Interpretability of deep neural networks for SYM-H prediction

JP Beukes

orcid.org/ 0000−0002−6302−382X

Dissertation accepted in fulfilment of the requirements for the degree *Master of Engineering in Computer and Electronic Engineering* at the North-West University

Supervisor:	Prof Marelie H Davel
Co-Supervisor:	Dr Stefan Lotz

Graduation:	June 2021
Student number:	26028107

Declaration

I, Jacques Pieter Beukes hereby declare that the dissertation entitled "Interpretability of deep neural networks for SYM-H prediction" is my own original work and has not already been submitted to any other university or institution for examination.

Jacques Pieter Beukes

Student number: 26028107

Signed on the 10th day of December 2020 at Potchefstroom.

Acknowledgements

This research was performed within the Multilingual Speech Technologies (MuST) research group of the North-West University, which is a member of the Centre for Artificial Intelligence Research (CAIR) of the Department of Science and Innovation. It was supervised by Professor Marelie Davel and Doctor Stefan Lotz in association with the South African National Space Agency (SANSA). It was truly a privilege to be part of such a skilled group of researchers.

I would like to thank:

- Ulrike Janke, who meticulously attended to administration and always ensured a healthy working environment. She always made me feel welcome in the group.
- My supervisors Prof. Marelie and Dr. Stefan, whose guidance greatly contributed to the quality of this work and who taught me much more than writing a good dissertation.
- The developers of Mustnet, our in-house codebase which proved to be an essential tool for training and testing the deep neural networks in this work.
- My fellow students, for their friendship and helpful conversations.

To my friends, family, parents, grandparents and two sisters, thank you for your patience, encouragement and timely distractions, especially during the final stretch of my studies. To Dewald, my roommate, thank you for the six years of camaraderie. May there be many more.

Psalm 111:2 (NIV): Great are the works of the Lord; they are pondered by all who delight in them.

Abstract

Deep neural networks (DNNs) have shown impressive performance on a wide variety of applications, but it remains difficult to interpret these models. For regression modelling, DNNs generally do not explicitly provide any information about the utility of each of the input parameters in terms of their contribution to model accuracy. With this in mind, we develop the *pairwise network*, an adaptation to the multilayer perceptron (MLP) that allows the ranking of input parameters according to their contribution to model output. The technique is developed using synthetic data, before its application is demonstrated in the context of a space physics problem.

Geomagnetic storms are multi-day events characterised by significant perturbations to the magnetic field of the Earth, driven by solar activity. Previous storm forecasting efforts typically use solar wind measurements as input parameters to a regression problem tasked with predicting a perturbation index such as the 1-minute cadence symmetric-H (SYM-H) index. We re-visit the task of predicting SYM-H from solar wind parameters, with two 'twists': (i) Geomagnetic storm phase information is incorporated as model inputs and shown to increase prediction performance. (ii) We describe the pairwise network structure and training process – first validating ranking ability on synthetic data, before using the network to analyse the SYM-H problem. We found that the proposed pairwise network achieved slightly better performance than an MLP on this task while providing feature rankings that correspond to our understanding of the underlying physics.

Keywords: *SYM-H prediction, interpretability, Deep neural networks, geomagnetic storm*

Contents

List of Figures

List of Tables

List of Acronyms

ANN Artificial neural network

BSN Bow shock nose

CME Coronal mass ejection

DNN Deep neural network

Dst Disturbance storm time

GMD Geomagnetic disturbances

HRO High resolution OMNI

IMF Interplanetary magnetic field

MLP Multilayer perceptron

MSE Mean squared error

PCC Pearson correlation coefficient

RMSE Root mean squared error

RNN Recurrent neural network

ReLU Rectified linear unit

SE Standard error

SGD Stochastic gradient descent

Chapter 1

Introduction

In this chapter we give an overview of the study and discuss why it is relevant. We provide the scope and objectives of this study and describe the methodology followed.

1.1 Background

For many machine learning applications, good performance is not enough. It is often also necessary to verify that a model captures the underlying essence of the data and do not exploit superfluous components. The measure in which the decision-making process of a model can be explained is referred to as the interpretability of a model. Interpretability is critical in tasks where reliance on the correct features are important, such as self-driving cars or healthcare [1]–[3].

Deep neural networks (DNNs) have excellent performance on a wide range of tasks, from computer vision to speech recognition and language modelling [4]. However, due to their complexity, they are notoriously difficult to interpret. In resent years there has been a surge of research to make DNNs more interpretable [5]. The proposed methods can be

categorised as either extrinsic (techniques that extract explanations from the model using a separate method) or intrinsic (structurally interpretable) models.

In this work we introduce the pairwise network, an intrinsically interpretable DNN, based on the multilayer perceptron (MLP, the simplest form of DNN). The structure of the pairwise network enables feature ranking alongside prediction. We apply this network to a popular space weather problem and show that it matches the performance of an MLP, and that the ranking produced by the network corresponds to the underlying physics.

Violent eruptions of electromagnetic energy (solar flares) and charged plasma (coronal mass ejections or CMEs) on the solar surface are propagated through interplanetary space and can impact the Earth's geomagnetic field. These perturbations can result in the disruption of various kinds of technological systems: satellite [6] and high-frequency radio communications [7] are affected by the increased energy and particle density in the atmosphere and near Earth space; electrical faults can develop on spacecrafts due to anomalous charging [8]; and power grids, oil pipelines and ground-based telecommunication are affected by low-frequency currents induced by the changing geomagnetic field [9], [10]. Due to the adverse effects that damage to critical technological infrastructure can have on modern society, major efforts are being made to effectively monitor and predict space weather and its impact on specific technologies [11].

The intervals of geomagnetic activity that routinely causes the most intense disturbances are known as geomagnetic storms [12] – prolonged periods of significant perturbation to the geomagnetic field usually driven by CMEs. The intensity of geomagnetic storms are quantified by the net disturbance to the field, measured on Earth by any of the dedicated geomagnetic observatories[1] found on all six continents. There are several indices derived from magnetic field measurements to quantify certain aspects of the disturbances. In this work we use the symmetric horizontal (SYM-H) index, calculated at 1-minute cadence. It serves as an indication of the strength of the ring current which circles the Earth – the main driver of magnetic storm activity on the ground.

[1]For more information on the the global network of observatories monitoring the Earth's magnetic field, visit `https://intermagnet.github.io`.

To understand the drivers of geomagnetic disturbances (GMD), characteristic parameters of the solar wind plasma and magnetic field are analysed. These are measured upstream of the Earth by several satellites that orbit the first Lagrangian point (L1, see Figure 2.1) about 1.5 million kilometres upstream of the Earth. Solar wind propagation speed near 1 astronomical unit ($\approx 1.5 \times 10^8$ km; the distance from sun to Earth) range from about 300 km/s (quiet periods) to over 1 000 km/s (severely disturbed), and yields a natural lead time for predictions ranging from about 20 to 90 minutes. Therefore the prediction of some terrestrial index of GMD from measurements taken in the solar wind naturally lends itself to modelling as a regression problem, and as such many attempts have been made to provide forecasts of a variety of disturbance indices tailored to specific space weather effects [13]–[20].

1.2 Problem statement

DNNs have excellent predictive performance, but their complexity makes them difficult to interpret. Several techniques have been proposed to make DNNs more interpretable, but it still remains an active area of research [5]. Our goal is to develop an intrinsically interpretable neural network – the *pairwise network* – that enables feature ranking without significant sacrifices in performance. Specifically, we aim to see whether modelling each pair of input features semi-independently, provides a more interpretable model. Our goal is to use this network on a popular space weather regression problem to show that the network achieves similar predictive performance as an MLP trained on the same task, and that the ranking produced by the pairwise network agrees with our current understanding of solar wind – magnetosphere coupling during geomagnetic storms.

1.3 Project scope

Given the problem statement, we restrict this study to regression modelling using a limited number of architecture types and data sets:

- **Data set:** To investigate the interaction between the solar wind and the SYM-H index, we develop a data set that consists of 8 plasma and magnetic field parameters (V_{sw}, N_p, P_d, E_M, $B_{X,Y,Z}$ and B_T) as input features and SYM-H as the target of a regression problem. The selected solar wind parameters all contribute to some extent to the dynamics involved in driving a geomagnetic storm. (See Section 2.2.2 for a description of the parameters.) This data set is the central focus of our study, but we also generate several synthetic data sets to validate the training and ranking procedure of the proposed pairwise network.

- **Architecture types:** The pairwise network is the main architecture under consideration, but to evaluate its performance on the solar wind/SYM-H data set, we also construct a baseline by training MLPs on this data set. We acknowledge that recurrent and convolutional structures are better suited for sequence modelling, but for the sake of simplicity, we decide to only focus on the above mentioned feed-forward neural networks.

With this study, it is not our goal to create a state-of-the-art interpretable network, but to explore a novel conceptual approach to intrinsically interpretable neural networks. We aim to show that the pairwise network and the MLP are both viable options for modelling geomagnetic storms and that the ranking produced by the pairwise network corresponds to existing knowledge in the field of space physics. It is outside the scope of the current study to gain new physical insight, but our intention is to develop and test a technique that could be used to do just that. We also do not aim to develop a feasible system for operational SYM-H forecasting.

1.4 Research questions

With the above mentioned aim in mind, we formulate the following research questions:

- How can one construct, train and evaluate an artificial neural network (ANN) that is capable of producing feature rankings alongside predictions?

- Can this network achieve approximately the same performance as an MLP on the solar wind/SYM-H task?
- Does the feature ranking produced by this network correspond to the current understanding of space physics?
- Does temporal and geomagnetic storm phase information improve the predictive performance of these models?

1.5 Objectives of the study

To address the research questions, we set the following objectives:

- Create a data set with solar wind parameters as input and the SYM-H index as output, curated for public release.
- Develop and optimise different MLP models on this data set, evaluating the effect of hyperparameter choices and experimenting with additional features. The MLP will be used as a benchmark for comparing the performance of pairwise networks.
- Develop the pairwise network:
  - Construct an architecture that enables feature ranking alongside prediction.
  - Generate several synthetic data sets for which the input-output relationship is known.
  - Validate the training and ranking procedure on the synthetic data: optimise the hyperparameters of the model on the data sets to determine if the network is able to reach acceptable performance on the data, extract feature rankings and compare it to the true data generating process.
- Develop a pairwise network for the solar wind/SYM-H data set:
  - Optimise different pairwise networks on this data set, evaluating the effect of hyperparameter choices and experimenting with additional features.

  - Compare the optimised pairwise network's performance to that of the baseline MLPs to determine if the network is a viable option for modelling geomagnetic storms.
  - Extract feature rankings and compare it to the known physical processes involved and determine if the network has learned to model the task in a way that corresponds with the physics.

1.6 Research methodology

This investigation is an empirical analysis of a novel neural network architecture. The following will form part of this study:

- **Literature study:** A review of space weather will be done to acquire the necessary knowledge of the solar wind, SYM-H index and geomagnetic storms. The basic principles pertaining to DNNs, such as the structure and optimisation of MLPs, as well as previous work done on the interpretability of DNNs will be reviewed. We will also review previous work on DNNs for modelling geomagnetic disturbance indices.
- **Development environment:** To do our experimental analysis, we will develop three codebases (see Section 2.4 for more information):
  - Mustnet, a PyTorch-based in-house codebase for DNNs training and analysis, will be used for MLP optimisation.
  - An independent Python package will be developed for downloading, generating and preprocessing the solar wind/SYM-H data set.
  - A second Python package will be developed for training and analysing pairwise networks.
- **Experimental procedure:** To develop and assess the pairwise network, we will:
  - Validate the pairwise network using synthetic data: optimise the hyperparameters on the network, determine if the network performs well, and extract the

feature ranking of each task to determine if it corresponds with the true data generating process.

  - Search for the optimal set of hyperparameters for both the pairwise network and the MLP on the solar wind/SYM-H data set.
  - Compare the performance of the optimised pairwise network to the MLP on the solar wind/SYM-H data set.
  - Extract and analyse the feature ranking produced by the pairwise network.

- **Evaluation:** Assess the feature ranking produced by the pairwise network by comparing it to the known physical processes involved.

1.7 Dissertation overview

This dissertation has the following structure:

- Chapter 2 provides the necessary background information for the rest of the dissertation and discusses work that relates to this study.
- Chapter 3 describes the main data set used for this study: how it is generated and the characteristics of the input features and target parameter.
- Chapter 4 describes the procedure and results for training MLP baselines on the data set.
- Chapter 5 introduces the pairwise network. We describe the training and ranking procedure and validate it on synthetic data sets.
- Chapter 6 applies the pairwise network to the solar wind/SYM-H data set and shows that it achieves similar performance. We also extract feature rankings and evaluate it against the existing physical understanding of the problem.
- Chapter 7 concludes with a summary of the key findings of the investigation.

1.8 Publications

Much of this dissertation's content has been published in 3 previous works:

- A poster titled "A neural network based method for input parameter selection" for the Helio ML conference of 2019 [21].
- A conference paper titled "Input parameter ranking for neural networks in a space weather regression problem" for FAIR2019 [22].
- An extended version of [22] published in the FAIR2019 special issue of the South African Computer Journal (SACJ) with the title "Pairwise networks for feature ranking of a geomagnetic storm model" [23].

The Helio ML poster overlaps with Chapter 6 of this dissertation, where we apply an improved version of the proposed network to the same solar wind/SYM-H task. Our literature study on space weather (Chapter 2), the MLP optimisation procedure (Chapter 4), and the application of the pairwise network on the solar wind/SYM-H data set (Chapter 6) is adapted from the FAIR2019 and SACJ article with some improvements. Development of the pairwise network using synthetic data sets (Chapter 5) is taken from the SACJ paper.

Chapter 2

Background

In this chapter we provide background information related to this study and introduce several concepts that is necessary for this dissertation. We also do a literature review of related work.

2.1 Introduction

In this chapter we provide background information for this dissertation. We discuss key concepts of space weather, such as the solar wind and SYM-H index, and review previous attempts at modelling geomagnetic storms with deep neural networks (DNNs). A short overview of DNNs is given, where we discuss the structure, optimisation and related work on the interpretability of DNNs. Other essential concepts pertaining to this investigation, such as sparsity and standardisation, are also defined here. The frameworks used for training and analysing DNNs are listed at the end of this chapter.

2.2 Space weather

Electromagnetic and corpuscular radiation are the two main types of energy emitted by the Sun. The latter constitutes the solar wind, a stream of charged atoms and sub-atomic particles expanding out into the Solar System and carrying the Sun's magnetic field with it. Upon reaching Earth, it interacts with the geomagnetic field, compressing its dipole shape on the day side and elongating the night side, as depicted in Figure 2.1 [24]. The bow shock is a shock wave that forms upstream of the magnetosphere, which slows the solar wind and diverts it around the Earth [24]. The first Lagrangian point (L1) is situated upstream of the magnetosphere. At this point, the gravitational forces of Earth and the Sun cancel out, allowing satellites on this point to orbit the Sun at the same rate as the Earth.

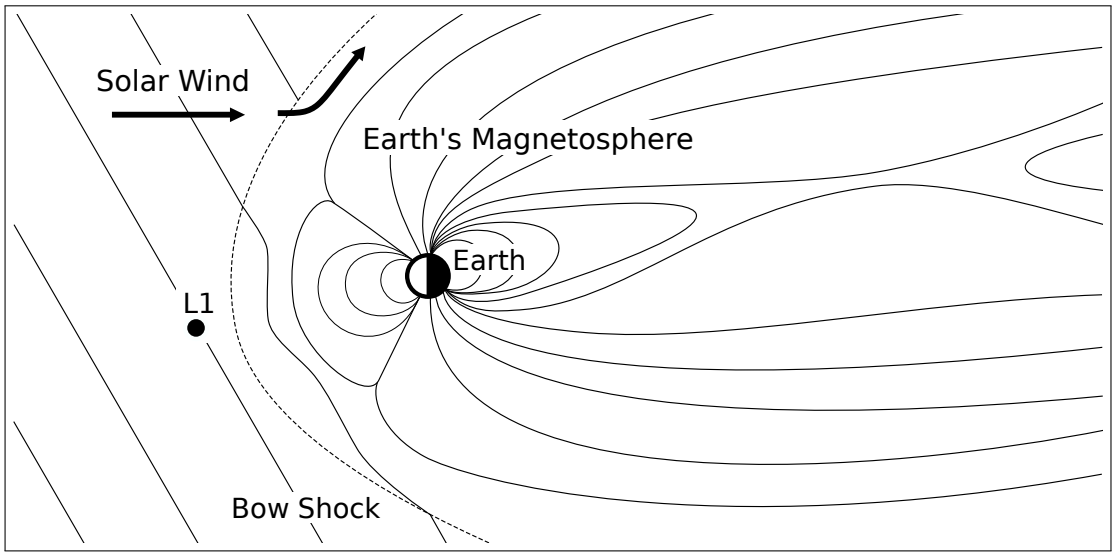


Figure 2.1: A simplified cross section of Earth's magnetosphere, showing how the solar wind compresses the geomagnetic field's dipole shape on the day side (left, facing the Sun) and elongates it on the night side (right, away from the Sun). L1 indicates the first Lagrangian point. This diagram is adapted from [24].

Violent eruptions on the Sun's surface, such as coronal mass ejections (CMEs) and solar flares eject large amounts of high-velocity particles into the solar wind. When impacting Earth's geomagnetic field it can cause prolonged periods of significant perturbation, known as geomagnetic storms.

A severe storm can result in the disruption of various kinds of technological systems: satellite [6] and high-frequency radio communications [7] are affected by the increased energy and particle density in the atmosphere and near Earth space; electrical faults can occur on spacecrafts due to anomalous charging [8]; and power grids, oil pipelines and ground-based telecommunication are affected by low-frequency currents induced by the changing geomagnetic field [9], [10]. Due to the adverse effects that damage to critical technological infrastructure can have on modern society, significant efforts are being made to effectively monitor and predict space weather and its impact on technology [11].

2.2.1 SYM-H index

Several indices exist to reflect certain aspects of geomagnetic disturbances. In this work we use the symmetric horizontal (SYM-H) index to quantify the intensity and progression of a storm. SYM-H is calculated at 1-minute cadence, derived from the horizontal (with regard to Earth's surface) component of the geomagnetic field measured at 11 middle latitude magnetic observatories [25]. The geographic locations of these observatories is depicted in Figure A.1. It serves as an indication of the strength of the ring current which circles the Earth and is the main driver of magnetic storm activity on the ground. This perturbation index is developed to remove local variation as a result of solar radiation from the ionosphere [24]. SYM-H is computed at WDC (World Data Center) for Geomagnetism[1] at Kyoto University, Japan and forms part of the high resolution OMNI HRO data set (Section 3.2). There also exists an asymmetric horizontal (ASY-H) index, derived by subtracting the SYM-H from each disturbance field at each station, but we chose not to incorporate it in this work [25].

2.2.2 Solar wind parameters

This study includes several plasma and magnetic field parameters, all of which contributes to some extent to the dynamics of a geomagnetic storm:

[1] `http://swdcwww.kugi.kyoto-u.ac.jp`

V_{sw} Solar wind speed [km/s] is the bulk speed of the plasma moving across the spacecraft.

N_p Proton number density [#/cc] measured in particles per cubic centimetre indicates the particle density of the plasma. Coronal mass ejecta are usually more dense than the ambient solar wind plasma.

P_d Dynamic flow pressure [nPa] is the flow pressure of the solar wind and is linearly related to $N_p V_{sw}^2$.

E_M Merging electric field in the solar wind [mV/m] serves as an indication of the coupling between the solar wind and magnetospheric plasmas and is linearly related to $-V_{sw} B_Z$.

$B_{X,Y,Z}$, B_T The three components of the interplanetary magnetic field (IMF), measured in nT, and the total field B_T.

These parameters are measured by several satellites, such as the advanced composition explorer[2] (ACE) and the deep space climate observatory[3] (DSCOVR), that orbit the first Lagrangian point (L1) about 1.5 million kilometres upstream of the Earth. The reason for this position is to allow satellites to always observe incoming solar wind before it reaches Earth. Solar wind propagation speed near 1 astronomical unit ($\approx 1.5 \times 10^8$ km; the distance from Sun to Earth) range from about 300 km/s (quiet periods) to over 1 000 km/s (severely disturbed), and yields a natural lead time for predictions ranging from about 20 to 90 minutes.

2.2.3 Storm phases

A typical geomagnetic storm is seen in the SYM-H curve of Figure 2.2. This event was caused by a CME that erupted at about 02:13 UTC on 15 March 2015 on the surface of the Sun and arrived at the magnetosphere at approximately 04:05 UTC on 17 March [26]. Upon the arrival of the CME, the storm starts with the *onset phase*, indicated by a

[2] `http://www.srl.caltech.edu/ACE`
[3] `https://www.nesdis.noaa.gov/content/dscovr-deep-space-climate-observatory`

clear peak in SYM-H. Next, B_Z turns southward and remains so for a prolonged period – indicative of the *main phase*. Southward IMF enables enhanced coupling between the solar wind and magnetospheric plasma, resulting in more efficient energy transfer from the solar wind to the geomagnetic field. After the IMF turns northward ($B_Z > 0$) and the bulk of the disturbed solar wind plasma has passed, the magnetosphere can recover (the *recovery phase* of the storm).

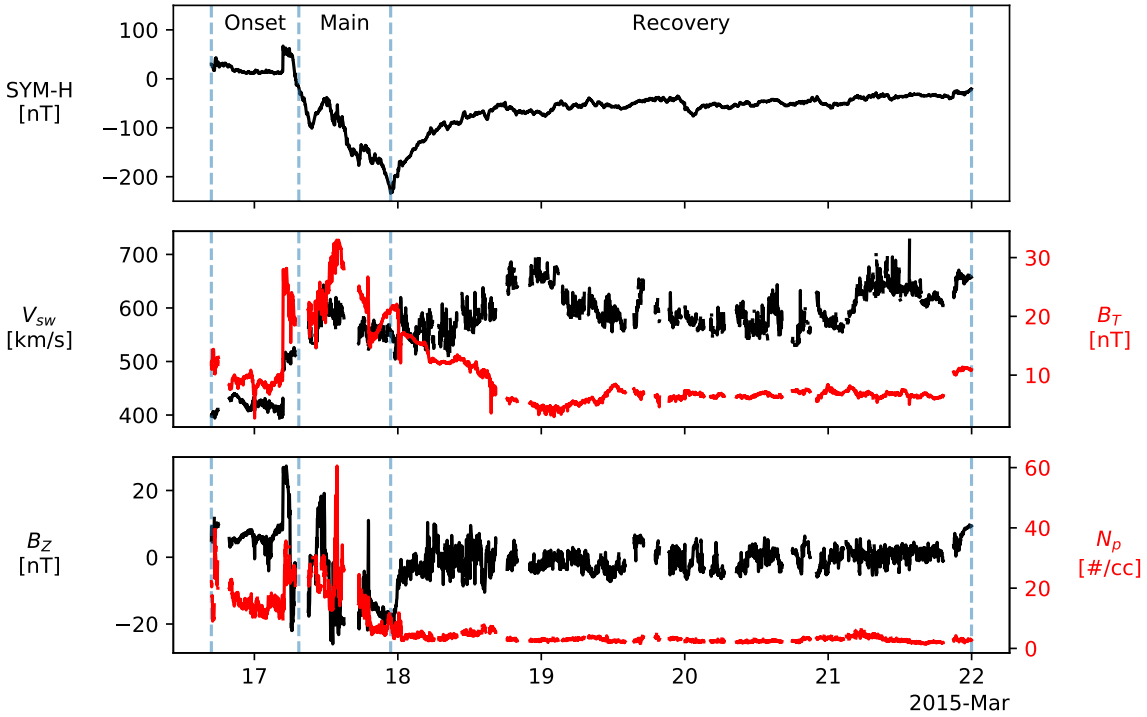


Figure 2.2: A typical geomagnetic storm driven by a single CME. This particular event occurred on 17-21 March 2015 and is known as the St. Patrick's day storm. The top panel shows SYM-H and the lower panels show solar wind parameters V_{sw}, B_T and B_Z, N_p respectively. Phase transitions are indicated with dashed vertical lines and each phase is labelled in the top panel.

2.2.4 Akasofu's ϵ

This parameter serves as an indication of the amount of energy (in Watts) deposited by the solar wind into the magnetosphere [27]. It is defined by

$$\epsilon = \frac{2\pi}{\mu_0} V_{sw} B_T^2 \sin^4(\frac{\theta}{2}) l_0^2, \tag{2.1}$$

with $\theta = \tan(B_Y/B_Z)$ the clock angle of the IMF, μ_0 the vacuum permittivity, and l_0 a scaling factor determined empirically. In Section 5.3.2 a simplified version of this parameter is modelled, with solar wind parameters as input:

$$\epsilon^* = V_{sw} B_T^2 \sin^4(\frac{\theta}{2}), \tag{2.2}$$

The ϵ^* parameter simply avoids the scaling constants, but the relationship with solar wind parameters remains intact. This regression problem enables the evaluation of our pairwise network on a complex, but known relationship between input and output parameters, utilising real-world data.

2.2.5 Storm prediction (previous work)

Several attempts have been made to predict geomagnetic indices from solar wind parameters using deep learning. Here we list studies that relate to ours. We include disturbance storm time (Dst) index prediction since Dst is approximately equivalent to SYM-H, but at hourly intervals [28]. The parameters used in the rest of this section are described in Section 2.2.2.

Lundstedt and Wintoft [13] were of the first to use an artificial neural network (ANN) for Dst prediction. They developed a multilayer perceptron (MLP) with B_Z, N_p and V_{sw} as input and were able to predict Dst 1 h ahead. The network modelled the onset and main phase of storms well, but could not predict the recovery phase accurately. Gleisner et al. [14] used the same input and target parameters, but were able to improve performance on the recovery phase by adding time-shifted inputs to the MLP. Kugblenu et al. [15] were also able to reproduce the recovery phase, by using past values of Dst, together with B_T,

B_Z and $\sqrt{N_p V_{sw}^2}$ as inputs. Bala and Reiff [16] managed to predict Dst 1 to 6 h ahead by using the Boyle index (combination of V_{sw}, B_T, and the clock angle of the IMF) as input.

Other ANN-based models used to predict Dst from solar wind parameters include recurrent neural networks (RNNs) [17]–[19] and ensemble methods [29]. Gruet et al. [20] developed a long short-term memory neural network (LSTM) capable of forecasting Dst 1 to 6 hours ahead from the solar wind, with a correlation coefficient higher than 0.873 and a root mean square error (RMSE) lower than 9.68 nT.

The so-called NARX (non-linear auto regressive with exogenous inputs) neural network has been shown to be capable of predicting SYM-H from solar wind parameters. Cai et al. [30] developed such a model for 1 hour ahead forecasting with a correlation of 0.91 and a RMSE of 14 nT. Their data set included moderate (-100 nT $<$ SYM-H $<$ -85 nT) and intense (SYM-H $\leq$ -100 nT) storms between 1998 and 2005. Bhaskar and Vichare [31] added storms from 2006 to 2015, but it did not translate to a significant increase in performance.

Sciliano et al. [32] compared the SYM-H forecasting ability of LSTMs to that of convolutional neural networks (CNNs). They identified extremely intense storms (SYM-H $<$ -200 nT) between 1998 and 2018 and was able to achieve 1 hour ahead forecasting with an RMSE of less than 10 nT for both models, when including SYM-H in the set of input features.

2.3 Deep neural networks

Deep learning has become a catch-all term for any machine learning technique that use ANNs, a type of machine learning model that consists of a collection of nodes connected by trainable weights and followed by non-linear activation functions. It gained much momentum in recent years, achieving state of the art performance on a wide range of applications: computer vision, natural language processing, speech recognition and many more [4].

The simplest form of an ANN is the MLP, a function approximator for both classification or regression tasks [33]. In this work we focus on regression, so the network takes the form $y = f(x;\theta)$ and learn parameters θ to best approximate f^* (the true function). It maps input parameters to output values through one or more hidden layers, each having multiple hidden nodes.

Every node in the network is the weighted sum of all the nodes in the preceding layer, followed by a non-linear activation function (σ). A popular activation function is the rectified linear unit (ReLU) [34]: $max(0\,;z)$, where z is the output of a node before the activation. Other activations exist (such as sigmoid, tanh, exponential linear unit [35] and leaky ReLU [36]), but we choose ReLU for its good performance on a wide range of applications [34], [37]–[39].

For an example of a simple MLP, consider the network depicted in Figure 2.3: 2 inputs, a single output, 2 hidden layers with 3 nodes per layer, and a bias (b) connected to the first hidden layer. The output of the network is then $y = \sigma(W_3^T h_2)$, where $h_2 = \sigma(W_2^T h_1)$ and $h_1 = \sigma(W_1^T x + b)$. W_i is the weight matrix of layer i, where the weight from node k in layer $i-1$ to node j in layer i is denoted as $w_{i,j,k}$. Notice that, when placed in the first layer, the bias is equivalent to a weight connected to an input feature with a constant value of one.

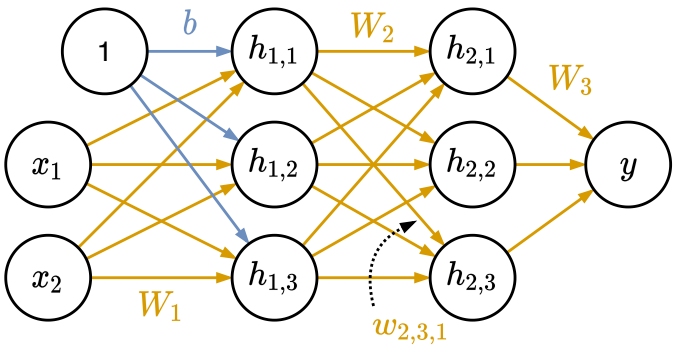


Figure 2.3: The architecture of a simple MLP with 2 inputs, 2 hidden layers and a single output.

2.3.1 Optimisation of deep neural networks

ANN optimisation refers to the process of adjusting network parameters so as to minimise the loss of the network. Loss is a measure of the error of a model's prediction relative to the observed target value. Mean squared error (MSE) is a common choice for calculating the loss of regression models [40].

When using gradient descent to optimise ANNs, each parameter of a network is updated proportional to the negative of the gradient of the loss with respect to that parameter:

$$\Delta w = -\eta \frac{\partial L}{\partial w}, \tag{2.3}$$

where L is the loss and η the learning rate. Unlike normal gradient descent, the expected gradient is approximated using stochastic gradient descent (SGD). A mini-batch of randomly selected samples are passed through the network and the average gradient is used to do parameter updates. SGD is an important optimisation technique for DNNs since it is more computationally efficient than normal gradient descent, and therefore enables the use of large training sets [33]. Adam [41] is a variant of SGD, popular for its adaptive estimates of lower-order moments [33].

The correct parameter initialisation scheme is important for good performance and stability during training [33]. A popular option for ReLU-activated feed forward networks is the Kaiming scheme [42]. The implementation of this scheme used in our work initialises each layer of a network by sampling the weights values from $\mathcal{U}(-b, b)$ where

$$b = 3\sqrt{\frac{2}{n}}, \tag{2.4}$$

when using a uniform distribution or $\mathcal{N}(0, \sigma^2)$ where

$$\sigma = \sqrt{\frac{2}{n}}, \tag{2.5}$$

when using a normal distribution. In this context, n is the *fan-in* – the number of connections coming into a given layer from the previous layer's output. Biases are initialised with a value of zero [42].

Batch normalisation is known to improve optimisation of DNNs and the noise introduced by it can also have a regularising effect [33]. To normalise the batch of activations of a layer, denoted with H, replace it with

$$H' = \frac{H - \mu}{\sigma}, \tag{2.6}$$

where μ and σ are vectors respectively containing the mean and standard deviation of each node in the layer. When using batch normalisation with an affine transform, H' is multiplied with α and subtracted with β, which are both learnable parameters to be optimised by the training algorithm. It can be applied to any input of hidden layer of a network and is usually an additional layer that normalises the output of the previous linear layer [33].

2.3.2 Interpretability of deep neural networks

In many applications, such as healthcare [2], [3] or self-driving cars [1], it is critical to verify that a machine learning model captures the underlying essence of the data and do not exploit superfluous components. Understanding what a model has learned would also allow practitioners to expose biases in data or extract new insights of the underlying physical process [43], [44].

Deep neural networks show excellent performance on a wide variety of tasks, but their complexity makes it difficult to realise how input features map to output values. Several attempts have been made to remedy this problem and can be categorised as either *intrinsic* or *extrinsic* interpretability techniques. For a comprehensive survey, we refer the reader to [5].

Extrinsic Interpretability

We define extrinsic interpretability (also known as "explainability" or "post-hoc interpretability" [45]) as the ability to explain model behaviour using techniques that do not

form part of the model's structure. A popular approach is the use of attribution methods that assign an importance score to each input feature through model inference.

The most intuitive of these are perturbation-based attribution methods, which occlude parts of a sample, pass it through the model and measure the difference with the original output [46], [47]. Perturbation-based methods provide a direct estimate of the marginal effect of a feature, but is very sensitive to the number of features removed and becomes computationally expensive when the total number of features increase [48].

Gradient-based methods compute the attribution of a feature using the gradient of the output with respects to that feature. Examples include Integrated Gradients [49], DeepLIFT [50], Layer-wise Relevance Propagation (LRP) [51], Grad-CAM [52] and SHAP [53]. These methods have the benefit of being incorporated in the backpropagation algorithm, making it generally faster than perturbation-based methods [48].

Attribution methods are attractive since they can be readily applied to a wide range of architectures with little to no modification. However, it remains an active area of research as several concerns have been raised with regards to their reliability, evaluation and clarity [45], [54]–[56].

Intrinsic Interpretability

The pairwise network introduced in this work are intrinsically interpretable, i.e. the structure of this network allows for feature ranking. Similar ANN architectures exist in the literature, most of which are based on the generalized additive model (GAM) [57], first introduced as "progression pursuit regression" [58]. GAMs take the form:

$$g(\mathbb{E}[y]) = \beta + \sum_{i=1}^{M} f_i(x_i), \tag{2.7}$$

where y is the target, $x = (x_1, x_2, \ldots, x_M)$ is the input with M features, g is called the link function and f_i is a univariate shape function.

Generalized Additive Neural Networks (GANNs) were introduced in [59] and is the first attempt to use neural networks for approximating the shape functions of GAMs. However, the GANNs used in [59] were not trained with back-propagation and only consisted of a single hidden layer. All of the networks that follow benefit from recent advances in deep learning (discussed in Section 2.3).

Neural Additive Models (NAMs) [60] improve GANNs significantly with several architectural and optimisation changes: Several regularisation techniques are employed; the hidden units within NAMs take the form $h(x) = f(e^w * (x - b))$ in order to learn weights in the logarithmic space; and a non-standard activation function (ReLU-n [61]) and initialisation scheme are used.

GAMI-Net (GAMs with structured Interactions) [62] bears the closest resemblance to our pairwise network and was developed in parallel to our work. It is based on GA^{2}Ms (GAMs plus Interactions) [63]:

$$g(\mathbb{E}[y]) = \beta + \sum_i f_i(x_i) + \sum_{i,j} f_{i,j}(x_i, x_j), \tag{2.8}$$

where $i \neq j$. Therefore, GAMI-Net essentially adds pairwise interactions ($f_{i,j}$) to the main effects (f_i) used in GANNs. Several interpretability constraints are applied during GAMI-Net training, so as to force sparse representations and more reliance on main effects than pairwise interactions. The differences between GAMI-Net and our pairwise network is discussed in Section 5.2.

Neural Interaction Transparency (NIT) [64] is a framework that starts with an MLP and then disentangles interactions to form a GAM, GA^{2}M or GA^{K}M (GAM with K interactions).

The Explainable Neural Network (xNN) [65], Adaptive xNN (AxNN) [66] and Enhanced xNN (ExNN) [67] are all based on the Generalized Additive Index Model (GAIM):

$$g(\mathbb{E}[y]) = \beta + \sum_{i=1}^{k} h_i(w_i^T x), \tag{2.9}$$

where w_i are known as the projection indices and k is chosen to be sufficiently large. In the latest version (ExNN), several regularisers are applied to force the projection indices to be sparse and the network to have smooth function approximation.

Except for GANN (1999), NIT (2018) and xNN (2018), all of these techniques were developed in parallel with ours, with ExNN and preprints of NAM, GAMI-Net and AxNN published in 2020. A comparative study on the strengths and weaknesses of these techniques are still lacking in the field.

2.3.3 Sparsity

Deep neural networks are often extremely large, especially state of the art models that have in the order of millions of parameters [68]. These models require a lot of computation and memory, increasing infrastructure costs and making it challenging to implement on resource-constrained environments such as embedded systems and mobile devices. Consequently, much effort has been made to develop smaller, but well-performing networks.

A popular approach to address this issue is to introduce sparsity within a model. Sparsity refers to the parameters removed from a model. It is achieved by starting with an over-parameterised model and pruning superfluous parameters. For ANNs, that means setting the appropriate weights to zero and preventing further updates. Previous work shows that deep neural networks can be pruned to high levels of sparsity without sacrificing performance [69]–[71].

Numerous pruning techniques exist. We refer the reader to a survey by Blalock et al. [72] for recent advances. A simple, yet effective method is proposed by Han et al. [71]: After the initial training phase, they remove all the weights that fall below a predetermined threshold and then train the remaining weights further (fine-tuning). This process of pruning and fine-tuning can be repeated several times to gradually increase sparsity.

2.4 Development environment

Experiments were carried out using PyTorch [73], an open source machine learning framework that enables the development of deep neural networks, with functionality such as automatic differentiation and GPU (graphical processing unit) acceleration. Mustnet[4], a PyTorch-based in-house codebase, was used for MLP optimisation. An independent Python package was developed for downloading, generating and preprocessing the solar wind/SYM-H data set[5]. This repository also contains code for the analysis done on this data set. Training and analysis of the pairwise network are contained in yet another repository[6], which is also structured as a Python package to allow for effortless importing of the pairwise network in future work. Weights and Biases[7] (W&B) was used extensively for pairwise network experiments. W&B provide several development tools for machine learning, such as experiment tracking, model optimisation and data set versioning.

2.5 Conclusion

In this chapter we provided an overview of the key concepts in space weather and deep learning that relates to this study. We described the SYM-H index and several solar wind parameters, which we will use in the next chapter to construct a data set for this study. Previous attempts at geomagnetic storm modelling using DNNs were discussed as well as related studies on the interpretability of DNNs. We introduced several concepts that will be used throughout the dissertation and listed the development environment used in this study.

[4] `https://bitbucket.org/must_research/mustnet3`
[5] `https://bitbucket.org/must_research/sansa`
[6] `https://bitbucket.org/must_research/pairwise_network`
[7] `https://wandb.com`

Chapter 3

Data set

In this chapter we introduce the solar wind/SYM-H data set developed for this study. We describe the input and target parameters, data source, preprocessing steps and provide useful statistics about the data.

3.1 Introduction

A new data set is created for the purpose of studying the relationship between the solar wind and Earth's geomagnetic field during intense geomagnetic storms using artificial neural networks (ANN). We frame it as a regression problem, with 8 solar wind parameters as input and the SYM-H index as the target. (A description of these parameters is provided in Sections 2.2.1 and 2.2.2.)

Data is collected from a publicly available source with some preprocessing steps taken care of. However, the data is still unbalanced, not partitioned or normalised and contain missing values. To balance and partition the data, we isolate geomagnetic storms and then randomly divide the set of storms into a training, validation and evaluation set. We

also add optional temporal and storm phase information with time-shifted inputs and a categorical variable, respectively. Linear interpolation is employed to replace most of the missing values and any remaining missing values are discarded. Lastly, we briefly analyse the data to verify that the data is sound and to get an idea of the nature of the problem.

3.2 OMNI data source

Data is collected from the high resolution OMNI (HRO) data set [74], which combines measurements taken from several satellites and includes both solar wind parameters and several geomagnetic disturbance indices, of which SYM-H is one.

This data set is already preprocessed in the following steps: Solar wind measurements taken at 16 or 64-second cadence (depending on the instrument) are averaged to 1-minute values and shifted in time to the estimated position of the magnetospheric bow shock nose (BSN) [74]. This ensures that the propagation time from the first Lagrange point (L1, see Figure 2.1) to the BSN does not have to be incorporated into model development.

3.3 Event selection

The time period chosen for this study is 2000-2018. It includes almost two full solar cycles, but since geomagnetic storms are fairly rare events, using all available data would then result in a very unbalanced data set, with the storm periods being under-represented. For example, during solar cycle 23 (1997 - 2007) 90 intense geomagnetic storms were identified, with minimum Dst of less than -100 nT and an average length of 80 hours each [75]. Moreover, it is known that during quiet times the connection between the solar wind and magnetosphere is very weak, as there is no direct coupling due to day-side reconnection [12].

Since it is not our goal to develop a feasible system for an operational setting, but to investigate the relationship between the solar wind and SYM-H during storms, only in-

tense events (storms with a minimum SYM-H < -100 nT) are selected out of the 19 year period.

Storm identification is done according to the algorithm described in [76]. The start of a storm's main phase is defined as the first instance before SYM-H < -100 nT, when SYM-H > -20 nT. To include the onset phase, the start time is pushed back to the point where SYM-H reaches a maximum in the 24 hours preceding the main phase. 12 hours of data are added prior to the onset phase and marks the start of a storm. The end of a storm is defined as the point where SYM-H returns to a level greater than -20 nT. No explicit padding is added to the end of a storm since this definition of the recovery phase ensures that a significant interval of diminished solar wind and magnetosphere coupling is already included in the event.

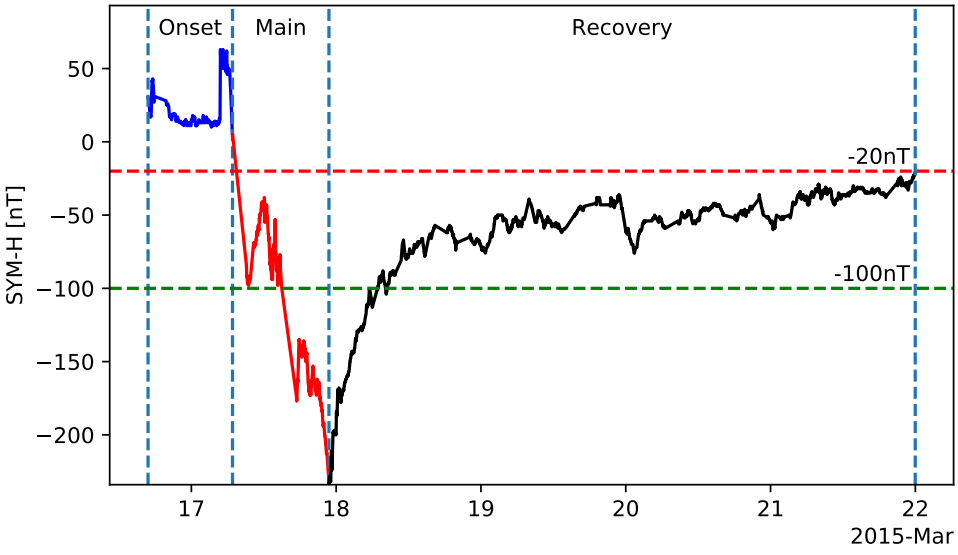


Figure 3.1: SYM-H progression during a typical geomagnetic storm. Horizontal lines indicate the -20 nT and -100 nT thresholds and the vertical lines show the boundaries of each storm phase. The onset, main and recovery phases are labelled as such.

Using the method described above, 96 storms are identified, resulting in 479 989 minutes of data (before preprocessing missing values) out of a possible $\sim 1.1 \times 10^7$ minutes (2000-2020). It should be noted that 8 of these events actually consist of 2 successive intense storms, but in practice we regard them as single events when identifying storm phases

(in Section 3.5). Storms that occurred in 2019, between 2007 and 2010 and between the start of 2020 and the time of this writing are not significant enough to be captured by this algorithm.

The collection of distinct storms are shuffled before dividing into training (68 storms), validation (14 storms) and evaluation (14 storms) sets. Keeping storm intervals separate ensures that the three sets are truly independent – every storm interval is wholly contained in only one of the three data sets. Shuffling ensures that each data set contains storms at different parts of the solar cycle. Table 3.1 summarises the number of samples and storms in each partition.

Table 3.1: Number of samples and storm events in each partition of the SYM-H data set, before missing values are processed.

Partition	**Ratio**	**Samples**	**Storms**
Training	73.91%	354 779	68
Validation	13.64%	65 485	14
Evaluation	12.44%	59 725	14
	Total	**479 989**	**96**

Measured across the entire data set, the shortest storm is 2 005 minutes, the longest is 14 487 minutes and the average storm length is 5 000 minutes, with a standard deviation of 2 213 minutes. Figure 3.2 presents other descriptive statistics in the form of a box plot.

3.4 Time-shifts

To enable the model to capture time information, we add temporally shifted versions of each parameter to the set of inputs. For each input parameter X_t, measured at time t, another input X_{t-m} is added, where m is the time-shift size and should be less than 720 minutes (12 hours) to fall within the amount of padding allocated to each storm. Time-shifts are added to each storm in isolation, ensuring that there is no overlap between storms. Missing values at the endpoints of X_t and X_{t-m}, resulting from the shifting operation, are removed by truncating the series.

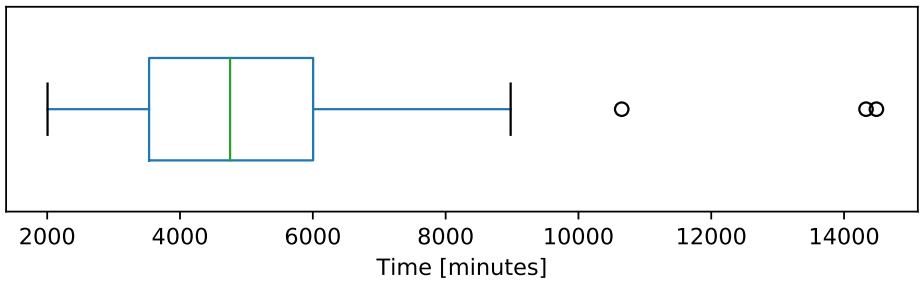


Figure 3.2: Box plot of the storm duration for the entire data set, before missing values are processed. The median and interquartile range (IQR) are indicated by the body of the box plot. Outliers (samples that do not fall within 1.5 IQR above the upper quartile) are indicated with circles and the whiskers of the box plot show the minimum and maximum storm duration if outliers are ignored.

3.5 Storm phases

It is known that different physical phenomena are at play during different phases of a geomagnetic storm (as discussed previously in Section 2.2). In our analysis we want to determine whether simply including phase information as a categorical variable in the set of input parameters improves prediction performance.

Within each storm the onset, main and recovery phases are identified by searching for the (i) interval around the positive increase in SYM-H (onset phase), (ii) the rapid decrease to storm minimum (main phase), and finally the (iii) recovery period from the minimum SYM-H until the end of the event. Figure 3.1 shows SYM-H during a typical storm, and the three phases identified by this algorithm. Note that this is a post-hoc process and cannot be implemented in an operational setting.

Figure 3.3 shows how the distribution of B_Z, V_{sw} and SYM-H change during each storm phase for the entire data set. B_Z is mostly negative during the main phase, but mixed during the onset and recovery phase. V_{sw} increases significantly after the onset phase, while SYM-H decreases from onset to main and then increases in the recovery phase. Statistics to indicate the typical duration of each phase of a storm is presented as box plots in Figure 3.4. The recovery phase is significantly longer than the other two.

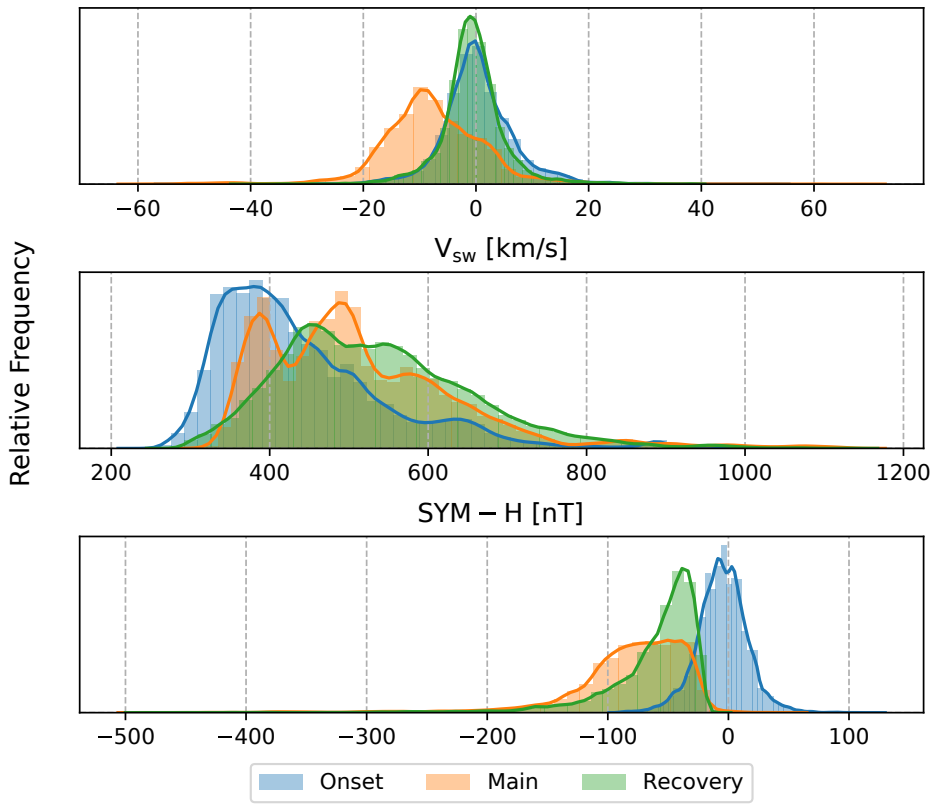


Figure 3.3: Distributions of B_Z, V_{sw} and SYM-H for each phase over the entire data set, before missing values are processed.

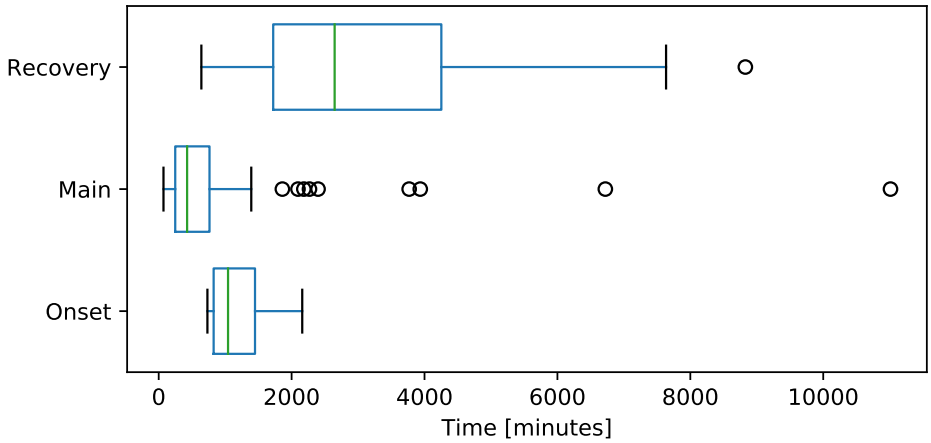


Figure 3.4: Box plot for the duration of each phase over the entire data set, before missing values are processed.

3.6 Missing values

In this section we analyse the number of missing values of the data set. We classify a sample as missing if at least one of its features are missing. None of the SYM-H values are missing, so we focus only on the solar wind parameters.

The percentage of missing values per feature per year and the percentage of missing samples per year are provided in Table 3.2. Note that two distinct groups are formed: interplanetary magnetic field (IMF) parameters ($B_{X,Y,Z}$, B_T) and plasma parameters (V_{sw}, N_p, P_d, E_M). Features in the same group typically have errors at the same time since the plasma or magnetic field instruments are independent [77].

Table 3.2: Percentage of missing features and samples per year.

Year	B_T	B_X	B_Y	B_Z	V_{sw}	N_p	P_d	E_M	Samples
2000	2	2	2	2	8	8	7	8	9
2001	11	11	11	11	17	17	16	16	17
2002	4	4	4	4	11	11	11	11	12
2003	12	12	12	12	19	19	15	16	19
2004	4	4	4	4	14	14	12	13	14
2005	6	6	6	6	15	15	15	15	16
2006	2	2	2	2	11	11	11	11	11
2011	6	6	6	6	23	23	23	24	24
2012	5	5	5	5	23	23	23	23	23
2013	8	8	8	8	24	24	24	25	25
2014	18	18	18	18	34	34	34	34	34
2015	7	7	7	7	21	21	21	22	22
2016	14	14	14	14	31	31	31	31	31
2017	26	26	26	26	36	36	36	36	36
2018	6	6	6	6	23	23	23	23	23

Most of the missing values have a very short duration. Figure 3.5 show this with distributions of the duration of consecutive missing values for the two main groups of parameters. The distributions are averaged over all parameters in a group with the average standard deviation given in the title of each plot.

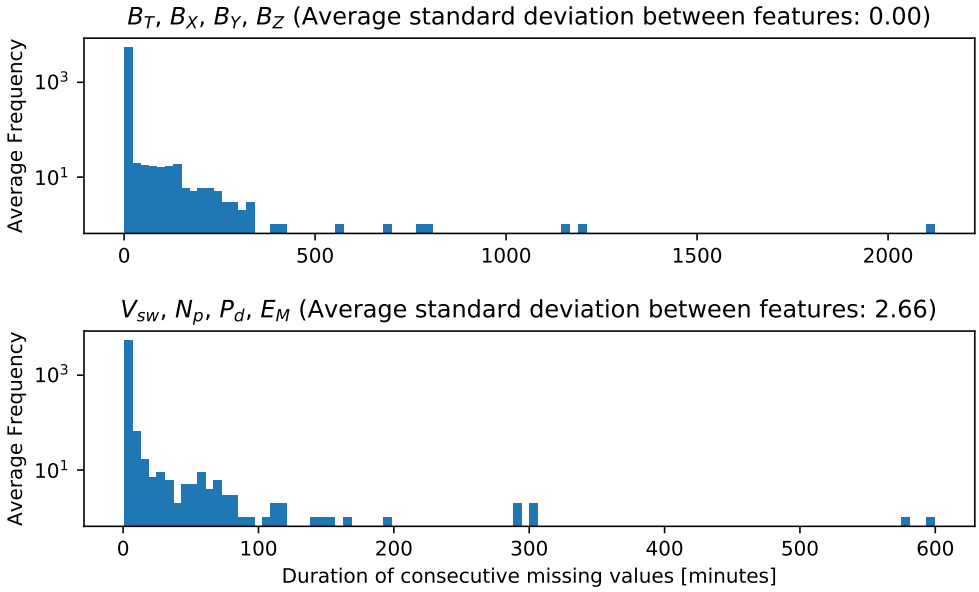


Figure 3.5: Distribution of the duration of consecutive missing values, grouped into IMF and plasma parameters. The average frequency is shown on a log scale.

To decrease the number of missing values, we interpolate over the missing values of each feature individually[1]. Note that, by handling features individually, interpolation should not make a significant difference in the relationship between parameters. Linear interpolation is chosen because it has a smaller chance of producing spurious peaks than other methods such as spline interpolation. We define an interpolation *limit* as the maximum number of consecutive missing features to interpolate over.

Table 3.3 shows the percentage of the entire data set that still consists of missing samples after interpolation. Results are shown for several time-shift sizes, as well as the number of samples removed by the time-shift procedure alone (i.e. without taking missing values into account). Interpolating with a limit of 10 minutes reduces the number of missing samples by more than 10% in all cases, but increasing the limit size further gives diminishing returns.

Adding too much interpolation may produce samples without enough variance (too linear) for a good representation of the true data generating process. To measure the approximate

[1]A simple test showed a small improvement in model performance (1% on the validation set) when interpolating missing values as shown here versus discarding all instances with missing values (not shown).

Table 3.3: Percentage of samples removed after applying time-shifts and interpolation, and removing remaining missing values. The *raw* column shows the percentage of samples removed by the time-shift procedure alone (that is, without interpolation or removing missing values).

Time-shift	Raw	Interpolation limit (minutes)								
		0	1	2	5	10	15	20	25	30
0	0.00	17	12	9	7	6	5	5	5	5
30	0.60	28	18	14	10	8	7	7	6	6
60	1.20	29	20	15	11	9	8	8	8	7
90	1.80	30	21	16	12	10	10	9	9	8
180	3.60	32	23	19	15	13	12	11	11	11
270	5.40	33	24	21	16	15	14	13	13	13
360	7.20	35	26	22	18	16	16	15	15	14
540	10.80	37	29	25	22	20	19	19	18	18
720	14.40	40	32	28	25	23	23	22	22	21

effect of interpolation on the variance of the data set, we perform the following analysis on SYM-H since this parameter has no missing values: First we artificially add missing values to SYM-H, corresponding to the missing features of the solar wind parameters. Then we process the missing values in the same way we would the solar wind parameters, that is we use linear interpolation with a predefined limit to replace a portion of the missing values and then remove any remaining missing samples. Finally we take the standard deviation of SYM-H and repeat the process for several interpolation limits. Results of this analysis (Table 3.4) show that interpolation have little effect on the variance of SYM-H.

Based on the analysis above, we choose an interpolation limit of 10 minutes. In most cases, it saves more than 50% of the samples that would have been removed if no interpolation had been used (see Table 3.3). Larger limits save slightly more samples, but at the risk of introducing too much linearity in the data.

In summary, missing values are handled as follows: If the total duration of consecutive missing values is less than 10 minutes, these values are filled via linear interpolation. Interpolation is done for each feature, and any time-shifted version of a feature, individually. The remaining missing values are removed by discarding any sample (both the input and target values) that has at least one missing feature. As mentioned in Section 3.4, if a

Table 3.4: Standard deviation of SYM-H values with induced missing values for several time-shifts and interpolation limits. The *raw* column show the case where no artificial missing values are added.

Phase	Time-shift	Raw	Interpolation limit (minutes)						
			0	5	10	15	20	25	30
Onset	None	20.70	20.91	20.62	20.60	20.59	20.59	20.59	20.58
	30	20.74	21.18	20.69	20.64	20.63	20.62	20.62	20.61
	90	20.79	21.30	20.79	20.72	20.70	20.68	20.67	20.66
	270	21.20	21.70	21.29	21.23	21.21	21.19	21.18	21.17
Main	None	51.13	46.49	46.23	46.23	46.27	46.31	46.36	46.40
	30	51.13	46.47	46.16	46.08	46.09	46.13	46.18	46.24
	90	51.13	46.04	45.40	45.32	45.33	45.36	45.40	45.44
	270	51.13	45.07	44.61	44.66	44.71	44.76	44.82	44.86
Recovery	None	41.49	41.27	40.42	40.34	40.29	40.27	40.25	40.24
	30	41.49	42.27	40.72	40.51	40.43	40.39	40.38	40.36
	90	41.49	42.20	40.75	40.60	40.52	40.48	40.46	40.43
	270	41.49	42.00	40.48	40.28	40.19	40.13	40.10	40.07
All	None	47.78	46.73	46.05	45.99	45.97	45.96	45.96	45.95
	30	47.79	47.39	46.22	46.07	46.01	46.00	45.99	45.99
	90	47.80	47.28	46.11	45.98	45.93	45.91	45.90	45.89
	270	47.79	46.89	45.78	45.66	45.62	45.61	45.62	45.63

time-shifted sample has one or more missing features remaining at either of its endpoints, the sample is removed. Our method for dealing with missing values has the following implication on the various data sets: Without any time-shifts, 17% of the samples contain at least one missing value. If we interpolate with a limit of 10 minutes it reduces to 6%. In other words, 11% of the data contains interpolated values. If we add a 270-minute time-shift, 33% of the data contains missing values and 18% of the data would contain interpolated values (when interpolating with the same 10-minute limit).

3.7 Data probing

In this section we extract statistical features from the data set to verify that the data is sound and to gain insight about the nature of the problem. The entire data set (the training, validation and evaluation set) is analysed here and therefore, we are careful not

to use these results for model development. Interpolation is applied and missing values are removed, as described in Section 3.6.

3.7.1 Parameter distribution

Figure 3.6 presents the distribution of solar wind parameters and the SYM-H index. Two distributions are shown for each parameter: The first captures the data set as described in this chapter (in other words, only intense storms) and the second is drawn from the remaining samples after intense storms are removed. Note that the latter still contains several storms that were not severe enough to be captured by our event selection algorithm, which only captured storms with a minimum SYM-H less than -100 nT. Alongside each distribution in the figure, miniature box plots are also illustrated.

From this graph it is clear that B_Z, P_d and E_M are approximately normally distributed for both intense storms and when excluding intense storms, albeit with extreme outliers. The distribution of B_Z is more negatively skewed during intense storms than during quieter times. (Outliers make it less apparent in the figure.) SYM-H is generally normally distributed, but skews to the negative direction during intense storms. This figure also shows that during intense storms, B_T and V_{sw} increase and the distributions of all the parameters are more spread out.

3.7.2 Parameter correlation

Figure 3.7 shows the correlation between solar wind parameters and SYM-H for the entire data set after removing missing values, but before standardisation. None of the solar wind parameters shows a strong correlation with the SYM-H index, which confirms that the relationship between solar wind and SYM-H is non-linear. The clear linear relationship of pairs (P_d, N_p) and (E_M, B_Z), respectively, is due to the relationships $P_d \sim N_p$ and $E_M \sim -B_Z$, described in Section 2.2.2. The absolute correlation between the rest of the parameter combinations fall below 0.5.

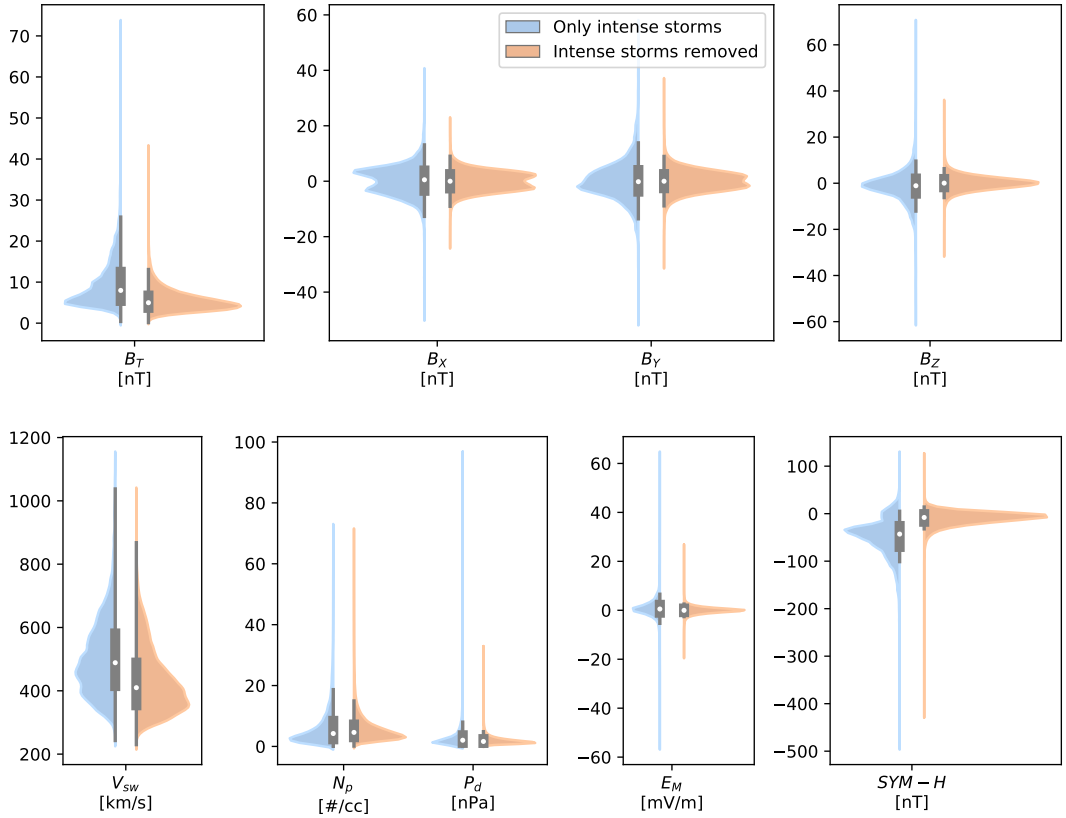


Figure 3.6: Distribution of the solar wind parameters and SYM-H index before standardisation. Units are shown below each parameter and small box plots are drawn on top of the distributions to show the minimum, maximum, median, first and third quartiles of the data.

3.8 Standardisation

It is common practice in machine learning to standardise the data set before model training or inference. In this work, standardisation is done for input data and target values separately, by removing the mean and scaling to unit variance. This technique is also known as z-normalisation. A feature x is standardised as follows:

$$z = \frac{x - \mu}{\sigma}, \tag{3.1}$$

where μ is the mean and σ the standard deviation of the training set for each feature individually. Standardisation scalers are only calculated on the training set and then applied to all partitions (training, validation and evaluation).

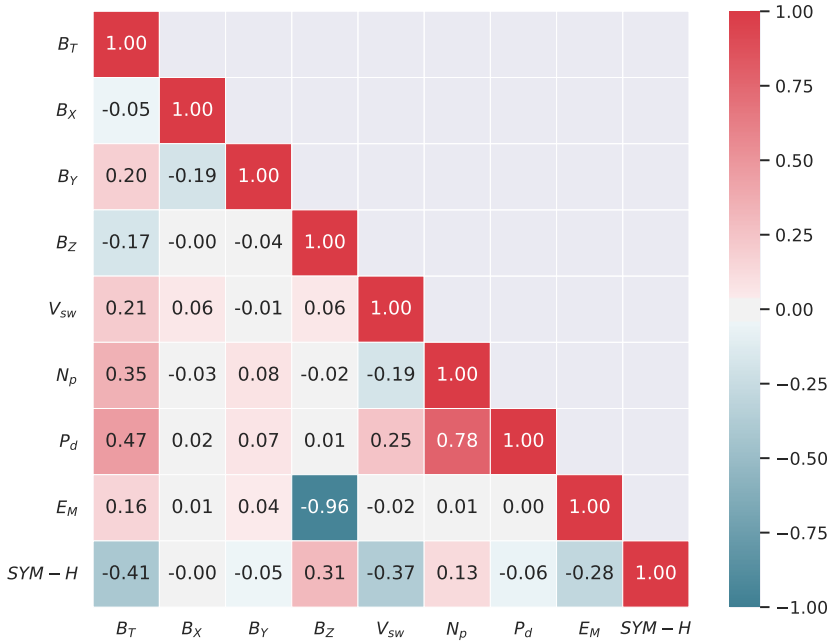


Figure 3.7: Correlation between the solar wind parameters and the SYM-H index of the entire data set before standardisation.

3.9 Preprocessing

The following preprocessing steps are taken in addition to the data preparation included in the HRO data set (Section 3.2):

1. Time-shifts are added, if required (Section 3.4).
2. Missing values, having a duration of less than 10 minutes, are filled using linear interpolation. Any remaining missing samples are removed (Section 3.6).
3. Every partition in the data set is standardised by removing the mean and scaling to unit variance. See Section 3.8 for the exact procedure.

3.10 Conclusion

In this chapter we described the development of the solar wind/SYM-H data set to be used in Chapters 4 and 6. An overview of the data source, time period and input and output parameters was given first. We then described the event selection algorithm used to isolate geomagnetic storms and specified the data partitioning method. Next, we discussed our approach to provide temporal and phase information to the model. The handling of missing values was a leading consideration during data set construction. We demonstrated that interpolation saves a lot of data and we expect the variance to be maintained based on analysis of the target parameter. Lastly, we provided several statistics, such as storm duration and parameter distributions, to verify that the data is sound and to gain insight about the nature of the problem.

Chapter 4

MLP for SYM-H

table

In this chapter, we train multilayer perceptrons (MLPs) on the solar wind/SYM-H data set to develop a baseline for comparing model performance in Chapter 6. We describe the experimental setup, discuss the optimisation results and show that temporal and phase information improve model performance on this task.

4.1 Introduction

In this chapter we develop four multilayer perceptron (MLP) baseline models for the solar wind/SYM-H data set described in Chapter 3. The first baseline uses the solar wind input parameters as is, the second adds temporal information in the form of time-shifts, and the third adds a calculated storm phase indicator as an additional parameter. The final baseline combines the features of the last two models. All baselines are optimised individually, as described in Section 4.3.

The first model under consideration (hereafter labelled B_1) is an MLP trained with 8 solar wind parameters as input and SYM-H as output. The MLP is our architecture of

choice because it is closely related to the proposed pairwise network (Chapter 5). It is also the simplest form of artificial neural network (ANN) [33] and therefore, relatively easy to optimise. However, an MLP cannot capture temporal information, while the magnetospheric response to solar wind energy input is non-linear in time and there are non-zero decay times determining the response of the ring current (and therefore the SYM-H index) [24], [78].

A time-shifted version of the input parameters is added in model B_2 as a statistical attempt at enabling the network to capture temporal information. For each input parameter X_t, measured at time t, another input X_{t-m} is added, doubling the number of inputs of model B_2 to 16. The value for m is chosen by a parametric search of shifts. Applying this time-shift results in a marked increase in performance. Note that a constant shift size is a limitation of the model because it incorrectly assumes that the time scale of energy deposition and magnetospheric processes is identical for all geomagnetic storms.

A further improvement is made by adding categorical input parameters to indicate phase since different physical processes are at play during the three storm phases. The relevant phase indicator is set to 1 when the appropriate phase (onset, main or recovery) is in progress, and set to 0 at other times. Therefore, model B_3 has $8+3 = 11$ input parameters. Model B_4 includes both phase and time-shifted inputs, resulting in $2(8 + 3) = 22$ inputs.

4.2 Experimental setup

In this section we motivate the decisions made and principles followed to develop a baseline through hyperparameter optimisation.

Every MLP constructed here has 8 or more input nodes, 1 or 2 hidden layers and a single output node. The rectified linear unit (ReLU) is used as activation function since it has been shown to have good performance in ANNs [37]. Networks are initialised with the Kaiming scheme [42] (using a uniform distribution), which is the default choice for ReLU-activated networks.

All networks are trained with the Adam [41] optimiser and mean squared error (MSE) loss function. Adam is chosen for its adaptive estimates of lower-order moments as opposed to standard stochastic gradient descent (SGD) [33], and MSE for its simplicity and popularity in regression problems. We report on the MSE and Pearson correlation coefficient (PCC) of the evaluation set averaged across several training seeds, and typically also report the standard error (SE) of both metrics.

The following hyperparameters are optimised: learning rate, batch size, weight decay (L2 norm penalty), batch normalisation and network size (number of hidden units). We chose these hyperparameters based on preliminary tests which showed that they could have a significant effect on model performance. Hyperparameter tuning is performed using grid searches that are specific to each experiment, and reported on per experiment. Without batch normalisation we add a bias term to the first layer of every network, but with batch normalisation no biases are added. When using batch normalisation we use an affine transform at each layer, which already includes extra learnable parameters with a bias, making additional bias terms redundant.

Early stopping is implemented by selecting the training epoch where the model achieved its lowest MSE on the validation set. To ensure that training is not terminated prematurely, we dynamically add epochs as required: Networks are trained for a minimum of 50 epochs, but if the best-performing network is found in the final $e = 30\%$ of epochs, the network is trained further for e more epochs. These steps are repeated until the network converges.

Optimisation is done on a single GPU (NVIDIA RTX 2080 Ti) to avoid floating point differences between multiple hardware devices and to ensure reproducibility.

4.3 Optimisation

The specific procedure followed to develop each baseline is described in this section. First of all, we construct B_1 by searching for the best set of hyperparameters (Section 4.3.1). Based on these results, we adapt the range of hyperparameters to search over and find

baselines B_2 (Section 4.3.2), B_3 and B_4 (Section 4.3.3). For B_2, we do an additional grid search over time-shift lengths. The same time-shift is also used in B_4.

4.3.1 MLP

In this section we construct baseline B_1, an MLP with 8 solar wind inputs, one or more hidden layers and a single output node (SYM-H). Our first step is to do a grid search over network layouts {2x10[1], 2x280}, batch sizes {128, 1 024, 16 384}, batch normalisation {with, without}, weight decay values {0.1, 0.01, 0.001, 0}, learning rates {10^{-1}, 10^{-2}, 10^{-3}, 10^{-4}, 10^{-5}} and 3 initialisation seeds. For this search, as well as for all of the hyperparameter tuning steps discussed in this chapter, the range of learning rates that we list is only a starting point. More learning rates are added whenever the best network lie on the edge of the grid or if there are large jumps in performance within the grid. In this manner we ensure that the learning rate is always optimised. We only show the result for the single best-performing learning rate per experiment. The results are presented in Table 4.1 and discussed below in separate sections, with additional optimisation steps where necessary. The final performance of baseline B_1 on the evaluation set is discussed in Section 4.4.

Batch size

The range of batch sizes are chosen based on the fact that the longest geomagnetic storm in our training set has 14 487 samples (see Section 3.3) and that it is common practice for the batch size to be a multiple of 2 for computational efficiency. The results in Table 4.1 show that the larger 16 384 batch size performs worse than the smaller two, and that a batch size of 128 and 1 024 have approximately similar performance when considering the best batch normalisation and weight decay settings for each batch size individually. Without a significant preference, we use a batch size of 1 024 from here onwards.

[1] An MLP with a layout DxH implies that it has D hidden layers of H nodes each. This convention is followed throughout this chapter.

Table 4.1: The validation MSE of MLPs trained with several hyperparameters on the solar wind/SYM-H data set. Results are averaged across 3 seeds in this preliminary run, and the learning rate is optimised in each case. No time-shifted or phase inputs are used. The best results are printed in bold.

Network Layout	Batch Size	Batch Normalisation							
		Without				With			
		Weight Decay							
		0	0.001	0.01	0.1	0	0.001	0.01	0.1
2x10	128	0.9341	0.9247	**0.9005**	0.9161	0.9681	0.9288	0.9230	**0.9187**
	1 024	0.9333	0.9341	0.9131	**0.9119**	0.9480	0.9377	**0.9277**	0.9279
	16 384	0.9330	0.9328	0.9290	**0.9191**	0.9628	**0.9350**	0.9401	0.9375
2x280	128	0.9347	0.9344	0.9313	**0.9131**	1.0272	0.9114	0.9330	**0.9075**
	1 024	0.9337	0.9334	0.9271	**0.9122**	1.0147	0.9289	**0.9173**	0.9316
	16 384	0.9347	0.9342	0.9316	**0.9231**	0.9375	0.9571	**0.9328**	0.9532

Batch normalisation

Batch normalisation does not improve the performance of the smaller 2x10 network, but the results in Table 4.1 indicate that it helps for the larger 2x280 network. Therefore, we decided to only use batch normalisation when training relatively large networks.

Weight decay

The results show that the right amount of weight decay leads to an improvement in performance over training without weight decay. For subsequent hyperparameter optimisation, we decided to treat weight decay the same as we do the learning rate: search over an initial range of values and then expand the grid if the best result is found at the edge of the grid or if there are large differences in performance within the grid.

Network layout

To investigate the effect of architecture size, we do an additional grid search over network layouts {1x10, 1x100, 2x10, 2x100, 2x280}, learning rates $\{10^{-1}, 10^{-2}, 10^{-3}, 10^{-4}, 10^{-5}\}$

and weight decay values {1, 0.5, 0.15, 0.1, 0.05, 0.01, 0.001, 0} across 3 seeds. These networks are trained with a batch size of 1 024 and we only tested batch normalisation when training the 2x100 and 2x280 models. The optimisation results of this grid are provided in Table 4.2, which shows that all the network layouts have very similar performance if the other hyperparameters are optimised, but that the smaller networks tend to perform better.

Table 4.2: Validation MSE of an MLP (B_1) on the solar wind/SYM-H data set for several network layouts and weight decay values, trained with a batch size of 1 024 across 3 seeds. In every case, we ensured that the learning rate is optimised. The best results are printed in bold.

Weight	Without Batch Normalisation (BN)					With BN	
Decay	1x10	1x100	2x10	2x100	2x280	2x100	2x280
0	0.9343	0.9321	0.9333	1.0352	0.9337	1.0352	1.0147
0.001	0.9331	0.9376	0.9341	0.9303	0.9334	0.9303	0.9289
0.01	0.9044	0.9121	0.9131	0.9304	0.9271	0.9304	0.9173
0.05	**0.8978**	**0.8987**	**0.9112**	0.9305	0.9177	0.9305	0.9374
0.1	0.8984	0.9054	0.9119	0.9374	**0.9122**	0.9374	0.9316
0.15	0.9085	0.9133	0.9311	**0.9175**	0.9235	**0.9175**	0.9192
0.5	0.9288	0.9227	0.9964	0.9214	0.9241	0.9214	**0.9108**
1	0.9960	0.9467	0.9692	0.9387	0.9430	0.9387	0.9366

4.3.2 MLP with time-shifts

Here we optimise the hyperparameters for baseline B_2 in a similar fashion as B_1, without showing all the intermediary results. First we find the optimal time-shift size (in minutes) with a grid search over shifts {5, 30, 90, 180, 260, 265, 268, 269, 270, 271, 272, 275, 280, 360, 540} and learning rates {0.5, 0.1, 0.01, 0.001} across 3 seeds. Based on our observations in the previous section, we select the 1x10 network layout, a batch size of 1 024, a weight decay value of 0.01 and no batch normalisation for this grid search. The best validation MSE of several time-shifts are provided in Table 4.3, which shows that a time-shift between 260 and 280 minutes results in low validation MSE. Without significant preference, we choose a 270 minute time-shift from here onward.

Table 4.3: Validation MSE of an MLP with time-shifted inputs (B_2) on the solar wind/SYM-H data set for several time-shift sizes. The 1x10 network is trained with a batch size of 1 024, weight decay value of 0.01 and no batch normalisation. Results are averaged across 3 seeds and the best learning rates are selected. Time-shift values not shown follow the same trend.

Time-shift (min)	30	90	260	265	270	275	280	540
Validation MSE	0.8393	0.6999	0.6050	0.6098	0.6078	0.6074	0.6068	0.7424

As a final step, we do a refined search over network layouts {1x10, 1x100, 2x280}, learning rates {0.5, 0.1, 0.01, 0.001}, weight decay values {0.1, 0.05, 0.01, 0.001, 0} and across 3 seeds using the 270 minute shift. Batch normalisation is employed for the 2x280 network.

4.3.3 MLP with storm phases

To establish baselines B_3 and B_4, we include phase inputs and do a grid search over network layouts {1x10, 1x100, 2x280}, learning rates {0.5, 0.1, 0.01, 0.001}, weight decay values {0.5, 0.15, 0.1, 0.05, 0.01, 0.001, 0}, with and without time-shifted inputs and across 3 seeds. A shift size of 270 minutes is used whenever time-shifted inputs are added and batch normalisation is tested with the 2x280 network.

The resulting performance on the validation set is shown in Table 4.4. For both B_3 and B_4, we select the model with the lowest validation MSE. That is the 2x280 layout, with batch normalisation for B_3 and without it for B_4. The performance of these networks on the evaluation set is discussed in the next section.

4.4 Baseline Performance

The average MSE and PCC of the baseline MLP models on every partition of the solar wind/SYM-H data set across 3 seeds, are presented in Table 4.5. The simplest model (baseline B_1) achieves an evaluation MSE of 0.42. By adding time-shifted (B_2) and phase

Table 4.4: Validation MSE of an MLP with phase inputs on the solar wind/SYM-H data set for several network layouts and weight decay values, with and without time-shifts across 3 seeds. In every case, we ensured that the learning rate is optimised. The best results are printed in bold.

Time-shift	Weight Decay	Without BN			With BN
		1x10	1x100	2x280	2x280
Without	0	0.5144	0.5204	0.5217	0.5693
	0.001	0.5034	0.5101	0.5207	**0.5011**
	0.01	**0.5025**	**0.5031**	**0.5051**	0.5040
	0.05	0.5282	0.5127	0.5120	0.5162
	0.1	0.5478	0.5301	0.5106	0.5100
	0.15	0.5659	0.5294	0.5204	0.5059
	0.5	0.7318	0.5502	0.5289	0.5250
With	0	0.4412	0.4378	0.4733	0.5086
	0.001	**0.4255**	0.4378	0.4473	0.4358
	0.01	0.4266	**0.4211**	**0.4120**	0.4600
	0.05	0.4299	0.4331	0.4275	0.4475
	0.1	0.4580	0.4414	0.4220	**0.4241**
	0.15	0.4692	0.4421	0.4357	0.4423
	0.5	0.5237	0.4579	0.4347	0.4715

inputs (B_3), the average MSE decreases to 0.33 and 0.26 respectively. When both time-shifted and phase inputs are included (B_4), the best MSE is 0.25. It is peculiar that B_3 and B_4 have approximately the same evaluation MSE. (To confirm, notice that the confidence intervals (SE) overlap: $0.26 - 0.0172 = 0.2428 < 0.2542 = 0.25 + 0.0042$.) This indicates that time-shifted inputs do not provide a significant benefit when phase inputs are already included in the model.

The prediction made by models B_1 and B_4 are compared to the observed SYM-H, for a single storm, in Figure 4.1. Notice the improvement gained when adding phase and temporal information: B_1 fails to capture the peak of the storm, but B_4 is able to model the onset, main and the start of the recovery phase well.

For a reference frame of the size of each network's error we measure ΔnT, the rescaled root mean squared error (RMSE), on the evaluation set and provide the results in Table 4.5. We determine ΔnT by passing the evaluation set through the trained network, rescaling

Table 4.5: Performance of the baseline MLPs on every partition of the SYM-H data set. Results are measured across 3 seeds and the SE of the evaluation set is provided in brackets. We report on ΔnT which interprets the MSE value in the original unit of SYM-H (nT).

Model	MSE			PCC			ΔnT
	Train	Valid	Eval (SE)	Train	Valid	Eval (SE)	Eval
B_1	0.61	0.90	0.42 (0.0115)	0.63	0.56	0.65 (0.0102)	29.82
B_2	0.35	0.59	0.33 (0.0161)	0.81	0.76	0.77 (0.0097)	25.75
B_3	0.34	0.50	0.26 (0.0172)	0.82	0.79	0.82 (0.0033)	23.51
B_4	0.19	0.41	0.25 (0.0042)	0.90	0.84	0.85 (0.0018)	22.52

the output to revert the scaling done during data standardisation (Section 3.8) and then calculating the RMSE between the prediction and the unscaled target value. To revert standardisation, we use the calculated values $\mu = -49.93$, $\sigma = 45.78$ for baselines B_1 and B_3 and $\mu = -52.14$, $\sigma = 45.05$ for baselines B_2 and B_4 for the target parameter. The way time-shifts are added (Section 3.4) is the reason for the difference in scaler values. The ΔnT values therefore interprets the MSE values in terms of the original unit of SYM-H.

There is a considerable difference between the validation and evaluation MSE of each baseline, with evaluation results better than validation results. This could be due to the specific distribution of storm events between the partitions.

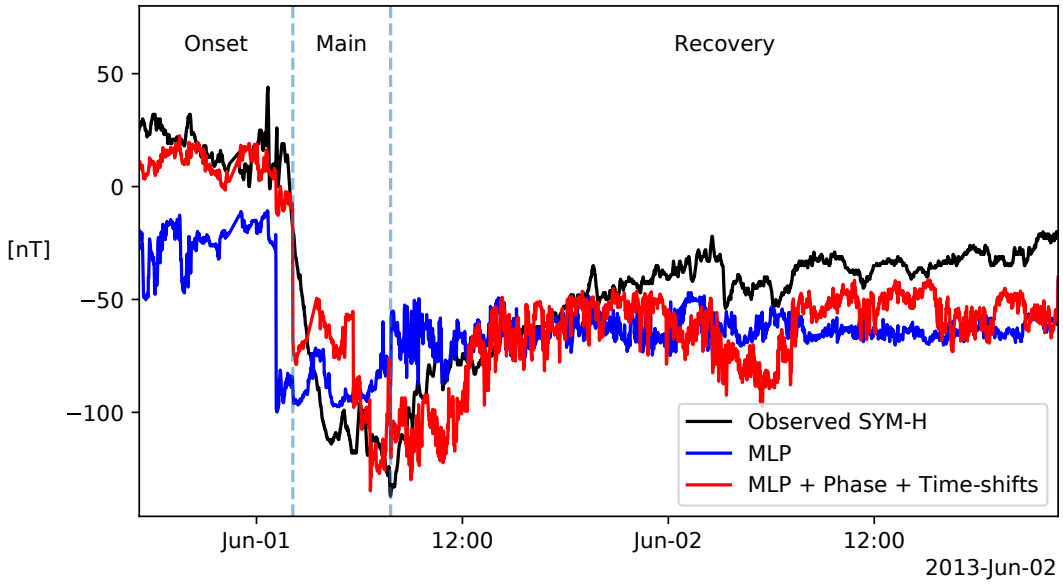


Figure 4.1: An example of observed SYM-H and the prediction made by baselines B_1 (MLP) and B_4 (MLP with phase and time-shifted inputs). This particular geomagnetic storm is taken from the evaluation set and occurred on 1-2 June 2013. The respective phases of the storm is labelled at the top of the panel.

4.5 Conclusion

In this chapter we showed that the MLP is a viable option for predicting the SYM-H index from solar wind parameters. We developed 4 baseline models by optimising the hyperparameters of several MLPs on the solar wind/SYM-H data set (introduced in Chapter 3). The following observations were made:

- The best model was of size 2x280 and achieved a RMSE of 22.52 nT and a correlation of 0.85 on the evaluation set.
- Adding time-shifted and phase inputs improve the performance significantly, but when used together the performance is approximately the same as when only phase inputs are included.

- The optimal time-shift size for the solar wind/SYM-H data set lie between 260 and 280 minutes. We decided to use a 270 minute shift from here onward.
- Batch normalisation resulted in a slight improvement in the performance of some of the larger models, but not for any of the smaller models.
- This task is relatively insensitive to changes in batch size, but sizes 128 and 1 024 perform better than the larger 16 384 size. We decided to use a batch size of 1 024 hereafter.
- Weight decay made a notable improvement in the performance of the MLPs.
- There exists a considerable difference between the validation and evaluation performance of these models.

In Chapter 6 we will use the MLP baselines to compare model performance of the pairwise network on this task.

Chapter 5

Pairwise Network

In this chapter we introduce the pairwise network, describe its layout and develop its training and ranking procedure using synthetic data sets. The pairwise network is the main focus of this dissertation.

5.1 Introduction

A multilayer perceptron (MLP) mixes signals from all inputs as information flows through the hidden layers to the output. This enables the training procedure to utilise all the different combinations of input parameters to find an efficient solution. Analysis of these sets of combinations are prohibitively complex since from the first layer of hidden nodes, every node is directly or indirectly connected to every input parameter.

By strategically removing connections, an MLP can be transformed into an intrinsically interpretable *pairwise network*. As introduced in [22] and [23], the pairwise network groups all combinations of input parameters into pairs of two and assigns a sub-network to each pair, allowing weights connected to pairs of inputs to be isolated and analysed

separately. Sub-network outputs are combined to produce the one-dimensional output of the model. We limit our investigation to regression modelling, but note that the pairwise network can be extended to a multi-class classifier by adapting the output layer. Similar models have been developed in recent research and are discussed in Section 2.3.2.

5.2 Pairwise network

This section describes the architecture, training and ranking procedure of the pairwise network. Some of the technical decisions described here are based on additional analysis that we present in more detail in Section 5.4.

5.2.1 Architecture

The pairwise network is constructed according to the following procedure (notation follows the conventions of Yang et al. [62]). Given an MLP with M input parameters, D hidden layers with H nodes per hidden layer, and a single output node:

1. Find all possible distinct pairs of inputs $\{x_i, x_j\}$ $(i \neq j)$ in the list $\{x_1, x_2, \ldots, x_M\}$ to form a set S containing N such pairs. Note that $\{x_i, x_j\} \equiv \{x_j, x_i\}$.

2. Divide the set of hidden nodes into N distinct groups, each group having D hidden layers with H/N nodes per hidden layer.

3. For each group, create a sub-network by fully connecting each pair of inputs to the first hidden layer of its corresponding group of hidden layers. Fully connect each set of hidden layers as well.

4. For each sub-network, fully connect the nodes of the final hidden layer to a single node (the summary node).

5. Connect the summary nodes of all sub-networks to a single output node.

The output of the pairwise network has the following form,

$$y = \sum_{(i,j)\in S} f_{ij}(x_i, x_j), \tag{5.1}$$

where $f_{ij}(x_i, x_j)$ is modelled by a sub-network.

The layout of an example pairwise network with three input parameters is shown in Figure 5.1. The *input weights* in the first layer of the pairwise network are set to have a value of 1 and frozen so as not to update during training. Biases are added to the second layer of the pairwise network and rectified linear unit (ReLU) activation functions [34] are placed after every layer except the first (input), second-last (summary) and final layer (output).

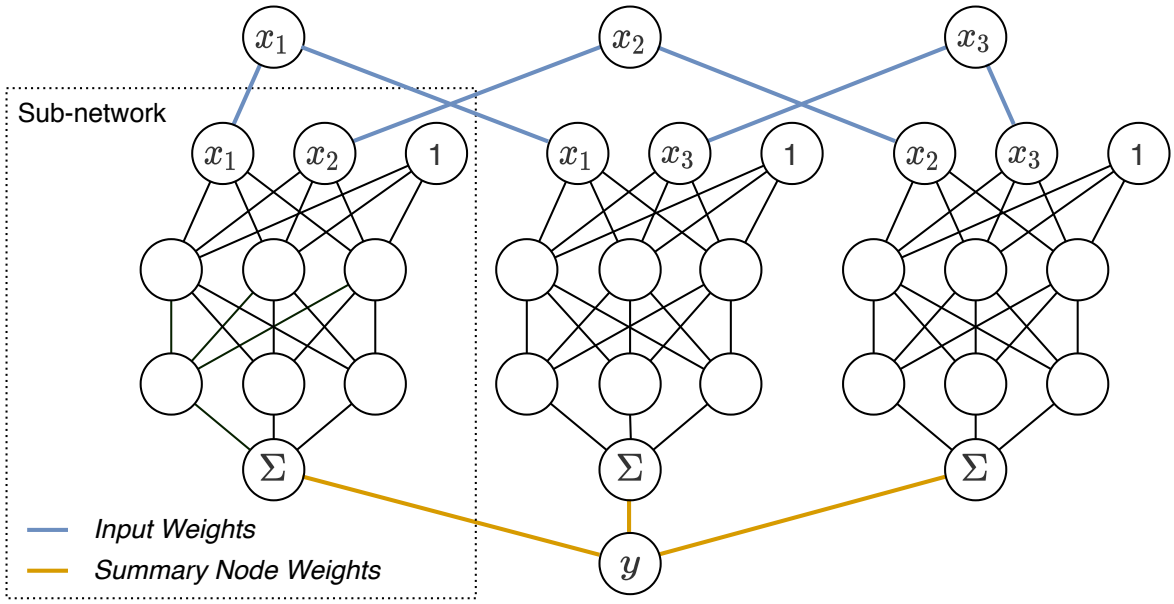


Figure 5.1: Pairwise network architecture for a simple case of 3 input features (x_i, top) and a single output (y, bottom). Input features are grouped into pairs of two and each pair is connected to a sub-network with 2 hidden layers and 3 nodes per hidden layer. Biases are added to the second layer of the pairwise network (i.e. the first layer of each sub-network). Summary nodes are indicated with Σ to show that they do not have an activation function.

5.2.2 Ranking

A desirable property of the pairwise network, with regards to interpretability, is that there are no shared parameters between sub-networks. This ensures that each sub-network is independent of the others, allowing separate analysis. In this section, we introduce a method for extracting feature rankings from the pairwise network.

For any given sample, the output of a pairwise network is the weighted sum of the summary node activation values (Equation 5.1), providing a direct indication of each pair's contribution to the model output. To quantify the importance of each pair relative to the others, we define the *weighted summary node activation ratio* of a variable pair (x_i, x_j) for a given sample as

$$\psi_{sample}(i,j) = \frac{|f_{ij}|}{\sum_{(i,j)\in S} |f_{ij}|}. \tag{5.2}$$

With S again all variable pairs. To estimate ψ, the parameter importance for a set of samples, ψ_{sample} is averaged over the set. The sum of ψ over all pairs is equal to one, hence ψ can be viewed as quantifying how importance is distributed across the different pairs, as used by Beukes et al. [23]. In this work we refer to it as the ψ-distribution. Larger ψ means greater importance and so input parameters can be ranked according to decreasing ψ.

This metric is similar to the Importance Ratio (IR) used by Yang et al. [62], with the difference being that they include sub-networks with only one input parameter. (Our ranking procedure was developed in parallel with theirs.)

5.2.3 Sub-network initialisation

We propose initialising the pairwise network to have identical weight values across sub-networks, with the idea that it will ensure that the network is not biased towards one of the pairs before training, and so improve the network's ranking capability.

In our implementation we first initialise the entire pairwise network with the Kaiming

scheme [42] (using a normal distribution) and then duplicate the weights of one sub-network to the others, assuming they have the same shape. Hereafter, we will refer to this scheme as *duplicated sub-network* initialisation and we will refer to *random* initialisation when initialising without duplicating sub-network weights.

In Section 5.4.2 we compare pairwise networks initialised with duplicated sub-networks to randomly initialised networks. The results show that both schemes produce networks with approximately the same performance and that, with regards to ranking, there are no significant and consistent benefit of using one over the other. Given a lack of empirical results to choose one over the other, we proceed to use the more principled duplicated sub-network technique as part of the standard pairwise network training procedure.

5.2.4 Pruning

In addition to the standard pairwise network, we investigate the iterative pruning of model weights, similar to the method described in [71]. In Section 5.4.3 we show that pruning results in a marked improvement in the ranking capability of pairwise networks by forcing models to be more sparse. We also show that the performance of pruned pairwise networks fall within acceptable margins of networks trained without pruning.

Our convention with regards to sparsity is to refer to the number of parameters removed from the network, for example a sparsity level of $s\%$ implies that $s\%$ of weights have been pruned. Pruning is done by setting weight values to 0 and freezing them so that they do not get updated in subsequent training steps.

The iterative pruning procedure we implement is as follows:

1. Initialise the pairwise network.
2. Train the network to completion[1].
3. Prune $p\%$ of the smallest unpruned weights according to their magnitude.

[1]The point where the network reaches its lowest validation mean squared error (MSE).

4. Fine-tune the remaining weights by training the network further.
5. Iteratively prune and fine-tune by repeating steps 3 and 4 until:
   (a) the network reaches the desired level of sparsity, or
   (b) an iteration's lowest validation MSE rises above a predetermined threshold k from the best value of the preceding iterations, in which case the previous iteration is selected.

Threshold k sets a trade-off between sparsity and performance. Sparser networks are favoured above better performing but larger networks, provided they perform within the margin required by the application. The percentage of remaining of weights to prune per iteration ($p\%$) is determined empirically. In Section 5.4.3 we experiment with several pruning amounts and investigate two fine-tuning strategies:

i. Training the network further for a fixed number of epochs.
ii. Training the network to completion.

Based on the results we choose a pruning amount of $p = 10$ and the first fine-tuning strategy (i) for training pairwise networks in Chapter 6.

5.2.5 Additional procedures

It can happen that the training procedure forces the fan-out weights of a particular feature in the input layer of a sub-network to zero, effectively pruning this feature from the sub-network. We rename such sub-networks to have the same name as the remaining input. If both features are pruned, such a sub-network becomes a bias. We found that for all the pairwise networks trained with pruning and for every data set in this work, these sub-networks have an output of exactly zero for all the samples in a particular set, and therefore do not contribute to the ψ-distributions.

Calculating the correlation between the input features of a sub-network and its output may reveal cases where sub-networks effectively only model one of the input pairs. The application of this technique is demonstrated in Section 5.4.5.

Lastly, seeds that produce a poor validation MSE are not used for ranking.

5.3 Synthetic data sets

To validate the pairwise network's ranking capability, we investigate two synthetic problems, with different degrees of complexity. This section describe these data sets.

5.3.1 Synthetic data set 1: $y = f(x)$

Here we generate two data sets with input features sampled from a standard normal distribution,

$$x_1, x_2, x_3, x_4 \sim \mathcal{N}(0, 1), \tag{5.3}$$

and target values,

$$y_1 = x_1 + x_2, \tag{5.4}$$

$$y_2 = x_1 x_2. \tag{5.5}$$

Each data set consists of 1 000 samples and are randomly partitioned as follows: 70% for training and 15% for both validation and evaluation. After partitioning, each data set is standardised as described in Section 3.8.

Our rationale behind this data set is to analyse the pairwise network's ranking capability on the simplest form of additive and multiplicative interaction between features. In both cases, the network is expected to rank pairs containing x_3 or x_4 lower than the (x_1, x_2) pair. For the multiplicative case (Equation 5.5) the network is expected to rely solely on the (x_1, x_2) pair and disregard the rest.

5.3.2 Synthetic data set 2: Akasofu's ϵ

Here we generate target values from a subsection of the SYM-H data set (Chapter 3), as described in Section 2.2.4, using the simplified ϵ^* parameter (Equation 2.2). We test two sets of input parameters:

$$a_1 = \{V_{sw}, B_T, \theta, r\}, \tag{5.6}$$

$$a_2 = \{V_{sw}, B_T, B_Y, B_Z, r\}, \tag{5.7}$$

where $r \sim \mathcal{N}(0, 1)$. In both cases the network is expected to identify r as an unwanted parameter, since it does not form part of the data generation. The same preprocessing steps followed for the SYM-H data set (Section 3.9) are applied here.

ϵ^* is determined by V_{sw}, B_T and θ. This interaction is reflected in Figure 5.2 which shows the progression of ϵ^* and the other parameters of this data set during a geomagnetic storm. Measured SYM-H is also presented for comparison. During the main phase of the storm ϵ^* reacts to greater change in the solar wind parameters, but as the recovery phase progresses and the solar wind returns to normal, ϵ^* decreases.

5.4 Analysis

In this section we validate the choices described in Section 5.2 through analysis of the pairwise network's training, pruning and ranking procedure on synthetic data sets.

5.4.1 Experimental setup

Here we describe the steps taken to perform hyperparameter optimisation for pairwise networks on synthetic data. From initial probing, we find a set of parameters that work well and then search for an optimal learning rate across several seeds. We do not search further over additional hyperparameters since the MSE of the networks trained on these data sets is very low. For each data set we optimise 2 models: a baseline pairwise

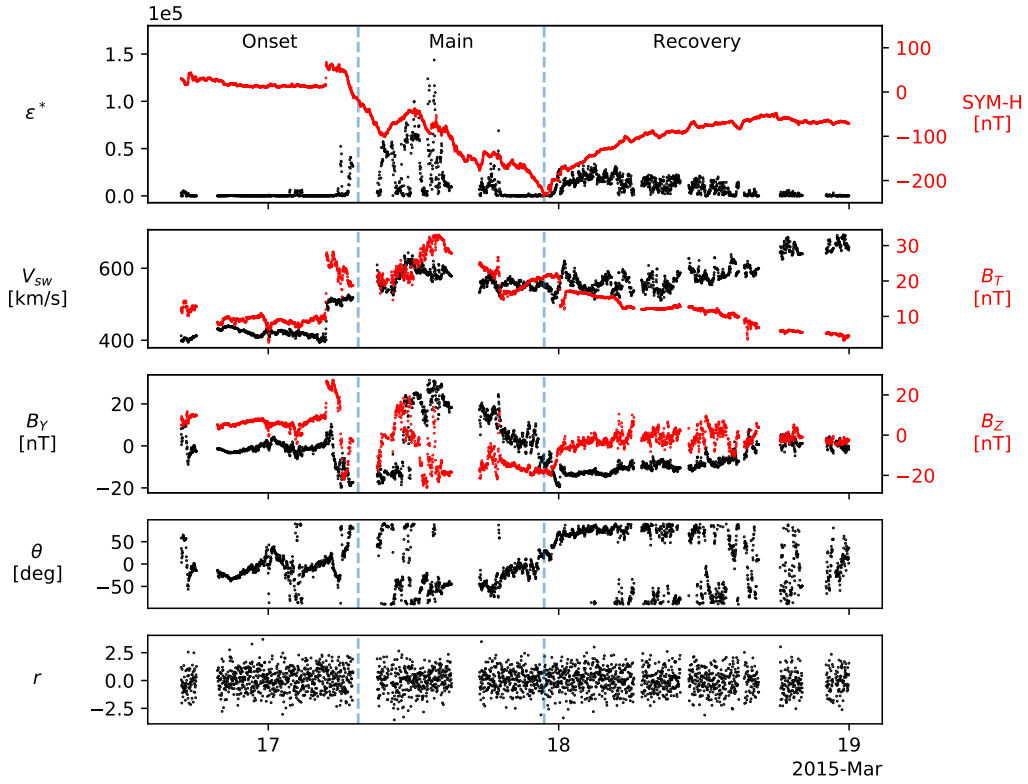


Figure 5.2: Synthetic data set 2 during the St. Patrick's day geomagnetic storm (17-21 March 2015). The ϵ^* parameter and the SYM-H index (provided for comparison) is shown in the top panel, solar wind parameters (V_{sw}, B_T, B_Y and B_Z) in the second and third panel, followed by θ and the random variable r. Storm phases are given in the top panel and transitions are indicated with dashed vertical lines.

network without pruning and one trained with pruning. We also test 2 types of fine-tuning strategies for the networks trained with pruning (as discussed in Section 5.2.4). For all experiments, Adam [41] is used as optimiser and MSE as loss function. Early stopping is applied by selecting models with the lowest MSE on the validation set. With regards to the architecture size, we choose a pairwise network with 2 hidden layers per sub-network and 10 nodes per hidden layer. Unless stated otherwise, we train all networks across 4 initialisation seeds and report on the average evaluation MSE, Pearson correlation coefficient (PCC) and the standard error (SE) of each metric. No explicit regularisers, such as batch normalisation or weight decay, are used. We choose mini-batch sizes 64 and 1 024 for data sets 1 and 2, respectively.

Baseline optimisation

Our baseline for comparing model performance is a pairwise network trained without pruning. To find this baseline, we do a grid search across 2 initialisation schemes, namely random (without duplicating sub-network weights) and duplicate sub-network initialisation, and learning rates {0.1, 0.05, 0.01} for synthetic data sets y_1 and y_2, and {0.005, 0.001, 0.0005, 0.0001} for data sets a_1 and a_2. In all cases, we ensure that the range of learning rates is complete by expanding the grid if results fall on the edge or refining the grid if there is a large difference between consecutive results in a range. We make sure that networks converge by dynamically adding epochs if the best network is found in the last 30% of epochs (as described in Section 4.2). We test both initialisation schemes to evaluate the effectiveness of duplicate sub-network initialisation with regards to performance and ranking.

Pruning optimisation

Two variants of the iterative pruning procedure (Section 5.2.4) are compared: Every fine-tuning step (Step 4) trains the network further (i) for a constant number of epochs or (ii) until completion. In both cases the best performing network of that iteration are selected by early stopping on the validation MSE.

The hyperparameters chosen for variant (ii) are selected according to both the performance and convergence time of the baseline pairwise network since training to completion at every iteration is computationally expensive. That is to say, we train the (ii)-networks across 4 seeds and use a single learning rate. For (i) we do a search across learning rates {0.1, 0.05, 0.01} for data sets y_1 and y_2, and {0.005, 0.001, 0.0005} for data sets a_1 and a_2. We choose the constant number of fine-tuning epochs as 100 for all synthetic data sets, based on initial tests.

For both pruning variants, the first set of networks are trained with a pruning amount (p%) of 10%. We then select the best learning rate and test several pruning amounts,

$p = \{10, 15, 20, 25, 30\}$. In all cases, a threshold of $k = 0.05$ is used, meaning that we select the sparsest network with a validation MSE within 0.05 of the best performing network, across all pruning iterations. All the networks trained with iterative pruning are initialised using the duplicated sub-network scheme.

5.4.2 Baseline system

In this section we present the performance and ranking results of the baseline pairwise networks trained on synthetic data sets. All the networks mentioned here are initialised with the duplicate sub-network scheme. Refer to Section 5.4.4 for a comparison between initialisation schemes.

Performance

The baseline pairwise network achieves near-perfect performance on all synthetic data sets with an MSE equal or less than 0.016 and a PCC greater than 0.98 on every evaluation set. Table 5.1 summarises these results. An example of how ϵ^* is modelled by the baseline pairwise network is shown in Figure 5.3.

Table 5.1: Performance of the baseline pairwise network on the synthetic data sets. Results are measured on the evaluation set with metrics averaged across 4 seeds and the standard error given in brackets.

Data Set	MSE (SE)	PCC (SE)
y_1	0 (0)	1 (0)
y_2	0.0070 (0.0015)	0.9961 (0.0008)
a_1	0.0153 (0.0013)	0.9828 (0.0015)
a_2	0.0083 (0.0003)	0.9902 (0.0005)

Ranking

Figure 5.4 presents the ψ-distribution of the optimised baseline pairwise networks for each of the 4 seeds on the evaluation set of every synthetic data set. For data sets y_1 and y_2, it

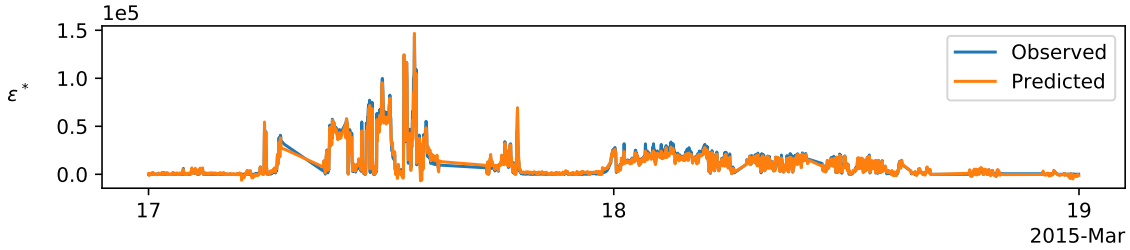


Figure 5.3: ϵ^* as predicted by baseline pairwise network, trained on data set a_1, against the true ϵ^* values of the St. Patrick's day storm. Results for data set a_2 are similar.

can be seen that the network relies more on the $\{x_1, x_2\}$ pair, but also assigns importance to pairs with a feature that do not form part of data generation. For the a_2 task, the model assigns more importance to pairs without the random variable (r), but for a_1 a high rank is given to the r-pairs. These results show that the ranking capability of the baseline model is fairly poor on the synthetic data sets.

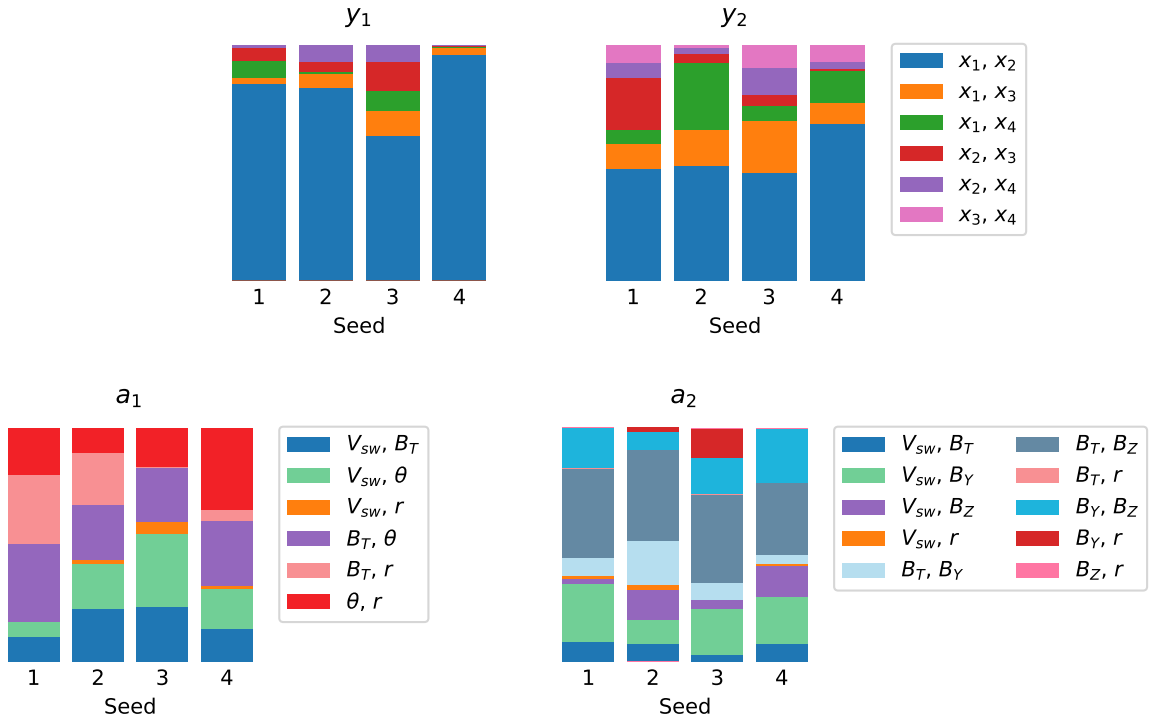


Figure 5.4: ψ-distributions of the baseline pairwise networks on the evaluation set of the respective synthetic data sets. Each bar represents a different initialisation seed and the length of each section (colour) is equal to the ψ of an input pair.

5.4.3 Pruning strategies

In this section we assess the effect of iterative pruning on the performance and ranking capability of pairwise networks on synthetic data sets. Unless stated otherwise, 10% of parameters are pruned each iteration and fine-tuning is done over 100 epochs every time. At the end of this section, we discuss the results of using other pruning amounts and compare fine-tuning strategies.

Performance

The performance of pairwise networks trained with pruning is given in Table 5.2. Except for y_1, pruning results in slightly worse predictive performance in comparison to the baseline models: A maximum increase of 0.0271 MSE and a maximum decrease of 0.0183 PCC on the evaluation set across all tasks.

Table 5.2: Performance of the pairwise networks on synthetic data sets when trained with iterative pruning. Networks are trained by pruning 10% of parameters and fine-tuning with a 100 epochs each iteration. Results are measured on the evaluation set with metrics averaged across 4 seeds and the SE given in brackets.

Data Set	MSE (SE)	PCC (SE)
y_1	0 (0)	1 (0)
y_2	0.0341 (0.0137)	0.9789 (0.0098)
a_1	0.0243 (0.0013)	0.9725 (0.0027)
a_2	0.0260 (0.0026)	0.9719 (0.0022)

To get an idea of the network's reaction to pruning, we plot the lowest validation MSE of each pruning iteration in Figure 5.5. Notice that for synthetic data sets y_1, y_2 and two seeds of a_1, the MSE remain approximately constant before increasing with a large amount in the last iteration. Results are shown for $p = 10$, but the same holds for other values of p in the range 10 to 30.

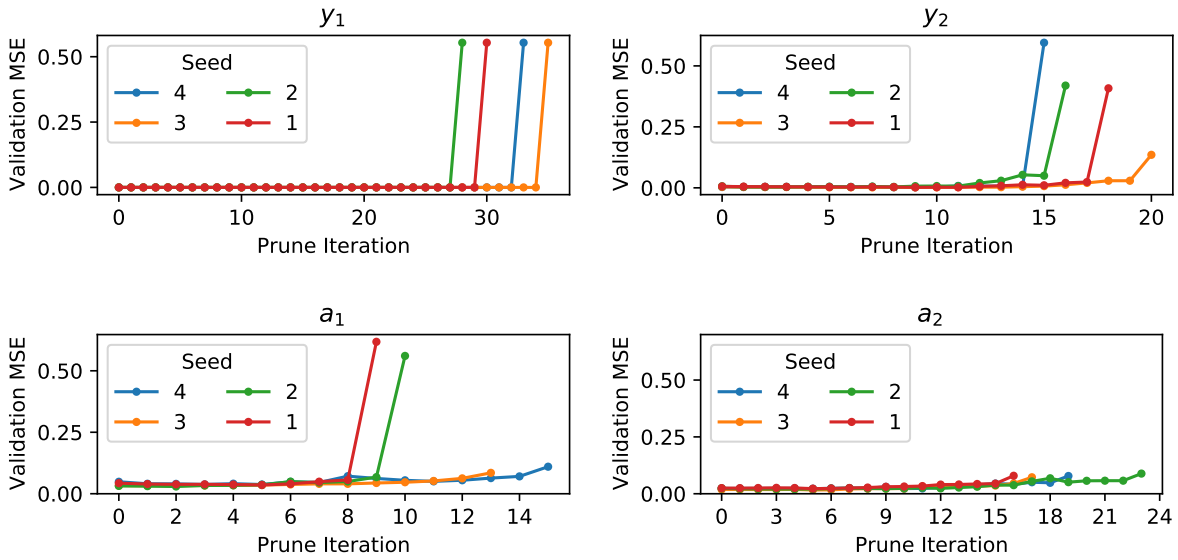


Figure 5.5: Validation MSE at each pruning iteration during pairwise network training on all synthetic data sets for 4 initialisation seeds. 10% of parameters are pruned per iteration and fine-tuning is done over 100 epochs every time.

Ranking

Iterative pruning improves the ranking capability of pairwise networks significantly. This is made clear in Figure 5.6, which shows the ψ-distribution of pruned pairwise networks trained on the synthetic data sets. For all cases, except seed 2 of a_1, a low ranking ($\psi = 0$) is attributed to parameters that do not form part of data generation. An analysis of results shows that an input feature can be completely pruned from a sub-network. In the ψ-distribution figures, we renamed such sub-networks to have the same name as the remaining input and marked them with a *.

Analysis of the $\{B_T, r\}$ sub-network's summary node activation values for task a_1, seed 2 (see Figure 5.6) reveal that this sub-network outputs a constant value for any sample. Therefore, it is essentially reduced to a bias on the output layer of the network.

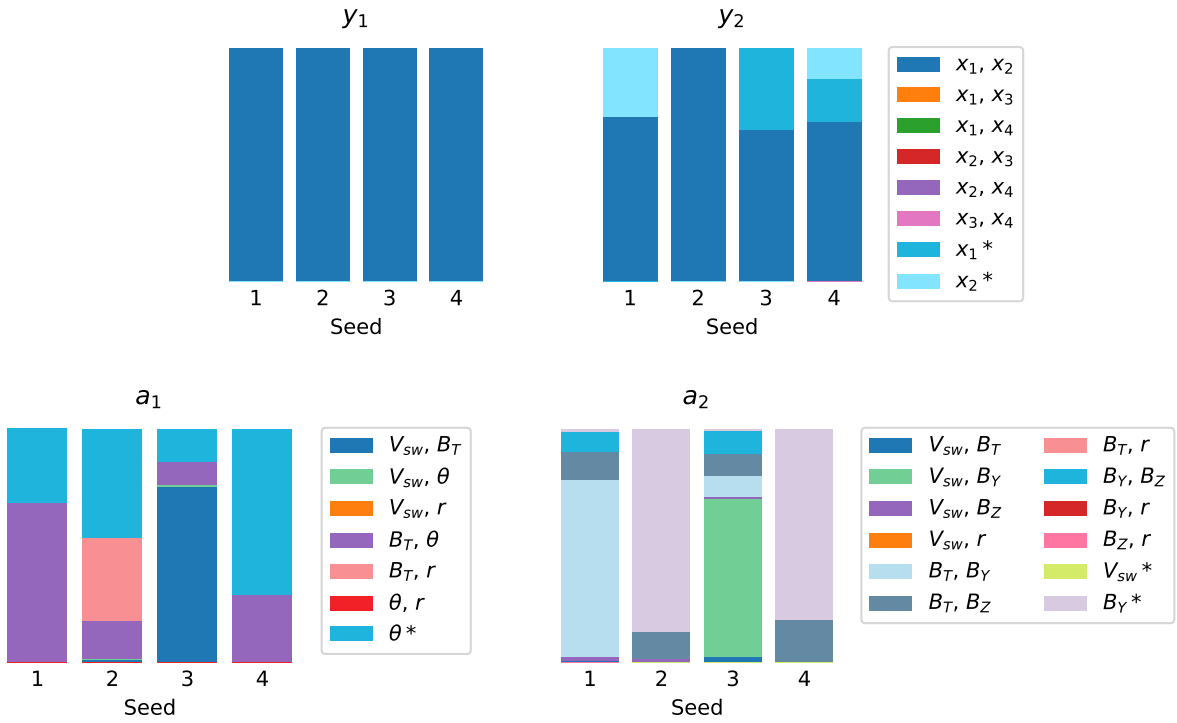


Figure 5.6: ψ-distributions of pairwise networks trained with iterative pruning by removing 10% of weights each iteration and fine-tuning for 100 epochs. These distributions are determined by passing the evaluation set of the respective data sets through the network. It can happen that the training algorithm completely prunes one of the inputs from a sub-network. We renamed such sub-networks to have the same name as the remaining input and marked them with a *.

Pruning amount

We tested pruning amounts $p = \{10, 15, 20, 25, 30\}$ on the synthetic tasks. The performance of the resulting networks are provided in Table A.1 and the pairwise rankings are compared in Figures A.2 and A.3. All of the pruning amounts produce networks with very similar performance, with an absolute difference in evaluation MSE less than 0.03 on each task. The rankings are also very similar, but $p = 10$ produces the best ranking in all cases (when fine-tuning for a fixed number of epochs). For this reason, we choose $p = 10$ for future pairwise network training.

Fine-tuning strategies

We investigate two fine-tuning strategies, namely (i) training networks further for a fixed number of epochs each iteration and (ii) training to completion each iteration. The performance and training time per iteration are provided in Table A.1 and the pairwise rankings are compared in Figures A.2 and A.3. Both strategies achieve approximately the same performance and ranking capability, but the former (i) trains faster in most cases. For this reason, the fine-tuning steps of subsequent pairwise network training will be done for a fixed number of epochs.

5.4.4 Initialisation

In this section we compare the effect of duplicate sub-network initialisation to that of random initialisation on pairwise networks trained without iterative pruning. Both schemes have approximately the same performance. (An absolute difference of no more than 0.001 MSE on the evaluation set across all synthetic tasks.) Figure 5.7 compares the ψ-distribution of networks initialised with the respective schemes. For the y_1 data set, duplicated sub-network initialisation results in a better ranking than random initialisation, but the reverse is true for y_2. Duplicated sub-network initialisation does not make a noticeable difference to the network's ranking on a_1 and a_2.

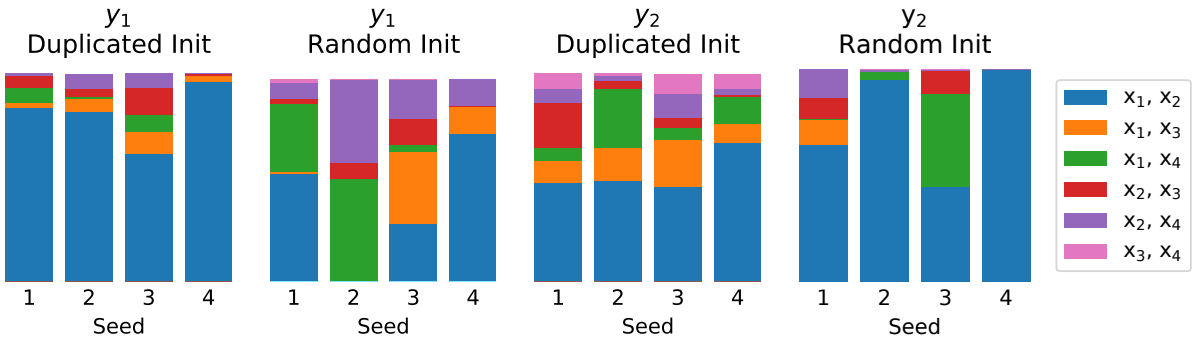


Figure 5.7: ψ-distributions of pairwise networks trained without pruning and initialised with random and duplicate sub-networks, respectively. Measurements are taken from the evaluation set of synthetic data sets y_1 and y_2.

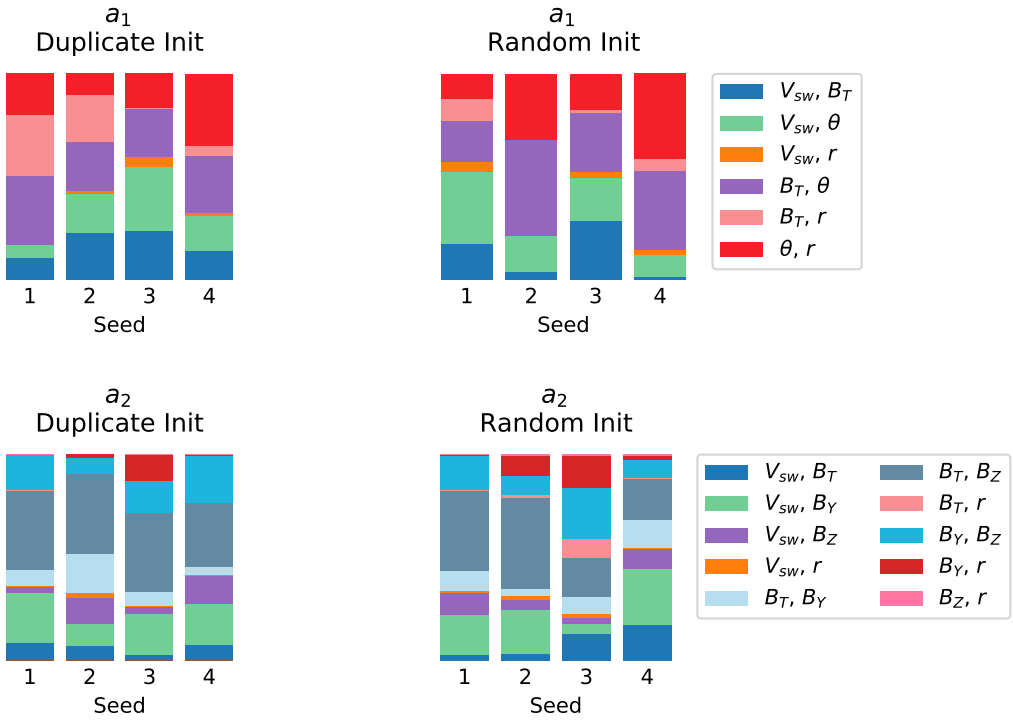


Figure 5.8: ψ-distributions of pairwise networks trained without pruning and initialised with random and duplicate sub-networks, respectively. Measurements are taken from the evaluation set of synthetic data sets a_1 and a_2.

5.4.5 Correlation Analysis

Our synthetic analysis shows that some initialisation seeds lead to networks with excellent predictive performance, but a poor ranking capability. To investigate these error cases, we analyse the correlation between the output of each sub-network and its input features (as proposed in Section 5.2.5).

For a demonstration, consider the pairwise network baseline for synthetic data set a_1. The ψ-distribution of this network is presented in Figure 5.4, which does not seem to match its MSE of 0.0153. This graph suggests that the network assigns significant importance to pairs $\{B_T, r\}$ and $\{\theta, r\}$, and hence the unwanted feature r. However, closer inspection reveals a weaker reliance on r. In Table 5.3 we show the PCC between the weighted summary node activation values of each sub-network and its corresponding input features. Notice that for every seed, the correlation between feature r and the output of its sub-

network is negligible, while the correlation between the output and the other feature of the r-pair, is high (barring pair $\{V_{sw}, r\}$, seed 1). The same effect can be seen for several of the other pairs. For example, the output of sub-network $\{V_{sw}, B_T\}$ (seed 1) has a strong correlation with B_T, but a weak correlation with V_{sw}. Many of these unnecessary relationships will be discarded during pruning, but ones that remain can be analysed using correlation analysis.

Table 5.3: PCC between the activation values of each sub-network and its corresponding input features for the baseline pairwise network of synthetic data set a_1. Measurements are taken on the evaluation set.

	V_{sw}, B_T	V_{sw}, θ	V_{sw}, r	B_T, θ	B_T, r	θ, r
Seed 1						
V_{sw}	-0.09	0.10	0.29	-	-	-
B_T	0.90	-	-	-0.18	1.00	-
θ	-	-0.25	-	0.40	-	-0.40
r	-	-	0.02	-	-0.01	0
Seed 2						
V_{sw}	-0.16	0.15	0.99	-	-	-
B_T	0.91	-	-	-0.29	1.00	-
θ	-	0.10	-	0.14	-	-0.32
r	-	-	0.10	-	-0.01	0
Seed 3						
V_{sw}	-0.51	0.16	1.00	-	-	-
B_T	0.69	-	-	-0.31	0.64	-
θ	-	-0.96	-	-0.67	-	0.87
r	-	-	0	-	-0.01	0
Seed 4						
V_{sw}	-0.09	-0.02	0.85	-	-	-
B_T	0.90	-	-	-0.13	0.98	-
θ	-	-0.96	-	0.98	-	-0.91
r	-	-	0	-	-0.10	0

5.5 Pairwise network summary

The pairwise network is an intrinsic interpretable model that enables the ranking of feature importance through the analysis of activation and weight values. It is constructed by grouping inputs in pairs of two and supplying each pair with a sub-network. All sub-networks are summed to produce a single output for regression modelling (Section 5.2.1).

Prior to training, each sub-network is initialised with the same set of weights, by initialising the entire network with the Kaiming [42] scheme and then duplicating the weights of a single sub-network to the others (Section 5.2.3). Iterative pruning and fine-tuning is applied during training as a method to improve ranking capability (Section 5.2.4). The network is first trained to convergence, then $p\%$ of weights are removed and the remaining weights are fine-tuned by training the network further and selecting the best network within a fixed number of epochs. Pruning and fine-tuning are repeated until the network reaches a desired level of sparsity or until the validation MSE falls below a predetermined threshold of the best performing network over all iterations. In some cases, one of the input features is entirely pruned from a sub-network, in which case we regard the output as a transformed version of the remaining feature (Section 5.2.5).

Feature ranking is performed by passing a set of samples through the trained pairwise network and capturing each sub-network's output to determine the ψ (Equation 5.2) of the corresponding input pair.

5.6 Conclusion

In this chapter, we introduced the pairwise network, an adapted MLP that enables feature ranking without significant sacrifices in performance. The pairwise network group input features into pairs and each pair receives a sub-network. The output of the network is the weighted sum of all sub-networks. To measure the rank of a pair, we defined the metric ψ, which is the relative importance of a pair based on the output of its sub-network.

We used synthetic data sets to validate the training and ranking procedure of the pairwise network. It was shown that the pairwise network alone is able to achieve near-perfect performance on the synthetic data sets, but requires iterative pruning during training to have an accurate ranking capability. With iterative pruning, the pairwise network is able to identify features that do not form part of data generation, at the cost of slightly worse performance (A maximum increase of 0.0193 MSE and a maximum decrease of 0.0183 PCC on the evaluation set across all synthetic tasks). Several pruning amounts and two fine-tuning strategies were tested. We found that pruning 10% of remaining parameters and then fine-tuning for a fixed number of epochs (as opposed to training to completion) each iteration, yields pairwise networks with a good ranking capability and reduced training time per iteration.

We tested two types of initialisation schemes and found no significant difference between them. Finally, we demonstrated how the correlation between the output of a sub-network and its input features can be used to reveal cases where a sub-network only models one feature and disregards the other.

During this analysis, we noticed two shortcomings of the current pairwise network: The pruning process is computationally expensive since the network has to be retrained at every pruning iteration. Since every combination of input pairs receives a sub-network, the model grows quickly, becoming unwieldy. Therefore, the current pairwise network is limited to smaller tasks.

In the next chapter, we will use the pairwise network to analyse a more complex task.

Chapter 6

Pairwise Network for SYM-H

In this chapter we investigate the performance and ranking capability of the pairwise network on the solar wind/SYM-H data set. We optimise the hyperparameters of several pairwise networks on this data set; compare their performance to baseline MLP models; and compare the network's feature rankings to existing space physics.

6.1 Introduction

In this chapter we train several pairwise networks on the solar wind/SYM-H data set described in Chapter 3. We show that these models perform slightly better than the multilayer perceptron (MLP) baselines developed in Chapter 4 and that the input parameter rankings produced by these networks agree with our current understanding of solar wind – magnetosphere coupling during geomagnetic storms.

Four pairwise models are optimised (P_1 to P_4, where each P_i is the pairwise equivalent of B_i): As with the MLP models, the first pairwise network uses the solar wind input parameters as is, the second adds temporal information in the form of time-shifts, and

the third adds a calculated storm phase indicator as an additional parameter. The final network combines the features of the last two models. When used, time-shifted parameters are included in the same sub-network as their unshifted counterparts. Likewise, phase inputs are included in every sub-network of the pairwise structure as well as their shifted versions (when time-shifts are used together with phase inputs).

6.2 Experimental setup

Our experimental setup for training pairwise networks is exactly the same as for the MLP optimisation in Section 4.2, except that we do not consider batch normalisation during hyperparameter tuning. The pairwise network architecture, initialisation scheme and training procedure are as described in Section 5.5. We chose a batch size of 1 024 for pairwise network training, based on our observations in Section 4.3. Following the results of Section 4.3.2, a shift size of 270 minutes is used whenever time-shifts are added to the network.

Since iterative pruning is computationally expensive, we first obtain approximate values of hyperparameters for a network without pruning. For the unpruned networks, we dynamically add training epochs as discussed in Section 4.2. The iterative pruning procedure is implemented by first training networks for a maximum of 200 epochs, then pruning 10% of remaining parameters and fine-tuning for 40 epochs each iteration. If any iteration has an early-stopped validation MSE above a predetermined threshold of the best network in the previous iterations, training is halted and the previous pruning iteration is selected.

The pruning threshold value is determined heuristically based on the learning curves during training, similar to what we did for the synthetic tasks in Section 5.4.3. The learning curves of our initial experiments showed that the validation performance either improves or stays approximately constant before degrading. Therefore, this threshold should account for the little variance between iterations (so that the network can become sparse enough), but prevent a significant loss in performance. From the learning curves we determined that a threshold of 0.02 satisfies our criteria.

6.3 Optimisation

In this section we discuss the procedure followed to optimise the pairwise network on the solar wind/SYM-H data set.

6.3.1 Pairwise network

First we obtain approximate values for hyperparameters by training several networks without pruning. For the unpruned P_1 network, we do a grid search over sub-network layouts {1x5, 1x10, 2x5, 2x10, 3x5}, weight decay values {0, 0.001, 0.01, 0.1}, learning rates $\{10^{-2}, 10^{-3}, 10^{-4}, 10^{-5}, 10^{-6}\}$ and 3 initialisation seeds. Note that a pairwise network with a sub-network layout DxH implies that every sub-network has D hidden layers of H nodes each.

The results for this search are provided in Table 6.1. It shows that the right amount of weight decay can lead to a minor performance improvement, but too much weight decay results in a significant loss of performance. Therefore, we choose not to use weight decay from here onward. The 1x10 and 2x10 sub-network layouts have approximately similar performance and both do better than the other layouts. From the table it is also clear that the performance is relatively insensitive to small changes in the learning rate.

Similarly, we construct P_2, P_3 and P_4 without pruning by training pairwise networks with and without phase or time-shifted inputs. For each model type we do a search over learning rates $\{5 \times 10^{-4}, 10^{-4}, 5 \times 10^{-5}, 10^{-5}\}$ and 3 seeds. We use a 2x10 sub-network layout without weight decay. The performance of these networks is provided in Table 6.3.

Table 6.1: Average MSE of the unpruned pairwise network, measured across 3 seeds on the validation set of the solar wind/SYM-H data set, for several sub-network layouts, weight decay values and learning rates. The networks trained with a weight decay value of 0.1 all have a validation MSE greater than 1.2.

Weight Decay	Learning Rate	Sub-network Layout				
		1x5	2x5	3x5	1x10	2x10
0	1E-6	1.0651	1.1880	1.3131	0.9274	0.9278
	1E-5	0.9249	0.9263	**0.9251**	0.9189	0.9192
	1E-4	0.9242	0.9251	0.9256	0.9194	0.9214
	1E-3	0.9392	0.9443	0.9390	0.9350	0.9430
	1E-2	0.9949	0.9689	0.9974	0.9808	0.9777
0.001	1E-6	1.0643	1.1888	1.3131	0.9266	0.9263
	1E-5	0.9242	1.0536	1.2141	**0.9176**	**0.9181**
	1E-4	0.9237	**0.9210**	1.1865	0.9184	0.9201
	1E-3	0.9394	0.9388	1.2162	0.9352	0.9424
	1E-2	0.9851	0.9754	1.2670	0.9815	0.9889
0.01	1E-6	1.0646	1.3118	1.3131	1.0579	1.3125
	1E-5	**0.9232**	1.2150	1.3130	0.9197	1.2123
	1E-4	0.9238	1.1860	1.3130	0.9214	1.1871
	1E-3	0.9426	1.1933	1.3131	0.9424	1.2210
	1E-2	0.9670	1.2701	1.3131	0.9643	1.3131

6.3.2 Pairwise network with pruning

In this section we optimise each of the pairwise networks (P_1 to P_4) with iterative pruning.

We choose the 2x10 layout, with the intention of making enough parameters available for pruning, and do a grid search over learning rates $\{5 \times 10^{-4}, 10^{-4}, 5 \times 10^{-5}, 10^{-5}\}$, across 3 seeds and without weight decay. The validation MSE of this grid is provided in Table 6.2. It shows that none of the networks lie on the edge of the grid and that the model remains relatively insensitive to learning rates. The performance of the best pruned pairwise networks are provided in Table 6.3, together with the unpruned versions and the MLP baselines. In all cases, both the pruned and unpruned pairwise networks achieve

lower evaluation MSE and higher PCC than the MLP baselines. With the exception of the pruned P_3, these differences are significant as they fall outside the confidence intervals, therefore we can confirm that all other pairwise networks perform slightly better than their MLP counterparts. Except for P_3, training pairwise networks with pruning results in approximately the same performance as without it.

Table 6.2: The validation MSE of pairwise networks trained with pruning on the solar wind/SYM-H data set for several learning rates, with and without time-shifted and phase inputs and averaged across 3 initialisation seeds.

Time-shift	Learning Rate	Phase	
		With	Without
With	1E-05	0.4414	0.6241
	5E-05	**0.4009**	**0.5847**
	1E-04	0.4052	0.5861
	5E-04	0.4180	0.6195
Without	1E-05	0.5431	0.9132
	5E-05	0.5048	0.9160
	1E-04	**0.4986**	**0.9113**
	5E-04	0.5220	0.9206

Temporal and phase information shows a marked improvement in performance and unlike the MLPs, the combination of both leads to better results than the separate elements. For all the models, there exists a substantial difference between the validation and evaluation performance. As noted in Section 4.4, this could be due to the specific distribution of storm events between the partitions.

For an example of the pairwise network's modelling capability, the observed and predicted SYM-H of two storms, taken from the evaluation set, are shown in Figure 6.1. Predictions are made using the best pairwise network trained with iterative pruning and having both phase and time-shifted inputs (P_4). The September/October 2012 event shows a peculiar response by the model: Predicted SYM-H remains relatively constant for most of the main phase while measured SYM-H follows the usual decreasing trend expected for the storm main phase. The model response may be due to the unique characteristic of this event – a second, prolonged enhancement in density and pressure (N_p and P_d) causing the magnetopause to be pushed towards Earth near geosynchronous orbit [79]. We speculate

that the model interprets this kind of constant high N_p or P_d as enhancing SYM-H, similar to what is observed at coronal mass ejection (CME) arrival or just before the storm onset, causing the effect of negative B_Z or high V_{sw} working against enhanced P_d. This shows that empirically modelled results need to be interpreted carefully since unique events such as this are not adequately represented in bulk data sets used to train the models.

Table 6.3: Average MSE and PCC of the pruned and unpruned pairwise network across 3 seeds on every partition of the SYM-H data set, with the SE of each metric on the evaluation set provided in brackets. The performance of the MLP baselines are provided for comparison. ΔnT interprets the MSE value in the original unit of SYM-H (nT).

Model	MSE			PCC			ΔnT
	Train	Valid	Eval (SE)	Train	Valid	Eval (SE)	Eval
MLP							
B_1	0.61	0.90	0.42 (0.0115)	0.63	0.56	0.65 (0.0102)	29.82
B_2	0.35	0.59	0.33 (0.0161)	0.81	0.76	0.77 (0.0097)	25.75
B_3	0.34	0.50	0.26 (0.0172)	0.82	0.79	0.82 (0.0033)	23.51
B_4	0.19	0.41	**0.25** (0.0042)	0.90	0.84	0.85 (0.0018)	22.52
Pairwise Network Without Pruning							
P_1	0.59	0.92	0.40 (0.0005)	0.64	0.54	0.67 (0.0005)	28.91
P_2	0.33	0.61	0.31 (0.0015)	0.82	0.76	0.78 (0.0008)	25.06
P_3	0.34	0.49	0.22 (0.0005)	0.81	0.79	0.84 (0.0004)	21.45
P_4	0.22	0.42	**0.21** (0.0002)	0.88	0.84	0.87 (0.0007)	20.46
Pairwise Network With Pruning							
P_1	0.60	0.92	0.39 (0.0084)	0.63	0.54	0.68 (0.0079)	28.55
P_2	0.34	0.60	0.30 (0.0034)	0.81	0.76	0.79 (0.0031)	24.63
P_3	0.35	0.50	0.24 (0.0146)	0.81	0.79	0.83 (0.0081)	22.46
P_4	0.23	0.43	**0.21** (0.0060)	0.88	0.84	0.86 (0.0036)	20.78

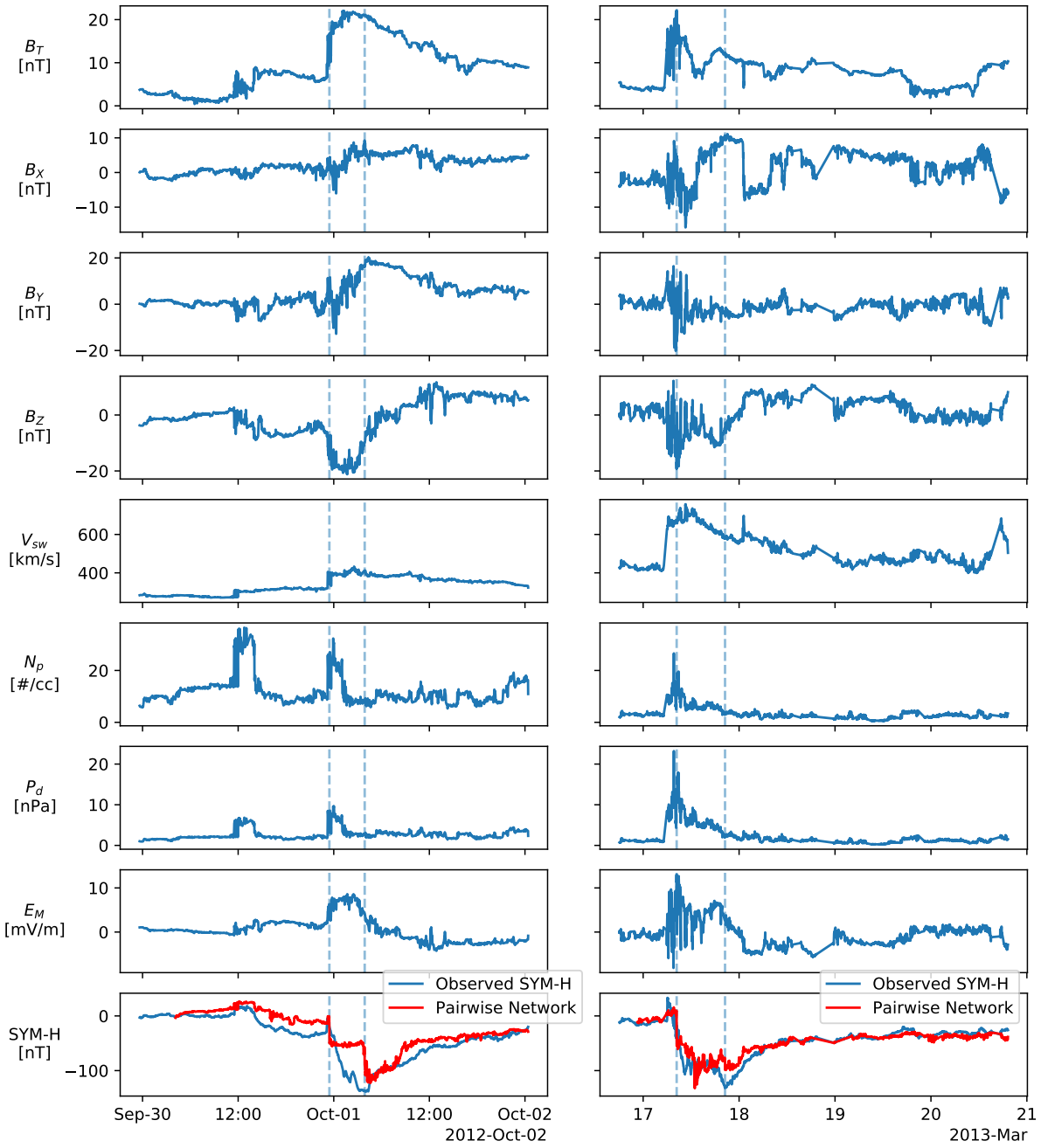


Figure 6.1: Two geomagnetic storms taken from the evaluation set. Each column is a separate event and each row shows a different solar wind parameter, except for the bottom row, which presents the observed SYM-H and the predictions made by the best performing pairwise network (P_4 – phase and time-shifted inputs included) trained with iterative pruning. The dashed vertical lines indicate phase transitions.

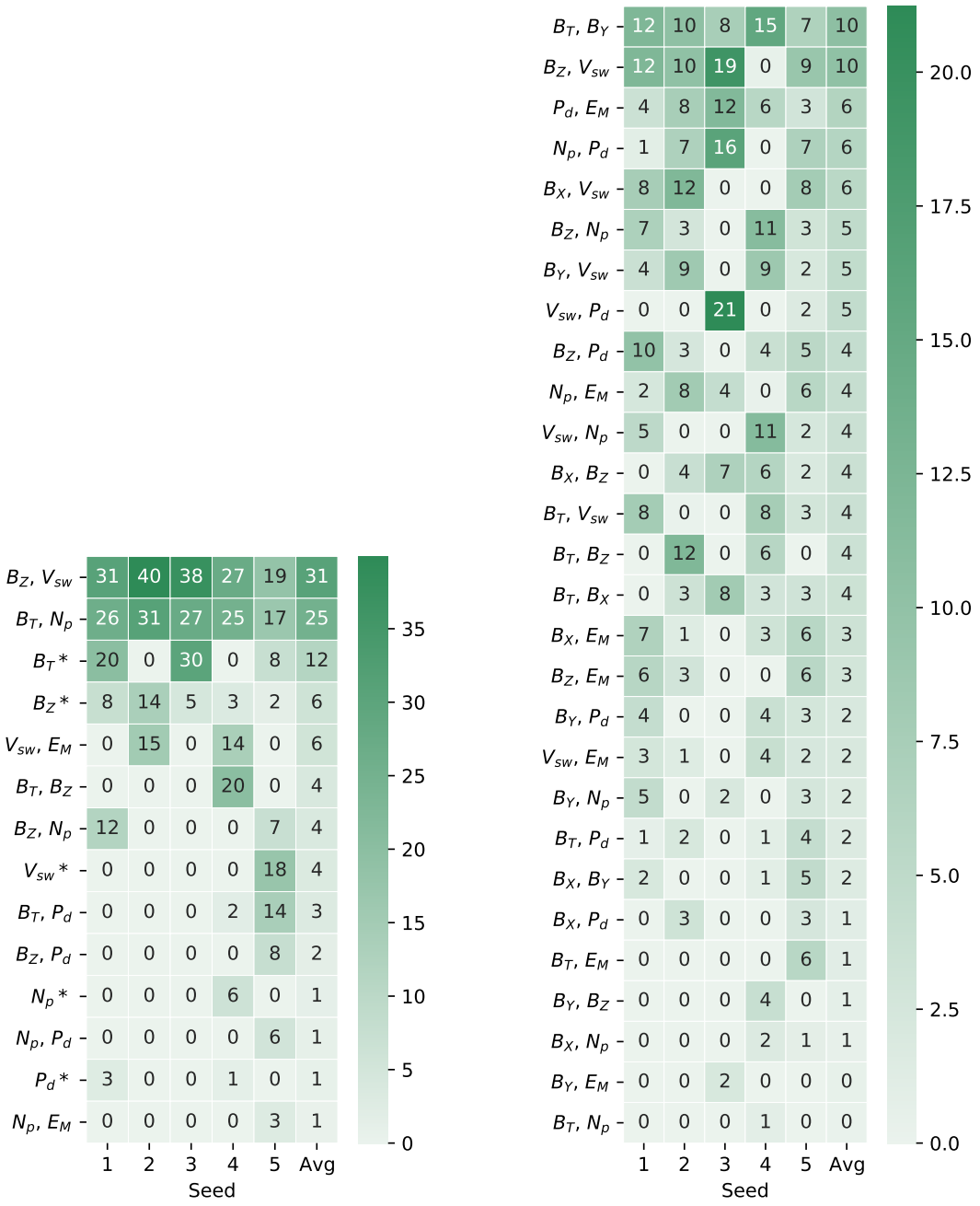
6.4 Pairwise Ranking

In this section, we evaluate the input parameter ranking produced by the pairwise network against our current understanding of the physical processes involved in the solar wind-geomagnetic field coupling during geomagnetic storms. Rankings are calculated by passing a set of samples through the network and then taking the weighted summary node activation ratio (ψ) of the set like we did in Chapter 5. Pairwise network ranking is measured on the evaluation set for (i) the entire set, and (ii) with the set divided according to the 3 storm phases. The latter is to see if the input ranking via the pairwise networks reflects the known differences in physical phenomena at play during the different phases.

The ranking produced by models P_1 and P_4 are discussed in this section, but the results for models P_2 and P_3 can be found in Appendix A.3. We added two more initialisation seeds for this analysis to show that there exist considerable variance between the rankings of different seeds. The absolute difference between the evaluation MSE of these additional networks and the corresponding results in Table 6.3 are less than 0.02.

6.4.1 Entire evaluation set

The pairwise rankings produced by P_1 and P_4, when passing the evaluation set through the networks, are presented in Figure 6.2. Rankings vary by seed, but as explained below, the top-ranked parameters for most seeds point to similar physical processes at work.



(a) P_1 (b) P_4

Figure 6.2: ψ values (in percentage) extracted by networks P_1 and P_4 on the evaluation set of the solar wind/SYM-H data set for 5 seeds, as well as the average across all seeds. A larger ψ value corresponds to a higher rank. Sub-networks for which one of the input parameters were pruned away during training are renamed to the remaining input and marked with a *.

In each of the five sets of ranks performed by P_1, the top four parameter pairs are related to the dynamic pressure applied by the solar wind to the magnetosphere (V_{sw} and N_p), the coupling of the solar wind plasma to the magnetospheric plasma (E_M, B_Z, and V_{sw}) and B_T – the total interplanetary magnetic field (IMF) – which indicates the amount of IMF flux at the dayside magnetopause [80]. It is peculiar that, for all networks except seed 5, the dynamic pressure parameter P_d ($\sim N_p V_{sw}^2$) does not feature in the top-ranked input parameters. This may be due to the influence of V_{sw} and N_p being strong indicators of solar activity and the training algorithm not responding to features duplicated by P_d. The sun-ward (X) and ecliptic (Y) components of the IMF (B_X and B_Y) consistently achieve low rankings. This is to be expected as B_Z is the only component that is directly related to reconnection between the IMF and the geomagnetic field, although recent results point to an apparent influence of B_Y on SYM-H (see next paragraph).

P_4 produced rankings with more variance between seeds. Seeds 1, 2 and especially seed 3 closely match our understanding of the physical process (as discussed above for P_1). The high ranking of B_Y is interesting as it does not generally feature as an important coupling parameter in its own right [12], [80]. However, B_Y is related to the clock angle of the IMF which does play a role in coupling efficiency, and recent results by [81] show that B_Y is related to the symmetry in ring current especially during times of negative B_Z. This does not fully explain the high rank of the $\{B_T, B_Y\}$ pair but may be of interest for a more detailed investigation. Seed 4 assigns a high rank to B_Y (although parameters B_Z, V_{sw} and B_T are also ranked high) and seed 5 shows no clear preference towards any particular set of parameters.

6.4.2 Separate phases

In this section we extract pairwise rankings from the solar wind/SYM-H data set for each of the three phases separately, using model P_1.

Figure 6.3 presents the ψ values produced by this model across 5 seeds. The precise values for each seed are provided in Figure A.5 in Appendix A.3, along with the rankings

of models P_2, P_3 and P_4. Ranking results of the other pairwise networks reveal similar trends as P_1, but with less clarity.

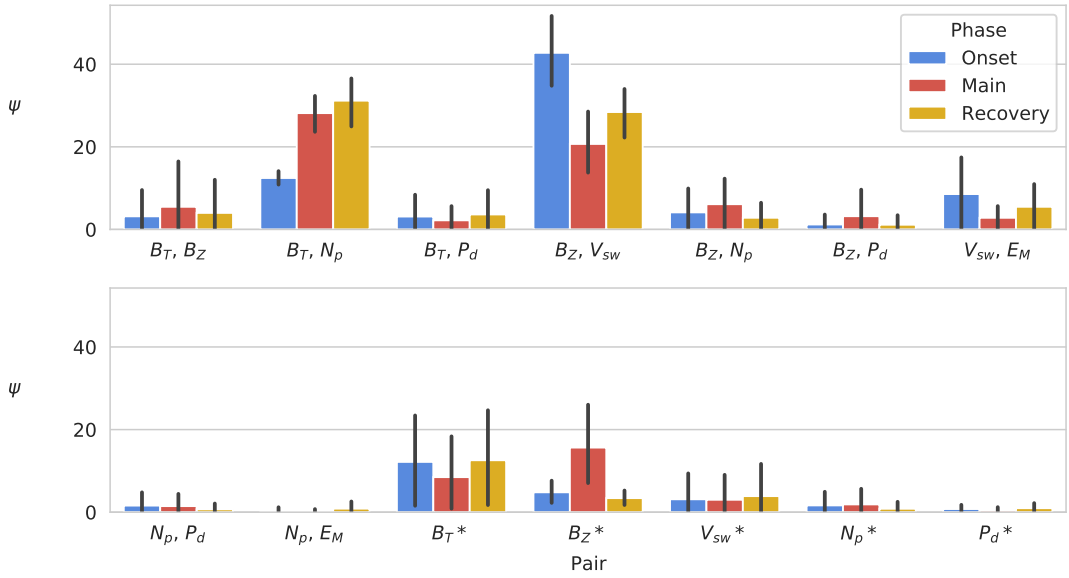


Figure 6.3: ψ values extracted by P_1 on the evaluation set of the solar wind/SYM-H data set for each phase separately. The results show the average across 5 seeds with error bars to indicate the 95% confidence interval.

Onset

On average it is V_{sw} and B_Z that dominate the onset phase in the rankings produced by P_1 in Figure 6.3. It makes sense that one of the pressure-related terms (P_d, N_p or V_{sw}) would dominate since for intense storms the onset phase is usually characterised by the so-called sudden storm commencement – caused by a sudden rise in solar wind dynamic pressure which compresses the day side magnetosphere, leading to an increase in horizontal magnetic field measured on the ground [12]. This increase manifests in SYM-H as a significant upward pulse lasting minutes to hours.

The onset phase is defined as the period before significant reconnection takes place and hence the B_Z and E_M components should not play a big role here. We speculate that the prevalence of B_Z is due to the way we defined the transition between the onset and main phase in our data set. On average, B_Z has a negative slope from approximately 80

minutes before the end of the onset phase to the start of the main phase (calculated over the entire dataset, as shown in Figure A.9). About 50 minutes later, SYM-H starts with its initial sharp decrease, which may cause an increased correlation with B_Z.

Main

The main phase of a storm is defined by the magnetic reconnection between the IMF and geomagnetic fields which happens when B_Z turns negative and maintains a large southward component for several hours, allowing field lines at the day side magnetopause to merge IMF lines. Therefore the high rankings of B_Z in Figure 6.3 is expected [78]. During this time the solar wind speed V_{sw} acts as a modulator of the energy input into the open magnetosphere [12]. The passage of the CME is typically characterised by high V_{sw}, with solar wind speed gradually returning to ambient levels after the CME has passed. During the first part of main phase B_T and N_p are typically high and fluctuating, before decreasing towards the end of main phase as the disturbed solar wind passes over the magnetosphere. (For an example see Figure 2.2.) This pair of parameters are indicative of the magnetic flux and particle density in the CME plasma.

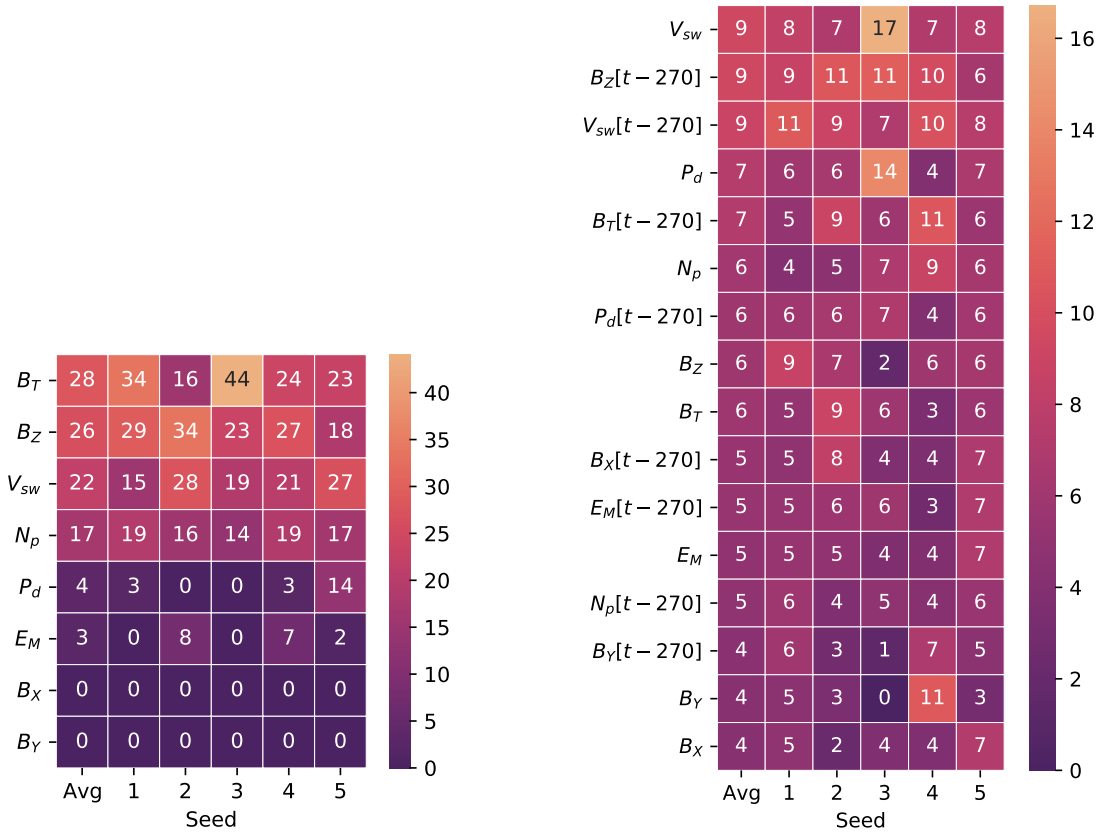
Recovery

During storm recovery, it is the absence of energy input from the solar wind that allows the magnetosphere to recover by various internal wave-particle interactions that drain energy from particle populations [78]. After a CME passes, the solar wind speed gradually decreases to ambient level and the fluctuations in the IMF decrease significantly. Here the top-ranking inputs are B_T, B_Z, N_p and V_{sw}. Total IMF B_T relates to the open flux at the IMF–magnetopause interface and is usually characterised by a gradual decrease during the recovery phase. During CME-driven disturbances it is typically the Z-component that varies more coherently than the B_X or B_Y components, and therefore there is high correlation between B_Z and B_T, especially during the main and recovery phases.

6.5 Feature Ranking

In this section we translate the pairwise rankings (determined in the previous section) to a ranking of individual input features: *Attribution values* are estimated for the input parameters by dividing each sub-network's ψ equally between every unpruned input connected to the sub-network (thereby assuming they contribute equally) and then summing over all sub-networks.

The attribution values assigned by P_1 and P_4, when passing the entire evaluation set through the networks, are presented in Figure 6.4. In Figures 6.5 and 6.6 we show the feature attributions by the same networks, but for separate storm phases.



(a) P_1 (b) P_4

Figure 6.4: Feature attribution values extracted by networks P_1 and P_4 on the evaluation set of the solar wind/SYM-H data set for 5 seeds, as well as the average across all seeds. The time-shifted version of parameter x is denoted as $x[t-270]$.

The feature attribution by P_1 exhibits the same behaviour as discussed in Section 6.4.1, but P_4 reveals that, in most cases, both the instantaneous and the time-shifted parameters achieve high rankings (as shown in Figure 6.4b). This confirms that the current SYM-H values are driven by current and previous conditions in the solar wind, which is indeed the case since the solar wind–magnetosphere coupling depends on various processes, with different time dependencies. (For an example see [80].)

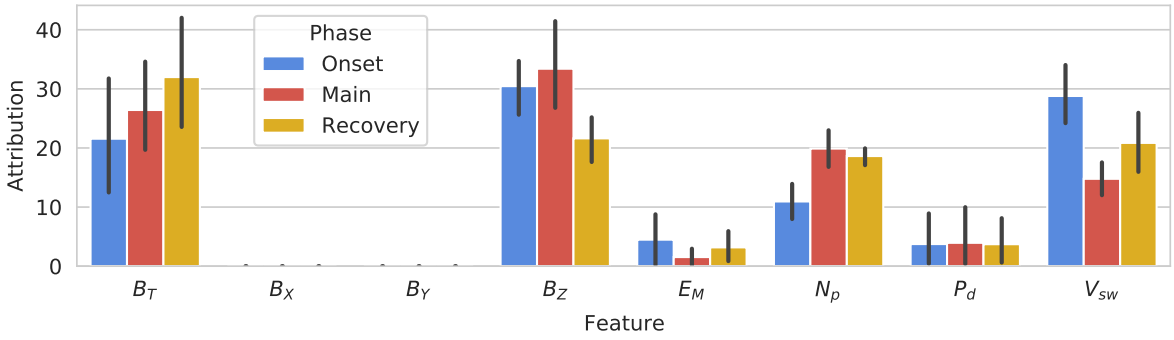


Figure 6.5: Feature attribution values extracted by P_1 on the evaluation set of the solar wind/SYM-H data set for each phase separately. The results show the average across 5 seeds with error bars to indicate the 95% confidence interval. The precise values for each seed are shown in Figures A.11 and A.12.

Feature rankings for the onset phase in Figures 6.5 and 6.6 align with our observations in the previous section, but more can be drawn from the feature rankings of the main phase. The main phase of a storm is defined by the energy input in to the magnetosphere facilitated by southward B_Z that allows reconnection and the pressure applied by the solar wind (defined by speed and density of the plasma), resulting in the ring current enhancement measured by SYM-H. The critical solar wind parameters that enable the coupling between the solar wind has been studied extensively and are understood to be pressure terms (P_d, N_p and V_{sw}) and IMF parameters B_Z, B_Y and B_T [12]. According to Figures 6.6 and A.12b we see that the pairwise network selects $B_Z[t-270]$ (indicating reconnection, with time delay), P_d (fluid pressure on the magnetosphere), V_{sw} (current speed and proxy for energy input), $E_M[t-270]$ (merging electric field indicating coupling, with time delay).

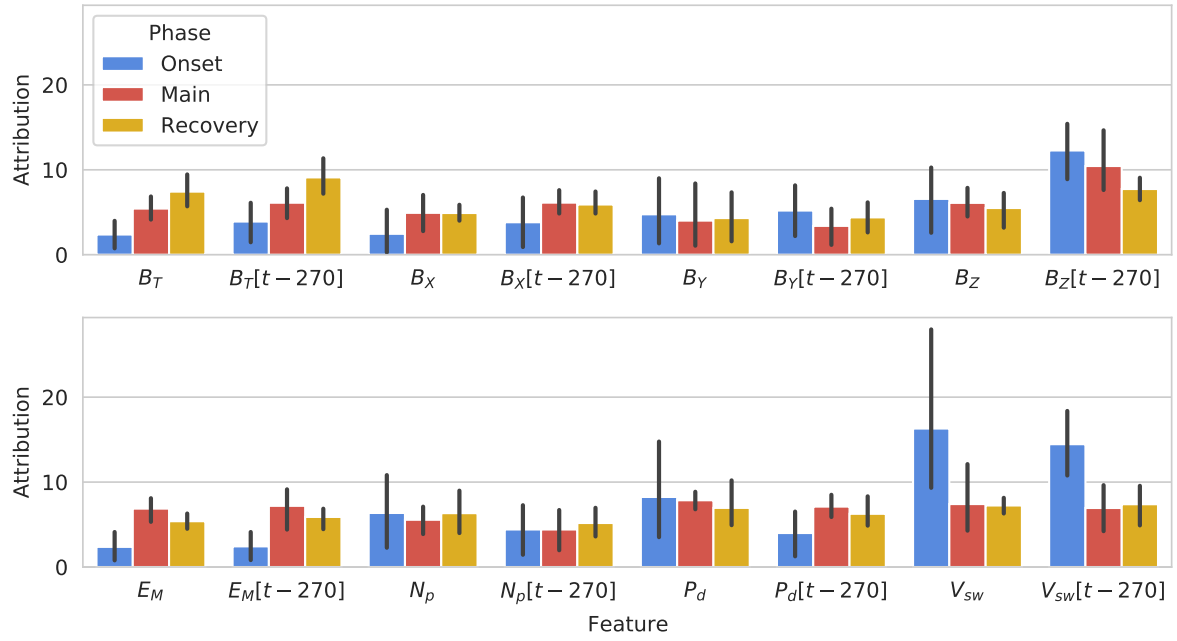


Figure 6.6: Feature attribution values extracted by P_4 on the evaluation set of the solar wind/SYM-H data set for each phase separately. The results show the average across 5 seeds with error bars to indicate the 95% confidence interval.

The top six of the ranked parameters in Figure A.12b are all in line with the physical mechanisms at play. There are some interesting candidates below the top six:

- $B_Z[t]$ at 10th place seems to indicate that historical (270 min in the past) B_Z is important but not the current B_Z. This seems to suggest that there is a time delay in the reaction of the inner magnetosphere to negative B_Z (and dayside reconnection).
- Note the interplay between V_{sw}, N_p and P_d: The network for which V_{sw} had its highest score (seed 3, score 16) N_p had its lowest score of 3; The network for which N_p has highest score (seed 2, score 8) V_{sw} had its lowest score of 3; All the while P_d had the same score (9 for both seeds 2 and 3). This seems to suggest the network is capable of drawing similar information from the combination of the three pressure-related terms.

The differences in attribution between shifted and unshifted parameters in Figures 6.6 and A.12b are quite interesting. For the pressure terms (V_{sw}, N_p and P_d), the attribution

values of unshifted parameters are greater than for shifted parameters during the onset phase, but approximately the same during the main and recovery phase. For B_T and B_Z, the attribution values of the unshifted parameters are less than for the shifted parameters across all of the phases. So historical IMF is important for all phases, but that immediate values are important for pressure terms. This is in agreement with our understanding of the time dependence of the various processes at play. Reconnection (B_Z) takes time to manifest on the ground while the effect of increased pressure at onset happens almost immediately, and the time *before* onset is quiet, so there is definitely no impact then.

6.6 Conclusion

In this chapter we developed several pairwise networks for predicting the SYM-H index from solar wind parameters and then compared the rankings produced by these networks to our understanding of the underlying physics.

Without iterative pruning, the best pairwise network achieved a root mean squared error (RMSE) of 20.46 nT and a correlation of 0.87 on the evaluation set, which is better than the best MLP baseline, with a RMSE of 22.52 nT and a correlation of 0.85. Pruning slightly decreased performance of the pairwise network to a RMSE of 20.78 nT and a correlation of 0.86, but it still performed better than the MLP baselines. Temporal and phase information resulted in a significant performance increase, but unlike the MLPs, using time-shifts together with phase inputs leads to better performance than the individual elements.

The ranking extracted by the pairwise networks corresponds to the existing understanding of space physics, albeit with some variance between seeds. We reported on the rankings drawn from the evaluation set, but the same conclusions can be made using the training and validation set. (Results not shown.) This is to be expected from the way the data is divided in Chapter 3.3. Without time-shifted or phase inputs (model P_1) the pairwise network correctly separated important parameters from those less important to the regression problem in the context of a set of highly mutually correlating input parameters

(given that all the solar wind parameters are characteristics of the same volume of plasma and its associated magnetic field). Duplicated information due to parameters representing similar physical parameters were not reflected in the ranking. Furthermore, extracting rankings for the three storm phases in isolation showed that parameter importance vary with storm progression in a manner that agrees with the physics.

Adding phase and time-shifted inputs (model P_4) lead to similar a ranking as P_1, but with more variance between seeds and with some seeds that produced a poor ranking. However, the rankings of P_4 did reveal that it considered historical IMF parameters as important for all phases, but that immediate values are important for pressure terms. This is in agreement with our understanding of the time dependence of the various processes at play.

We found that with the 8 solar wind parameters and careful model configuration, the pairwise network trains in approximately the same amount of time as the corresponding MLP if no pruning is used. Iterative pruning increases the training time proportional to the number of pruning iterations and fine-tuning steps. An increase in the data set size would scale the computation time of the pairwise network with the same measure as the MLP, but (as discussed in Section 5.6) increasing the number of features would quickly grow the pairwise network to a point where it becomes unwieldy.

Chapter 7

Conclusion

In this chapter we review the key findings of this dissertation, discuss the implications thereof and suggest future research directions.

7.1 Introduction

The goal of this study (as discussed in Chapter 1) was to develop an intrinsically interpretable artificial neural network (ANN) that enables feature ranking without significant sacrifices in performance and then apply it to a popular space weather regression problem. In this chapter we examine the degree to which our original objectives were met. We discuss the key findings of the study, the implications of these findings and future research directions.

7.2 Key findings

In this section we discuss the key findings of this study and the implications thereof. We start with the observations made during the development of an intrinsically interpretable ANN using synthetic data:

- Structural adaptions to an MLP can transform it into a *pairwise network*, an ANN that enables feature ranking without significant sacrifices in performance. The pairwise network groups input features into pairs of two and assigns a sub-network to each pair. The output of the network is the weighted sum of all sub-networks.

- From the structure of the pairwise network, we introduced the *weighted summary node activation ratio* (ψ), which measures how much a pair contributed to the output of the network, relative to the others. Using this metric, the input features can be ranked according to decreasing ψ.

- We used synthetic data sets to validate the training and ranking procedure of the pairwise network. It was shown that the pairwise network alone is able to achieve near-perfect performance on these tasks, but requires iterative pruning during training to have an accurate ranking capability. With iterative pruning, the pairwise network is able to identify features that do not form part of data generation, at the cost of slightly worse performance.

- Several pruning amounts and two fine-tuning strategies were tested. We found that, for the task studied, pruning 10% of remaining parameters and then fine-tuning for a fixed number of epochs each iteration, yields pairwise networks with a good ranking capability and reduced training time per iteration (opposed to fine-tuning to completion per iteration).

- We tested two types of initialisation schemes: initialising the entire network with random weight values and initialising the network so that each sub-network has identical weight values. However, we found no significant difference between the two schemes.

- Finally, we demonstrated how the correlation between the output of a sub-network and its input features can be used to reveal cases where a sub-network only models one feature and disregards the other.

- The proposed process has certain limitations: the pruning process is computationally expensive since the network has to be retrained at every pruning iteration. Also, the architecture itself is limited to smaller tasks: since every combination of input pairs receives a sub-network the model grows quickly, becoming unwieldy.

We trained several multilayer perceptrons (MLPs) and pairwise networks on the solar wind/SYM-H data set that we created for this study and found the following:

- Both pairwise networks and MLPs are viable options for predicting the SYM-H index from solar wind parameters. Without iterative pruning, the best pairwise network achieved a root mean squared error (RMSE) of 20.46 nT and a correlation of 0.87 on the evaluation set, which is better than the best MLP baseline, with a RMSE of 22.52 nT and a correlation of 0.85. Pruning slightly decreased performance of the pairwise network to a RMSE of 20.78 nT and a correlation of 0.86, but it still performed better than the MLP baselines.

- Temporal and phase information resulted in a significant performance increase and using them together leads to a better performance than the individual elements for the pairwise networks, but not for the MLPs.

- The rankings extracted from the pairwise networks correspond to our understanding of space physics, albeit with some variance between initialisation seeds:

  - Without time-shifted or phase inputs the pairwise network correctly separated important parameters from those less important to the space weather regression problem.

  - Duplicated information due to parameters representing similar physical parameters were not reflected in the ranking.

  - Extracting rankings for the three storm phases in isolation showed that parameter importance vary with storm progression in a manner that agrees with the physics.
  - Adding phase and time-shifted inputs lead to similar rankings as without it, but with more variance between seeds and with some networks that produced a poor ranking. However, the differences in ranking between current and historical values of solar wind parameters reflect our understanding of the time dependence of the various processes at play.

7.3 Contributions

In this work we introduced the pairwise network, a novel neural network that implements a form of intrinsic interpretability through layout constraints to allow an, admittedly crude, way of feature ranking.

Revisiting the well-known solar wind/SYM-H regression problem, we showed that:

- The architectural modifications to the pairwise network do not decrease performance when compared to an MLP trained on this task. In fact, on our data set the pairwise network performs slightly better than the MLP.
- The training procedure followed to produce networks with good ranking ability might decrease the performance, but these networks still performed better than the MLPs.
- Adding storm phase and time-shifted solar wind parameters increases model performance, as would be expected given the current understanding of the problem.
- The rankings produced by this pairwise network generally agrees with our current understanding of solar wind – magnetosphere coupling during geomagnetic storms.

The following artifacts that we developed for this study will be publicly released:

- A curated data set with solar wind parameters as input and SYM-H as output.
- A Python package for downloading, generating and preprocessing the solar wind/SYM-H data set[1]. This repository also contains code for the analysis done on this data set.
- The source code for the training and analysis of the pairwise network[2]. This repository is structured as a Python package to allow for effortless importing of the pairwise network in future work.

7.4 Future work

Further development of the pairwise network will concentrate on improving ranking and computational tractability. We hope to address the following in future work:

- Optimal pruning strategies have not yet been investigated. It is of particular interest to develop a method that produces sparse networks early in training since iteratively pruning is computationally expensive. A possible solution could include ending every fine-tuning iteration once the network reached a certain performance level.
- Alternative methods for enforcing sparsely-trained networks can be considered. For instance, one can add a term to the loss function that ensures sparsity in certain layers of the network. This approach will decrease training time since no pruning iterations are required.
- Existing attribution methods could be added to the ranking procedure. For example, DeepLIFT [50] or Integrated Gradients [49] can be used to get an estimate for the importance of each input parameter to a sub-network.
- In this work we base the pairwise network on the MLP for its relative simplicity with regards to both implementation and analysis. Architectures that are more suitable for sequence modelling, such as recurrent or convolutional neural networks, can be incorporated in future versions of the pairwise network.

[1] `https://bitbucket.org/must_research/sansa`
[2] `https://bitbucket.org/must_research/pairwise_network`

- A more rigorous analysis on a broader range of synthetic data sets with variable parameter importance are required, but eventually the development will shift to the application of these ideas to other, less well-understood tasks.
- A shortcoming of the current pairwise network is that its use is limited to smaller tasks. Since every combination of input pairs receives a sub-network, the model quickly becomes unwieldy. Methods to reduce the number of input pairs before training can be investigated.
- In more exploratory work, we would also like to measure the pairwise ranking for various sets of samples (natural sample clusters formed by the network itself [82]) to see if there is a difference in behaviour among the sets.

7.5 Conclusion

In the current age of rapidly increasing machine learning capability, researchers and practitioners are becoming more cognisant of the dangers of well-performing but unexplainable models. To this end, we illustrated how careful model design can inform domain knowledge by developing a novel neural network that implements a form of intrinsic interpretability through layout constraints to allow a way of feature ranking. We showed that this network marginally outperforms an MLP on a popular space weather task while providing feature rankings that correspond with our understanding of the problem. We hope that a future version of this network will eventually be used to aid in scientific discovery by reliably extracting information from its task.

Bibliography

[1] M. Bojarski, P. Yeres, A. Choromanska, *et al.*, "Explaining how a deep neural network trained with end-to-end learning steers a car," 2017. arXiv: 1704.07911.

[2] R. Caruana, Y. Lou, J. Gehrke, P. Koch, M. Sturm, and N. Elhadad, "Intelligible models for healthcare: Predicting pneumonia risk and hospital 30-day readmission," in *Proceedings of the ACM SIGKDD*, vol. 2015-August, 2015, pp. 1721–1730, ISBN: 9781450336642. DOI: 10.1145/2783258.2788613.

[3] Z. Che, S. Purushotham, R. Khemani, and Y. Liu, "Distilling knowledge from deep networks with applications to healthcare domain," 2015. arXiv: 1512.03542.

[4] M. Raghu and E. Schmidt, "A survey of deep learning for scientific discovery," 2020. arXiv: 2003.11755.

[5] A. Adadi and M. Berrada, "Peeking inside the black-box: A survey on explainable artificial intelligence (XAI)," *IEEE Access*, vol. 6, pp. 52 138–52 160, 2018. DOI: 10.1109/ACCESS.2018.2870052.

[6] Y. Béniguel and P. Hamel, "A global ionosphere scintillation propagation model for equatorial regions," *Journal of Space Weather and Space Climate (JSWSC)*, vol. 1, no. 1, A04, 2011. DOI: 10.1051/swsc/2011004.

[7] N. A. Frissell, J. S. Vega, E. Markowitz, *et al.*, "High-frequency communications response to solar activity in September 2017 as observed by amateur radio networks," *Space Weather*, vol. 17, no. 1, pp. 118–132, 2019. DOI: 10.1029/2018SW002008.

[8] D. N. Baker, "The occurrence of operational anomalies in spacecraft and their relationship to space weather," *IEEE Transactions on Plasma Science*, vol. 28, no. 6, pp. 2007–2016, 2000. DOI: 10.1109/27.902228.

[9] D. H. Boteler, "Assessment of geomagnetic hazard to power systems in Canada," *Natural Hazards*, vol. 23, no. 2, pp. 101–120, 2001. DOI: 10.1023/A:1011194414259.

[10] L. Trichtchenko and D. H. Boteler, "Modelling of geomagnetic induction in pipelines," *Annales Geophysicae*, vol. 20, no. 7, pp. 1063–1072, 2002. DOI: 10.5194/angeo-20-1063-2002.

[11] E. J. Oughton, M. Hapgood, G. S. Richardson, *et al.*, "A risk assessment framework for the socioeconomic impacts of electricity transmission infrastructure failure due to space weather: An application to the United Kingdom," *Risk Analysis*, vol. 39, no. 5, pp. 1022–1043, 2019. DOI: 10.1111/risa.13229.

[12] W. D. Gonzalez, J. A. Joselyn, Y. Kamide, *et al.*, "What is a geomagnetic storm?" *Journal of Geophysical Research*, vol. 99, no. A4, p. 5771, 1994. DOI: 10.1029/93ja02867.

[13] H. Lundstedt and P. Wintoft, "Prediction of geomagnetic storms from solar wind data with the use of a neural network," *Annales Geophysicae*, vol. 12, no. 1, pp. 19–24, 1994. DOI: 10.1007/s00585-994-0019-2.

[14] H. Gleisner, H. Lundstedt, and P. Wintoft, "Predicting geomagnetic storms from solar-wind data using time-delay neural networks," *Annales Geophysicae*, vol. 14, no. 7, pp. 679–686, 1996. DOI: 10.1007/s00585-996-0679-1.

[15] S. Kugblenu, S. Taguchi, and T. Okuzawa, "Prediction of the geomagnetic storm associated Dst index using an artificial neural network algorithm," *Earth, Planets and Space*, vol. 51, no. 4, pp. 307–313, 1999. DOI: 10.1186/BF03352234.

[16] R. Bala and P. Reiff, "Improvements in short-term forecasting of geomagnetic activity," *Space Weather*, vol. 10, no. 6, p. 6001, 2012. DOI: 10.1029/2012SW000779.

[17] H. Lundstedt, H. Gleisner, and P. Wintoft, "Operational forecasts of the geomagnetic Dst index," *Geophysical Research Letters*, vol. 29, no. 24, pp. 34-1-34–4, 2002. DOI: 10.1029/2002gl016151.

[18] J. G. Wu, "Geomagnetic storm predictions from solar wind data with the use of dynamic neural networks," *Journal of Geophysical Research A: Space Physics*, vol. 102, no. A7, pp. 14 255–14 268, 1997. DOI: 10.1029/97JA00975.

[19] S. Watanabe, E. Sagawa, K. Ohtaka, and H. Shimazu, "Prediction of the Dst index from solar wind parameters by a neural network method," *Earth, Planets and Space*, vol. 54, no. 12, pp. 1263–1275, 2002. DOI: 10.1186/bf03352454.

[20] M. A. Gruet, M. Chandorkar, A. Sicard, and E. Camporeale, "Multiple-hour-ahead forecast of the Dst index using a combination of long short-term memory neural network and Gaussian process," *Space Weather*, vol. 16, no. 11, pp. 1882–1896, 2018. DOI: 10.1029/2018SW001898.

[21] S. Lotz, J. P. Beukes, and M. H. Davel, "A neural network based method for input parameter selection," in *Helio ML*, 2019.

[22] S. Lotz, J. P. Beukes, and M. H. Davel, "Input parameter ranking for neural networks in a space weather regression problem," in *Proceedings of the South African Forum for Artificial Intelligence Research (FAIR)*, Cape Town, South Africa, 2019, pp. 133–144. [Online]. Available: http://ceur-ws.org/Vol-2540/FAIR2019_paper_50.pdf.

[23] J. P. Beukes, S. Lotz, and M. H. Davel, "Pairwise networks for feature ranking of a geomagnetic storm model," *South African Computer Journal*, 2020. DOI: 10.18489/sacj.v32i2.860.

[24] M. Moldwin, *An introduction to space weather*. Cambridge: Cambridge University Press, 2008, pp. 1–142. DOI: 10.1017/CBO9780511801365.

[25] T. Iyemori, M. Takeda, M. Nose, Y. Odagi, and H. Toh, "Mid-latitude geomagnetic indices ASY and SYM for 2009 (provisional)," *Data Analysis Center for Geomagnetism and Space Magnetism, Graduate School of Science, Kyoto University, Japan*, 2010. [Online]. Available: http://wdc.kugi.kyoto-u.ac.jp/aeasy/asy.pdf.

[26] C. E. Navia, M. N. de Oliveira, and C. R. A. Augusto, "The highest geomagnetic storms of the solar cycle observed at ground level," in *Extreme Weather*, P. J. Sallis, Ed., Rijeka: IntechOpen, 2018, ch. 3. DOI: 10.5772/intechopen.75688.

[27] H. E. Koskinen and E. I. Tanskanen, "Magnetospheric energy budget and the epsilon parameter," *Journal of Geophysical Research: Space Physics*, vol. 107, no. A11, 2002. DOI: 10.1029/2002JA009283.

[28] J. A. Wanliss and K. M. Showalter, "High-resolution global storm index: Dst versus SYM-H," *Journal of Geophysical Research: Space Physics*, vol. 111, no. 2, pp. 1–10, 2006. DOI: 10.1029/2005JA011034.

[29] S. B. Xu, S. Y. Huang, Z. G. Yuan, X. H. Deng, and K. Jiang, "Prediction of the Dst index with bagging ensemble-learning algorithm," *The Astrophysical Journal Supplement Series*, vol. 248, no. 1, p. 14, 2020. DOI: 10.3847/1538-4365/ab880e.

[30] L. Cai, S. Y. Ma, and Y. L. Zhou, "Prediction of SYM-H index during large storms by NARX neural network from IMF and solar wind data," *Annales Geophysicae*, vol. 28, no. 2, pp. 381–393, 2010. DOI: 10.5194/angeo-28-381-2010.

[31] A. Bhaskar and G. Vichare, "Forecasting of SYMH and ASYH indices for geomagnetic storms of solar cycle 24 including St. Patrick's day, 2015 storm using NARX neural network," *Journal of Space Weather and Space Climate*, vol. 9, 2019. DOI: 10.1051/swsc/2019007.

[32] F. Siciliano, G. Consolini, R. Tozzi, M. Gentili, and F. Giannattasio, "Forecasting SYM-H index : A comparison between long short-term memory and convolutional neural networks," *Space Weather*, 2020. DOI: 10.1029/2020SW002589.

[33] I. Goodfellow, Y. Bengio, and A. Courville, *Deep Learning*. MIT Press, 2016. [Online]. Available: http://www.deeplearningbook.org.

[34] V. Nair and G. E. Hinton, "Rectified linear units improve Restricted Boltzmann machines," in *ICML*, 2010, pp. 807–814, ISBN: 9781605589077. [Online]. Available: https://icml.cc/Conferences/2010/papers/432.pdf.

[35] D. A. Clevert, T. Unterthiner, and S. Hochreiter, "Fast and accurate deep network learning by exponential linear units (ELUs)," *ICLR*, pp. 1–14, 2016. arXiv: 1511.07289.

[36] A. L. Maas, A. Y. Hannun, and A. Y. Ng, "Rectifier nonlinearities improve neural network acoustic models," *in ICML Workshop on Deep Learning for Audio, Speech and Language Processing*, vol. 28, 2013.

[37] X. Glorot, A. Bordes, and Y. Bengio, "Deep sparse rectifier neural networks," in *Journal of Machine Learning Research*, vol. 15, Fort Lauderdale, FL, USA: JMLR Workshop and Conference Proceedings, 2011, pp. 315–323. [Online]. Available: http://proceedings.mlr.press/v15/glorot11a.html.

[38] T. Szandała, "Review and comparison of commonly used activation functions for deep neural networks," in *Bio-inspired Neurocomputing*, Singapore: Springer, 2021, pp. 203–224, ISBN: 978-981-15-5495-7. DOI: 10.1007/978-981-15-5495-7_11.

[39] C. E. Nwankpa, W. Ijomah, A. Gachagan, and S. Marshall, "Activation functions: Comparison of trends in practice and research for deep learning," 2018. arXiv: 1811.03378.

[40] A. Botchkarev, "A new typology design of performance metrics to measure errors in machine learning regression algorithms," *Interdisciplinary Journal of Information, Knowledge, and Management*, vol. 14, pp. 45–76, 2019. DOI: 10.28945/4184.

[41] D. P. Kingma and J. L. Ba, "Adam: A method for stochastic optimization," in *ICLR*, 2015. arXiv: 1412.6980v9.

[42] K. He, X. Zhang, S. Ren, and J. Sun, "Delving deep into rectifiers: Surpassing human-level performance on imagenet classification," in *Proceedings of the IEEE International Conference on Computer Vision*, 2015, pp. 1026–1034, ISBN: 9781467383912. DOI: 10.1109/ICCV.2015.123.

[43] K. T. Schütt, M. Gastegger, A. Tkatchenko, and K.-R. Müller, "Quantum-chemical insights from interpretable atomistic neural networks," in *Explainable AI: Interpreting, Explaining and Visualizing Deep Learning*, Cham: Springer International Publishing, 2019, pp. 311–330, ISBN: 978-3-030-28954-6. DOI: 10.1007/978-3-030-28954-6_17.

[44] K. Preuer, G. Klambauer, F. Rippmann, S. Hochreiter, and T. Unterthiner, "Interpretable deep learning in drug discovery," in *Explainable AI: Interpreting, Explaining and Visualizing Deep Learning*, Cham: Springer International Publishing, 2019, pp. 331–345, ISBN: 978-3-030-28954-6. DOI: 10.1007/978-3-030-28954-6_18.

[45] C. Rudin, "Stop explaining black box machine learning models for high stakes decisions and use interpretable models instead," *Nature Machine Intelligence*, vol. 1, no. 5, pp. 206–215, 2019. DOI: 10.1038/s42256-019-0048-x.

[46] M. D. Zeiler and R. Fergus, "Visualizing and understanding convolutional networks," in *Computer Vision – ECCV 2014*, Springer International Publishing, 2014, pp. 818–833, ISBN: 978-3-319-10590-1. DOI: 10.1007/978-3-319-10590-1_53.

[47] L. M. Zintgraf, T. S. Cohen, T. Adel, and M. Welling, "Visualizing deep neural network decisions: Prediction difference analysis," in *ICLR – Conference Track Proceedings*, 2017. arXiv: 1702.04595.

[48] M. Ancona, E. Ceolini, C. Öztireli, and M. Gross, "Towards better understanding of gradient-based attribution methods for deep neural networks," in *ICLR*, 2018. arXiv: 1711.06104.

[49] M. Sundararajan, A. Taly, and Q. Yan, "Axiomatic attribution for deep networks," in *ICML*, vol. 70, 2017, pp. 3319–3328. arXiv: 1703.01365.

[50] A. Shrikumar, P. Greenside, and A. Kundaje, "Learning important features through propagating activation differences," in *ICML*, vol. 70, Sydney, NSW, Australia, 2017, pp. 3145–3153. arXiv: 1704.02685.

[51] S. Bach, A. Binder, G. Montavon, F. Klauschen, K. R. Müller, and W. Samek, "On pixel-wise explanations for non-linear classifier decisions by layer-wise relevance propagation," *PLoS ONE*, vol. 10, no. 7, 2015. DOI: 10.1371/journal.pone.0130140.

[52] R. R. Selvaraju, M. Cogswell, A. Das, R. Vedantam, D. Parikh, and D. Batra, "Grad-CAM: Visual explanations from deep networks via gradient-Based localization," *International Journal of Computer Vision*, vol. 128, no. 2, pp. 336–359, 2019. DOI: 10.1007/s11263-019-01228-7.

[53] S. M. Lundberg and S. I. Lee, "A unified approach to interpreting model predictions," in *NeurIPS*, vol. 30, Curran Associates, Inc., 2017, pp. 4766–4775. arXiv: 1705.07874.

[54] L. Sixt, M. Granz, and T. Landgraf, "When explanations lie: Why many modified BP attributions fail," in *ICML*, vol. 119, PMLR, 2020, pp. 9046–9057. arXiv: 1912.09818.

[55] S. Hooker, D. Erhan, P. J. Kindermans, and B. Kim, "A benchmark for interpretability methods in deep neural networks," in *NeurIPS*, vol. 32, 2019, pp. 9737–9748. arXiv: 1806.10758.

[56] J. Adebayo, J. Gilmer, M. Muelly, I. Goodfellow, M. Hardt, and B. Kim, "Sanity checks for saliency maps," in *NeurIPS*, 2018, pp. 9505–9515. arXiv: 1810.03292.

[57] T. Hastie and R. Tibshirani, "Generalized additive models," *Statistical Science*, vol. 1, no. 3, pp. 297–310, 1986. DOI: 10.1214/ss/1177013604.

[58] J. H. Friedman and W. Stuetzle, "Projection pursuit regression," *Journal of the American Statistical Association*, vol. 76, no. 376, pp. 817–823, 1981. DOI: 10.1080/01621459.1981.10477729.

[59] W. J. E. Potts, "Generalized additive neural networks," in *Proceedings of the ACM SIGKDD*, New York, USA: ACM Press, 1999, pp. 194–200. DOI: 10.1145/312129.312228.

[60] R. Agarwal, N. Frosst, X. Zhang, R. Caruana, and G. E. Hinton, "Neural additive models: Interpretable machine learning with neural nets," *arXiv preprint*, 2020. arXiv: 2004.13912.

[61] A. Krizhevsky, "Convolutional deep belief networks on cifar-10," 2010.

[62] Z. Yang, A. Zhang, and A. Sudjianto, "GAMI-Net: An explainable neural network based on generalized additive models with structured interactions," 2020. arXiv: 2003.07132.

[63] Y. Lou, R. Caruana, J. Gehrke, and G. Hooker, "Accurate intelligible models with pairwise interactions," in *Proceedings of the ACM SIGKDD*, vol. Part F1288, 2013, pp. 623–631, ISBN: 9781450321747. DOI: 10.1145/2487575.2487579.

[64] M. Tsang, H. Liu, S. Purushotham, P. Murali, and Y. Liu, "Neural interaction transparency (NIT): Disentangling learned interactions for improved interpretability," in *NeurIPS*, Curran Associates, Inc., 2018, pp. 5804–5813. [Online]. Available: https://proceedings.neurips.cc/paper/2018/file/74378afe5e8b20910cf1f939e57f0480-Paper.pdf.

[65] J. Vaughan, A. Sudjianto, E. Brahimi, J. Chen, and V. N. Nair, "Explainable neural networks based on additive index models," 2018. arXiv: 1806.01933.

[66] J. Chen, J. Vaughan, V. Nair, and A. Sudjianto, "Adaptive explainable neural networks (Axnns)," *SSRN Electronic Journal*, 2020. DOI: 10.2139/ssrn.3569318.

[67] Z. Yang, A. Zhang, and A. Sudjianto, "Enhancing explainability of neural networks through architecture constraints," *IEEE Transactions on Neural Networks and Learning Systems*, 2020. DOI: 10.1109/tnnls.2020.3007259.

[68] Y. Huang, Y. Cheng, A. Bapna, *et al.*, "GPipe: Efficient training of giant neural networks using pipeline parallelism," in *NeurIPS*, vol. 32, 2019. arXiv: 1811.06965.

[69] S. Narang, G. Diamos, S. Sengupta, and E. Elsen, "Exploring sparsity in recurrent neural networks," in *ICLR*, 2017. arXiv: 1704.05119.

[70] K. Ullrich, M. Welling, and E. Meeds, "Soft weight-sharing for neural network compression," in *ICLR*, 2017. arXiv: 1702.04008.

[71] S. Han, J. Pool, J. Tran, and W. J. Dally, "Learning both weights and connections for efficient neural networks," in *NeurIPS*, 2015. arXiv: 1506.02626.

[72] D. Blalock, J. J. G. Ortiz, J. Frankle, and J. Guttag, "What is the state of neural network pruning?," 2020. arXiv: 2003.03033.

[73] A. Paszke, S. Gross, S. Chintala, *et al.*, *Automatic differentiation in pytorch*. 2017. [Online]. Available: https://openreview.net/forum?id=BJJsrmfCZ.

[74] *OMNIWeb Homepage*. [Online]. Available: https://omniweb.gsfc.nasa.gov/.

[75] S. K. Pandey and S. C. Dubey, "Characteristic features of large geomagnetic storms observed during solar cycle 23," *Indian Journal of Radio and Space Physics*, vol. 38, no. 6, pp. 305–312, 2009.

[76] S. I. Lotz and D. W. Danskin, "Extreme value analysis of induced geoelectric field in South Africa," *Space Weather*, vol. 15, no. 10, pp. 1347–1356, 2017. DOI: 10.1002/2017SW001662.

[77] E. C. Stone, A. M. Frandsen, and R. A. Mewaldt, "The advanced composition explorer," *Sensors (Peterborough, NH)*, vol. 86, no. 1/4, pp. 1–22, 1998. DOI: 10.1023/A:1005082526237.

[78] D. Vassiliadis, A. J. Klimas, J. A. Valdivia, and D. N. Baker, "The Dst geomagnetic response as a function of storm phase and amplitude and the solar wind electric field," *Journal of Geophysical Research: Space Physics*, vol. 104, no. A11, pp. 24 957–24 976, 1999. DOI: 10.1029/1999ja900185.

[79] D. L. Turner, V. Angelopoulos, W. Li, *et al.*, "Competing source and loss mechanisms due to wave-particle interactions in Earth's outer radiation belt during the 30 September to 3 October 2012 geomagnetic storm," *Journal of Geophysical Research: Space Physics*, vol. 119, no. 3, pp. 1960–1979, 2014. DOI: https://doi.org/10.1002/2014JA019770.

[80] P. T. Newell, T. Sotirelis, K. Liou, C. I. Meng, and F. J. Rich, "A nearly universal solar wind-magnetosphere coupling function inferred from 10 magnetospheric state variables," *Journal of Geophysical Research: Space Physics*, vol. 112, no. 1, 2007. DOI: 10.1029/2006JA012015.

[81] S. Kumar, B. Veenadhari, D. Chakrabarty, S. Tulasi Ram, T. Kikuchi, and Y. Miyoshi, "Effects of IMF By on ring current asymmetry under southward IMF Bz conditions observed at ground magnetic stations: case studies," *Journal of Geophysical Research: Space Physics*, vol. 125, no. 10, e2019JA027493, 2020. DOI: 10.1029/2019JA027493.

[82] D. G. Haasbroek and M. H. Davel, "Exploring neural network training dynamics through binary node activations," *Proceedings of the Southern African Conference for Artificial Intelligence Research*, 2020, Accepted for publication.

Appendix A

Supplemental Results

A.1 Appendix: Chapter 2

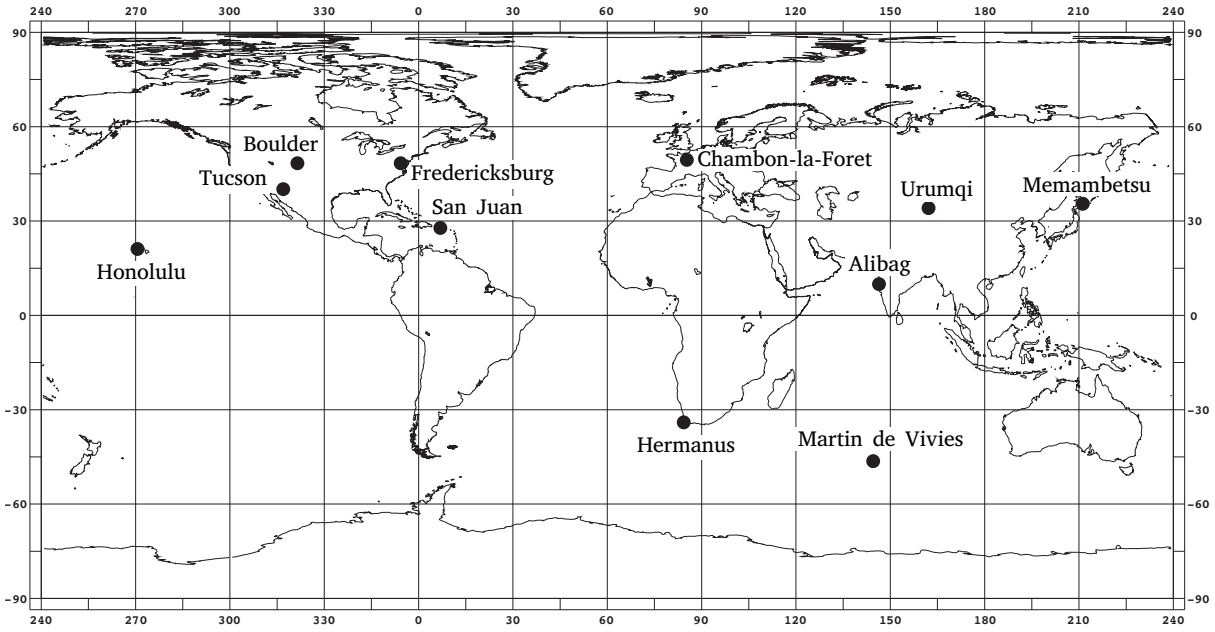


Figure A.1: Geographic distribution of the 11 geomagnetic observatories who's measurements are used to derive the SYM-H index. The diagram is adapted from [25].

A.2 Appendix: Chapter 5

Table A.1: The evaluation MSE, SE and computation time (in minutes) per pruning iteration for every synthetic data set and several pruning amounts (p), averaged across 4 seeds.

Data Set	p	Fine-tune to completion		Fine-tune for 100 epochs	
		Eval MSE (SE)	Min/Iteration	Eval MSE (SE)	Min/Iteration
y_1	10	0 (0)	65	0 (0)	51
	15	0 (0)	80	0 (0)	59
	20	0 (0)	80	0 (0)	59
	25	0 (0)	86	0 (0)	47
	30	0 (0)	94	0 (0)	57
y_2	10	0.0356 (0.0067)	97	0.0341 (0.0137)	54
	15	0.0254 (0.0051)	92	0.0306 (0.0120)	55
	20	0.0278 (0.0088)	104	0.0270 (0.0056)	63
	25	0.0234 (0.0035)	132	0.0397 (0.0120)	63
	30	0.0510 (0.0070)	112	0.0326 (0.0036)	63
a_1	10	0.0223 (0.0020)	1 125	0.0243 (0.0013)	405
	15	0.0225 (0.0021)	1 467	0.0235 (0.0021)	874
	20	0.0210 (0.0046)	2 412	0.0196 (0.0041)	1 092
	25	0.0281 (0.0057)	3 637	0.0252 (0.0060)	2 470
	30	0.0186 (0.0023)	2 058	0.0170 (0.0025)	2 043
a_2	10	0.0277 (0.0014)	718	0.0260 (0.0026)	133
	15	0.0213 (0.0053)	881	0.0239 (0.0015)	558
	20	0.0308 (0.0041)	850	0.0228 (0.0011)	543
	25	0.0244 (0.0020)	930	0.0274 (0.0038)	700
	30	0.0287 (0.0045)	841	0.0312 (0.0062)	692

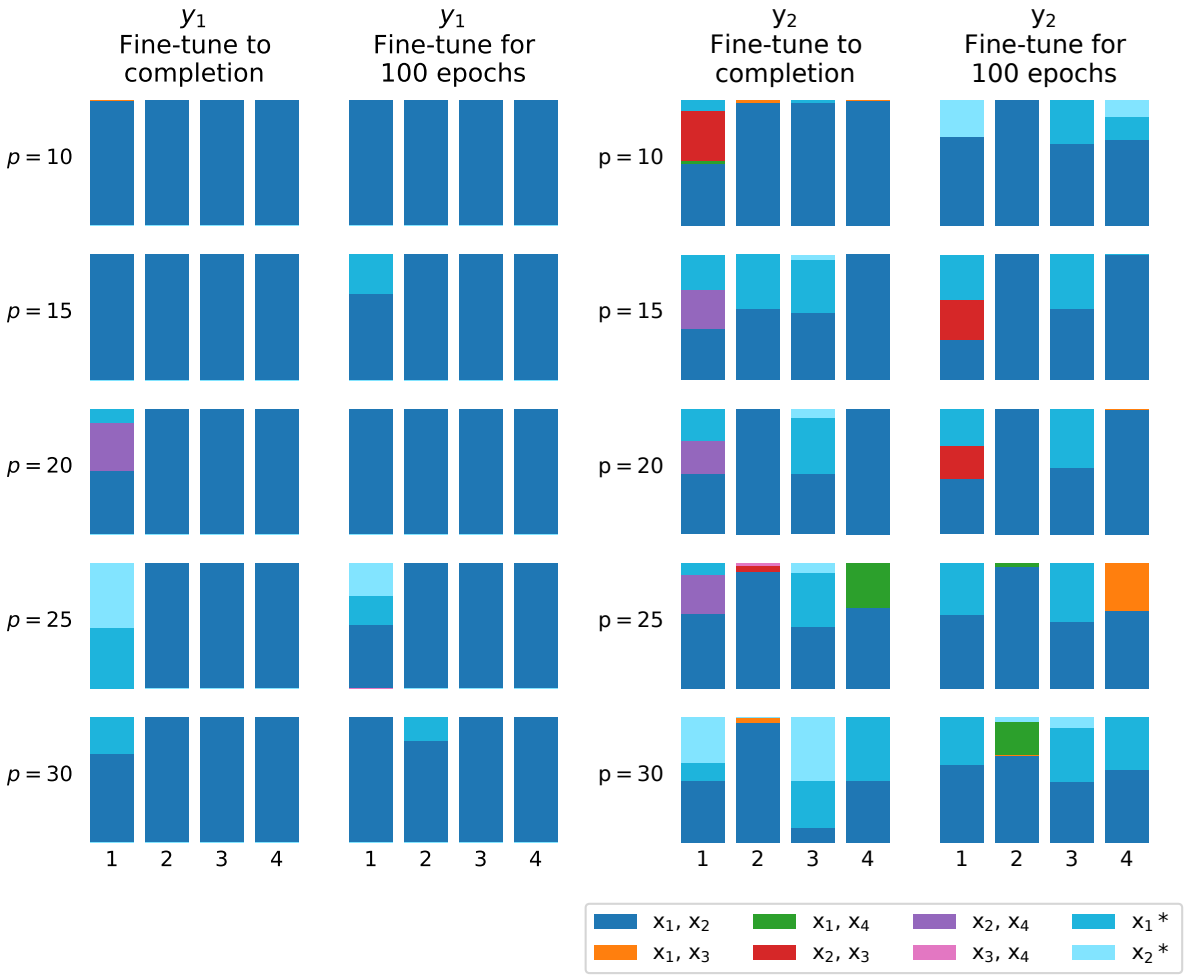


Figure A.2: ψ-distributions of synthetic data sets y_1 and y_2 for the two pruning variants and several pruning amounts (p).

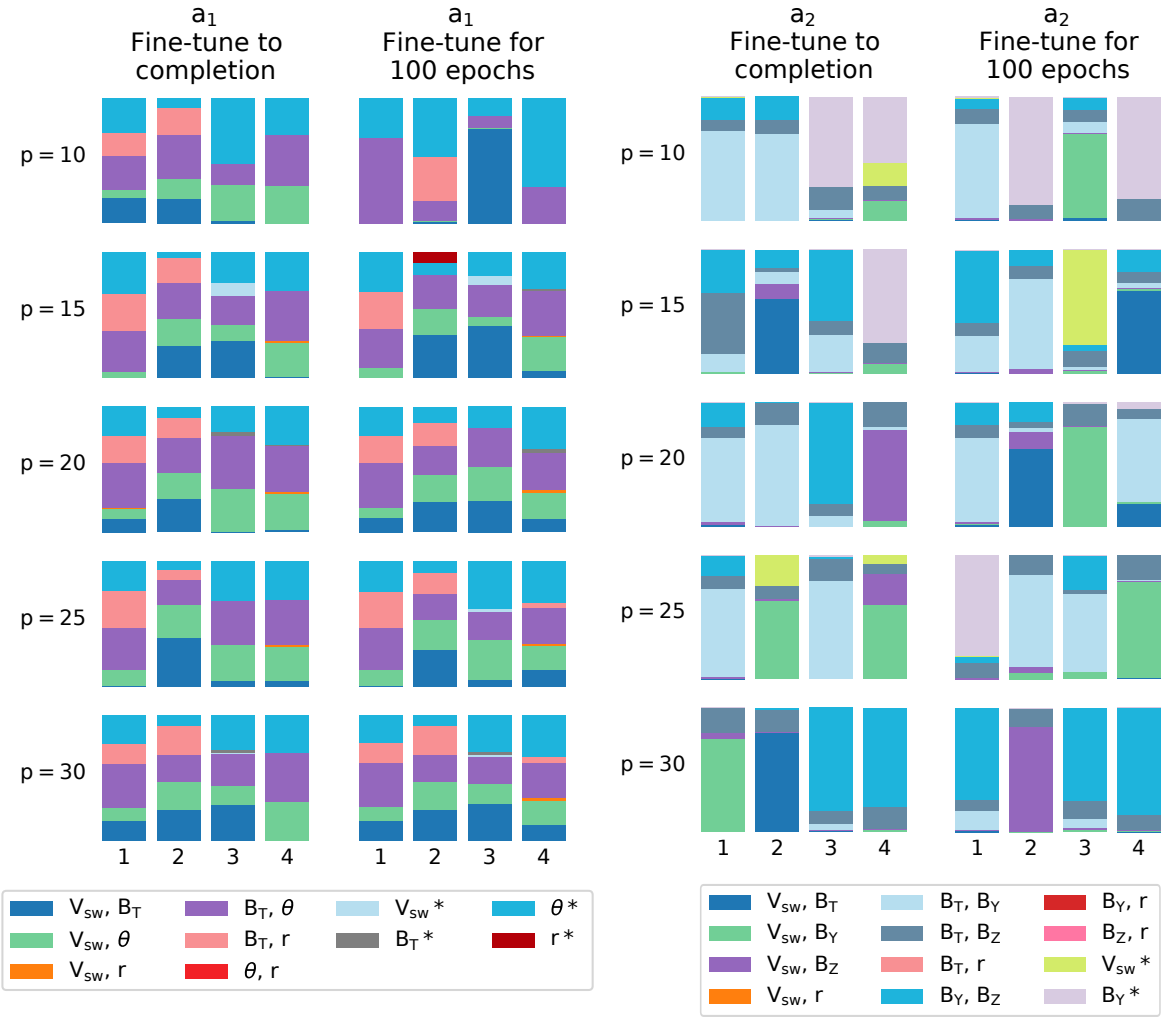
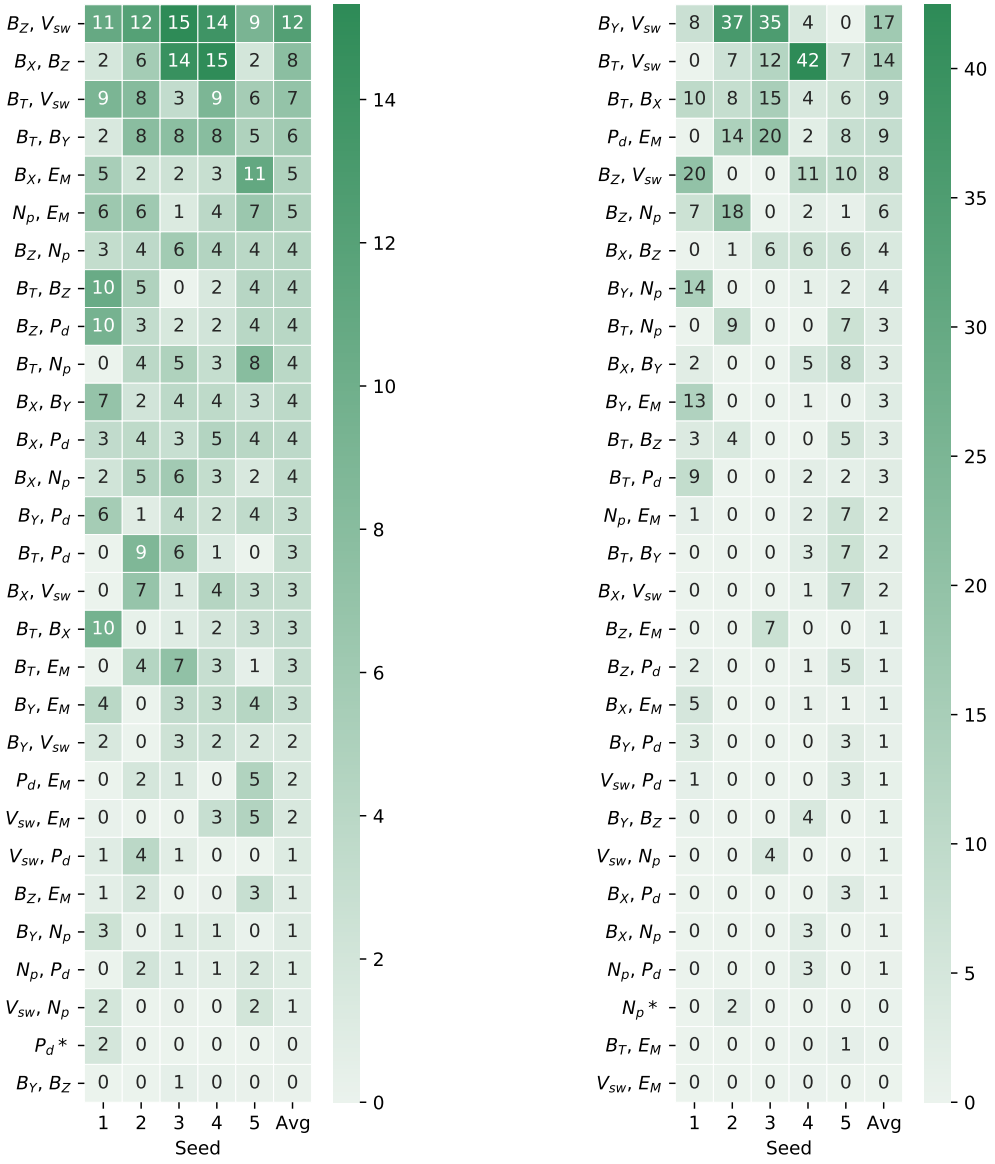


Figure A.3: ψ-distributions of synthetic data sets a_1 and a_2 for the two pruning variants and several pruning amounts (p).

A.3 Appendix: Chapter 6



(a) P_2 (pairwise with time-shifts) (b) P_3 (pairwise with phase inputs)

Figure A.4: ψ values extracted by pairwise networks P_2 and P_3 on the evaluation set of the solar wind/SYM-H data set for 5 seeds, as well as the average across all seeds. Sub-networks for which one of the input parameters were pruned away during training are renamed to the remaining input and marked with a *.

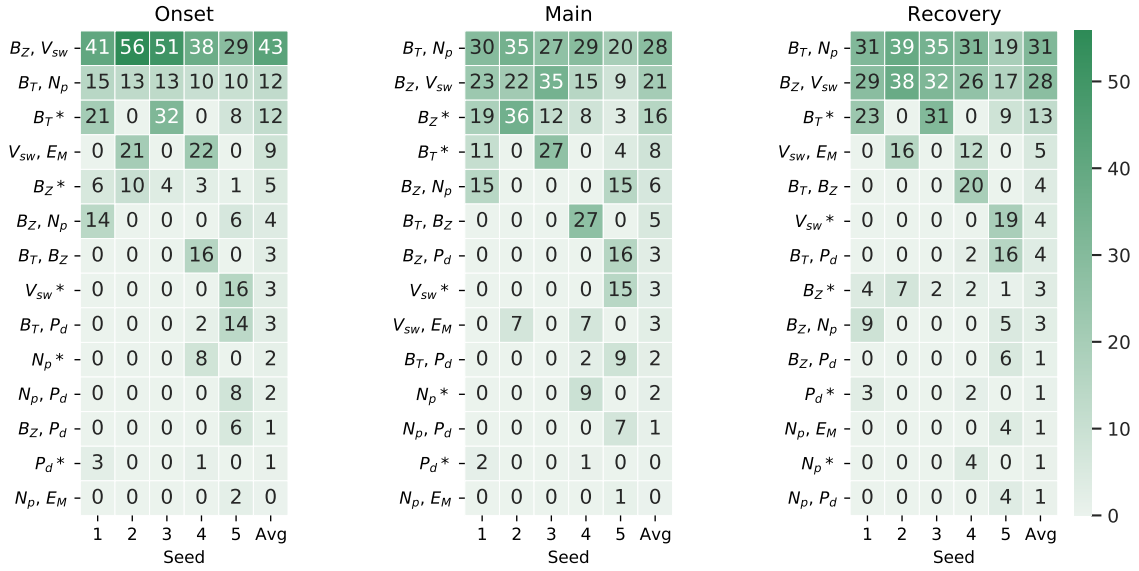


Figure A.5: ψ values extracted by pairwise network P_1 on the evaluation set of the solar wind/SYM-H data set for each phase separately. 5 seeds are shown, as well as the average across all seeds. Sub-networks for which one of the input parameters were pruned away during training are renamed to the remaining input and marked with a *.

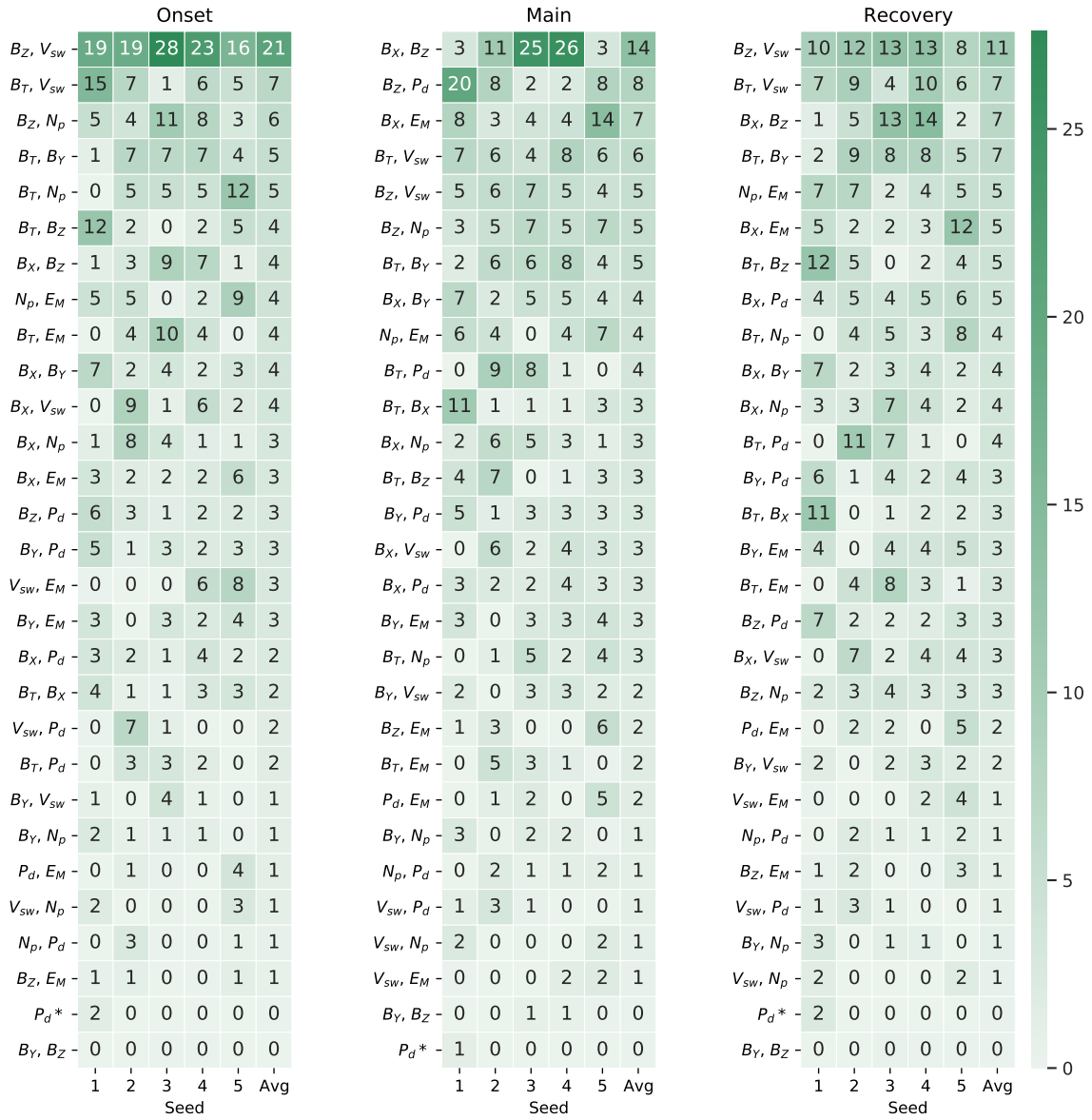


Figure A.6: ψ values extracted by pairwise network P_2 on the evaluation set of the solar wind/SYM-H data set for each phase separately. 5 seeds are shown, as well as the average across all seeds.

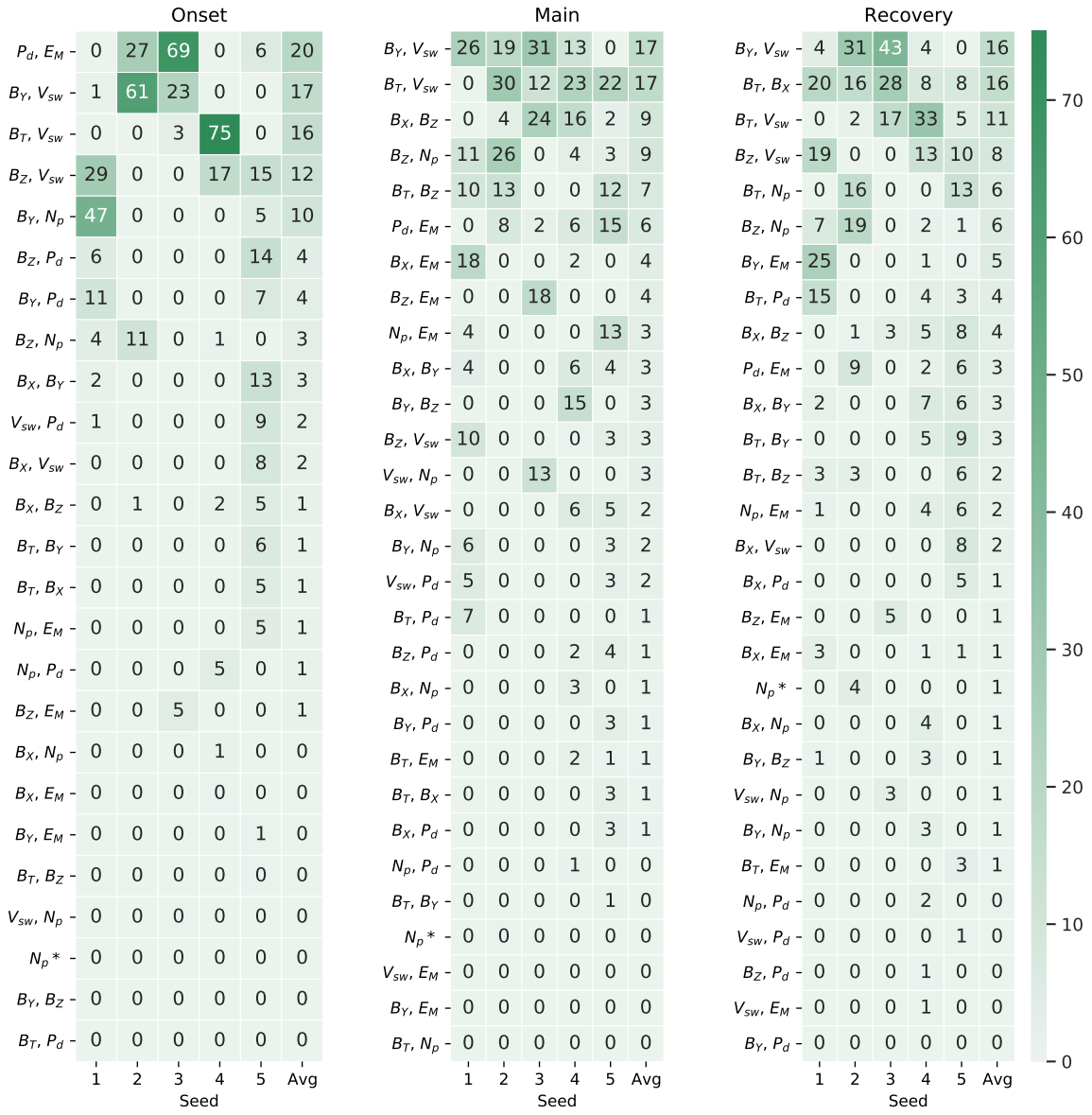


Figure A.7: ψ values extracted by pairwise network P_3 on the evaluation set of the solar wind/SYM-H data set for each phase separately. 5 seeds are shown, as well as the average across all seeds.

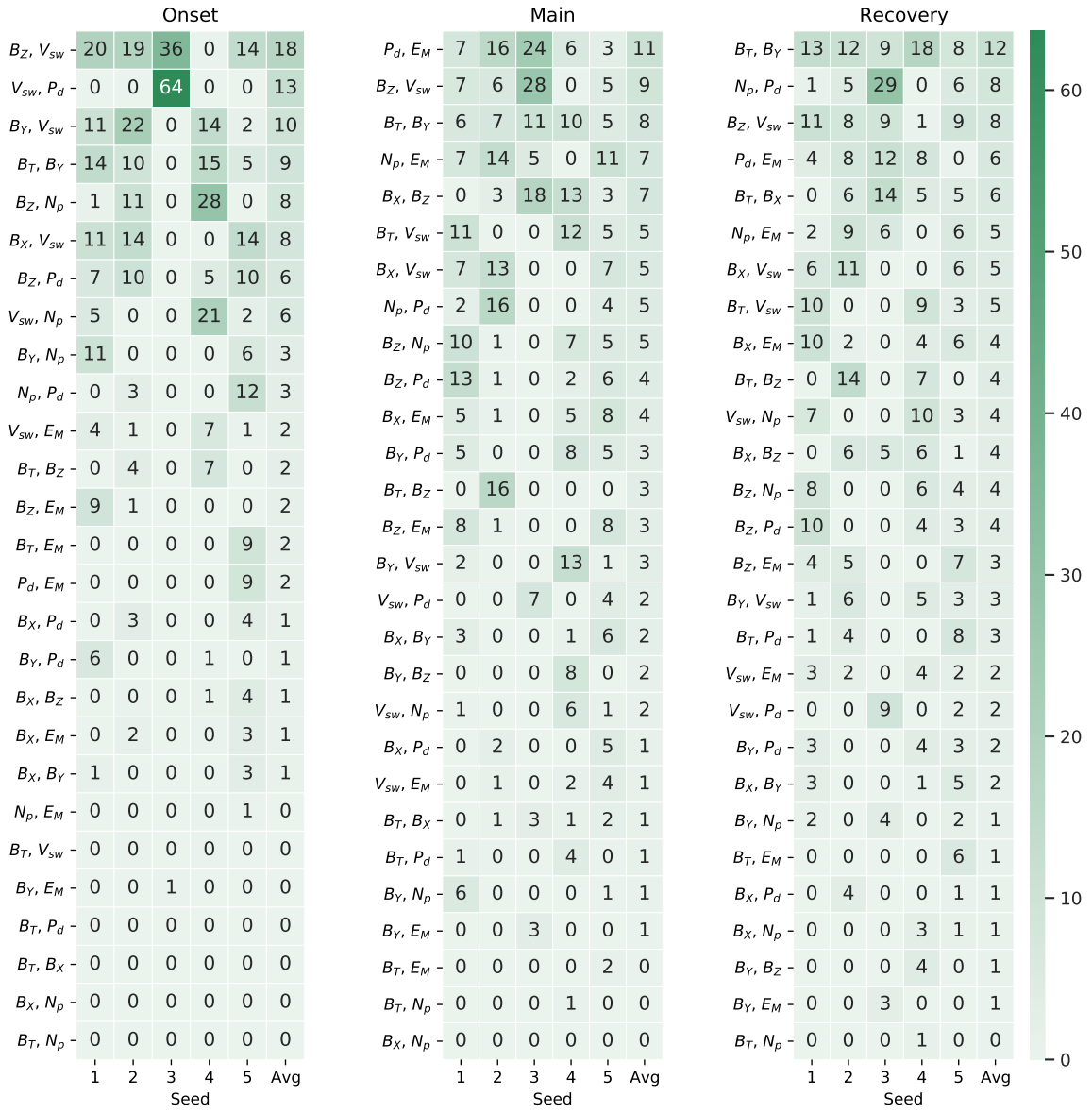


Figure A.8: ψ values extracted by pairwise network P_4 on the evaluation set of the solar wind/SYM-H data set for each phase separately. 5 seeds are shown, as well as the average across all seeds.

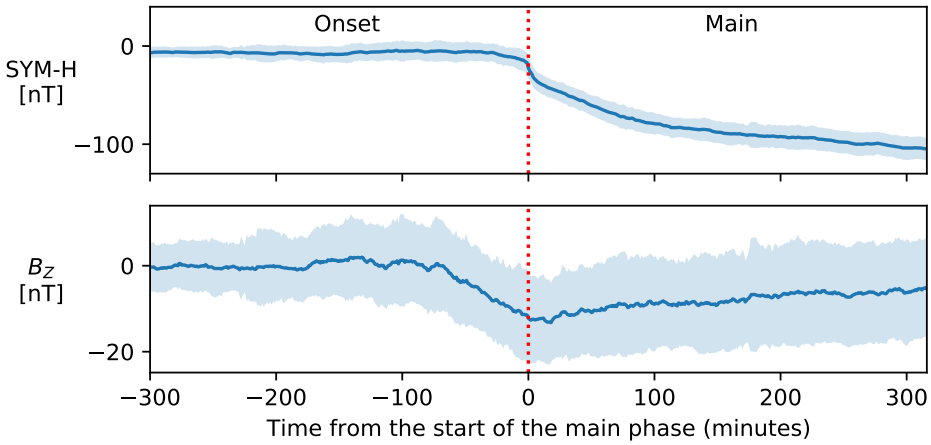
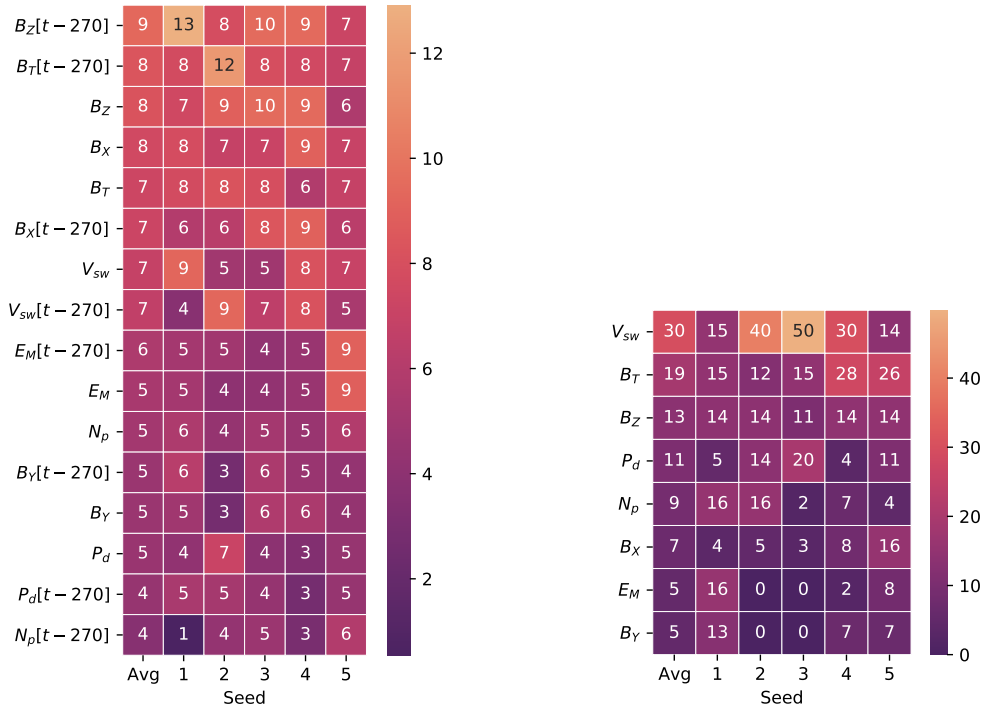


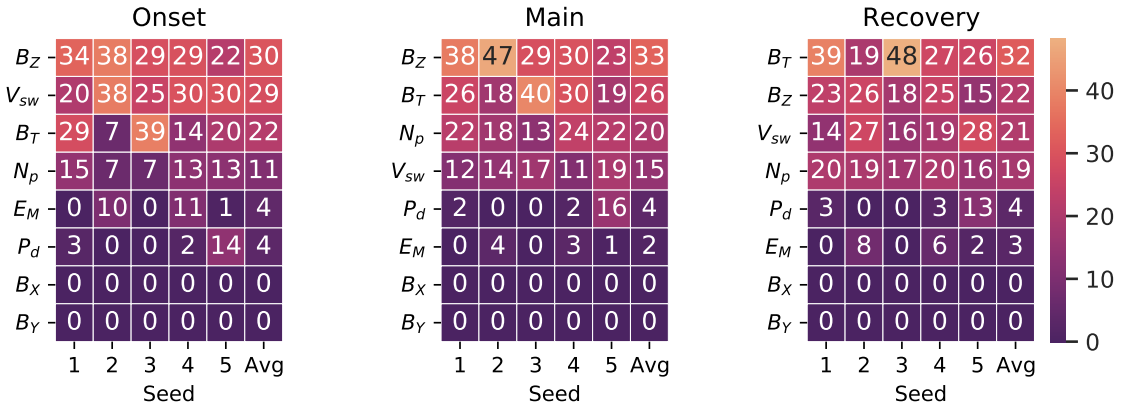
Figure A.9: B_Z and SYM-H during the onset and main phase when averaged across the entire solar wind/SYM-H data set. The shaded areas show one standard deviation above and below the mean and the dotted vertical lines indicate the start of the main phase.



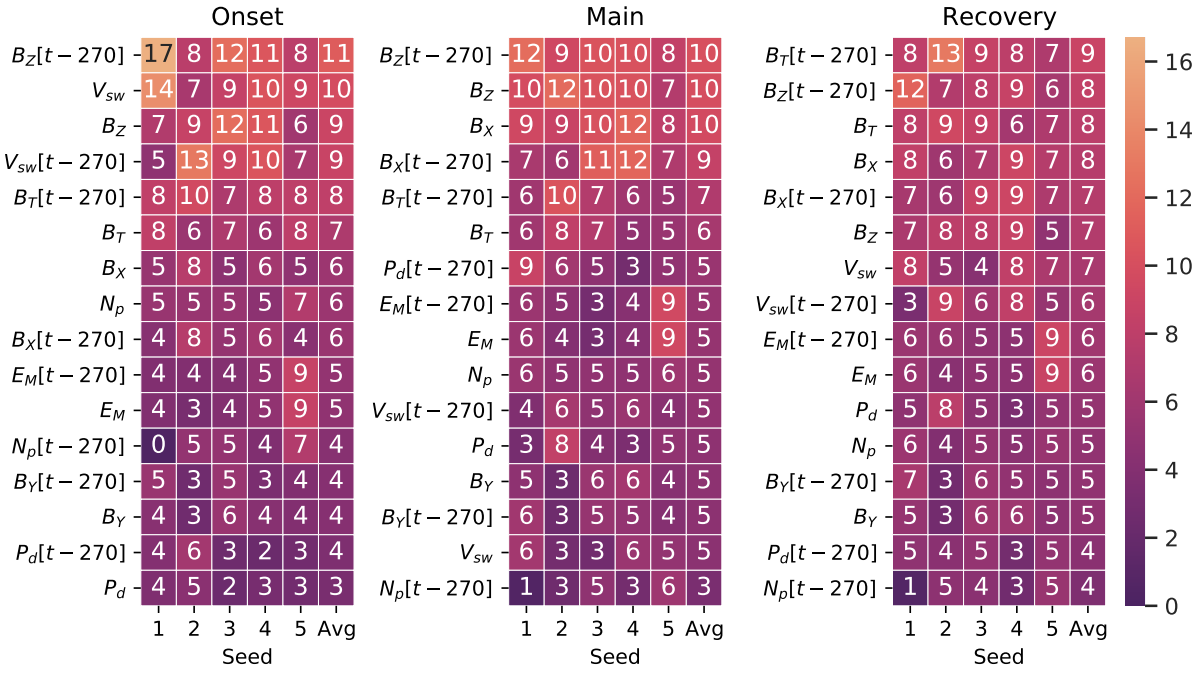
(a) P_2 (pairwise with time-shifts)

(b) P_3 (pairwise with phase inputs)

Figure A.10: Feature attribution values extracted by pairwise networks P_2 and P_3 on the evaluation set of the solar wind/SYM-H data set for 5 seeds, as well as the average across all seeds. The time-shifted version of parameter x is denoted as $x[t-270]$.

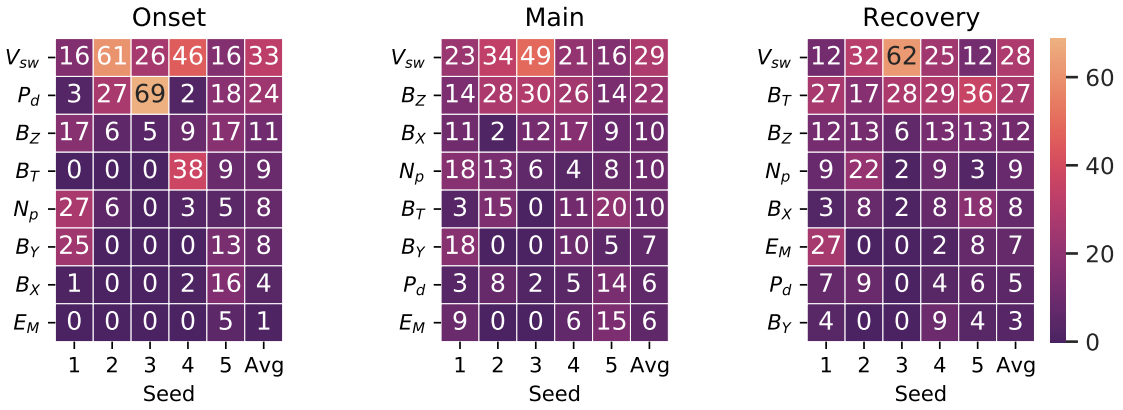


(a) P_1 (pairwise network without phase or time-shifted inputs)

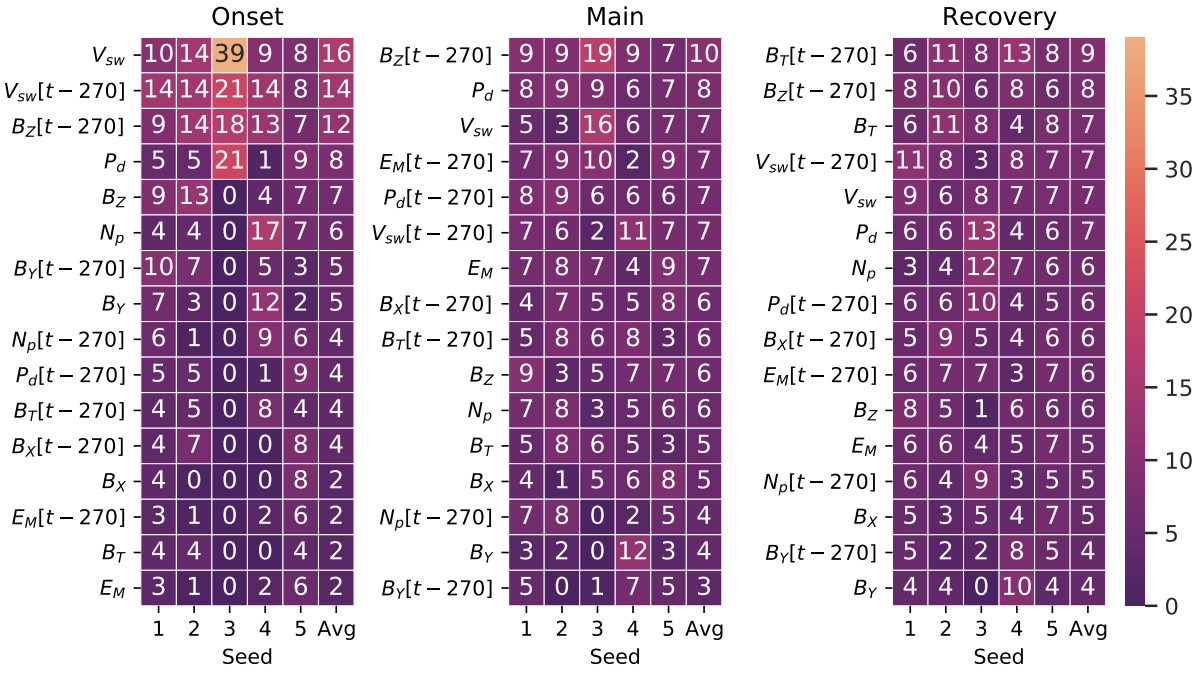


(b) P_2 (pairwise network with time-shifts)

Figure A.11: Feature attribution values produced by pairwise networks P_1 and P_2 across 5 initialisation seeds on the evaluation set of the solar wind/SYM-H data set for each phase separately. The average across all seeds are also shown.



(a) P_3 (pairwise network with time-shifts)



(b) P_4 (pairwise network with both phase and time-shifted inputs)

Figure A.12: Feature attribution values produced by pairwise networks P_3 and P_4 across 5 initialisation seeds on the evaluation set of the solar wind/SYM-H data set for each phase separately. The average across all seeds are also shown.