the deep learning models. Rather than using whole magnetogram images, an alternative approach could involve leveraging extracted physical attributes related to solar flares, as suggested by Nishizuka et al. (2018). An open question remains: would these extracted features enhance model performance, or would they primarily introduce noise to the deep learning model?

4 Conclusion

In this study, we explore how the upcoming space mission Vigil can leverage Machine Learning to enhance its potential for providing early warnings of potentially hazardous solar activity. A key advantage of Vigil is its planned position at the L5 Lagrange point, which allows continuous monitoring of the region between the Sun and Earth. This unique position will provide unprecedented observational data, presenting both an opportunity and a challenge for Machine Learning approaches, which traditionally rely on extensive historical datasets.

We have identified data sources that closely resemble the instruments onboard Vigil and have selected periods of extreme space weather events for analysis. Based on these periods, we have constructed the MESWE dataset, which integrates in situ measurements of solar wind parameters, Extreme Ultraviolet and Coronagraph images, and geomagnetic activity represented by the Dst index. This preliminary step is essential for assessing the feasibility of Machine Learning applications for Vigil. To the best of our knowledge, this is the first study to undertake such an approach.

To enhance Vigil’s capability in recognizing threats, it would be beneficial for the mission to predict Earth’s geomagnetic state, represented by the Dst index. We have developed deep learning models designed for this time series forecasting task. We explored multiple models that vary in input data and processing methods. Our findings indicate that the model incorporating both in situ measurements and processed remote sensing data achieved the highest performance.

Our findings have been published in a peer-reviewed journal (Majirský et al., 2025), where we discuss the potential data from Vigil and present methods for obtaining it. Additionally, we have prepared a manuscript that provides a more in-depth analysis of the Machine Learning approaches explored in our study, which we plan to submit to the *Astrophysical Journal*.

This work was conducted in collaboration with RNDr. Šimon Mackovjak, PhD.,

affiliated with the Institute of Experimental Physics, Slovak Academy of Sciences. Outcomes were part of deliverables for the RPA SKR1-23 project "Study toward enhancing reliability and timeliness of Vigil mission predictions through Machine Learning", funded by the European Space Agency.

References

- Abadi, M., Barham, P., Chen, J., Chen, Z., Davis, A., Dean, J., Devin, M., Ghemawat, S., Irving, G., Isard, M. et al. (2016). {TensorFlow}: a system for {Large-Scale} machine learning, *12th USENIX symposium on operating systems design and implementation (OSDI 16)*, pp. 265–283.
- Abboush, M., Knieke, C. and Rausch, A. (2023). Gru-based denoising autoencoder for detection and clustering of unknown single and concurrent faults during system integration testing of automotive software systems, *Sensors* **23**(14): 6606.
- Adlam, B. and Pennington, J. (2020). Understanding double descent requires a fine-grained bias-variance decomposition, *Advances in neural information processing systems* **33**: 11022–11032.
- Aggarwal, C. C. (2023). Neural networks and deep learning: A textbook, 2nd editon, Cham, Switzerland: Springer International Publishing .
- Allen, J., Sauer, H., Frank, L. and Reiff, P. (1989). Effects of the march 1989 solar activity, *Eos, Transactions American Geophysical Union* **70**(46): 1479–1488.
- Bala, R. and Reiff, P. (2012). Improvements in short-term forecasting of geomagnetic activity, *Space Weather* **10**(6): 0–1.
- Bates, S., Hastie, T. and Tibshirani, R. (2024). Cross-validation: what does it estimate and how well does it do it?, *Journal of the American Statistical Association* **119**(546): 1434–1445.
- Belkin, M., Hsu, D., Ma, S. and Mandal, S. (2019). Reconciling modern machine-learning practice and the classical bias-variance trade-off, *Proceedings of the National Academy of Sciences* **116**(32): 15849–15854.
- Bengio, Y., Simard, P. and Frasconi, P. (1994). Learning long-term dependencies

- with gradient descent is difficult, *IEEE transactions on neural networks* **5**(2): 157–166.
- Bergstra, J. and Bengio, Y. (2012). Random search for hyper-parameter optimization., *Journal of machine learning research* **13**(2).
- Berndt, D. J. and Clifford, J. (1994). Using dynamic time warping to find patterns in time series, *Proceedings of the 3rd international conference on knowledge discovery and data mining*, pp. 359–370.
- Borovsky, J. E. and Shprits, Y. Y. (2017). Is the dst index sufficient to define all geospace storms?, *Journal of Geophysical Research: Space Physics* **122**(11): 11–543.
- Camporeale, E. (2019). The challenge of machine learning in space weather: Nowcasting and forecasting, *Space weather* **17**(8): 1166–1207.
- Cho, K. (2014). On the properties of neural machine translation: Encoder-decoder approaches, *arXiv preprint arXiv:1409.1259* **0**(0): 0–1.
- Chung, J., Gulcehre, C., Cho, K. and Bengio, Y. (2014). Empirical evaluation of gated recurrent neural networks on sequence modeling, *arXiv preprint arXiv:1412.3555* **0**(0): 0–1.
- Cid, C., Palacios, J., Saiz, E., Guerrero, A. and Cerrato, Y. (2014). On extreme geomagnetic storms, *Journal of Space Weather and Space Climate* **4**: A28.
- Cybenko, G. (1989). Approximation by superpositions of a sigmoidal function, *Mathematics of control, signals and systems* **2**(4): 303–314.
- Dang, T., Li, X., Luo, B., Li, R., Zhang, B., Pham, K., Ren, D., Chen, X., Lei, J. and Wang, Y. (2022). Unveiling the space weather during the starlink satellites destruction event on 4 february 2022, *Space weather* **20**(8): e2022SW003152.

- Domingo, V., Fleck, B. and Poland, A. I. (1995). The soho mission: an overview, *Solar Physics* **162**: 1–37.
- Eastwood, J. P., Brown, P., Magnes, W., Carr, C. M., Agu, M., Baughen, R., Berghofer, G., Hodgkins, J., Jernej, I., Möstl, C., Oddy, T., Strickland, A. and Vitkova, A. (2024). The Vigil Magnetometer for Operational Space Weather Services From the Sun-Earth L5 Point, *Space Weather* **22**(6): e2024SW003867.
- Efitorov, A., Myagkova, I., Shirokii, V. and Dolenko, S. (2018). The prediction of the dst-index based on machine learning methods, *Cosmic Research* **56**: 434–441.
- Elsken, T., Metzen, J. H. and Hutter, F. (2019). Neural architecture search: A survey, *Journal of Machine Learning Research* **20**(55): 1–21.
- Fozzard, R., Bradshaw, G. and Ceci, L. (1988). A connectionist expert system that actually works, *Advances in neural information processing systems* **1**.
- Giegengack, R. (2015). The carrington coronal mass ejection of 1859, *Proceedings of the American Philosophical Society* **159**(4): 421–433.
- Glorot, X. and Bengio, Y. (2010). Understanding the difficulty of training deep feed-forward neural networks, *Proceedings of the thirteenth international conference on artificial intelligence and statistics*, JMLR Workshop and Conference Proceedings, pp. 249–256.
- Glorot, X., Bordes, A. and Bengio, Y. (2011). Deep sparse rectifier neural networks, *Proceedings of the fourteenth international conference on artificial intelligence and statistics*, JMLR Workshop and Conference Proceedings, pp. 315–323.
- Goodman, S. J., Schmit, T. J., Daniels, J. and Redmon, R. J. (2019). *The GOES-R series: a new generation of geostationary environmental satellites*, Elsevier.
- Gu, A. and Dao, T. (2023). Mamba: Linear-time sequence modeling with selective state spaces, *arXiv preprint arXiv:2312.00752* .

-
- Hapgood, M., Angling, M. J., Attrill, G., Bisi, M., Cannon, P. S., Dyer, C., Eastwood, J. P., Elvidge, S., Gibbs, M., Harrison, R. A. et al. (2021). Development of space weather reasonable worst-case scenarios for the uk national risk assessment.
- Hendrycks, D. and Gimpel, K. (2016). Gaussian error linear units (gelus), *arXiv preprint arXiv:1606.08415*.
- Hochreiter, S. (1997). Long short-term memory, *Neural Computation MIT-Press* **0**(0): 0–1.
- Hornik, K. (1991). Approximation capabilities of multilayer feedforward networks, *Neural networks* **4**(2): 251–257.
- Hu, A., Camporeale, E. and Swiger, B. (2023). Multi-hour-ahead dst index prediction using multi-fidelity boosted neural networks, *Space Weather* **21**(4): e2022SW003286.
- Hu, A., Shneider, C., Tiwari, A. and Camporeale, E. (2022). Probabilistic prediction of dst storms one-day-ahead using full-disk soho images, *Space weather* **20**(8): e2022SW003064.
- Ioffe, S. (2015). Batch normalization: Accelerating deep network training by reducing internal covariate shift, *arXiv preprint arXiv:1502.03167* **0**(0): 0–1.
- Jankovičová, D., Dolinský, P., Valach, F. and Vörös, Z. (2002). Neural network-based nonlinear prediction of magnetic storms, *Journal of Atmospheric and Solar-Terrestrial Physics* **64**(5-6): 651–656.
- KA, N. D., Jusoh, M. H., Mashohor, S., Sali, A., Yoshikawa, A., Kasuan, N., Hashim, M. H. and Hairuddin, M. A. (2023). Bibliographic dataset of literature for analysing global trends and progress of the machine learning paradigm in space weather research, *Data in Brief* **51**: 109667.

-
- Kaiser, M. L., Kucera, T., Davila, J., St Cyr, O., Guhathakurta, M. and Christian, E. (2008). The stereo mission: An introduction, *Space Science Reviews* **136**: 5–16.
- Kingma, D. P. (2014). Adam: A method for stochastic optimization, *arXiv preprint arXiv:1412.6980* .
- Krizhevsky, A., Sutskever, I. and Hinton, G. E. (2012). Imagenet classification with deep convolutional neural networks, *Advances in neural information processing systems* **25**.
- Kugblenu, S., Taguchi, S. and Okuzawa, T. (1999). Prediction of the geomagnetic storm associated dst index using an artificial neural network algorithm, *Earth, planets and space* **51**(4): 307–313.
- Lafon, M. and Thomas, A. (2024). Understanding the double descent phenomenon in deep learning, *arXiv preprint arXiv:2403.10459* .
- Laperre, B., Amaya, J. and Lapenta, G. (2020). Dynamic time warping as a new evaluation for dst forecast with machine learning, *Frontiers in Astronomy and Space Sciences* **7**: 540918.
- Lazzús, J., Vega, P., Rojas, P. and Salfate, I. (2017). Forecasting the dst index using a swarm-optimized neural network, *Space Weather* **15**(8): 1068–1089.
- Lecun, Y. (1985). Une procedure d’apprentissage pour reseau a seuil asymmetrique (a learning scheme for asymmetric threshold networks), *Proceedings of Cognitive 85, Paris, France*, pp. 599–604.
- LeCun, Y., Bengio, Y. and Hinton, G. (2015). Deep learning, *nature* **521**(7553): 436–444.
- LeCun, Y., Boser, B., Denker, J., Henderson, D., Howard, R., Hubbard, W. and Jackel, L. (1989). Handwritten digit recognition with a back-propagation network, *Advances in neural information processing systems* **2**.
-

- LeCun, Y., Bottou, L., Bengio, Y. and Haffner, P. (1998). Gradient-based learning applied to document recognition, *Proceedings of the IEEE* **86**(11): 2278–2324.
- Lederer, J. (2021). Activation functions in artificial neural networks: A systematic overview, *arXiv preprint arXiv:2101.09957*.
- Lethy, A., El-Eraki, M. A., Samy, A. and Deebes, H. A. (2018). Prediction of the dst index and analysis of its dependence on solar wind parameters using neural network, *Space Weather* **16**(9): 1277–1290.
- Lilensten, J., Dumbović, M., Spogli, L., Belehaki, A., Van der Linden, R., Poedts, S., Barata, T., Bisi, M. M., Cessateur, G., De Donder, E. et al. (2021). Quo vadis, european space weather community?, *Journal of Space Weather and Space Climate* **11**: 26.
- Lo, M. W., Llanos, P. J. and Hintz, G. R. (2010). An l5 mission to observe the sun and space weather, part i, *AAS/AIAA Astrodynamacist Specialist Conference, San Diego*, Vol. 136, pp. 10–121.
- Loewe, C. and Prölss, G. (1997). Classification and mean behavior of magnetic storms, *Journal of Geophysical Research: Space Physics* **102**(A7): 14209–14213.
- Lu, L., Shin, Y., Su, Y. and Karniadakis, G. E. (2019). Dying relu and initialization: Theory and numerical examples, *arXiv preprint arXiv:1903.06733*.
- Lundstedt, H. (1991). Neural network predictions of gemagnetic activity, *IAGA Programs and Abstracts, XX General Assembly, IUGG, Vienna 1991* **0**(0): 0–1.
- Maas, A. L., Hannun, A. Y., Ng, A. Y. et al. (2013). Rectifier nonlinearities improve neural network acoustic models, *Proc. icml*, Vol. 30, Atlanta, GA, p. 3.
- Mackovjak, Š., Harman, M., Maslej-Krešňáková, V. and Butka, P. (2021). Scss-net: solar corona structures segmentation by deep learning, *Monthly Notices of the Royal Astronomical Society* **508**(3): 3111–3124.

- Majirský, A., Mackovjak, Š., Kostárová, S. and Amrich, S. (2025). Extreme space weather events of the past 30 years: Preparation for data from mission vigil, *Earth and Space Science* **12**(2): e2024EA003937.
- Minsky, M. and Papert, S. (1969). An introduction to computational geometry, *Cambridge tiass.*, *HIT* **479**(480): 104.
- Motúzová, V. (2023). Predikcia geomagnetických búrok pomocou hlbokého učenia.
- Nakkiran, P., Kaplun, G., Bansal, Y., Yang, T., Barak, B. and Sutskever, I. (2021). Deep double descent: Where bigger models and more data hurt, *Journal of Statistical Mechanics: Theory and Experiment* **2021**(12): 124003.
- Ng, A. Y. (2004). Feature selection, l_1 vs. l_2 regularization, and rotational invariance, *Proceedings of the twenty-first international conference on Machine learning*, p. 78.
- Nishizuka, N., Sugiura, K., Kubo, Y., Den, M. and Ishii, M. (2018). Deep flare net (defn) model for solar flare prediction, *The Astrophysical Journal* **858**(2): 113.
- Panaretos, V. M. and Zemel, Y. (2019). Statistical aspects of wasserstein distances, *Annual review of statistics and its application* **6**(1): 405–431.
- Paralič, J. (2003). Objavovanie znalostí v databázach.
- Park, W., Lee, J., Kim, K.-C., Lee, J., Park, K., Miyashita, Y., Sohn, J., Park, J., Kwak, Y.-S., Hwang, J. et al. (2021). Operational dst index prediction model based on combination of artificial neural network and empirical model, *Journal of Space Weather and Space Climate* **11**: 38.
- Parker, D. (1985). Learning-logic: Casting the cortex of the human brain in silicon.
- Paszke, A., Gross, S., Chintala, S., Chanan, G., Yang, E., DeVito, Z., Lin, Z., Desmaison, A., Antiga, L. and Lerer, A. (2017). Automatic differentiation in pytorch.

-
- Peng, B., Alcaide, E., Anthony, Q., Albalak, A., Arcadinho, S., Biderman, S., Cao, H., Cheng, X., Chung, M., Grella, M. et al. (2023). Rwkv: Reinventing rnns for the transformer era, *arXiv preprint arXiv:2305.13048* .
- Penza, V., Bertello, L., Cantoresi, M., Criscuoli, S. and Berrilli, F. (2023). Prediction of solar cycle 25: applications and comparison, *Rendiconti Lincei. Scienze Fisiche e Naturali* **34**(3): 663–670.
- Pesnell, W. D., Thompson, B. J. and Chamberlin, P. (2012). *The solar dynamics observatory (SDO)*, Springer.
- Prince, S. J. (2023). *Understanding deep learning*, MIT press.
- Purnomo, H. D., Trihandaru, S. et al. (2021). Disturbance storm time index prediction using long short-term memory machine learning, *2021 4th International Conference of Computer and Informatics Engineering (IC2IE)*, IEEE, pp. 311–316.
- Rosenblatt, F. (1958). The perceptron: a probabilistic model for information storage and organization in the brain., *Psychological review* **65**(6): 386.
- Rotter, T., Veronig, A., Temmer, M. and Vršnak, B. (2012). Relation between coronal hole areas on the sun and the solar wind parameters at 1 au, *Solar physics* **281**: 793–813.
- Rumelhart, D. E., Hinton, G. E. and Williams, R. J. (1986). Learning representations by back-propagating errors, *nature* **323**(6088): 533–536.
- Samara, E., Laperre, B., Kieokaew, R., Temmer, M., Verbeke, C., Rodriguez, L., Magdalenic, J. and Poedts, S. (2022). Dynamic time warping as a means of assessing solar wind time series, *The Astrophysical Journal* **927**(2): 187.
- Schafer, R. W. (2011). What is a savitzky-golay filter?[lecture notes], *IEEE Signal processing magazine* **28**(4): 111–117.
-

- Senin, P. (2008). Dynamic time warping algorithm review, *Information and Computer Science Department University of Hawaii at Manoa Honolulu, USA* **855**(1-23): 40.
- Sermanet, P., Chintala, S. and LeCun, Y. (2012). Convolutional neural networks applied to house numbers digit classification, *Proceedings of the 21st international conference on pattern recognition (ICPR2012)*, IEEE, pp. 3288–3291.
- Shallue, C. J., Lee, J., Antognini, J., Sohl-Dickstein, J., Frostig, R. and Dahl, G. E. (2019). Measuring the effects of data parallelism on neural network training, *Journal of Machine Learning Research* **20**(112): 1–49.
- SPACE::LAB, Majirský, A., Mackovjak, S., Kostárová, S. and Amrich, S. (2024). Software: space-lab-sk/vigil-like-data: First release.
- Spogli, L., Alberti, T., Bagiacchi, P., Cafarella, L., Cesaroni, C., Cianchini, G., Coco, I., Di Mauro, D., Ghidoni, R., Giannattasio, F. et al. (2024). The effects of the may 2024 mother’s day superstorm over the mediterranean sector: from data to public communication, *Annals of Geophysics* **67**(2): PA218–PA218.
- Srivastava, N., Hinton, G., Krizhevsky, A., Sutskever, I. and Salakhutdinov, R. (2014). Dropout: a simple way to prevent neural networks from overfitting, *The journal of machine learning research* **15**(1): 1929–1958.
- Staub, J., Fernandez-Rico, G., Gandorfer, A., Gizon, L., Hirzberger, J., Kraft, S., Lagg, A., Schou, J., Solanki, S. K., del Toro Iniesta, J. C., Wiegelmann, T. and Woch, J. (2020). PMI: The Photospheric Magnetic Field Imager, *Journal of Space Weather and Space Climate* **10**: 54.
- Stepanova, M. and Pérez, P. (2000). Autoprediction of dst index using neural network techniques and relationship to the auroral geomagnetic indices, *Geofísica Internacional* **39**(1): 143–146.

-
- Sugiura, M. and Kamei, T. (1991). Equatorial dst index 1957–1986, *iaga bull.*, 40, *Berthelier and M. Menville (Int. Serv. Geomagn. Indices Publ. Off., Saint Maur, 1991)* **0**(0): 0–1.
- Sugiura, M. and Wilson, C. R. (1964). Oscillation of the geomagnetic field lines and associated magnetic perturbations at conjugate points, *Journal of Geophysical Research* **69**(7): 1211–1216.
- Tappin, S. J., Davies, J. A. and Eyles, C. J. (2023). Simulating diffraction effects in heliospheric imagers, *RAS Techniques and Instruments* **2**(1): 510–531.
- Vaswani, A. (2017). Attention is all you need, *Advances in Neural Information Processing Systems* .
- Verbeeck, C., Delouille, V., Mampaey, B. and De Visscher, R. (2014). The spoca-suite: Software for extraction, characterization, and tracking of active regions and coronal holes on euv images, *Astronomy & Astrophysics* **561**: A29.
- Vourlidas, A. (2015). Mission to the sun-earth l5 lagrangian point: An optimal platform for space weather research, *Space Weather* **13**(4): 197–201.
- Weigel, B., Batta, H. N., Faden, J. and jvandegriff (2021). hapi-server/client-python:.
- Weigel, R. S., Vandegriff, J., Faden, J., King, T., Roberts, D. A., Harris, B., Candey, R., Lal, N., Boardsen, S., Lindholm, C. et al. (2021). Hapi: An api standard for accessing heliophysics time series data, *Journal of Geophysical Research: Space Physics* **126**(12): e2021JA029534.
- Werbos, P. (1974). New tools for prediction and analysis in the behavioral science, *Ph. D. dissertation, Harvard University* .
- Wintoft, P. and Wik, M. (2018). Evaluation of kp and dst predictions using ace and dscovr solar wind data, *Space Weather* **16**(12): 1972–1983.
-

-
- Xu, S., Huang, S., Yuan, Z., Deng, X. and Jiang, K. (2020). Prediction of the dst index with bagging ensemble-learning algorithm, *The Astrophysical Journal Supplement Series* **248**(1): 14.
- Zboray, M. (2024). Využitie vysvetliteľných techník modelov hlbokého učenia v oblasti spracovania geofyzikálnych dát.
- Zeiler, M. (2014). Visualizing and understanding convolutional networks, *European conference on computer vision/arXiv*, Vol. 1311.
- Zhang, H., Verscharen, D. and Nicolaou, G. (2024). The Impact of Non-Equilibrium Plasma Distributions on Solar Wind Measurements by Vigil's Plasma Analyser, *Space Weather* **22**(2): e2023SW003671.
- Zhang, J., Feng, Y., Zhang, J. and Li, Y. (2023). The short time prediction of the dst index based on the long-short time memory and empirical mode decomposition–long-short time memory models, *Applied Sciences* **13**(21): 11824.
- Zhang, J., Richardson, I., Webb, D., Gopalswamy, N., Huttunen, E., Kasper, J., Nitta, N., Poomvises, W., Thompson, B., Wu, C.-C. et al. (2007). Solar and interplanetary sources of major geomagnetic storms ($\text{dst} \leq -100$ nt) during 1996–2005, *Journal of Geophysical Research: Space Physics* **112**(A10).

Appendices

Appendix A CD medium - final thesis in electronic form, manuals in electronic form and source code

Appendix B System manual

Appendix C supplementary .ipynb files in PDF format

Appendix D Publication