

Knowledge discovery with additive attribution methods for geomagnetic index prediction

C Grant



orcid.org/ 0000-0003-4861-9876

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 MH Davel

Co-supervisor: Dr SI Lotz

Graduation: October 2023

Student number: 29458447

Declaration

I, Chara Grant, hereby declare that the dissertation entitled “Knowledge discovery with additive attribution methods for geomagnetic index prediction” is my own original work and has not already been submitted to any other university or institution for examination.

Acknowledgements

This study was conducted under the leadership of the MUST Deep Learning research group at the North-West University (NWU) Potchefstroom Campus, and with the financial and academic support of both the South African National Space Agency (SANSA) and the MUST group. We acknowledge and thank the NWU for access to its High-Performance Computing (HPC) cluster, and MUST for the provision of its ‘Turmite’ server, which, without either platform, none of the compute-heavy tasks essential to this study could have been undertaken.

First and foremost, I owe thanks to Prof. Marelie Davel and Prof. Stefan Lotz, who were excellent academic supervisors, and taught me more than how to complete a master’s study. Ulrike, thank you for the consistent administrative support that made my experience with MUST enjoyable, and for your hard-working and diligent example that you set for all of us. To my personal friends and fellow students at MUST, thank you for the reliable supply of humour, encouragement, and companionship.

I would like to thank my family for being a consistent source of confidence, and especially for their patience during the long final months. To my parents in particular: thank you for teaching me to always do what is difficult, and to persevere in every good thing.

To Willem Venter, my best friend and my partner in life, love, and crime: thank you for the much-needed motivation and support. You are truly an example of perseverance, integrity, and kindness, and I am so excited to share this work with you.

Last but not least, I give thanks to Jesus Christ, the one by whom all things are, and the author and finisher of my faith. To paraphrase: The heart of man plans his way, but the LORD establishes his steps ... in Him we live and work and have our being.

*This dissertation is dedicated to **Gideon Johannes Kahl** (14/01/1945 - 28/05/2022). My very dear friend, mentor, and honorary parent who taught me how to think like a ‘real engineer’. Thank you for participating in making me who I am today. I miss you very much, and I hope this makes you proud.*

***For Granddad** (Hugh Grant, 11/06/1940 - 15/06/2022), who taught me that ‘the tallest trees get the most wind’.*

Abstract

During a solar cycle, the Sun emits various ejections such as Coronal Mass Ejections (CME) and higher-pressure streams caused by Corotating Interaction Regions (CIR). These ejections induce geomagnetic storms that are known to cause various measurable and often adverse effects. These effects are often modeled with a proxy such as a Geomagnetic Disturbance Index (GMD). One such GMD index is e_h , which is derived from the horizontal component of the locally measured ground-based geomagnetic field, and is a proxy index for induced electric field, and a direct driver of Geomagnetically Induced Currents (GIC). Solar physics has a strong reliance on empirical modeling, to which deep learning techniques such as Deep Neural Networks (DNNs) have been applied with moderate success. DNNs have great potential to accurately model these complex relationships due to their high capacity. However, DNNs are black boxes, and the correlations used by these models are nearly impossible to discern. With the advent of additive attribution methods such as DeepSHAP, it is now possible to analyse some of the underlying patterns and relationships used by these models in making their predictions. This study documents the use of a DNN model to predict the e_h index at two different latitudes (Abisko, Sweden and Hermanus, South Africa) using solar wind drivers as input and subsequent DeepSHAP attribution. We evaluate the feasibility of such a set of techniques as a knowledge discovery tool to be applied to problems with an abundance of data, but limited knowledge of the effective relationships between input and output drivers of that system. We trained a Multi-layer Perceptron (MLP) with time-shifted inputs to predict the e_h index individually at each latitude, simultaneously at both latitudes, and the explicit difference in e_h between the latitudes. These models were interpreted using DeepSHAP, and the resulting attributions were compared to the domain literature and known relationships between the solar wind drivers and e_h at these latitudes. We found that the DeepSHAP attributions, in the cases where the models accurately predicted e_h , consistently met the expectations set by the domain literature.

Keywords: *Deep neural networks, Additive attribution methods, Geomagnetic disturbance index, Knowledge discovery, DeepSHAP*

Contents

List of Figures	x
List of Tables	xv
List of Acronyms	xvii
1 Introduction	1
1.1 Background	1
1.2 Problem statement	3
1.3 Project scope	3
1.4 Research questions	4
1.5 Objectives	5
1.6 Research methodology	5
1.7 Dissertation structure	7
1.8 Publications	8
2 Background and literature review	9
2.1 Overview	9
2.2 Geomagnetic storms and GIC formation	10
2.2.1 The geomagnetic disturbance index	10
2.2.2 Drivers of geomagnetic storms	11

2.2.3	Geomagnetic storm modeling	13
2.3	Deep learning techniques	13
2.3.1	Feed-forward multi-layer perceptron	14
2.3.2	Time-series modeling	17
2.4	eXplainable AI: techniques and definitions	17
2.4.1	Local explanations: additive feature attribution techniques	19
2.4.2	Evaluating explanations	20
2.5	SHAP	22
2.5.1	Shapley values	23
2.5.2	Vanilla SHAP	24
2.5.3	DeepSHAP	26
2.6	Previous work	27
2.7	Chapter summary	29
3	Evaluation of DeepSHAP on a synthetic data set	30
3.1	Introduction	30
3.2	DeepSHAP	31
3.2.1	Background: DeepLIFT	31
3.2.2	DeepSHAP definition	33
3.2.3	The selection of reference values	35
3.3	Synthetic data set definition	35
3.4	Model selection and training	37
3.4.1	Model selection procedure	39
3.4.2	Model performance	40
3.5	DeepSHAP attributions	41
3.6	Discussion	43
3.7	Chapter summary	44

4	Data set definition	45
4.1	Introduction	45
4.2	Data set preparation	46
4.2.1	Source data	46
4.2.2	Initial pre-processing	48
4.2.3	Wintoft index calculation	49
4.2.4	Storm identification	50
4.3	Exploratory data analysis	53
4.3.1	Empirical distributions	54
4.3.2	Approximate distribution fits	56
4.3.3	Parameter correlations and comparisons	59
5	Time-shift multi-layer perceptron models	63
5.1	Introduction	63
5.2	Approach	64
5.3	Data set	65
5.3.1	Feature normalisation	65
5.3.2	Output target normalisation	65
5.3.3	Time-shift application	66
5.4	Model selection	67
5.5	Predictive performance and modelling results	69
5.6	Results when including auto-regressive input features	73
5.7	Chapter summary	76
6	DeepSHAP attributions and analysis	81
6.1	Introduction	81
6.2	Expectations	82

6.3	Experimental protocol	83
6.3.1	DeepSHAP review	83
6.3.2	Data sets	84
6.3.3	Reference data set selection	84
6.3.4	Attribution procedure	88
6.3.5	Attribution visualisation	88
6.3.6	Storm selection	90
6.4	Analysis and results	90
6.4.1	Aggregate results	91
6.4.2	A CME-driven storm	94
6.4.3	A CIR-driven storm	101
6.4.4	The effect of auto-regressive input features	106
6.5	Discussion	107
6.6	Chapter summary	110
7	Conclusion	112
7.1	Introduction	112
7.2	Key findings	113
7.3	Contributions	115
7.4	Future work	116
7.5	Conclusion	116
	References	118
A	Supplemental content	124
A.1	Appendix: Chapter 3	124
A.2	Appendix: Chapter 4	127
A.2.1	Dates of storms referenced in text	127

A.2.2	Interpolation procedure	127
A.2.3	Intensity-wise empirical histograms	130
A.2.4	Pairwise parameter 2-D histograms	131
A.3	Appendix: Chapter 5	140
A.3.1	SMAC characterisation	140
A.3.2	Optimisation results	143
A.3.3	Output plots	150
A.4	Appendix: Chapter 6	190
A.4.1	Reference set selection scheme comparisons	190
A.4.2	Supplemental CIR-driven storm attributions	200
A.4.3	Supplemental CME-driven storm attributions	214
A.4.4	Aggregate relative contribution distributions	250

List of Figures

2.1	Simplified view of the Earth's magnetosphere showing the solar wind encroaching from the left hand side. The L1 lagrange point, where the sattelites and space-crafts recording the solar wind paramaters are located, is indicated on the image. Adapted from Moldwin et al. [11]	12
2.2	Illustration of an MLP with two hidden layers and a single output node - showing flow of model parameters such as weights (w) and biases (b), and the application of activation functions to each hidden layer and the output node (σ).	15
3.1	Input feature values, 'weighted' input feature values, which are each input feature value as passed through Equation 3.15 , and the output values of the synthetic dataset. Each of the curves depict the weighted input feature values arranged from smallest input feature value to largest input feature value. The output value curve is arranged from smallest to largest output value.	37
3.2	Scenario plot for hyperparameter configuration selection on synthetic dataset. . .	40
3.3	Output values ordered by output value from smallest to largest vs. corresponding target values for train and evaluation sets.	41
3.4	Compilation of ordered input features and corresponding transformed input features vs their respective DeepSHAP attributions.	42
3.5	Compilation of input feature values, transformed input feature values, and each input feature's respective DeepSHAP attributions as sorted from smallest to largest according to target output value.	43
4.1	Geocentric Solar Ecliptic (GSE) coordinate system shown in the context of the solar wind, magnetospere, and Bow-Shock Nose (BSN). Adapted from Moldwin [11]	48
4.2	SYM-H, e_h , and $ d_h $ at Abisko (ABK) for a severe storm in 1996. e_h and $ d_h $ are calculated with a time window N of 30 minutes and a scaling factor a of 1. e_h is proportional to $ d_h $ [10].	50

4.3	SYM-H, B_z , and V_{sw} during a severe storm in 2015 with the onset, main, and recovery phases annotated. From Beukes et al. [4], used with permission.	51
4.4	SYM-H alongside B_z , V_{sw} and N_ρ for storm 1998_severe_006. The Sudden Storm Commencement (SSC) is the ‘spike’ in SYM-H at the beginning of the storm when the Coronal Mass Ejection (CME) hits the magnetosphere. One can notice the spikes in B_z , V_{sw} , and N_ρ coinciding with the SSC.	53
4.5	Empirical histograms of all input parameters and e_h at both latitudes, with all available data, and separately, all isolated storms, from 1996 to 2019. The relative frequency of samples is shown on the y-axis, and the value range is shown on the x-axis.	56
4.6	Theoretical Probability Density Function (PDF) vs. empirical distribution of each input parameter and e_h at both latitudes.	58
4.7	Correlation heatmap for the 10 input parameters and $e_{h_{HER}}$ and $e_{h_{ABK}}$	59
4.8	2-dimensional histograms of each input feature vs. $e_{h_{ABK}}$	60
4.9	2-dimensional histograms of each input feature vs. $e_{h_{HER}}$	61
5.2	Distribution of e_h at Hermanus (HER) and ABK after z-norm standardisation without centering at the mean, as well as $ \Delta e_h $ calculated with post-standardisation e_h values, for training and validation sets.	66
5.3	Best validation loss values for each time-shift size. The time-shift size in minutes is indicated on the x-axis, and the best validation Mean Squared Error (MSE) loss for the task at that time-shift is annotated at each point.	68
5.6	Target vs. predicted outputs of evaluation set storms with the strongest/best Pearson’s r coefficient values and corresponding best relative Root Mean Squared Error (RMSE) values.	71
5.7	Target vs. predicted outputs of evaluation set storms with the weakest/worst Pearson’s r coefficient values.	72
5.8	Target vs. predicted values of evaluation set storms with the strongest/best Pearson’s r coefficient values and a lag of -30.	75
5.1	Distributions of input features after z-norm standardisation for training and validation sets.	78
5.4	Scenario plot for $e_{h_{ABK}}$ with a time-shift of 30 minutes. The hyperparameter values and best validation loss value obtained at each run are shown as scattered points. Runs that strictly improve relative to previous runs are connected with a line that terminates at the final, best configuration.	79

5.5	Scenario plot for $e_{h_{HER}}$ with a time-shift of 30 minutes. The hyperparameter values and best validation loss value obtained at each run are shown as scattered points. Runs that strictly improve relative to previous runs are connected with a line that terminates at the final, best configuration.	80
6.1	Time-series attribution plot of $B_{z_{GSM}}$ for 1998_mild_001. Shown is the relative contribution of $B_{z_{GSM}}$ at each time step to the difference-from-reference of $e_{h_{ABK}}$	86
6.2	Feature ranking plot according to mean relative contribution for the set of all mild intensity storms at ABK.	86
6.3	Time-series attribution plot of $B_{z_{GSM}}$ for 1998_mild_001. Shown is the relative contribution of $B_{z_{GSM}}$ at each time step to the difference-from-reference of e_{ABK}	87
6.4	Feature ranking plot according to mean relative contribution for the set of all mild intensity storms at ABK with auto-regressive input features included. . . .	87
6.5	Feature ranking plot according to mean relative contribution for a set of CME-driven storms at HER and ABK	91
6.6	Feature ranking plot according to mean relative contribution for a set of Corotating Interaction Region (CIR)-driven storms at HER and ABK	92
6.7	Feature ranking plot according to mean relative contribution for a set of CIR-driven storms at HER and ABK	93
6.8	Feature ranking plot according to mean relative contribution for a set of CIR-driven storms at HER and ABK	93
6.9	Feature ranking plot according to mean relative contribution for a set of CME-driven storms and a set of CIR-driven storms for $ \Delta e_h $	94
6.10	Time-series attribution plot of all input features for the prediction of $e_{h_{HER}}$ for 2001_severe_005. Each individual input feature is represented by a colour map - with the y-axis of each being the time axis into the past (0 minutes and -30 minutes respectively) and the x-axis of the plot being the time axis in the direction of the storm's progression.	95
6.11	Time-series attribution plot of all input features for the prediction of $e_{h_{ABK}}$ for 2001_severe_005.	96
6.12	Input features ranked by mean relative contribution to the output for $e_{h_{HER}}$ and $e_{h_{ABK}}$ predicted individually during 2001_severe_005.	97
6.13	Attributions of B_z , $ B $, P_d , and N_ρ at 0 and -30 minutes for $e_{h_{HER}}$ during the storm 2001_severe_005. A time-series plot of each input parameter as well as SYM-H are shown at a lag of 0 and -30 minutes alongside the attribution feature maps.	97

6.14	Attributions of B_z , $ B $, B_Y , and N_ρ at 0 and -30 minutes for $e_{h_{ABK}}$ during the storm 2001_severe_005.	98
6.15	Input feature rankings for $e_{h_{HER}}$ and $e_{h_{ABK}}$ predicted simultaneously.	100
6.16	Input feature rankings for $ \Delta e_h $	100
6.17	Time-series attribution plot of all input features for the prediction of $e_{h_{HER}}$ for 1997_moderate_001.	102
6.18	Time-series attribution plot of all input features for the prediction of $e_{h_{ABK}}$ for 1997_moderate_001.	103
6.19	Input features ranked by mean relative contribution to the output for $e_{h_{HER}}$ and $e_{h_{ABK}}$ predicted individually during 1997_moderate_001.	103
6.20	Attributions of $ B $, P_d , and V_{sw} at 0 and -30 minutes for $e_{h_{HER}}$ during the storm 1997_moderate_001.	104
6.21	Attributions of $ B $, P_d , and V_{sw} at 0 and -30 minutes for $e_{h_{ABK}}$ during the storm 1997_moderate_001.	104
6.22	Input feature rankings for $e_{h_{HER}}$ and $e_{h_{ABK}}$ predicted simultaneously during 1997_moderate_001.	105
6.23	Input feature rankings for $ \Delta e_h $ predicted during 1997_moderate_001.	106
6.24	Input features ranked by mean relative contribution to the output for $e_{h_{HER}}$ and $e_{h_{ABK}}$ predicted individually during 2001_severe_005 with auto-regressive input features included.	107
6.25	Input features ranked by mean relative contribution to the output for $e_{h_{HER}}$ and $e_{h_{ABK}}$ predicted individually during 1997_severe_001 with auto-regressive input features included.	107
A.1	Scenario plot for the selection of Multi-Layer Perceptron (MLP) hyperparameters on the synthetic data set.	125
A.2	Output values ordered by target output value from smallest to largest vs. target output values for train and evaluation sets.	125
A.3	Compilation of ordered features and their mappings vs their respective DeepSHAP attributions.	126
A.4	Compilation of input feature values, transformed input feature values, and each input feature's respective DeepSHAP as sorted from smallest to largest according to target output value.	126
A.5	The ratio of points gained vs. interpolation limit for the years 1996, 1998, 2000, 2002, 2004, 2006 and 2008	129

A.17 Scenario plot portraying the configurations searched over in each scenario’s configuration space. The scenarios portrayed here are Scenario 1: early stopping enabled with patience, 3 rounds of intensification, Scenario 2: early stopping enabled without patience, 3 rounds of intensification, and Scenario 3: early stopping enabled with patience, 2 rounds of intensification. Please note that the intensification rounds in Scenario 3 are rendered ‘hobbled’ by setting a constant epoch budget. 142

List of Tables

3.1	Hyperparameter search configuration space for MLP model selection.	39
4.1	Description of input features. Name, symbol, unit, min, max, and median of each input feature are listed, based on isolated storm data for the years 1996 to 2019.	47
4.2	The extreme minimum values of SYM-H for severe, moderate, and mild storms as seen in [13]	52
4.3	Summary of the data points in the Hermanus-Abisko Storms (HAS) data set. Listed is the total number of points, minimum, maximum, and average number of points for the entire data set and each storm intensity, as well as the total and average number of remaining data points containing Not-a-Numbers (NaNs) for the entire data set and each storm intensity.	54
4.4	List of theoretical distributions to be fit to each parameter	57
4.5	List of selected theoretical distributions, their ‘rank’ out of 10 based on Root Sum of Squares (RSS) score, and their RSS score.	58
5.1	Hyperparameter search configuration space for Time-Shift Multi-Layer Perceptron (TSMLP) model selection.	67
5.2	Training and validation loss at best epoch with hyperparameter values for best model configurations for each task.	69
5.3	Performance evaluation with root-mean-squared error and Pearson’s r coefficient on the evaluation set of each task’s selected model. For all tasks except $ \Delta e_h $, predictions were scaled up to original units. $ \Delta e_h $ remains unitless: although nT is listed as the RMSE unit for all rows, take $ \Delta e_h $ to have no units.	70
5.4	Training and validation loss at best epoch with hyperparameter values for best model configurations for each task.	73
5.5	Performance evaluation with root-mean-squared error and Pearson’s r coefficient on training, validation, and evaluation sets of each task’s selected model.	74

5.6	Mode strongest correlated time-lags for training, validation, and evaluation set storms for each task where auto-regressive input features at -30 minutes are appended to the input.	76
A.1	Start/end times of storms by Storm ID	127
A.2	Ratio of points gained per interpolation limit in minutes for the years 1996, 1998, 2000, 2002, 2004, 2006, 2008	128

List of Acronyms

DNN Deep Neural Network

ANN Artificial Neural Network

ReLU Rectified Linear Unit

SGD Stochastic Gradient Descent

MSE Mean Squared Error

MLP Multi-Layer Perceptron

TSMLP Time-Shift Multi-Layer Perceptron

TCN Temporal Convolution Network

HER Hermanus

ABK Abisko

CME Coronal Mass Ejection

CIR Corotating Interaction Region

IMF Inter-Planetary Magnetic Field

BSN Bow-Shock Nose

NaN Not-a-Number

SMAC3 Sequential Model Algorithm Configuration 3

BO Bayesian Optimisation

EDA Exploratory Data Analysis

ReLU Rectified Linear Unit

RMSE Root Mean Squared Error

SANSA South African National Space Agency

NITheCS National Institute for Theoretical and Computational Sciences

NASA National Aeronautics and Space Administration

HAS Hermanus-Abisko Storms

HRO High-Resolution OMNI

GSE Geocentric Solar Ecliptic

GSM Geocentric Solar Magnetic

SHAP SHapley Additive exPlanation

LRP Layer-Wise Relevance Propagation

SSC Sudden Storm Commencement

PDF Probability Density Function

RSS Root Sum of Squares

MLE Maximum Likelihood Estimation

NLL Negative Log Likelihood

GIC Geomagnetically Induced Current

GMD Geomagnetic Disturbance

ACE Advanced Composition Explorer

LSTM Long Short-Term Memory

XAI eXplainable Artificial Intelligence

CNN Convolutional Neural Network

RNN Recurrent Neural Network

HCI Human-Computer Interaction

MAPE Maximum Absolute Percentage Error

NLP Natural Language Processing

NITheCS National Institute for Theoretical and Computational Science

Chapter 1

Introduction

An introduction to the study, the research questions, problem statement, and an overview of the key findings.

1.1 Background

Geomagnetic storms, induced by solar activity such as Coronal Mass Ejections (CMEs) and Corotating Interaction Regions (CIRs), are known to adversely effect various infrastructure. One such adverse effect is the production of Geomagnetically Induced Currents (GICs), which have the potential to damage to electricity infrastructure, possibly resulting in power outages [1]. Geomagnetic storms are described using several ground-based Geomagnetic Disturbance (GMD) indices, which have been predicted with the use of deep learning techniques to moderate levels of success [2, 3, 4, 5]. However, the majority of geomagnetic storm and geomagnetic activity modeling is empirical. As demonstrated by the lack of truly accurate physics-based models or a complete theoretical framework in many cases, several unknowns persist in the realm of solar physics [3].

Deep learning techniques are very powerful when provided with sufficient data and computing power [6]. The large capacity of Deep Neural Networks (DNNs) means that even a simple

architecture such as an Multi-Layer Perceptron (MLP) would reasonably be able to capture many nuances and correlations in a data set that traditional statistical modeling or machine learning techniques would not typically be able to detect [6]. This capacity of DNNs reveals the potential for utilising these empirical modeling techniques to create new knowledge about these complex solar physics systems (knowledge discovery), or at least confirm existing knowledge with the potential for later knowledge creation as knowledge discovery techniques involving DNNs evolve.

While DNNs have the capacity to accurately model complex relationships in a provided data set, these models are by nature ‘black boxes’, and do not naturally lend themselves to ‘knowledge discovery’ in this way. However, interpretability techniques, such as Layer-Wise Relevance Propagation (LRP) and SHapley Additive exPlanation (SHAP), were specifically developed to assist with explaining the decision-making processes of a DNN by ranking the contributions of the input features to each of the DNN model’s predictions [7, 8, 9].

In this study, we investigate the feasibility of using an MLP and an interpretability technique such as SHAP in a knowledge discovery process of a solar physics problem: the response of local GMD indices at two different latitudes to solar wind plasma and the Inter-Planetary Magnetic Field (IMF) during geomagnetic disturbances. We train a modified MLP model, a Time-Shift Multi-Layer Perceptron (TSMLP), which is created by time-shifting each input feature, to predict the Wintoft index, e_h , (a local GMD index derived from the horizontal component of the geomagnetic field) as it is induced at Hermanus (HER), South Africa, and Abisko (ABK), Sweden. We then explain the relationships between each input feature and the prediction it produced by applying an interpretability technique to the TSMLP. Since the response of e_h to geomagnetic disturbances is different at the two geographical locations in question, whether or not the intricacies of these differences will be uncovered by this technique is of particular interest. The differences in e_h at these two latitudes are thoroughly explored in Section 4.3.

The data set is derived from solar wind data - with the input features describing both the solar wind plasma and IMF during a geomagnetic storm event. The particular input features are selected following an intensive data analysis process wherein expectations of the relationships of these features to the predicted index are formed. Interpretations are generated using SHAP, an additive attribution technique, and are compared to domain insights from the space physics

literature.

1.2 Problem statement

Solar physics is home to several problems that have incomplete theoretical models. Deep learning techniques have been applied to problems in this domain, however, the application of interpretability techniques to explain these resulting DNN models have been explored to a limited extent. It is not known to what extent interpretability techniques, such as additive attribution methods, are able to assist with knowledge discovery in solar physics problems that have non-stationary spatio-temporal elements – such as the way GMD indices are induced at different latitudes.

The formation of geomagnetic storms and the prediction of GMD indices is a well-studied problem. It is a problem that has many spatio-temporal dependencies and interesting interactions between the solar wind parameters used to model these events and the indices they predict. Specifically, the prediction of local ground-based geomagnetic indices at two different latitudes, which possesses the properties of temporal dependency and non-stationarity, is a suitable surrogate problem which will test the feasibility of applying additive attribution methods to deep neural networks for knowledge discovery in the solar physics domain.

1.3 Project scope

Throughout the project, the scope of data sets, techniques, and tools were limited to:

- **Data sets:** Solar wind and IMF data curated by National Aeronautics and Space Administration (NASA) and available at OMNI¹ will provide the input features of our models. Our target, the e_h index [10], will be derived from magnetic observatory data hosted by INTERMAGNET² for Hermanus, South Africa, and Abisko, Sweden.
- **Techniques:** DeepSHAP [8], an implementation of SHAP [7] (a post-hoc, additive feature attribution technique) that is combined with DeepLIFT for use with DNNs, will be the

¹<http://omniweb.gsfc.nasa.gov>

²<https://intermagnet.github.io/>

interpretability technique of choice for this project. An MLP architecture with time-shifted inputs will be used as the modeling technique.

- **Tools:** Existing frameworks such as Pytorch³, Pandas⁴, and Numpy⁵ will be used to create data processing and modeling tools with the Python⁶ programming language. For the majority of the modeling use cases, an existing toolkit developed by the MUST research group, ‘mustnet 3’, and an open source hyperparameter optimisation package, ‘SMAC3’⁷, will be used. Scott Lundberg’s implementation⁸ of SHAP, which includes an implementation of DeepSHAP, will be used for interpretation use cases.

Given that the application of *interpretability techniques* to the domain in an end-to-end knowledge discovery process is the main focus of the study, the particular modeling of the time-series will be simplified as much as possible to allow for more focus on the interpretation-generation process itself. For this reason an MLP with time-shifted inputs, which we will call a TSMLP, will be used as the DNN architecture of choice rather than more involved architectures such as a Temporal Convolution Network (TCN) or Recurrent Neural Network (RNN).

While the DeepSHAP attributions will be evaluated for faithfulness, the evaluation process should be reasonable for the study’s time frame. The evaluation processes will be limited to two main techniques: *ground-truth comparison* (as a baseline example with a synthetic data set) and *domain-expert agreement*, with extensive reference to and comparison with the domain scientific literature. The *consistency* of the attributions will be evaluated through comparison between variations of the same data set as a consequence of the experimental process.

1.4 Research questions

The following research questions were posed for this study:

1. Does DeepSHAP, when interpreting an MLP trained on a synthetic data set, accurately attribute the input feature relationships in that data set?

³<https://pytorch.org/>

⁴<https://pandas.pydata.org/>

⁵<https://numpy.org/>

⁶<https://www.python.org/>

⁷<https://github.com/automl/SMAC3>

⁸<https://github.com/slundberg/shap>

-
2. Is an MLP, trained with time-shifted input features to capture the time-dimension, able to predict e_h at both latitudes?
 3. Do the TSMLP models, when interpreted by DeepSHAP, assign the same importance to input features at different latitudes that would be expected based on current knowledge of geomagnetic storms?

1.5 Objectives

This study aimed to determine the feasibility of applying DNNs with attributions generated by an interpretability method for knowledge discovery in a problem with spatio-temporal dependencies and non-stationarity. To fulfil this aim, and answer the preceding research questions, the following objectives were devised:

- Curate a data set with selected solar wind parameters as the input features and the e_h GMD index as derived with data measured at HER and ABK as the output targets.
- Conduct an Exploratory Data Analysis (EDA) to analyse the relationships between various input features and the target e_h at each latitude during geomagnetic storm events.
- Train MLPs that include time-shifted inputs (TSMLPs) on the data set to predict e_h in a variety of different tasks.
- Generate attributions for the TSMLP models using DeepSHAP and compare the feature attributions for e_h at each latitude to the expectations of feature importance.
- Evaluate the feasibility of using a framework similar to this study as a possible knowledge discovery mechanism in further work.

1.6 Research methodology

This is an exploratory research study wherein existing techniques will be applied to a solar physics problem in order to determine the feasibility of using a DNN-interpretability technique pipeline as a knowledge discovery tool.

This study is designed as follows:

- **Literature review:** A literature review is conducted to provide a background to geomagnetic storm events, deep neural networks, and interpretability techniques. Particular emphasis is placed on the details of geomagnetic storm progression, and the particulars of and differences between CME-driven and CIR-driven storms. Interpretability techniques, the terms and definitions for these techniques, and various evaluation methods for interpretability techniques will be discussed. Previous work where interpretability techniques were used with various DNN architectures in similar applications will also be reviewed.
- **Development:** Besides the in-house mustnet 3 toolkit and the open source SMAC 3 package, the following scripts, toolsets, and programs are developed in the Python programming language:
 - **Data processing pipeline:** A custom data processing pipeline was developed that reads the particular data file formats of High-Resolution OMNI and both observatories (HER and ABK), handles interpolation, e_h derivation, geomagnetic storm selection, and other pre-processing steps automatically, is able to create and manage synthetic data sets, and seamlessly integrates with the mustnet 3 toolkit.
 - **Exploratory Data Analysis (EDA) scripts:** Assorted scripts that conduct various EDA tasks, such as calculating correlations, determining best distribution fit, and drawing and saving various plots.
 - **Utility scripts:** Assorted utility scripts that manage model training results, the storage of various model performance metrics, and automatic organisation of various other miscellaneous outputs from other scripts and toolkits.
 - **Assorted plot creation scripts:** A set of scripts that automatically draw and save plots for various purposes: such as model output, input features and SYM-H values during storms, histograms, etc.
 - **SMAC3 result manager:** A set of scripts/toolkits that automatically obtains performance metrics and hyperparameter values of all models trained in a finished SMAC3 run and draws and saves plots that summarise the results of these runs.
 - **SHAP wrapper:** A wrapper for the calculation of DeepSHAP attributions that manipulates calculated attributions as needed and draws and saves all necessary result plots.

-
- **Experimental design:** In order to meet the objectives of this study and answer the research questions, the following experiments were conducted:
 - Using a custom synthetic data set with clear relationships between input features and output targets, a ground-truth evaluation of DeepSHAP attributions is conducted to determine if the DeepSHAP attributions match the true underlying relationships in the data set.
 - An EDA process is conducted to analyse the correlations and distributions of solar wind plasma/IMF input features, as well as to set the groundwork for the expectations of the TSMLP training results and the DeepSHAP attribution results.
 - TSMLPs are trained to predict e_h at either latitude, at both latitudes, and to predict the explicit difference between e_h at these latitudes.
 - The best-performing TSMLP for each prediction task is then used for the interpretation phase. Here, DeepSHAP attributions are calculated for each of these TSMLPs over all storms in the evaluation data set. Particularly representative storms with CME and CIR drivers are discussed and compared to the expectations based on the physical realities as well as the expectations produced by the EDA process.

1.7 Dissertation structure

This dissertation is structured as follows:

- **Chapter 2** contains the literature review, and provides the background for the study of solar weather and GMDs, deep learning techniques (DNNs and MLPs in particular), and DNN interpretability techniques, with a focus on the “DeepSHAP” SHAP implementation.
- **Chapter 3** contains the implementation of a baseline synthetic data set, provides an example of the use of SMAC3 for hyperparameter selection, and reports the process and results of a ground-truth evaluation of DeepSHAP attributions generated for the synthetic data set.
- **Chapter 4** walks through a description of the solar wind data set, its pre-processing steps, and an EDA process.

-
- **Chapter 5** describes the TSMLP architecture that is then trained to predict e_h at either latitude, both latitudes or the explicit difference in e_h on the solar wind data set developed and analysed in Chapter 4, and discusses the results and the performance of these models.
 - **Chapter 6** handles the generation of DeepSHAP attributions for the models trained in Chapter 5 and evaluates these attributions based on the expectations of input feature importance at either latitude based on the underlying physics.
 - **Chapter 7** concludes the dissertation with a summary of this study’s findings as well as future work.

1.8 Publications

Some content of this study has been presented to peers and external audiences at two opportunities:

- South African National Space Agency (SANSA) Student Workshop, 18-22 October 2021, project concept.
- A presentation drawn from Chapters 4, 5, and 6 was presented at National Institute for Theoretical and Computational Science (NITheCS) Machine Learning Workshop, SACAIR 2022, 6 December 2022.

Chapter 2

Background and literature review

This chapter provides the background to and literature review for the content of our study, a description and explanation of our problem domain (solar weather), knowledge discovery, explainable AI (XAI), and our eXplainable Artificial Intelligence (XAI) technique of choice: DeepSHAP.

2.1 Overview

This chapter provides the literature review of and background to this study. We discuss solar weather, and in particular the formation of Geomagnetically Induced Currents (GICs) during geomagnetic storms and geomagnetic storm drivers in Section 2.2. Deep learning techniques are discussed in Section 2.3, and a general overview of XAI with an explanation of various techniques and evaluation methods of XAI outputs in Section 2.4. A particular focus on our XAI technique of choice, SHapley Additive exPlanation (SHAP), is provided in Section 2.5. The DeepSHAP implementation of SHAP will be used in this study and is explained in Section 3.2. An overview of previous work is provided in Section 2.6.

2.2 Geomagnetic storms and GIC formation

Solar weather, solar wind, and geomagnetic storms are physical phenomena that depend on a cyclical process, namely a solar cycle. A solar cycle is defined as the approximately 11-year cycle in which the number of sunspots, and thereby the level of solar activity, waxes and wanes. Near the maximum of the solar cycle the active areas (sunspots) become more in number and more complex, resulting in increased frequency of eruptions such as the Coronal Mass Ejections (CMEs) that are the drivers of some geomagnetic storms [11].

As a geomagnetic storm progresses, it has several effects on the Earth itself. One major effect is the formation of GICs, which are quasi-DC (1Hz) currents that are able to pass through transformers and cause damage to electrical grids. One notable instance is a geomagnetic storm in October of 2003 causing a power outage in Sweden as a result of these GICs [12]. There a large body of work that has gone into investigating the relationship between GICs, and other geomagnetic effects, and geomagnetic storm drivers [13, 14, 15, 16, 1, 17]. In general, the understanding is that higher latitudes, such as at Sweden, experience more severe GIC formations and other geoeffective effects than low and mid-low latitudes, such as in South Africa [14, 15, 16, 18]. However, the GIC formations produced in these lower latitudes, while less likely to take out a grid outright, are still able to accelerate the aging of infrastructure. However, it is difficult to directly study the link between GIC formation and geomagnetic storms themselves, so a GMD index is often used as a proxy for GICs.

2.2.1 The geomagnetic disturbance index

Geomagnetic storms and general geomagnetic disturbances are described by several ground-based GMD indices. These indices include the K_p , or the planetary K-index, a 3-hourly measure of the perturbation of the global geomagnetic field at the Earth's surface, local K-indices (like K_p , but only for particular locations), the Dst index, a measure of the ring current strength based on an average of global ground-based measurements, and several others [13, 3, 19].

The manifestation of geomagnetic disturbances and their effects have strong spatio-temporal dependency due to the shape of the Earth's dipole magnetic field and the dynamic nature of solar wind components, as well as the progression of the geomagnetic disturbance itself [19, 11,

15, 14]. For this reason, it is known that geomagnetic disturbances will manifest differently at different times and at different locations. In particular, the more exposed regions of the higher latitudes will have stronger, faster, and more extreme reactions to the magnetospheric coupling and subsequent strengthening of the ring current caused by the inflow of ions compared to regions in lower-middle latitudes (for example: the manifestation of aurora) [15, 14].

The particular formation and measurement of GIC during storms depend on these spatio-temporal factors as well as other information such as the geology of certain regions and the availability and layout of electricity infrastructure. As a result, GIC and their effects are difficult to measure or estimate accurately. For this reason, a geomagnetic disturbance index based on the time-derivative of the horizontal geomagnetic field measured at any particular region is used as a proxy index for GICs [10]. The derivation of this index, which is known as the e_h index, is walked through in Chapter 4.

2.2.2 Drivers of geomagnetic storms

The Earth's *magnetosphere* is the region of near-Earth space dominated by the geomagnetic field. The shape of the magnetosphere is altered by the solar wind flow and CMEs cause even larger perturbations to the magnetosphere than ambient solar wind. Both solar wind and CMEs contain magnetized ions - thereby carrying the Sun's magnetic field to Earth. This carried magnetic field couples with the Earth's magnetic field, forming new magnetic field lines, through a process known as magnetic reconnection, and exposing the Earth's magnetosphere to the CME plasma [14, 20]. This instream of ions strengthens the already-existing ring current in the ionosphere, weakening the Earth's magnetic field, strengthening a counter-magnetic field, and further allowing more ions into the ionosphere. This cycle continues until the CME passes. The fluctuations caused by the re-organisation of current systems due to CME impact is known as a geomagnetic storm [11, 13].

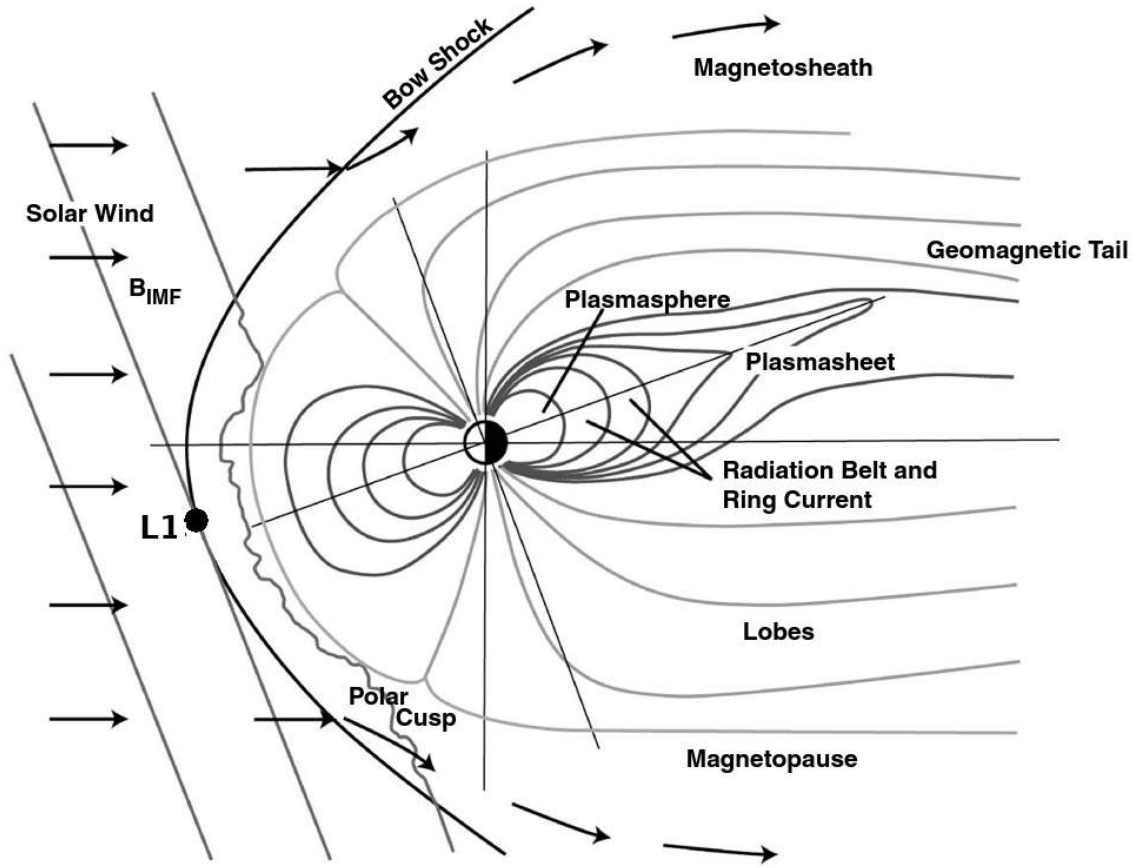


Figure 2.1: Simplified view of the Earth's magnetosphere showing the solar wind encroaching from the left hand side. The L1 lagrange point, where the satellites and spacecrafts recording the solar wind paramaters are located, is indicated on the image. Adapted from Moldwin et al. [11]

CMEs are not the only major storm drivers - there exists another plasma ejection stream, Corotating Interaction Regions (CIRs), that is formed by the compression of the solar wind plasma at the interface of high and ambient speed solar wind. It is ejected from the Sun like a 'sprinkler', and is modulated to some extent by the 27-day rotation period of the Sun [21]. While CMEs most often occur close to the solar maximum, CIRs occur mostly during the declining and minimum phases of a solar cycle. CIRs cause a more gradual, steady increase in solar wind pressure over the duration of the storm rather than a sudden increase as is seen with CMEs, and have different interactions with the Earth's magnetosphere than CME [21]. CIRs often result in less severe geomagnetic storms that last for a longer duration than CME-driven storms [21].

2.2.3 Geomagnetic storm modeling

Statistical modeling for geomagnetic index prediction and the study of geomagnetic storms is well established. Techniques such as NARMAX (Nonlinear Auto Regressive Moving Average With Exogenous Inputs) and multi-linear regression based techniques have been applied to both Kp prediction and GIC modelling respectively [2, 1]. Neural-network based techniques, such as the popular Costello Neural-Network model (one of the first artificial neural network based Kp prediction models) which used solar wind parameters from the Advanced Composition Explorer (ACE) spacecraft and was developed in 1997 [2], MLPs with time-shifted inputs [3, 4, 5], Recurrent Neural Networks (RNNs) [19], and Long Short Term Memory models (LSTMs) [22] among others, are also well-utilised methods of GMD prediction.

Lotz et al. describe the development of an MLP with time-shifted inputs to accurately predict SYM-H using solar wind parameters [23]. Gruet et al. make use of an Long Short-Term Memory (LSTM) with Gaussian Process modeling for Dst index prediction, where the LSTM makes a forecast of 1 to 6 hours into the future. This forecast is then used as the mean function of a Gaussian Process to create a probabilistic forecast of Dst values [22]. Wintoft uses RNN models in his process of predicting fluctuations in induced geomagnetic field as measured at two locations: Uppsala, Sweden, and Brorfelde, Denmark [19].

While all of these applications and examples are quite interesting in itself, it should be noted that the majority of solar physics and geomagnetic storm understanding is based more on empirical modeling than it is on a complete theoretical framework [3]. With the potential of neural network interpretation techniques to provide insight into problems and data sets it is applied to [9], the use of models, coupled with good explanations, may eventually provide a means to bolster current theoretical understanding with new insights gained from an empirical process.

2.3 Deep learning techniques

Deep learning techniques are machine-learning algorithms applied to both classification and regression tasks that are based on Deep Neural Networks (DNNs), a variation of Artificial Neural Networks (ANNs) that typically have two or more layers. ANNs are a technology with a long history, with McCollough and Pitts proposing the first mathematical model of a neuron in

1943 [24]. A neuron forms the most basic part of an ANN: a network of these neurons, or nodes, that are connected between an input and an output [25]. In the most basic implementation of an ANN is a plain *linear neural network*, where each node performs a linear transformation of its input. This, however, has the additional property that if there is more than one layer of nodes, the linear transformations of each layer can all be reduced down to a single linear transformation for the entire model [25]. The strength of an ANN, however, lies in the addition of an *activation function*, which, when applied to a node, non-linearly transforms the output of that node in a variety of ways that enables the approximation of diverse non-linear relationships [25, 6].

Each layer added to an ANN increases its *depth*, hence the term deep learning, and by extension, Deep Neural Network [6]. The term DNN describes an ANN that consists of two or more layers [25]. The layers in a DNN that are not directly connected to either the input or the output are known as hidden layers, and are often densely connected: i.e., each node in a preceding layer is connected to each and every node in a following layer, and each node in that following layer is connected to each and every node in its preceding layer [25, 6].

There is a theoretical set of parameters (Θ) that will enable a ANN or DNN to perfectly approximate an output distribution based on an input distribution [6]. This set of parameters is approximated as closely as possible to minimise the error between the target output and predicted output with an optimisation function, $J(\Theta)$ [6]. This optimisation function conducts *backpropagation*, which takes an error term for an output and, using the chain-rule of derivatives, propagates this error back through the DNN to update the model’s parameters accordingly.

2.3.1 Feed-forward multi-layer perceptron

The simplest DNN architecture is a densely-connected feed-forward network, also known as a feed-forward multi-layer perceptron, or simply, an MLP [25, 6]. The MLP consists of several (two or more) hidden layers that forward-propagate an input through each layer to the output [25, 6]. There are no back-connections or loops, hence ‘feed-forward’. This is the DNN architecture that we will use in this study, as it is a relatively simple, yet powerful architecture for a variety of regression and classification applications [6]. We will use the description of this architecture to explain the concepts touched on in the previous section in some more depth.

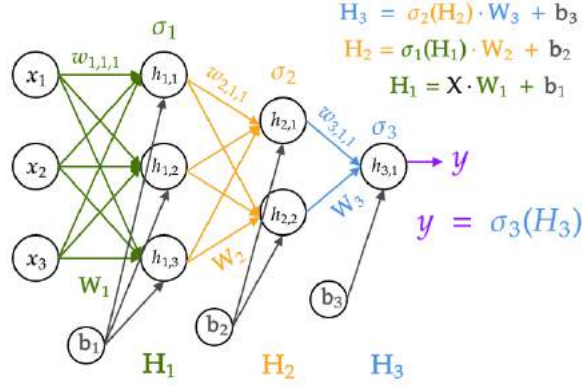


Figure 2.2: Illustration of an MLP with two hidden layers and a single output node - showing flow of model parameters such as weights (w) and biases (b), and the application of activation functions to each hidden layer and the output node (σ).

Figure 2.2 illustrates an MLP with 2 hidden layers and an output node. An MLP consists of two or more hidden layers (H) of nodes connected from an input (X) to an output (y), with an activation function (σ) applied to each layer [25]. A node applies a linear transformation to an input. For example, hidden layer H_1 in Figure 2.2 applies a dot product between the vector of input features $\mathbf{X}$ and the vector of weights $\mathbf{W}_1$, and adds to that a bias vector $\mathbf{b}_1$.

$$\mathbf{H}_1 = \mathbf{X} \cdot \mathbf{W}_1 + \mathbf{b}_1 \quad (2.1)$$

The following layer, $\mathbf{H}_2$, applies a similar transformation to its input. However, an activation function σ_1 has been applied to layer $\mathbf{H}_1$. For this reason, the input to layer $\mathbf{H}_2$ is $\sigma_1(\mathbf{H}_1)$, so the linear transformation applied by $\mathbf{H}_2$ to this input is as follows:

$$\mathbf{H}_2 = \sigma_1(\mathbf{H}_1) \cdot \mathbf{W}_2 + \mathbf{b}_2 \quad (2.2)$$

This continues to the final layer, H_3 , which consists of a single output node. The model output, y , is provided by:

$$y = \sigma_3(\mathbf{H}_3) = \sigma_3(\sigma_2(\mathbf{H}_2) \cdot \mathbf{W}_3 + \mathbf{b}_3) \quad (2.3)$$

The output of the model is therefore a chain function of all the previous layers leading from the input features to the output value.

The model is trained by optimising the objective function $J(\Theta)$ to minimise the model loss L [6, 25]. The model loss is calculated by a loss function l such as Mean Squared Error (MSE) loss. This is done by updating the model parameters Θ (which are the weights and biases) by way of

back-propagation. Back-propagation updates the values of the model parameters relative to the model loss by way of the chain rule of derivatives. The magnitude and direction in which these parameters are updated is proportional to the negative of the gradient of the loss with respect to that parameter [25, 6]. For example, if:

$$J(\Theta) = L = l(y', y) \quad (2.4)$$

with L the model loss calculated via the loss function of the model output (y) and the target output (y'), and:

$$y = \sigma(H_3) = \sigma_3(\sigma_2(\sigma_1(X \cdot \mathbf{W}_1 + \mathbf{b}_1) \cdot \mathbf{W}_2 + \mathbf{b}_2) \cdot \mathbf{W}_3 + \mathbf{b}_3) \quad (2.5)$$

in the case of our example MLP, then, according to the chain rule, if we wanted to update a weight $w_{1,1,1}$ we could do so as follows:

$$\Delta w_{1,1,1} = -\eta \frac{\partial L}{\partial w_{1,1,1}} \quad (2.6)$$

with η being the *learning rate*, a proportional parameter that is set by the optimisation function being used.

The most commonly used optimisation function is Stochastic Gradient Descent (SGD), which updates the parameters according to the average gradient of a mini-batch of randomly selected input samples that have been passed through the model [6]. The learning rate used by SGD may either be fixed, or gradually increase over time. A popular optimisation function with an adaptive learning rate that is based on SGD that is used in this study is known as Adam [26, 6]. This name is derived from the term ‘adaptive moments’, and it makes use of the estimated first and second order moments of the gradients that adapt the learning rate during the optimisation process [26].

The two activation functions used in this study are Rectified Linear Unit (ReLU) [6] and leaky ReLU [27]. These activation functions suppress negative outputs of hidden nodes and passes their positive outputs. ReLU suppresses negative output values by forcing all negative outputs to zero. This negative output value suppression helps to prevent vanishing gradients, where the hidden layer outputs shrink to such a degree that the model is no longer able to effectively update its parameters. However, overly aggressive suppression of hidden node outputs may lead to dead nodes, where nodes fail to activate. To counteract this, leaky ReLU passes negative output values that have been scaled to minimise their effect on following nodes, but still allows certain nodes to activate and eventually ‘recover’ during training.

2.3.2 Time-series modeling

Besides MLPs and other feed-forward networks (such as Convolutional Neural Networks (CNNs)), there are DNNs that have some feed-back of previous output values built in, and which are often used for time-series or sequence modeling tasks. Some of these sequence-modeling DNN architectures include, but are not limited to:

- **RNNs:** Recurrent neural networks incorporate a *hidden state* that summarises all sequential information up to $t - 1$ as an input to its hidden layers [25, 6, 28]. This is a family of neural networks that incorporates auto-regression for sequence modeling, and is used as a basis for a multitude of Natural Language Processing (NLP) tasks [6].
- **LSTMs:** LSTM networks are RNNs that incorporate ‘long short-term memory’ modules that allow for longer sequences to be emphasized as important before the model ‘forgets’ that input [25].
- **TCNs:** TCNs encode time-series data by performing 1-dimensional convolutions along the time-axis of the input sequence. The receptive field size, or length of the input sequence it is able to process, depends on the depth of the TCN. In order to prevent over-sized models that are far too computationally expensive, ‘residual blocks’ are implemented to increase receptive field size without increasing model capacity too much [29, 30].

However, not only these comparatively complex architectures are capable of sequence-modeling tasks. In the work of Lotz et. al [23], temporal dependencies were included in the modeling process by including an additional set of time-shifted inputs alongside the input features to an MLP. We similarly elect to use a similar MLP architecture in this study as is described in Section 5.2. This sort of architecture was used by Beukes et. al. ([4]) in a task similar to this one, and is also a relatively simple architecture that will allow focus on the interpretation aspects of the problem.

2.4 eXplainable AI: techniques and definitions

Over the past several years, eXplainable AI (XAI) has become ever more prominent in the context of AI fairness and the general mistrust of the ‘black box’ model class: DNN. XAI has

alternated between being termed ‘Interpretable Artificial Intelligence’ and ‘Explainable Artificial Intelligence’ in the literature, and similarly, ‘explainability’ and ‘interpretability’ are often used interchangeably, however, these terms are understood to have important distinctions from one another [31].

Arrieta et al. define interpretability as ‘the ability to explain or to provide the meaning in understandable terms to a human.’ and explainability as an accurate and comprehensible ‘interface between humans and a decision maker’ [31]. Since the definitions of these terms tend to be interchangeable in some contexts and very distinct in others, we will enforce our own understanding of these terms as follows. We use ‘interpretability techniques’ to refer to the actual techniques used to interpret or probe models, and the comprehensions produced by these techniques as ‘explanations’. An ‘interpretation’ is then a numerical artefact produced by a technique that calculates feature influence on an output, often leveraging the aspects of a model (such as activations or gradients) to come to this result. ‘Interpretability’ will be used to refer to the concept of XAI and as a blanket term for the techniques involved. ‘Explainability’ or ‘explainable’ will refer to the degree of transparency (literally, explainability) of a model. An ‘explanation’ refers to a human-comprehensible ‘why statement’, such as noting that one feature activates a model more strongly than another, and therefore must be more important.

There are two main branches of interpretability: the use of models that are ‘transparent by design’ to approximate less transparent models, and the application of ‘post-hoc’ techniques to an opaque system. An opaque system could refer to a proprietary model or pipeline, or to a black-box model such as a DNN. This study focuses on post-hoc techniques.

Several post-hoc interpretability techniques exist. Post-hoc techniques may be model-specific (techniques designed for a particular model type/architecture) or model-agnostic (techniques designed to be applicable to any model type/architecture). Many post-hoc techniques may fall in either category, but generally, a clear division can be drawn [31].

Post-hoc techniques may also be classified as ‘local’ or ‘global’ methods, which could each be made up of either intrinsic or post-hoc techniques. A global method generates explanations that are applied to the entire data set, while a local method generates explanations a single or a particular set of individual output instances or for a certain smaller subset of the data set [31].

This work is focused on the following kinds of model-agnostic methods for this study [31]:

-
- *Model simplification.* These methods approximate a surrogate model based on the model being explained. This is often done by creating a new linear model for each output instance of the target model [7]. Another approach to this approximates shallow or transparent models side-by-side to the more complex target model to produce the same or similar outputs, and explaining these approximate predictions with the inherent transparency of the approximated, more explainable, surrogate model. These techniques are often considered to be *local explanations*, since often the explanations produced account for only a small part of the target model system or for individual output instances.
 - *Feature relevance.* Feature relevance techniques rank and/or measure the importance of each input feature to a particular output instance. Feature relevance is a more recent and currently popular technique type, with many of them fitting the definitions of *model simplification*, and producing *local explanations*.

Feature relevance techniques are most readily applied to deep neural networks, such as MLPs, CNNs, and RNNs to name a few [31].

2.4.1 Local explanations: additive feature attribution techniques

Some post-hoc feature relevance techniques are often considered as ‘additive attribution methods’. The additive attribution methods are defined as being ‘local’ methods (creating explanations sample-by-sample) that fit a **linear** explanation model g to explain the predictions of a target model f [7]. These techniques, in this sense, are essentially a combination of ‘model simplification’ and ‘feature relevance’ techniques. According to Lundberg et al. [7], in order for a post-hoc feature relevance technique to be classified as an additive attribution method, they must also meet one or more of three desirable properties based on traditional Shapley-value based attribution methods, namely [7]:

- **Local accuracy:** the simplified outputs of the explanation model for corresponding simplified inputs must at least match the outputs of the target model for those same simplified inputs.
- **Missingness:** Features that are missing in the original input set should have no impact on explanations of other features.

-
- **Consistency:** If a feature’s *contribution* changes in any way, that feature’s *attribution* should change accordingly.

Some interpretability methods that conform to the definition of additive attribution methods, as well as meet one or more of the desirable properties, are Local Interpretable Model-Agnostic Explanations (LIME), Deep Learning Important FeaTures (DeepLIFT), Layer-wise Relevance Propagation (LRP), Shapley Regression Values, Shapley Sampling Values, Quantitative Input Influence (QII) and SHapley Additive exPlanations (SHAP). Of these methods, LRP and SHAP both have strong theoretical bases for their use and development [9]. LRP is based on Deep Taylor Decomposition, and SHAP Shapley analyses’ influence on feature attribution for interpretability and Shapley analysis itself [9]. Additionally, SHAP satisfies both the definition of an additive attribution method as well as all three desirable properties [7].

An additive attribution method is any interpretability technique that:

- Generates an attribution value for each input feature, or a value that quantifies the contribution of an input feature to the output.
- Adheres to the efficiency principle, i.e., the input feature attributions generated by that method, when summed together, at least add up to the output value less some baseline value.
- Adheres to the properties of missingness and local accuracy by necessity (since otherwise the *efficiency principle* cannot be adhered to), and optionally, to consistency.

2.4.2 Evaluating explanations

Ensuring the fidelity of an explanation - or ‘explaining the explanations’! - provided by an interpretability technique is an important and active field of study [32, 9, 31]. While interpretability evaluation methods were, and are still, often based on psychological, Human-Computer Interaction (HCI) , and social measures [31, 33, 34], a concrete measure of the **correctness** of an explanation, or a ‘computational measure’, is necessary for many use cases [33].

A reliable, believable, or trustworthy interpretability technique should meet the criteria of **faithfulness** (how reliably an explanation represents the model’s reality), **human explainability**

(how understandable an explanation is to a non/semi-expert human user), and **applicability** (the prospects of the technique to be applied in practice).

Faithfulness, which is often the computationally measured criterion, can be further broken down [35]:

- *Accuracy*. Is the explanation reliable for instances occurring outside the training-evaluation context of the model? i.e.: Does this method *generalise* well to real-world/out-of-sample data?
- *Fidelity*. Do these explanations reasonably reflect the expected and known behaviour of the model?
- *Consistency*. Does this method produce similar/comparable explanations for multiple models trained on the same task?
- *Stability*. Are similar explanations provided for similar predictions (with similar input features)?
- *Novelty*. If an out-of-distribution or novel instance is produced by the model, will the explanations reflect this? (See *Accuracy* for generalising to real-world data)

Some examples of commonly-used computational measures to fulfill some of these rigorous faithfulness criteria include [32, 33]:

- *Perturbation techniques*. This comprises the removal or augmentation of features seen as ‘relevant’ by the explanations, and evaluating the effect of this input perturbation on the accuracy of the model’s predictions [32, 35].
- *Ground-truth comparisons*. Typically, the components of a ground-truth comparison are: generating a synthetic data set that will serve as a known baseline [32, 36], creating an arithmetic toy-problem and corresponding implementation pipeline with which to test an end-to-end application [32], and user-generated (typically subject-matter expert) benchmarks for comparison with the explanation [33]. Ground-truth comparisons have various levels of demand with regard to implementation and complexity, and are not always feasible in all contexts. This is especially true when taking into account the complexity of more

advanced models and real-world data [9]. However, in a controlled context, ground-truth comparisons are useful as a base evaluation of whether a technique functions as expected.

- *Comparison with transparent models.* Another method of ensuring the quality of explanations is comparing simpler models with the target model. This entails applying a target black-box model and an ‘explainer’ transparent model to the same data set and comparing the explanations of the target model produced by the technique in question to the ‘standard’ set by the explainer model [32, 33]. This method is, however, severely limited by the capabilities of the transparent model used for comparison which might not capture all the nuances [32]. In addition, in the case of a task where an explainer model were to accurately explain the output, as well as produce similar results as a black box model, the argument could be made that that explainer model would be more suitable than the target black box model in the first place.
- *Expert agreement.* This requires evaluating explanations using domain knowledge and human intuition of the data set or problem it is applied to [37, 33] combined with expert knowledge of the inner workings of the target model [33, 7]. This more closely links to the HCI methods mentioned before than computation measures at first glance, but, when combined with one or more of the aforementioned techniques, is a powerful way to ensure trust in a particular technique.

None of these evaluation techniques, nor the interpretability techniques that they are evaluating, are perfect or comprehensive. XAI is a new field that was born and had to grow quickly out of necessity to keep pace with the rapidly developing, and rapidly more opaque, models that are being used in several sectors at an exponential adoption rate. Fairness, explainability, and trust are fundamental components in the development of any good A.I. model, and while these techniques are not necessarily perfect, many are well-understood (at least scrutinised!) enough to be used in practice and further studied in the ongoing quest for fairness.

2.5 SHAP

SHAP is an additive attribution method first proposed by Lundberg et al. [7], which estimates Shapley values for each input feature to a model. This is to bring the useful properties of Shap-

ley values to the computational efficiency provided by other state-of-the-art additive attribution methods. Since Shapley values are challenging and computationally expensive to calculate precisely, implementations of SHAP make use of certain shortcuts to estimate these values as closely as possible. These implementations make certain assumptions, such as conditional independence between input features, and use other additive attribution methods such as LIME and DeepLIFT as a computational backbone for its Shapley value estimations.

Here we briefly describe Shapley values, Shapley value estimation, and general implementations of vanilla SHAP. For our study, we choose to use DeepSHAP, a DNN-specific implementation of SHAP based on DeepLIFT which is described in detail in Chapter 3.

2.5.1 Shapley values

Shapley values were defined by Lloyd Shapley in 1953 for cooperative game theory as a means to calculate the *contribution* of each player to a game’s outcome, and thereby determine fairly to which portion of the result they are entitled [38].

If we are given a set of N *players*, who arrange themselves in a *coalition* that we call S to participate in a game with a characteristic function v , the contribution ϕ_i of a single player i to the game’s outcome $v(S)$ is determined by calculating the outcome of the game with that player absent, with that player present, and finding the difference.

Specifically, the contribution, or Shapley value ϕ_i , of player i , for all possible coalitions S , is obtained as follows [38]:

$$\phi_i = \frac{1}{N!} \sum_{S \subseteq N \setminus \{i\}} |S|!(|N| - |S| - 1)!(v(S \cup i) - v(S)) \quad (2.7)$$

The coalition S is a subset of the N players where player i is absent, where the players in coalition $S \subseteq N$ can be ordered in $|S|!(|N| - |S| - 1)!$ ways. The value ϕ_i is calculated by finding the difference between the results of the game function v with player i present and player i absent for all possible combinations of the players in coalition $S \subseteq N$, weighted by the number of possible combinations of players in coalition $S \subseteq N$. The resultant sum is averaged by the number of possible arrangements of the total N players.

This is to ensure that any *conditionality* that is introduced by the particular arrangement of the

players in the game is controlled for, since the order in which the other players are introduced into the coalition $S \subseteq N$ effect the game’s outcome. Shapley values can then be described as the *marginal contribution* of a feature to the output, that is, the contribution of that feature to the output on its own merit, regardless of the contribution/influence of other features on the output or itself. The Shapley values ϕ_i for each player i can be expressed as an additive function $\bar{v}(S)$ that approximates the game outcome $v(S)$ for all $S \subseteq N$ for N players [38]:

$$\bar{v}(S) = \sum_{i \in S} \phi_i \quad (2.8)$$

2.5.2 Vanilla SHAP

SHAP estimates a Shapley value describing the contribution of each feature in a particular sample to the model’s output, where the model is the analogue of a *game*, and each feature represents a *player* [7]. These values are known as the feature *attributions*, or *SHAP values*.

Once the feature attributions are calculated, they are expressed as an additive function similar to the one defined in Equation 2.8, with the exception that the contribution of the *baseline output* of the model is also included.

The baseline output is the output of the model where *all* players are *absent* – or where all features are set to some background/reference value. In the case of Shapley values, if all players are absent from a coalition, the game outcome $v(S)$:

$$v(S) = v(\emptyset) = 0 \quad (2.9)$$

However, the models that we are trying to attribute features for do not necessarily deal well with an input feature that is ‘missing’. To work around that, instead of having the feature (player) as absent, that feature is set to some background or reference value. This background or reference value can be arbitrarily chosen, or it may be chosen based on the nature of the model and the problem that the model is solving. Most often, if interpretations are generated using a validation or test set, the background input value is either a randomly selected value from the training set, or the mean value of that feature in the training set.

Therefore, for a given model $f(x)$, with output value y , SHAP creates a surrogate model $g(x')$ such that:

$$g(x') = \phi_0 + \sum \phi_i x'_i \quad (2.10)$$

where ϕ_0 is the baseline output of $f(x)$, and ϕ_i is the Shapley value for each i th feature in a set of N features [7].

If we consider the model $f(x)$ to be an estimator of an output value distribution given an input value distribution, we can discuss SHAP values in the following way as seen in [7]. The baseline output ϕ_0 in equation 2.10 is the *expectation* of the output of the function/model $f(x)$, or $E[f(x)]$. The feature attribution values, or ϕ_i , are the conditional expectations of $f(x)$, averaged over all other possible combination of input feature values, where x_i remains fixed at a particular value. This is the marginal of the feature x_i for the function $f(x)$, given feature x_i fixed at some value. This is $E[f(x)|x_i]$, or the expectation of $f(x)$ on the condition of x_i . Knowing this, we can restate equation 2.10 as [7]:

$$\begin{aligned} g(x') = & E[f(x)] + E[f(x)|x_1] + E[f(x)|x_1, x_2] \\ & + \dots + E[f(x)|x_1, \dots, x_n] \end{aligned} \quad (2.11)$$

for all N features of a data set, where $g(x')$ is a surrogate additive attribution function of function f , and x' is the simplified *feature mapping* of an input x . It can be seen that in equation 2.11, conditional independence is assumed for all features [7]. If conditional independence between features is not assumed, it would be required to know the joint distribution of all features in order to compute the conditional expectations of each feature [39]. This is not always possible, and introduces a high computational cost. Since this is the case, the majority of SHAP implementations assume conditional independence of all features in a data set [7]. This is allowable since we are evaluating the expectation of the *function* of the data, $f(x)$, and not the values of the features in the data set itself. No matter the dependencies in the input data distribution, the input features from the data distribution are assumed to be conditionally independent, and the marginals are so computed.

Conditionality may still be introduced by the non-linearity of the model being explained. To

counter this introduction of conditionality, SHAP should ideally compute Shapley values over all $N!$ orderings of features - as is done for the Shapley values computed for N players of a game with an outcome $v(S)$ [7], which ensures that the Shapley values adhere to the properties of [7]:

- **Local accuracy:** the output of the explanation model $g(x')$ matches the output of the model $f(x)$ for every input x and its mapping x' , which is a simplification of the input x that makes allowance for ‘missing’ features.
- **Consistency:** if a model changes in a way that a feature’s contribution to the output increases or remains the same in spite of the behaviour of other inputs, that feature’s attribution value cannot decrease, but must at least increase or stay the same.
- **Missingness:** if a feature that was missing in the original feature set is added to the model’s input after the model has already been trained on the original feature set, it must have no attributable impact.

The ideal computation of exact SHAP values, where Shapley values must be computed over all $N!$ orderings of features, is computationally costly and difficult. However, additive attribution methods may be used to simplify the calculation of approximate SHAP values. Therefore SHAP is often combined with other additive attribution methods to estimate Shapley values using their calculation mechanisms and modifying them in a way so that all three properties are met.

The implementation of SHAP used in this study is DeepSHAP, which takes advantage of DeepLIFT’s backpropagation method to estimate Shapley values for Deep Neural Networks (DNN).

2.5.3 DeepSHAP

DeepSHAP is an example of simplifying the Shapley value estimation process for SHAP by combining the Shapley value calculation with the backpropagation-based interpretability method DeepLIFT.

DeepLIFT calculates the contribution of the change in input feature values from a baseline to the change in output of the neural network from some baseline output. The change in input values, as well as the change in output value, is known as a *difference-from-reference*. A reference

input value is selected as some baseline or background input feature. The choice of reference value for input features depends on the nature of the problem that the model is solving, or the desired control for an interpretation.

DeepSHAP and DeepLIFT are explained with more depth in Chapters 3 and 6.

2.6 Previous work

Our aim is to determine whether an additive attribution method, when applied to a specific time-series neural network, provides an accurate explanation of the model’s decision making process, by extension determining if the model shows feature importances that correspond to the current understanding of a physical problem. Our study is closely related to several of the efforts detailed here – in the use of time-series neural networks, additive attribution methods, and the verification of known scientific knowledge.

Our study differs in that we are applying our methods to a solar physics problem (the prediction of a GMD index) at two different latitudes, which differs from the most similar study in Kratzert et al, which focused on a hydrology problem [37]. In the problem space of measuring feature importance in solar physics problems, Beukes et al. [5, 4] make use of a novel feedforward neural network architecture, the pairwise neural network, to rank feature importance when predicting a certain geomagnetic index. Our work differs here in that we are using and evaluating post-hoc feature relevance techniques, instead of attempting to develop an inherently more transparent model.

Application of additive attribution methods to time-series neural networks

Additive attribution methods have been widely applied to time-series neural networks – such as LSTM based networks [36, 40, 37], RNNs in general [32, 31], and Temporal Convolution Networks (TCNs) [41, 42]. According to Arrieta et. al, the most commonly used interpretability techniques for time-series neural networks are post-hoc feature relevance techniques [31].

LRP has been specifically extended to deal with LSTM blocks, and has been applied in a variety of problems using LSTM-based models [9]. One notable example is the use of LRP to interpret

the therapy predictions for cancer patients given by an LSTM-based model, which showed the model’s explanations coinciding with the clinical recommendations for therapy decisions [36]. LRP has further been applied to RNNs for Sentiment Analysis and other Natural Language Processing (NLP) tasks [32]. LRP has also been applied to TCNs [41].

SHAP has also been applied in this area, for example, to a TCN that estimates electric motor temperatures to validate these estimations [42]. Among other time-series problems, SHAP has also been applied to LSTM-based models to explain prognoses of remaining useful machinery life [40].

In one instance, IG has been applied to an LSTM to verify that the features used for river-runoff predictions coincide with realistic expectations [37].

Additive attribution methods for knowledge discovery

Deep learning techniques have the potential to reveal interesting patterns and correlations in data. This is an aspect that enables scientists to exploit interpretability techniques to gain useful and novel insights into problems [9, 43]. These methods of knowledge discovery has recently seen a surge of adoption in the physical and natural sciences [43]. It has been applied tasks involving in image-based, time-series, and structured data [9]. Some applications include but are not limited to:

- **Image-based tasks:** explaining diabetic retinopathy detection predictions to determine if new biomarkers for diabetic retinopathy in medical images can be discovered [44], pathology report generation to assist in cancer diagnosis in under-resourced hospitals and provide expert second opinions [45] and meteorological predictions using satellite images to potentially assist in meteorologist-driven discovery processes [46] with the application of LRP.
- **Time-series tasks:** prediction of medical conditions based on EEG time series analysis [41] with a combination of Deep Taylor Decomposition and LRP applied to a TCN, explaining unique human gait patterns found by deep learning models [47] with LRP, and showing that explanations for a rainfall-runoff model using meteorological data are consistent with established hydrological system knowledge [37] with the application of **IG!** (**IG!**) to an LSTM.

-
- **Structured data tasks:** analysing models used for molecular property and bioactivity predictions to develop new molecular structures for medications [48] with **IG!**, visualising the means by which CNNs ‘interpret’ protein-ligand complexes to aid in future CNN design for molecular modelling problems [49], and showing that a neural network trained for a quantum-chemical problem aligns with established chemical knowledge (such as the learned embeddings of chemicals exhibiting partial ordering similar to that of the periodic table) [50].

The state-of-art post-hoc attribution methods most commonly utilised are LRP, which shows up in the majority of the applications of knowledge discovery reviewed, followed by IG. In the presence of time-series data, a notable combination is LRP with an LSTM-based model (first referenced in section 2.6), which aimed to make the therapy predictions for cancer patients a transparent process, and the features used by the model were verified to be mostly in alignment with clinical guidelines [36]. IG was applied to an LSTM-based model in a meteorological knowledge discovery problem with the goal of evaluating whether the insights gathered from that analysis were in alignment with current understanding [37].

2.7 Chapter summary

In this chapter, several key concepts were introduced, such as: the influence of solar activity and the different drivers of geomagnetic storms, deep learning techniques, and XAI. We created a foundational understanding of deep learning techniques, in particular, MLPs and their functioning, and briefly introduced the use of an MLP with time-shifted inputs for sequence modeling. We provided an in-depth overview of XAI techniques, introduced our terminology in terms of these techniques, and in particular focused on introducing evaluation methods of the attributions produced by these techniques and introduced the SHAP technique that we will be using for this study.

Chapter 3

Evaluation of DeepSHAP on a synthetic data set

Here we describe DeepSHAP in detail, and demonstrate its use in a small toy problem with a synthetic data set.

3.1 Introduction

The fidelity of explanations generated by an interpretability method may be evaluated in several ways, including domain-expert evaluation and ground truth comparison, which is the focus for this study. The main method used to evaluate the fidelity of our explanations in Chapter 6 is by comparison to the expectations fostered by both the literature on the subject as well as the results of the Exploratory Data Analysis in Chapter 4.

In this chapter, however, we first describe DeepSHAP in detail and then evaluate the faithfulness of DeepSHAP explanations via ground truth comparison in the context of a *toy problem*. Ground truth comparison often involves creating a toy problem with a synthetic data set that has clear, observable patterns. A simplified version of a black-box model being studied is trained on the synthetic data set as well as it possibly can be, and the explanations generated by the

interpretability method in question are compared to the patterns observed in the synthetic data set [32, 36, 33].

DeepSHAP and SHAP were both briefly described in Chapter 2, but a further in-depth description of DeepSHAP is found in Section 3.2. The synthetic data set is motivated, described, and created in Section 3.3, the MLP model and its hyperparameter selection process is presented in Section 3.4, and the DeepSHAP explanations generated in this evaluation are presented and discussed in Sections 3.5 and 3.6, respectively.

3.2 DeepSHAP

This description of DeepLIFT and DeepSHAP are sourced from Shrikumar et. al.’s paper defining DeepLIFT [51] and Chen et al.’s work defining DeepSHAP [8, 52].

DeepSHAP is an implementation of SHAP that is based on DeepLIFT, another additive attribution method that exploits the backpropagation process of DNN models to calculate feature attributions [7, 8, 52, 51]. DeepSHAP simplifies the highly computationally expensive vanilla Shapley value estimation process by combining the Shapley value calculation with DeepLIFT’s calculation rules. DeepSHAP utilises a modified version of DeepLIFT’s reveal-cancel rule, and estimates the Shapley values layer-by-layer, and backpropagates these layer-wise results back onto the input features [8, 52].

3.2.1 Background: DeepLIFT

DeepLIFT, and by extension, DeepSHAP, calculates the contribution of the ‘difference-from-reference’ of input features values from some reference values to the ‘difference-from-reference’ of the model’s output from the output that would have been produced with those reference input feature values. The choice of reference values for input features depends on the nature of the problem, or the desired ‘control’ for the feature attributions being calculated. Typically, as with DeepSHAP, the reference values are sampled from the training set, where the feature attributions are calculated using data from the validation or evaluation sets.

DeepLIFT assigns a contribution, $C_{\Delta x_i \Delta t}$, of each difference-from-reference for feature x_i (Δx_i)

to a difference-from reference for the output value t (Δt).

These contributions follow a ‘summation-to-delta’ property (Equation 3.1) that states that each feature contribution adds up to the difference in the model output from its baseline output [7].

$$\sum_i C_{\Delta x_i \Delta t} = \Delta t \quad (3.1)$$

For any contribution $C_{\Delta x \Delta t}$, there exists a *multiplier* that is consistent with the summation-to-delta property:

$$m_{\Delta x \Delta t} = \frac{C_{\Delta x \Delta t}}{\Delta x} \quad (3.2)$$

This multiplier is the contribution of Δx to Δt with respect to Δx . The multiplier $m_{\Delta x \Delta t}$ can also be viewed as the partial derivative $\frac{\partial t}{\partial x}$ over finite differences. This enables us to use the *chain rule for multipliers*.

The *chain rule for multipliers* states that, assuming that we have a neural network with a set of inputs $x_1, \dots, x_n$, a hidden layer with nodes $y_1, \dots, y_m$ and some target output node t ; given values for $m_{\Delta x_i \Delta y_j}$ and $m_{\Delta y_j \Delta t}$, the following:

$$m_{\Delta x_i \Delta t} = \sum_j m_{\Delta x_i \Delta y_j} m_{\Delta y_j \Delta t} \quad (3.3)$$

is consistent with the summation-to-delta property.

Given the above definitions for contributions, multipliers, and the summation-to-delta property, we are able to compute the contributions of a set of input features $C_{\Delta x \Delta t}$ to the output of any neural network.

However, it cannot be assumed that all these contributions will be positive, or that any contribution will be completely positive or completely negative. In order to determine the positive and negative components of the contribution of a particular *difference from reference* Δy to a Δt , this Δy is divided into its positive and negative components:

$$\Delta y = \Delta y^+ + \Delta y^- \quad (3.4)$$

Similarly, the contribution $C_{\Delta y \Delta t}$ is composed of two components:

$$C_{\Delta y \Delta t} = C_{\Delta y^+ \Delta t} + C_{\Delta y^- \Delta t} \quad (3.5)$$

There are three rules by which these positive and negative components can be handled: the linear rule, the rescale rule, and the reveal-cancel rule. However, since DeepSHAP adapts the reveal-cancel rule, we describe only the reveal-cancel rule.

The reveal-cancel rule calculates the positive and negative components of $C_{\Delta x \Delta y}$ separately, by first considering the impact of the positive component of Δx alone, and then the impact of the negative component of Δx alone. This ensures that the positive and negative components of $C_{\Delta x \Delta y}$ are accurately calculated. This system of individual consideration is quite similar to the Shapley value calculation as shown in Equation 2.8 in Chapter 2.

DeepLIFT defines the reveal-cancel rule as follows: If we are given a nonlinear neuron $y = f(x)$, with input value x , reference input value x^0 , and the positive and negative components of difference-from-reference Δx , (Δx^+ and Δx^-), then the positive and negative components of $C_{\Delta x \Delta y}$ (Δy^+ and Δy^-) are computed:

$$\begin{aligned}\Delta y^+ &= \frac{1}{2}(f(x^0 + \Delta x^+) - f(x^0)) \\ &\quad + \frac{1}{2}(f(x^0 + \Delta x^- + \Delta x^+) - f(x^0 + \Delta x^-)) \\ \Delta y^- &= \frac{1}{2}(f(x^0 + \Delta x^-) - f(x^0)) \\ &\quad + \frac{1}{2}(f(x^0 + \Delta x^+ + \Delta x^-) - f(x^0 + \Delta x^+))\end{aligned}\tag{3.6}$$

The multipliers for the positive and negative components of the contribution $C_{\Delta x \Delta y}$ are then defined as:

$$\begin{aligned}m_{\Delta x^+ y^+} &= \frac{C_{\Delta x^+ \Delta y^+}}{\Delta x^+} = \frac{\Delta y^+}{\Delta x^+} \\ m_{\Delta x^- y^-} &= \frac{C_{\Delta x^- \Delta y^-}}{\Delta x^-} = \frac{\Delta y^-}{\Delta x^-}\end{aligned}\tag{3.7}$$

3.2.2 DeepSHAP definition

Observing Equation 3.6, we can see a similarity to how Shapley values are defined in Equation 2.8 in Section 2.4.1. This is essentially computing the marginal contributions of Δx^+ and Δx^- to Δy^+ and Δy^- , and, by extension, to Δy . For this reason, DeepLIFT's reveal-cancel rule is able to stand in as a more computationally efficient solution for Shapley values in the context of a DNN model. Looking at Equation 3.7, we can see that $C_{\Delta x^+ \Delta y} = \Delta y^+$ and $C_{\Delta x^- \Delta y} = \Delta y^-$, so Δy^+ and Δy^- as defined in Equation 3.6 are the *contributions* of the positive and negative components of an input x to the output y .

The adaptation of this rule to DeepSHAP computes the contribution of an input value x_i to the output value y of a node (assumed to be passed through an activation function f) in two parts. First, it will consider the result of a linear function $h = \sum x_i w_i$ as a single input to the activation function f . This input h is divided into its positive and negative components h_+ and h_- based on whether or not $w_i \Delta x_i > g$ for each x_i , with w_i being the weight for that input value, and with the threshold g typically being 0:

$$h = \sum w_i x_i = h_+ + h_- \quad (3.8)$$

$$\begin{aligned} h_+ &= \sum_i w_i x_i [w_i \Delta x_i > g] \\ h_- &= \sum_i w_i x_i [w_i \Delta x_i \leq g] \end{aligned} \quad (3.9)$$

The baseline value for h , h^0 , is equal to:

$$h^0 = \sum w_i x_i^0 = h_+^0 + h_-^0 \quad (3.10)$$

where h_+^0 and h_-^0 are calculated as follows:

$$\begin{aligned} h_+^0 &= \sum_i w_i x_i^0 [w_i \Delta x_i > g] \\ h_-^0 &= \sum_i w_i x_i^0 [w_i \Delta x_i \leq g] \end{aligned} \quad (3.11)$$

with the difference-from-reference for h contributing to Δy being:

$$\begin{aligned} \Delta h &= h - h^0 \\ \Delta h_+ &= h_+ - h_+^0 \\ \Delta h_- &= h_- - h_-^0 \end{aligned} \quad (3.12)$$

Thus, considering h as the new input to the activation function f , we calculate the Shapley values of h_+ and h_- 's contributions to Δy using an adaptation of DeepLIFT's reveal-cancel rule as follows:

$$\begin{aligned}
\phi_{h_+}(\Delta y) &= \frac{1}{2}[(f(h_+ + h_-) - f(h_+^0 + h_-)) + \\
&\quad (f(h_+ + h_-^0) - f(h_-^0 + h_+))] \\
\phi_{h_-}(\Delta y) &= \frac{1}{2}[(f(h_+ + h_-) - f(h_+ + h_-^0)) + \\
&\quad (f(h_+^0 + h_-) - f(h_+^0 + h_-^0))]
\end{aligned} \tag{3.13}$$

These Shapley values are then back-propagated linearly to the input value x_i to compute the exact Shapley value of its contribution to Δy . The Shapley value of the positive or negative contribution calculated in Equation 3.13 is propagated depending on whether $w_i \Delta x_i > g$:

$$\phi_{x_i}(\Delta y) = \begin{cases} \frac{\phi_{h_+}}{\Delta h_+} w_i \Delta x_i & \text{if } w_i \Delta x_i > g \\ \frac{\phi_{h_-}}{\Delta h_-} w_i \Delta x_i & \text{otherwise} \end{cases} \tag{3.14}$$

This process is repeated for each node in a layer, for each layer, with the resulting contributions are propagated back onto the input features of the neural network.

3.2.3 The selection of reference values

DeepSHAP makes use of a theorem proposed by Chen et al. [8] to justify its use of background values. This theorem states that the mean of SHAP attribution values for a single sample calculated with each background value approximates the ‘true’ SHAP value for a given distribution [8]. This method of calculating SHAP attribution values relative to a background distribution, while being shown to be less biased than calculating SHAP values based on an average of a background distribution or a single reference value, is computationally intensive. Typically, a limited number of background samples are randomly selected from the training set, while calculating attributions for samples from the evaluation set.

3.3 Synthetic data set definition

Here we aim to create a synthetic data set with clear feature weights that can be easily mapped to the general form of additive feature attribution methods (see Equation 2.8 in Section 2.5.1). This can be accomplished with a simple polynomial regression data set, which is defined with Equation 3.15. This mapping function allows for negative and positive contributions of the

features to y , and does not include any interacting features, simplifying the comparison to the DeepSHAP attributions that will be calculated:

$$4a^4 + 3b^3 + 2c^2 + d = y \quad (3.15)$$

The mapping function of Equation 3.15 creates a data set with input features $\{a, b, c, d\} \in \mathbb{R}$, and output target $y \in \mathbb{R}$. The values for each of the input features are selected separately from a random uniform distribution $\mathcal{U}[-1, 1]$. Each of the terms in this mapping function corresponds to a single input feature, and each term, when plotted with the input feature values monotonically increasing, will have its own characteristic curve. This provides a clear basis for comparison with DeepSHAP attribution values, which should, when plotted with the corresponding input feature values monotonically increasing, match the characteristic curves of the synthetic feature's mapping function term. The values of the output targets are calculated for each sample by passing the input feature values through Equation 3.15. In this way, we generate a data set of 50 000 samples, from which we select 28 000 samples for a training set, 7000 samples for a validation set, and 15 000 samples for an evaluation set.

Figure 3.1 shows the curve of each input feature's term with the corresponding input feature values arranged from -1 to 1, and the output values arranged from smallest to largest value, for the training set.

Since DeepSHAP is an implementation of SHAP, it is an additive attribution method. This means that the attributions for each feature add up to the difference-from-reference for each output sample, such as in Equation 3.1. If we were to simplify the symbol for the contribution of feature i to the output y as ϕ_i , then Equation 3.1 can also be written as:

$$\sum_{i=1}^N \phi_i = y - \phi_0 \quad (3.16)$$

where N is the number of features in the feature space, ϕ_i is the contribution input feature i , and $y - \phi_0$ is the output target's 'difference-from-reference'.

Considering Equation 3.16, and Equation 3.15 we could expect the attribution values generated by DeepSHAP to correspond to Equation 3.17:

$$\phi(a) + \phi(b) + \phi(c) + \phi(d) = y - \phi_0 \quad (3.17)$$

where $\phi(a)$, $\phi(b)$, $\phi(c)$ and $\phi(d)$ are the attributions of each feature, and ϕ_0 is the reference output value.

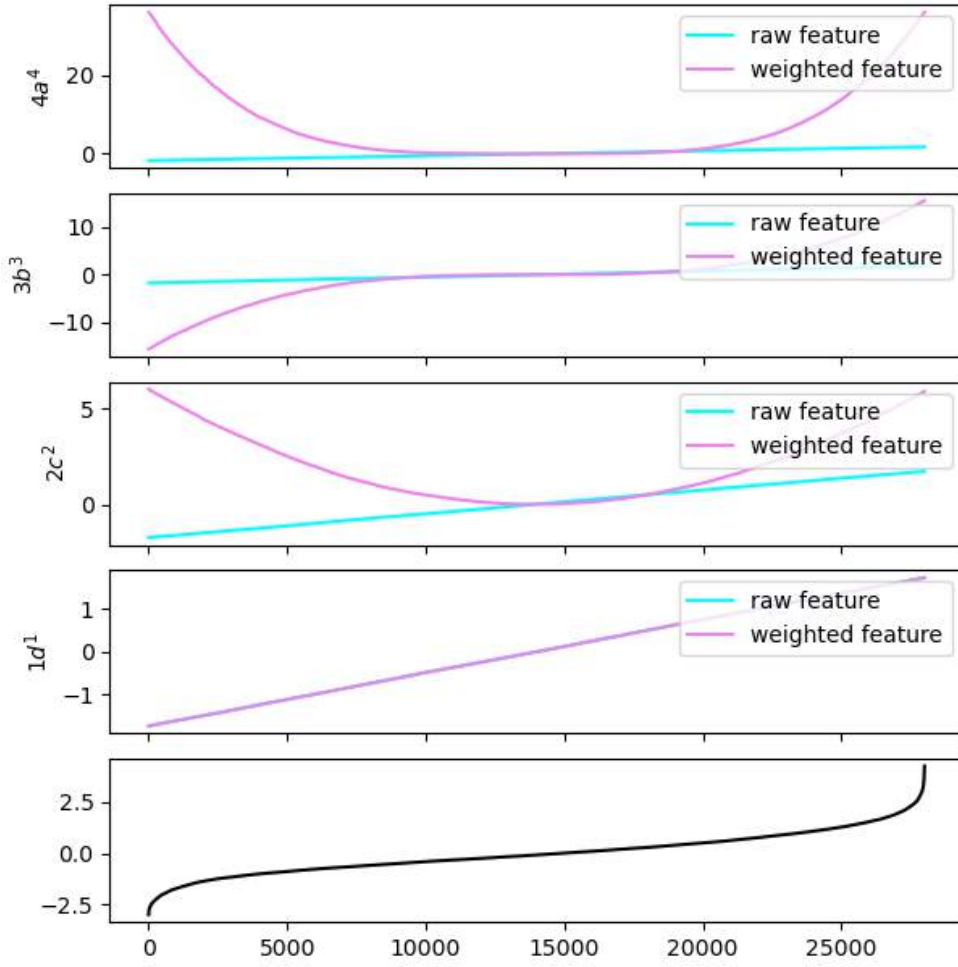


Figure 3.1: Input feature values, ‘weighted’ input feature values, which are each input feature value as passed through Equation 3.15, and the output values of the synthetic dataset. Each of the curves depict the weighted input feature values arranged from smallest input feature value to largest input feature value. The output value curve is arranged from smallest to largest output value.

3.4 Model selection and training

An MLP model is selected by conducting a hyperparameter search with the Sequential Model Algorithm Configuration 3 package¹, a configuration tool that enables automatic optimisation of hyperparameters for a model using a combination of Bayesian Optimisation (BO) and an intensification method [53].

The Sequential Model Algorithm Configuration 3 (SMAC3) package offers several different im-

¹<https://github.com/automl/SMAC3>

plementations, which are described in its internal documentation as facades. The SMAC3 facade that we use is the SMACM4F facade, which approximates ‘expensive’ functions such as deep learning algorithms with a cheaper proxy. It does this using a budget, or a maximum limit that is set to the value of any hyperparameter that increases the expense of the function’s computation [53]. We set a budget for total training epochs.

This SMAC3 facade uses a random forest to create several configurations with a random initial design that are trained for only a fraction of the training epoch budget. Further, BO with hyperband is used to intensify the hyperparameter search with successive halving to eventually fit a ‘final’ configuration on the maximum training epoch budget after a sufficient set of observations (possible configurations), which can be specified beforehand, have been trained [53].

Successive halving is the continuous narrowing of possible configuration candidates by successively dropping half of the ‘lesser performing’ configuration possibilities from a previous training round, re-selecting configurations from the new set of possibilities, and training for double the previous fraction of the training epoch budget. Eventually, a very small set of possibilities remain in the final training round, and these final configurations are trained to the maximum value of the training epoch budget.

A maximum number of observations, the number of intensification rounds, a minimum budget, and an objective (the desired metric to be minimised) is set in a SMAC3 configuration file along with a possible set of hyperparameter values that can be selected. These settings and configuration file is known as a configuration space. The range of possible hyperparameter values are expressed as distributions. This configuration space can contain hyperparameter value distributions that are either categorical or continuous (random uniform, random normal), and provides great flexibility when searching for a good model configuration.

When a SMAC3 scenario is run, all possible model configurations selected from the configuration space and their performance results (the value of the objective metric) at the completion of their training is saved in a run history file. A run history file is a text file that records the performance results obtained at each observation made and its corresponding configuration during the scenario². This information may be used to determine and select the best model configuration from that scenario.

²SMAC3 documentation: <https://automl.github.io/SMAC3/main/>

3.4.1 Model selection procedure

This model selection procedure aims to set a standard for the hyperparameter selection process for all MLP models trained during this study. This procedure involves setting up a SMAC configuration space with a certain number of maximum observations, intensification rounds, and a minimum budget. The objective metric in every case will be best validation loss. Generally, the hyperparameters that are varied when selecting hyperparameter configurations for an MLP are layout (i.e., number of layers and number of nodes per layer), learning rate, weight decay, and batch size. Factors such as the choice of optimisation function, loss function and activation functions are fixed to Adam, MSE loss, and ReLU respectively. The application of bias nodes (or not) is also decided and kept constant throughout the model selection process.

Once the configuration space is set up, a SMAC scenario is run. Once the scenario's run is complete, the best hyperparameters are selected based on lowest validation loss, and the run is summarised in a scenario plot. The configuration space for this model selection is set with a maximum of 50 observations, 3 intensification rounds, a minimum budget of 10 epochs, and an objective of best validation (MSE) loss. The maximum budget of epochs is set as 100 epochs. Early stopping is enabled without patience, and MSE loss is selected as the loss function, Adam is selected as the optimisation function, and a ReLU activation function is used after each hidden layer. Bias nodes are applied to all layers. The hyperparameters that were searched in this scenario, along with their values and distributions, are listed in Table 3.1.

Table 3.1: Hyperparameter search configuration space for MLP model selection.

Hyperparameter	Range	Selection distribution
Layout (depth x width)	2x100, 3x100	Categorical
Batch size	32, 64	Categorical
Learning rate	0.001 to 0.01	Random uniform
Weight decay	0 to 0.01	Random uniform

A scenario plot showing the best final validation loss, weight decay, learning rate, batch size, and layout at each run is found in Figure 3.2. The best configuration is seen at run 50, with a layout of 2x100 nodes, a batch size of 32, a weight decay of 0, a learning rate of 0.002, and a validation loss of 0.00016.

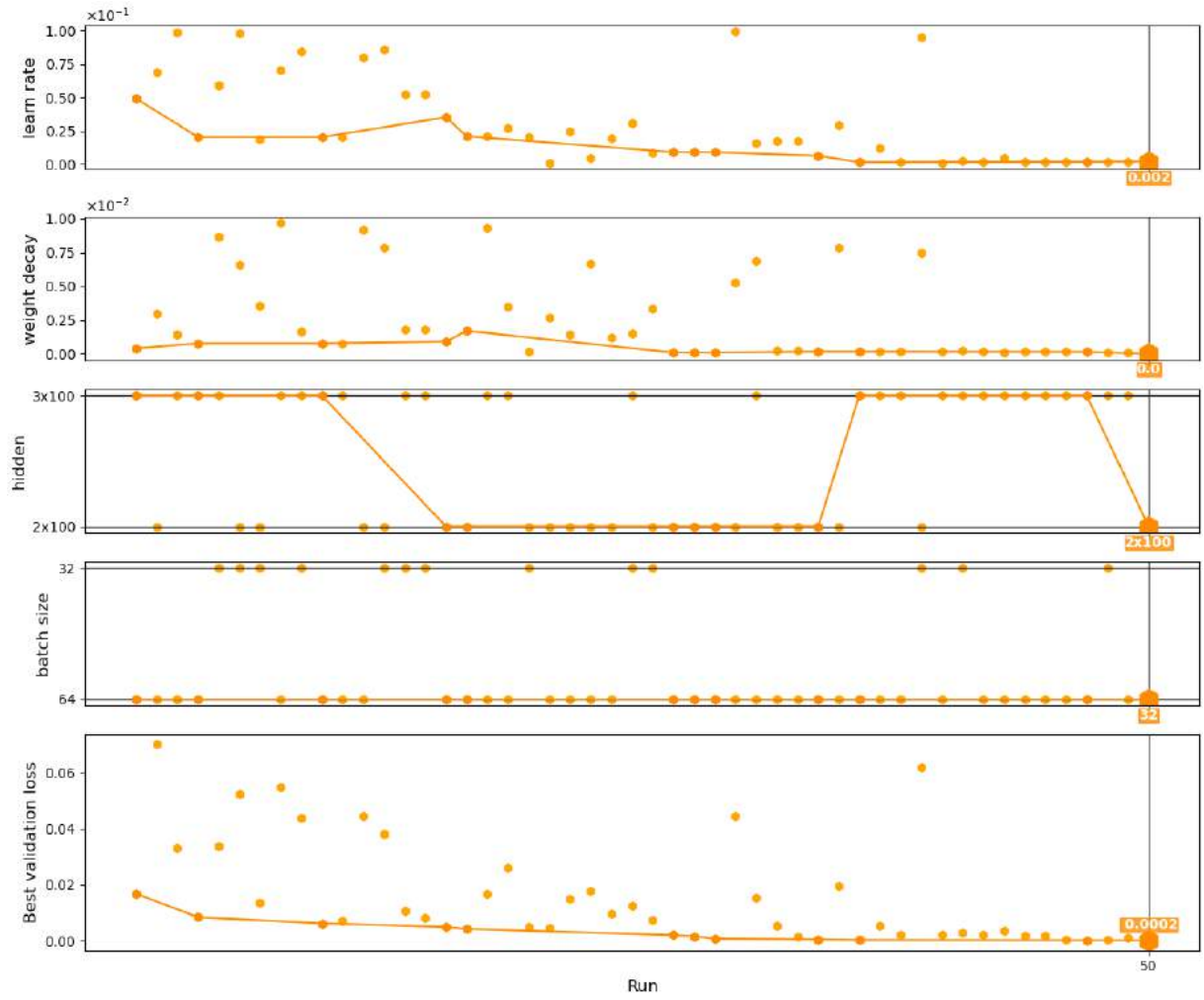


Figure 3.2: Scenario plot for hyperparameter configuration selection on synthetic dataset.

3.4.2 Model performance

Outputs are generated using the input features from the training, validation, and evaluation sets. These predicted output values, as well as the target output values, are sorted from smallest to largest according to the target output values. Thus sorted, predicted output values and target output values for the train and evaluation sets are shown in Figure 3.3. The Root Mean Squared Error (RMSE) and Pearson's r is annotated on each plot.

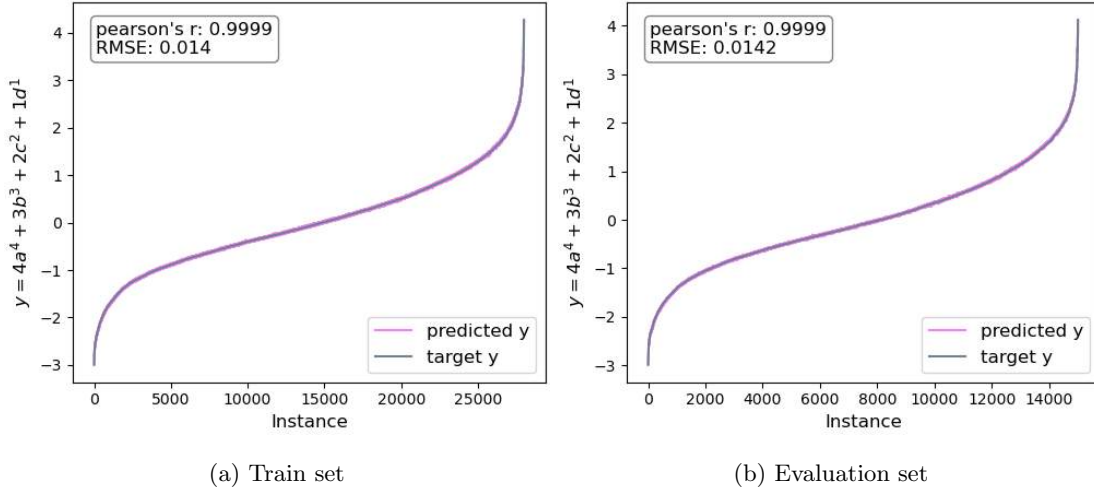


Figure 3.3: Output values ordered by output value from smallest to largest vs. corresponding target values for train and evaluation sets.

3.5 DeepSHAP attributions

A random selection of 1 000 training samples are taken for the reference set, and DeepSHAP attributions are generated for each feature using the evaluation set. The outcomes of this are shown in Figures 3.4 and 3.5.

In Figure 3.4, attribution values are sorted according to the value of their corresponding input feature from smallest to largest. The sum of all attributions for each output sample is similarly ordered according to the target output values from smallest to largest. Additionally, the transformed input feature values as passed through equation 3.15 are sorted according to input feature value for comparison to the attribution values.

In Figure 3.5, the input feature values, transformed input feature values, feature attributions, as well as the sum of all attributions for each output sample are sorted by target output value from smallest to largest.

In both Figures 3.4 and 3.5 the DeepSHAP attributions (green) correspond very closely to the transformed input features (violet). The difference in magnitude is due to the fact that the output targets were standardised with z-norm standardisation prior to training the model, and the individual effects on the scale of the transformed input features as a result would not

translate to the scale of the DeepSHAP attributions directly.

It can be inferred, however, that since the sum of the attributions very closely correspond to the values of the predicted values, that the DeepSHAP attributions would have corresponded similarly closely in magnitude to the transformed input features were these scaling effects directly comparable.

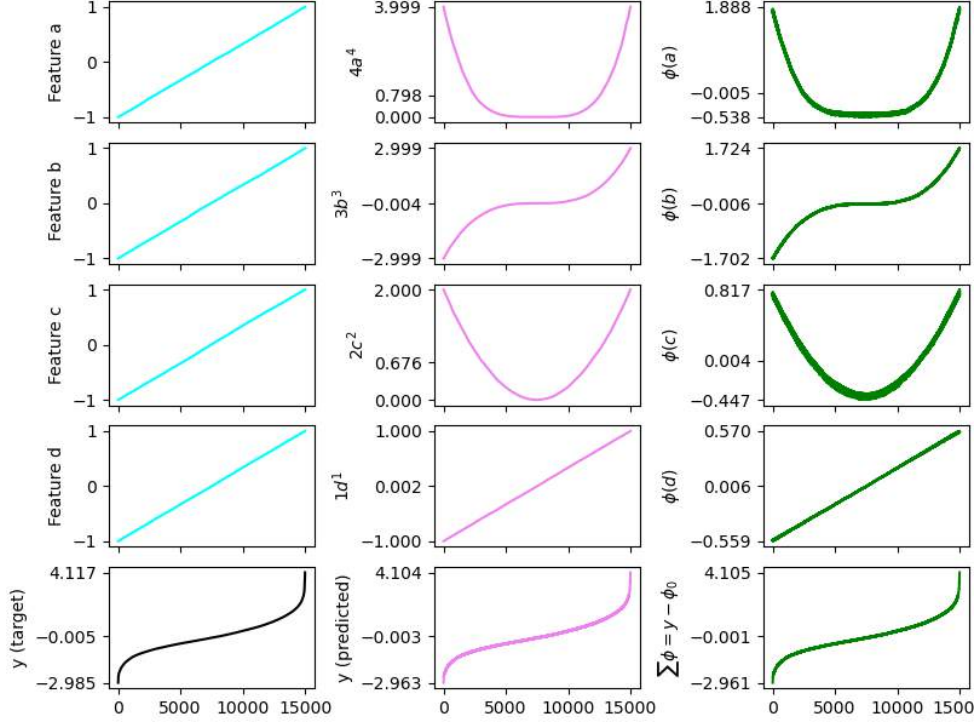


Figure 3.4: Compilation of ordered input features and corresponding transformed input features vs their respective DeepSHAP attributions.

In particular, when observing Figure 3.5, we can see that when the feature values, feature mappings, and attributions are arranged according to output value, feature *b* has a very strong impact on the output's shape. Observing the DeepSHAP attribution values when ordered according to output value, features *a*, *b* and *c* contribute most to the output's magnitude.

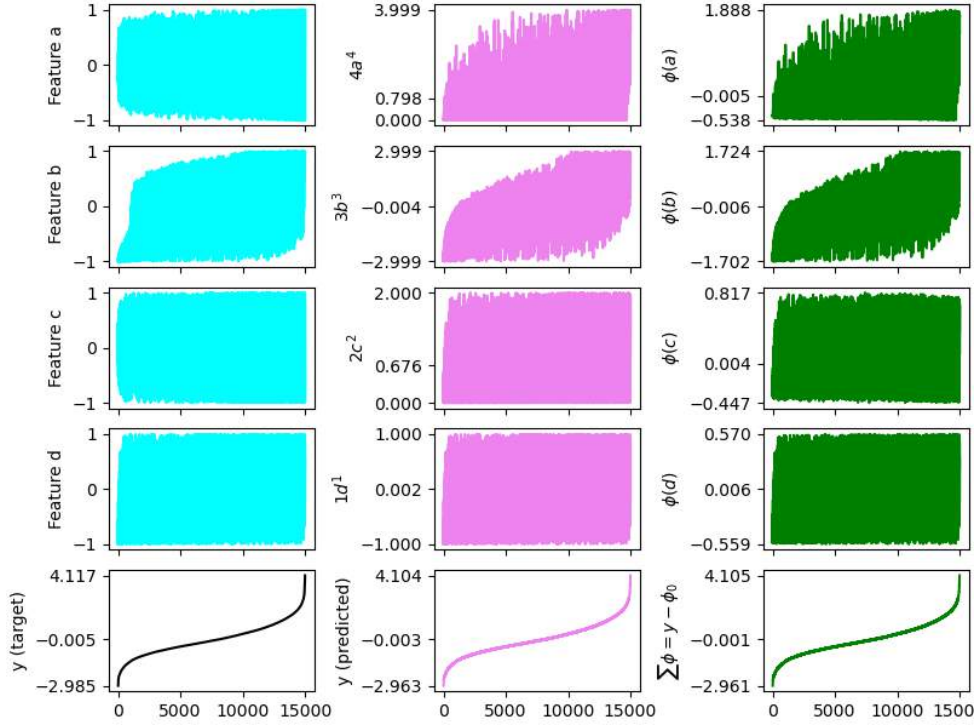


Figure 3.5: Compilation of input feature values, transformed input feature values, and each input feature’s respective DeepSHAP attributions as sorted from smallest to largest according to target output value.

3.6 Discussion

The results shown in Section 3.5 show that a simple MLP model, sufficiently well-trained, is able to pick up on feature patterns and dependencies inherent in the dataset, without having access to these feature patterns explicitly. It is also shown by virtue of $\sum \phi \approx y$ as well as the shapes of each feature’s attribution in Figure 3.4 that DeepSHAP was able to reveal that this model uncovered the underlying patterns, and that the attributions that DeepSHAP was able to calculate for each feature is reasonably faithful to reality.

This is however, a case where the model performed close to perfectly (with a 0.99+ accuracy). In Appendix A.1, a similar model training process to that in Section 3.4 is discussed, where bias nodes are not applied to any of the layers. This resulted in a MLP model of similar layout, learning rate, and weight decay that performed significantly worse than the model selected in this chapter.

It can be inferred that this model would be less able to accurately approximate the feature

patterns, and that DeepSHAP would be able to pick up on these difficulties. In Appendix A.1, we see an example of this happening: the model there is not as accurate as the model discussed here, and as a result, its learning of the inherent patterns is limited. The DeepSHAP attributions still indicate approximate learning of these patterns with a decent level of fidelity, however, the attributions are quite noisy, as would be expected from a model which did not perfectly capture the underlying data.

3.7 Chapter summary

The goal of this chapter was to introduce the use of the DeepSHAP interpretability technique, and demonstrate its use in a ground truth evaluation toy problem. We detailed the inner working of DeepSHAP in the context of its use of DeepLIFT calculation rules, its use of reference values taken from a background distribution, and demonstrated a data set selection, model training, and attribution calculation pipeline that we will be using in the remainder of our study.

We saw in Figures 3.4 and 3.5 in Section 3.5 that when an MLP model, trained to high accuracy on a clearly explainable synthetic data set, recognises and uses underlying feature importances that it did not have access to prior to training, and that DeepSHAP was accurately able to attribute and show these feature importances.

Chapter 4

Data set definition

In this chapter we define the data set which is used in this study. Its source data, its pre-processing steps, and its derivations are described. Exploratory data analysis is conducted to characterise the data set.

4.1 Introduction

In this chapter, we describe the data set used in this study, which we call the Hermanus-Abisko Storms (HAS) data set, to compare solar wind drivers with the e_h geomagnetic disturbance index [10] derived from data measured at two geomagnetic observatories: one at high latitude (Abisko, Sweden: 68.358° N, 18.823° E) and one at low middle latitude (Hermanus, South Africa: 34.43° S, 19.23° E). We discuss the source data, data preparation process, parameter selection, derivations, and characteristics of the final data set.

Section 4.2 outlines the data preparation process, including a description of the source data, pre-processing steps, geomagnetic storm identification, and e_h index calculation. In Section 4.3, we conduct EDA before further modelling and interpretation done in Chapters 5 and 6. This involves creating and analysing empirical distributions of all parameters and output targets, fitting theoretical distributions, and calculating correlations between parameters and e_h at the

high and low latitude stations. Empirical and theoretical distributions of the parameters and output targets are studied to gain an understanding of what they look like, and also to potentially adjust parameters for modeling purposes according to said distributions. Correlations between parameters were calculated so that especially strong relationships could be identified, as this could later influence interpretations of the models.

4.2 Data set preparation

This data set is obtained and derived from two data sources: solar wind plasma and Interplanetary Magnetic Field (IMF) data obtained from the High-Resolution OMNI (HRO) data set¹, and geomagnetic observations from HER and ABK, hosted by Intermagnet².

Both of these sources contain data that spans several decades across several instruments. This represents aging data formats, evolving storage requirements, and disparate measurement procedures. In addition, the 8 spacecraft orbiting the L1 Lagrange point responsible for a majority of the measurements from which the HRO data set is derived are prone to outages. This results in a large amount of missing values present in the derived solar wind parameters. For this reason, extensive pre-processing is required to create a suitable data set for this study.

4.2.1 Source data

The HRO data set provides solar wind plasma and IMF parameters which are used as input features in our data set. Ground-based geomagnetic observation data measured at HER and ABK are used to derive the target output value (e_h , derived by Wintoft et. al [10]) for our data set as described in Section 4.2.3. All source data is available at a resolution of 1 minute, which is consequently the resolution of our data set.

The measurements of the solar wind parameters are already shifted forward in time to when the Bow-Shock Nose (BSN) of the solar wind makes contact with the Earth’s magnetic field as they are obtained from the HRO data set. This allows the input features to be directly associated with ground-based geomagnetic effects represented by e_h occurring at HER and ABK in real

¹<http://omniweb.gsfc.nasa.gov>

²<https://intermagnet.github.io/>

time.

The input features of our data set consist of 10 solar wind parameters. These parameters are detailed in Table 4.1. This table includes the names, symbols, units, and some descriptive statistics of each parameter. Note that the descriptive statistics are based on the input features as extracted for the years 1996 to 2019, with geomagnetic storms identified by the method described in Section 4.2.4.

Table 4.1: Description of input features. Name, symbol, unit, min, max, and median of each input feature are listed, based on isolated storm data for the years 1996 to 2019.

Name	Symbol	Unit	Min	Max	Median
Solar wind speed	V_{sw}	km/s	138.64	1076.00	509.48
IMF magnitude	$ B $	nT ³	−3.49	69.79	8.63
IMF-x	B_x	nT	−52.35	39.95	0.05
IMF-y	B_y	nT	−46.84	49.03	0.24
IMF-z	B_z	nT	−93.67	69.70	−1.19
Proton density	N_ρ	#/cc ⁴	0.00	84.66	7.25
Flow pressure	P_d	nPa	0.00	96.20	3.05
Velocity-x	V_x	km/s	−1073.00	−93.40	−507.53
Velocity-y	V_y	km/s	−864.90	384.07	4.28
Velocity-z	V_z	km/s	−355.73	836.61	−1.57

The solar wind data (specifically, the 10 chosen input features) consists of three major components: velocity/speed components, particle components, and IMF components.

The velocity components of the solar wind plasma are measured according to GSE coordinates, which are centered at the Earth and have the X axis pointing towards the Sun, the Z axis pointing northward (geographic north), and the Y axis in the ecliptic plane completing the cartesian system. This coordinate system is contextualised in Figure 4.1. A solar wind speed scalar (V_{sw}) is also included.

The particle components, N_ρ and P_d , measure particle density and flow pressure of protons carried by the solar wind.

The IMF components are measured in both GSE and Geocentric Solar Magnetic (GSM) coordinates. GSM coordinates are similarly defined to GSE coordinates, except that the Z axis is aligned to magnetic north pole instead of the geographic north pole as is done with GSE coordinates. We opt to use IMF components as measured in GSM coordinates for this study.

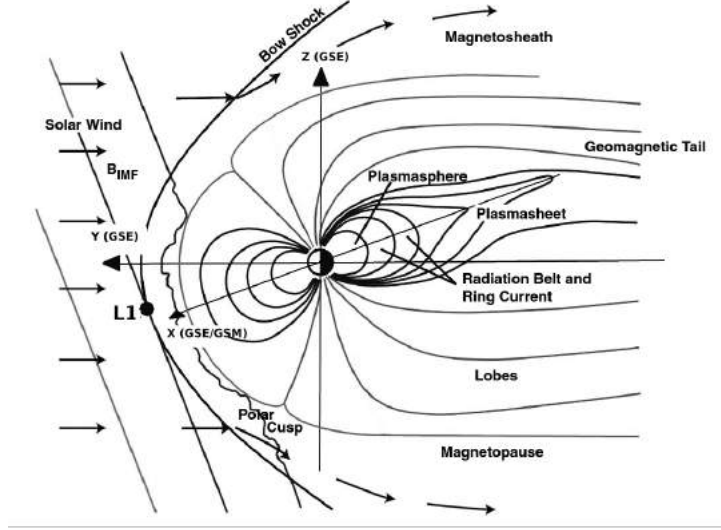


Figure 4.1: Geocentric Solar Ecliptic (GSE) coordinate system shown in the context of the solar wind, magnetosphere, and BSN. Adapted from Moldwin [11]

An IMF magnitude quantity ($|B|$) is also included.

4.2.2 Initial pre-processing

Since the HRO, HER, and ABK data sets are based on measurements from a wide variety of instruments over several decades, missing values are often represented with the value that can be stored and/or handled by the instrument in question. These values vary from parameter to parameter, and vary both between instruments and over time. While HRO provides a meta-data file for each year of data downloaded containing a list of missing value placeholders for each parameter, Intermagnet data does not have such a luxury, and missing value placeholders had to be manually identified. The missing value placeholders and the data points that contain them need to be identified for each parameter and replaced with NaN values before any interpolation or missing value handling can take place.

There were a total of 18 different missing value placeholders identified for the 10 input parameters in the HRO data set for the time-span from 1996 to 2019, and 3 different missing value placeholders were identified for the ABK and HER data sets in that period.

Once each parameter's placeholder values are replaced with NaN values, they are then interpolated with a second-order polynomial up to a gap size limit of 10 minutes. The second-order

polynomial is selected to provide more flexibility than linear interpolation, as well as for being less computationally intensive than spline interpolation. This gap limit size was selected after testing the ratio of points gained over simply dropping all samples containing NaN values for a set of gap limit sizes. In the majority of cases, interpolating over gaps larger than 10 minutes provided diminishing returns. Interpolation over too large of a gap would risk suppressing true patterns in the underlying data and introduce unwanted artefacts. More details on this test and its results are found in Appendix A.2.2.

4.2.3 Wintoft index calculation

The e_h index, defined in [10], is utilised as a proxy index for GIC, since it approximates the electric field induced by geomagnetic disturbances (see Wintoft et. al [10] and references therein).

The e_h index is derived using the northerly and easterly components of the horizontal geomagnetic field (B_h) that is measured at either ABK or HER. In the case that it is not measured directly, B_h can be calculated using its northerly B_x and B_y easterly components [10]:

$$B_h = \sqrt{B_x^2 + B_y^2} \quad (4.1)$$

The derivative of B_h with respect to time t , d_h , is given by:

$$d_h = \frac{dB_h}{dt} \approx \Delta B_h = B_h(t+1) - B_h(t) \quad \forall t \quad (4.2)$$

e_h is calculated using the following formula [10]:

$$e_h = \sqrt{\left(\frac{dB_x}{dt}\right)^2 + \left(\frac{dB_y}{dt}\right)^2} \quad (4.3)$$

According to the method proposed by Wintoft et al. [10], e_h is passed through a time window of length N and a scaling factor of value a as is shown in Equation 4.4.

$$e_h^{(a)}(t) = \left(\frac{1}{N} \sum_{t'=t+1}^{t+N} e_h^a(t')\right)^{\frac{1}{a}} \quad (4.4)$$

A scaling factor of 1 corresponds to simply obtaining the average value for e_h in a time window of N minutes, where a scaling factor tending to ∞ results in the maximum value for e_h in a time window of N minutes [10]. We use $a = 1$ and $N = 30$, obtaining the 30-minute average of e_h at each time step.

e_h is proportional to $|d_h|$. For this reason, d_h and the resultant $|d_h|$ is a good comparative check to validate the calculation of e_h . To validate this calculation of e_h as an example, $|d_h|$ was calculated alongside e_h for the same time period. The two are compared in Figure 4.2 at corresponding time-steps. Since e_h is calculated so that its 30-minute average at each time step is used, $|d_h|$ is similarly passed through Equation 4.4 with $a = 1$ and $N = 30$ to retrieve its 30-minute average at each time step.

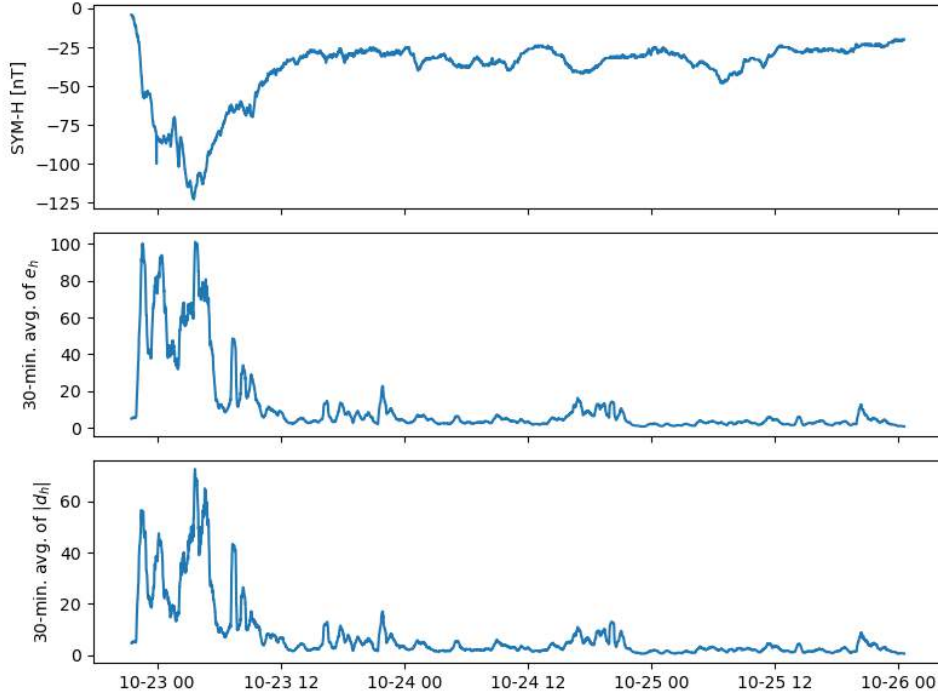


Figure 4.2: SYM-H, e_h , and $|d_h|$ at ABK for a severe storm in 1996. e_h and $|d_h|$ are calculated with a time window N of 30 minutes and a scaling factor a of 1. e_h is proportional to $|d_h|$ [10].

4.2.4 Storm identification

The symmetric-H index (SYM-H) is a geomagnetic index that quantifies the symmetric disturbance in the horizontal (H) component of the geomagnetic field at a number of ground-based magnetic field measurements taken at several locations around the world [54]. This index pro-

vides a global indication of the effect of geomagnetic disturbance on the ground, and may be used to determine whether or not a solar storm is occurring. The SYM-H index is available in the HRO data set, with the unit nT.

Geomagnetic storms are typically characterised by three phases: the storm onset, the main phase, and the recovery phase. Figure 4.3 shows SYM-H during a severe storm with these phases marked using the method described in [4], alongside additional parameters B_z and V_{sw} . We use the term ‘onset phase’ to refer to the brief period leading up to the main phase, up to the point when $\text{SYM-H} < -20\text{nT}$ at the start of the main phase.

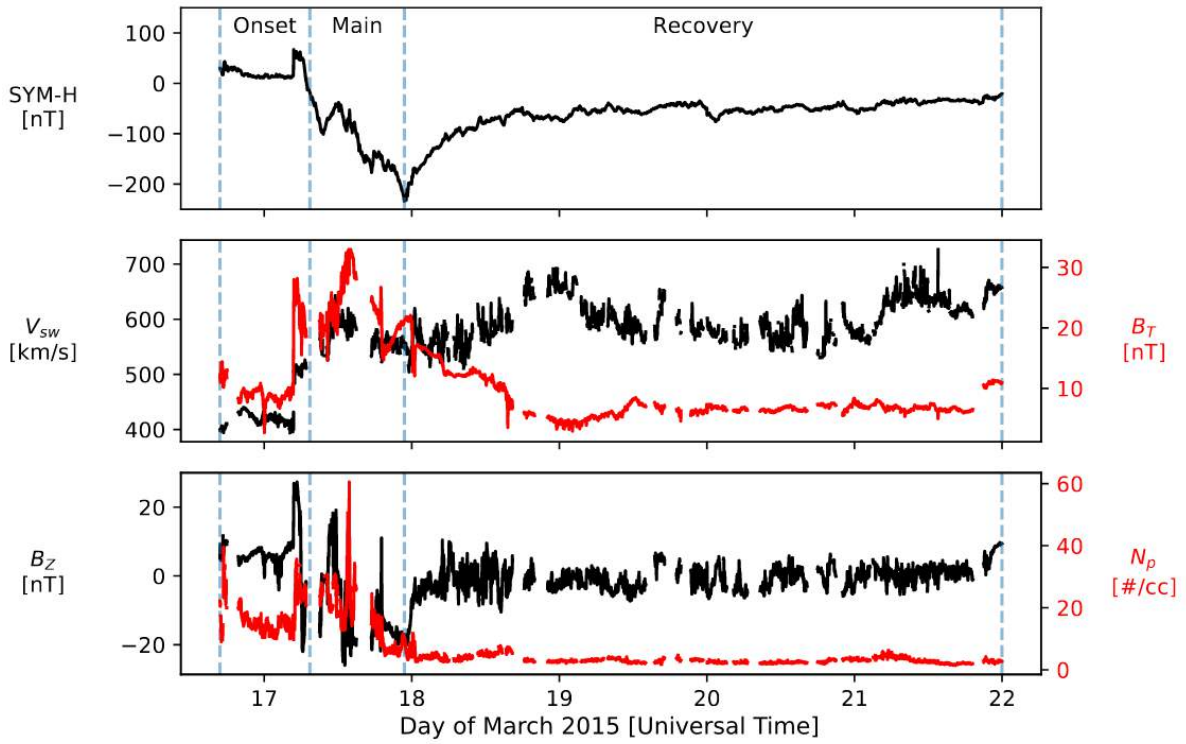


Figure 4.3: SYM-H, B_z , and V_{sw} during a severe storm in 2015 with the onset, main, and recovery phases annotated. From Beukes et al. [4], used with permission.

SYM-H typically hovers around zero during non-stormy (quiet periods) with minor fluctuations caused by the ambient solar wind or other transients such as high speed streams. During the **onset phase** of most CME-driven storms, SYM-H peaks in intensity after a period of build-up (called the sudden commencement), and then immediately afterward plunges below -20 nT to an extreme minimum value during the **main phase** of the storm. Once SYM-H reaches the minimum extreme value for that storm, and the geomagnetic event passes, it enters the

recovery phase. This phase is where it slowly recovers to a moderate -20 nT over several hours, or sometimes, days.

Table 4.2: The extreme minimum values of SYM-H for severe, moderate, and mild storms as seen in [13]

Event intensity	SYM-H minimum threshold
Quiet/recovery (no storm)	-20 nT
Mild	-50 nT
Moderate	-80 nT
Severe	-100 nT

We use the procedure described in [1] to identify storms of severe intensity, and adapt it to also select storms of mild and moderate intensity, from commencement until SYM-H recovers to above -20 nT.

A mild event has a minimum threshold SYM-H value between -50 nT and -80 nT, a moderate event a minimum threshold value between -80 nT and -100 nT, and a severe event a minimum threshold of any value -100 nT and below [13] (see Table 4.2).

During storm identification, the SYM-H values are searched through year-by-year to find where the values drop below the minimum threshold for at least mild storms and where the values rise above the recovery threshold of -20 nT. The preliminary start and end points of the storms are the last recovery point before an extreme minimum value and the first recovery point after an extreme minimum value respectively. Once the preliminary start and end points for each valid storm are defined, the storms are sorted into their respective intensities based on the minimum SYM-H reached.

The Sudden Storm Commencement (SSC) point is searched for and appended to each storm by searching within a 24 hour window preceding the preliminary start points for the maximum SYM-H value above the recovery threshold. The first occurrence of the maximum value in that time window is taken as the SSC point, and the preliminary start points of each storm are updated with the start of the additional 24 hour window that now includes that point. As long as a valid SSC point is not found, the length of the standard preceding window is incrementally increased until the manually set maximum window size is reached. If a valid SSC point is still not found, a preceding window of the maximum size is simply appended to the initial start point.

Figure 4.4 shows the SYM-H index alongside B_z , V_{sw} and N_p for a selected severe storm from 1998.

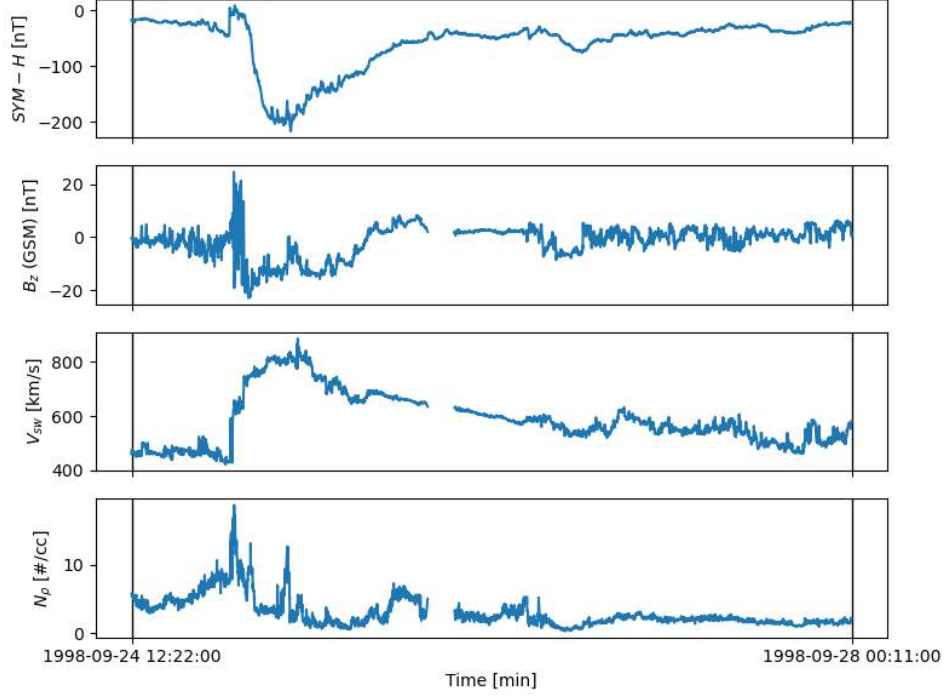


Figure 4.4: SYM-H alongside B_z , V_{sw} and N_p for storm 1998_severe.006. The SSC is the ‘spike’ in SYM-H at the beginning of the storm when the CME hits the magnetosphere. One can notice the spikes in B_z , V_{sw} , and N_p coinciding with the SSC.

Once events are selected, they are each labeled with a unique identifier, or storm ID. The storm IDs take the general form of ‘YYYY_intensity_NNN’, which enables quick and easy reference to individual storms. This storm ID will be used to refer to all storms throughout this study.

4.3 Exploratory data analysis

There is a total of 660 identified geomagnetic storm events of all intensities from 1996 to 2019 in the HAS data set, spanning 2 solar cycles. Table 4.3 provides a summary of the data set, listing the total storms, data points in each storms, and those remaining data points containing Not-a-Number (NaN) values.

An important step in the modelling process is to understand and explore the training data set

Table 4.3: Summary of the data points in the HAS data set. Listed is the total number of points, minimum, maximum, and average number of points for the entire data set and each storm intensity, as well as the total and average number of remaining data points containing NaNs for the entire data set and each storm intensity.

	All storms	Severe storms	Moderate storms	Mild storms
Total points	1 423 810	459 022	236 162	726 267
Total NaNs	62 607	23 227	12 501	24 508
Total storm events	660	122	87	451
Avg. points per storm	2 150	3 762	2 715	1 610
Avg. NaNs per storm	91	190	144	54
Min. points per storm	73	632	222	73
Max. points per storm	9 932	9 932	7 147	6 018

thoroughly before embarking on any further modelling steps. In this section, Exploratory Data Analysis is conducted to characterise the data set and gain an understanding of its contents, behaviour, and underlying correlations. This information is used to improve judgement of the explanations generated in Chapter 6. Specifically, we find the empirical distributions of each parameter and fit them to the closest theoretical distribution. We then compare the distributions of each parameter between stormy and quiet periods, as well as the distributions of e_h between the HER and ABK observatories as well as between quiet and stormy periods. Finally, we calculate pairwise linear correlations between the input parameters and e_h .

It is expected that there will be differences in both intensity and distribution of e_h values between HER and ABK as well as between stormy and quiet periods. In addition, the correlations between input parameters and e_h should similarly differ between latitude region and between quiet and stormy periods. Furthermore, parameters identified as important in this study by domain expert are expected to have strong correlations with e_h at both stations, which should align with the relative importance assigned to them by DeepSHAP in Chapter 6.

4.3.1 Empirical distributions

In this section, we analyze the empirical distributions of the input parameters listed in Table 4.1 and e_h at both high and low latitude stations.

Histograms are drawn to compare the effect of geomagnetic storm identification from the bulk data set. We show the distribution of each parameter for all available data as well as for identified

storms (the HAS set itself) from the years 1996-2019. To ease visualisation the outliers falling outside the 1st and 99th percentile are dropped, except for e_h at ABK (85th percentile), and HER (98th percentile). Note that these outliers are only dropped to prevent obscuring the overall distribution shape but were not excluded from further analysis. 60 bins are used to create the empirical distribution of each parameter, with 80 bins used for $e_{h_{ABK}}$. Figure 4.5 presents the histograms drawn for the resulting distributions.

Upon visual inspection of Figure 4.5, it is evident that the empirical distributions of V_{sw} and V_x are mirror images of each other. This is due to the x-coordinate in the GSE system pointing exactly along the Sun-Earth line, therefore approximately aligning with the dominant solar wind direction. This similarity is further confirmed in Figure 4.7 of Section 4.3.3, which shows that these two parameters are fully negatively correlated. As V_{sw} is the scalar value of the solar wind speed traveling toward Earth, and V_x the velocity vector of the same as measured in the same plane, it could be argued to keep one and exclude the other. However, both parameters will be included to explore whether their evident similarity is highlighted in the modeling and interpretation process.

The similarity between V_{sw} and V_x is clear; both seem to share a skewed-normal or gamma distribution. Figure 4.5 also reveals that V_z and V_y follow a skewed-normal or similar distribution, with a more pronounced skew being exhibited for isolated storms. N_ρ and P_d appear to have an exponential or log-normal distribution, with P_d possessing a heavier left-tail for isolated storms.

Figure 4.5 shows that the distributions of e_h at both stations are similarly exponentially distributed. e_h , and in particular, $e_{h_{ABK}}$ develops a longer/heavier tail as storms are isolated from the rest of the data. It should also be noted that $e_{h_{ABK}}$ is of much stronger intensity than $e_{h_{HER}}$.

In Figure 4.5, both B_y and B_z have similar dual-peaked distributions resembling Double-Weibull distributions. As storms are isolated, the tails of both parameters become heavier and the peaks flatter. The distribution of $|B|$ appears to be skewed, extreme or lognormal, with a heavy tail that grows heavier when storms are isolated. This is due to the bias introduced by selecting geomagnetic storms – strong negative B_z is critical to magnetic reconnection that is an important factor in the transfer of energy from the solar wind to the magnetosphere. For B_z , the distribution appears to be either normal or skewed normal for all available data, with a heavier left tail and left-skew that becomes more pronounced as storms are isolated.

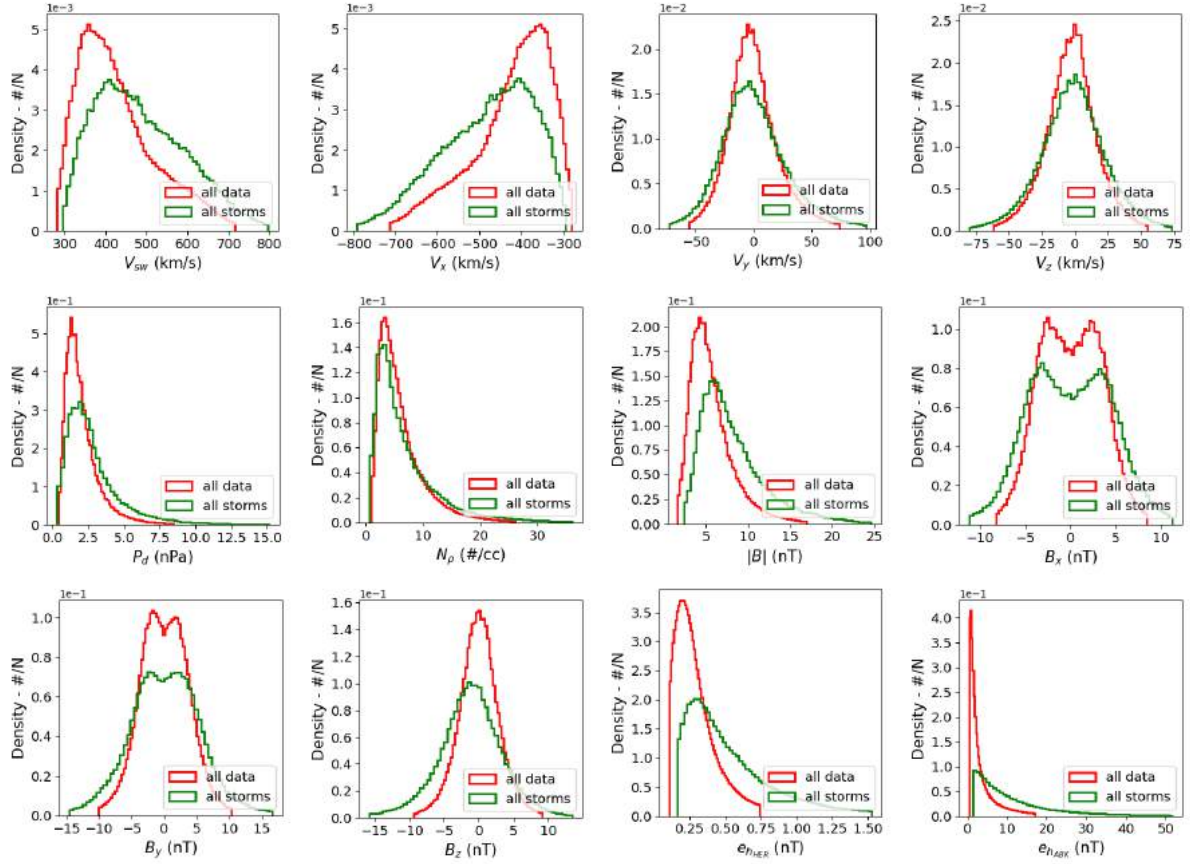


Figure 4.5: Empirical histograms of all input parameters and e_h at both latitudes, with all available data, and separately, all isolated storms, from 1996 to 2019. The relative frequency of samples is shown on the y-axis, and the value range is shown on the x-axis.

In Section 4.3.2, an automatic distribution fitting function is used to approximate the distributions of each parameter in the HAS data set. Further empirical distributions comparing the selection of each storm intensity in the HAS data set (severe, moderate, mild) are available in Appendix A.2.3.

4.3.2 Approximate distribution fits

In this section, we run an automatic distribution fitting function on each input feature and e_h , and compare the resulting selected Probability Density Functions (PDFs) to each parameter's empirical distribution.

We make use of scipy’s ‘continuous fit’ function⁵, that uses Maximum Likelihood Estimation (MLE) to estimate the relevant parameters of the selected theoretical distribution given the data by minimising its Negative Log Likelihood (NLL) function. The parameters returned by the ‘continuous fit’ function are used to create the PDF of this theoretical distribution. The Root Sum of Squares (RSS) between the empirical distribution and the PDF is calculated as a ‘goodness of fit’ measure. This process is repeated for each of the 10 theoretical distributions listed in Table 4.4. The fits for each of these distributions are ranked by their RSS score, and a final fit is chosen by visual inspection of the PDF vs. the empirical distribution prioritising better RSS score.

As with the empirical distributions, the outliers outside the 1st and 99th percentiles were dropped before fitting the distributions for each parameter (excepting $e_{h_{ABK}}$, 85th percentile, and $e_{h_{HER}}$, 98th percentile). 60 bins are used for each parameter, with 80 bins used for $e_{h_{ABK}}$.

The selected distribution, its RSS score, and its ranking out of 10 according to RSS score for each input feature and e_h is provided in Table 4.5. The PDFs of each of the selected theoretical distributions for all storm severities are shown in Figure 4.6.

Table 4.4: List of theoretical distributions to be fit to each parameter

Distribution name	Description
gennorm	Generalised Normal distribution
lognorm	Logarithmic Normal distribution
skewnorm	Skewed Normal distribution
dweibull	Double Weibull distribution
exponweib	Exponential Weibull distribution
expon	Exponential distribution
beta	Beta distribution
genextreme	Generalised extreme distribution
gamma	Gamma distribution
t	Student’s T distribution

⁵Scipy ‘continuous fit’ documentation: https://docs.scipy.org/doc/scipy/reference/generated/scipy.stats.rv_continuous.fit.html#scipy.stats.rv_continuous.fit

Table 4.5: List of selected theoretical distributions, their ‘rank’ out of 10 based on RSS score, and their RSS score.

Parameter	Selected distribution	Rank/10	RSS score
V_{sw}	skewnorm	2	$4.08 * 10^{-6}$
V_x	skewnorm	3	$4.07 * 10^{-6}$
V_y	skewnorm	1	$2.49 * 10^{-5}$
V_z	skewnorm	3	$4.07 * 10^{-6}$
P_d	genextreme	1	$3.46 * 10^{-3}$
N_ρ	exponweibull	3	$1.06 * 10^{-3}$
$ B $	lognorm	1	$4.92 * 10^{-4}$
B_x	dweibull	2	$4.98 * 10^{-3}$
B_y	dweibull	8	$1.31 * 10^{-3}$
B_z	t	2	$2.63 * 10^{-4}$
$e_{h_{HER}}$	exponweibull	1	$1.08 * 10^{-2}$
$e_{h_{ABK}}$	exponweibull	1	$4.04 * 10^{-5}$

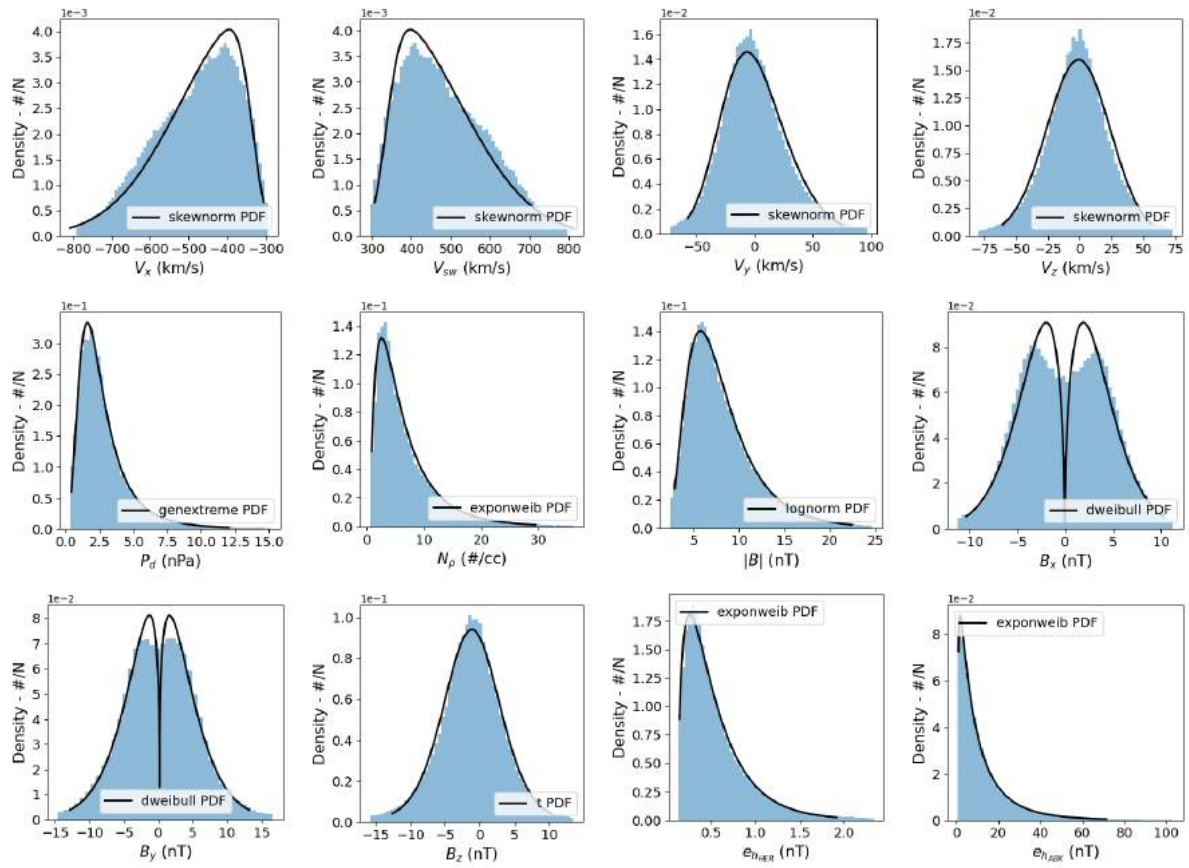


Figure 4.6: Theoretical PDF vs. empirical distribution of each input parameter and e_h at both latitudes.

4.3.3 Parameter correlations and comparisons

In this section, the Pearson's r correlation between each of the 10 parameters and the e_h index are calculated and compared between latitudes. Two-dimensional distribution plots are also created between each input parameter and e_h . All storm data between 1996 and 2019 is used, and each input parameter was standardised using z-norm standardisation. e_h (both HER and ABK) are standardised using z-norm standardisation, but without subtracting the mean. This was done to ensure that the standardised e_h values will always be greater than zero, since as a physical quantity, there can never be a negative e_h . This standardisation procedure is also followed before training any models, and is detailed further in Sections 5.3.1 and 5.3.2 in Chapter 5.

Figure 4.7 shows a heatmap of the resulting correlation values.

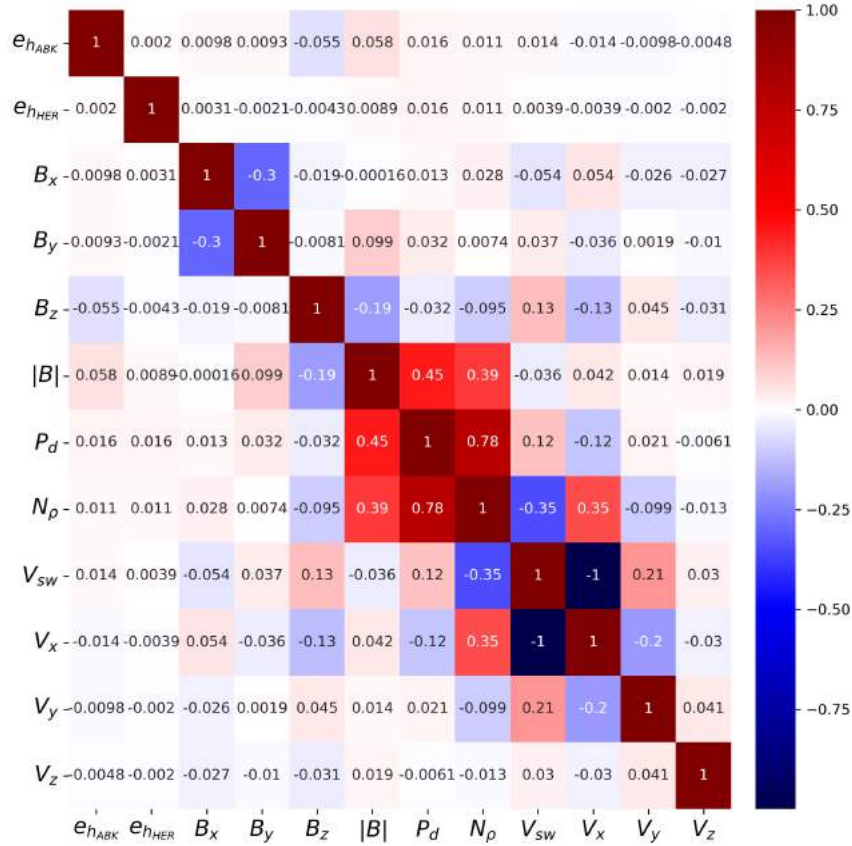


Figure 4.7: Correlation heatmap for the 10 input parameters and $e_{h_{HER}}$ and $e_{h_{ABK}}$.

As seen in Figure 4.7, the parameters that are relatively strongly correlated with $e_{h_{HER}}$ are less so than with $e_{h_{ABK}}$, with the exception of B_z , whose correlation is slightly stronger with $e_{h_{ABK}}$.

than with $e_{h_{HER}}$. Note that all B_z was included, i.e. positive and negative values. Restricting to only southward pointing IMF would have resulted in increased correlation with $e_{h_{ABK}}$ and $e_{h_{HER}}$. In particular, $|B|$ and pressure-related components ($|P_d|$, N_ρ , V_{sw} , and V_x) are far more strongly correlated with $e_{h_{HER}}$. This discrepancy could possibly be explained twofold:

1. $e_{h_{ABK}}$ is a lot more exposed than $e_{h_{HER}}$ magnetically, and would also have much more complex interactions with the solar wind that is not completely captured by a linear correlation, and
2. that $e_{h_{HER}}$, being a bit more shielded by the magnetosphere, would depend more on the reactions of the ring current and pressure-related parameters than $e_{h_{ABK}}$ would be.

Figures 4.8 and 4.9 show the 2-dimensional histograms of each of the input features vs. e_h . A very clear difference in the relationships between these parameters and $e_{h_{ABK}}$ and $e_{h_{HER}}$ can be observed.

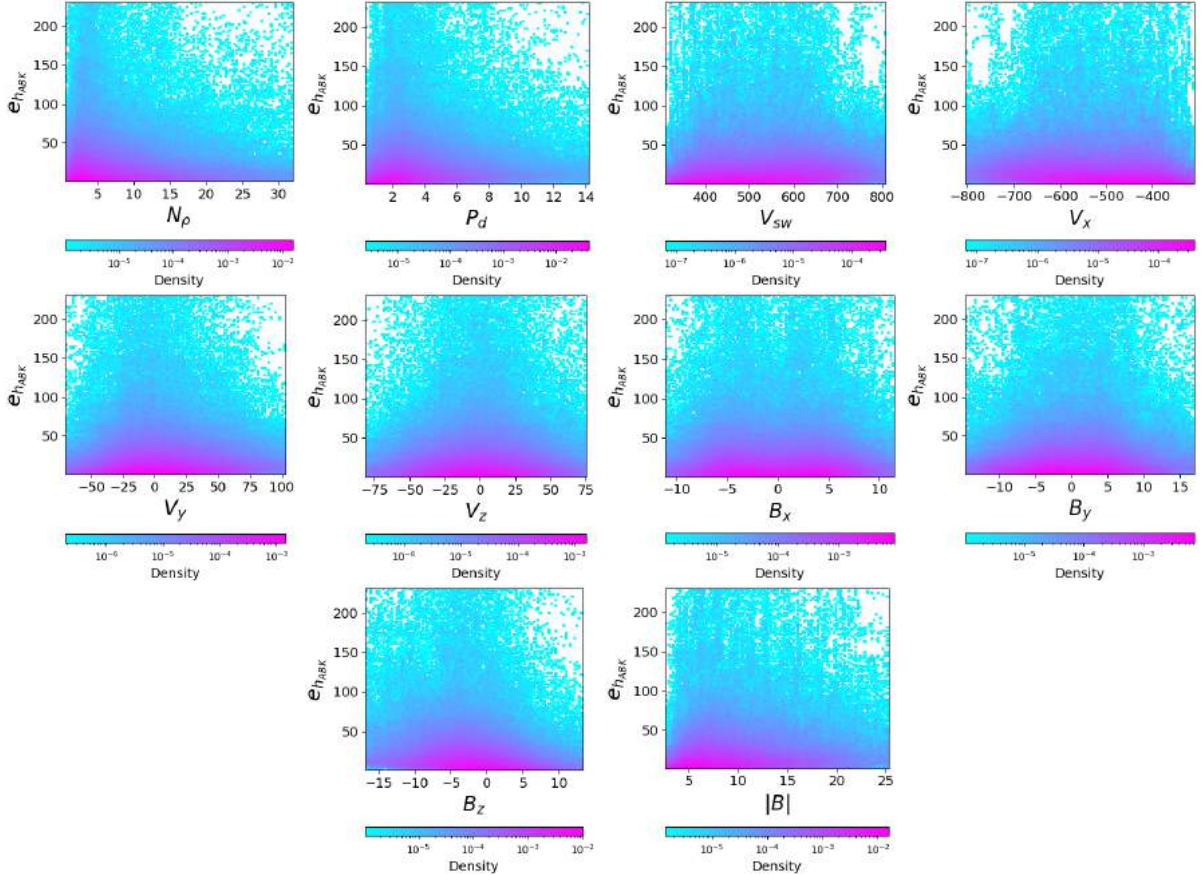


Figure 4.8: 2-dimensional histograms of each input feature vs. $e_{h_{ABK}}$.

For $e_{h_{ABK}}$, there is a very even concentration in the areas covered by the distribution. All magnetic parameters also have a very similar shape to one another, with B_z and $|B|$ skewing more to the left (negatively) as $e_{h_{ABK}}$ increases. On the other hand, for $e_{h_{HER}}$, there is a very clear difference in concentration in the covered areas. $|B|$ is highly negatively concentrated for lower values of $e_{h_{HER}}$, and is middle-negatively skewed still as the value of $e_{h_{HER}}$ increases, however, there is a higher positive concentration present than for $e_{h_{ABK}}$. All three magnetic components are heavily middle-concentrated around 0. B_z is less negatively skewed for increasing $e_{h_{HER}}$ than for $e_{h_{ABK}}$.

While the general shapes of the covered areas are similar between hemispheres, $e_{h_{HER}}$ has stronger concentrations. As the value of e_h increases in either hemisphere, similar distribution skews of the particle-associated parameters are observed.

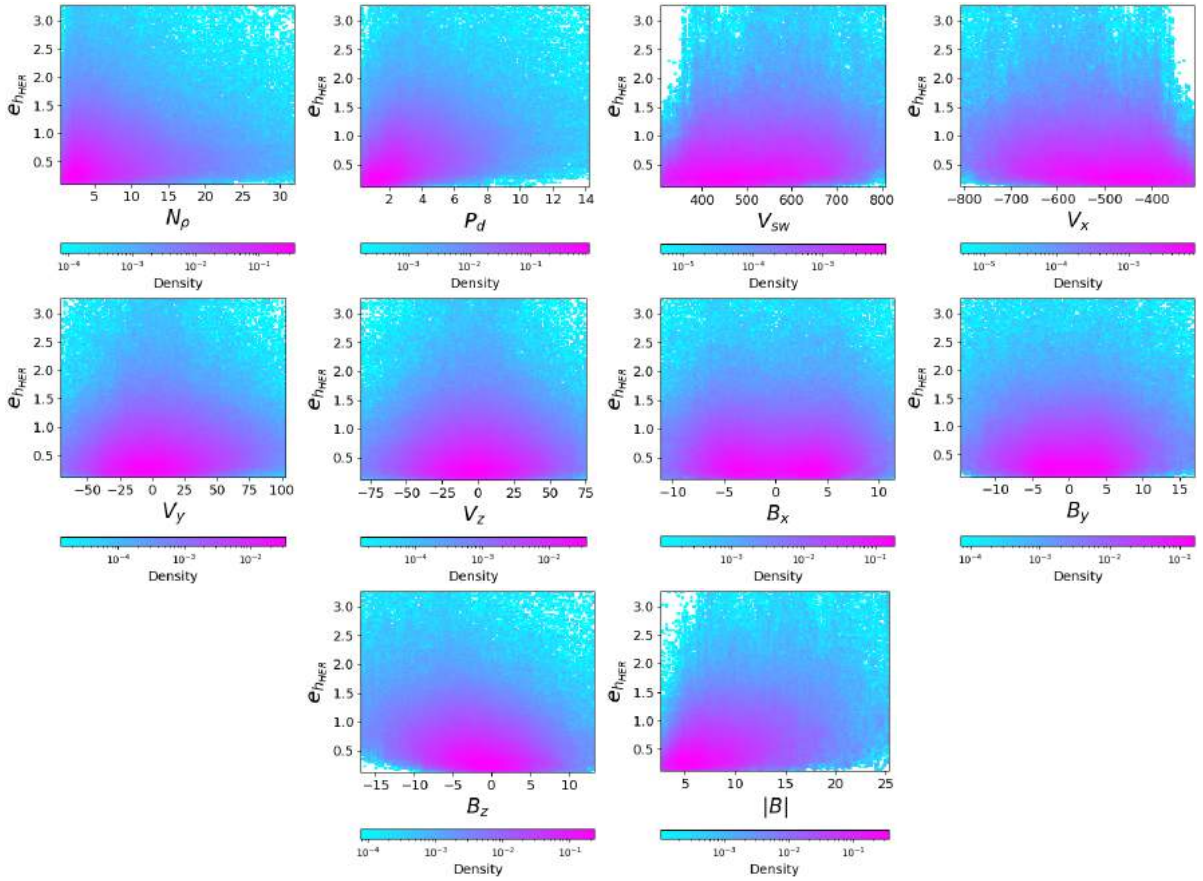


Figure 4.9: 2-dimensional histograms of each input feature vs. $e_{h_{HER}}$.

For $e_{h_{ABK}}$, N_ρ and P_d 's distributions appear to make a sloped curve as $e_{h_{ABK}}$ decreases in value and these two values simultaneously decrease in value. A small slope of this shape is also evident

as $e_{h_{HER}}$ decreases in value, however, there is a much stronger concentration in the corner of each distribution. Both of these parameters show a trend of being low in value when the value of e_h is high.

In V_{sw} and V_x both increase in their relative value as e_h increase in value, with V_x being more skewed to the middle for $e_{h_{ABK}}$, and heavier concentrations to the ‘lower’ end in each for $e_{h_{HER}}$. V_z has very similar distribution for both hemispheres, being quite centred around the middle. V_y , on the other hand, has a more pronounced left-skew as $e_{h_{ABK}}$ increases in value than for $e_{h_{HER}}$.

Additional two-dimensional histograms (for all input feature pairs) are available in Appendix A.2.4.

Chapter 5

Time-shift multi-layer perceptron models

Here we define, select, and train TSMLP models for the three main prediction tasks, evaluate these models, and discuss their performance.

5.1 Introduction

This chapter will describe the selection and training of suitable MLP models using time-shift delayed data as input (TSMLP) for a set of e_h prediction tasks. We use the MLP architecture due to its simplicity as a fundamental ANN architecture [6] and because a similar approach was used by Beukes et al. [4] for another solar wind related prediction task. Our goal is to model both the implicit and explicit differences in the response of e_h to the physical drivers between latitudes.

The general approach taken in this chapter is described in Section 5.2. The data set preparation and model selection process are presented in Sections 5.3 and 5.4, respectively, and the predictive performance of the models is discussed in Section 5.5. The inclusion of auto-regressive input features in further modeling is discussed in Section 5.6.

5.2 Approach

This study aimed to model, and thereby understand, the difference in e_h induced by the same drivers between two latitudes. To achieve this, three approaches are taken: creating a model that simultaneously predicts e_h at both latitudes, creating models that predict e_h at each latitude individually, and another model that predicts the absolute difference ($|\Delta e_h|$) between them. Additionally, since this is a time-series problem, multiple time-shifts are considered to account for the potential multi-temporal dependency of e_h on the input drivers.

Therefore we select and train a TSMLP model with four different time-shift sizes: 30, 60, 90, and 120 minutes, to predict:

- e_h for either HER or ABK alone,
- e_h for both HER and ABK simultaneously,
- and finally, $|\Delta e_h|$.

Recall that HER refers to e_h as derived from data measured at Hermanus, South Africa ($e_{h_{HER}}$), ABK refers to e_h as derived from data measured at Abisko, Sweden ($e_{h_{ABK}}$), and that $|\Delta e_h|$ is the difference between these two quantities once they have been standardised.

We anticipate that a model with sufficient depth, well-selected hyperparameters, and appropriate time-shift sizes will enable modeling the explicit difference between latitudes by accurately predicting $|\Delta e_h|$ or e_h at both latitudes simultaneously, with the simultaneous e_h predictions being of similar quality to those predictions of e_h at either latitude alone.

The TSMLP architecture described in this chapter is a simple MLP architecture with the addition of a time-shift delay of m minutes. Specifically, to create this time-shift delay, input features at time $t - m$ are appended to the vector of input features at time t . Additionally, previous output targets at time $t - m$ may optionally be included as auto-regressive input features as is done in Section 5.6.

The software tools utilized include a self-developed data pipeline written in Python which uses Pandas for much of the data manipulation, as well as extensive standalone Python scripts for handling model predictions, visualisation of results, and related tasks. mustnet3, an in-research-

group modeling toolkit developed in Python and based on Pytorch, is used to create, train, and validate these models.

5.3 Data set

All storms from the period of 1996 to 2019, of all intensities, were used for training, validation, and evaluation. The input features selected are those described in Section 4.2.1 and listed in Table 4.1 in Chapter 4, providing an initial set of 10 input features. Training and validation data will consist of a random selection of 70% of all storms, with a fifth of the storms in this portion randomly selected as validation data. The remaining 30% of all storms will be used as evaluation data. This method of random storm selection and data set splitting is kept consistent for the rest of this study.

After the data has been split into training, validation, and evaluation sets, the input features and the output targets are standardised (Sections 5.3.1 and 5.3.2), the output targets appropriately restricted to positive values post-normalisation (Section 5.3.2), and time-shifts are implemented (Section 5.3.3). Once the time-shifts are implemented, all samples containing features with NaN values remaining after the interpolation process (see Section 4.2.2) are dropped.

5.3.1 Feature normalisation

The ten input features are each standardised with z-norm standardisation to have a mean of 0 and a variance of 1. A z-norm standardisation scaler is created for each input feature separately, and fit to the training data. These scalers are then applied to their individual input features in the training, validation, and evaluation sets. The distributions of each input feature post z-norm standardisation are available in Figure 5.1.

5.3.2 Output target normalisation

Similarly to the input features, the e_h output targets are standardised with z-norm standardisation, with the standard scaler/s fit to the output targets of the training set. Unlike the process for standardising the input features, the output targets are standardised without centering the

mean at 0. Therefore, while the resulting output targets will have unit variance, they will be restricted to values greater than 0 and have a nonzero mean. This is desired, since e_h is a strictly positive quantity, and this will help to restrict the output space of the TSMLP models appropriately. Additionally, a ReLU activation will be applied to the output node/s of each model to further restrict the output space.

$|\Delta e_h|$ is calculated as the absolute difference between e_h at each latitude post-standardisation.

Figure 5.2 shows e_h at each latitude and $|\Delta e_h|$ after z-norm standardisation without centering the mean for training and validation data.

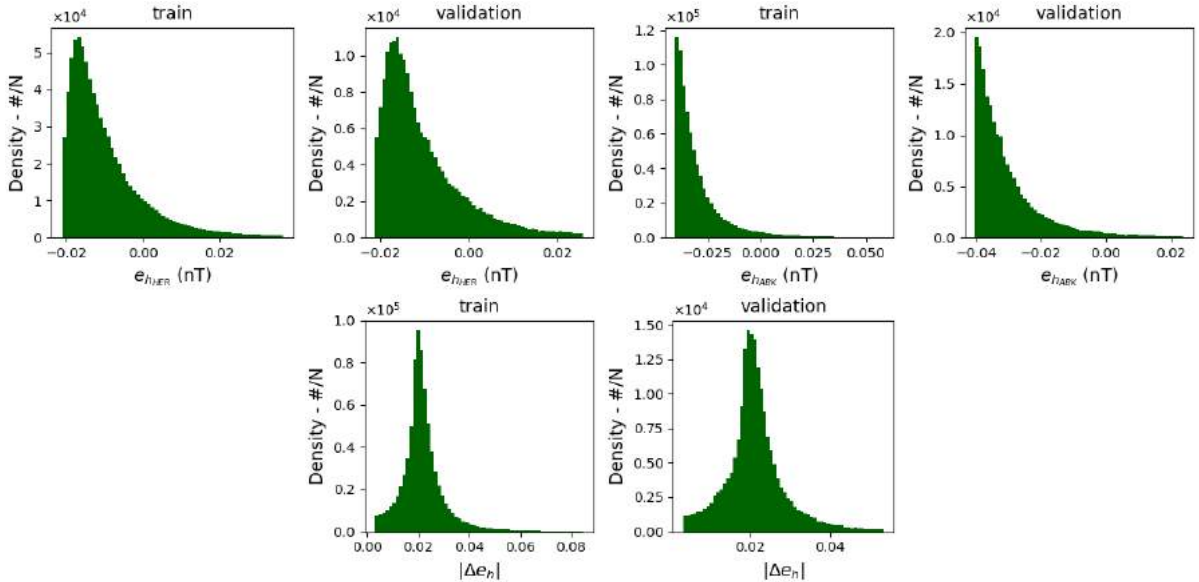


Figure 5.2: Distribution of e_h at HER and ABK after z-norm standardisation without centering at the mean, as well as $|\Delta e_h|$ calculated with post-standardisation e_h values, for training and validation sets.

5.3.3 Time-shift application

The time shifts are implemented by appending a previous instance of each input feature at time $t - m$, where m is the size of the time shift delay in minutes, to the input feature vector that is otherwise at time t . This essentially doubles the number of input features from 10 to 20 input features. Additionally, when applicable, a previous instance of the output target/s at $t - m$ is also appended to the input vector at time t as an auto-regressive input feature. This increases the previously doubled number of input features from 20 to 21 or 22, depending on whether there are one or two output targets.

5.4 Model selection

A series of hyperparameter searches are conducted to select an appropriate TSMLP configuration for each task. Here we follow the procedure described in Section 3.4 using SMAC3.

Here we describe the configuration space setup for the model selection process. This configuration space is utilised in each task’s search scenario. We search over model layout, learning rate, weight decay, and batch size. Table 5.1 lists the hyperparameters and their value ranges for this configuration space.

Table 5.1: Hyperparameter search configuration space for TSMLP model selection.

Hyperparameter	Range	Selection distribution
Layout (depth x width)	2x100, 2x200, 3x100, 3x200	Categorical
Batch size	128, 1024, 2048	Categorical
Learning rate	0.01 to 0.00001	Random uniform
Weight decay	0 to 0.001	Random uniform

In all cases, by default, leaky ReLU is used as hidden layer activation function, bias nodes are applied to all layers, Adam with betas set to $\{0.99, 0.999\}$ is used as the optimiser, early stopping is enabled without patience, MSE loss is used as the loss function, and ReLU is applied to the output node. Some additional information on the choice of early stopping settings with regards to SMAC3 is available in Appendix A.3.1. In each search scenario, the objective metric to be minimised is MSE loss on validation data, the minimum epoch budget is set to 10 epochs, three intensification rounds are specified, and the total number of model training runs is limited to 60.

Hyperparameter searches using the configuration space listed in Table 5.1 are conducted first for $e_{h_{HER}}$ and $e_{h_{ABK}}$ alone over each time shift size, and then only for the remaining tasks over the best performing time-shift size. At present, previous auto-regressive instances of output targets are *not* included in the input features. This is done only in Section 5.6.

The resulting models with the best final validation loss for each task are selected, their hyperparameter values are shown in Table 5.2, and their prediction performance is evaluated in Section 5.5 using storms from the validation and evaluation sets. Different units are reported in the communication of model performance. ‘Validation’ or ‘training’ loss refers to MSE as calculated using the raw model predictions and the standardised output targets. Further metrics, such as

RMSE, which are used to discuss model results are calculated using raw model predictions and output targets both re-scaled to the units of e_h .

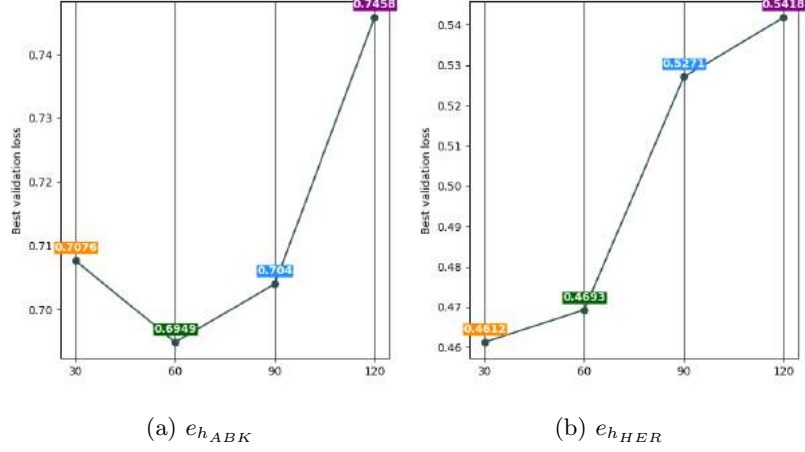


Figure 5.3: Best validation loss values for each time-shift size. The time-shift size in minutes is indicated on the x-axis, and the best validation MSE loss for the task at that time-shift is annotated at each point.

As seen in Figure 5.3, the time-shift size of 30 minutes performed the best in the case of $e_{h_{HER}}$. The time windows of 30, 60, and 90 minutes performed closely together in the case of $e_{h_{ABK}}$, with 60 minutes being the best performer. However, since the performance of 30, 60, and 90 minutes for $e_{h_{ABK}}$ is so close together, and that the following prediction tasks incorporate e_h at both latitudes, a time-shift of 30 minutes is selected for the hyperparameter searches conducted for the simultaneous e_h prediction and $|\Delta e_h|$ prediction tasks.

Figures 5.4 and 5.5 show the ‘scenario plots’, a scatter plot of hyperparameter values and MSE loss for each sequential run in a SMAC3 scenario, for HER and ABK at a time-shift of 30 minutes. The scenario plots for all tasks with a 30 minute time-shift are available in Appendix A.3.2, with the scenario plot for e_h ABK and e_h HER with all the time-shift sizes available in addition.

The hyperparameter values and final validation MSE loss of the winning configurations for each task are shown in Table 5.2. Single e_h prediction and $|\Delta e_h|$ tasks have shallower layouts (2 layers) than the simultaneous task (3 layers), while the $e_{h_{ABK}}$ and $e_{h_{HER}}$ tasks have narrower layouts (100 nodes per layer) than the simultaneous and $|\Delta e_h|$ tasks (300 nodes per layer).

Table 5.2: Training and validation loss at best epoch with hyperparameter values for best model configurations for each task.

Task	Valid. loss	Train. loss	Layout (depth x width)	Batch size	Learn. rate	Weight decay
e_h HER	0.4563	0.4005	2x100	128	0.0097	0.0010
e_h ABK	0.6871	0.6262	2x100	128	0.0070	0.0008
e_h HER and e_h ABK	0.5785	0.5506	3x300	128	0.0094	0.0010
Δe_h	0.4448	0.4346	2x300	128	0.0094	0.0010

It can also be seen in Table 5.2 that the tasks where $e_{h_{ABK}}$ is involved generally seem to do worse in terms of validation MSE loss than tasks where $e_{h_{HER}}$ alone is predicted. This is suspected to be due to higher difficulty in modeling $e_{h_{ABK}}$ because of the more ‘direct’ coupling between the solar wind drivers and the magnetosphere at ABK rather than the more magnetospherically ‘sheltered’ HER. This more direct coupling results in more complex relationships between the solar wind and e_h at ABK.

An evaluation of each model’s predictive performance according to various metrics is provided in Section 5.5.

5.5 Predictive performance and modelling results

Table 5.3 depicts the performance of each of these models on the train, validation, and evaluation sets in terms of root-mean-squared error and Pearson’s r coefficient. It can be observed that $e_{h_{HER}}$ performs comparatively well in terms of Pearson’s r coefficient for training, validation, and evaluation data. It similarly outperforms $e_{h_{ABK}}$ in single-latitude and both-hemisphere contexts. $|\Delta e_h|$ also does comparatively well across data splits.

Table 5.3: Performance evaluation with root-mean-squared error and Pearson’s r coefficient on the evaluation set of each task’s selected model. For all tasks except $|\Delta e_h|$, predictions were scaled up to original units. $|\Delta e_h|$ remains unitless: although nT is listed as the RMSE unit for all rows, take $|\Delta e_h|$ to have no units.

	RMSE [nT]			Pearson		
Task	Train. set	Valid. set	Eval. set	Train. set	Valid. set	Eval. set
e_h HER	0.3725	0.4103	0.3364	0.7712	0.7224	0.6856
e_h ABK	13.1305	12.5050	11.8387	0.5938	0.5520	0.5346
Both targets ($e_{h_{HER}}$)	0.3282	0.3473	0.3904	0.7548	0.7530	0.7265
Both targets ($e_{h_{ABK}}$)	12.0172	12.0777	11.6874	0.5806	0.5723	0.5642
$ \Delta e_h $ (no unit)	0.6671	0.6778	0.6943	0.5727	0.5659	0.5706

An individual storm is selected from the evaluation set for each task’s final model based on strongest and weakest Pearson’s r coefficients and corresponding best RMSE value. Since RMSE is based on the actual e_h quantities, it varies based on the e_h value ranges of a particular storm and is relative to each storm. Smaller is better within the context of each storm’s actual e_h or $|\Delta e_h|$ values, and Maximum Absolute Percentage Error (MAPE) is calculated to further assist with performance-based storm selection for discussion.

Figure 5.6 depicts the target and predicted values for the selected best-performing storms according to Pearson’s r and RMSE from the evaluation set, with Figure 5.7 depicting the same for the corresponding worst-performing storms. The storms that were most accurately modeled according to this criteria typically repeated themselves across tasks: with 2005_severe_002 dominating in all categories, with the exception of $|\Delta e_h|$, where the best-performing storm selected according to the criteria was 2017_mild_004.

When predicting a single e_h target, the best performers followed the general peak shapes well, and were somewhat congruent to the amplitude of the target. $e_{h_{ABK}}$ did better at approximating the target amplitude, with $e_{h_{HER}}$ ‘over-shooting’ at the maximum peaks. In the simultaneous task, similar behaviour is perceived. $e_{h_{HER}}$ over-shoots at the maximum peak, with $e_{h_{ABK}}$ over-shooting at the start of the peak. In general, e_h is similarly approximated in the simultaneous task to the individual e_h prediction task. Arguably, $|\Delta e_h|$ does the best visually here - with

both the placement and amplitude of the peaks well-approximated. However, it also over-shoots the amplitude at several points. It should be noted that $|\Delta e_h|$ is a unitless quantity based on the standardised values of $e_{h_{HER}}$ and $e_{h_{ABK}}$.

Additional output plots for randomly selected storms across all tasks are available in Appendix A.3.3.

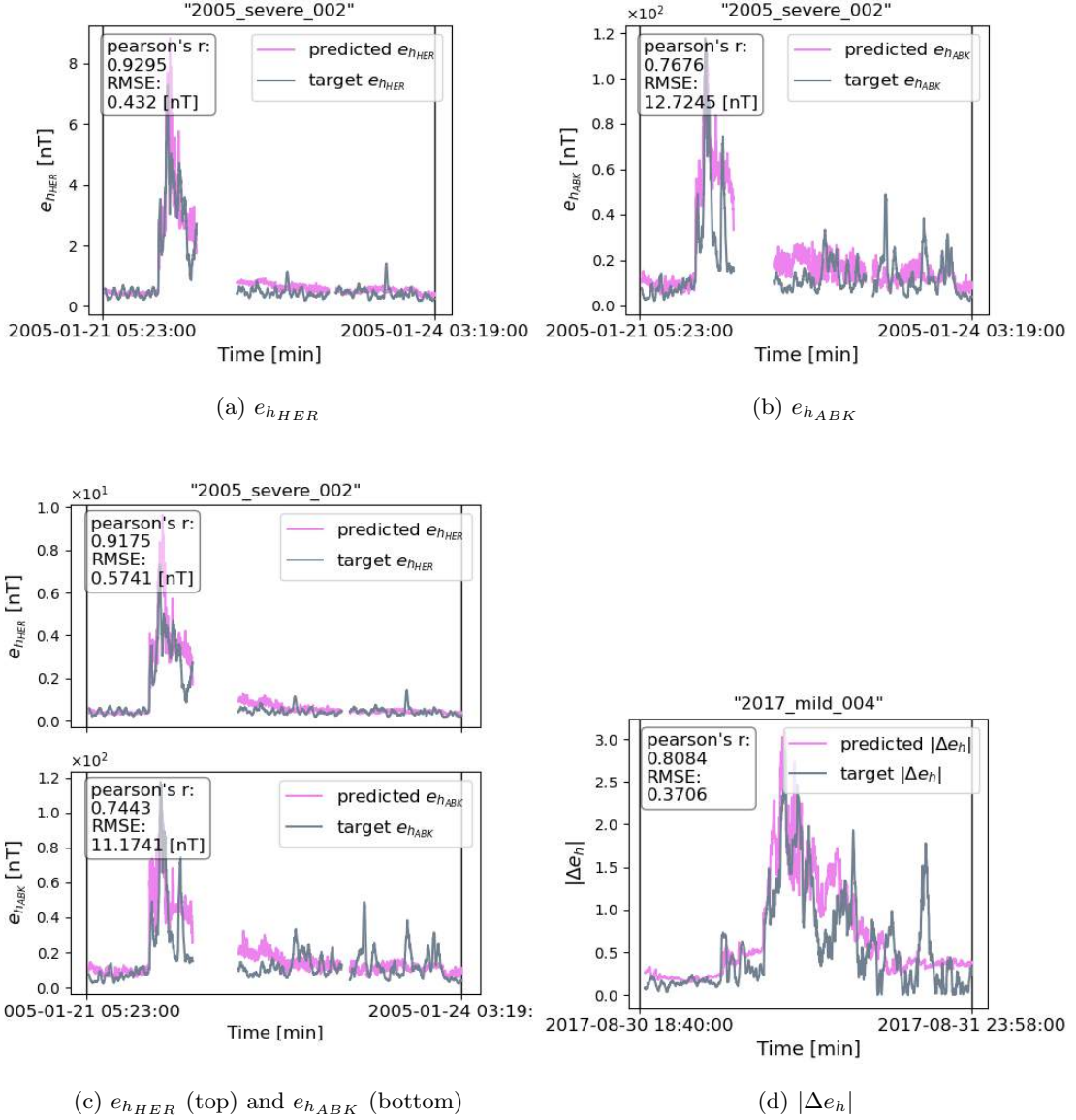


Figure 5.6: Target vs. predicted outputs of evaluation set storms with the strongest/best Pearson's r coefficient values and corresponding best relative RMSE values.

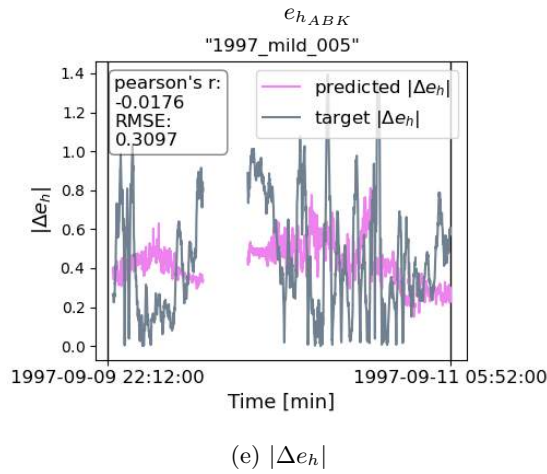
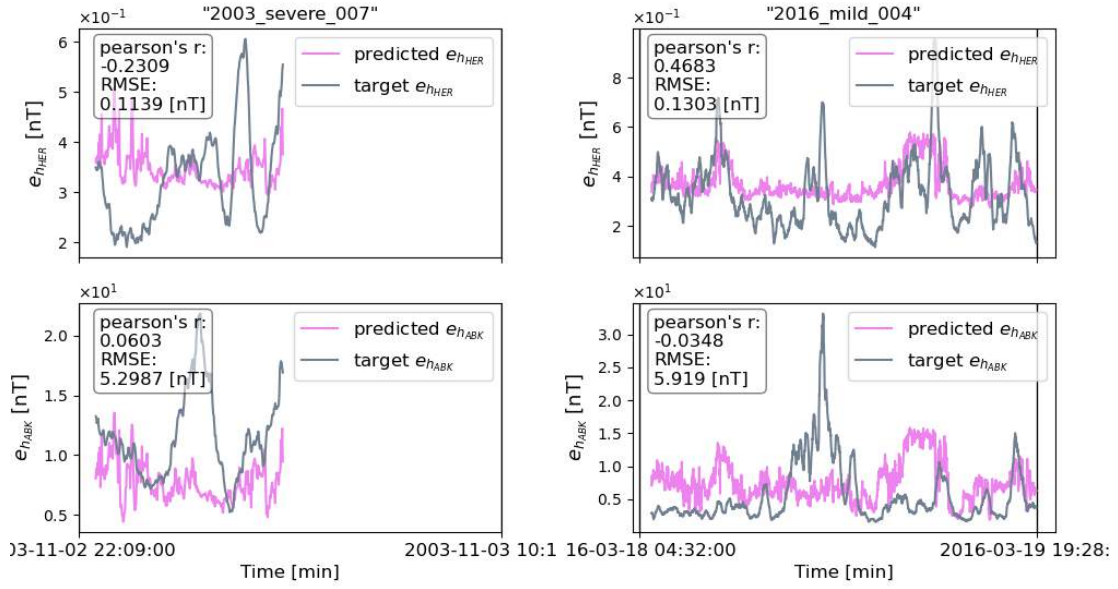
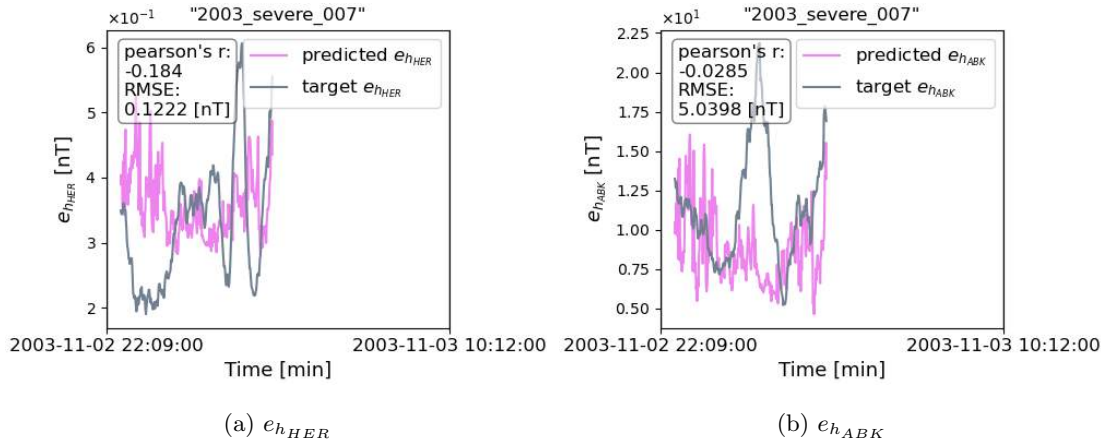


Figure 5.7: Target vs. predicted outputs of evaluation set storms with the weakest/worst Pearson's r coefficient values.

5.6 Results when including auto-regressive input features

The model selection procedure is run again on each task with the same data sets and approach, with an instance of e_h or $|\Delta e_h|$ at time $t - m$ appended to the input features for prediction at time t .

Once again, all time-shifts were searched over for the first two tasks of $e_{h_{HER}}$ alone and $e_{h_{ABK}}$, with the best-performing time-shift size used for the two remaining tasks. This best performing time-shift size with auto-regressive input features included was 30 minutes.

The hyperparameter values, along with the respective training and validation MSE loss values of the selected models for each task, are shown in Table 5.4.

Table 5.4: Training and validation loss at best epoch with hyperparameter values for best model configurations for each task.

Task	Valid. loss	Train. loss	Layout (depth x width)	Batch size	Learn. rate	Weight decay
e_h HER	0.3594	0.2883	3x300	1024	0.0040	0.0010
e_h ABK	0.3735	0.3694	3x300	128	0.0035	0.0010
e_h HER and e_h ABK	0.3667	0.3154	3x100	2048	0.0030	0.0006
$ \Delta e_h $	0.3440	0.3457	3x300	128	0.0073	0.0009

All tasks ended up with architectures of the same depth (3 layers), while the simultaneous task ended up with narrower architectures than other tasks. All tasks had moderately similar weight decay values. $e_{h_{ABK}}$, $e_{h_{HER}}$ and the simultaneous task all had similar learning rates, with $|\Delta e_h|$ boasting the highest learning rate.

Overall, when compared to the exclusion of auto-regressive input features, smaller learning rates, larger batch sizes, and more similar validation and train losses across tasks are present when auto-regressive input features are included.

Table 5.5 depicts the predictive performance of each model as reported by RMSE and Pearson’s r coefficient. All tasks performed better according to these metrics than without auto-regressive input features included but not with a very large margin. Similarly to before, $e_{h_{HER}}$ performs

relatively well based on Pearson’s r values across data set splits, with the performance difference between $e_{h_{HER}}$ and $e_{h_{ABK}}$ not as apparent in comparison to the same with auto-regressive input features excluded.

Table 5.5: Performance evaluation with root-mean-squared error and Pearson’s r coefficient on training, validation, and evaluation sets of each task’s selected model.

	RMSE [nT]			Pearson		
Task	Train. set	Valid. set	Eval. set	Train. set	Valid. set	Eval. set
$e_{h_{HER}}$	0.3152	0.3474	0.2789	0.8418	0.8069	0.7958
$e_{h_{ABK}}$	10.3600	9.9976	8.9562	0.7814	0.7574	0.7718
Both targets ($e_{h_{HER}}$)	0.2570	0.3072	0.3275	0.8579	0.8124	0.8179
Both targets ($e_{h_{ABK}}$)	8.2397	9.1114	9.2029	0.8316	0.7877	0.7652
$ \Delta e_h $ (no unit)	0.5899	0.5892	0.6103	0.6888	0.6972	0.6911

Similarly as before, storms are selected from the evaluation set based on strongest individual Pearson’s r coefficients while prioritising low RMSE and considering MAPE. The output targets vs. predicted values for each of the best performing storms selected from the evaluation set are shown per task in Figure 5.8.

In general, the peaks are much better followed, and the second prediction, be it $e_{h_{HER}}$ or $e_{h_{ABK}}$, fares much better in the simultaneous task than without the inclusion of auto-regressive input features. However, it should be noted that the output signal appears slightly delayed when compared to the target.

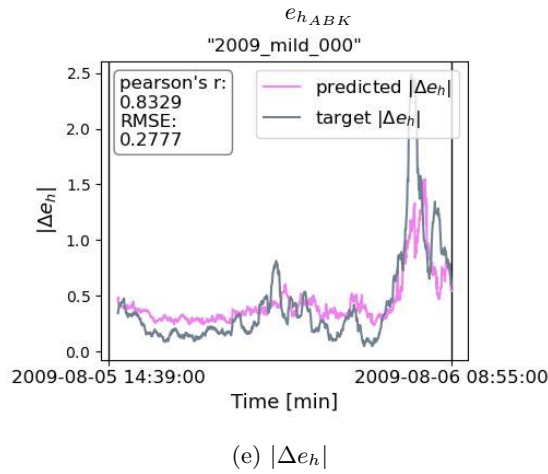
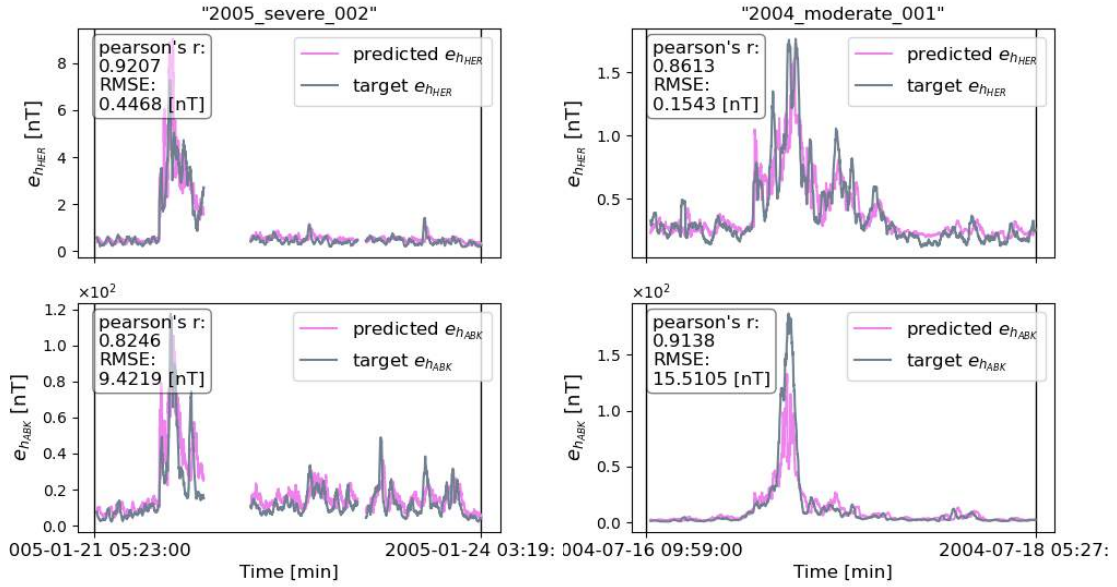
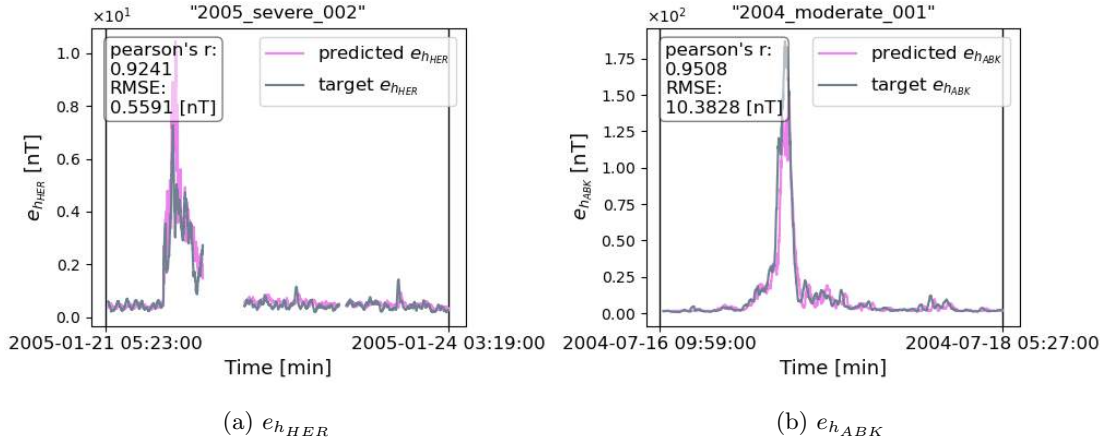


Figure 5.8: Target vs. predicted values of evaluation set storms with the strongest/best Pearson's r coefficient values and a lag of -30.

It is reasonable to suspect that the inclusion of an auto-regressive instance of either e_h or $|\Delta e_h|$ might cause the model to overfit on these easily-mappable input features. To investigate this, a cross-correlation test is run using the predicted values and the output targets of training, validation, and evaluation set storms to find at which time-lag the predicted values and targets are most commonly strongest-correlated. Table 5.6 shows the mode time-lags for the training, validation, and evaluation set storms of each task. Not all storms have this time-lag, and there are varying time-lags throughout each data split. However, when reporting on the mode, it is seen that across data splits the most common time-lag was -30/-29 minutes - indicating that there is most likely a much stronger emphasis placed by the models on the auto-regressive inputs than on other input features.

Table 5.6: Mode strongest correlated time-lags for training, validation, and evaluation set storms for each task where auto-regressive input features at -30 minutes are appended to the input.

Task	Train. set	Valid. set	Eval. set
$e_{h_{HER}}$	-29	-29	-29
$e_{h_{ABK}}$	-30	-30	-30
Both targets ($e_{h_{HER}}$)	-29	-29	-29
Both targets ($e_{h_{ABK}}$)	-29	-30	-30
$ \Delta e_h $	-30	-30	-30

Additional output plots of a random selection of storms for all tasks where auto-regressive input features are included are available in Appendix A.3.3.

5.7 Chapter summary

The goal of this chapter was to select and train a TSMLP model for four prediction tasks, with different time-shift sizes, and the inclusion/exclusion of auto-regressive input features. We expected that it would be possible to model the explicit difference between latitudes given a sufficiently large architecture and well-selected hyper-parameters, and that the performance of the explicit difference models would be comparable to the models predicting each latitude's e_h values individually.

When simply using time-shifted inputs without including auto-regressive input features, it was possible to both model e_h at each individual latitude, for both latitudes simultaneously, and $|\Delta e_h|$. $e_{h_{HER}}$ performed better than $e_{h_{ABK}}$ overall. As discussed in the results, this difference in approximation performance could be due to e_h at HER being generally less extreme than at ABK, as well as being relatively easier to model because of its more magnetospherically shielded location when compared to ABK.

When including auto-regressive input features, the performance of models across the board greatly improved in performance. However, this was tempered by the fact that the influence of the auto-regressive input features seemed to be over-emphasised by these models. This phenomenon was investigated by computing the mode time-lag of the predicted values when compared to the output targets, and this mode time-lag came to be -30 minutes, which was the time-shift size of the input and the lag of the auto-regressive input features.

While the performance increase afforded by including previous output targets is desirable, this could potentially have an adversarial impact to the relevance of different solar wind drivers, and is not ideal. Whether or not the inclusion of these previous output targets truly does adversely impact the relative contributions of the solar wind drivers or not will likely be seen in Chapter 6.

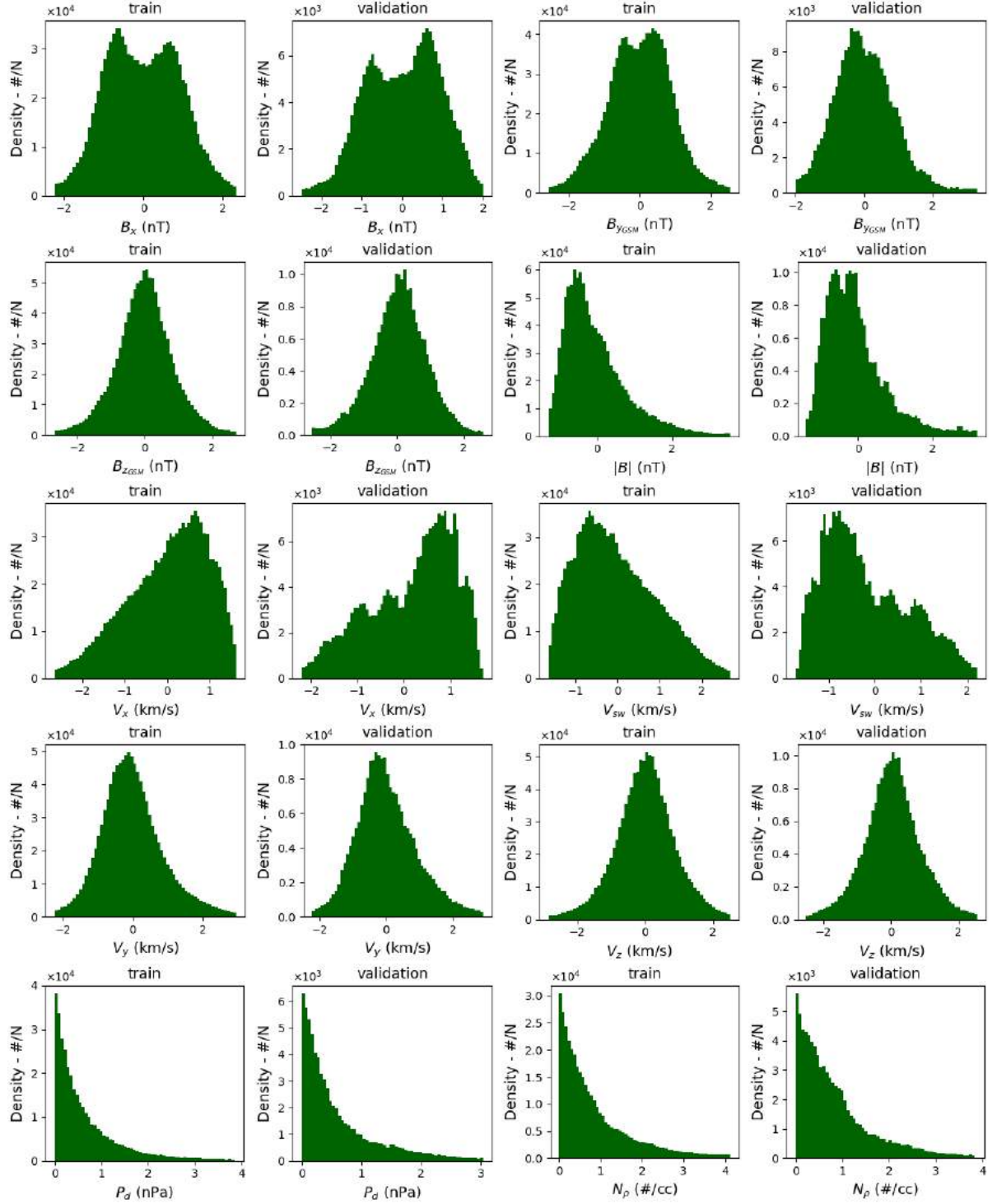


Figure 5.1: Distributions of input features after z-norm standardisation for training and validation sets.

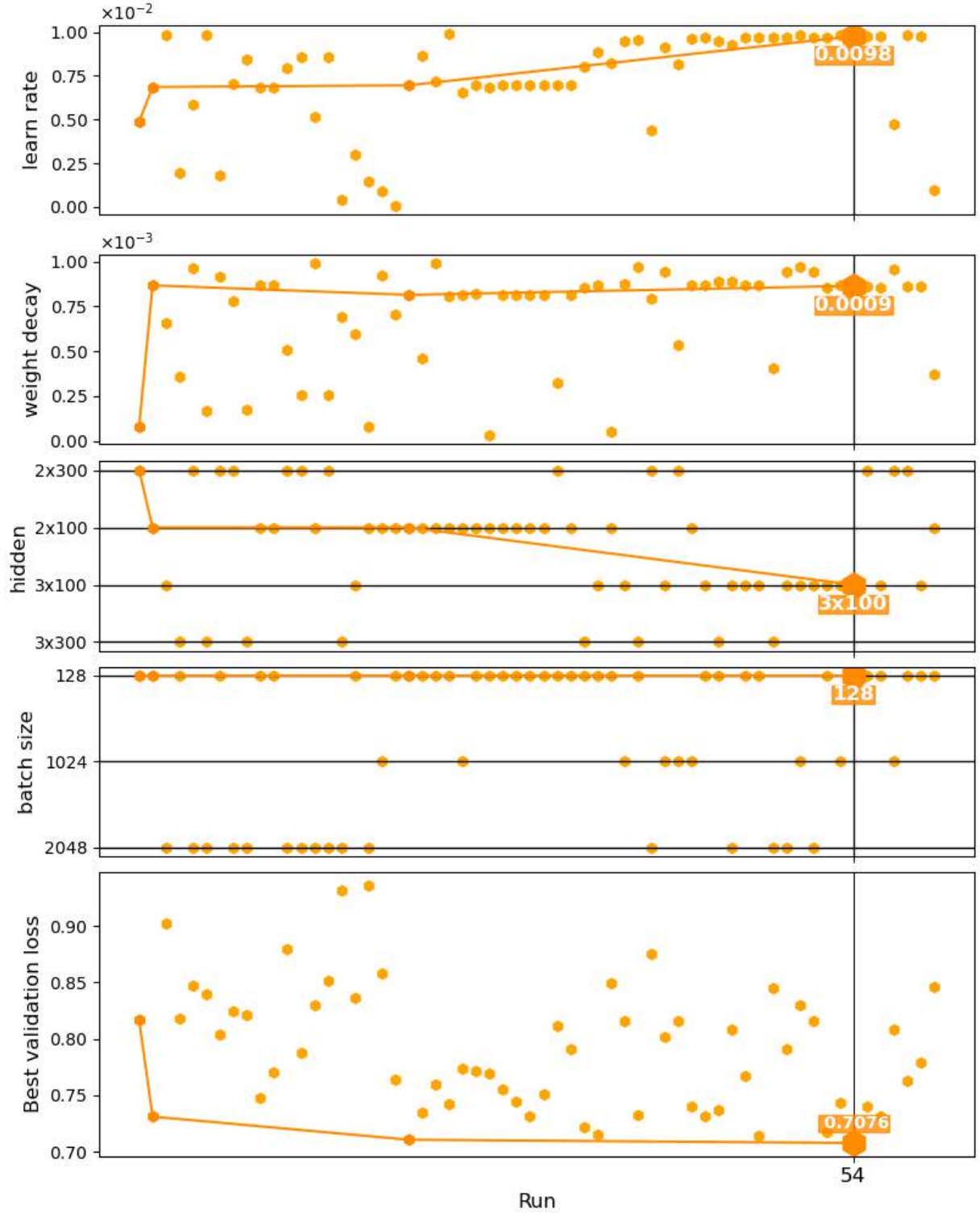


Figure 5.4: Scenario plot for $e_{h_{ABK}}$ with a time-shift of 30 minutes. The hyperparameter values and best validation loss value obtained at each run are shown as scattered points. Runs that strictly improve relative to previous runs are connected with a line that terminates at the final, best configuration.

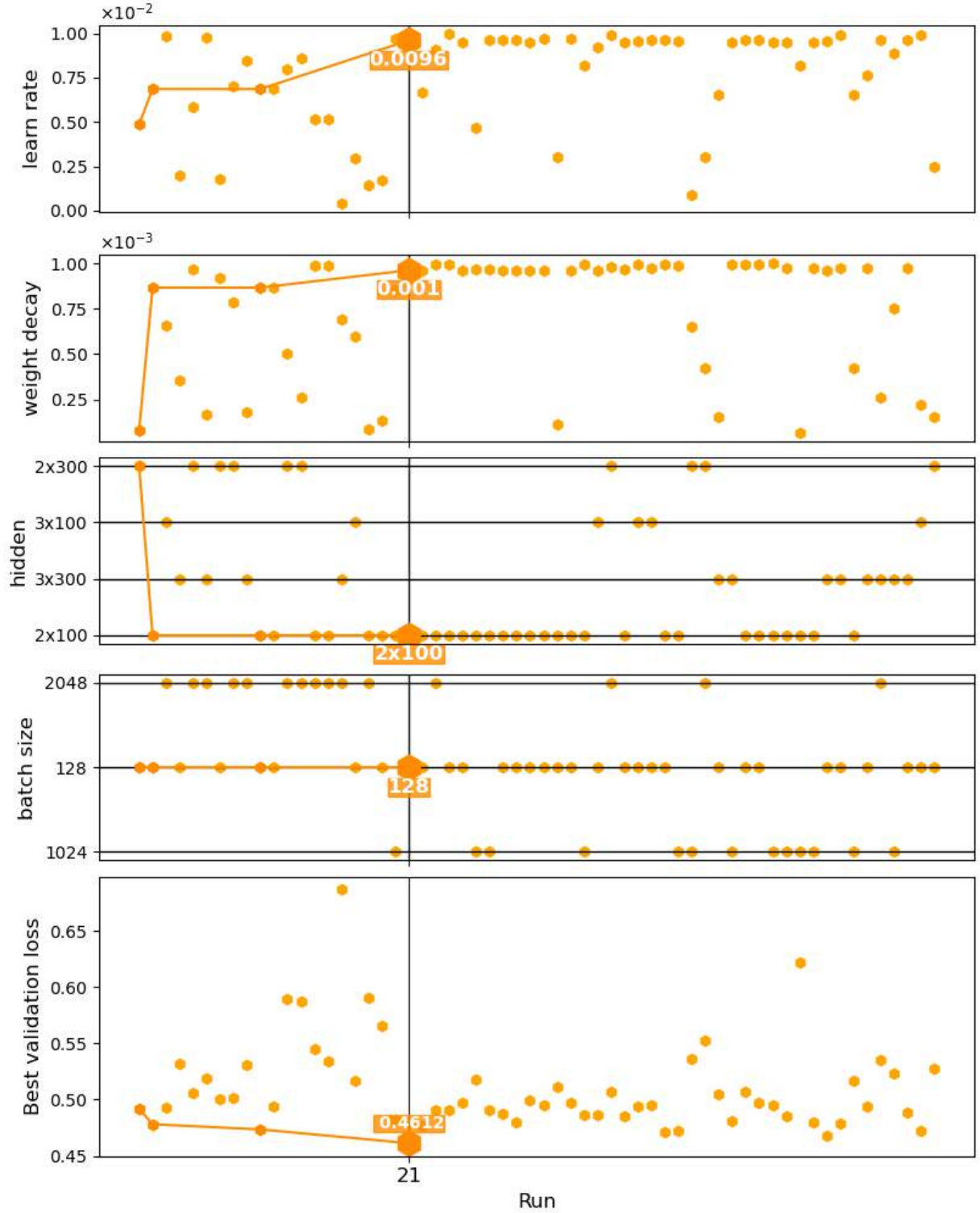


Figure 5.5: Scenario plot for $e_{h_{HER}}$ with a time-shift of 30 minutes. The hyperparameter values and best validation loss value obtained at each run are shown as scattered points. Runs that strictly improve relative to previous runs are connected with a line that terminates at the final, best configuration.

Chapter 6

DeepSHAP attributions and analysis

Using DeepSHAP, we interpret each of the selected TSMLPs for a selected set of CIR-driven and CME-driven storms, compare the attributions between models/tasks, and discuss the implications of the results.

6.1 Introduction

All ten input parameters described in Table 4.1 of Chapter 4 have some importance to the process of inducing geomagnetic disturbances quantified by the e_h parameter. Our expectations of how these input parameters will be regarded by our models and a brief description of the main drivers (Coronal Mass Ejections (CMEs) and Corotating Interaction Regions (CIRs)) of different types of geomagnetic storms are provided in Section 6.2.

We use DeepSHAP (briefly reviewed in Section 6.3.1, first described in Section 3.2) to generate attributions of each input parameter for each storm in the evaluation data set following the procedure outlined in Section 6.3.4. These attributions for selected CME-driven and CIR-driven storms across prediction tasks are displayed, analysed, and compared in Section 6.4. These results are discussed in Section 6.5.

6.2 Expectations

Following the description in Section 2.2 - we know that the Sun continually ejects ionised particles at varying intensities in the form of solar wind. However, as more activity appears toward solar maximum, more CMEs are released (compared to near solar minimum). A CME is a portion of the Sun's coronal mass that is separated from the Sun by interacting magnetic currents and is then hurtled in to space (including toward the Earth) at supersonic speed. This is a common driver of typically strong and/or damaging geomagnetic disturbances [21]. During other phases of the solar cycle, and in particular, the final phases of a declining solar cycle, streams of plasma begin to be ejected from 'coronal holes'. This constant stream of plasma associated with 'coronal holes' create CIRs, which last for several rotations of the Sun, and are another common driver of geomagnetic disturbances [21].

In the case of both CMEs and CIRs, when contact is made with the Earth's geomagnetic field, perturbations to the geomagnetic field are caused that induce effects that are observed at ground-based magnetic observatories. However, depending on the main driver, geomagnetic disturbances caused by either a CME or a CIR have different effects on the ground and on spacecraft as well as different progressions [55].

In the case of CME-driven events, the magnetosphere is perturbed by the speed, density and magnetic structure of the arriving ejected mass. This causes a compression of the magnetosphere, and the ensuing coupling with the Inter-Planetary Magnetic Field (IMF) by means of magnetic reconnection means that, along with $B_{z_{GSM}}$, N_ρ , P_d and V_{sw} should have the strongest impact on the geomagnetic disturbance during the onset and main phases of the storm. Unlike CIR-driven storms, CME-driven storms have a relatively short duration of impact, typically lasting one or two days at most [55]. Once the CME passes the Earth the IMF de-couples relatively quickly after the storm reaches its maximum onset, and the recovery phase of the storm essentially takes place in 'quiet', uncoupled, non-stormy conditions.

In terms of CIR-driven storms, there is typically less dramatic magnetospheric compression, and the plasma sheet is both less dense and with a weaker southward IMF component than CME-driven storms [55]. In this case, while V_x , N_ρ , and P_d would still have some effect, $B_{z_{GSM}}$ and/or $|B|$ overall would have a relatively stronger effect than these features would during the onset and main phases of the storm. CIR-driven storms typically last longer than CME-driven storms,

and they do not exhibit the same distinct recovery phase as with CME-driven storms [55]. This means, during the recovery phase, much of the plasma and magnetic coupling is still present, and will continue to induce perturbations until the Sun turns sufficiently on its axis so that the CIR stream is no longer aimed at the Earth. This means that solar wind drivers will continue to have direct effect throughout the recovery phase. Severe and some moderate intensity storms are typically caused by CMEs, while a majority of mild intensity and some moderate intensity storms are generally caused by CIRs [55].

Since CME-driven storms are more likely to cause longer and more intense aurora [55], and that Abisko (ABK) is located at high magnetic latitude, it is expected that N_ρ , P_d and magnetic drivers will have stronger effects at ABK than at Hermanus (HER) during CME-driven storms. It is also expected that HER will react relatively more strongly to flow pressure related terms (plasma speed and density) than to IMF components, since it is located at lower latitude.

When previous output target instances are included as auto-regressive input feature/s, it is expected that these inputs will have the strongest attribution across tasks, and in particular, at ABK. In general, for all tasks, it is expected that input features at delayed time-shifts will have stronger/more expected attributions according to our physical expectations, since there is inherently a delay in the reactions of the ground-based magnetic field to the solar-wind drivers.

6.3 Experimental protocol

6.3.1 DeepSHAP review

As described in Chapter 3, DeepSHAP is an implementation of SHAP that combines DeepLIFT’s use of backpropagating the contributions of difference-from-reference activation values to the difference-from-reference output onto the difference-from-reference input features.

DeepSHAP computes its SHAP attributions for a particular sample repeatedly over a set of ‘background’ or ‘reference’ input samples, and returns the mean of the attributions as the final set of SHAP attributions for that sample. The SHAP attributions add up to a ‘difference-from-reference’ value corresponding to the output value of the model for that sample. The output value of the model for that sample is approximately equal to that ‘difference-from-reference’

value added to some baseline value. When dealing with a model that has several simultaneous outputs, these attributions are calculated relative to each output value independently.

The choice of ‘background’ or ‘reference’ samples influence the insights that can be gathered from the attributions. This requires careful attention to be paid to the selection of the reference samples to prevent unwanted biases in the attributions [52]. Typically, to counter these potential biases, a representative set of training data set samples are selected as reference samples at random, and attributions are calculated using the validation or evaluation sets [52, 8]. This random selection from the training set can be stratified explicitly at the user’s discretion to yield interesting results: for example, when attributing the ‘cause of death’ on a general medical data set, the highest contributing factor may be age and/or gender. However, when a subset of older males are used as the reference samples, the highest ranked attribution becomes cardiovascular disease [8].

While the different strata of storm intensities in our data set may not have such a drastic, or any, discernable influence on the calculated attributions, it would be useful to determine if different selections of background data have a material effect on the attributions for this data set and what those effects are before we proceed with the attributions in earnest. This is done in Section 6.3.3.

6.3.2 Data sets

All TSMLP models were trained using the same storm data set (barring differences in autoregressive input feature inclusion/exclusion), with the same training-validation-evaluation split. For this reason, the same version of the data set used for each model type will be used in the attribution process of each model. Our attributions will make use of the data set as prepared in Section 5.3. The reference samples will be selected from the training set for each model, while attributions will be generated for storms selected from the evaluation set.

6.3.3 Reference data set selection

24 hours-worth of reference samples are selected with time-shifts included from the training set at a rate of 60 minutes from each storm. This translates to 60 samples (each sample represents

a minute’s worth of data) being randomly selected from 24 randomly selected storms for the reference set. This selection is done once for all attributions using the same randomisation seed for each model type and storm intensity selection type.

The reference data set selections (storm intensity selection types) are defined as follows:

- **All storms:** The reference set is randomly selected as described from the training set with all storm intensities included and no discrimination against or preference for any particular storm severity.
- **Opposite intensity storms:** The reference set is randomly selected as described, with the exception that the storm intensities are limited to the two storm intensities not currently being attributed. For example, if mild intensity storms are currently being attributed, the reference samples are selected only from moderate and severe intensity storms in the training set.
- **Same intensity storms:** The reference values are randomly selected as described with the storm intensity limited to the current storm intensity being attributed. For example, severe intensity reference values for severe intensity storms.

Attributions are generated for the individual $e_{h_{ABK}}$ prediction task with and without auto-regressive input features included, for storms of mild intensity, and for different reference set selections. The full attribution plots for each background set selection for selected mild intensity storms are available in Appendix A.4.1.

For each storm, the attributions corresponding to each time-step between reference sets are visually compared. For each reference set, the feature rankings by mean relative contribution to the difference-from-reference in the output are calculated over the whole evaluation set.

The attributions of $B_{z_{GSM}}$ (for the case where auto-regressive input features are excluded) corresponding to each time step in the output for the same storm over different reference set selections are shown in Figure 6.1, and the feature rankings by mean relative contribution to the output across background set selections are shown in Figure 6.2.

The same is done for the case where auto-regressive inputs are included in Figures 6.3 and 6.4.

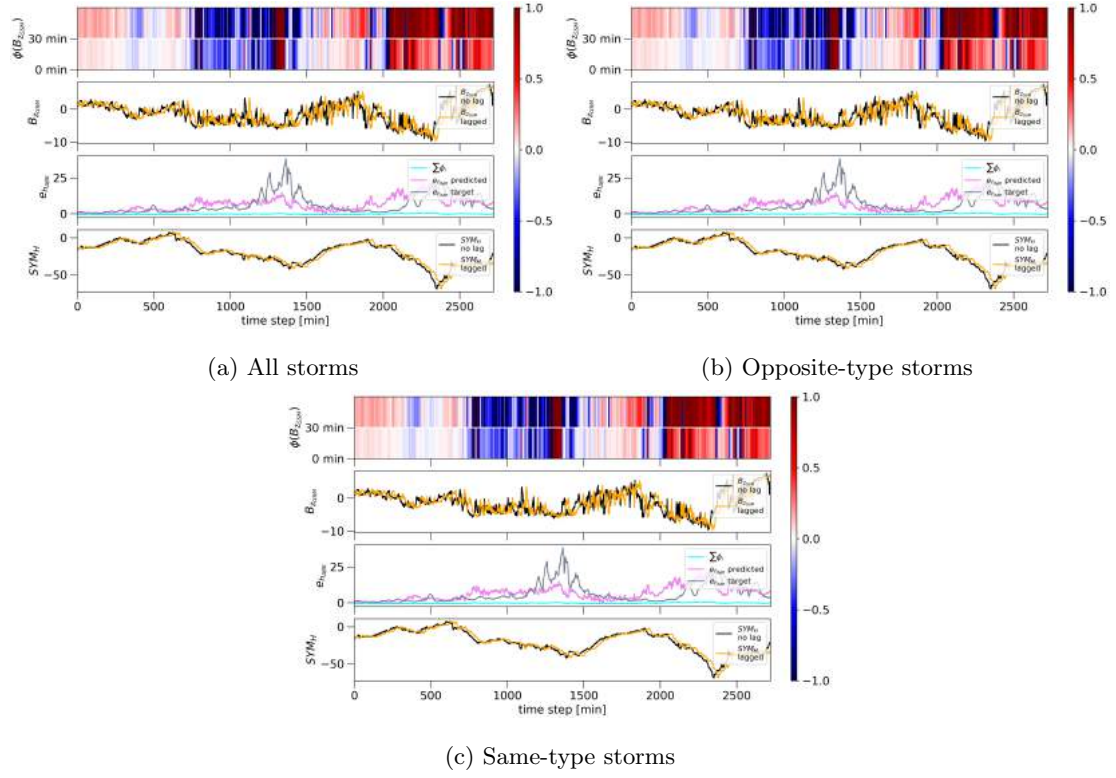


Figure 6.1: Time-series attribution plot of $B_{z_{GSM}}$ for 1998_mild.001. Shown is the relative contribution of $B_{z_{GSM}}$ at each time step to the difference-from-reference of e_{hABK}

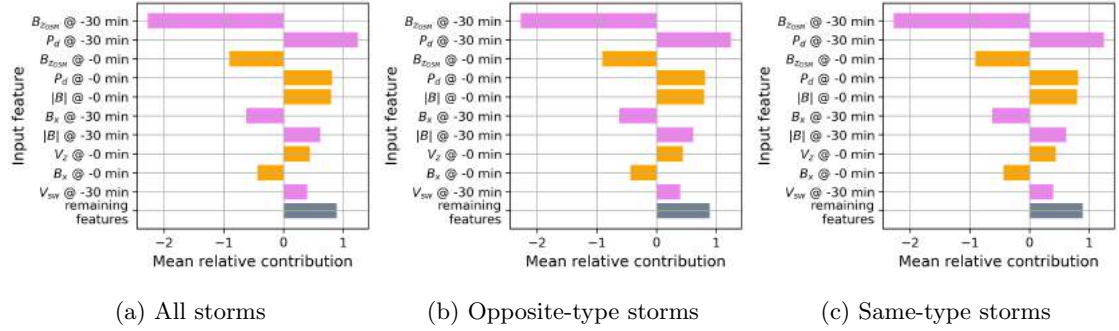


Figure 6.2: Feature ranking plot according to mean relative contribution for the set of all mild intensity storms at ABK.

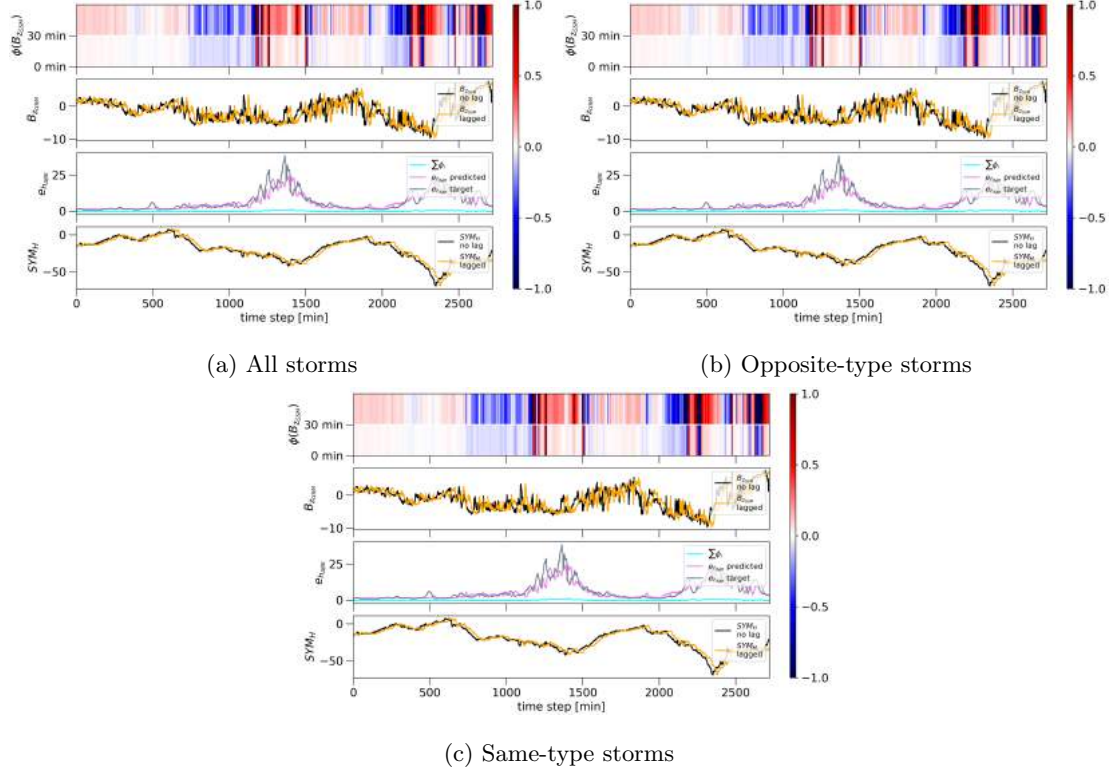


Figure 6.3: Time-series attribution plot of $B_{z_{GSM}}$ for 1998_mild.001. Shown is the relative contribution of $B_{z_{GSM}}$ at each time step to the difference-from-reference of e_{ABK}

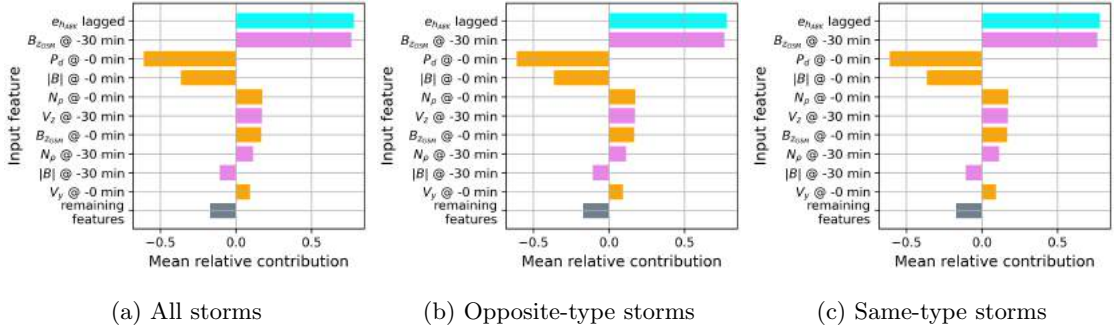


Figure 6.4: Feature ranking plot according to mean relative contribution for the set of all mild intensity storms at ABK with auto-regressive input features included.

As can be derived from Figures 6.1, 6.2, 6.3, and 6.4, the background selection makes very little if any material difference to the generated attributions. For the purposes and scope of this study, we will use a single reference set selection method for the sake of brevity. With no particular preference, ‘same type storms’, as described here, is chosen as the reference set selection method.

6.3.4 Attribution procedure

The reference set is selected according to the method described in Section 6.3.3, using ‘same type storms’ for each storm intensity being attributed (mild, moderate, and severe). Each storm intensity’s reference set is re-used for each storm that is attributed of that particular intensity.

The attributions are generated sample by sample and concatenated into a single array, which spans the entire time-series of each individual storm. To ensure that the attributions are comparable to one another for the entirety of the storm and are not only specific to their respective output samples, each sample’s attributions are normalised by the sum of attributions for that sample. This ensures that relative contributions of each feature over time is not obscured by the magnitude of the output sample that it is contributing to. This means that the magnitude of the attributions will indicate the intensity of each feature’s contribution to e_h (or $|\Delta e_h|$) comparably throughout the duration of an entire storm or even across an entire data set regardless of the magnitude of e_h or $|\Delta e_h|$. Attributions are generated in this way for all storms in the evaluation set.

6.3.5 Attribution visualisation

Attributions will be visualised in three main ways: with so-called time-series attribution plots, feature attribution plots, and feature ranking plots. We first describe these visualisations here before discussing the results.

Time-series attribution plots present the relative contribution of each feature at each time-shift over time for all output samples in the progression of a storm. These relative contributions are presented in a colour map that maps the attribution values from -1 to 1, corresponding to blue and red respectively. These colour maps show the relative intensity and direction of the feature contributions in the context of the whole storm over time - blue would mean that the feature at that time-step would contribute to lowering the output value, while red would mean that the feature at that time step would contribute to increasing the output value. The degree to which this contribution to the increase or decrease of an output value is indicated with the intensity of the respective colour. These colour maps are shown alongside a time-series graph of corresponding target and output values, sums of attributions for each output sample, and an

additional time-series graph of SYM-H at 0 minutes of lag and -30 minutes of lag to contextualise the attributions in the storm progression. The sums of attributions for each output sample are included as a proxy for the difference-from-reference, and should match the graph of the output values with a difference in amplitude. This visualisation provides a master summary of all the contributions of each input feature for a particular storm at a glance, but is, however, a very information-dense plot.

Feature attribution plots are very similar to time-series attribution plots in that they show the time-series attributions alongside the output, target, attribution sums and SYM-H in exactly the same way as time-series attribution plots, but for only one feature at a time. Feature attribution plots show the colour map of the input feature at both its time-shifts over time for the whole progression of the storm, with time-series graphs of that input feature at its time-shifts alongside this colour map in addition to the other aforementioned elements. This visualisation helps contextualise the contributions of each individual feature to the output over time with the feature’s behaviour itself.

Feature ranking plots are used to present a very rough-grained summary of the overall behaviour of each feature at each time-lag throughout a storm, or in a certain set of storms. Feature ranking plots are created by obtaining the mean relative contribution of each input feature at each time-lag to the output for a particular storm or set of storms and ranking these mean contributions in descending order by magnitude of mean relative contribution. These visualisations are high-level summaries of average relative feature importance. It is important to keep in mind, however, that these mean relative contributions for the feature ranking plots are calculated using attributions from all phases of a storm, including the recovery phase. For CME-driven storms in particular, the majority of the influence of the solar wind parameters on e_h occurs during the onset and main phases of the storm. During the recovery phase, when the solar wind has already passed and the IMF has been decoupled, the majority of the influence on e_h is actually due to internal magnetospheric processes that cannot be measured with the same instruments that are measuring the solar wind [56]. For this reason, while the feature ranking plots will show to a great extent which solar wind parameters are most important during a particular storm, or a set of storms, the degree to which, and whether the influence is mostly positive or negative, is not necessarily reflective of the entire physical reality, and time-series or feature attribution plots are required to truly contextualise the relative contributions of the input features concerned.

6.3.6 Storm selection

Particular storms are selected for discussion from the evaluation set based on their driver types as well as visual inspection of their performance at both latitudes. Driver types are determined by manually matching event dates to corresponding dates in CIR-event [57, 58, 55, 59, 21, 60] and CME-event [58, 61, 21, 55, 60] catalogs. Storms that are either CIR- or CME-driven (i.e., not ambiguous or having mixed drivers) and are well-predicted in both the individual $e_{h_{HER}}$ and individual $e_{h_{ABK}}$ tasks are selected for discussion.

The selection criteria for discussed storms is based on the observation of the storm’s output vs. target value graphs at both latitudes for models where auto-regressive input features were not included, as well as based on the performance criteria used in Chapter 5.5. Ideally, the output e_h values should at least ‘catch’ the peaks of the target e_h values during the main phase and reasonably match their amplitude, and should otherwise match the ‘idle’ or non-peak amplitude of the target e_h . This ensures that at least the initial onset and/or main phase of the storm is adequately captured - since most of the magnetospheric coupling occurs during this time.

In the interest of brevity, only a single storm per driver type is discussed in detail throughout this chapter as a representative example of storms of that driver type. A full list of manually selected evaluation set storms according to these criteria, along with their feature ranking and time-series attribution plots for each task are found in Appendix A.4.3 and A.4.2.

6.4 Analysis and results

Here we present and discuss the relative contributions of each feature as calculated from the DeepSHAP attributions for a set of CME-driven storms and a set of CIR-driven storms (Section 6.4.1). The storms in these sets are selected based on the described selection criteria. A single example CME-driven storm (Section 6.4.2) a single example CIR-driven storm (Section 6.4.3) are further discussed in detail.

The majority of storms in the evaluation data set are CME-driven storms, and, typically, the CME-driven storms in the evaluation set tend to perform better than the CIR-driven storms according to the performance criteria set out in Section 5.5. There are fewer CIR-storms, and

even fewer well-performing CIR-storms in the evaluation set, with only one storm performing to satisfaction according to our selection criteria on both $e_{h_{HER}}$ and $e_{h_{ABK}}$. This particular storm is used as the CIR-driven storm example. As was mentioned in Section 6.3.6, more examples of CIR-storms that perform well at either $e_{h_{HER}}$ or $e_{h_{ABK}}$ with their attribution plots available in Appendix A.4.2, as well as the set of selected CME storms and their individual attribution plots in Appendix A.4.3.

6.4.1 Aggregate results

8 CME-driven storms and 3 CIR-driven storms that conform to the selection criteria are selected, and their attributions are compared. For each task, the relative contribution of each input feature is calculated as described in Section 6.3.4. The mean relative contributions of each feature are ranked for each task and driver type, and the distributions of each feature's relative contributions are drawn.

Figure 6.5 shows the feature ranking based on mean relative contribution for the individual e_h prediction task for HER and ABK.

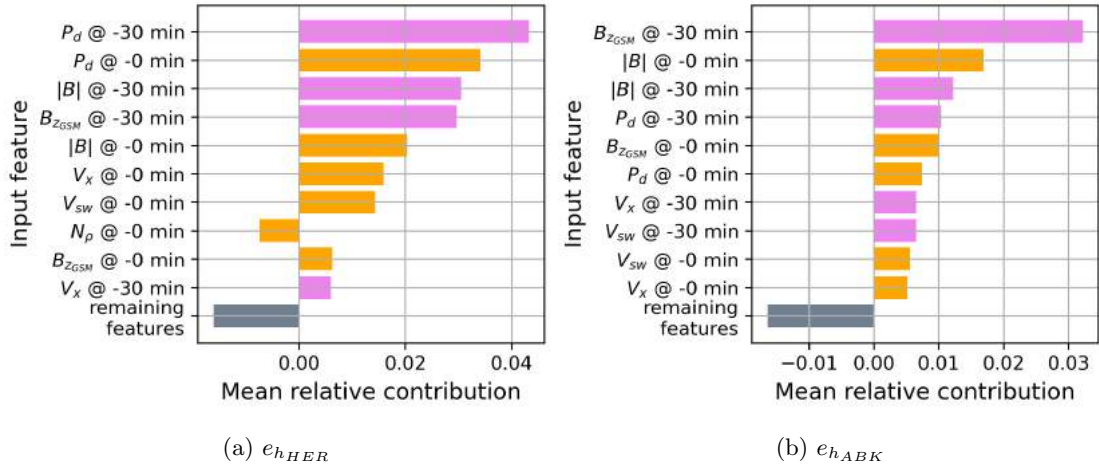


Figure 6.5: Feature ranking plot according to mean relative contribution for a set of CME-driven storms at HER and ABK

During CME storms, we can determine from Figure 6.5 that flow pressure (P_d) and overall magnetic field magnitude ($|B|$) are more important at HER, and that directional magnetic field components such as B_z are more important at ABK. Additionally, at ABK, time-lagged B_z at

-30 minutes is the highest contributing feature, with both lagged and immediate flow pressure P_d being the highest contributing features at HER. Solar wind speed V_{sw} and V_x have higher relative contributions at HER than at ABK.

Figure 6.6 shows the feature ranking plots for the individual e_h prediction task at both HER and ABK by mean relative contribution for a set of 3 CIR-driven storms that meet the selection criteria.

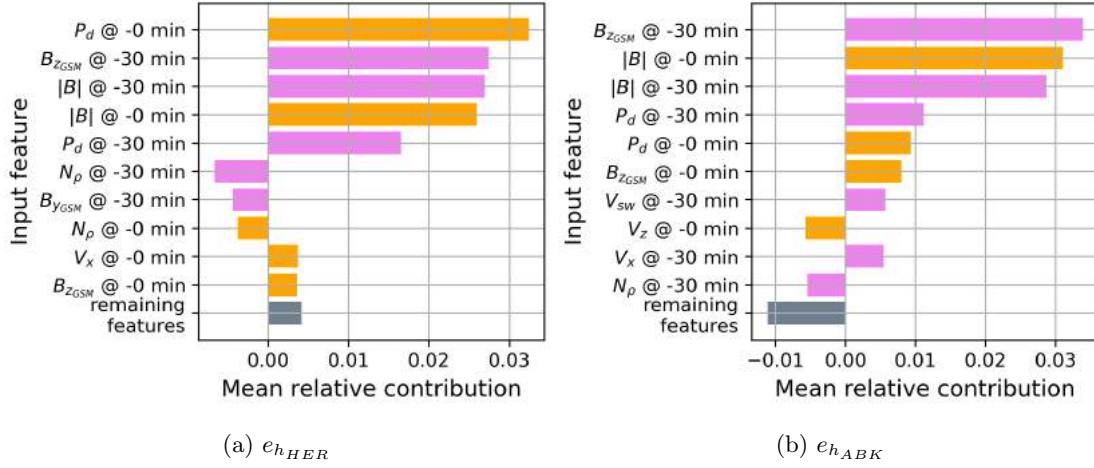


Figure 6.6: Feature ranking plot according to mean relative contribution for a set of CIR-driven storms at HER and ABK

While directional magnetic field component B_z is still quite dominant at ABK, for CIR-driven storms, flow pressure P_d and overall magnetic field magnitude $|B|$ become more dominant at both latitudes.

When considering the same for simultaneous prediction tasks, the feature ranking plots are remarkably similar to the respective individual prediction task plots across both CME-driven and CIR-driven storm sets. Figures 6.7 and 6.8 show the feature ranking plots according to mean relative contribution for the simultaneous e_h prediction task for CME-driven and CIR-driven storms respectively.

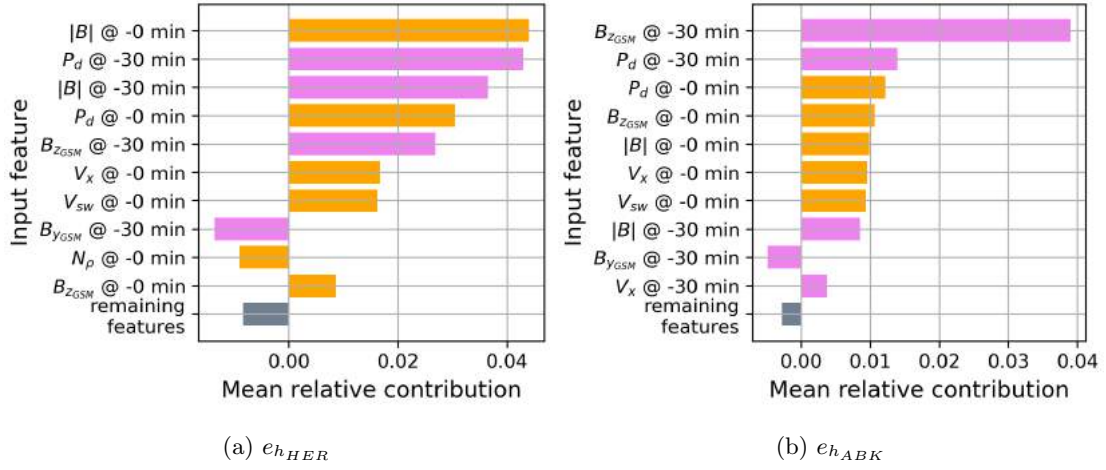


Figure 6.7: Feature ranking plot according to mean relative contribution for a set of CIR-driven storms at HER and ABK

When observing Figure 6.7, the majority of the differences in the feature rankings for HER when compared to the feature rankings for HER in Figure 6.5 are in the 'shuffling' of the relative importance of P_d and $|B|$, with immediate $|B|$ taking over the top spot from lagged P_d . The feature rankings for ABK remain the same in terms of B_z , with P_d otherwise increasing in importance when compared to the individual prediction task. The general trend of P_d and $|B|$ importance at HER and B_z importance at ABK still holds, with these shifts otherwise ascribable to differences in the respective TSMLP models.

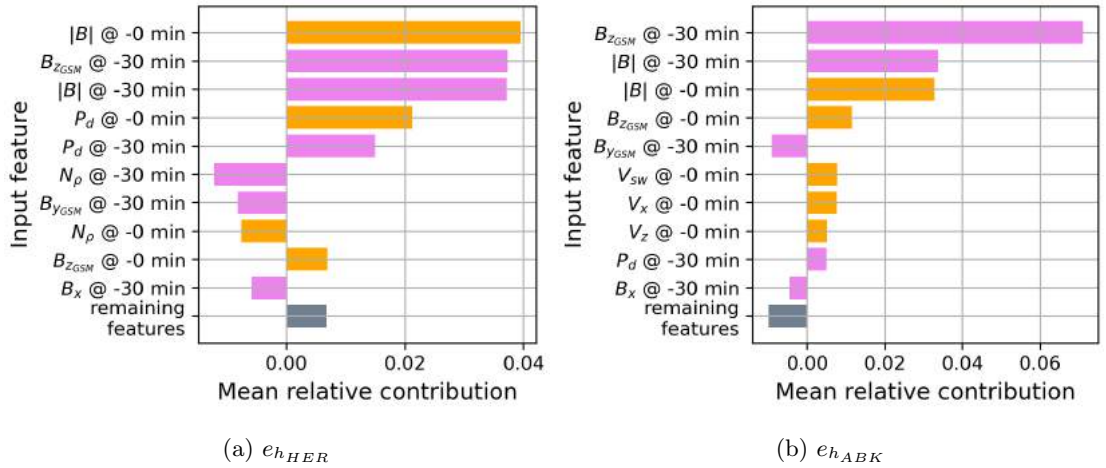


Figure 6.8: Feature ranking plot according to mean relative contribution for a set of CIR-driven storms at HER and ABK

When observing Figure 6.8, the trends seen in Figure 6.6 at each latitude hold remarkably well,

with some differences in importance between lagged features vs. immediate features. At $e_{h_{HER}}$, immediate flow pressure (P_d) and immediate overall magnetic field ($|B|$) 'swop' importance with the majority of other features remaining the same. At $e_{h_{ABK}}$, B_z retains its overall ranking and increases in importance, with P_d decreasing in importance in favour of other features.

The feature rankings according to the mean relative contribution to the explicit difference in e_h ($|\Delta e_h|$) for CME-driven and CIR-driven storms is shown in Figure 6.9.

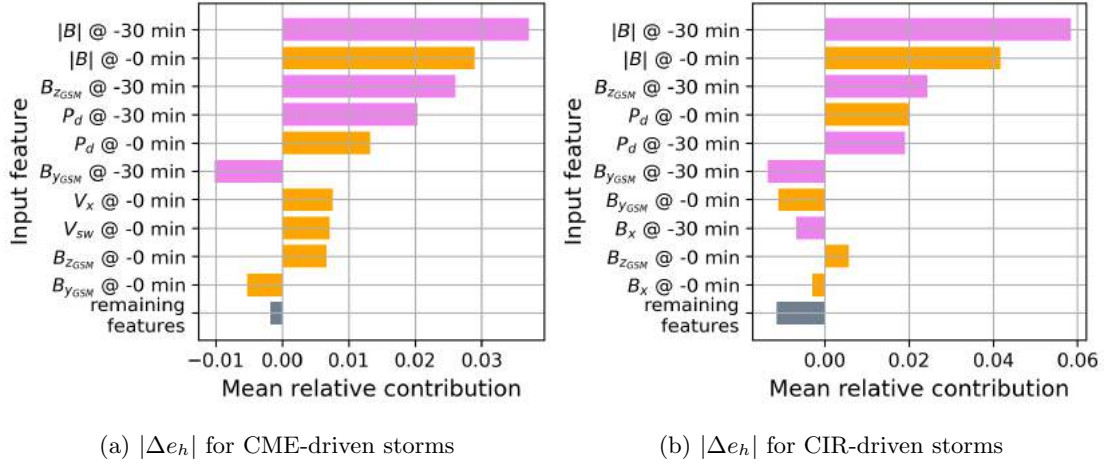


Figure 6.9: Feature ranking plot according to mean relative contribution for a set of CME-driven storms and a set of CIR-driven storms for $|\Delta e_h|$

When observing these rankings, it appears that the contribution to the explicit absolute difference between latitudes is the importance of both immediate and lagged $|B|$ during storms of either driver type, followed by the importance of lagged B_z and both immediate and lagged P_d .

The distribution of the relative contributions of each input feature for all e_h prediction tasks for these sets of CME-driven and CIR-driven storms are available in Appendix A.4.4.

6.4.2 A CME-driven storm

The example CME-driven storm we use to discuss the DeepSHAP attributions here is 2001_severe_005. All the attribution and feature ranking plots for this storm are in Appendix A.4.3.

Figure 6.10 shows the full time-series attribution plot of all input features for when $e_{h_{HER}}$ is predicted during 2001_severe_005, and Figure 6.11 shows the same for $e_{h_{ABK}}$, in the individual e_h

prediction task. Figure 6.12 shows the feature rankings according to mean relative contribution for the individual e_h prediction tasks at HER and ABK. Figures 6.13 and 6.14 show the feature attribution plots of the individual ‘top 4’ input features.

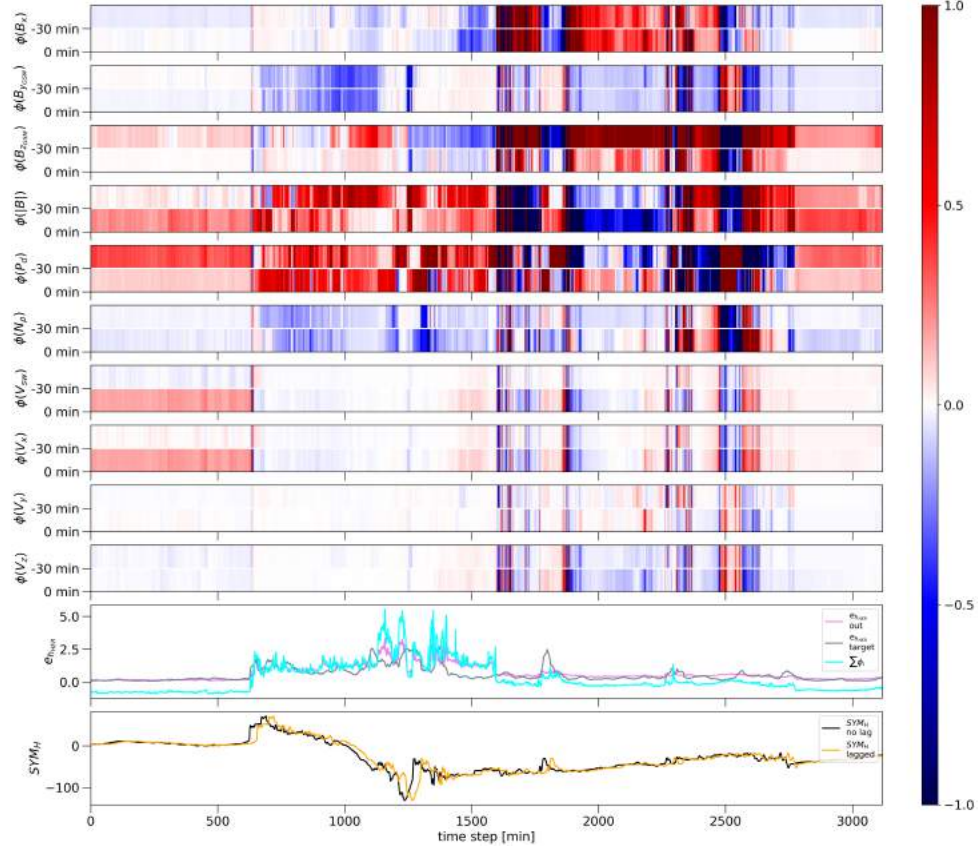


Figure 6.10: Time-series attribution plot of all input features for the prediction of $e_{h_{HER}}$ for 2001_severe_005. Each individual input feature is represented by a colour map - with the y-axis of each being the time axis into the past (0 minutes and -30 minutes respectively) and the x-axis of the plot being the time axis in the direction of the storm’s progression.

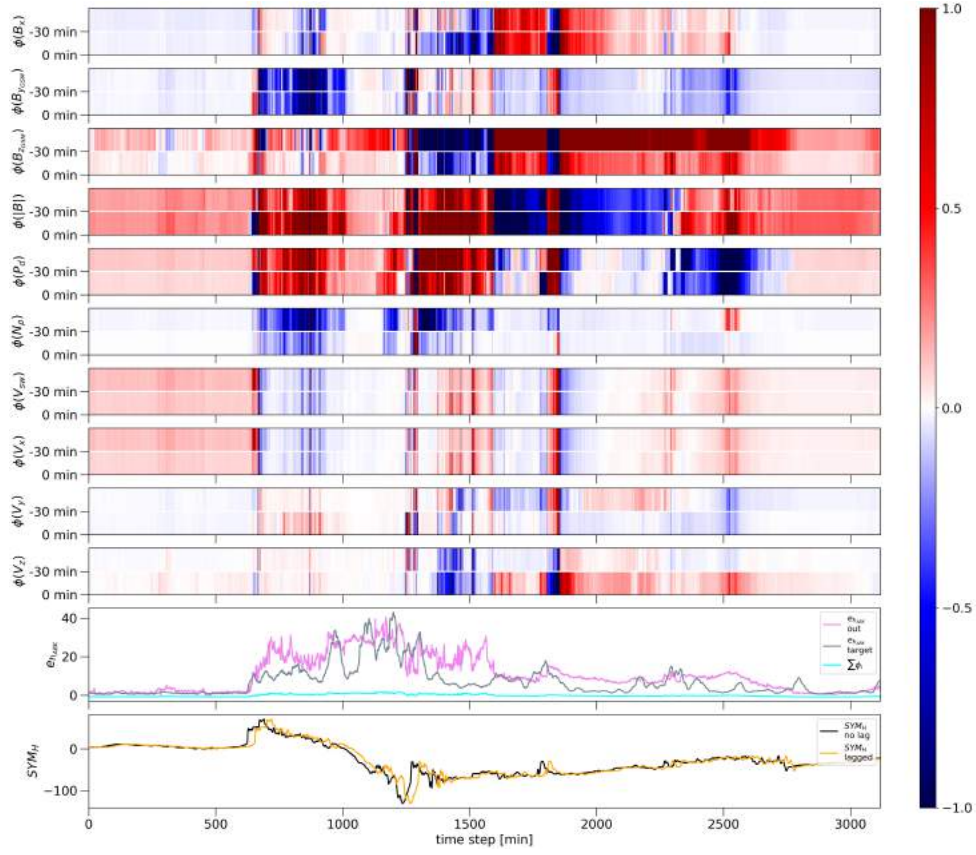


Figure 6.11: Time-series attribution plot of all input features for the prediction of $e_{h_{ABK}}$ for 2001_severe_005.

Observing the feature rankings in Figure 6.12, we note that there are four features in particular at each latitude that have strong relative contributions to e_h on average, with lagged P_d having the largest contribution to $e_{h_{HER}}$, and B_Z having the largest contribution to $e_{h_{ABK}}$. To provide additional insight into the dynamics of the topmost influential features at each latitude, individual feature attribution plots for B_z , $|B|$, and P_d , along with N_ρ and V_{sw} , with B_Y replacing P_d at ABK, are shown in Figures 6.13 and 6.14.

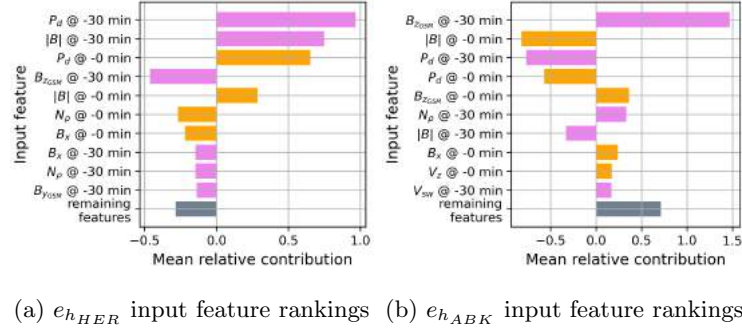


Figure 6.12: Input features ranked by mean relative contribution to the output for e_{hHER} and e_{hABK} predicted individually during 2001_severe_005.

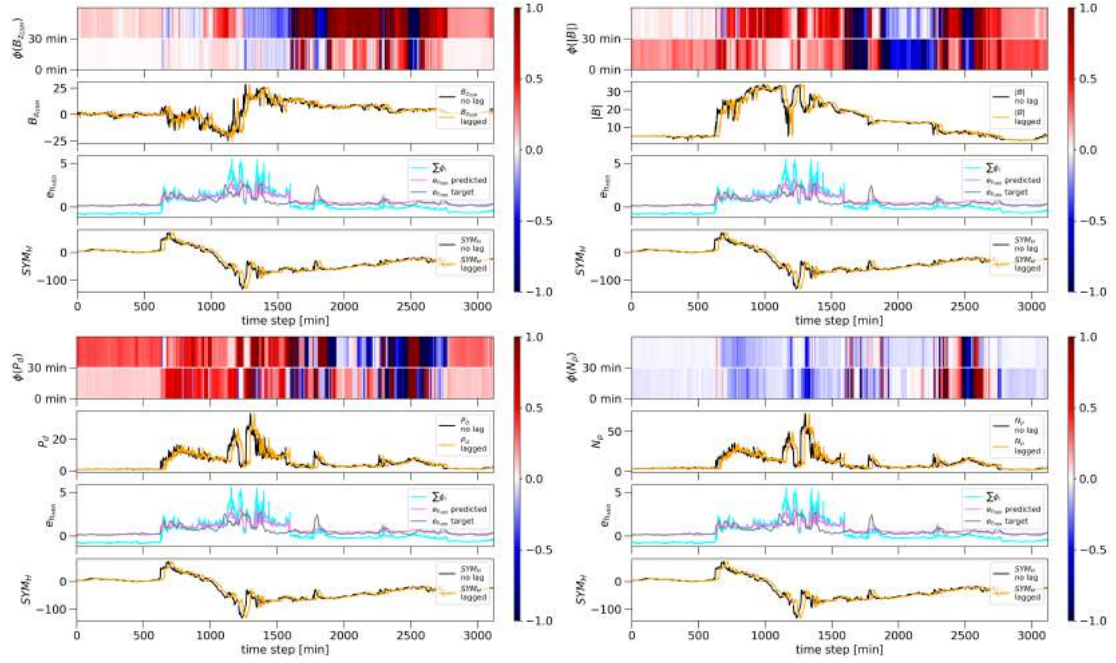


Figure 6.13: Attributions of B_z , $|B|$, P_d , and N_p at 0 and -30 minutes for e_{hHER} during the storm 2001_severe_005. A time-series plot of each input parameter as well as SYM-H are shown at a lag of 0 and -30 minutes alongside the attribution feature maps.

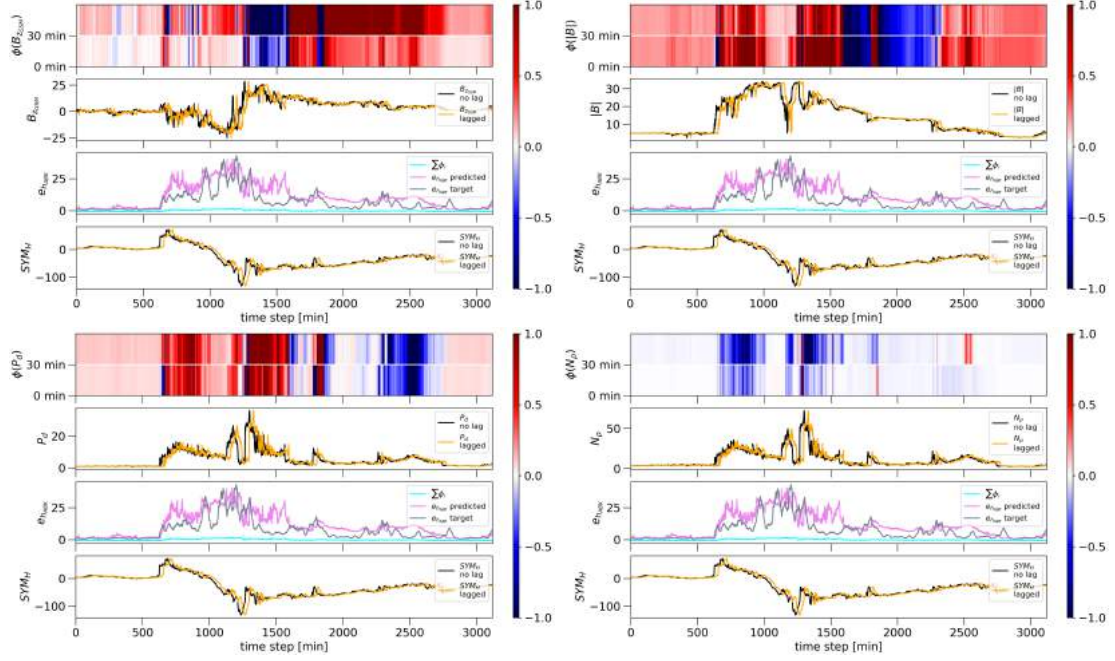


Figure 6.14: Attributions of B_z , $|B|$, B_Y , and N_p at 0 and -30 minutes for $e_{h_{ABK}}$ during the storm 2001_severe_005.

This being a CME-driven storm, the storm sudden commencement (SSC) is clearly visible at around minute 600 by the positive spike in SYM-H but also by in the changes in input attribution at the same time.

B_Z suddenly drops in value to -25 nT during the main phase of the storm, before switching polarity (jumping in value to +25nT) at the end of the main phase as the CME passes Earth and decouples from the magnetosphere - with B_Z slowly returning to a value of 0 nT during the recovery phase. $|B|$, on the other hand, spikes in value massively with a 20+ nT increase at the storm's onset, steadily increasing in value throughout the main phase of the storm. As soon as the CME passes, $|B|$ once again slowly recovers to 10 nT.

P_d and N_p , being flow pressure and proton density, are very closely related to one another ($P_d \sim V_{sw}^2 N_p$) and have approximately the same progression throughout the storm, with sudden spikes in value at the storm onset that is sustained before two large spikes and falls in value coinciding with the end of the main phase and the predicted spikes in e_h at both latitudes, before returning to a level close to what it was before the storm, with intermittent, short, small increases in value throughout the recovery phase.

At both latitudes, B_Z is similarly attributed over time, with B_Z at -30 minutes bearing more

importance than B_Z at 0 minutes at both latitudes. This would suggest a sort of memory or loading mechanism at play during dayside magnetic reconnection. However, the difference between time-lags at HER is a bit more stark. Additionally, at HER, B_Z 's contribution only truly begins to strengthen during the main phase, where the declining value of B_Z is positively attributed. The small 'spike' in B_Z at the storm's onset and the polarity flip signaling the start of the recovery phase both make strong negative contributions to e_h , with B_Z generally being positively attributed during the rest of the recovery phase. Note that a negative attribution means that at that time the predicted value of e_h is *decreased* due to the influence of B_Z , while a positive attribution means that the predicted value of e_h is *increased* due to the influence of B_Z . This goes for all input features.

At ABK the contributions to e_h before the main phase are larger than they are at HER. The attribution of B_Z as dayside reconnection ceases and the CME plasma decouples from the magnetosphere is much stronger at ABK than at HER, showing a much stronger reliance on the magnetospheric coupling/decoupling itself at the high latitude station.

While being similar at both time-lags, the attributed behaviour of $|B|$ at the two latitudes is somewhat different. At HER, we once again see a brief negative attribution at the initial onset 'spike', with $|B|$ otherwise strongly positively contributing during the main phase of the storm. During the recovery phase, as the CME passes and $|B|$ steadily recovers in value, there is initially a strong negative contribution that gradually weakens, and eventually, the recovery phase tapers out with a positive contribution containing one rogue 'burst' of negative attributions. At ABK, there are stronger positive attributions over-all (especially in the main phase), with similar behaviour at the initial onset 'spike' and the start at the recovery phase to HER.

Although P_d and N_ρ have a similar progression throughout the storm, they are attributed quite differently from one another at HER in particular. P_d is generally given positive attributions throughout the storm, with short bursts of negative attributions around spikes in its value at the start of the recovery phase and throughout. There is an increase in intensity of the positive attributions during the main phase, with little difference between time-lags. N_ρ on the other hand, is negatively attributed throughout the storm, with a brief positive attribution at the storm onset and at scattered points during the recovery phase. At ABK, N_ρ shows concentrated negative attributions during the main phase of the storm, with some bursts of positive attributions during the recovery phase with much weaker attributions everywhere else. P_d at

ABK shows greatly similar attributions between time-lags: with strongly positive attributions during the main phase, bursts of negative attribution during the recovery phase, and weaker positive/negative attributions everywhere else.

Attributions were also generated for the simultaneous e_h task, whose input feature rankings are shown in Figure 6.15, and for $|\Delta e_h|$, whose input feature rankings are shown in 6.16.

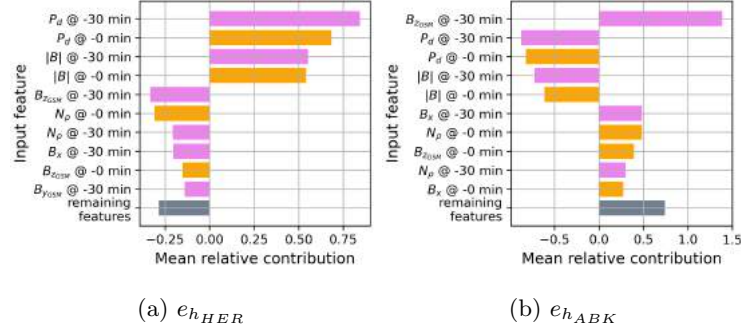


Figure 6.15: Input feature rankings for $e_{h_{HER}}$ and $e_{h_{ABK}}$ predicted simultaneously.

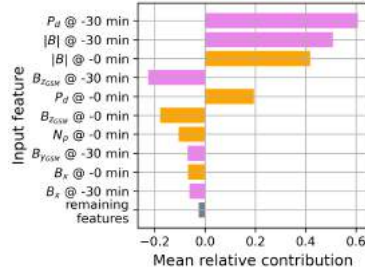


Figure 6.16: Input feature rankings for $|\Delta e_h|$

Observing the feature rankings in Figures 6.15 and 6.16 and comparing them with the feature rankings in Figure 6.12, there are a few points to highlight.

Namely, P_d , B_Z , and $|B|$ all still enjoyed ‘top 4’ status in these new rankings. B_Z was still the top contributor to $e_{h_{ABK}}$, and P_d remained highly attributed at $e_{h_{HER}}$. In general, the ‘top 4’ ranked features at both latitudes experienced some re-ordering, but still contributed relatively to the same degree as they did when e_h at each latitude was predicted individually.

6.4.3 A CIR-driven storm

The example storm selected to discuss CIR-driven storms here is 1997_moderate.001. Figure 6.17 shows the time-series attribution plot of all input features when $e_{h_{HER}}$ is predicted, and Figure 6.18 shows the same for $e_{h_{ABK}}$. Figure 6.19 shows the feature rankings during 1997_moderate.001 for when $e_{h_{HER}}$ and $e_{h_{ABK}}$ are each predicted individually.

When observing Figures 6.17 and 6.18, it is immediately apparent that while the CME example had its strongest attributions concentrated during the main and onset phases of the storm predominantly, the attributions for this CIR storm are quite a bit more ‘spread out’. One can notice especially the consistent treatment of P_d , $|B|$, B_Z and V_{sw}/V_x at both latitudes during the storm’s progression, with the major differences being the higher emphasis on V_{sw}/V_x at 0 minutes than at -30 minutes when considering HER, and the difference in strength of the negative attributions of $|B|$ and P_d between the two latitudes.

As shown in Figure 6.19, features that are strongest across both latitudes are: $|B|$, P_d , and V_x/V_{sw} at both 0 and -30 minutes. (We see that at ABK, B_Z is stronger than V_{sw}/V_x , and N_ρ has the same ranking as V_{sw}/V_x .) Interestingly, these features, $|B|$ in particular, are quite well grouped at the top of both feature rankings, with P_d having a stronger contribution at $e_{h_{HER}}$ than at $e_{h_{ABK}}$.

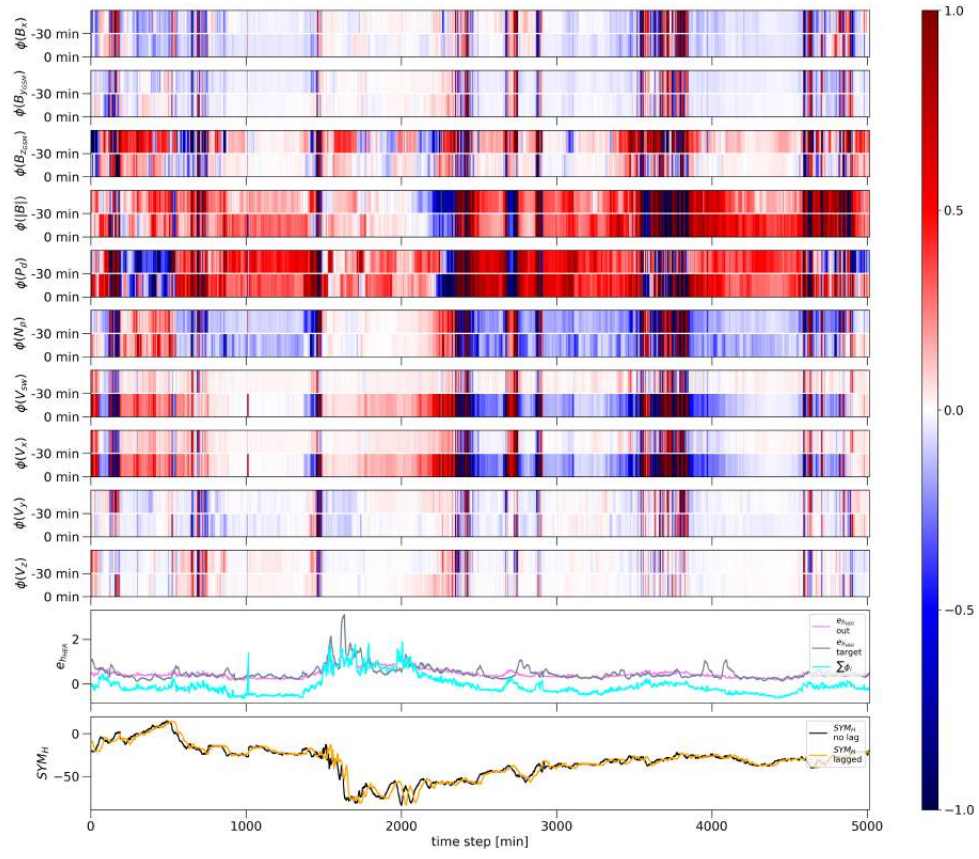


Figure 6.17: Time-series attribution plot of all input features for the prediction of e_{hHER} for 1997_moderate_001.

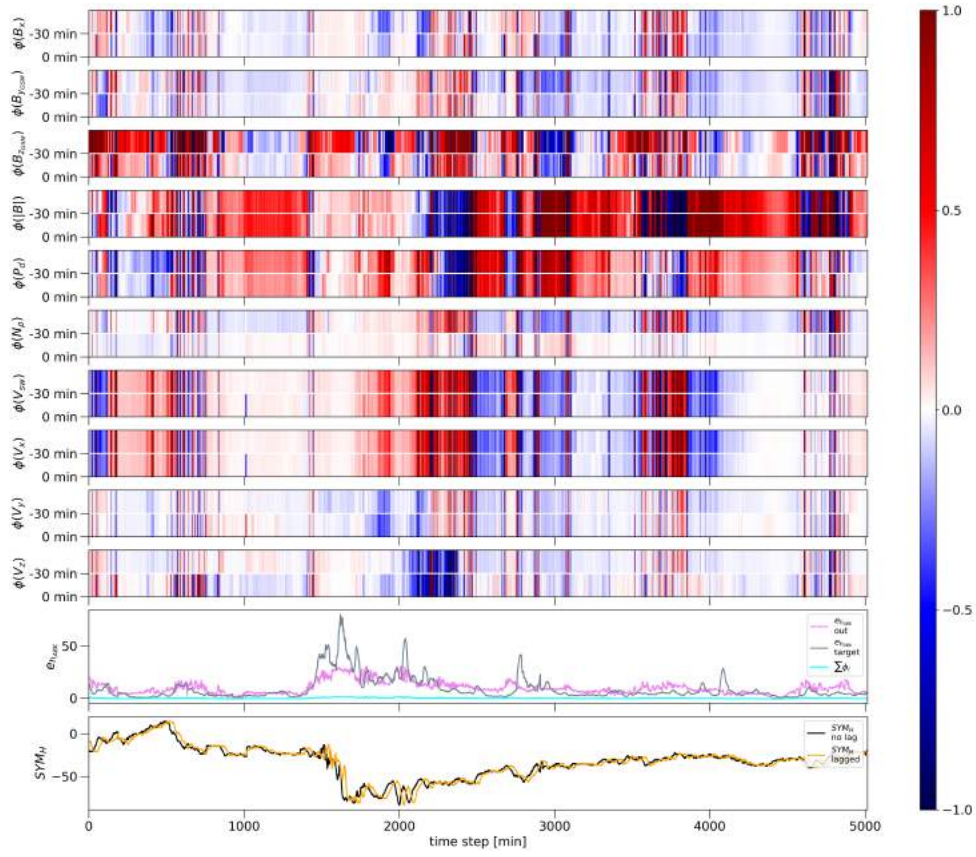
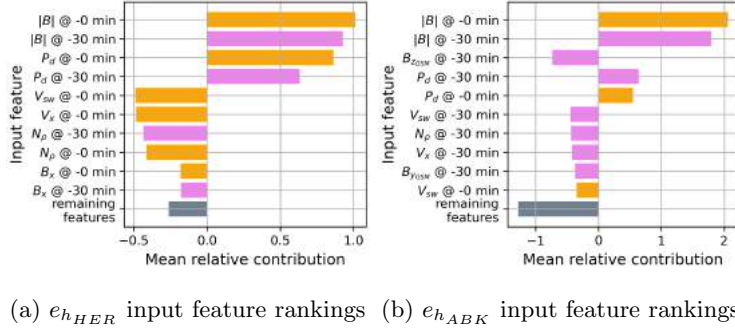


Figure 6.18: Time-series attribution plot of all input features for the prediction of e_{hABK} for 1997_moderate_001.



(a) e_{hHER} input feature rankings (b) e_{hABK} input feature rankings

Figure 6.19: Input features ranked by mean relative contribution to the output for e_{hHER} and e_{hABK} predicted individually during 1997_moderate_001.

Figures 6.13 and 6.14 provide the time-wise attribution plots of $|B|$, P_d , and V_{sw} at each latitude, offering a more granular view. As was observed in Figures 6.17 and 6.18, the top three features are quite similarly attributed across latitudes, with some differences in intensity.

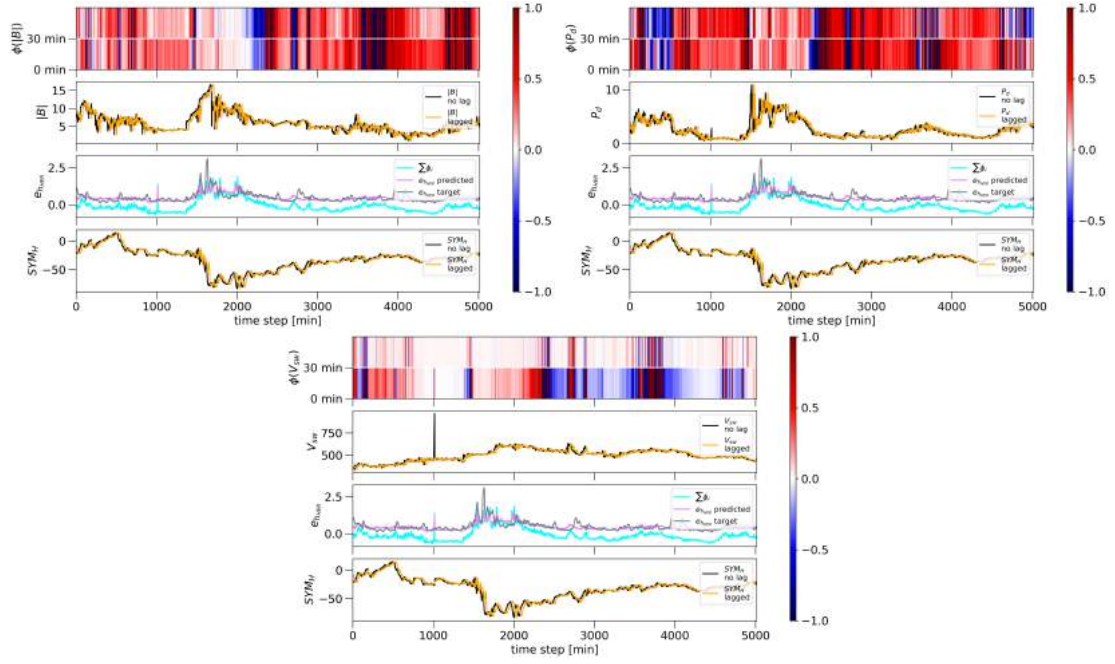


Figure 6.20: Attributions of $|B|$, P_d , and V_{sw} at 0 and -30 minutes for $e_{h_{HER}}$ during the storm 1997_moderate_001.

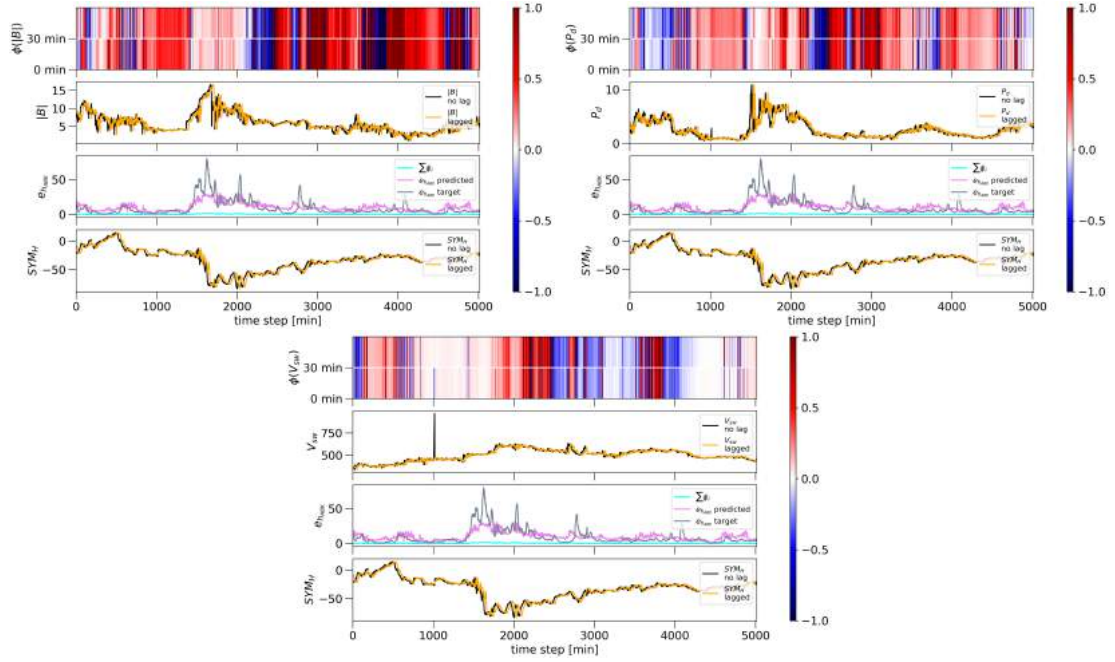


Figure 6.21: Attributions of $|B|$, P_d , and V_{sw} at 0 and -30 minutes for $e_{h_{ABK}}$ during the storm 1997_moderate_001.

During the onset phase, there is no sudden spike/shock and drop in SYM-H, but a gradual increase in particles and flow at different points throughout the storm (approx. 0-500 minutes). As a strong increase of flow pressure (P_d) commences at around 1700 minutes into the storm,

SYM-H drops as the main phase begins. This corresponds with an almost mirrored increase in $|B|$. Both of these increases are followed by a decrease in value at about 2500 minutes into the storm, and the recovery phase commences. V_{sw} shows a steady increase in value during the storm onset and main phase, peaking and then slowing down very gradually as the recovery phase commences, with the solar wind mostly maintaining its speed during the storm recovery.

At both latitudes, the elevated value of V_{sw} during the recovery phase is negatively attributed to the same degree at both time-lags for ABK and at the immediate time at HER, with lagged V_{sw} being generally lightly positively attributed at HER. Before the storm onset, V_{sw} is positively attributed at ABK, and negatively attributed during the same period of time at HER, while both cases are followed by a period of basically no attribution at all until the very start of the main phase. At both latitudes, the V_{sw} is positively attributed during the main phase.

Both P_d and $|B|$ mostly share attributions. Before the onset phase, there is an initial blip in flow pressure and magnetic field magnitude. This blip is negatively attributed (this blip for $|B|$ is mostly positively attributed at HER), and is then followed with a period of mostly positive attribution right through to the end of the main phase. At HER, $|B|$ experiences much weaker attributions at the period of 1800 to 2000 minutes in favour of P_d , however, both are very strongly negatively attributed at the start of the recovery phase, followed by positive attribution interspersed with negative attributions at the same time-points for the remainder of the the storm. At ABK, the negative attributions for P_d are weaker, and the difference in attribution between P_d and $|B|$ during the main phase of the storm are quite small. However, during the recovery phase, $|B|$ enjoys a stronger positive attribution than P_d does.

Attributions are also generated for the simultaneous e_h task, whose input feature rankings are shown in Figure 6.22, and for $|\Delta e_h|$, whose input feature rankings are shown in 6.23.

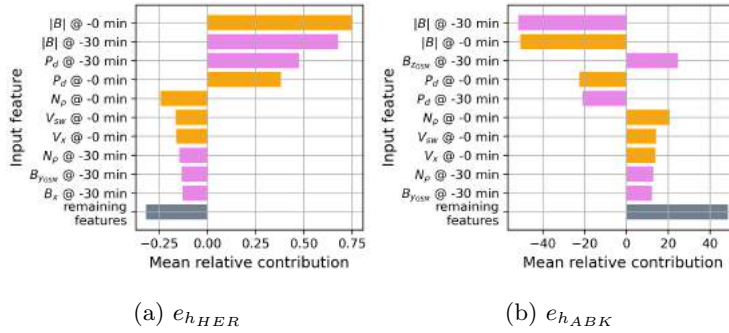


Figure 6.22: Input feature rankings for $e_{h_{HER}}$ and $e_{h_{ABK}}$ predicted simultaneously during 1997_moderate_001.

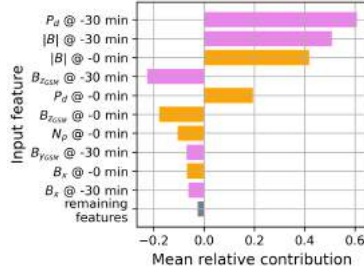


Figure 6.23: Input feature rankings for $|\Delta e_h|$ predicted during 1997_moderate_001.

Once again, as we saw in Section 6.4.2, the same features that were highly ranked in the individual prediction tasks once again enjoy more or less the same ranking and relative attribution value, with some differences.

6.4.4 The effect of auto-regressive input features

Taking the very same tasks and data set split, the attributions generated for TSMLP models trained with additional auto-regressive input features included are discussed here using the storms 2001_severe_005 and 1997_moderate_001.

It was suspected, and confirmed by calculating the mode time-lag of the output values in Section 5.6, that these TSMLPs relied heavily on the auto-regressive input features to improve their accuracy. When observing Figures 6.24 and 6.25, this reliance on the auto-regressive feature is especially obvious.

However, interestingly, when comparing the feature rankings of both examples with their non auto-regressive counterparts, the same features generally still rank in the same positions. This is barring some differences that may be induced by the individual attributions differing at particular time-steps. Also interesting to note is the relative dependence on auto-regressive $e_{h_{ABK}}$ vs. auto-regressive $e_{h_{HER}}$. In both the CME and CIR examples, auto-regressive $e_{h_{ABK}}$ is ranked as far more important to the $e_{h_{ABK}}$ prediction task than auto-regressive $e_{h_{HER}}$ is to the $e_{h_{HER}}$ prediction task. This once again speaks to the relative complexity of the magnetospheric coupling at ABK compared to the coupling at HER, which is more magnetically sheltered and has less extreme reactions to changes in the magnetosphere.

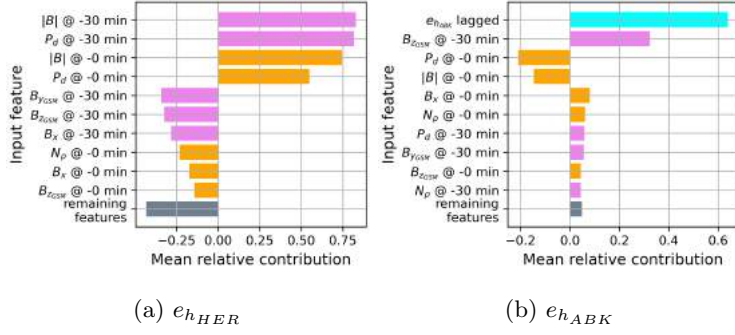


Figure 6.24: Input features ranked by mean relative contribution to the output for $e_{h_{HER}}$ and $e_{h_{ABK}}$ predicted individually during 2001_severe_005 with auto-regressive input features included.

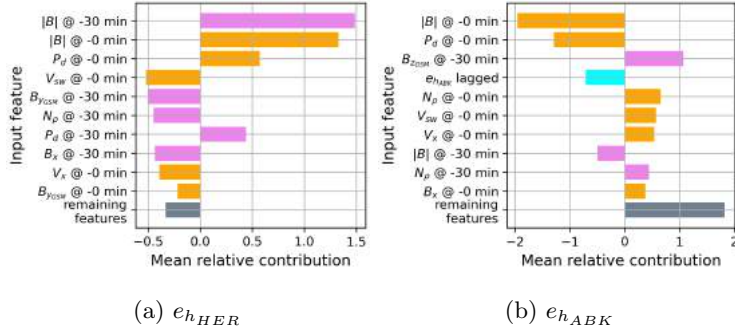


Figure 6.25: Input features ranked by mean relative contribution to the output for $e_{h_{HER}}$ and $e_{h_{ABK}}$ predicted individually during 1997_severe_001 with auto-regressive input features included.

Here again, the differences in important features between the two latitudes are consistent with the suggestion that $e_{h_{HER}}$ is more reliant on flow pressure, particle density, and overall field strength, with $e_{h_{ABK}}$ more reliant on individual magnetic field components such as B_Z . Similarly, barring more individual differences between hemispheres being highlighted with auto-regressive inputs included, the most important features for the CIR-driven example remain overall magnetic field strength, proton density, flow pressure, and solar wind speed, with proton density (N_p) once again also featuring more strongly here when auto-regressive input features are included.

6.5 Discussion

In general, the storms discussed here adhered to the expectation that magnetic components would feature more strongly at ABK than at HER during CME-driven storms. When auto-regressive input features were included, the general ‘global’ importance of features remained

similar in terms of mean relative contribution throughout a whole storm. The ABK models relied far more heavily on lagged $e_{h_{ABK}}$ when auto-regressive inputs were included than the HER models did on lagged $e_{h_{HER}}$.

This supported the fact that the relationship between $e_{h_{ABK}}$ and the solar wind drivers is more complex than the relationship between $e_{h_{HER}}$ and those same drivers. This expectation is created both by an understanding of the more exposed nature of ABK, as well as the performance of the TSMLP models trained to predict $e_{h_{ABK}}$ compared to that of the models trained to predict $e_{h_{HER}}$ as seen in Chapter 5.

The general picture painted by the attributions discussed in this chapter (and those in the Appendix), is that during CME-driven events, IMF components have a stronger and more direct effect on the induction of e_h at ABK, with overall magnetic field magnitude and solar wind flow pressure/proton density/speed having a greater effect at HER, which is less exposed to direct solar wind coupling and more to ring current enhancement [15]. This was also apparent when considering the aggregate feature ranking plots for CME-driven and CIR-driven storms in Section 6.4.1. This held when generating input feature attributions for TSMLP models that predicted e_h at both latitudes simultaneously - even when regarding the differences in intensity and/or time-lag importance, input features were similarly ranked across individual and simultaneous e_h prediction tasks.

For CIR-driven events, the difference between latitudes in terms of *which* input features were more important at ABK vs. HER was replaced with the difference in to what *degree* the same major input features had importance at each latitude. During CIR-driven events, P_d , $|B|$, N_p , V_x/V_{sw} and B_Z all featured heavily at both latitudes, with $|B|$, directional B components such as B_Z , and N_p featuring more strongly at ABK, but, in general, $|B|$, P_d and V_{sw} contributed the most to e_h during these CIR-driven events.

Upon visual inspection, the feature attributions for the models predicting $|\Delta e_h|$ very closely matched the would-be differences in the attributions for those same features between HER and ABK when comparing the feature attribution rankings of singular storms for individual e_h at each latitude to the feature attribution rankings for $|\Delta e_h|$. To a lesser degree, this also holds when comparing the feature attribution rankings for simultaneous e_h at each latitude to $|\Delta e_h|$'s feature attribution rankings. When comparing the same for the aggregate results in Section

6.4.1, however, this observation did not hold. This could be due to the effect of combining individual storm attributions before calculating the feature rankings.

It can be seen in the majority of cases that separate models trained on variations of the same data set (i.e., for different tasks) have variations of the same attributions for shared input features. This was especially apparent when comparing the attributions between aggregate results in Section 6.4.1. This indicates a *consistency* (Section 2.4.2) of the generated attributions besides the alignment with the domain-expert expectations of the attributions. Furthermore, when observing any of the attributions of V_x and V_{sw} in Figures 6.10, 6.11, 6.17 and 6.18 or in the Appendix, one notes that two input features that are precisely negatively correlated with one another (Figure 4.7 in Section 4.3.3) and describe the same quantity have the same attributions at each time-step. This further points to a *consistency* across models for treating these two features as if they describe the same quantity measured from two different directions in all of these cases.

One can see when observing those same attribution plots that there are some parameters, especially $|B|$ and P_d , that are heavily correlated with one another at different times throughout the storm’s progression, and this apparent correlation, again, is consistent across storms of the same driver/intensity. If one refers to the attribution plots in the Appendix, P_d and $|B|$ correlate quite strongly during the pre-onset and recovery phases in CIR-driven storms and to some extent at those same points during several CME-driven storms, as indicated by their shared attributions during these periods. At CME-driven storms, however, there are several cases where only P_d at -30 minutes shares attributions at these points with $|B|$.

This is expected as the IMF magnitude $|B|$ typically increases as the higher density CME or CIR plasma passes over the magnetosphere. The direction of the IMF, mainly in the Z component is a known feature of CME magnetic field and serves to “unlock” the magnetosphere by dayside reconnection. The main driver of middle latitude (i.e. HER) geomagnetic disturbances is the enhancement of the ring current, which occurs through pressure increases (CIR and CME) as well as through reconnection (CME storms) [15].

6.6 Chapter summary

In this chapter, we discussed and interpreted the explanations produced by DeepSHAP when attributing the input feature contributions to e_h at two latitudes. Two example storms were used in this discussion: one CME-driven storm, and one CIR-driven storm. This was done to exemplify the relationships of the solar wind drivers to e_h in two discrete situations and enable a comprehensive comparison of the DeepSHAP attributions to the expectations of those relationships based on the physical realities.

CME and CIR-driven storms were identified by comparing their commencement and end dates to the dates of events available in several CME and CIR event catalogs. CME-driven storms are characterised by a sudden compression of the dayside magnetosphere due to the increased pressure of the CME ejecta, followed by rapid changes in directional B (B_Z , B_Y) components, $|B|$, and flow pressure (P_d). It was expected that, due to the coupling of the solar wind with the magnetosphere by reconnection, B_Z at both latitudes will be more important at -30 minutes than at 0 minutes [16]. It was also expected that directional B components, B_Z in particular, $|B|$, and N_p would have a more pronounced effect at ABK than at HER, due to ABK being more directly exposed to the solar wind, with P_d more influential at HER than the magnetic parameters since it is more magnetospherically shielded. These expectations held when observing the DeepSHAP attributions generated for the CME-driven storm in the individual e_h prediction task, as well as for the attributions generated in the simultaneous e_h prediction task.

For CIR-driven storms, it was expected that overall magnetic field, flow pressure, and solar wind speed would have the highest importance to different extents at both latitudes. This held true for the CIR-driven storm example, with these parameters featuring at both latitudes to different degrees (with V_{sw} more important than P_d at ABK and P_d more important than $|B|$ at ABK).

Additionally, when observing the feature rankings for $|\Delta e_h|$ according to mean relative contribution, the mean relative contributions of these features (in both the CME and CIR-driven storm cases) were visually congruent with the difference between the mean relative contributions of the same feature between latitudes.

While the TSMLP model was not the most perfect approximation of the solar wind/ e_h relationship, and there were some differences in the attributions of the same features between tasks, in

general, the DeepSHAP attributions made sense when regarding the physical reality, matched up with the expectations in most cases, and even showed consistency across tasks and when attributing the explicit difference in e_h , $|\Delta e_h|$.

Chapter 7

Conclusion

In this chapter we summarize key findings and discuss their implications. We also reflect on possibilities for future work.

7.1 Introduction

The goal of this study was to determine the extent to which an additive attribution method, specifically DeepSHAP, in combination with a DNN architecture such as an MLP, would be able to assist with knowledge discovery on a solar physics task. In particular, we trained a set of TSMLP models (MLP with time-shifted inputs) to predict the target geomagnetic index, e_h , at two latitudes, to determine if the DeepSHAP attributions generated for these models would describe what is expected from a physics perspective.

In this chapter we list our key findings and examine how well our objectives set out in Chapter 1 were met. We also list the contributions that this study made, and examine possible avenues for future work.

7.2 Key findings

This section lists and discusses the implications of the key findings of this study. We begin with the findings from Chapter 3, where a synthetic data set was used to provide a ground-truth evaluation of DeepSHAP:

- When an MLP model was trained to the best possible accuracy on a synthetic data set, the trends of the DeepSHAP attributions exactly matched the trends of the transformed input features.
- When this same MLP model was trained with some differences (the exclusion of a bias node) to a lesser accuracy, the DeepSHAP attributions trends no longer exactly matched the trends of the transformed input features, but the underlying trends of the transformed input features were still discernible in the attributions.

During the EDA process conducted in Chapter 4, we found that:

- Particle-related solar wind parameters, such as P_d , N_ρ , and V_{sw} , as well as overall magnetic field magnitude, $|B|$, had a stronger relation to e_h at HER than to e_h at ABK.

When training the TSMLP models to predict e_h at either or both latitudes, or to predict $|\Delta e_h|$ in Chapter 5, we found:

- In general, TSMLPs that were trained to predict e_h at HER alone did better than the TSMLPs trained to predict e_h at ABK alone whether or not auto-regressive input features were included. This supported the fact that the relationship between e_h at ABK and the solar wind drivers is more complex than the relationship between e_h at ABK and those same drivers.
- TSMLPs trained to predict e_h at both hemispheres simultaneously, as well as those trained to predict $|\Delta e_h|$, performed with a ranking similar to the TSMLPs trained to predict e_h at either hemisphere alone. This indicates that it is possible to model the *explicit difference* in e_h between latitudes to a satisfactory degree.

-
- Several time-shift lengths were tested for these TSMLPs models, at 30, 60, 90, and 120 minutes. However a time-shift size of 30 minutes consistently out-performed the larger time-shift sizes. This coincides with the length of e_h 's smoothing window, but may also indicate that shorter time-shift intervals are more meaningful than longer-term changes.
 - Including auto-regressive inputs (time-shifted instances of e_h) improved the performance of the TSMLPs, and for TSMLPs trained to predict e_h at ABK in particular. However, we also found (understandably) that the TSMLPs relied very heavily on these auto-regressive inputs. This could be due to the ‘missing link’ in e_h prediction in the form of a lack of measurements of the actual magnetospheric coupling being filled by the inclusion of these auto-regressive input features, or due to the models simply over-fitting on the auto-regressive input features.

When generating DeepSHAP attributions for these TSMLPs we found that:

- For CME-driven storms, directional B components were much more important to e_h at ABK than to e_h at HER, and overall $|B|$, P_d , and other particle-related features were more important to e_h at HER. This supported the expectation of this behaviour since HER would be more magnetospherically shielded than ABK, so would be less reactive to directional B components and more dependent on overall magnetic field magnitude ($|B|$) and particle-related features such as flow pressure, particle density and solar wind speed.
- For CIR-driven storms whose drivers are generally a gradual increase in intensity of ambient solar wind throughout the progression of the storm, we noted that the same features were broadly attributed as important at both latitudes. $|B|$, P_d , and (to a lesser extent) V_{sw} were the most important features at these latitudes, which corresponds with the expectations set by the mechanics of CIR-driven storms described in the solar physics literature. These particular features, when zoomed into on a time-series level, had different degrees of influence at different time-steps throughout the storm, but were generally similarly attributed across both latitudes.
- When considering the importance of historical feature values with attributed time-shifted features, magnetic components were more important at 30 minutes in the past than at present at both latitudes. This corresponds to the phenomenon of ‘magnetospheric loading’, where the changes wrought in the magnetosphere by the solar wind and geomagnetic

disturbance has a delayed effect on the ground-based geomagnetic field. Additionally, the differences between the attributions of the time-shifted magnetic components at ABK and HER were significant enough to say that while the time-lag is important at HER, the more magnetospherically exposed ABK is more sensitive to magnetospheric loading.

- Over all, even though the TSMLP models did not perfectly predict e_h or $|\Delta e_h|$ in its various tasks, we saw that the DeepSHAP attributions were consistent with the expectations set by the solar physics literature. Additionally, consistencies in attributions emerged between both variations in the data set (across e_h prediction tasks) as well as across storms within the same driver group (CME-driven storms were consistent with each other, and CIR-driven storms were consistent with one another). Interestingly, the average attributions input features for prediction of $|\Delta e_h|$ were often visually consistent with the difference in the average attributions for the same input features between latitudes. With this considered, it can be asserted that using DeepSHAP attributions as a knowledge discovery method is feasible.

7.3 Contributions

We found that under certain conditions, the use of DeepSHAP attributions generated for an MLP trained on a particular task for knowledge discovery is indeed feasible. While it was subject to many challenges (i.e., the TSMLPs not always perfectly modeling the solar wind driver/ e_h interaction), the DeepSHAP attributions fared well when evaluated according to the solar physics literature. We hereby introduce this attribution-based knowledge discovery methodology to be adapted to a variety of use cases.

Pending the finalisation of this study, the following data sets and software packages will be made publicly available:

- The data set that was curated and processed for this particular study (HAS) will be released for general use, and the code used for the pre-processing and conversion to PyTorch-compatible tensors will be released as an open source project.
- The methodology for transforming and displaying the local DeepSHAP attributions in the context of a time-series and in a more global sense with regards to the relative contribution

ranking plots seen in Chapter 6 and its associated code will be released as an open source project.

- The code used for the management and drawing of summary plots for SMAC3 hyper parameter optimisation results will be released as an open source project.

7.4 Future work

While the TSMLP proved sufficient to model the solar wind driver/ e_h interaction with the addition of a previous time-step, more in-depth and high-capacity time-series modeling will be required to fully capture these spatio-temporally dependent interactions. Further development of this project should focus on more high-fidelity modeling techniques and more quantitative evaluation of calculated attributions.

Future work could include the following:

- Training TCNs or transformer-based architectures to predict e_h with the same drivers to better encompass the relationship with greater time resolution and higher capacity.
- Developing attribution summarising and presentation techniques (in the same way as this study’s time-series attribution plots) for more complex time-series attributions.
- Create a more rigorous consistency-based attribution evaluation system based on attribution comparison between similar models and data sets and more quantitative comparison to known in-data relationships.

7.5 Conclusion

Recent rapid development of massive and opaque language models such as ChatGPT and the ever-increasing sophistication of deep recommender systems used on various social media platforms has made the need and desire for XAI and AI fairness an imperative. In the same vein, this rapid development of more powerful models by various affluent groups with access to massive amounts of computing power continues to anecdotally discourage some in the field of deep learning with the feeling that whatever one can do with the resources at hand is already obsolete.

We have demonstrated that a relatively simple TSMLP model, when trained on a solar physics task, and interpreted with a technique such as DeepSHAP, is able to accurately capture physical realities that correspond well to established knowledge of the area. The desire is that this knowledge-discovery technique will be recognised for its potential, adopted, and taken further to inspire many more niche and intriguing applications with the resources that are at hand.

Bibliography

- [1] S. I. Lotz, M. J. Heyns, and P. J. Cilliers, “Regression-based forecast model of induced geoelectric field,” *Space Weather*, vol. 15, no. 1, pp. 180–191, Jan. 2017, ISSN: 1542-7390. DOI: 10.1002/2016SW001518.
- [2] S. Wing *et al.*, “Kp forecast models,” *Journal of Geophysical Research: Space Physics*, vol. 110, no. A4, p. 4203, Apr. 2005, ISSN: 2156-2202. DOI: 10.1029/2004JA010500.
- [3] J. Uwamahoro and J. Habarulema, “Empirical modeling of the storm time geomagnetic indices: a comparison between the local K and global Kp indices,” *Earth, Planets and Space*, vol. 66, no. 1, p. 95, 2014, ISSN: 1880-5981. DOI: 10.1186/1880-5981-66-95.
- [4] J. P. Beukes, S. Lotz, and M. H. Davel, “Pairwise networks for feature ranking of a geomagnetic storm model,” *South African Computer Journal*, vol. 32, no. 2, pp. 35–55, 2020, ISSN: 23137835. DOI: 10.18489/SACJ.V32I2.860.
- [5] 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*, 2019, pp. 133–144.
- [6] I. Goodfellow, Y. Bengio, and A. Courville, *Deep Learning*, T. Dietterich, C. Bishop, D. Heckerman, M. Jordan, and M. Kearns, Eds. Cambridge: The MIT Press, 2016, vol. 53.
- [7] S. Lundberg and S.-I. Lee, “A unified approach to interpreting model predictions,” in *Advances in Neural Information Processing Systems*, vol. 30, 2017, pp. 4766–4775.
- [8] H. Chen, S. Lundberg, and S.-I. Lee, “Explaining Models by Propagating Shapley Values,” *Arxiv*, Nov. 2019. [Online]. Available: <https://arxiv.org/pdf/1911.11888.pdf>.
- [9] W. Samek, G. Montavon, S. Lapuschkin, C. J. Anders, and K. R. Müller, “Explaining Deep Neural Networks and Beyond: A Review of Methods and Applications,” *Proceedings*

-
- of the *IEEE*, vol. 109, no. 3, pp. 247–278, 2021, ISSN: 15582256. DOI: 10.1109/JPROC.2021.3060483.
- [10] P. Wintoft, M. Wik, and A. Viljanen, “Solar wind driven empirical forecast models of the time derivative of the ground magnetic field,” *Journal of Space Weather and Space Climate*, vol. 5, A7–P9, 2015, ISSN: 21157251. DOI: 10.1051/swsc/2015008.
 - [11] M. Moldwin, *An Introduction to Space Weather*. 2008, ISBN: 978-0-521-71112-8.
 - [12] J. G. Kappenman, “An overview of the impulsive geomagnetic field disturbances and power grid impacts associated with the violent sun-earth connection events of 29–31 october 2003 and a comparative evaluation with other contemporary storms,” *Space Weather*, vol. 3, no. 8, 2005. DOI: <https://doi.org/10.1029/2004SW000128>.
 - [13] W. D. Gonzalez *et al.*, “What is a geomagnetic storm?” *Journal of Geophysical Research*, vol. 99, pp. 5771–5792, 1994.
 - [14] L. Holappa and K. Mursula, “Explicit imf by dependence in high-latitude geomagnetic activity,” *Journal of Geophysical Research: Space Physics*, vol. 123, no. 6, pp. 4728–4740, 2018. DOI: <https://doi.org/10.1029/2018JA025517>.
 - [15] E. Echer, W. D. Gonzalez, and M. V. Alves, “On the geomagnetic effects of solar wind interplanetary magnetic structures,” *Space Weather*, vol. 4, 2006. DOI: <https://doi.org/10.1029/2005SW000200>.
 - [16] S. E. Milan *et al.*, “Timescales of dayside and nightside field-aligned current response to changes in solar wind-magnetosphere coupling,” *Journal of Geophysical Research: Space Physics*, vol. 123, no. 9, pp. 7307–7319, 2018. DOI: <https://doi.org/10.1029/2018JA025645>.
 - [17] A. W. P. Thomson *et al.*, “Present day challenges in understanding the geomagnetic hazard to national power grids,” *Advances in Space Research*, vol. 45, no. 9, pp. 1182–1190, 2010, ISSN: 0273-1177. DOI: <https://doi.org/10.1016/j.asr.2009.11.023>.
 - [18] J. B. Habarulema *et al.*, “Ionospheric response at conjugate locations during the 7–8 september 2017 geomagnetic storm over the europe-african longitude sector,” *Journal of Geophysical Research: Space Physics*, vol. 125, 2020.
 - [19] P. Wintoft, “Study of the solar wind coupling to the time difference horizontal geomagnetic field,” *Annales Geophysicae*, vol. 23, pp. 1949–1957, 2005.
-

-
- [20] A. Boudouridis, H. K. Connor, D. Lummerzheim, A. J. Ridley, and E. Zesta, “Changes in the magnetic field topology and the dayside/nightside reconnection rates in response to a solar wind dynamic pressure front: A case study,” *Journal of Geophysical Research: Space Physics*, vol. 126, 2021.
- [21] M. Revallo, F. Valach, P. Hejda, and J. Bochníček, “Modeling of cme and cir driven geomagnetic storms by means of artificial neural networks,” *Contributions to Geophysics and Geodesy*, vol. 45, no. 1, pp. 53–65, Mar. 2015, ISSN: 13380540. DOI: 10.1515/CONGEO-2015-0013.
- [22] 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, Nov. 2018, ISSN: 1542-7390. DOI: 10.1029/2018SW001898.
- [23] S. Lotz, J. P. Beukes, and M. H. Davel, “Input parameter ranking for neural networks in a space weather regression problem,” in *Fundamentals of Artificial Intelligence Research*, CEUR workshop proceedings, 2019, ch. 133-144.
- [24] W. S. McCulloch and W. H. Pitts, “A logical calculus of the ideas immanent in nervous activity,” *Bulletin of Mathematical Biophysics*, vol. 5, pp. 115–133, 1943.
- [25] A. Zhang, Z. C. Lipton, M. Li, and A. J. Smola, “Dive into deep learning,” *arXiv preprint arXiv:2106.11342*, 2021.
- [26] D. P. Kingma and J. Ba, “Adam: A method for stochastic optimization,” *International Conference on Learning Representations*, 2014.
- [27] A. L. Maas, “Rectifier nonlinearities improve neural network acoustic models,” in *International Conference on Machine Learning*, vol. 30, 2013, p. 3.
- [28] J. L. Elman, “Finding structure in time,” *Cognitive Science*, vol. 14, pp. 179–211, 1990.
- [29] A. F. de Arajou, W. Norris, and J. Sim, “Computing receptive fields of convolutional neural networks,” *Distill*, 2019.
- [30] S. Bai, J. Z. Kolter, and V. Koltun, “An empirical evaluation of generic convolutional and recurrent networks for sequence modeling,” *ArXiv*, vol. abs/1803.01271, 2018.
- [31] A. Barredo Arrieta *et al.*, “Explainable Explainable Artificial Intelligence (XAI): Concepts, taxonomies, opportunities and challenges toward responsible AI,” *Information Fusion*, vol. 58, pp. 82–115, 2020, ISSN: 15662535. DOI: 10.1016/j.inffus.2019.12.012.
-

-
- [32] L. Arras, G. Montavon, K.-R. Müller, and W. Samek, “Explaining Recurrent Neural Network Predictions in Sentiment Analysis,” pp. 159–168, Jan. 2018. DOI: 10.18653/v1/w17-5221.
- [33] A. Brasoveanu, M. Moodie, and R. Agrawal, “Textual evidence for the perfunctoriness of independent medical reviews,” *CEUR Workshop Proceedings*, vol. 2657, pp. 1–9, 2020, ISSN: 16130073.
- [34] R. R. Hoffman, S. T. Mueller, G. Klein, and J. Litman, “Metrics for Explainable AI: Challenges and Prospects,” Tech. Rep., 2018. eprint: 1812.04608.
- [35] M. Robnik-Šikonja and M. Bohanec, “Perturbation-Based Explanations of Prediction Models,” pp. 159–175, 2018. DOI: 10.1007/978-3-319-90403-0_9.
- [36] Y. Yang, V. Tresp, M. Wunderle, and P. A. Fasching, “Explaining therapy predictions with layer-wise relevance propagation in neural networks,” *Proceedings - 2018 IEEE International Conference on Healthcare Informatics, ICHI 2018*, pp. 152–162, Jul. 2018. DOI: 10.1109/ICHI.2018.00025.
- [37] F. Kratzert, M. Herrnegger, D. Klotz, S. Hochreiter, and G. Klambauer, “NeuralHydrology – Interpreting LSTMs in Hydrology,” *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, vol. 11700 LNCS, pp. 347–362, 2019, ISSN: 16113349. DOI: 10.1007/978-3-030-28954-6_19. arXiv: 1903.07903.
- [38] L. S. Shapley, “a Value for n-Person Games,” *Classics in Game Theory*, vol. 2, pp. 69–79, 2020. DOI: 10.2307/j.ctv173f1fh.12.
- [39] C. Bishop, *Pattern recognition and machine learning*. Springer, 2006, ISBN: 978-0-387-31073-2.
- [40] “Explainable artificial intelligence for the remaining useful life prognosis of the turbofan engines,” *Proceedings of the 3rd IEEE International Conference on Knowledge Innovation and Invention 2020, ICKII 2020*, pp. 144–147, Aug. 2020. DOI: 10.1109/ICKII50300.2020.9318912.
- [41] S. M. Lauritsen *et al.*, “Explainable artificial intelligence model to predict acute critical illness from electronic health records,” *Nature Communications*, vol. 11, no. 1, 2020, ISSN: 20411723. DOI: 10.1038/s41467-020-17431-x. arXiv: 1912.01266.
-

-
- [42] W. Kirchgassner, O. Wallscheid, and J. Bocker, “Estimating Electric Motor Temperatures with Deep Residual Machine Learning,” *IEEE Transactions on Power Electronics*, vol. 36, no. 7, pp. 7480–7488, Jul. 2021, ISSN: 19410107. DOI: 10.1109/TPEL.2020.3045596.
- [43] R. Roscher, B. Bohn, M. F. Duarte, and J. Garcke, “Explainable Machine Learning for Scientific Insights and Discoveries,” *IEEE Access*, vol. 8, pp. 42 200–42 216, 2020, ISSN: 21693536. DOI: 10.1109/ACCESS.2020.2976199. arXiv: 1905.08883.
- [44] G. Quellec, K. Charrière, Y. Boudi, B. Cochener, and M. Lamard, “Deep image mining for diabetic retinopathy screening,” *Medical Image Analysis*, vol. 39, pp. 178–193, Jul. 2017, ISSN: 13618423. DOI: 10.1016/j.media.2017.04.012. eprint: 1610.07086.
- [45] Z. Zhang *et al.*, “Pathologist-level interpretable whole-slide cancer diagnosis with deep learning,” *Nature Machine Intelligence*, vol. 1, no. 5, pp. 236–245, 2019. DOI: 10.1038/s42256-019-0052-1.
- [46] I. Ebert-Uphoff and K. Hilburn, “Evaluation, tuning, and interpretation of neural networks for working with images in meteorological applications,” *Bulletin of the American Meteorological Society*, vol. 101, no. 12, Dec. 2020, ISSN: 00030007. DOI: 10.1175/BAMS-D-20-0097.1.
- [47] F. Horst, S. Lapuschkin, W. Samek, K. R. Müller, and W. I. Schöllhorn, “Explaining the unique nature of individual gait patterns with deep learning,” *Scientific Reports*, vol. 9, no. 1, pp. 1–13, Feb. 2019, ISSN: 20452322. DOI: 10.1038/s41598-019-38748-8. eprint: 1808.04308.
- [48] K. Preuer, G. Klambauer, F. Rippmann, S. Hochreiter, and T. Unterthiner, “Interpretable Deep Learning in Drug Discovery,” *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, vol. 11700 LNCS, pp. 331–345, 2019, ISSN: 16113349. DOI: 10.1007/978-3-030-28954-6_18. arXiv: 1903.02788.
- [49] J. Hochuli, A. Helbling, T. Skaist, M. Ragoza, and D. R. Koes, “Visualizing convolutional neural network protein-ligand scoring,” *Journal of Molecular Graphics and Modelling*, vol. 84, pp. 96–108, Sep. 2018, ISSN: 18734243. DOI: 10.1016/j.jmgm.2018.06.005. eprint: 1803.02398.
- [50] K. T. Schütt, M. Gastegger, A. Tkatchenko, and K. R. Müller, “Quantum-Chemical Insights from Interpretable Atomistic Neural Networks,” *Lecture Notes in Computer Science*
-

-
- (including subseries *Lecture Notes in Artificial Intelligence* and *Lecture Notes in Bioinformatics*), vol. 11700 LNCS, pp. 311–330, 2019, ISSN: 16113349. DOI: 10.1007/978-3-030-28954-6_17. arXiv: 1806.10349.
- [51] A. Shrikumar, P. Greenside, and A. Kundaje, “Learning Important Features Through Propagating Activation Differences,” 2019. arXiv: 1704.02685v2.
 - [52] H. Chen, S. M. Lundberg, and S.-I. Lee, “Explaining a Series of Models by Propagating Shapley Values,” *Nature communications*, vol. 13, 2022. DOI: <https://doi.org/10.1038/s41467-022-31384-3>.
 - [53] M. Lindauer *et al.*, “SMAC3: A Versatile Bayesian Optimization Package for Hyperparameter Optimization,” vol. 23, pp. 1–9, 2022.
 - [54] J. Wanliss and K. M. Showalter, “High-resolution global storm index: Dst versus sym-h,” *Journal of Geophysical Research*, vol. 111, 2006.
 - [55] J. E. Borovsky and M. H. Denton, “Differences between cme-driven storms and cir-driven storms,” *Journal of Geophysical Research: Space Physics*, vol. 111, no. A7, pp. 7–08, Jul. 2006, ISSN: 2156-2202. DOI: 10.1029/2005JA011447.
 - [56] K. Choraghe, A. Raghav, D. Chakrabarty, S. Kasthurirangan, and N. Bijewar, “Properties of the recovery phase of extreme storms,” *Journal of Geophysical Research: Space Physics*, vol. 126, no. 9, 2021. DOI: <https://doi.org/10.1029/2020JA028685>.
 - [57] U. of Helsinki, *Database of interplanetary shocks - ipshocks*. [Online]. Available: <http://ipshocks.fi/database> (visited on 04/11/2022).
 - [58] TWINS, *TWINS Geomagnetic Storm Catalog*. [Online]. Available: <https://auburn.edu/cosam/twins/> (visited on 04/11/2022).
 - [59] L. Jian, *Stream Interaction Regions (SIRs) from Wind and ACE Data during 1995-2009*, 2021. (visited on 03/11/2022).
 - [60] W. D. Cramer, N. E. Turner, M.-C. Fok, and N. Y. Buzulukova, “Effects of different geomagnetic storm drivers on the ring current: CRCM results,” *Journal of Geophysical Research: Space Physics*, vol. 118, pp. 1062–1073, 2013. DOI: 10.1002/jgra.50138.
 - [61] I. G. Richardson and H. V. Cane, “Near-Earth Interplanetary Coronal Mass Ejections During Solar Cycle 23 (1996–2009): Catalog and Summary of Properties,” *Solar Physics* 2010 264:1, vol. 264, no. 1, pp. 189–237, May 2010, ISSN: 1573-093X. DOI: 10.1007/S11207-010-9568-6.
-

Appendix A

Supplemental content

We provide supplementary results related to an additional synthetic data set baseline experiment referred to in Chapter 3, data set preparation and additional analysis results from Chapter 4, modelling results of and predictions from the TSMLPs in Chapter 5 and the attributions generated in Chapter 6

A.1 Appendix: Chapter 3

An MLP model is trained on the synthetic data set defined in Section 3.3, with the same train/test/eval split, according to the same SMAC3 process with the same configuration space, but with bias nodes excluded, and with the addition of a third batch size option (128). The final model trained to an MSE loss of 0.2145, and the hyperparameter configuration included a layout of 3 layers of 100 nodes, batch size of 128, learning rate of 0.0016, and a weight decay of 0.00001. The scenario plot for this model configuration selection is shown in Figure A.1 and the output of the model in comparison to the target values for the training and evaluation sets is available in Figure A.2.

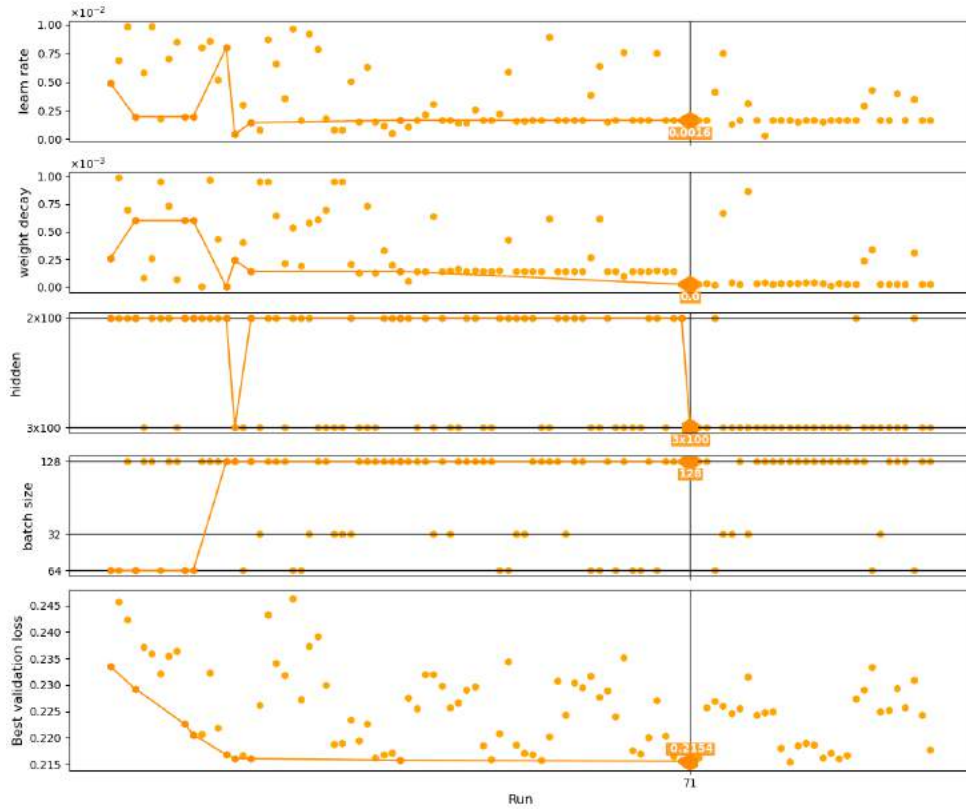


Figure A.1: Scenario plot for the selection of MLP hyperparameters on the synthetic data set.

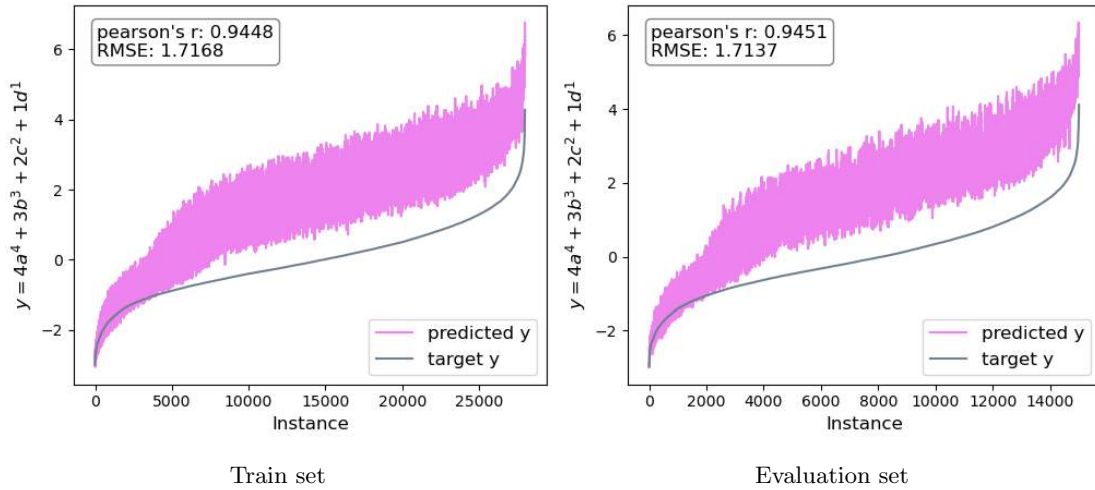


Figure A.2: Output values ordered by target output value from smallest to largest vs. target output values for train and evaluation sets.

Feature attributions are generated for this model in the same way as was described in Section 3.5, and the results are shown in Figures A.3 and A.4.

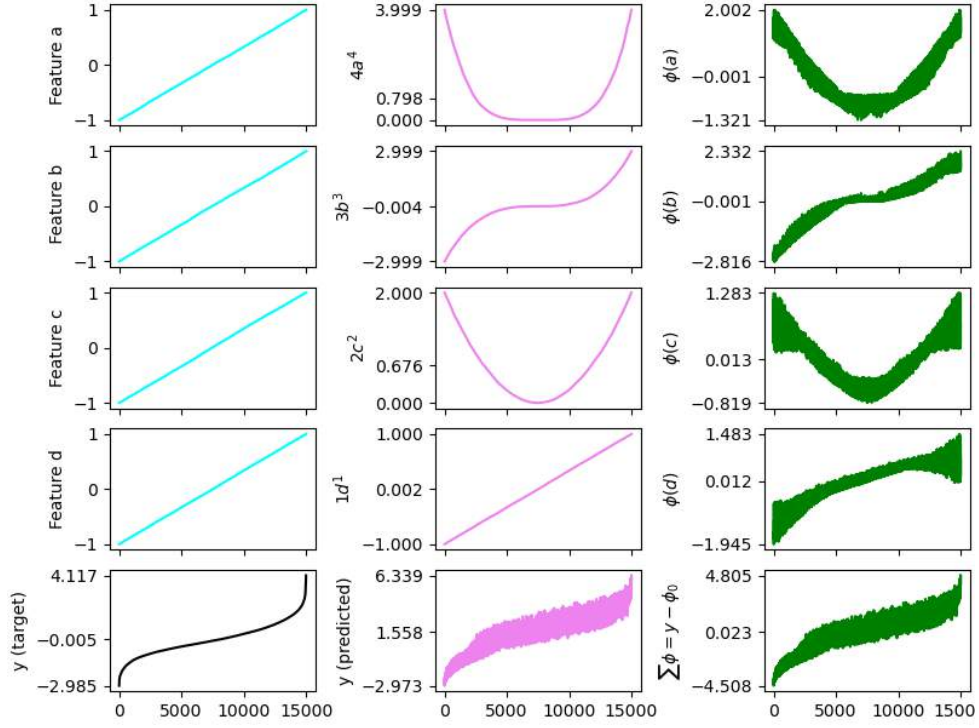


Figure A.3: Compilation of ordered features and their mappings vs their respective DeepSHAP attributions.

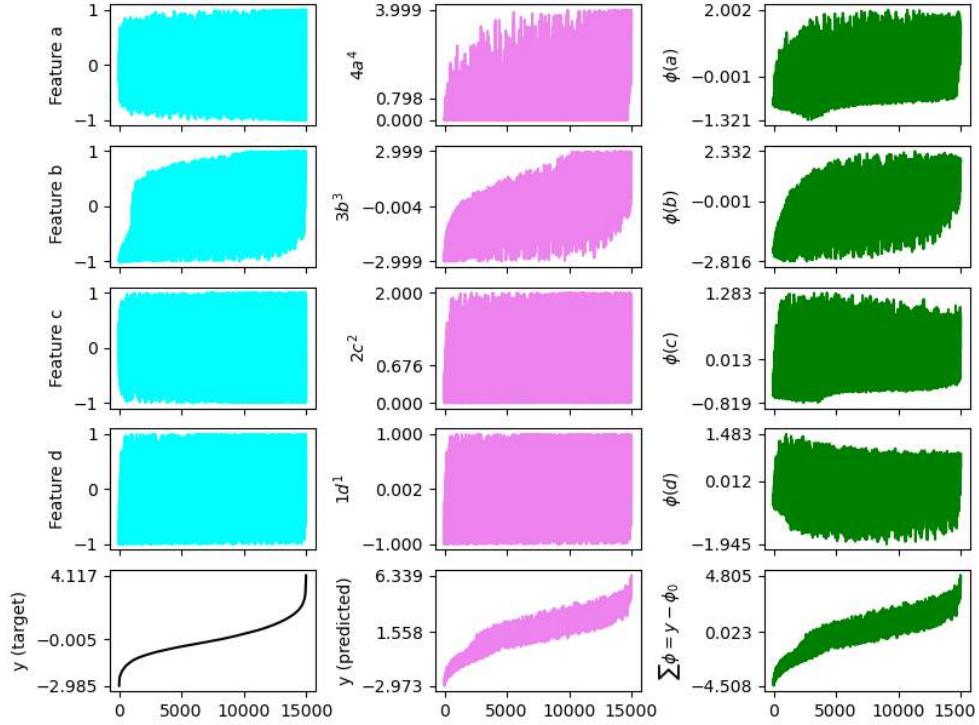


Figure A.4: Compilation of input feature values, transformed input feature values, and each input feature's respective DeepSHAP as sorted from smallest to largest according to target output value.

A.2 Appendix: Chapter 4

A.2.1 Dates of storms referenced in text

Table A.1: Start/end times of storms by Storm ID

Storm ID	Start datetime	End datetime	CME/CIR
1996_severe_000	1996-10-22 09:30:00	1996-10-26 00:32:00	CME
1997_moderate_001	1997-02-26 11:57:00	1997-03-02 10:52:00	CIR
1998_mild_001	1998-03-23 23:53:00	1998-03-25 21:54:00	Combined
1998_severe_006	1998-09-24 12:22:00	1998-09-28 00:11:00	CME
2001_severe_005	2001-08-17 00:08:00	2001-08-19 04:35:00	CME
2003_severe_000	2003-05-29 04:35:00	2003-05-31 17:37:00	CME
2003_severe_007	2003-11-02 22:09:00	2003-11-03 10:12:00	CIR
2004_moderate_001	2004-07-16 09:59:00	2004-07-18 05:27:00	CIR
2005_mild_004	2005-04-29 06:24:00	2005-04-30 05:39:00	CIR
2005_severe_002	2005-01-21 05:23:00	2005-01-24 03:19:00	CME
2014_mild_007	2014-10-26 11:14:00	2014-10-28 18:47:00	CME

A.2.2 Interpolation procedure

Once each column's placeholder values for each data set (OMNI and INTERMAGNET (HER and ABK)) are replaced with NaN values, each data set is then interpolated with a second-order polynomial. The reason for choosing polynomial interpolation is that linear interpolation assumes that the gaps in the data are linearly spaced. A polynomial interpolation uses the index values as a guide in the pandas implementation. While it may cause some interpolation artefacts, it is far less computationally demanding than a spline interpolation.

The size of the gaps interpolated over are limited by two factors: the diminishing returns of usable points gained when increasing gap size, and the potential for 'hiding' true and salient patterns in the data by interpolating over too large of a gap. To determine optimal gap limit size, data of the OMNI data set from every second year in the period of 1996 to 2008 were interpolated using various gap limits, and the ratio of usable points gained for each gap limit were recorded.

These results are listed in Table A.2, and are visualised in Figure A.5. Since the point of diminishing returns for interpolation gap size remains at 10 minutes, the solar wind parameters are interpolated with a 2nd order polynomial up to a limit of 10 minutes at a time.

Table A.2: Ratio of points gained per interpolation limit in minutes for the years 1996, 1998, 2000, 2002, 2004, 2006, 2008

	1996	1998	2000	2002	2004	2006	2008
0	0	0	0	0	0	0	0
1	$4.64x10^{-4}$	$5.08x10^{-4}$	$1.01x10^{-3}$	$4.46x10^{-4}$	$2.19x10^{-4}$	$2.59x10^{-4}$	$2.75x10^{-4}$
2	$5.06x10^{-4}$	$5.53x10^{-4}$	$1.27x10^{-3}$	$6.27x10^{-4}$	$3.04x10^{-4}$	$3.74x10^{-4}$	$3.80x10^{-4}$
3	$5.20x10^{-4}$	$5.60x10^{-4}$	$1.38x10^{-3}$	$7.19x10^{-4}$	$3.50x10^{-4}$	$4.11x10^{-4}$	$4.35x10^{-4}$
5	$5.61x10^{-4}$	$5.69x10^{-4}$	$1.45x10^{-3}$	$7.84x10^{-4}$	$3.70x10^{-4}$	$4.37x10^{-4}$	$4.77x10^{-4}$
10	$5.88x10^{-4}$	$5.71x10^{-4}$	$1.48x10^{-3}$	$7.98x10^{-4}$	$3.86x10^{-4}$	$4.79x10^{-4}$	$5.12x10^{-4}$
15	$5.88x10^{-4}$	$5.71x10^{-4}$	$1.49x10^{-3}$	$6.27x10^{-4}$	$3.86x10^{-4}$	$4.93x10^{-4}$	$5.24x10^{-4}$
20	$5.88x10^{-4}$	$5.71x10^{-4}$	$1.49x10^{-3}$	$6.27x10^{-4}$	$3.86x10^{-4}$	$5.05x10^{-4}$	$5.31x10^{-4}$
25	$5.88x10^{-4}$	$5.71x10^{-4}$	$1.49x10^{-3}$	$6.27x10^{-4}$	$3.86x10^{-4}$	$5.05x10^{-4}$	$5.31x10^{-4}$
30	$5.88x10^{-4}$	$5.71x10^{-4}$	$1.49x10^{-3}$	$6.27x10^{-4}$	$3.86x10^{-4}$	$5.05x10^{-4}$	$5.31x10^{-4}$

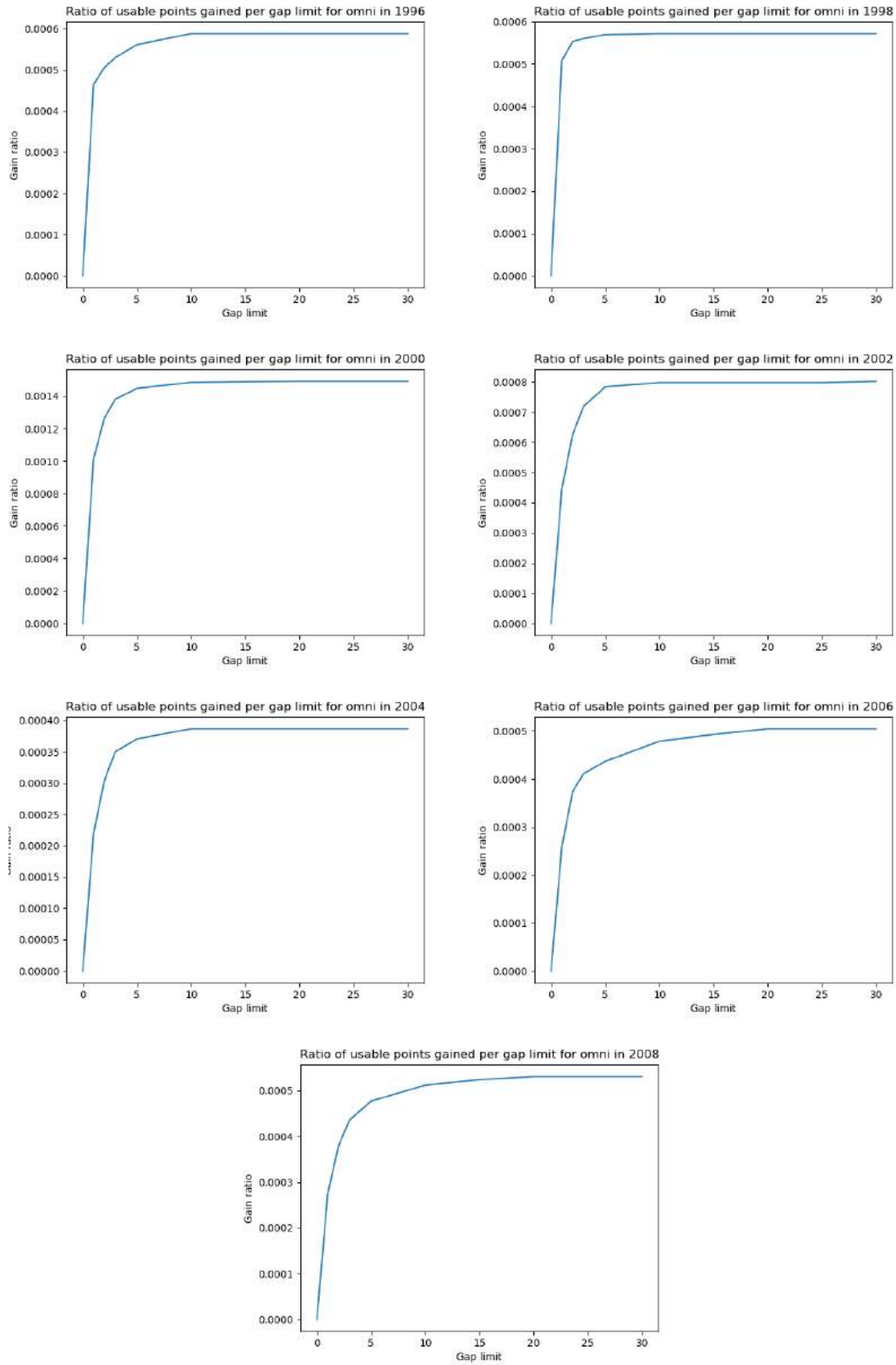
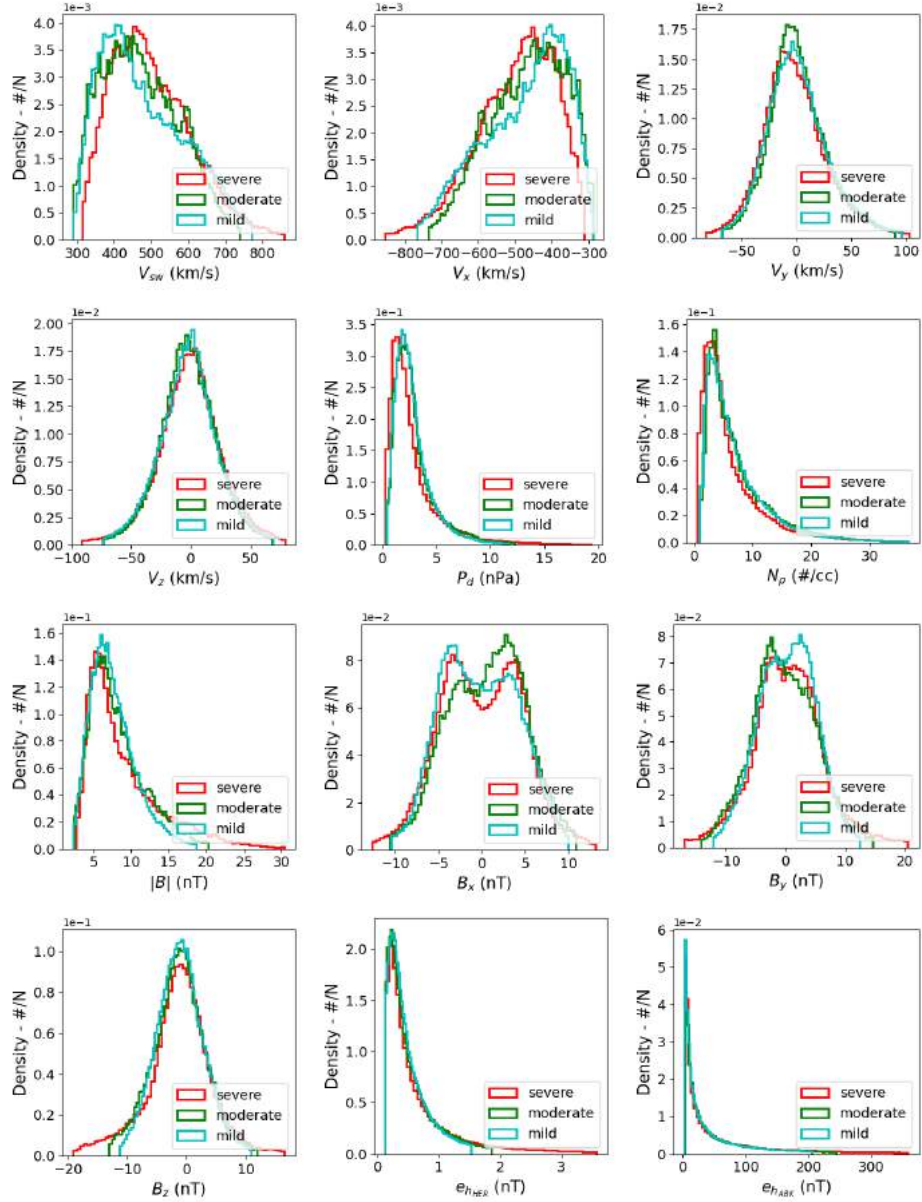


Figure A.5: The ratio of points gained vs. interpolation limit for the years 1996, 1998, 2000, 2002, 2004, 2006 and 2008

A.2.3 Intensity-wise empirical histograms

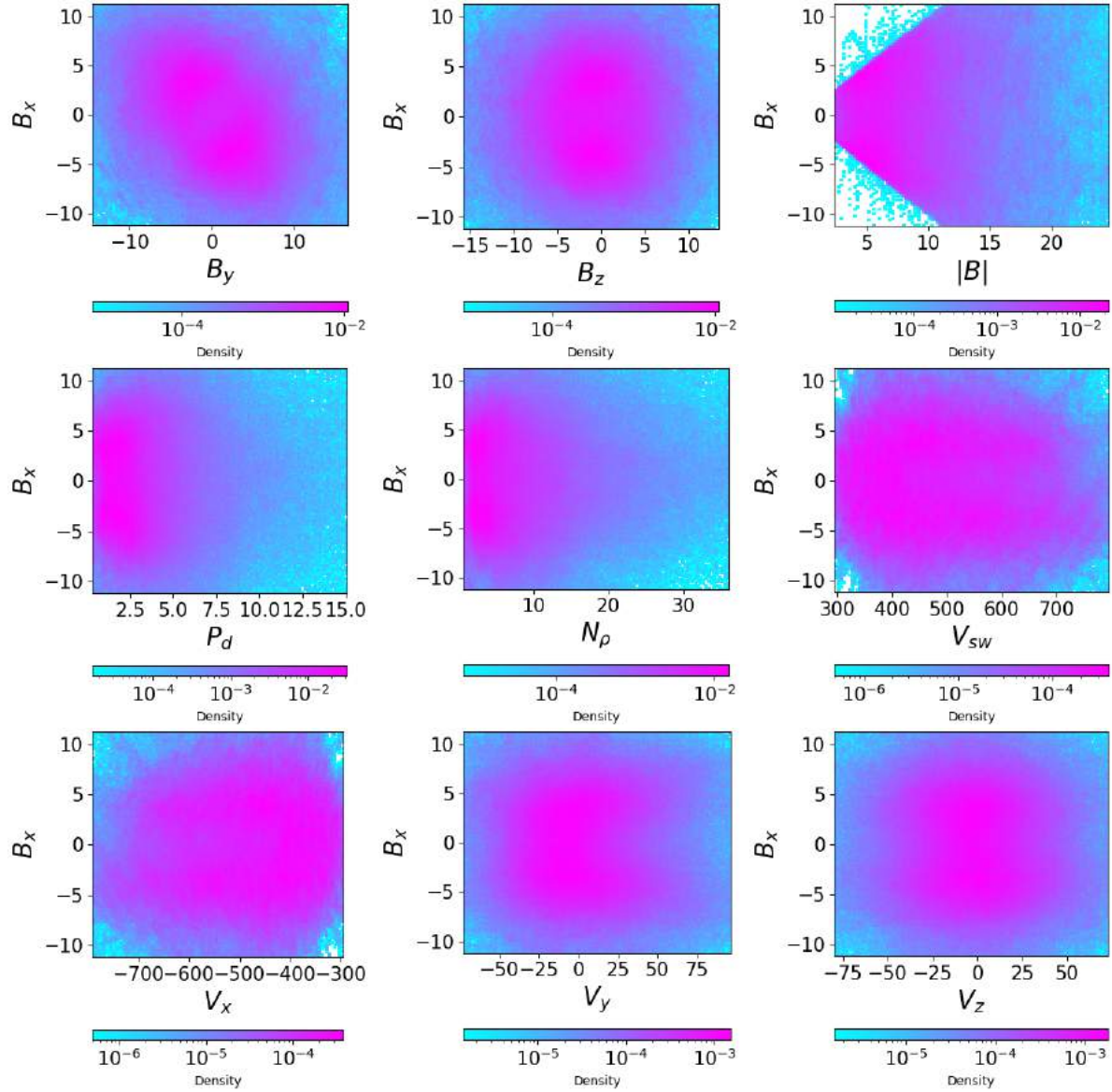
Histograms are drawn to compare the empirical distributions for each input feature listed in Section 4.2.1 between storm intensities. We show the distribution of each parameter for identified storms (the HAS set itself) from the years 1996-2019 of severe, moderate, and mild intensity. Outliers and bin counts are handled in the same way for each parameter as it was done in Section 4.3.1.



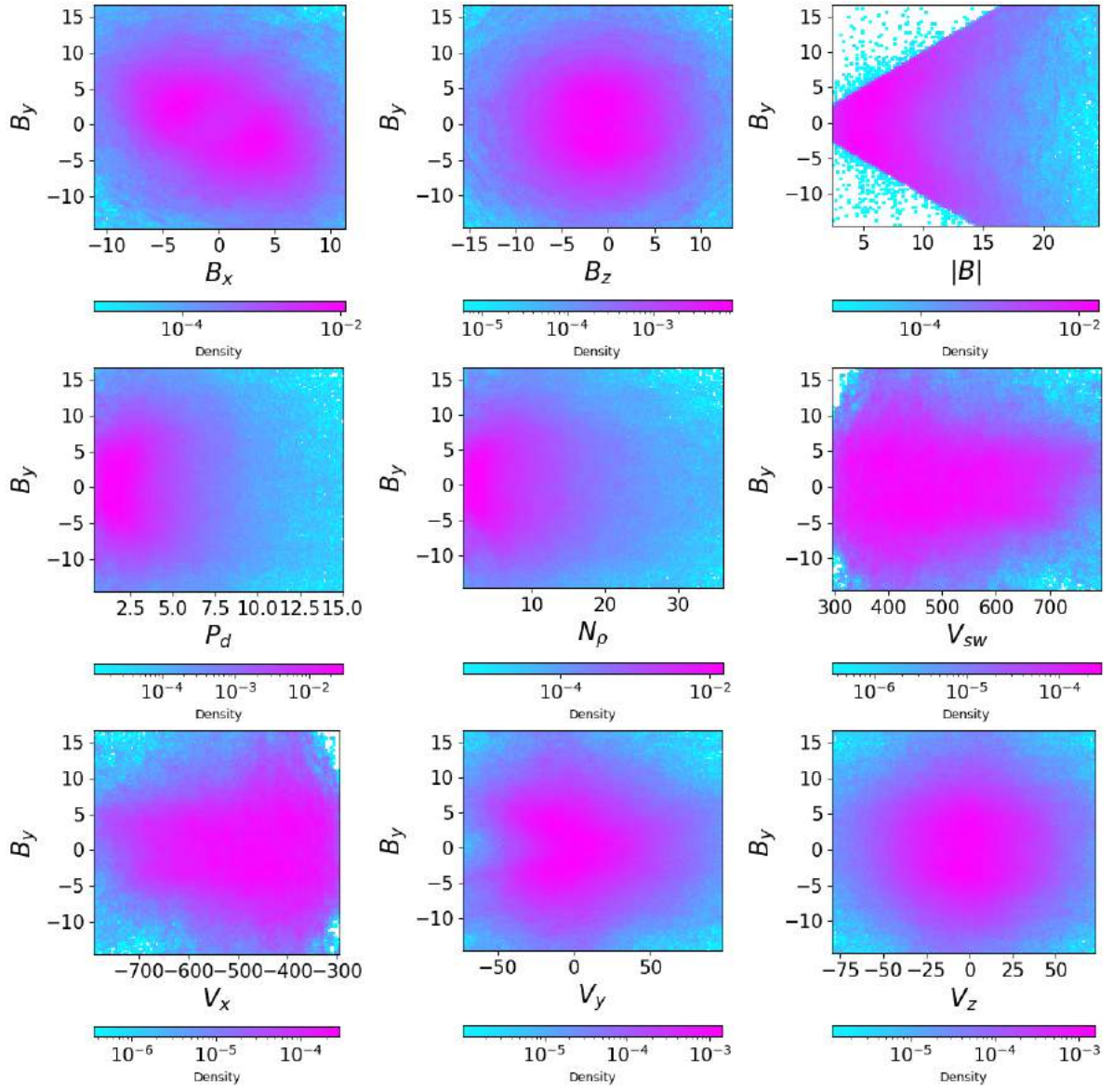
Empirical histograms of all input parameters and e_h , for each storm intensity, from 1996 to 2019. The relative frequency of samples is shown on the y-axis, and the value range is shown on the x-axis.

A.2.4 Pairwise parameter 2-D histograms

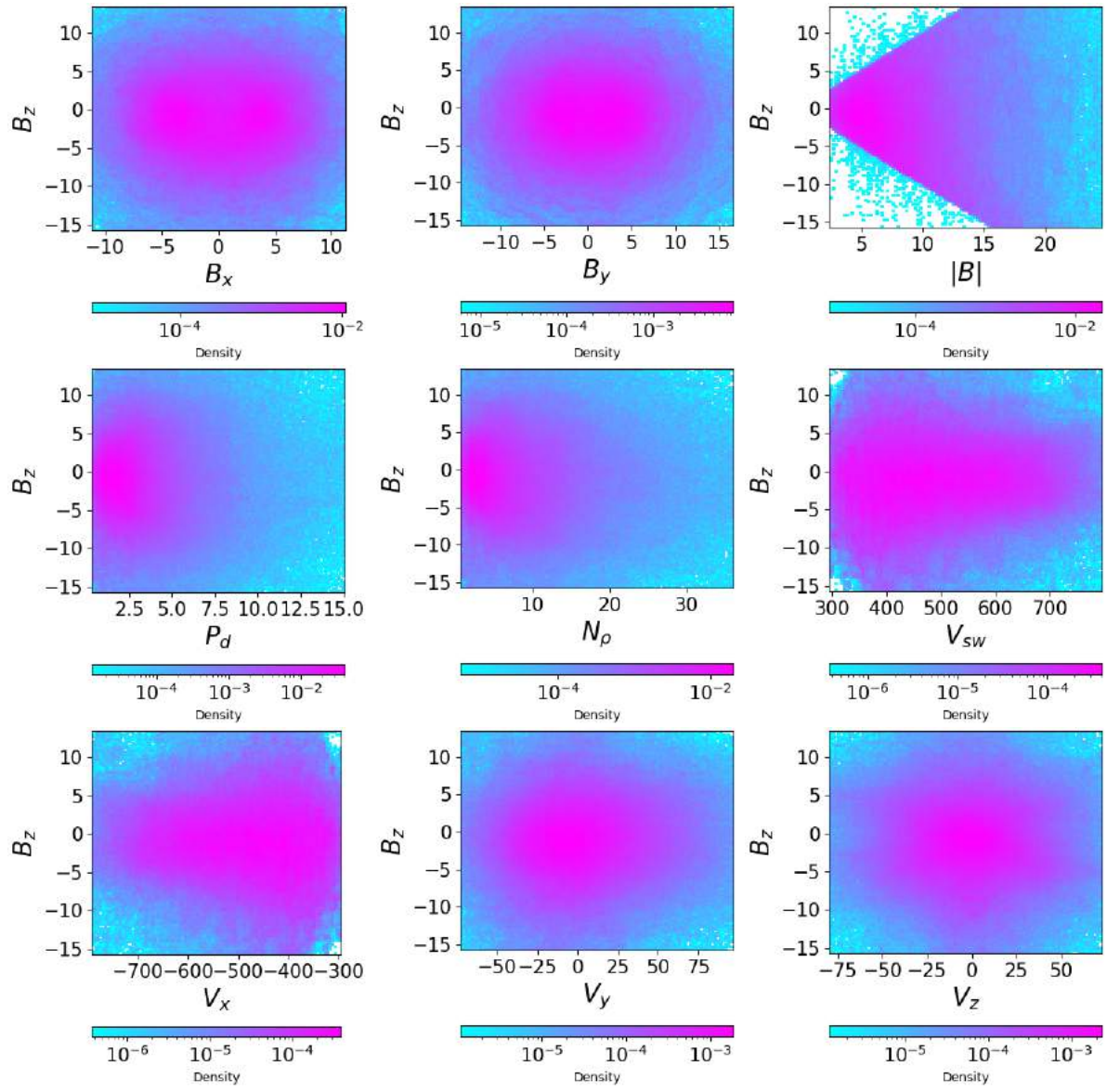
Additional pairwise 2-dimensional histograms for each input feature pair generated in the same way as for each input feature vs. output target in Section 4.3.3 are displayed here.



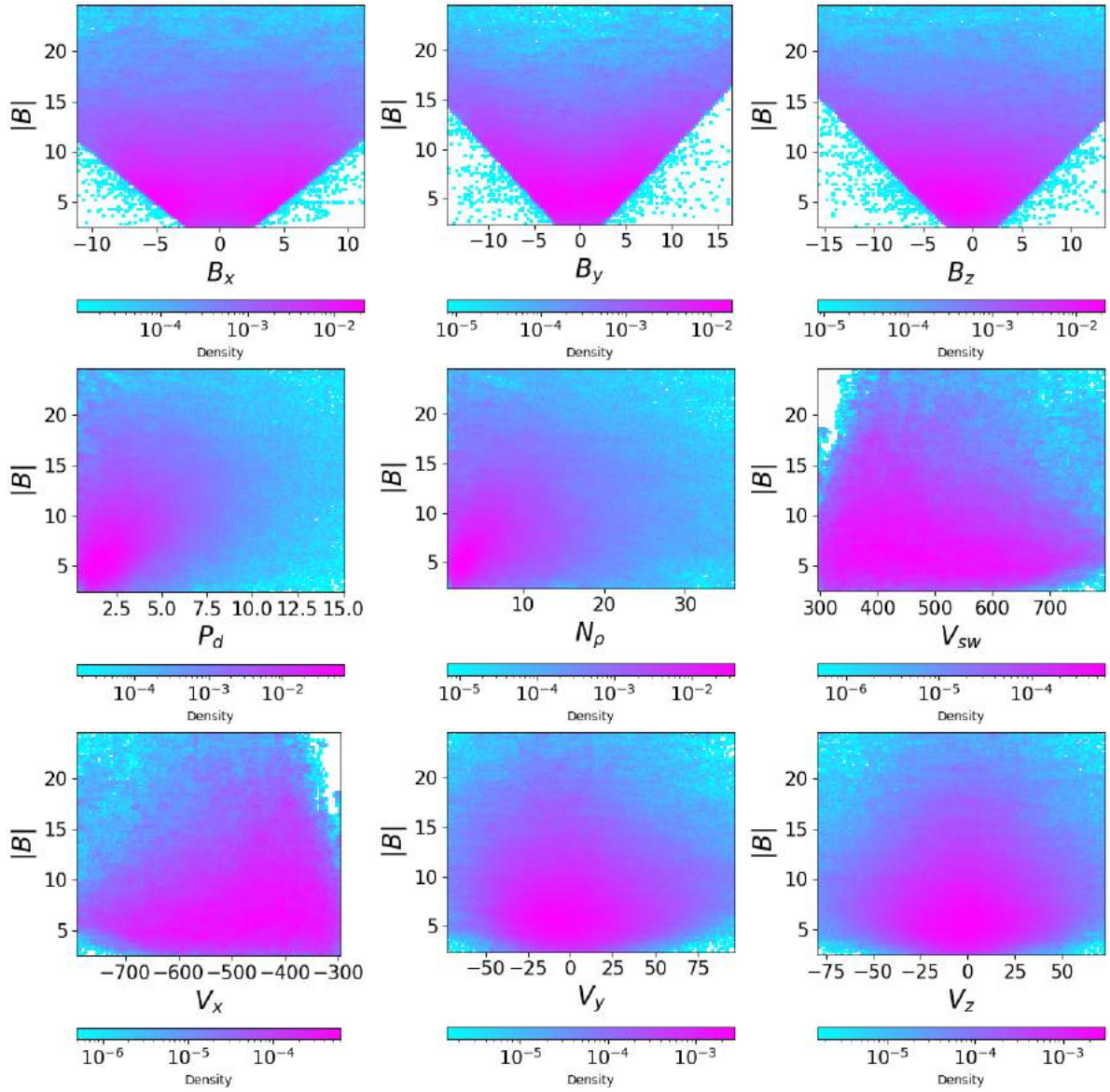
2-dimensional histograms of input features vs. B_x .



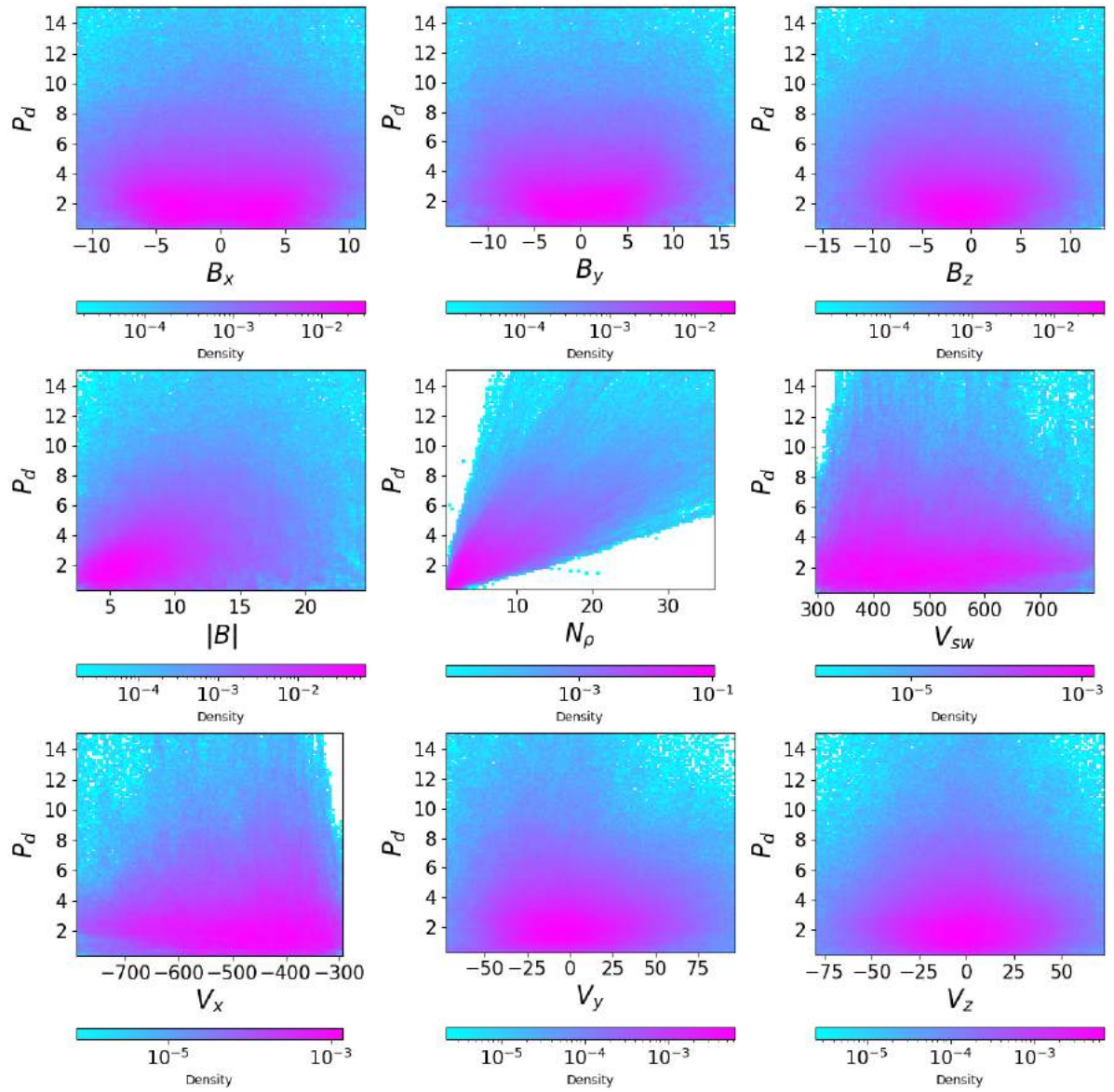
2-dimensional histograms of input features vs. B_y .



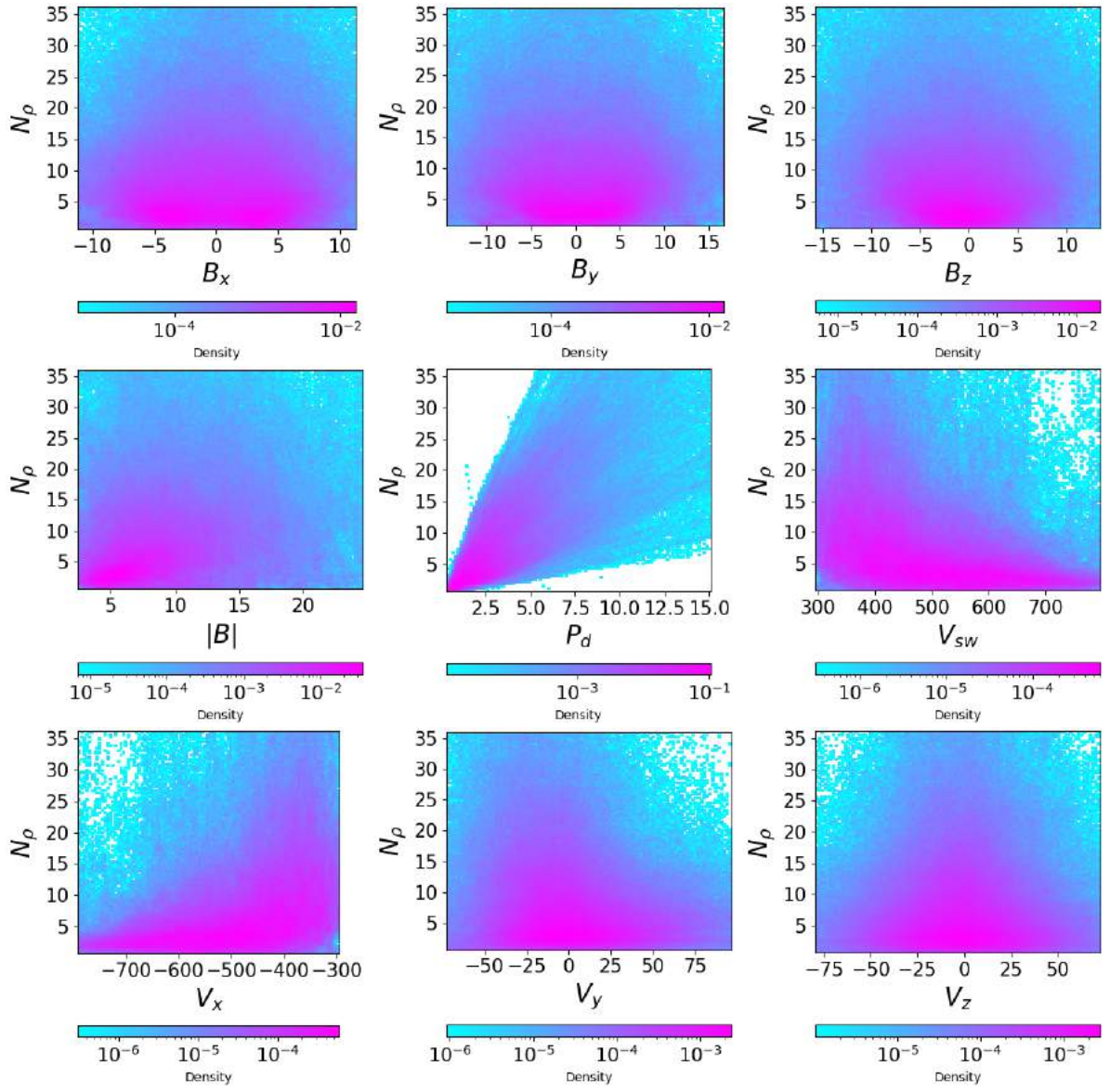
2-dimensional histograms of input features vs. B_z .



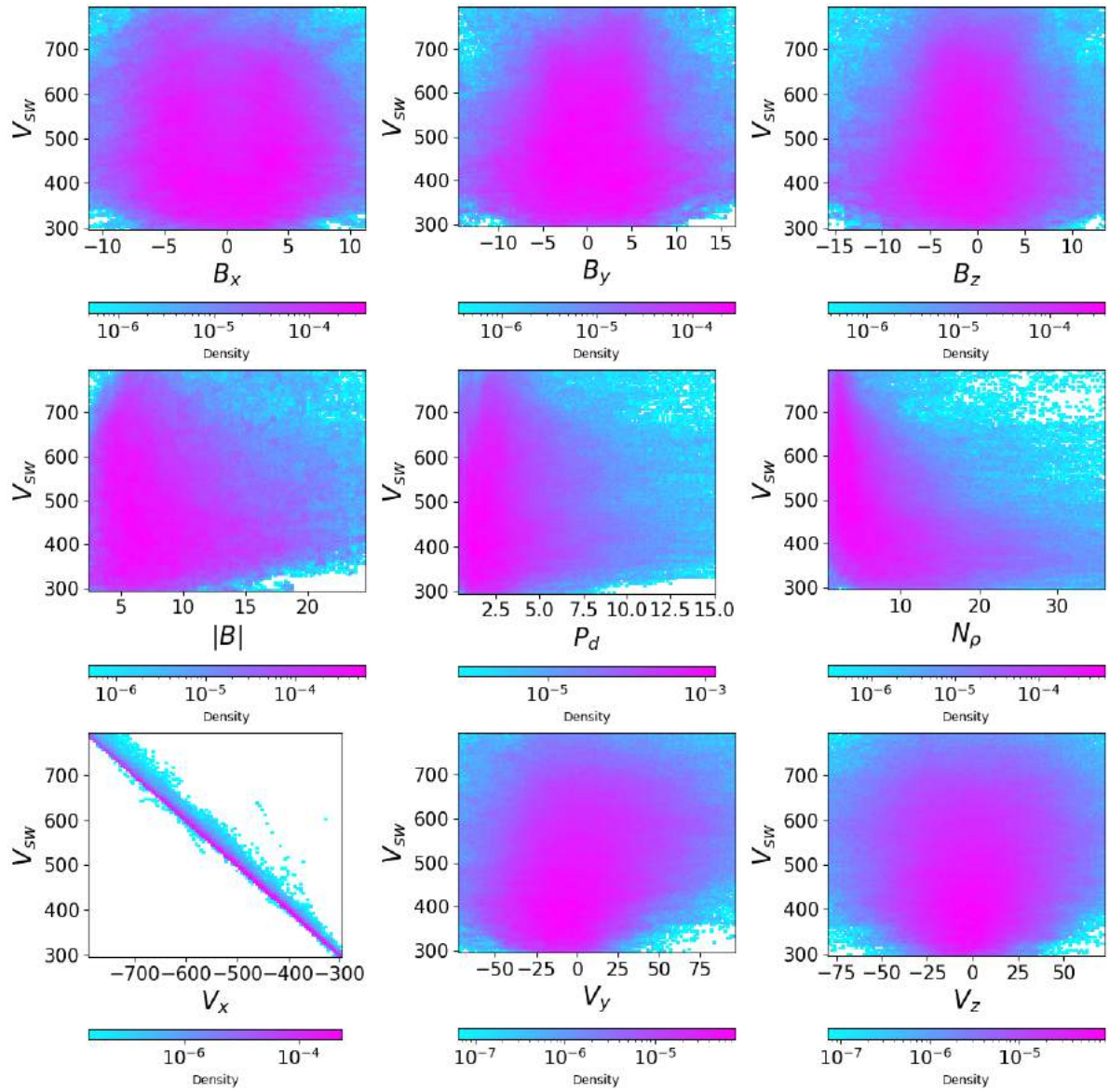
2-dimensional histograms of input features vs. $|B|$.



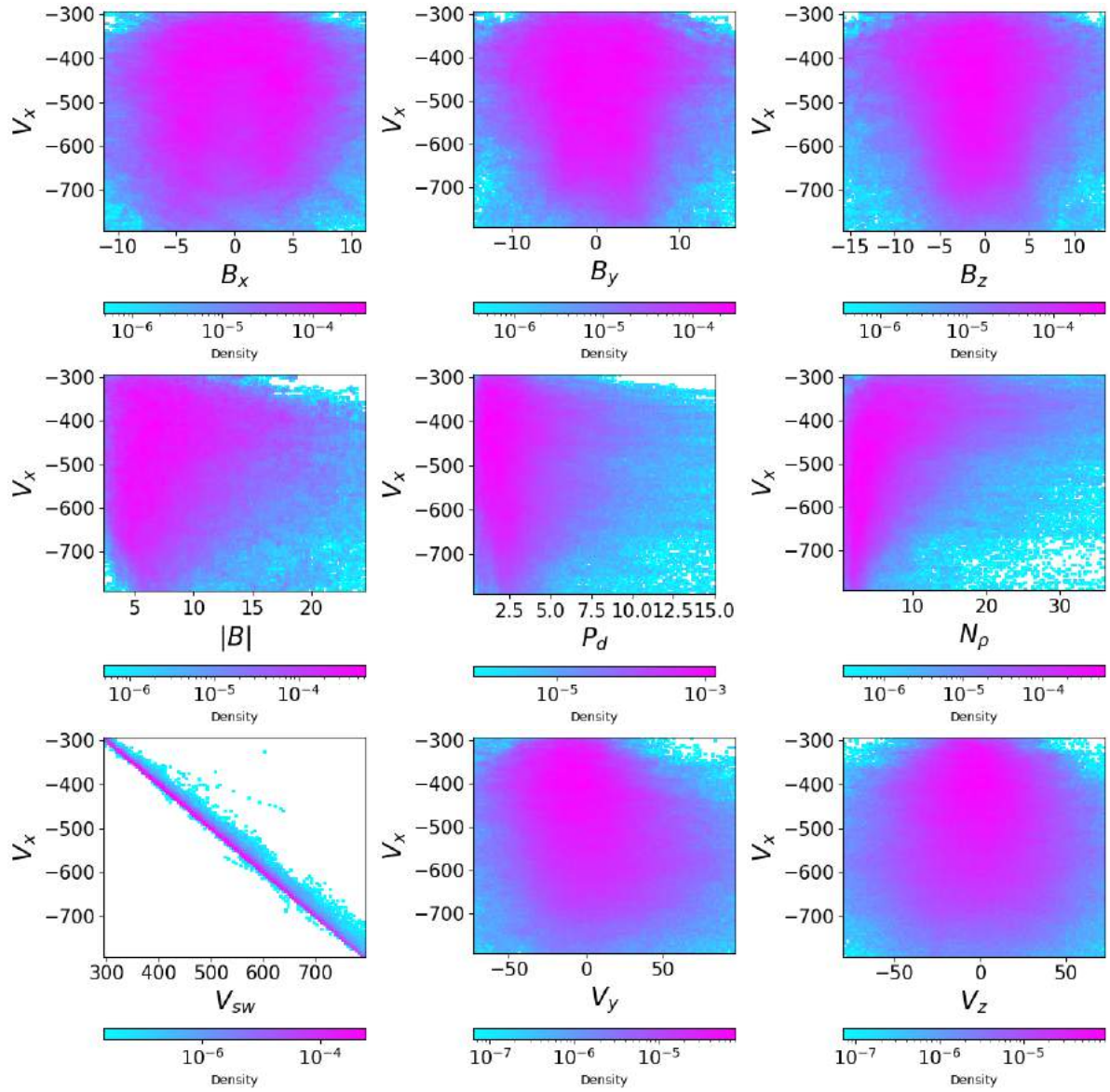
2-dimensional histograms of input features vs. P_d .



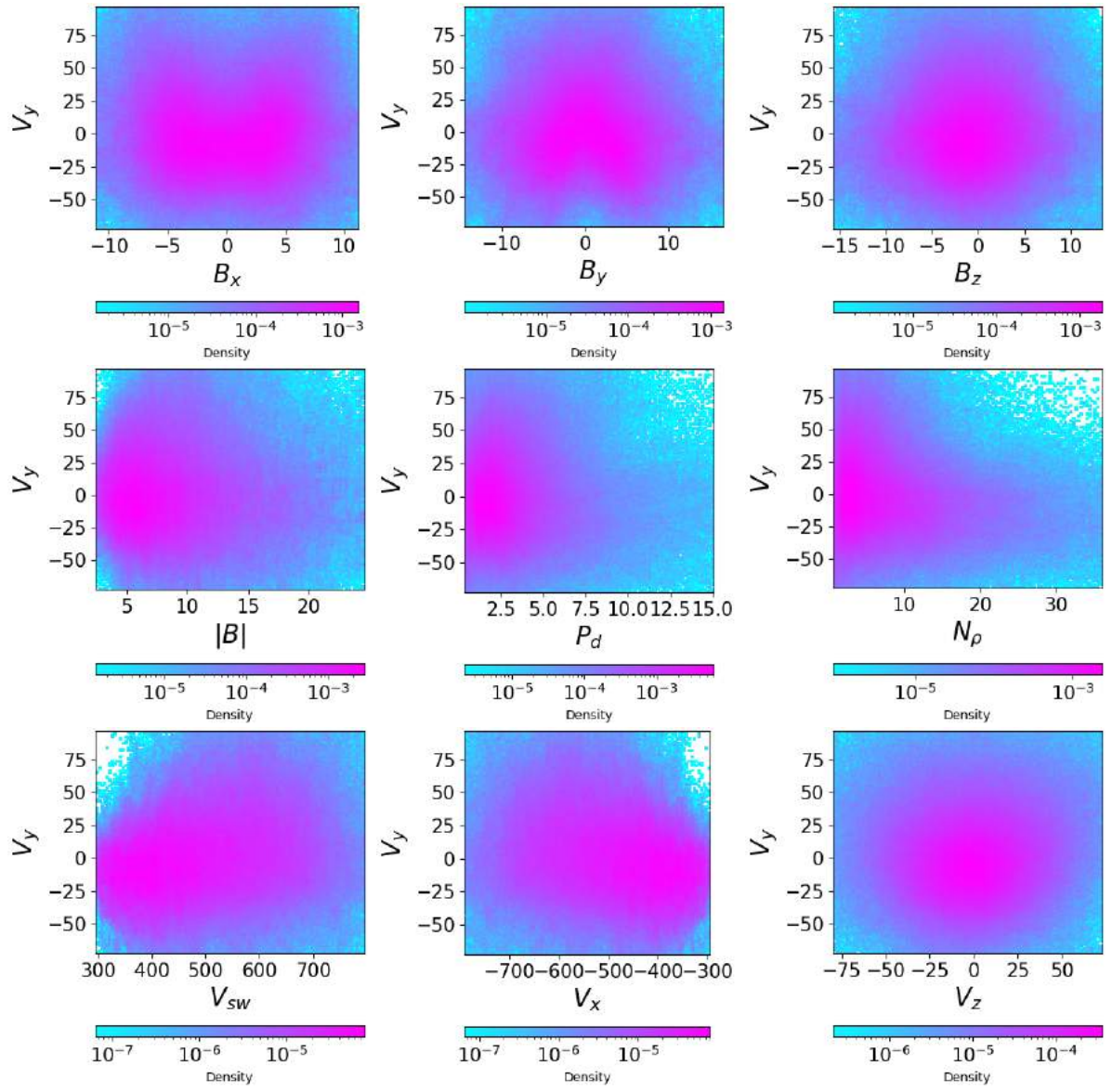
2-dimensional histograms of input features vs. N_ρ .



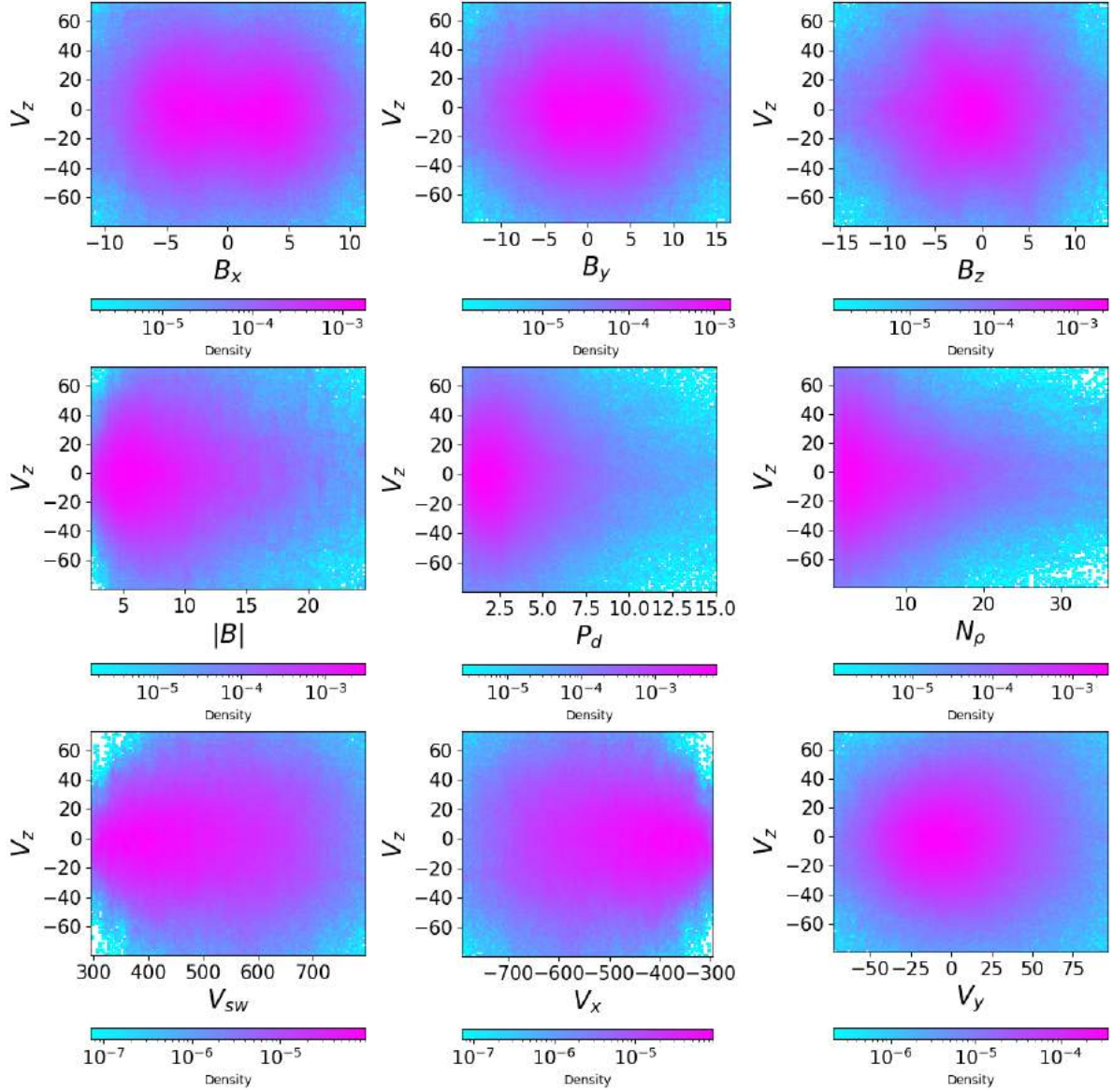
2-dimensional histograms of input features vs. V_{sw} .



2-dimensional histograms of input features vs. V_x .



2-dimensional histograms of input features vs. V_y .



2-dimensional histograms of input features vs. V_z .

A.3 Appendix: Chapter 5

A.3.1 SMAC characterisation

There are several ways to set up the initial configuration space for the SMAC3 algorithm, and there are variations on how this algorithm is able to optimise a model configuration for an optimal validation loss. Most of this variation comes in due to how the SMAC3 algorithm is integrated into our mustnet toolkit; our mustnet toolkit, when using early stopping (with or

without patience), will save the model with best validation loss. This saved model is loaded and scored by SMAC3, and this score is used to further the search. For this reason, even if early stopping is not explicitly enabled by SMAC3, it is in effect used.

The difference comes in when patience is enabled. Since SMAC3 only defines a minimum budget of epochs at each round, it does not place a hard cap on maximum epochs run. For this reason, if patience is enabled, it is conceivable that some models will train unnecessarily long within their respective rounds, and will possibly ‘pollute’ the results with validation losses that were in fact obtained in effectively larger budgets than was specified by SMAC3.

In addition, intensification does not necessarily have to mean increasing the number of epochs in each round. There could be value in having the maximum budget of epochs be used from the beginning, albeit this may lead to longer training times. This could level the playing field between rounds, and enable model configurations to ‘mature’ before being incorporated into or rejected from the search space prematurely.

Here, we run a small test to characterise SMAC3’s behaviour in tandem with mustnet with the same configuration space and data set in the following three scenarios:

- Early stopping enabled with patience, 3 intensification rounds, minimum budget less than mustnet’s maximum epoch setting.
- Early stopping enabled without patience, 3 intensification rounds, minimum budget less than mustnet’s maximum epoch setting.
- Early stopping enabled with patience, 2 intensification rounds, minimum budget equal to mustnet’s maximum epoch setting.

The data set used for this experiment is a ‘mini’ version of the solar wind data set used in this study: with all storm severities from 1996 to 2000, all 10 input parameters, and e_h , Hermanus as the target. A time-shift size of 30 minutes is used, and a previous instance of the output label at 30 minutes in the past is appended to the input. The configuration space includes model layout {100 100; 100 100 100}, learning rate {0.0001 to 0.01 (uniform distribution)}, weight decay {0 to 0.001 (uniform distribution)} and batch size {32, 64, 128}. Bias nodes are applied to all layers, Adam is the optimisation algorithm, the loss function is MSE loss, ReLU is used

as the activation on the output node, and leaky ReLU is used as the activation on each hidden layer.

Figure A.17 portrays the configurations searched over in each scenario, their final best validation losses, and indicates the best configurations for each scenario. The Scenario that produced the best final validation loss selection is Scenario 2. For this reason, a setup similar to Scenario 2, with early stopping enabled without patience and 2+ intensification rounds, will be used as the optimisation scheme of this study.

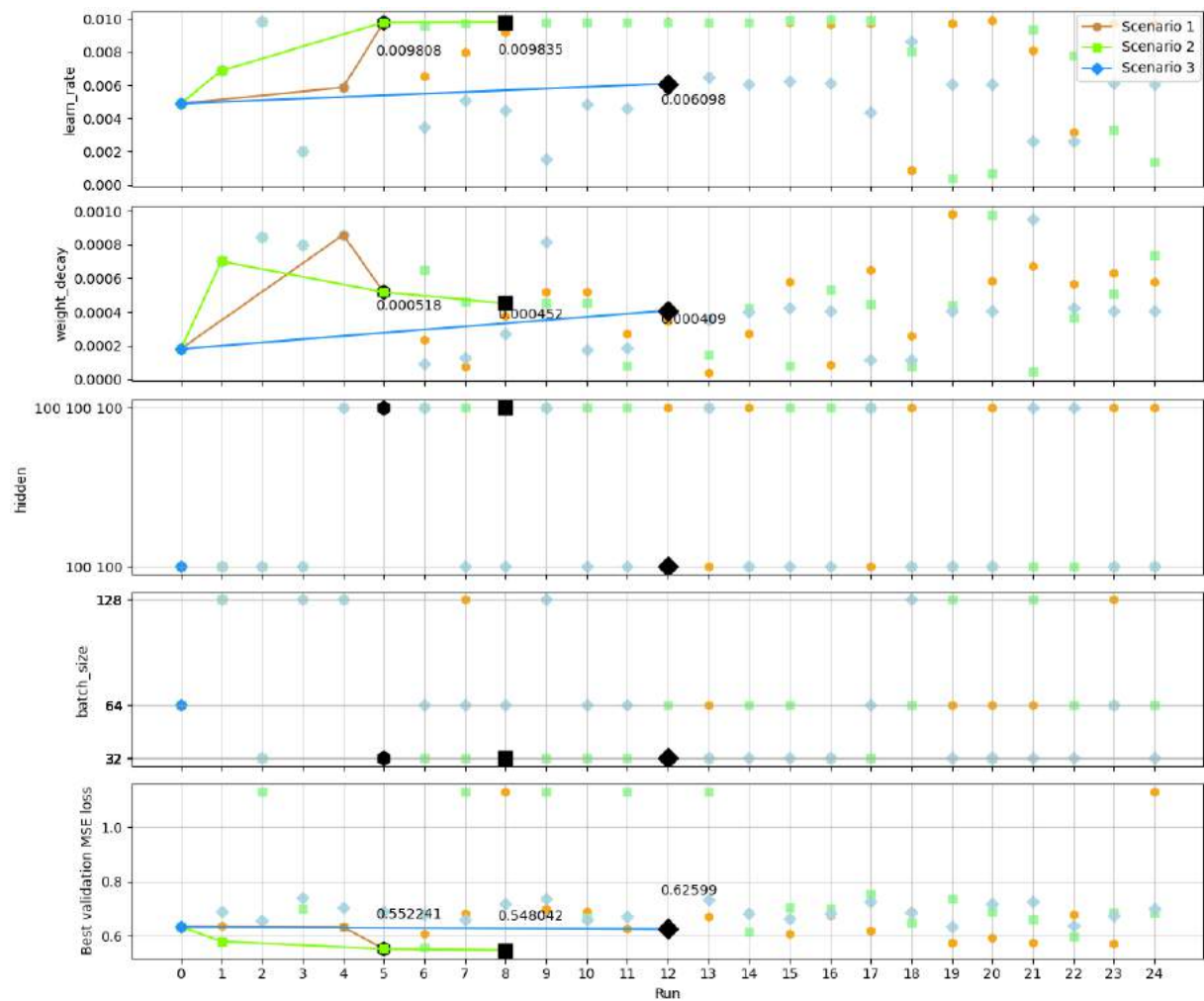
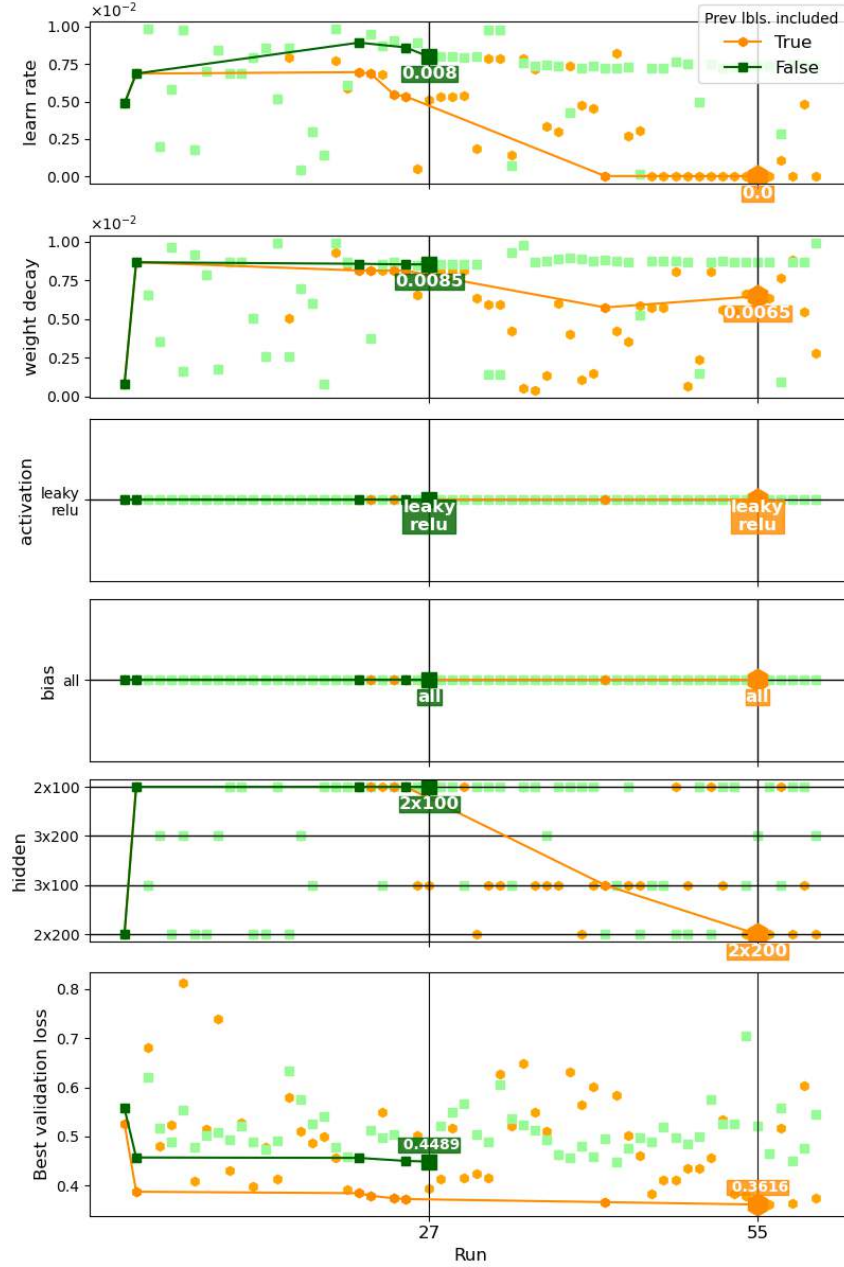


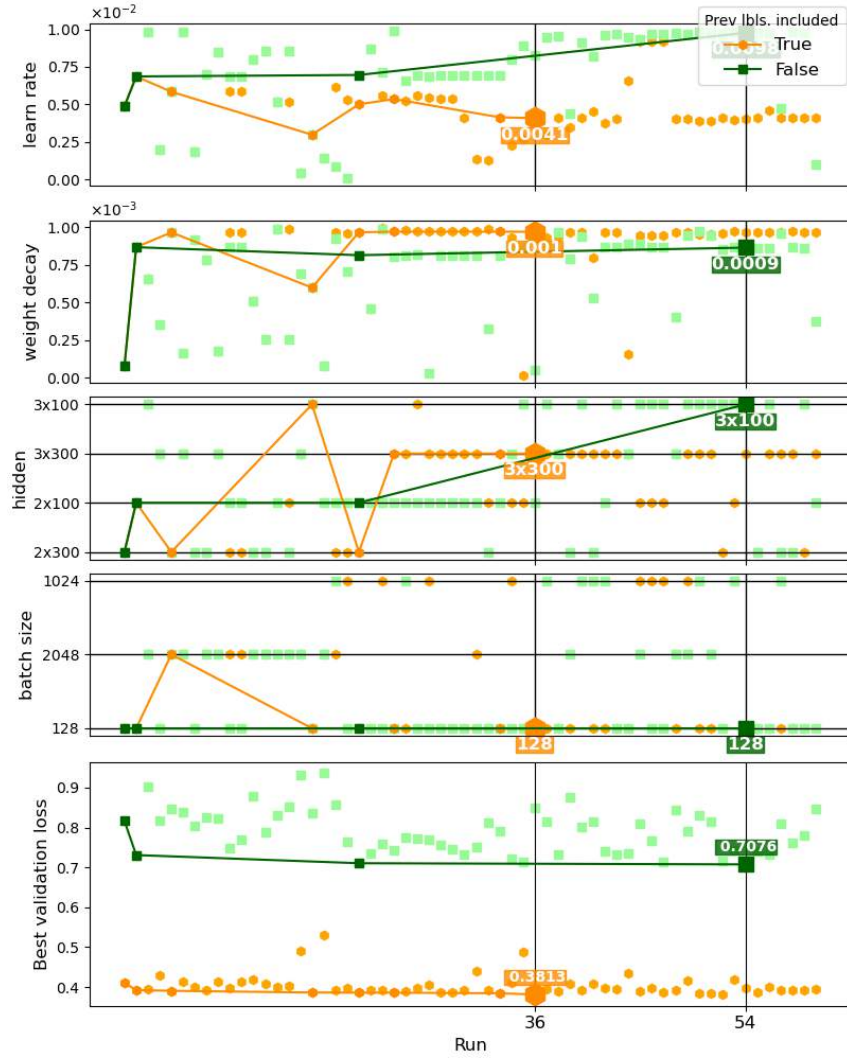
Figure A.17: Scenario plot portraying the configurations searched over in each scenario's configuration space. The scenarios portrayed here are Scenario 1: early stopping enabled with patience, 3 rounds of intensification, Scenario 2: early stopping enabled without patience, 3 rounds of intensification, and Scenario 3: early stopping enabled with patience, 2 rounds of intensification. Please note that the intensification rounds in Scenario 3 are rendered 'hobbled' by setting a constant epoch budget.

A.3.2 Optimisation results

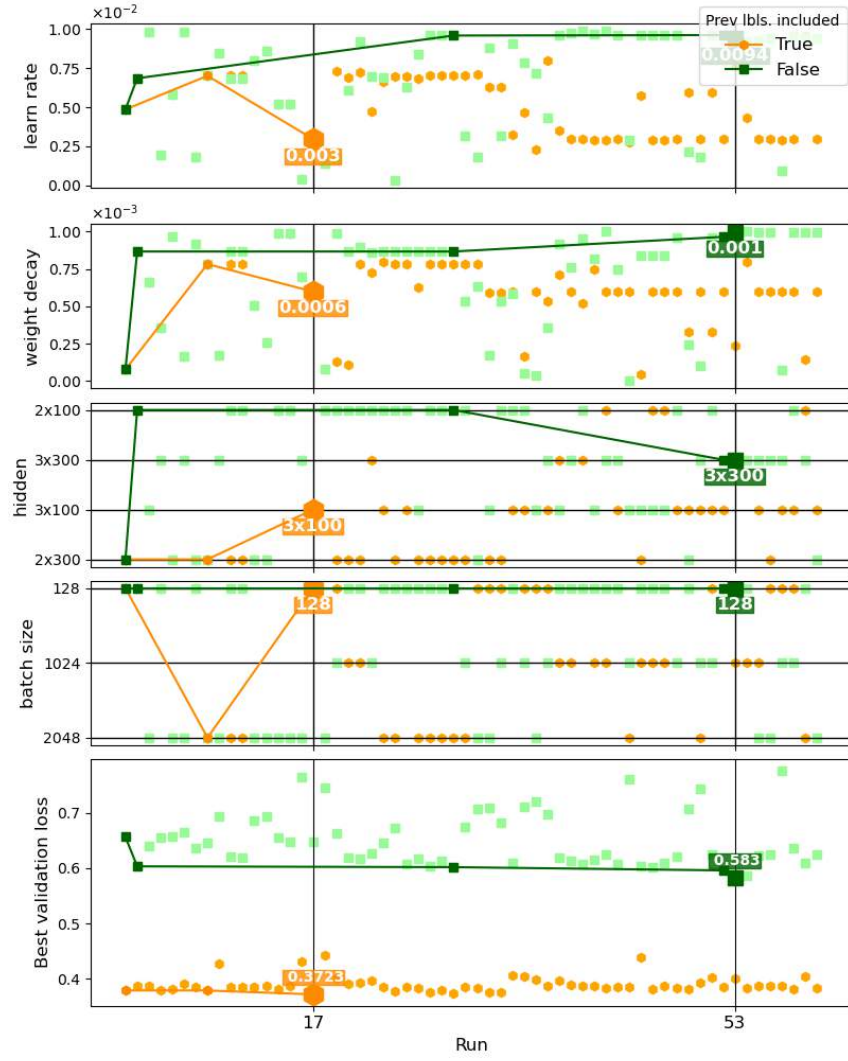
Here we display the scenario plots for all tasks with 30 minute time-shift sizes, for cases where auto-regressive input features are both included and excluded. In addition, the individual $e_{h_{HER}}$ and $e_{h_{ABK}}$ prediction tasks have scenario plots available for all four time-shift sizes, and are displayed here as well.



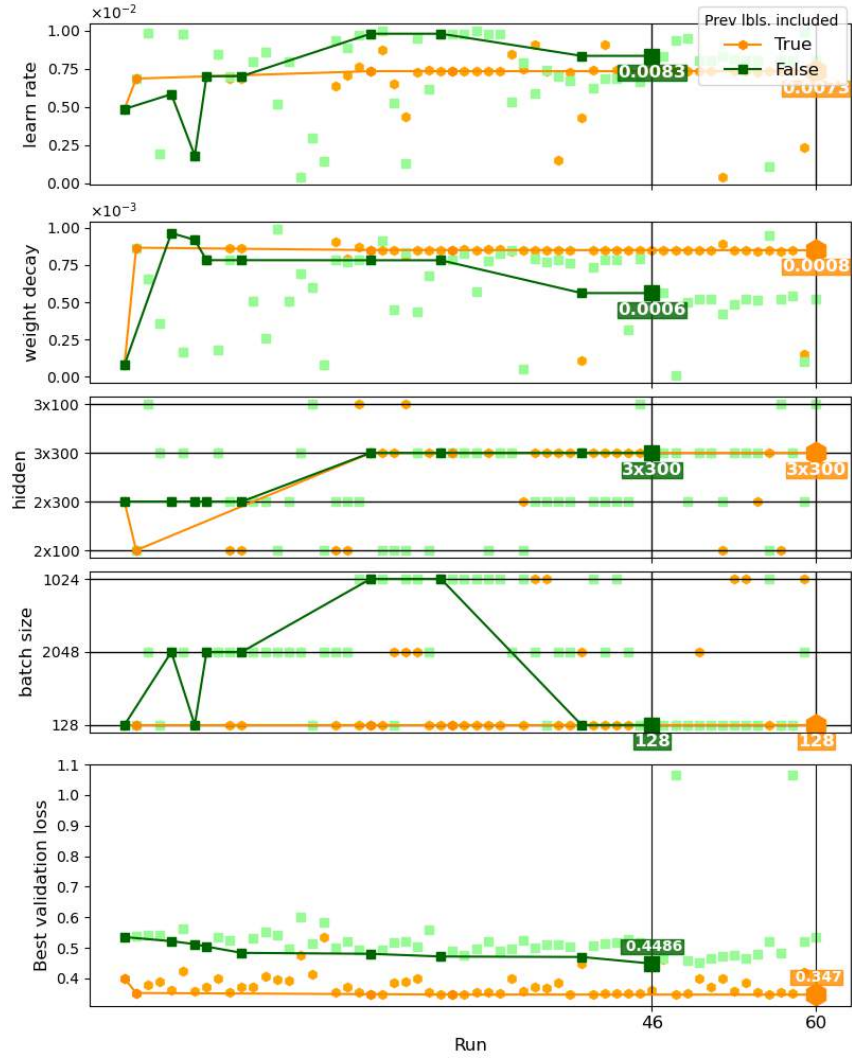
Scenario plot for the individual $e_{h_{HER}}$ prediction task, with auto-regressive input features included (True) and excluded (False).



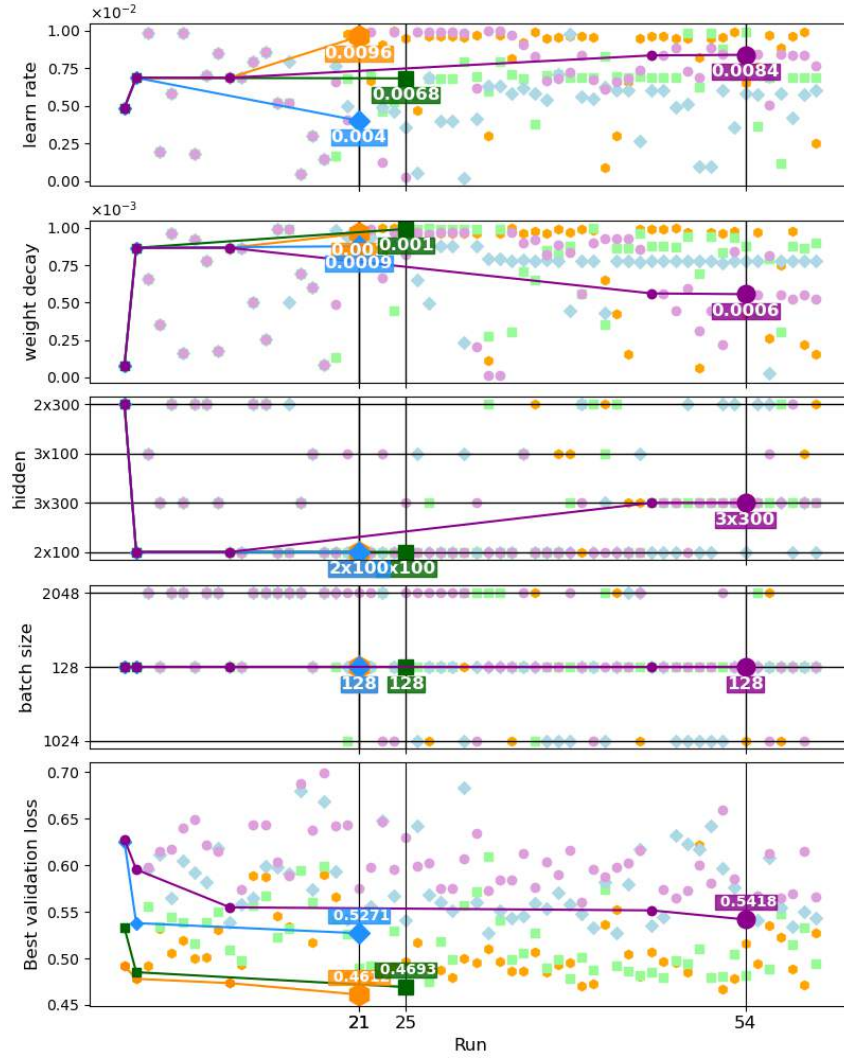
Scenario plot for the individual $e_{h_{ABK}}$ prediction task, with auto-regressive input features included (True) and excluded (False).



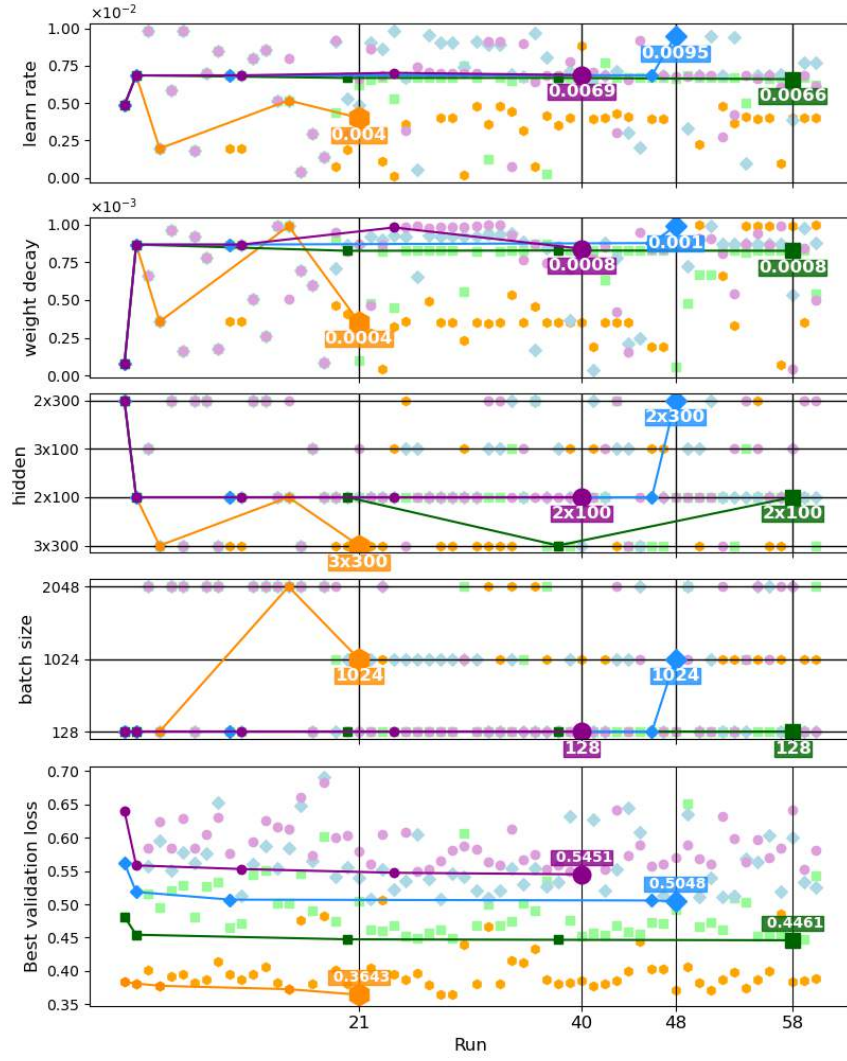
Scenario plot for the simultaneous $e_{h_{HER}}$ and $e_{h_{ABK}}$ prediction task, with auto-regressive input features included (True) and excluded (False).



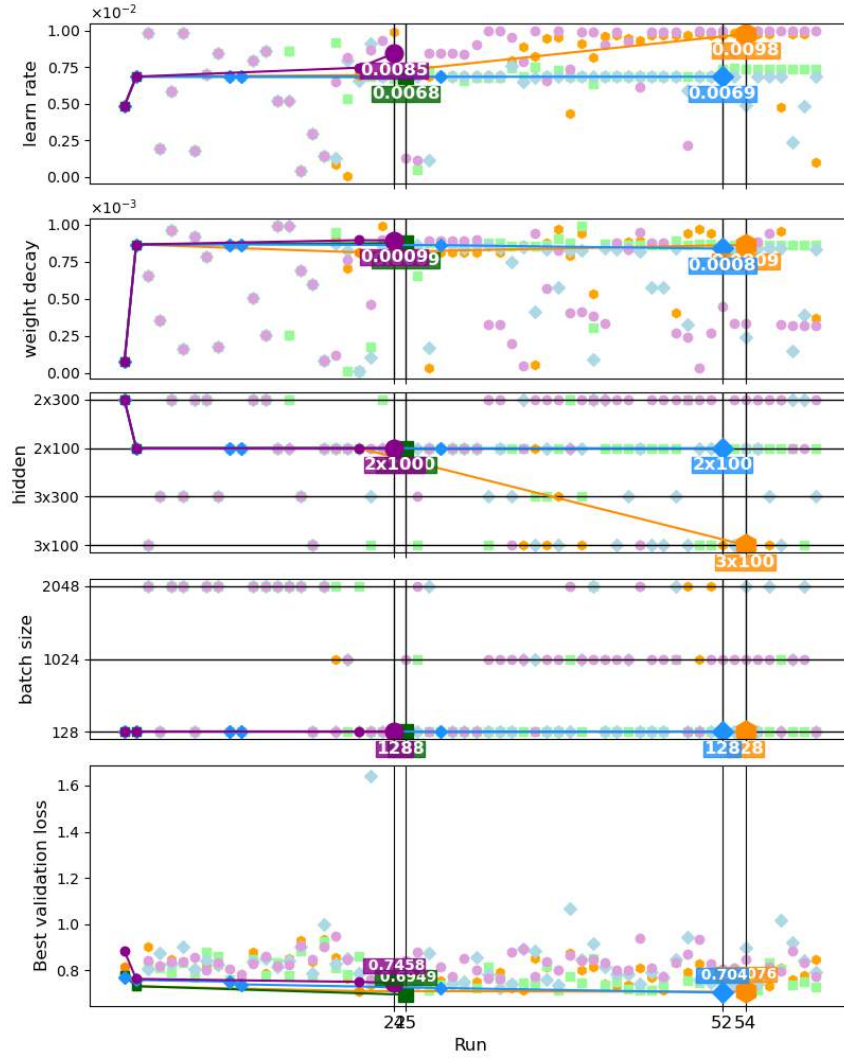
Scenario plot for the $|\Delta e_h|$ prediction task, with auto-regressive input features included (True) and excluded (False).



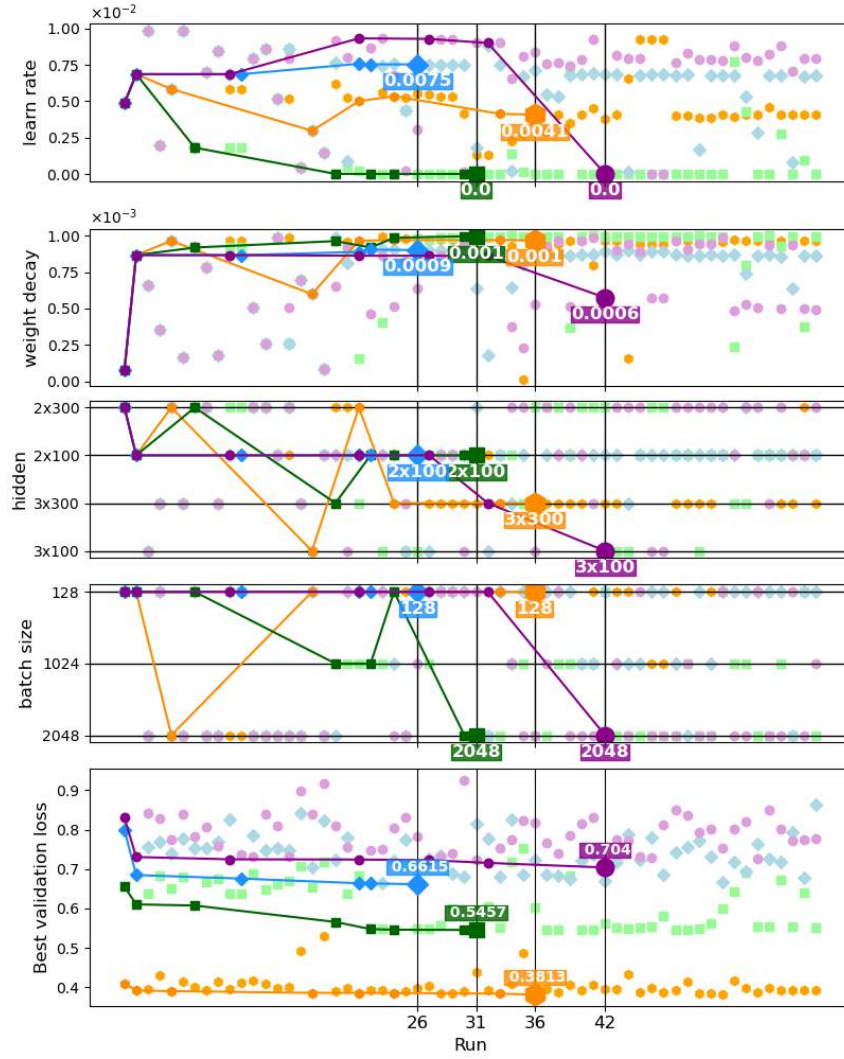
Scenario plot of all four time-shift sizes for the $e_{h_{HER}}$ prediction task, with auto-regressive input features excluded.



Scenario plot of all four time-shift sizes for the $e_{h_{HER}}$ prediction task, with auto-regressive input features included.



Scenario plot of all four time-shift sizes for the $e_{h_{ABK}}$ prediction task, with auto-regressive input features excluded.

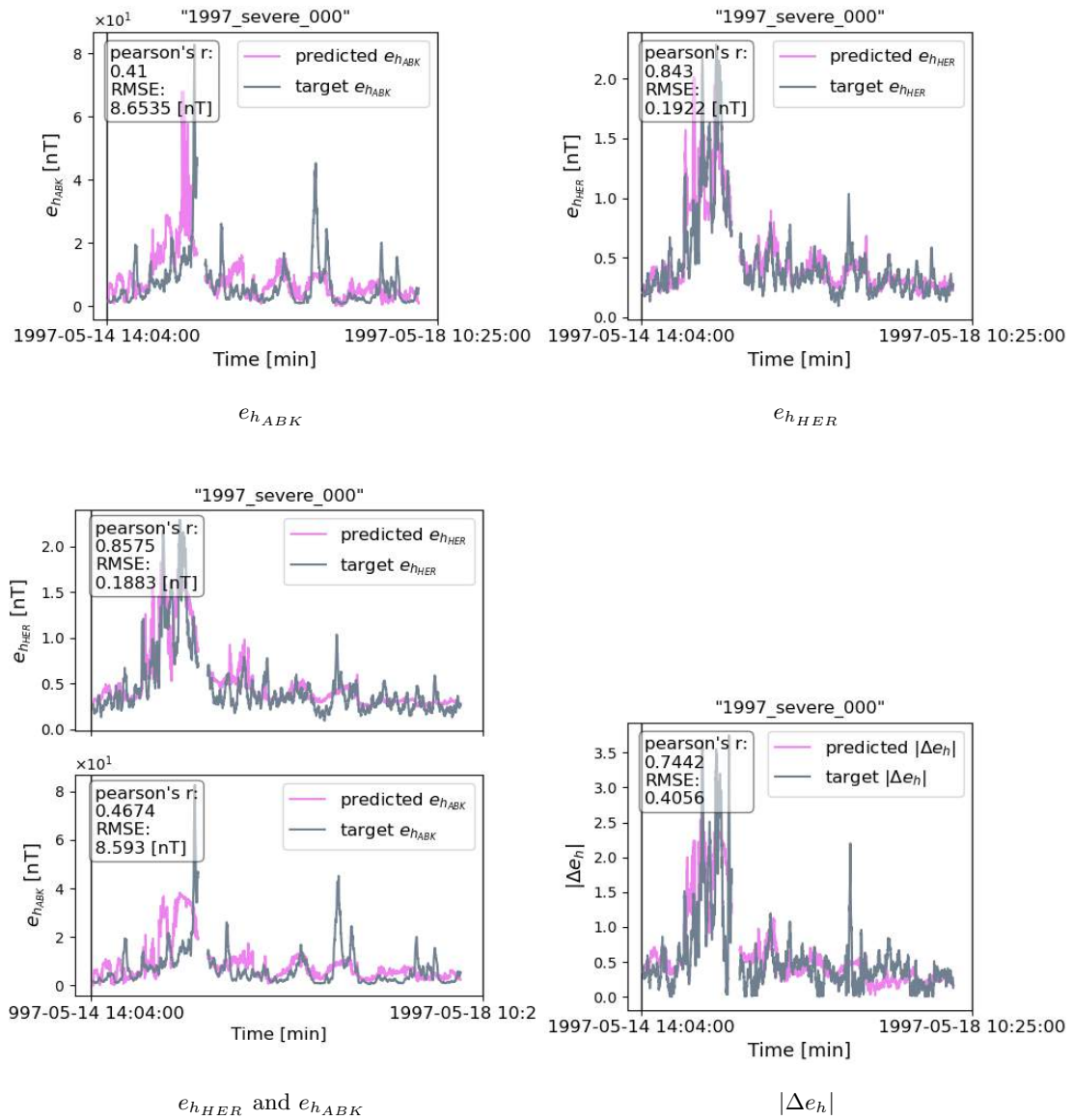


Scenario plot of all four time-shift sizes for the $e_{h_{ABK}}$ prediction task, with auto-regressive input features included.

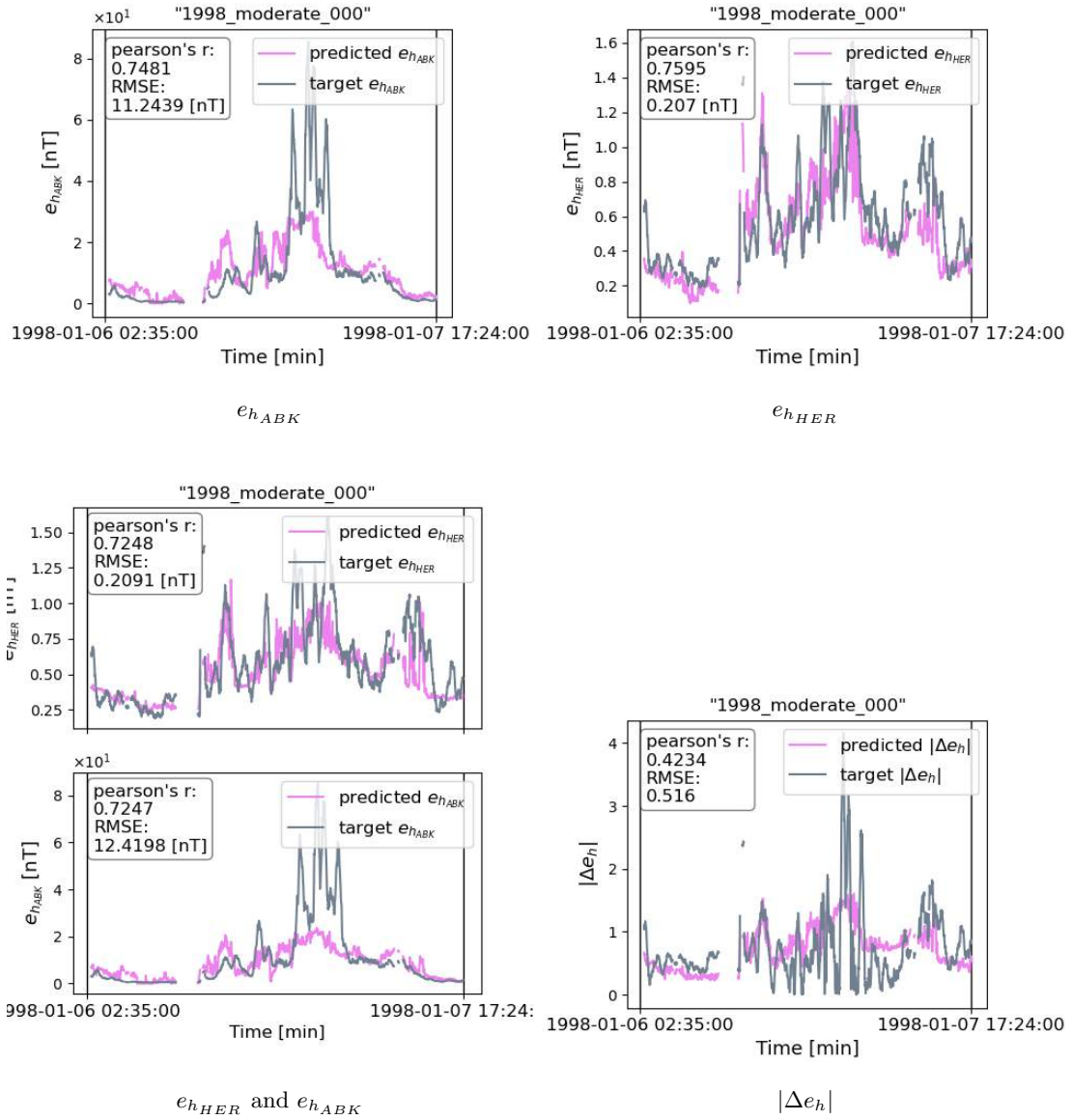
A.3.3 Output plots

Here we display the output plots for 10 randomly selected CME-driven storms and 10 randomly selected CIR-driven storms, with and without auto-regressive input features included, for each of the four prediction tasks. These storms are selected from the evaluation set.

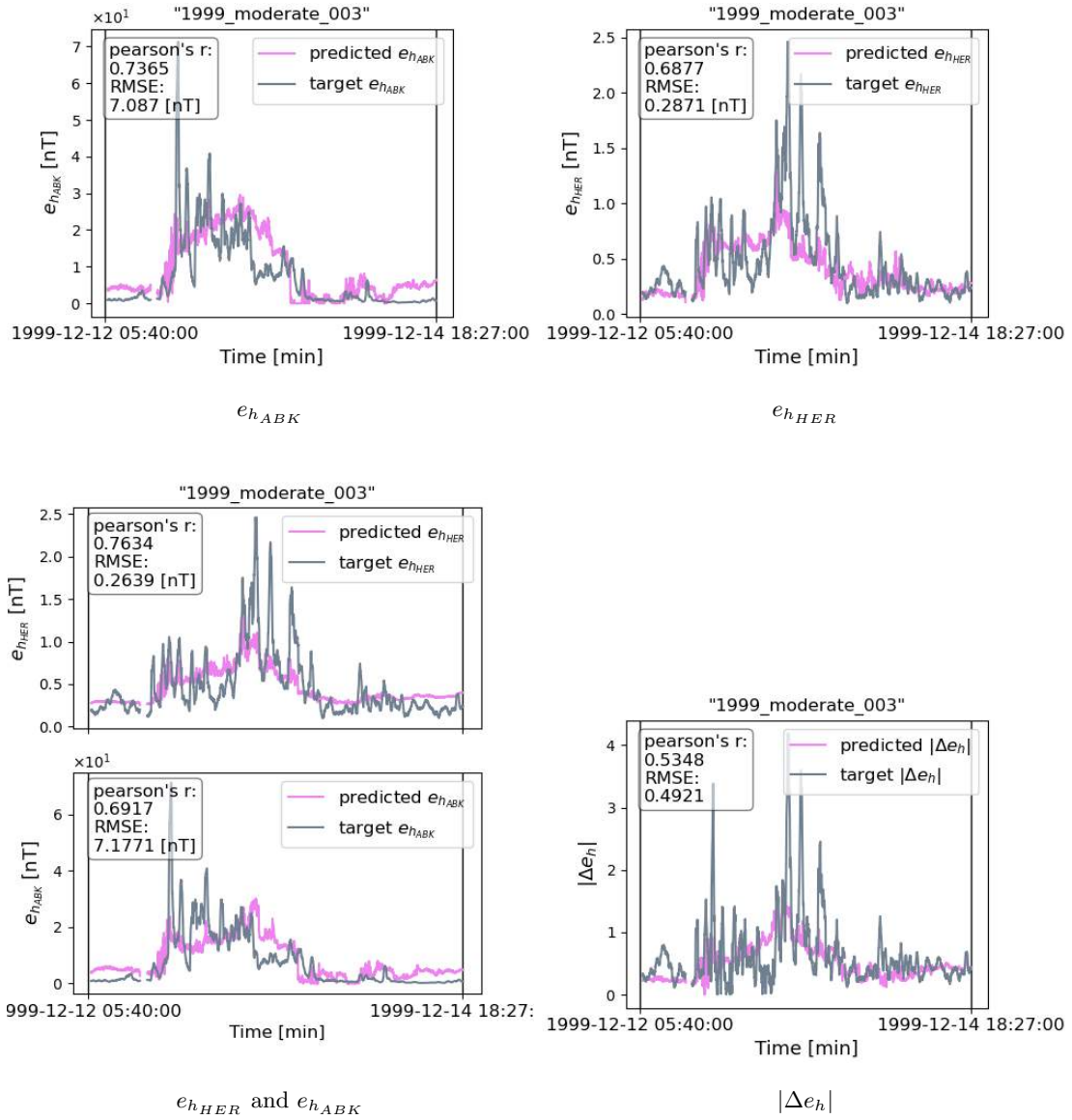
Output plots for CME-driven storms (no auto-regressive input features)



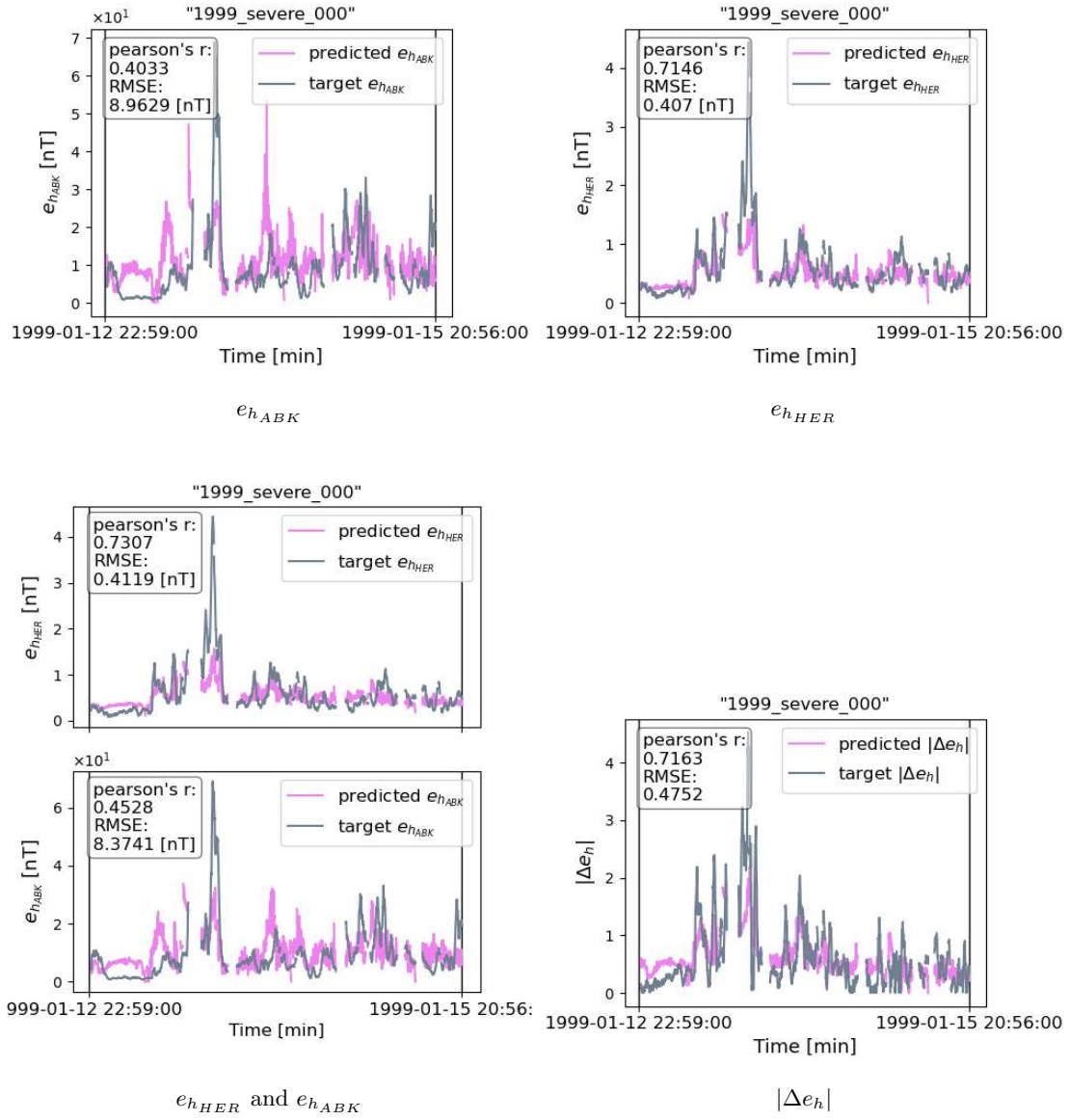
Output plots of 1997_severe_000 for all four tasks, no auto-regressive input features included.



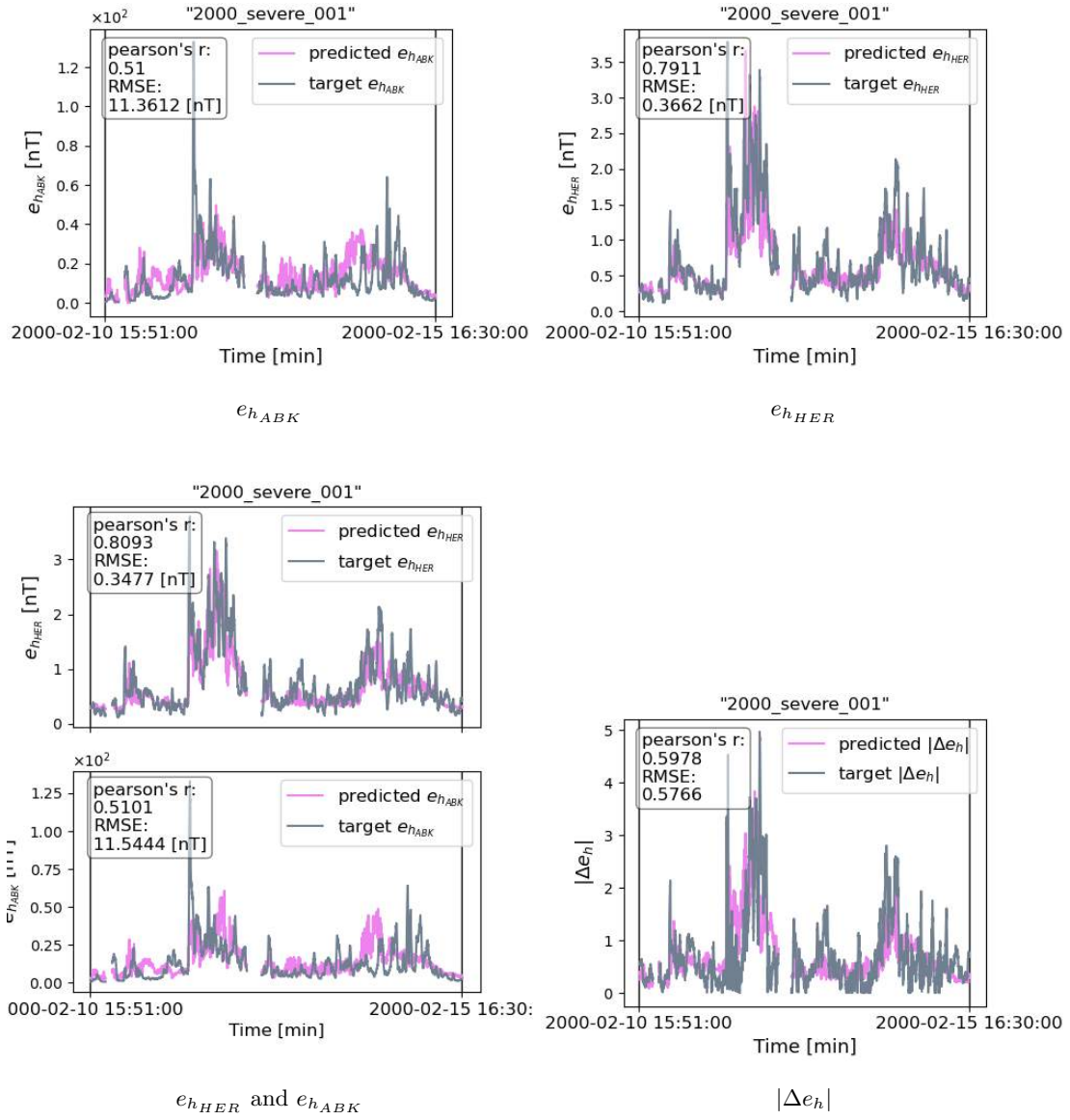
Output plots of 1998.moderate.000 for all four tasks, no auto-regressive input features included.



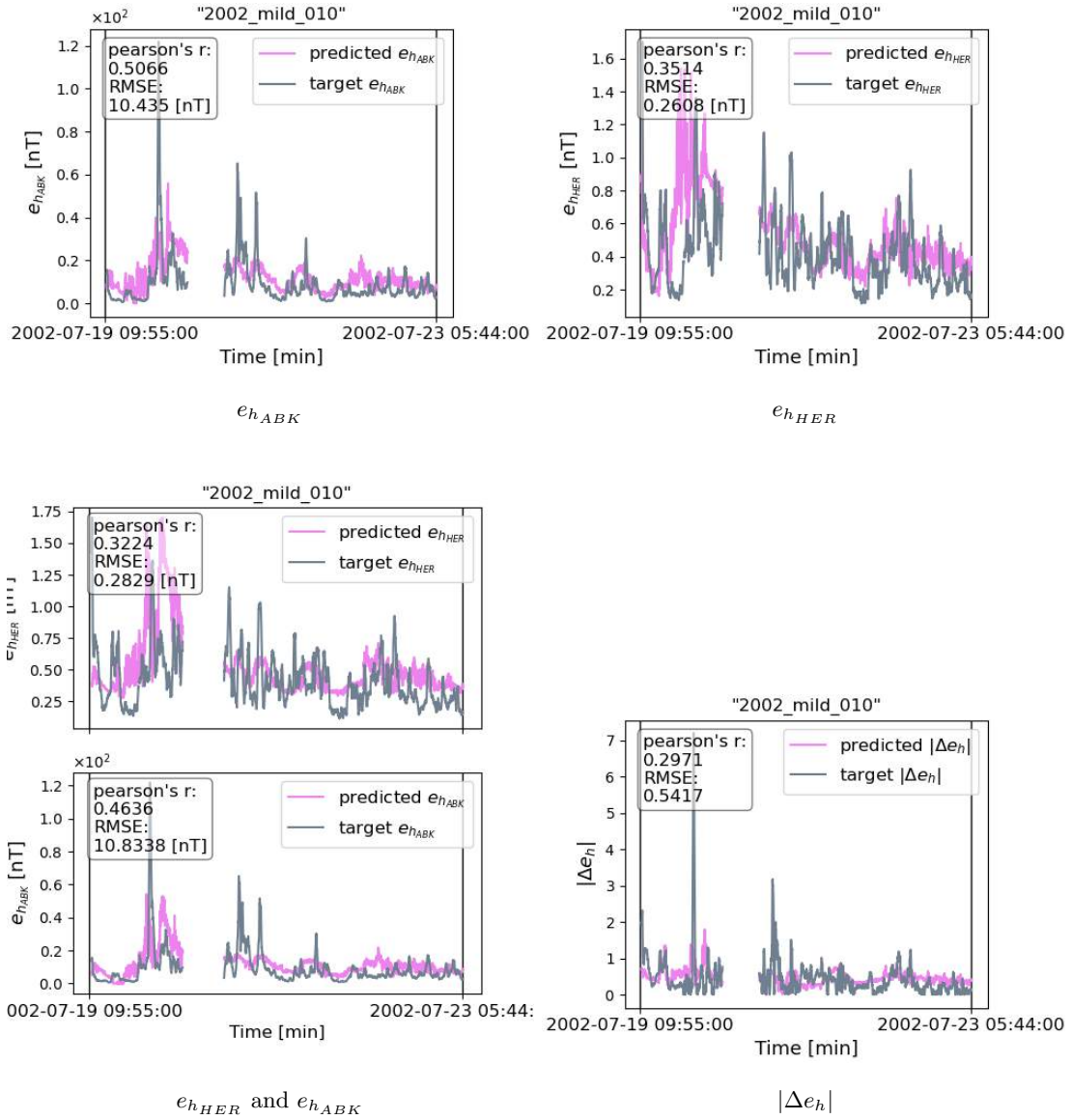
Output plots of 1999_moderate.003 for all four tasks, no auto-regressive input features included.



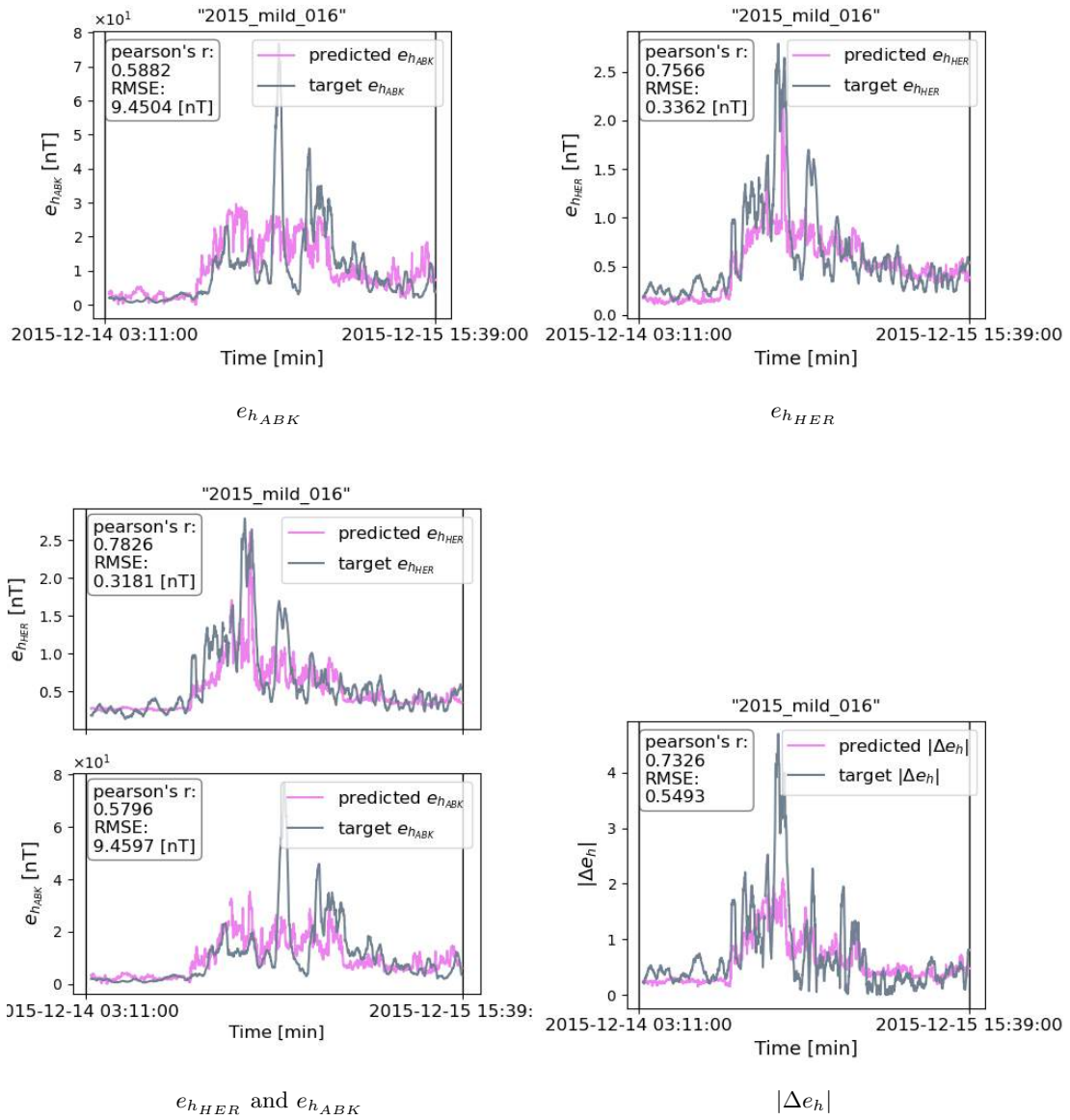
Output plots of 1999_severe_000 for all four tasks, no auto-regressive input features included.



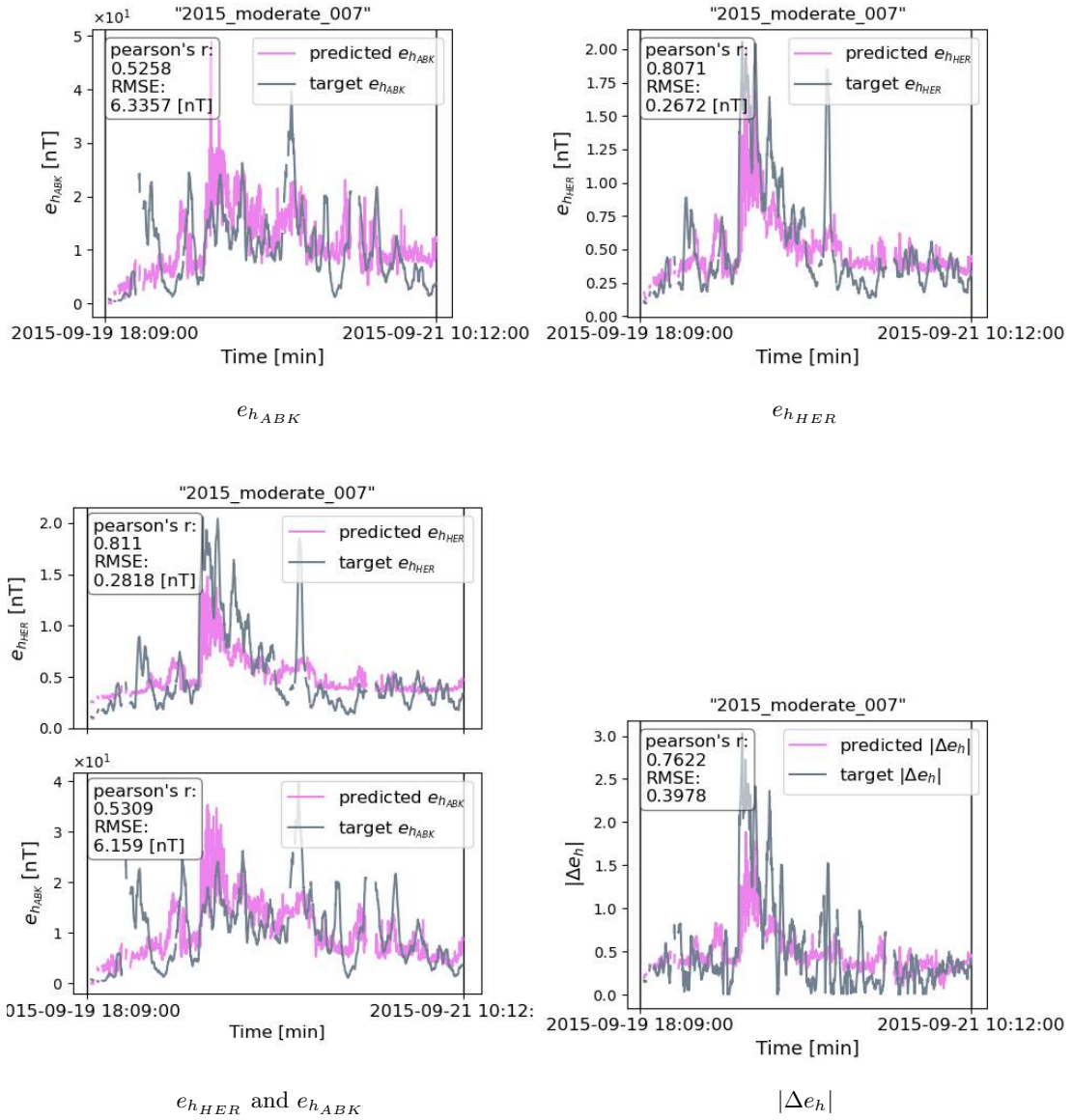
Output plots of 2000.severe.001 for all four tasks, no auto-regressive input features included.



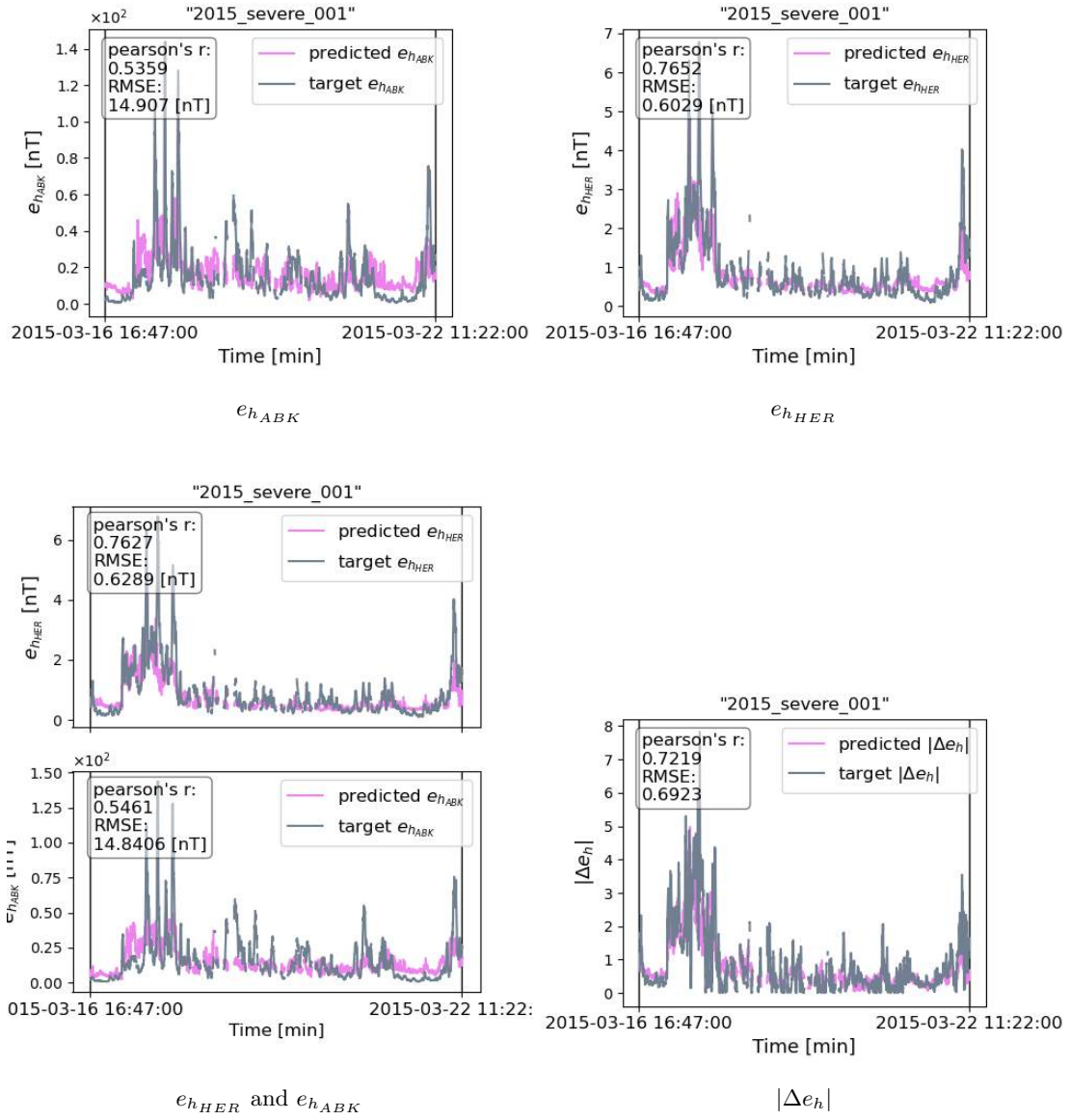
Output plots of 2002_mild.010 for all four tasks, no auto-regressive input features included.



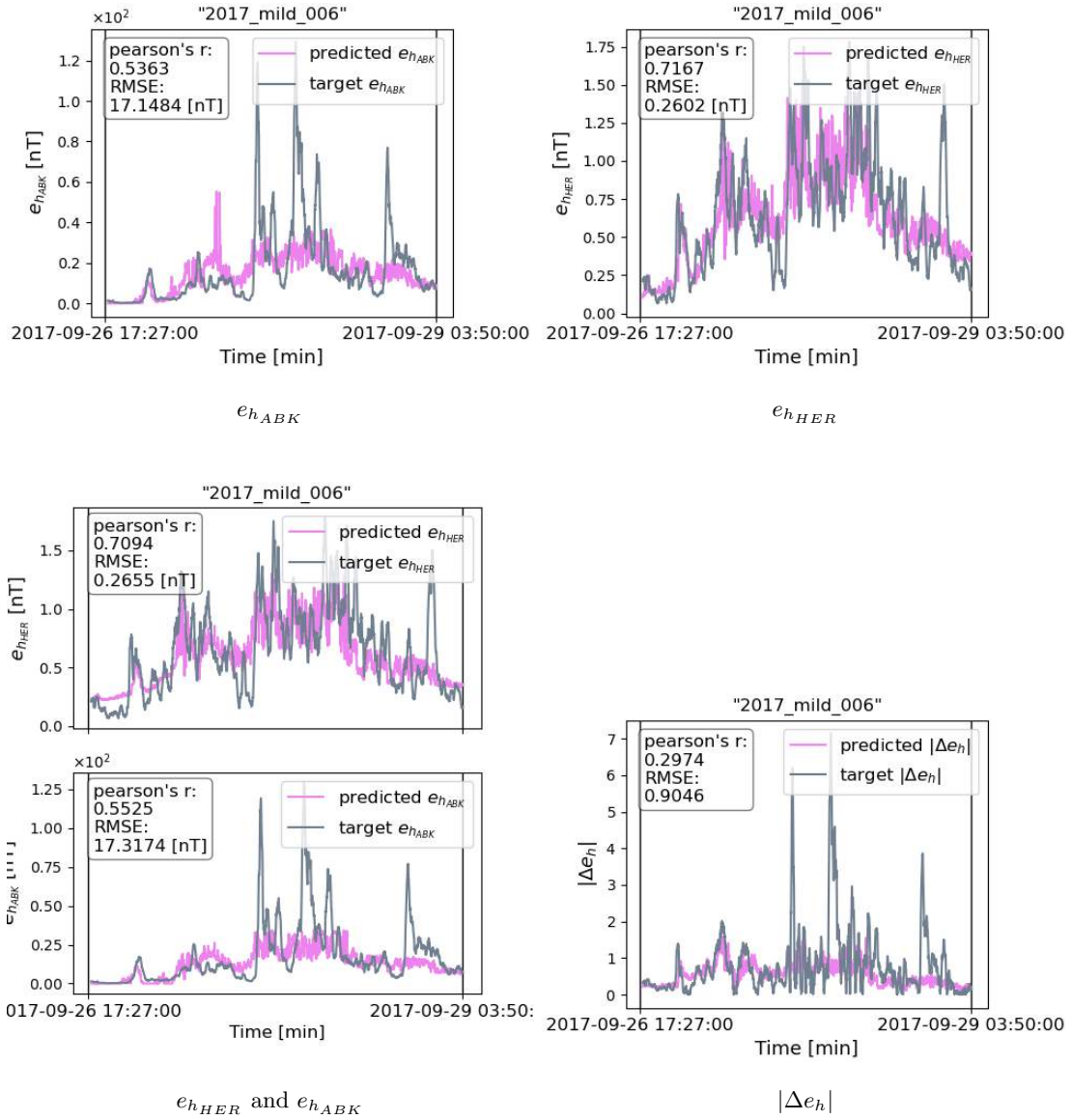
Output plots of 2015_mild.016 for all four tasks, no auto-regressive input features included.



Output plots of 2015.moderate.007 for all four tasks, no auto-regressive input features included.

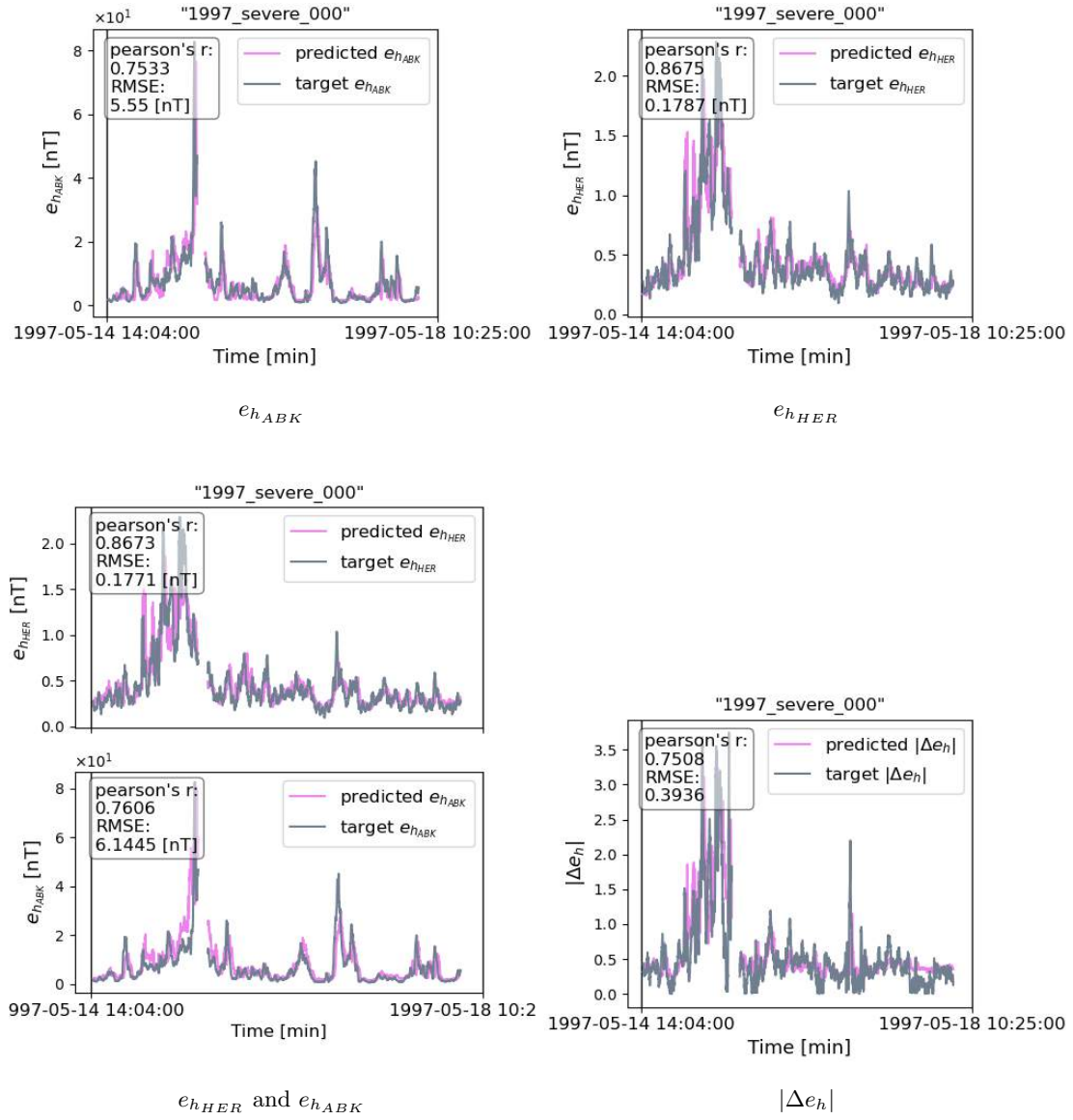


Output plots of 2015_severe_001 for all four tasks, no auto-regressive input features included.

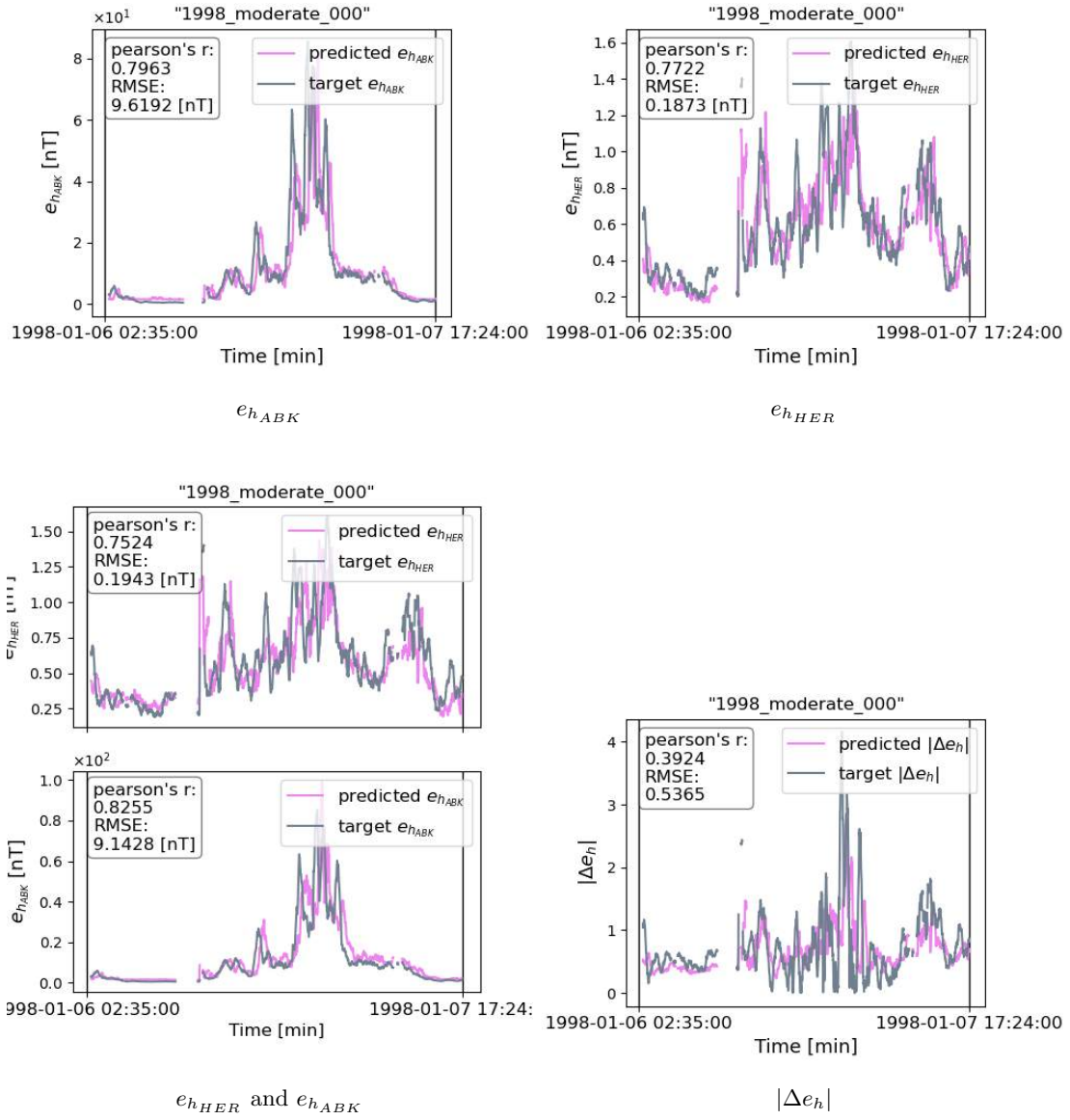


Output plots of 2017_mild.006 for all four tasks, no auto-regressive input features included.

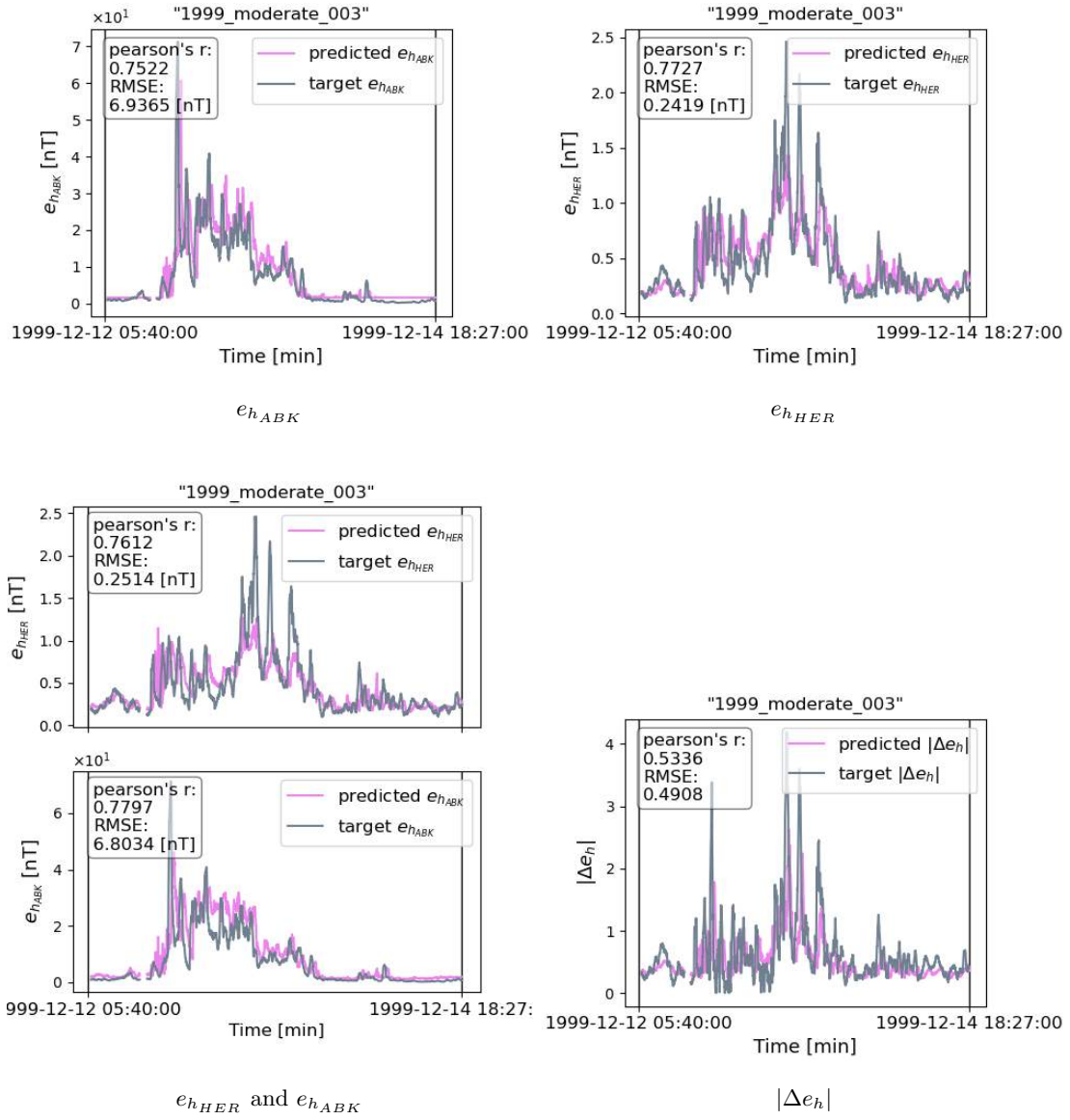
Output plots for CME-driven storms (auto-regressive input features included)



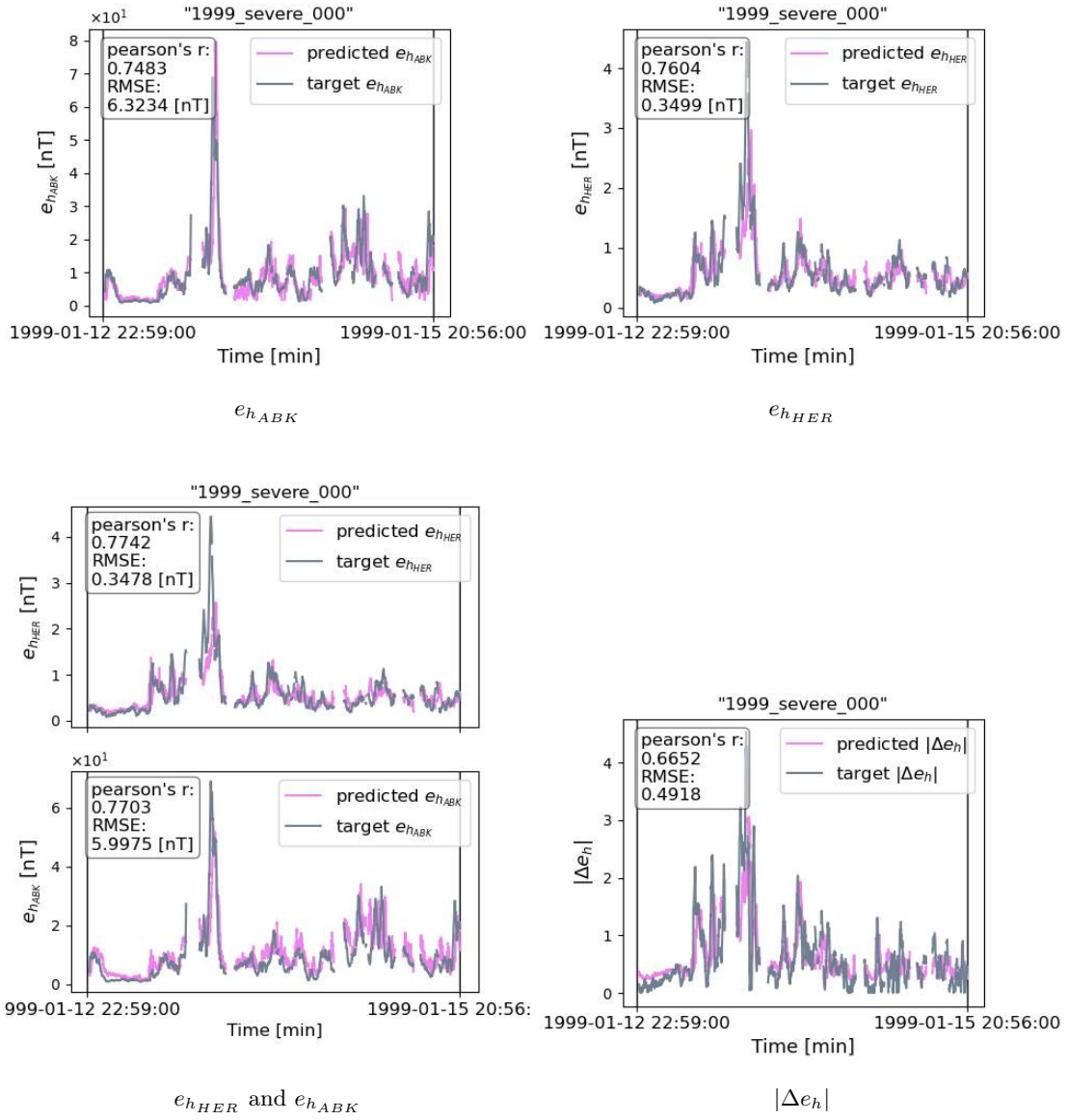
Output plots of 1997_severe_000 for all four tasks, auto-regressive input features included.



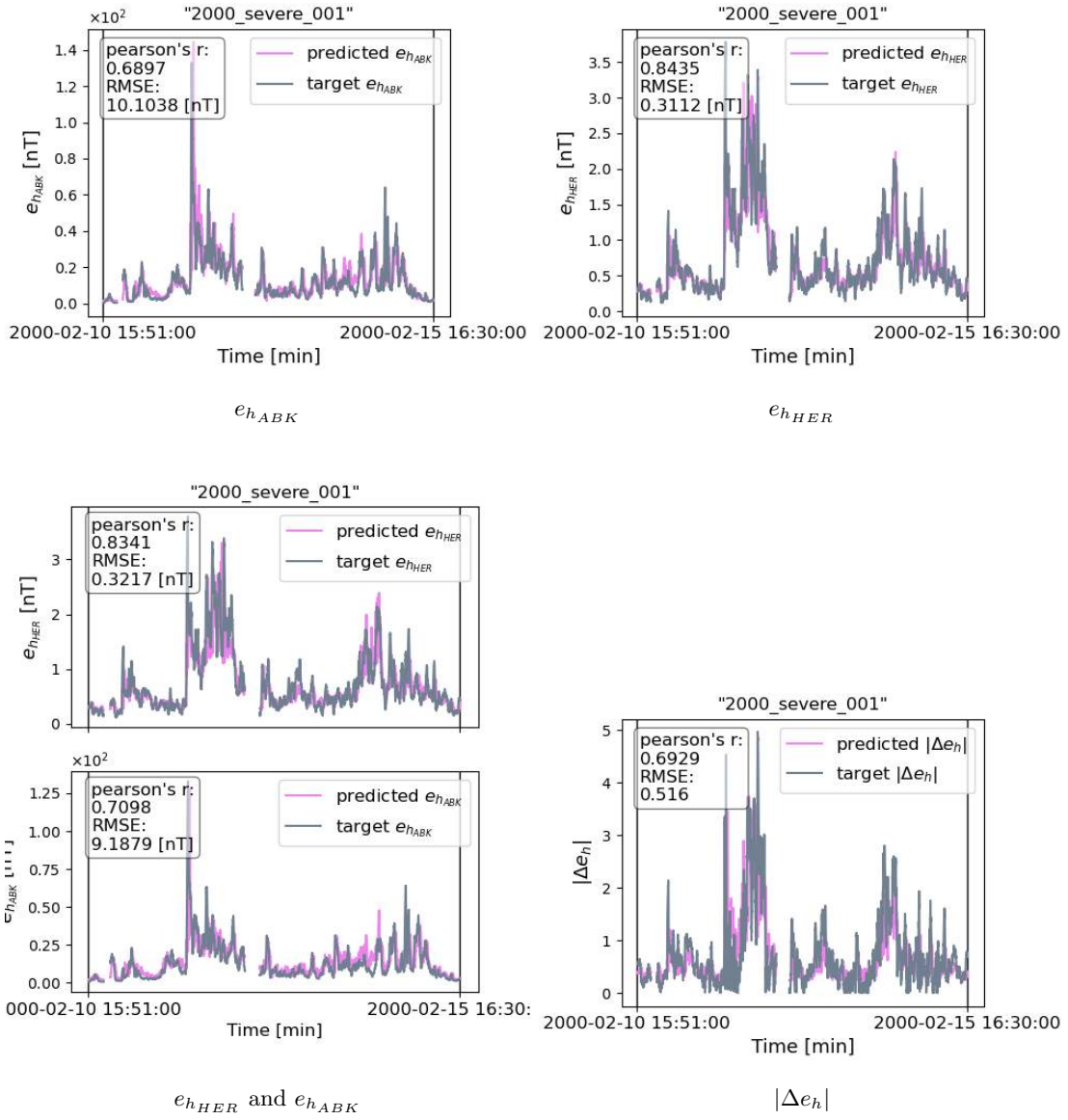
Output plots of 1998.moderate.000 for all four tasks, auto-regressive input features included.



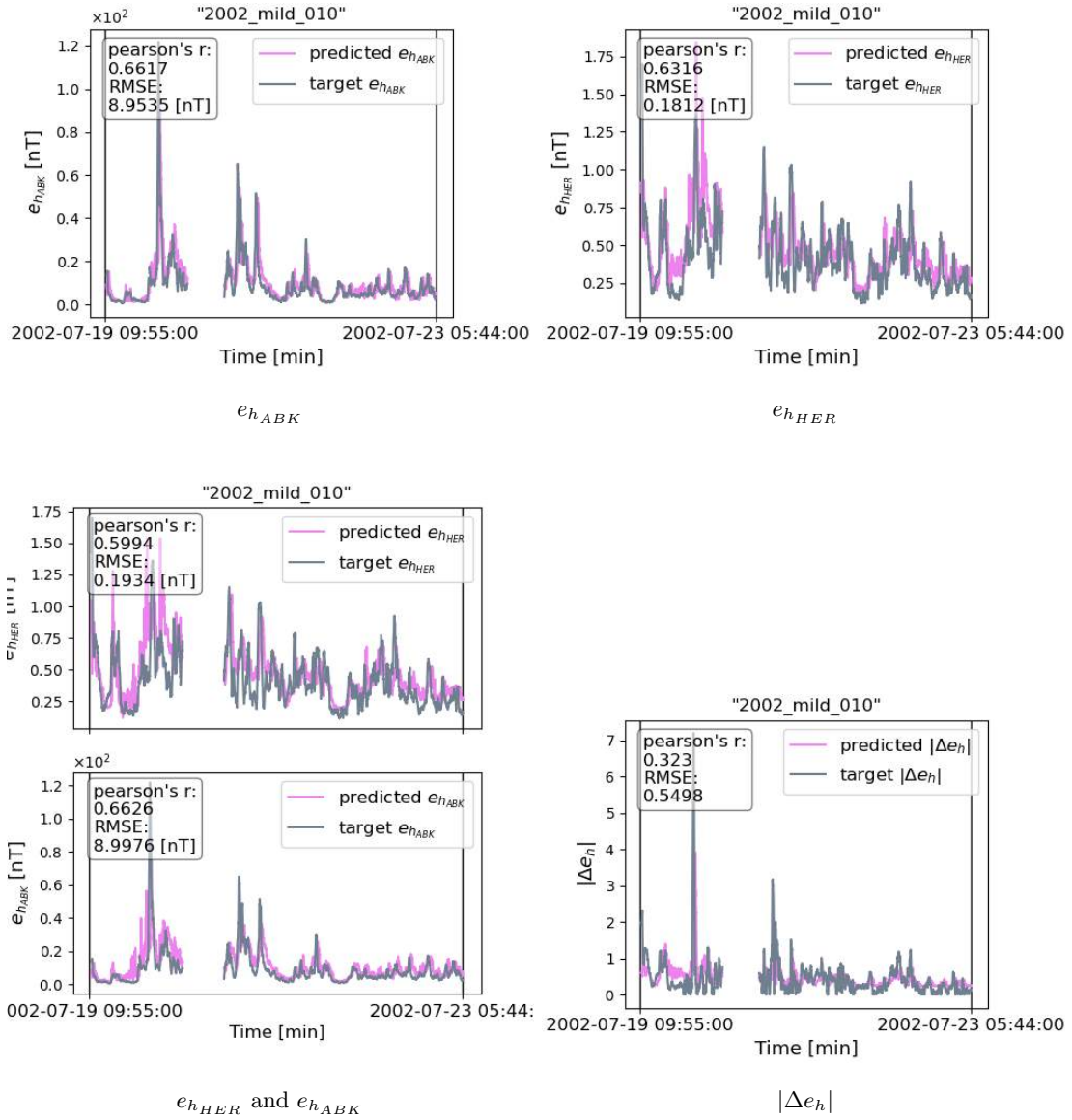
Output plots of 1999.moderate.003 for all four tasks, auto-regressive input features included.



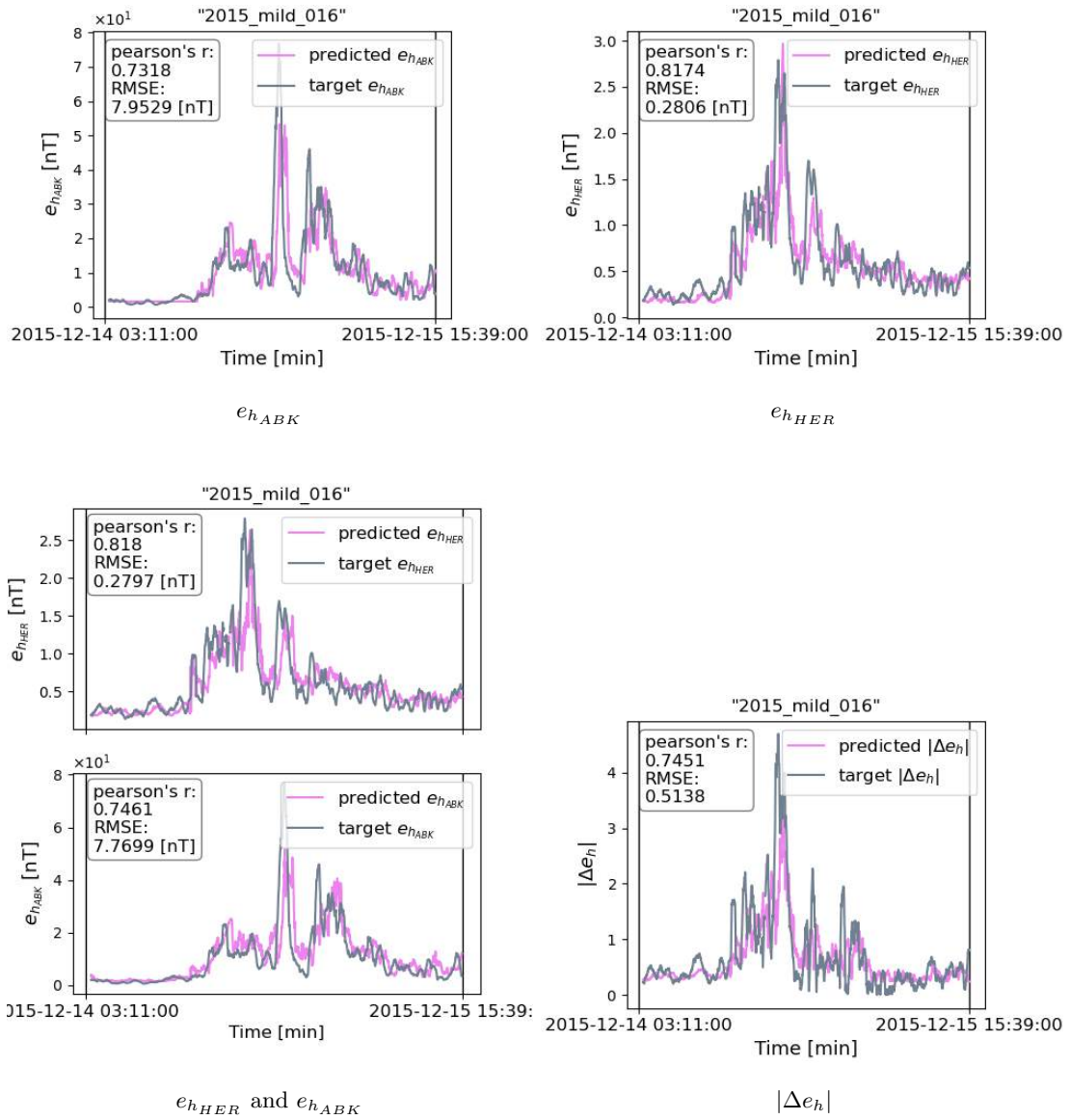
Output plots of 1999_severe.000 for all four tasks, auto-regressive input features included.



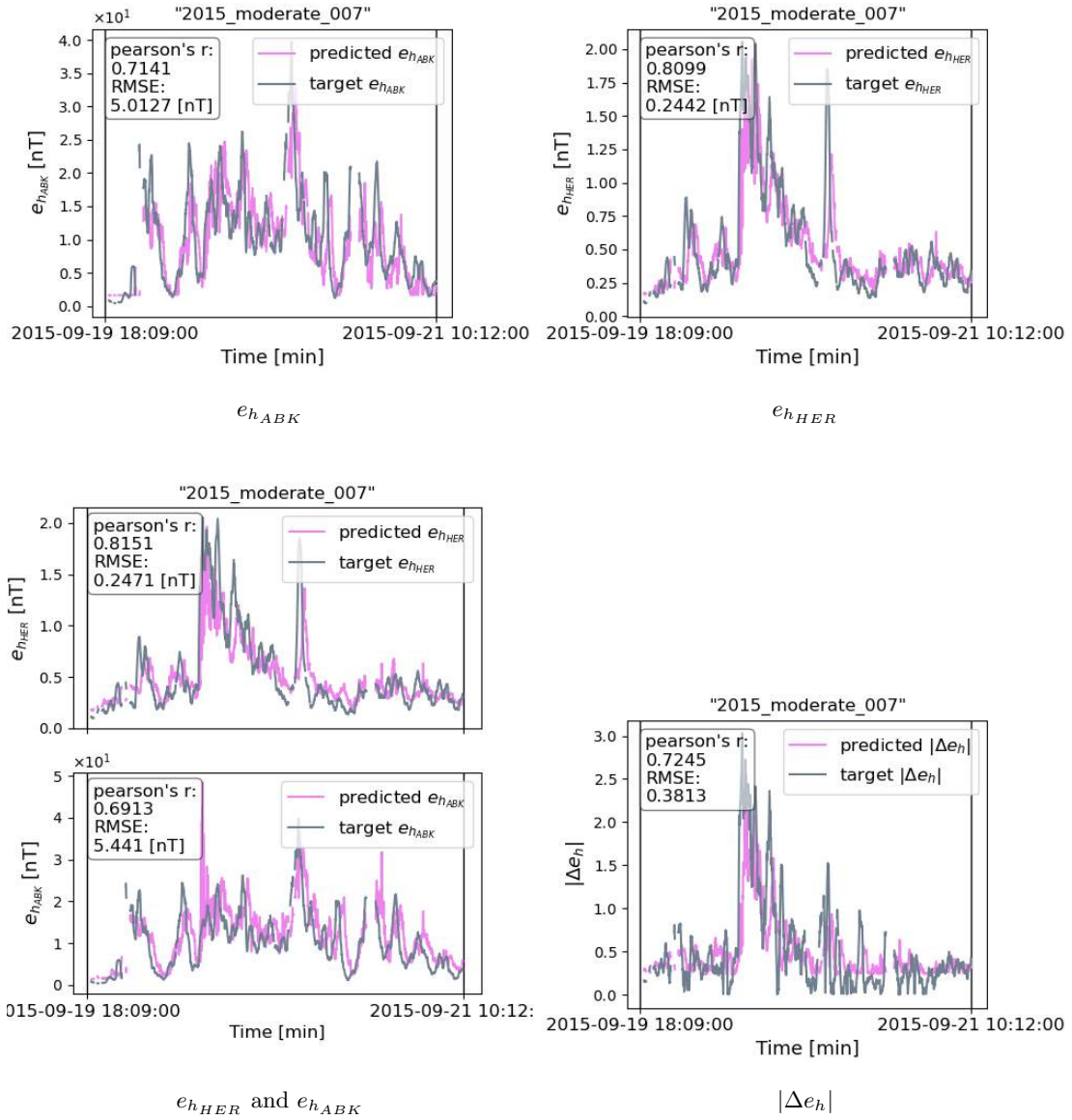
Output plots of 2000.severe.001 for all four tasks, auto-regressive input features included.



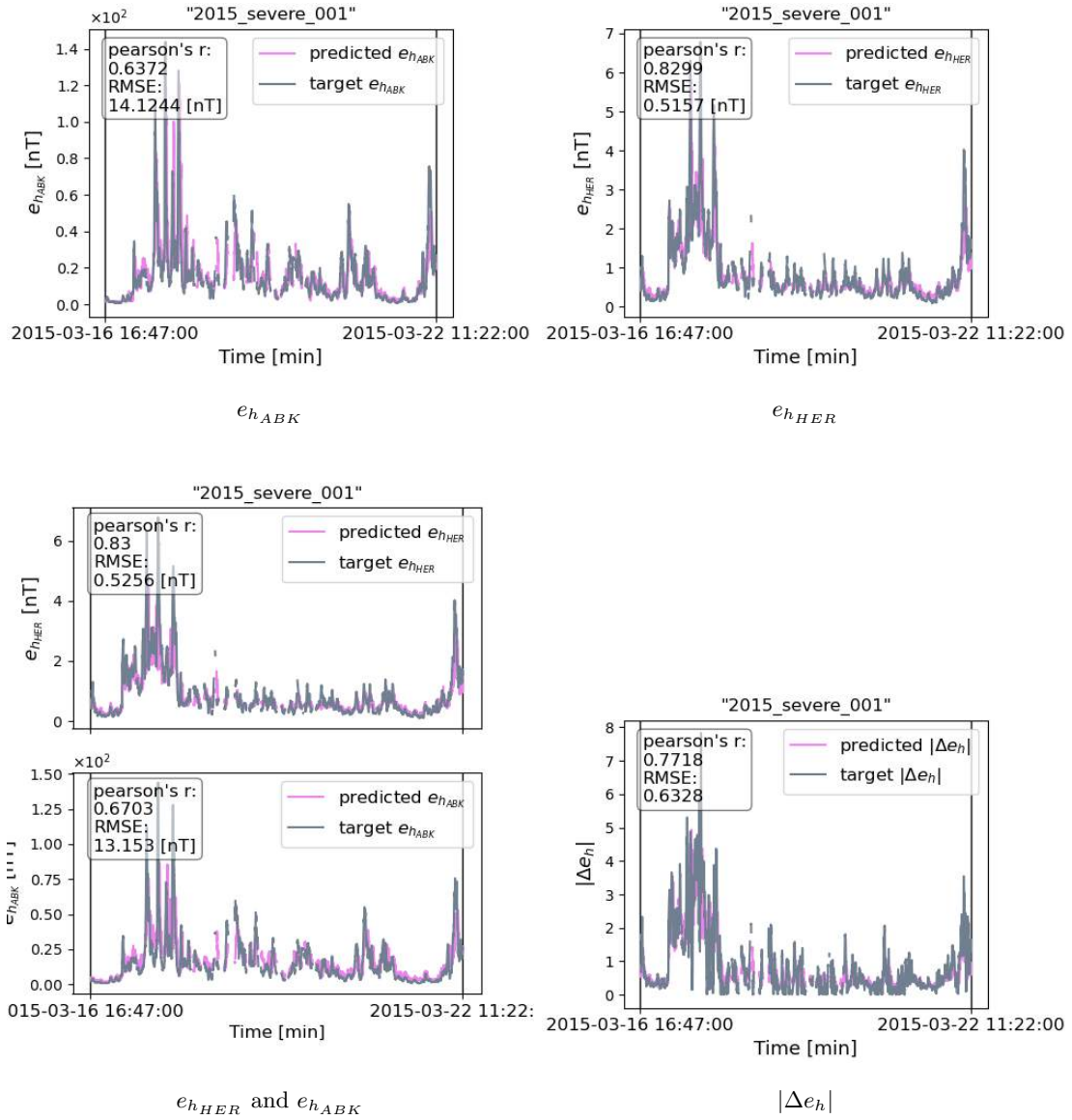
Output plots of 2002_mild.010 for all four tasks, auto-regressive input features included.



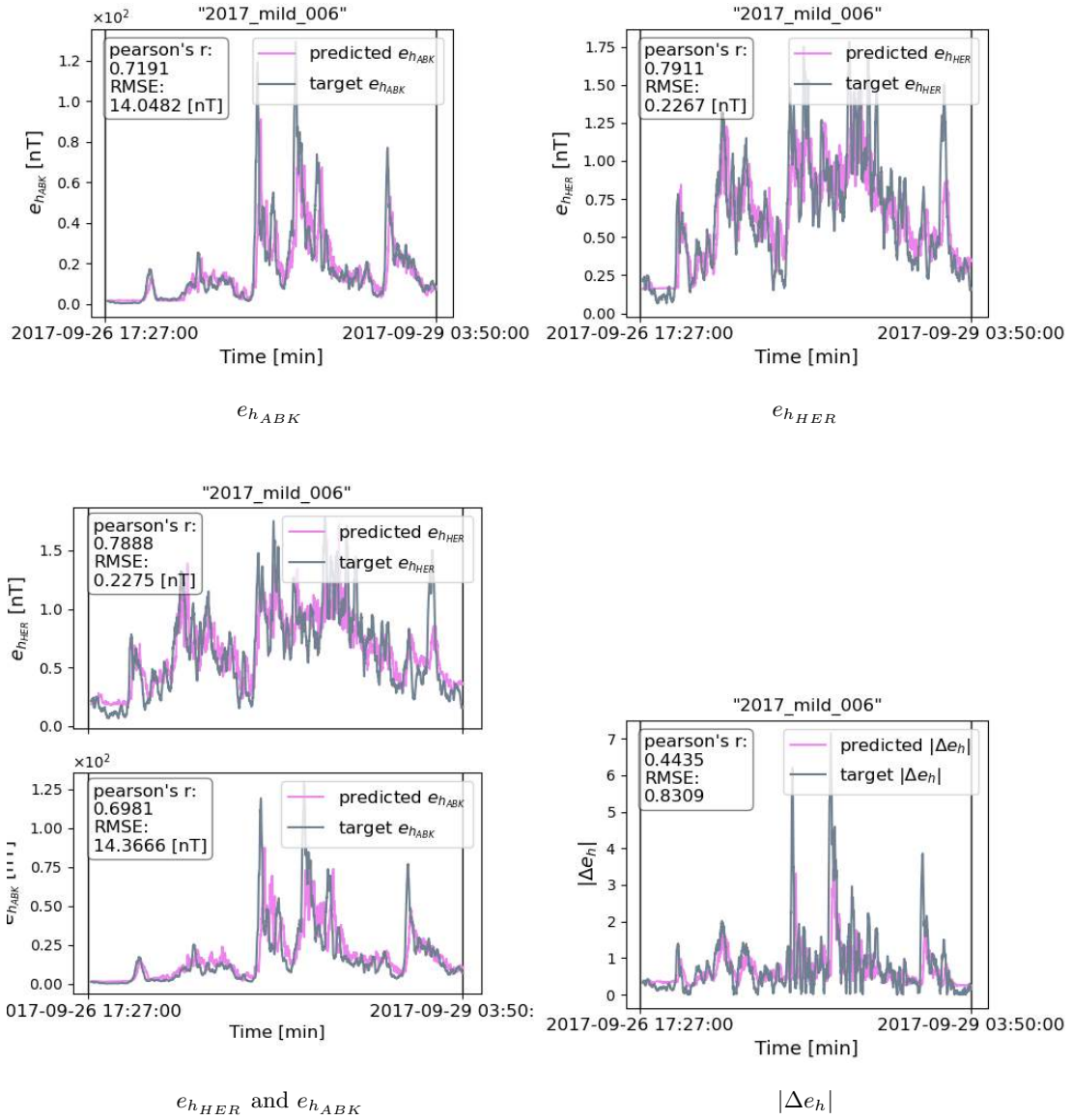
Output plots of 2015_mild.016 for all four tasks, auto-regressive input features included.



Output plots of 2015_moderate.007 for all four tasks, auto-regressive input features included.

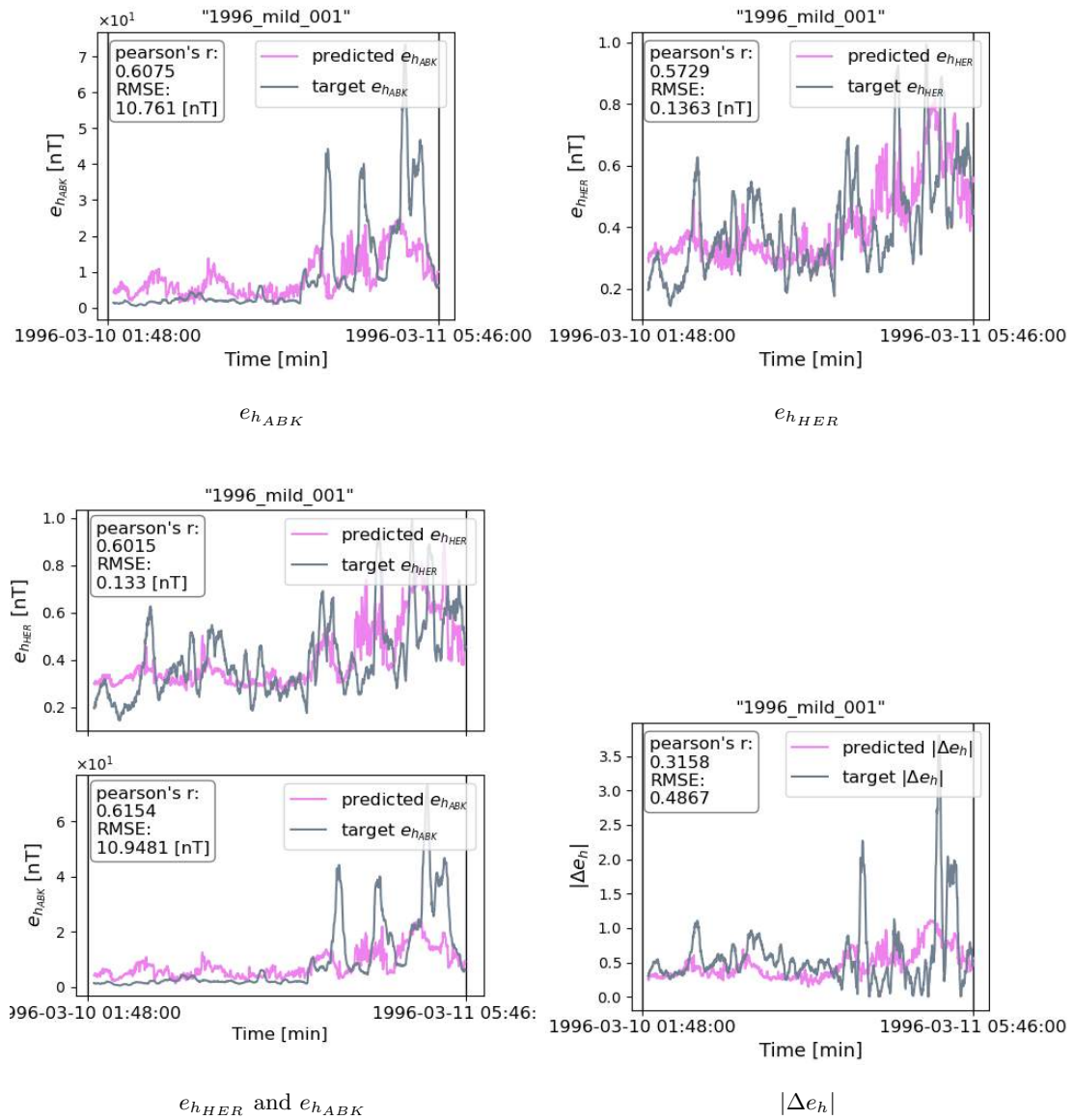


Output plots of 2015_severe.001 for all four tasks, auto-regressive input features included.

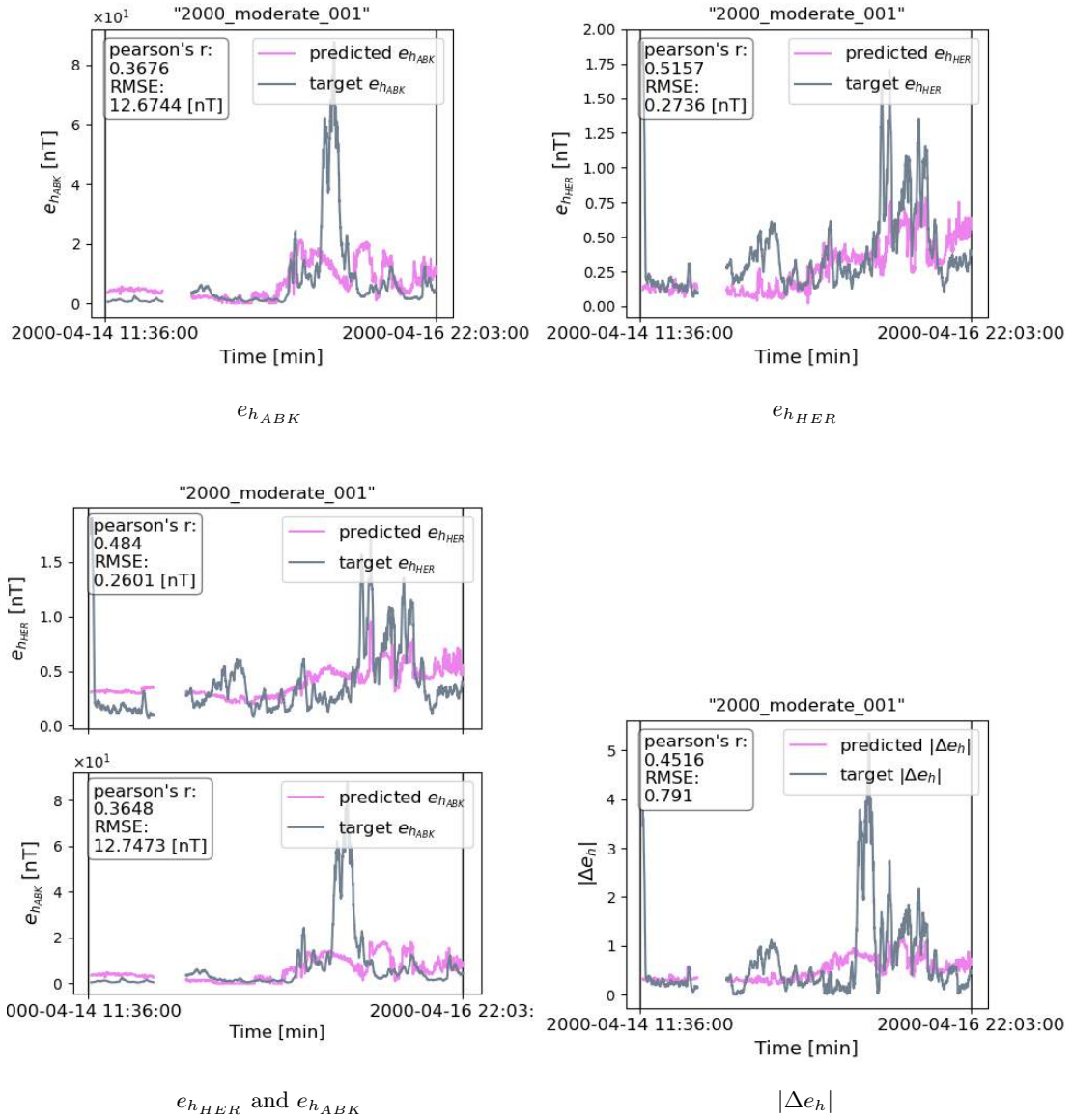


Output plots of 2017_mild.006 for all four tasks, auto-regressive input features included.

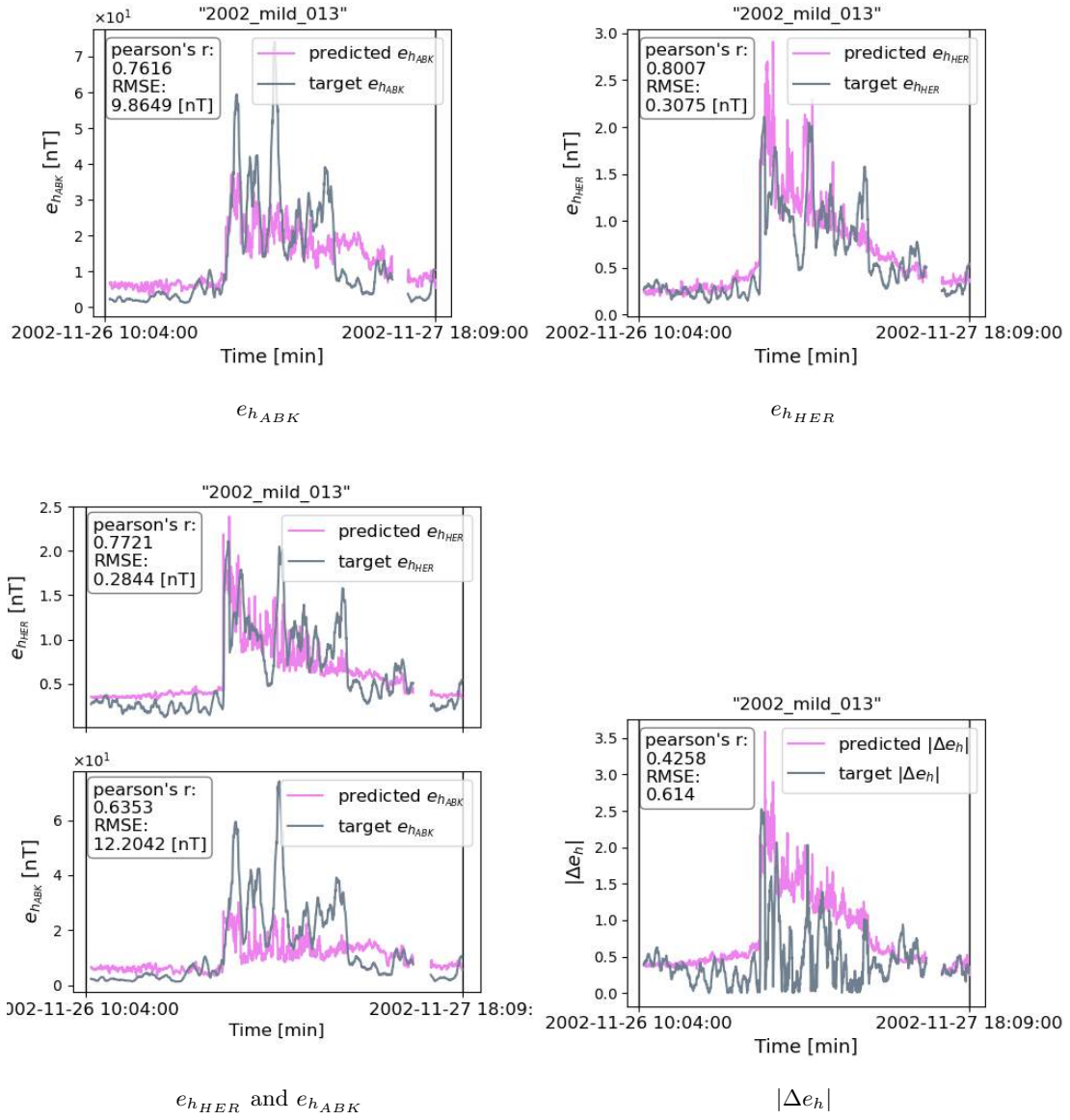
Output plots for CIR-driven storms (no auto-regressive input features)



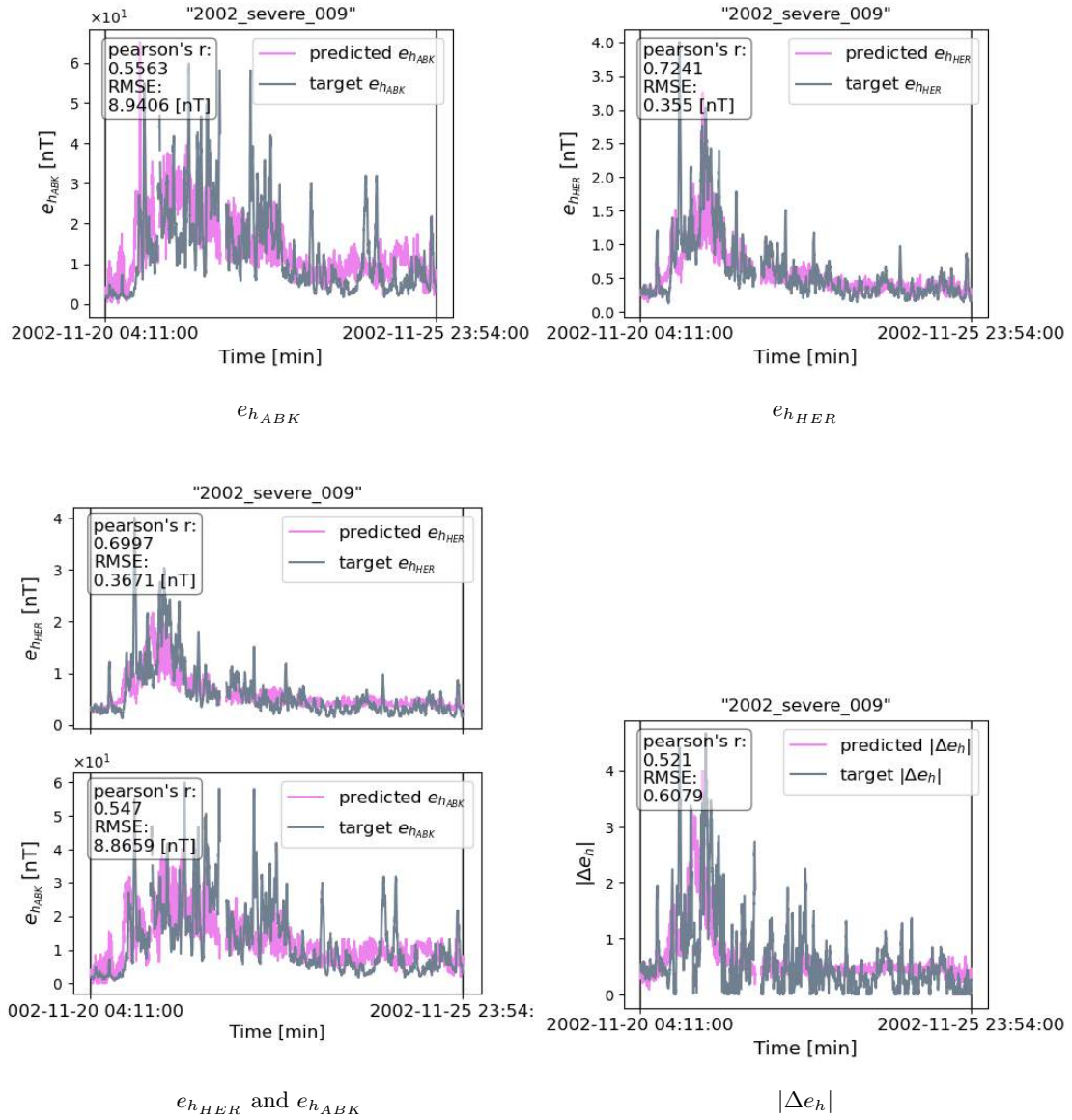
Output plots of 1996_mild_001 for all four tasks, no auto-regressive input features included.



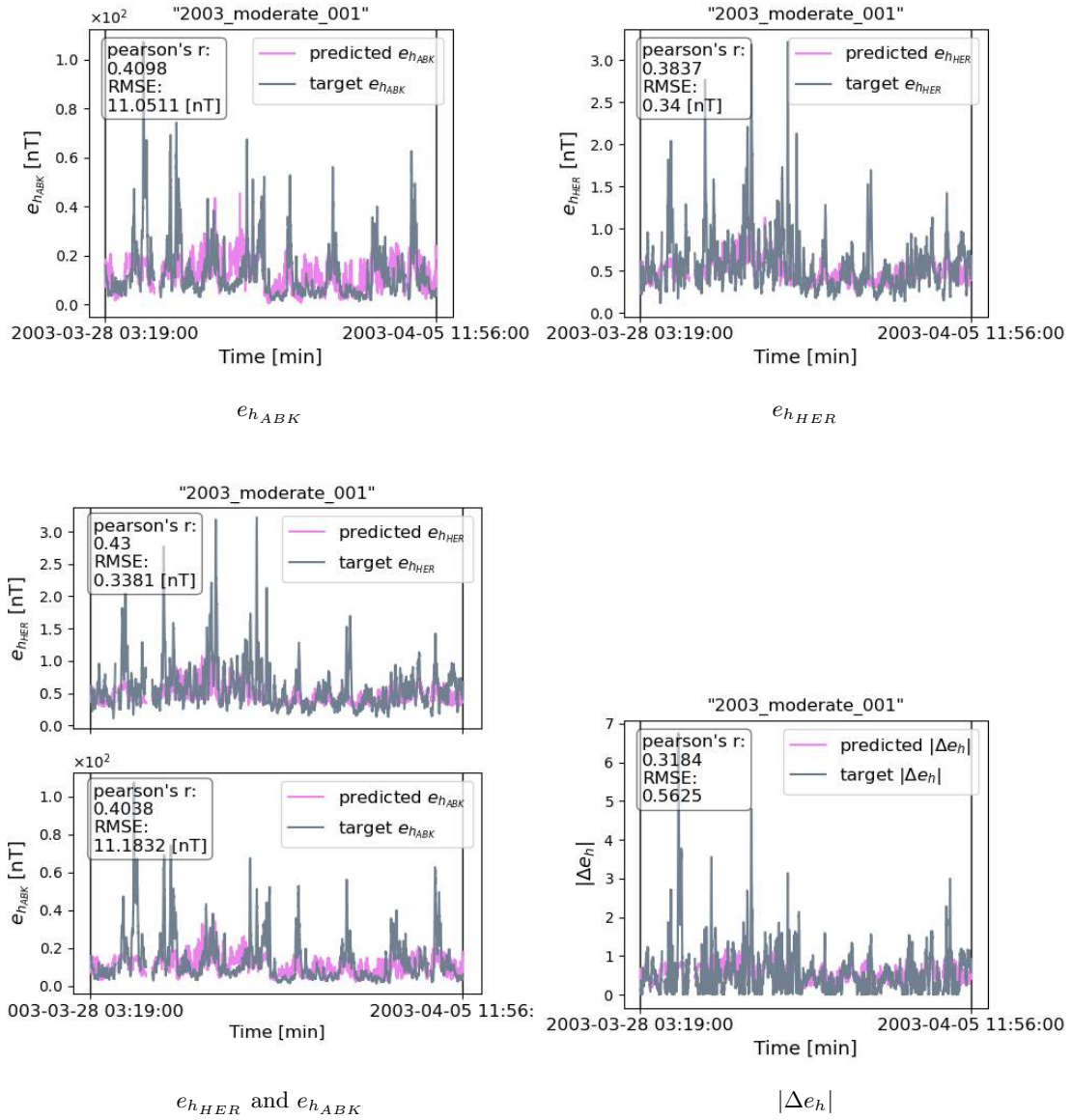
Output plots of 2000.moderate.001 for all four tasks, no auto-regressive input features included.



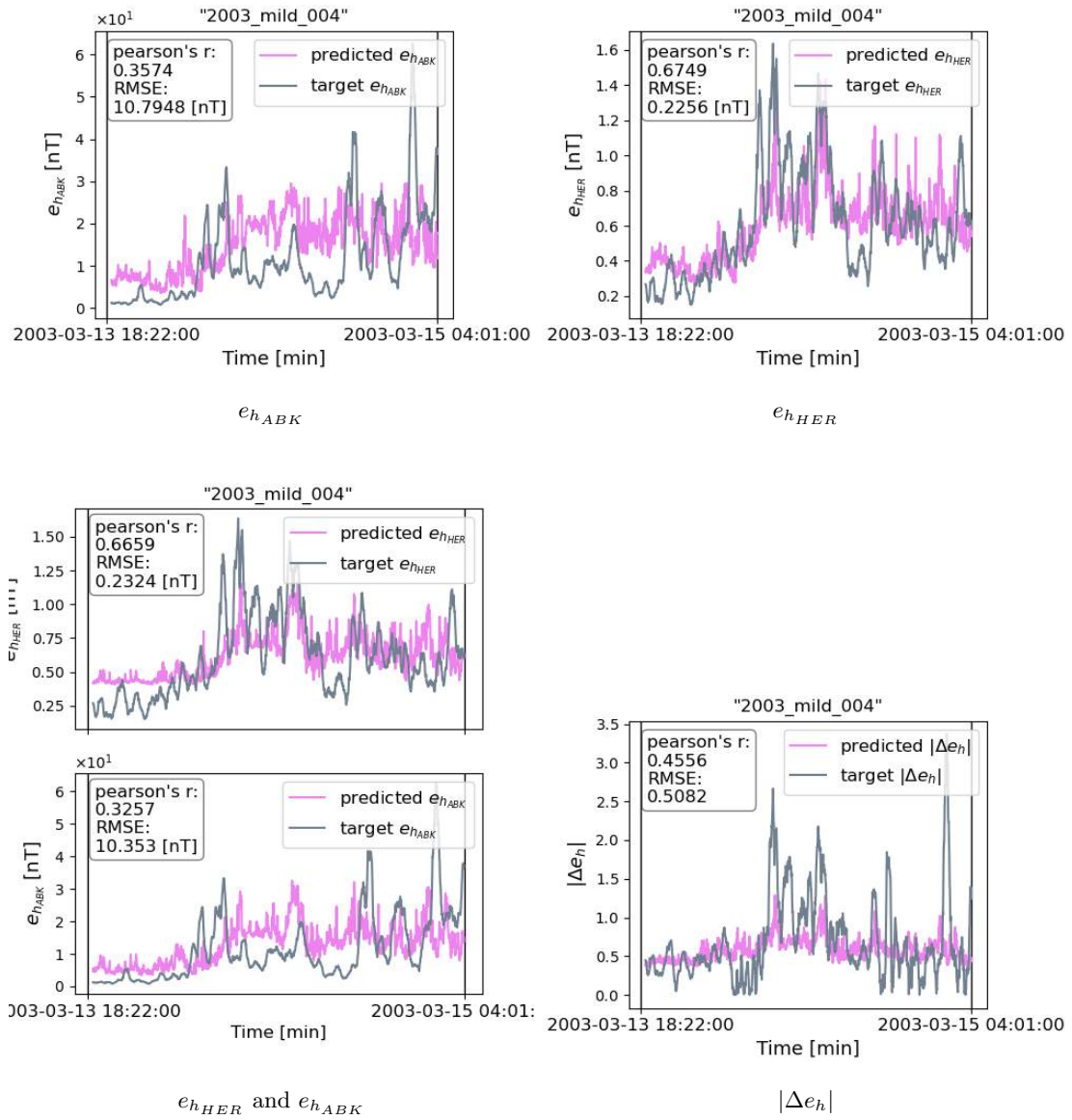
Output plots of 2002_mild.013 for all four tasks, no auto-regressive input features included.



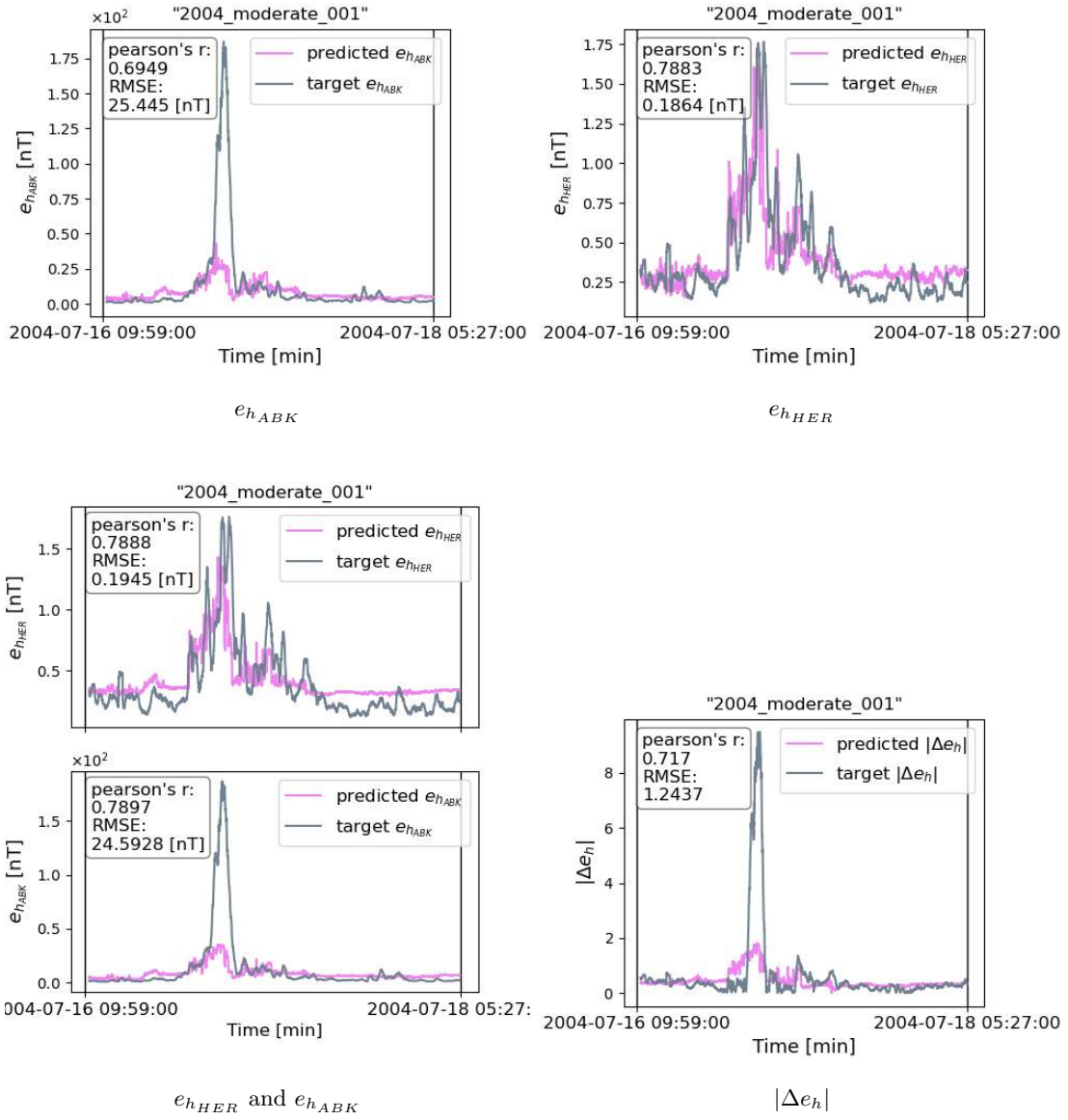
Output plots of 2002_severe_009 for all four tasks, no auto-regressive input features included.



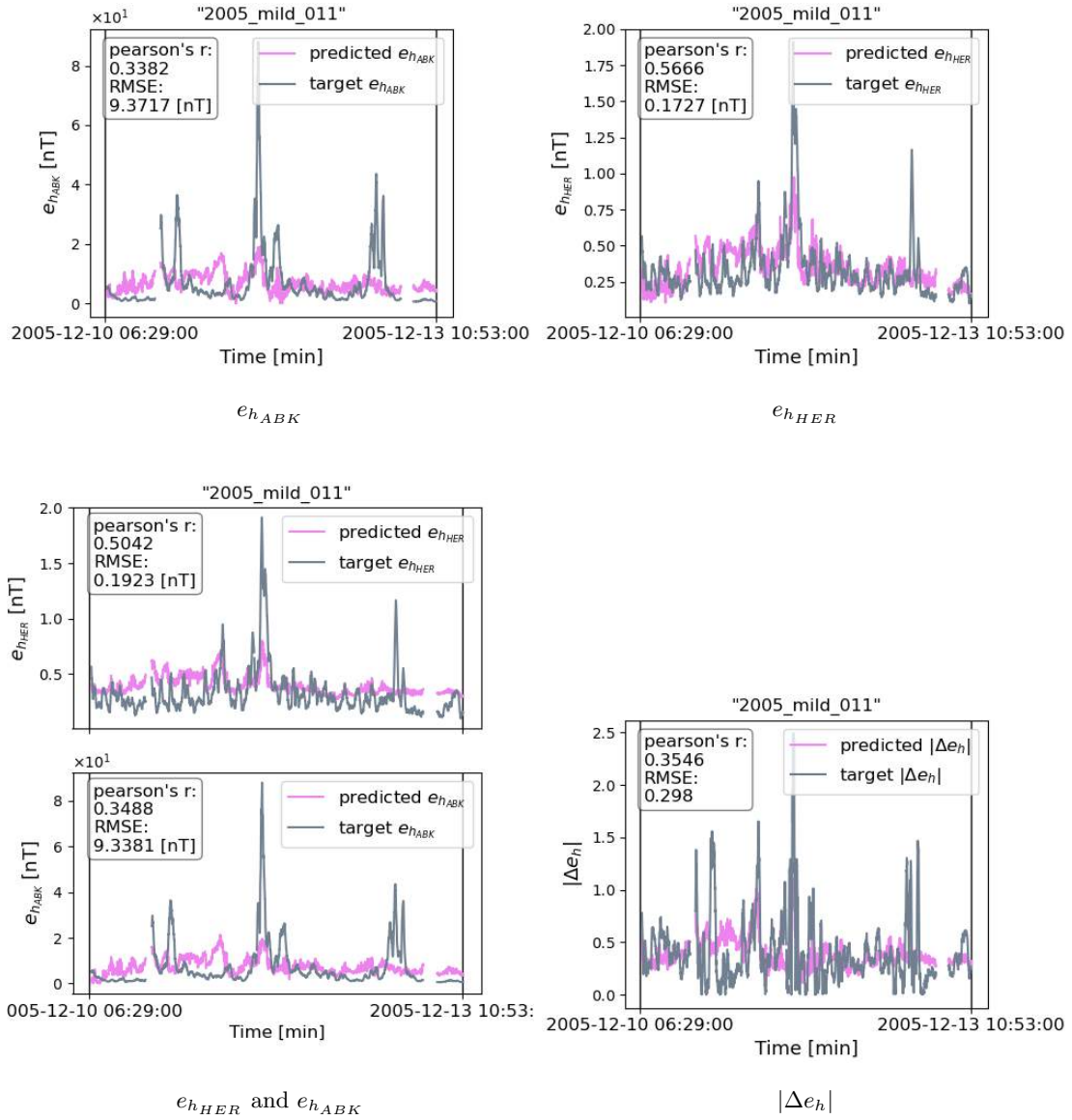
Output plots of 2003.moderate.001 for all four tasks, no auto-regressive input features included.



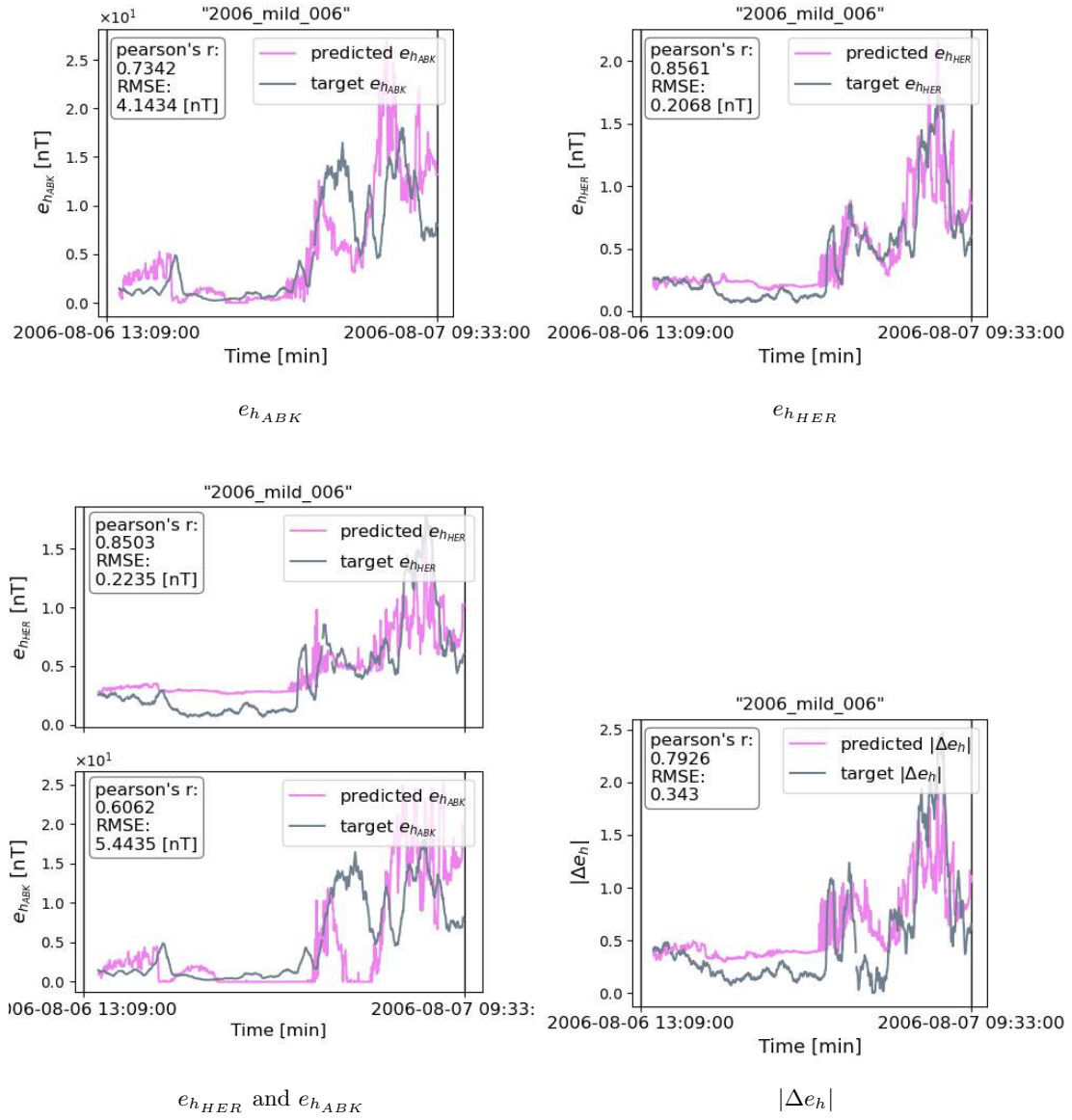
Output plots of 2003_mild.004 for all four tasks, no auto-regressive input features included.



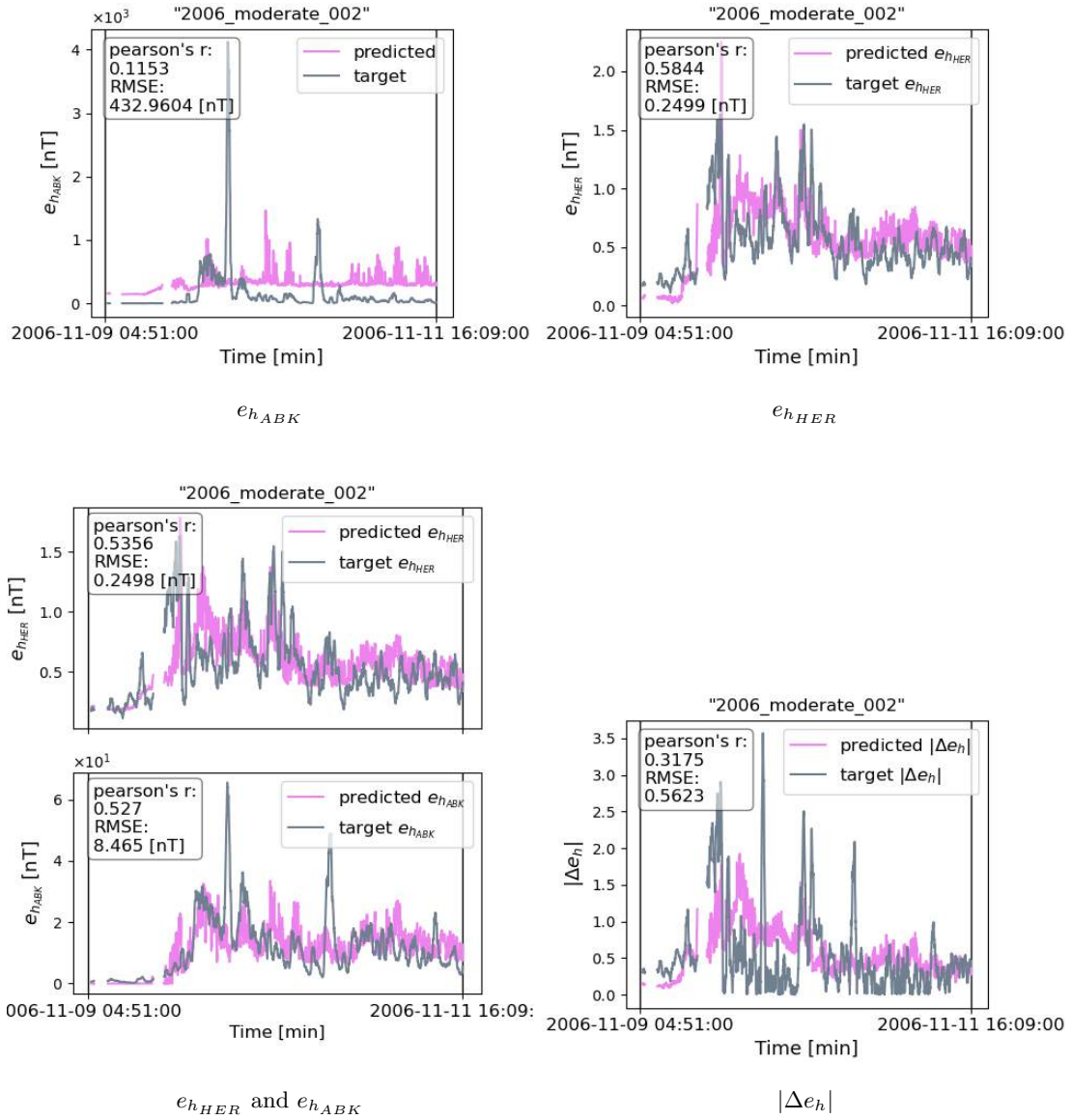
Output plots of 2004.moderate.001 for all four tasks, no auto-regressive input features included.



Output plots of 2005_mild.011 for all four tasks, no auto-regressive input features included.

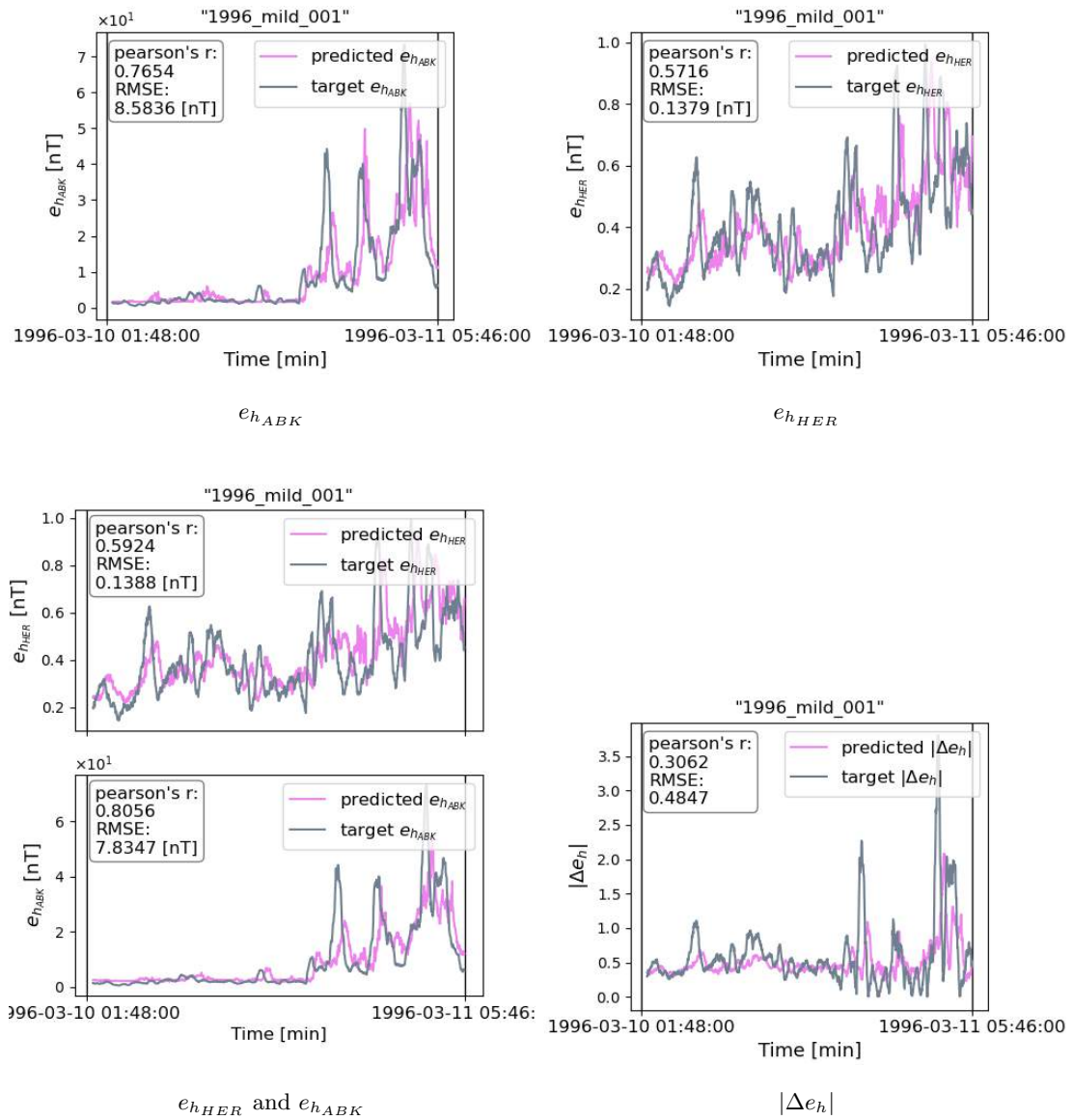


Output plots of 2006_mild.006 for all four tasks, no auto-regressive input features included.

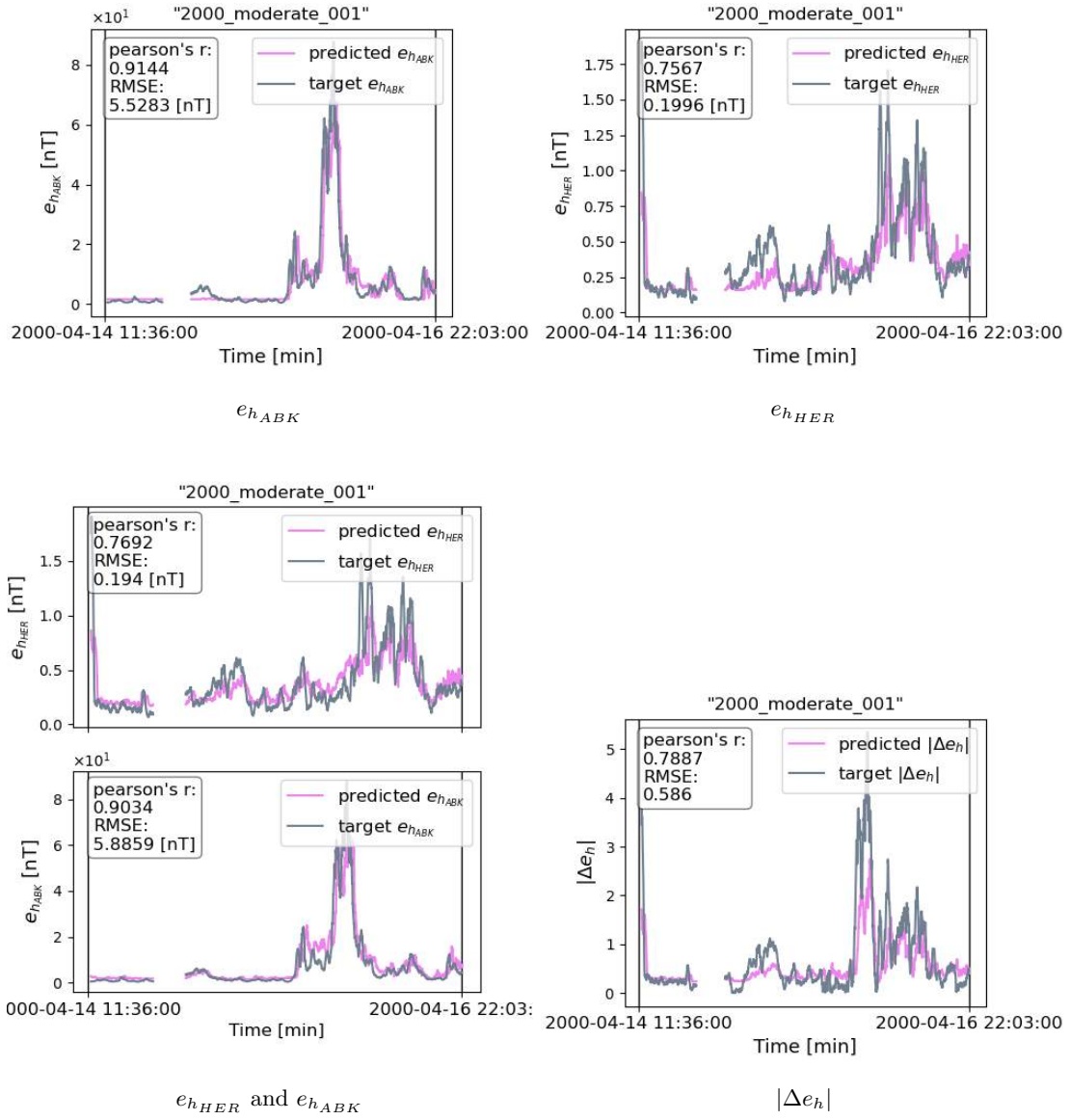


Output plots of 2006.moderate.002 for all four tasks, no auto-regressive input features included.

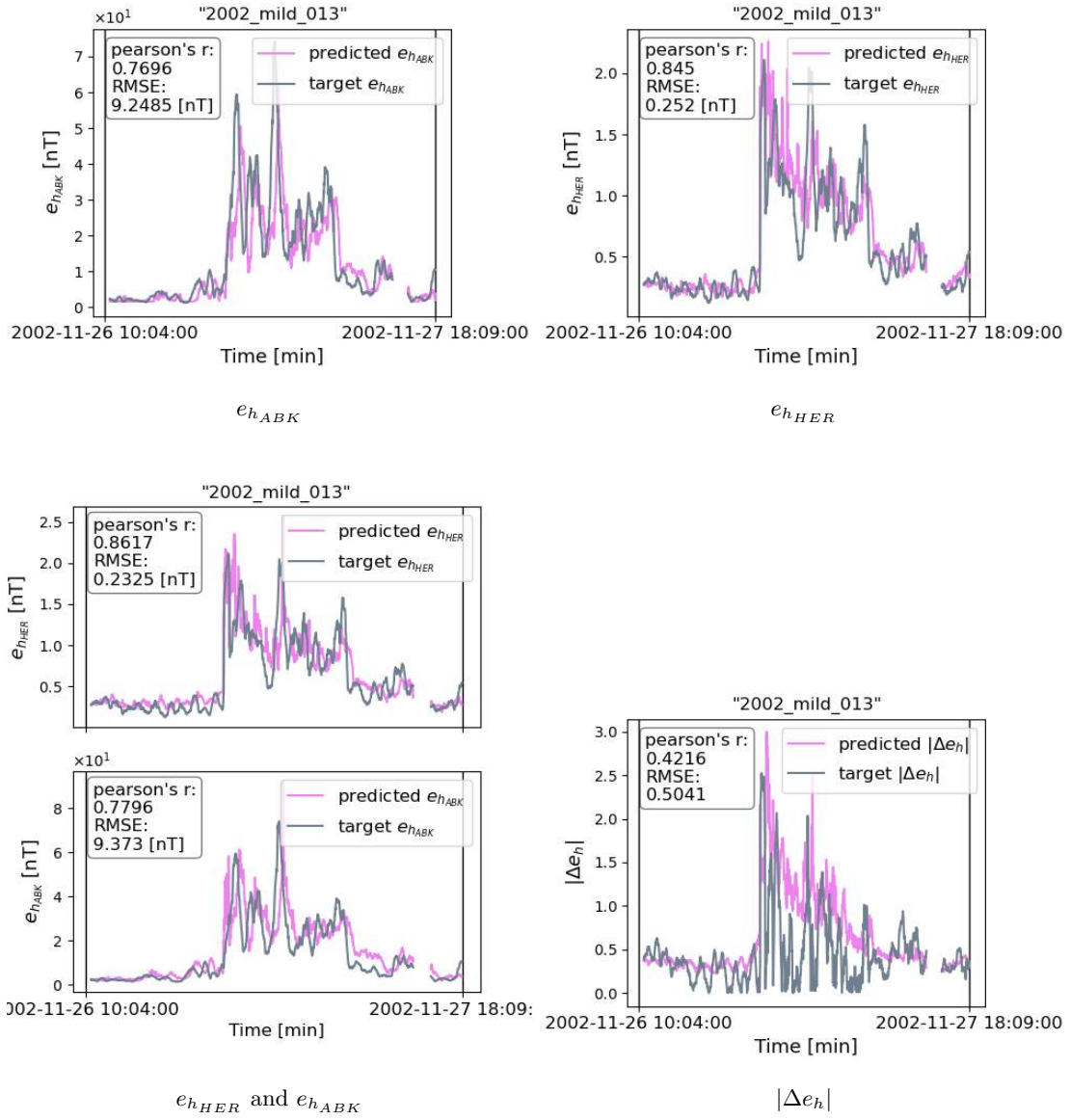
Output plots for CIR-driven storms (auto-regressive input features included)



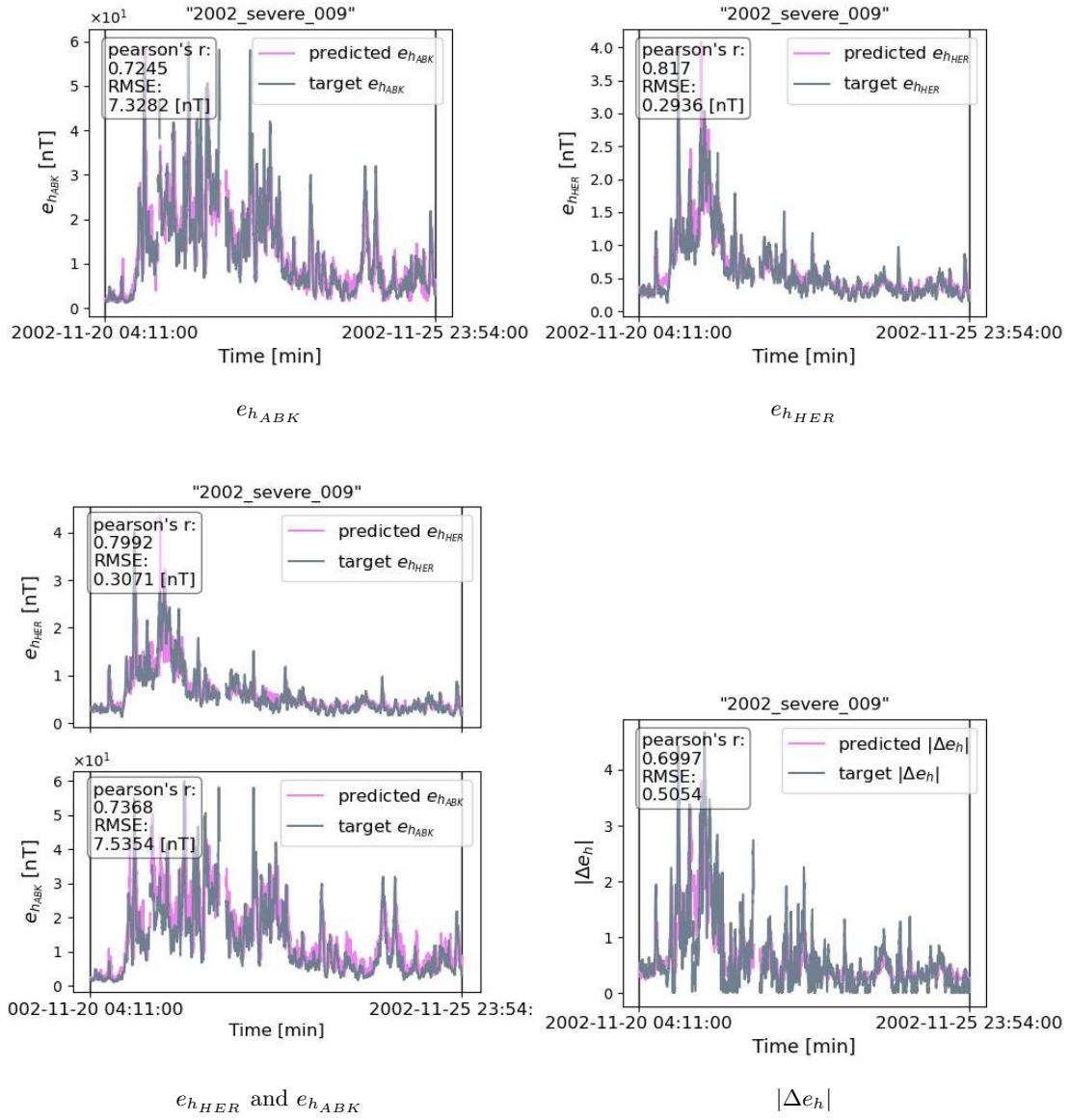
Output plots of 1996_mild_001 for all four tasks, auto-regressive input features included.



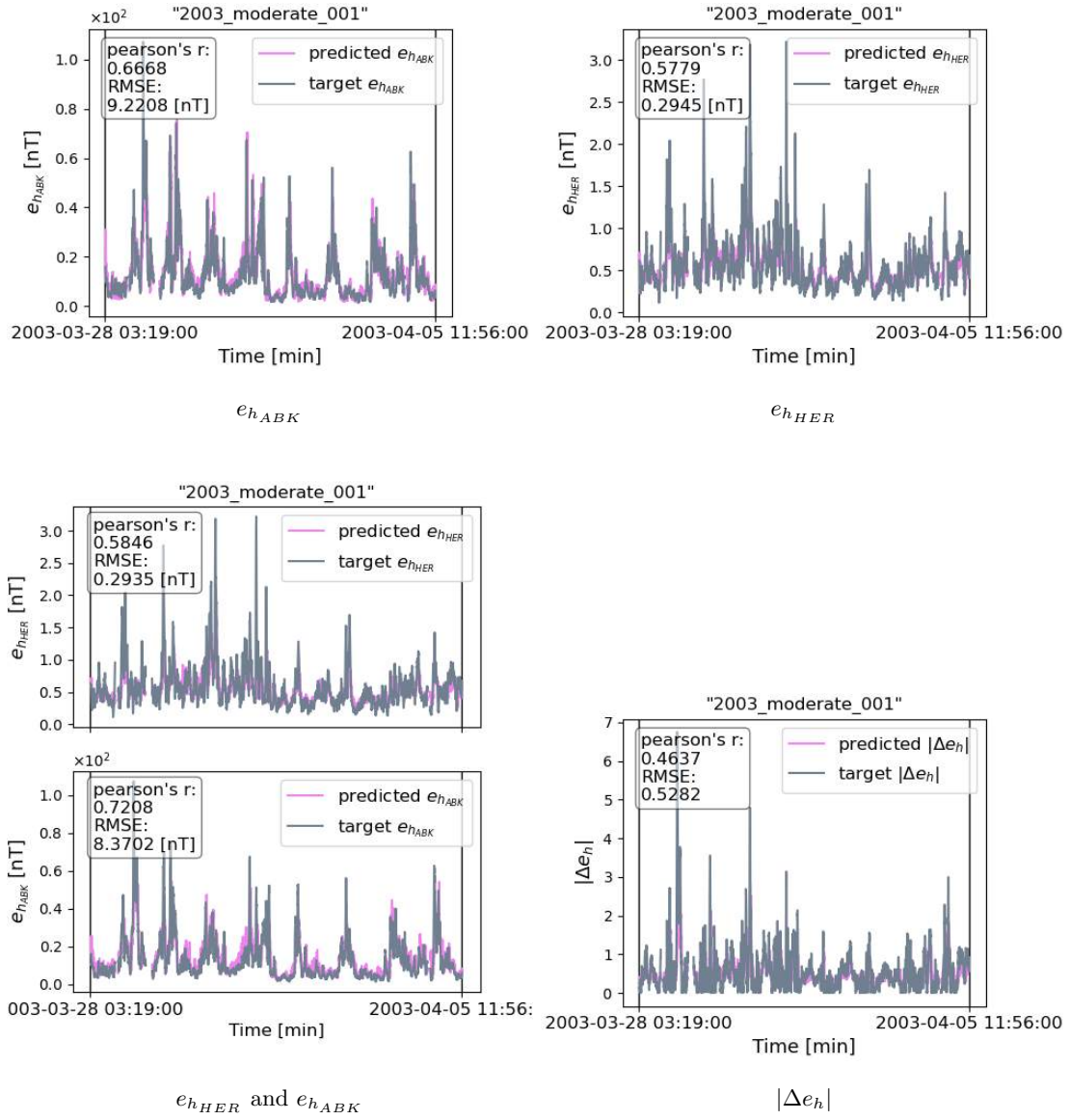
Output plots of 2000.moderate.001 for all four tasks, auto-regressive input features included.



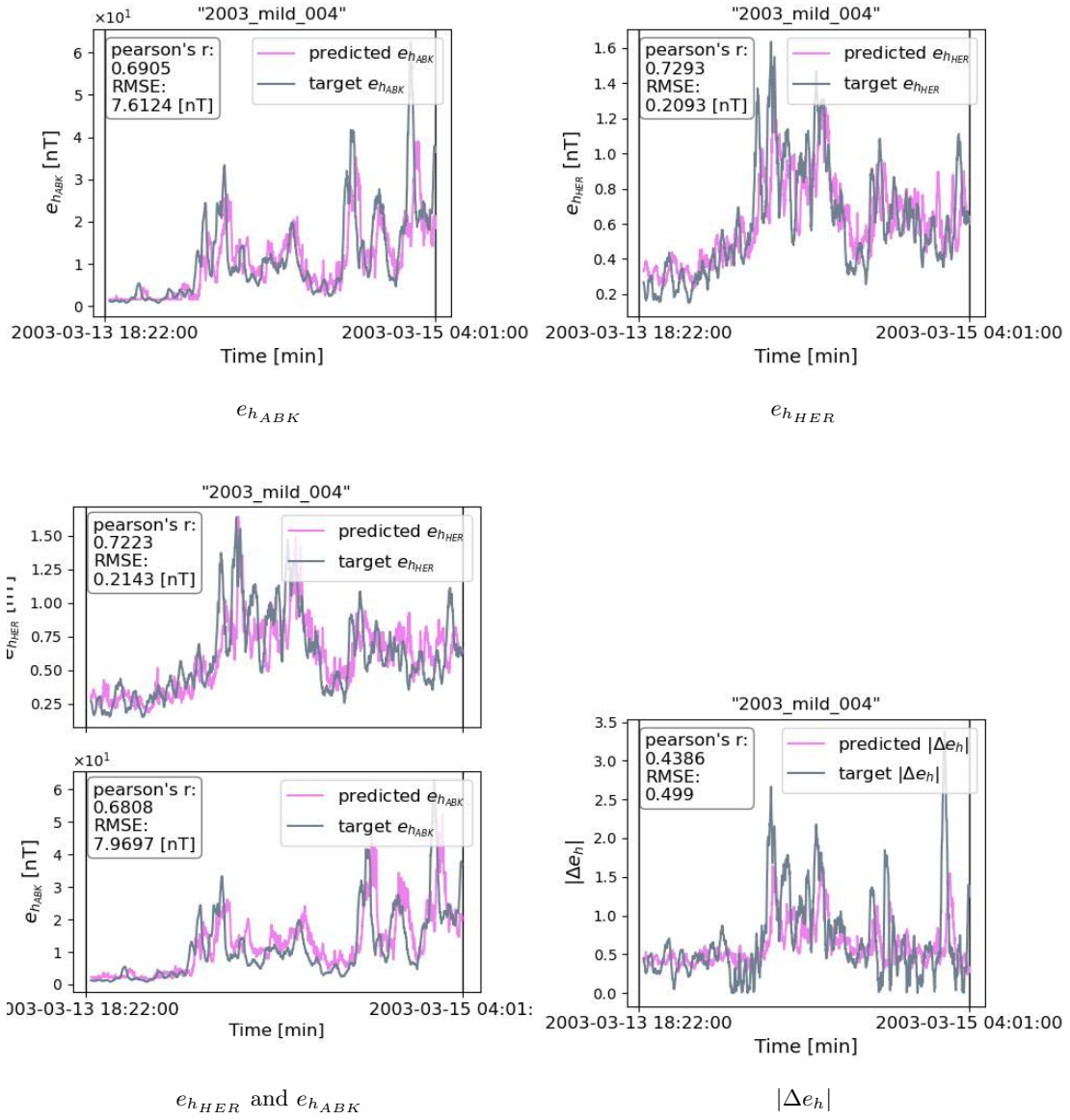
Output plots of 2002_mild.013 for all four tasks, auto-regressive input features included.



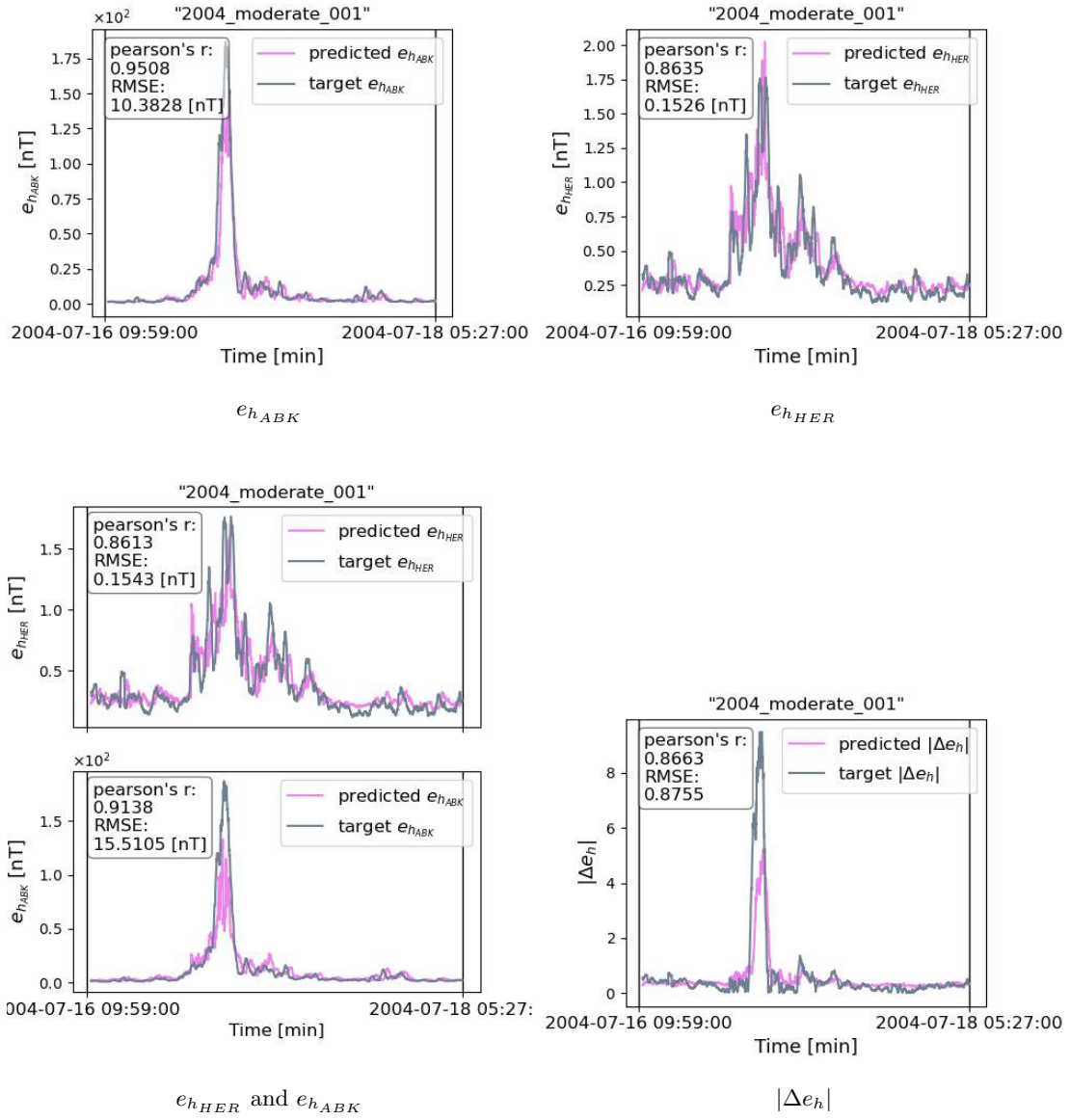
Output plots of 2002_severe_009 for all four tasks, no auto-regressive input features included.



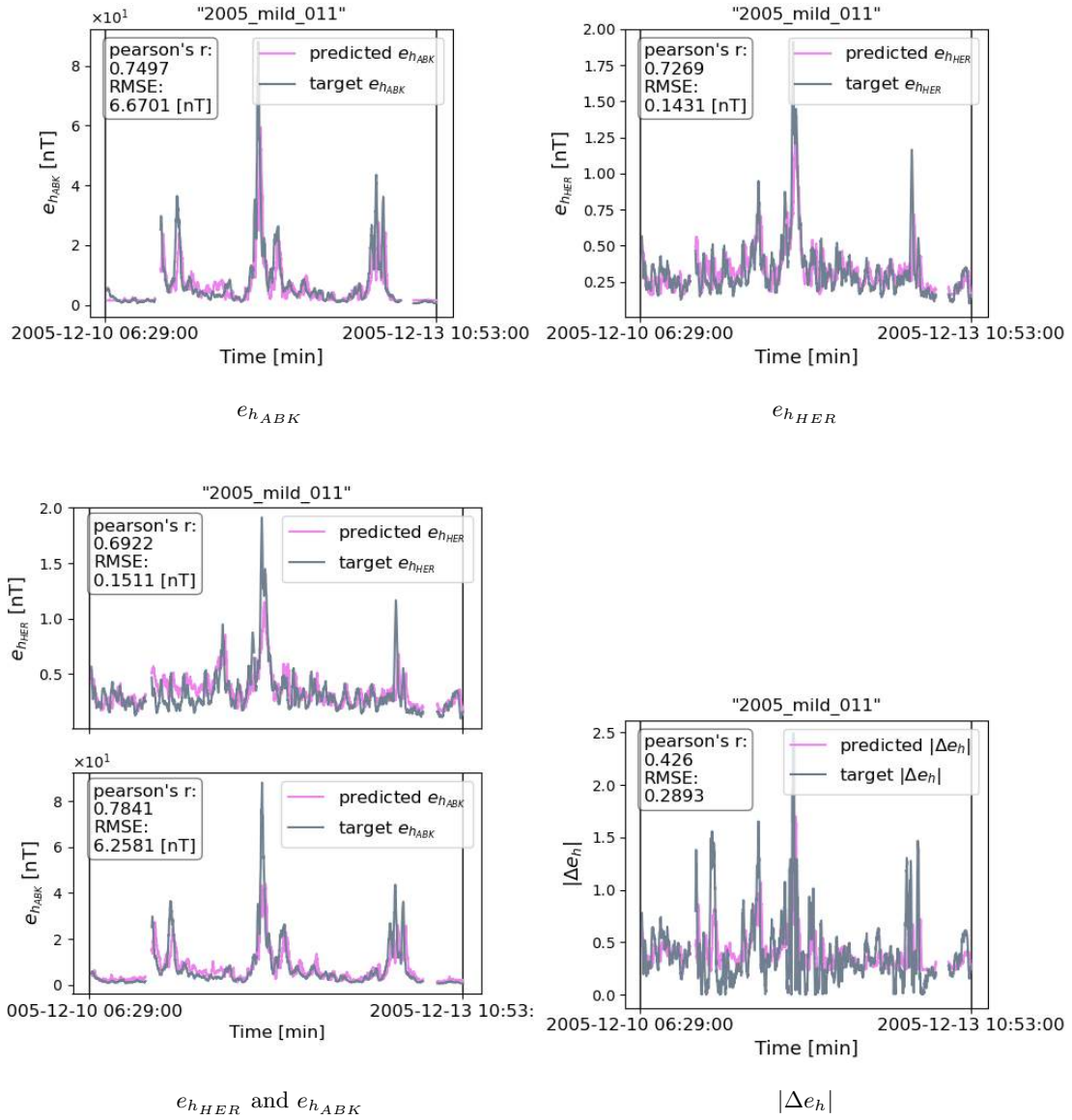
Output plots of 2003.moderate.001 for all four tasks, auto-regressive input features included.



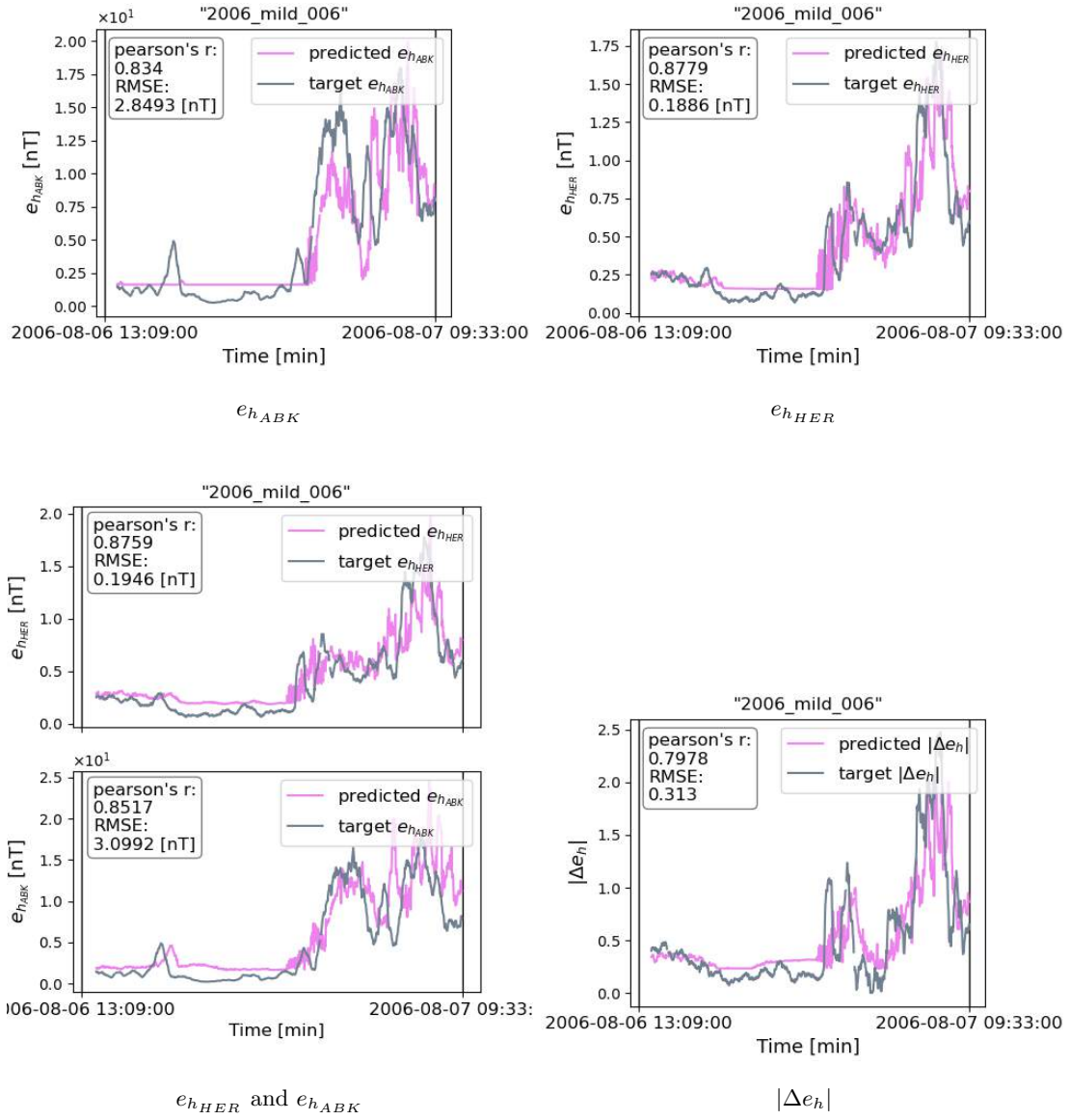
Output plots of 2003_mild.004 for all four tasks, auto-regressive input features included.



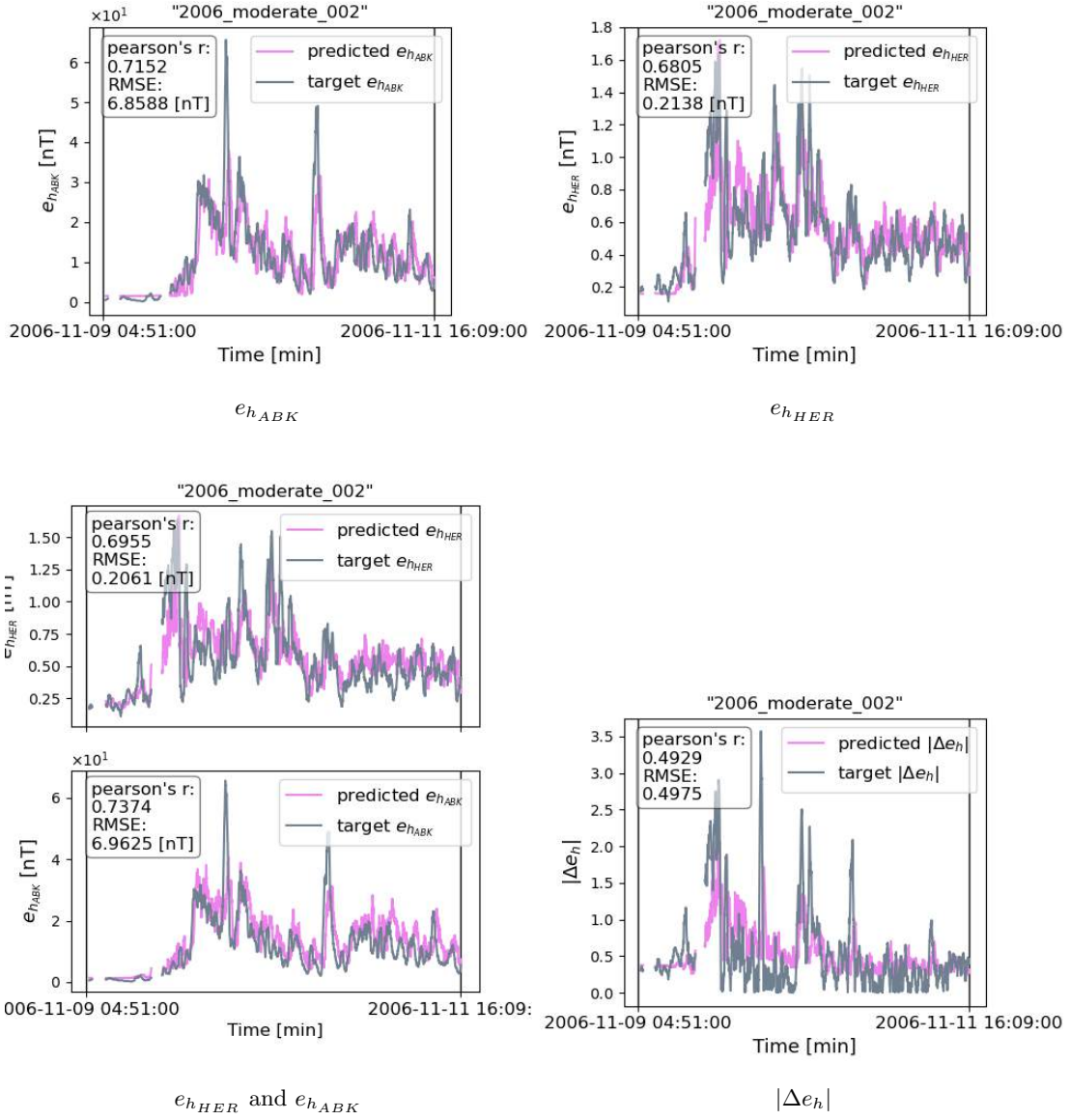
Output plots of 2004.moderate.001 for all four tasks, auto-regressive input features included.



Output plots of 2005_mild.011 for all four tasks, auto-regressive input features included.



Output plots of 2006_mild.006 for all four tasks, auto-regressive input features included.



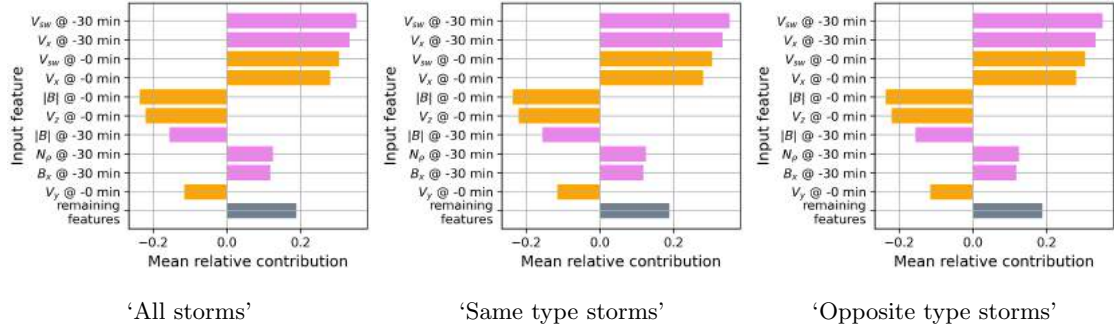
Output plots of 2006_moderate.002 for all four tasks, auto-regressive input features included.

A.4 Appendix: Chapter 6

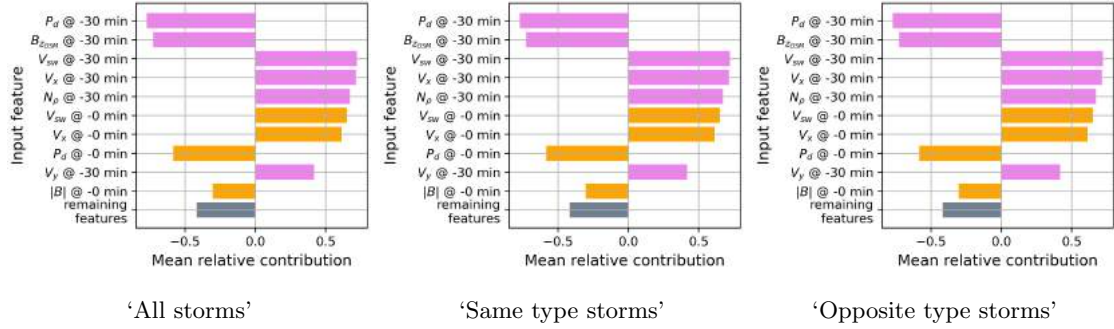
A.4.1 Reference set selection scheme comparisons

The attribution and feature ranking plots for selected mild intensity storms at ABK with different reference set selection schemes as described in Section 6.3.3 are shown here.

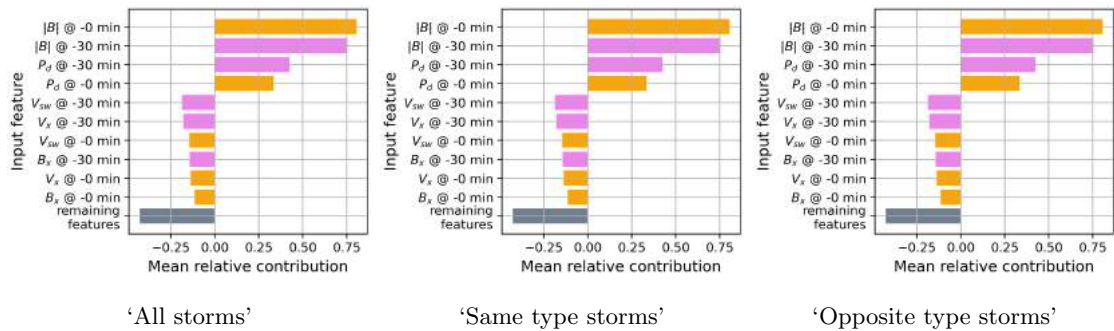
Feature ranking plots



Feature ranking plots of all non auto-regressive input features for e_{hABK} predicted during 2000_mild.009 with varying reference set selection schemes.

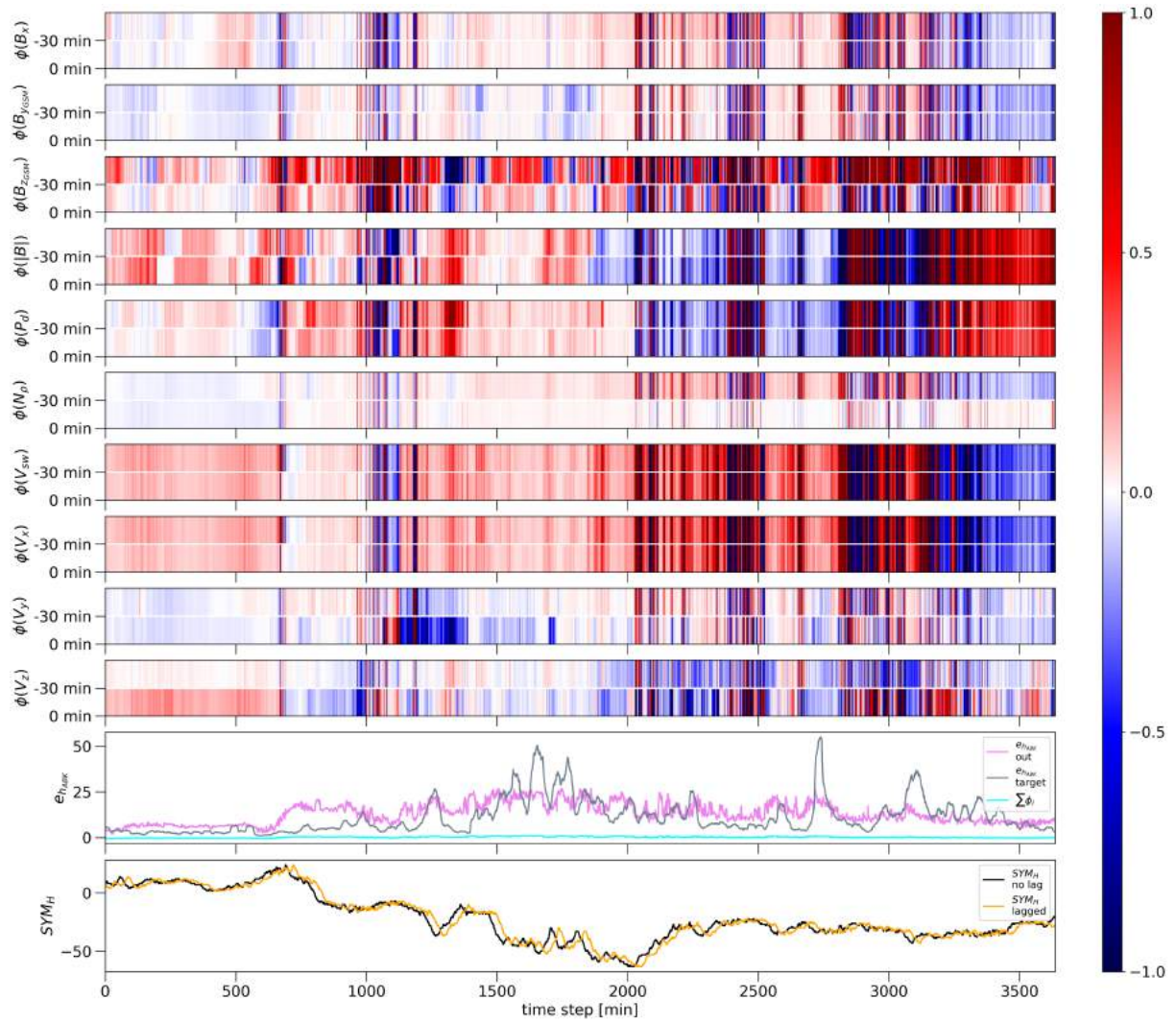


Feature ranking plots of all non auto-regressive input features for e_{hABK} predicted during 2005_mild.004 with varying reference set selection schemes.

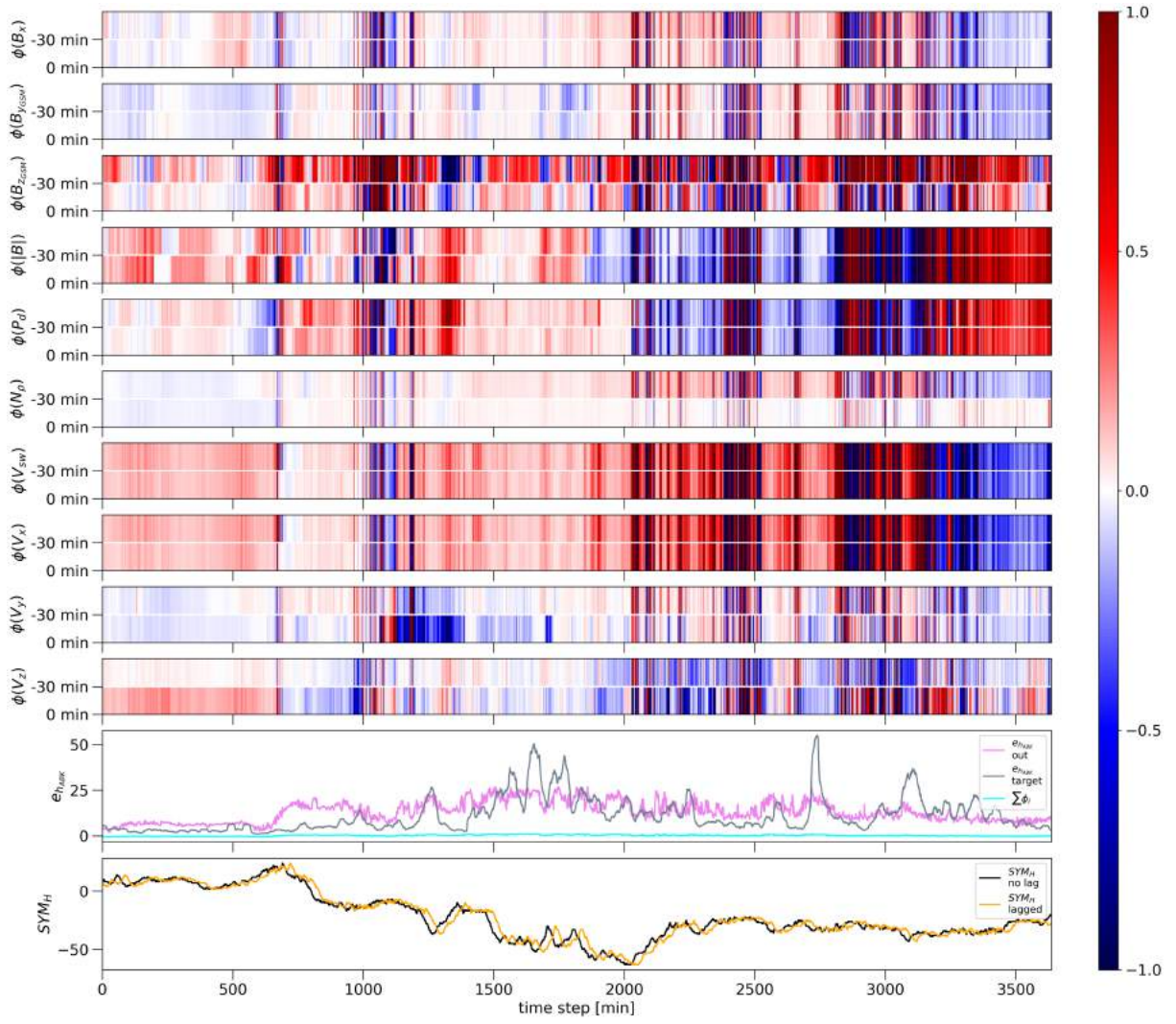


Feature ranking plots of all non auto-regressive input features for e_{hABK} predicted during 2019_mild.003 with varying reference set selection schemes.

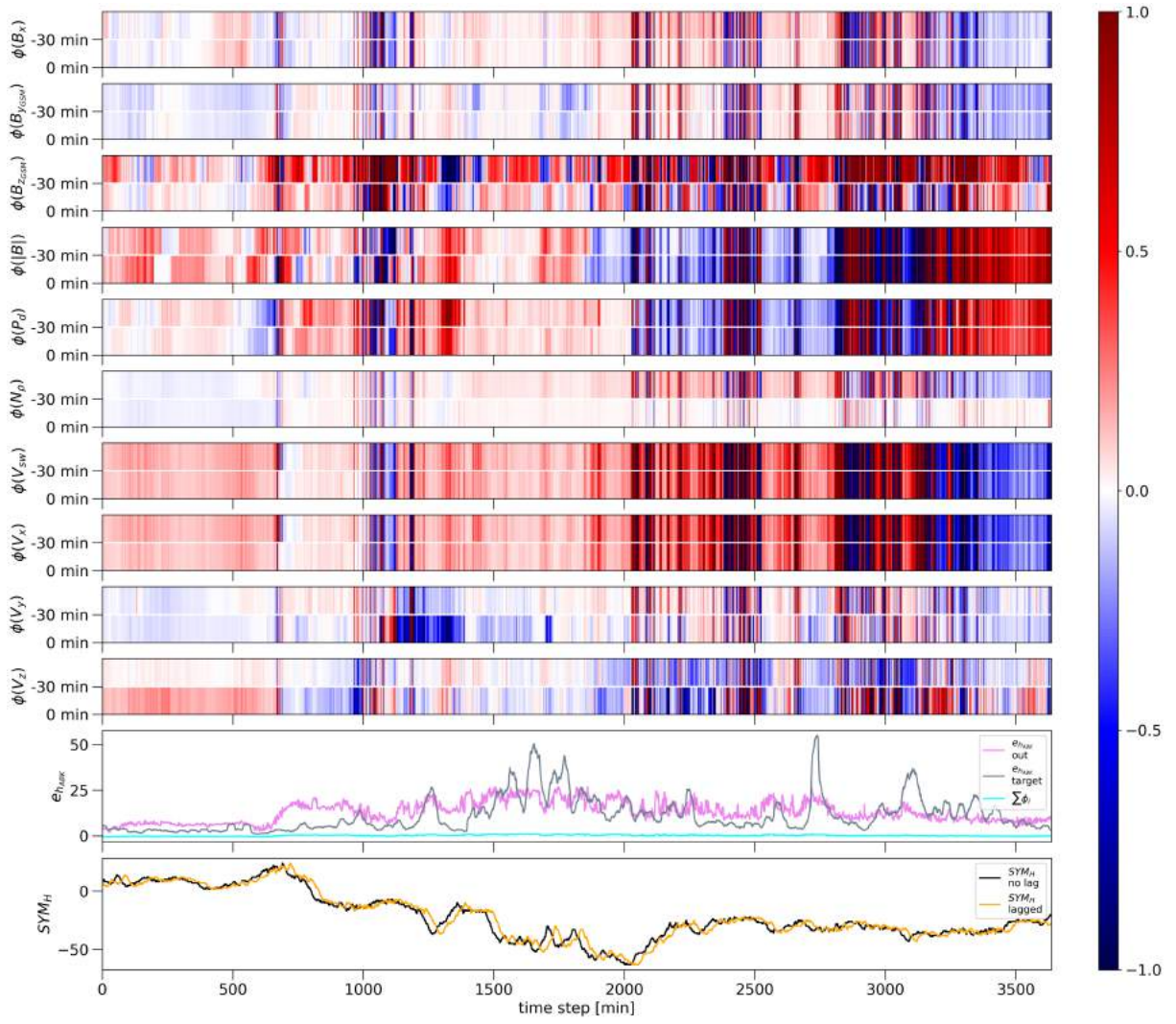
Attribution plots



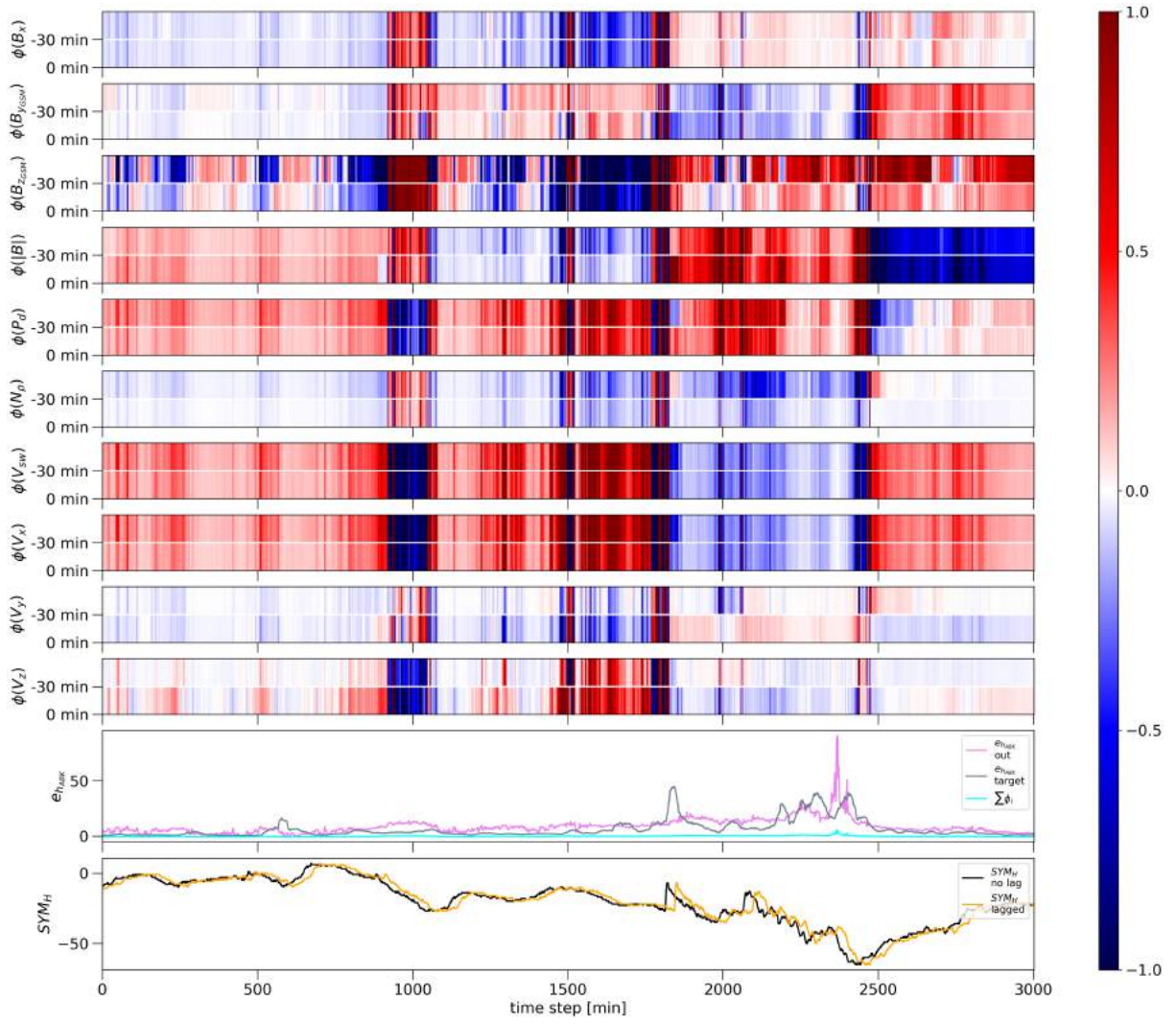
Attribution plot of all non auto-regressive input features for e_{hABK} predicted during 2000_mild_009 with attributions generated using an ‘all storms’ reference set selection scheme.



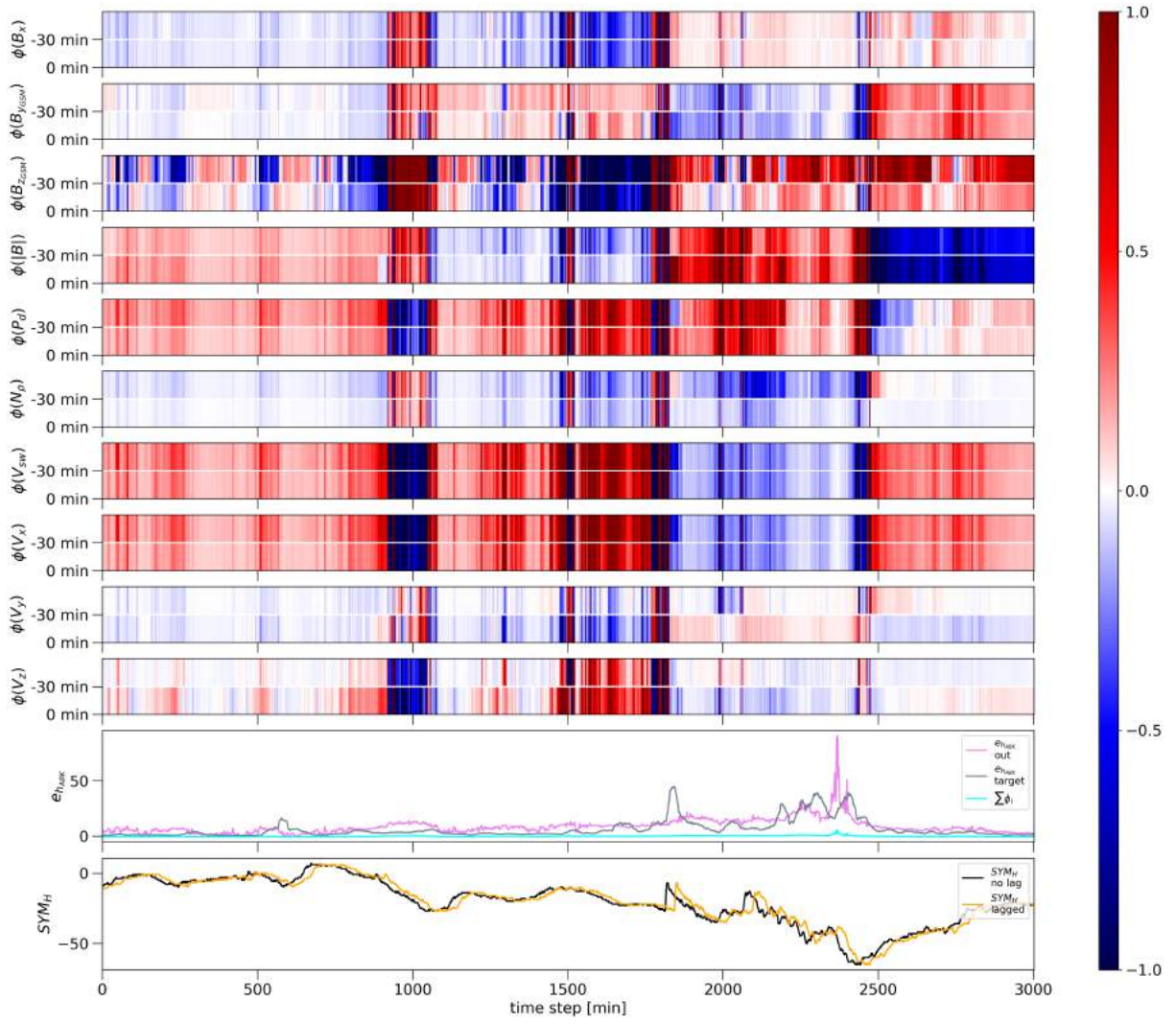
Attribution plot of all non auto-regressive input features for e_{hABK} predicted during 2000_mild_009 with attributions generated using a ‘same type storms’ reference set selection scheme.



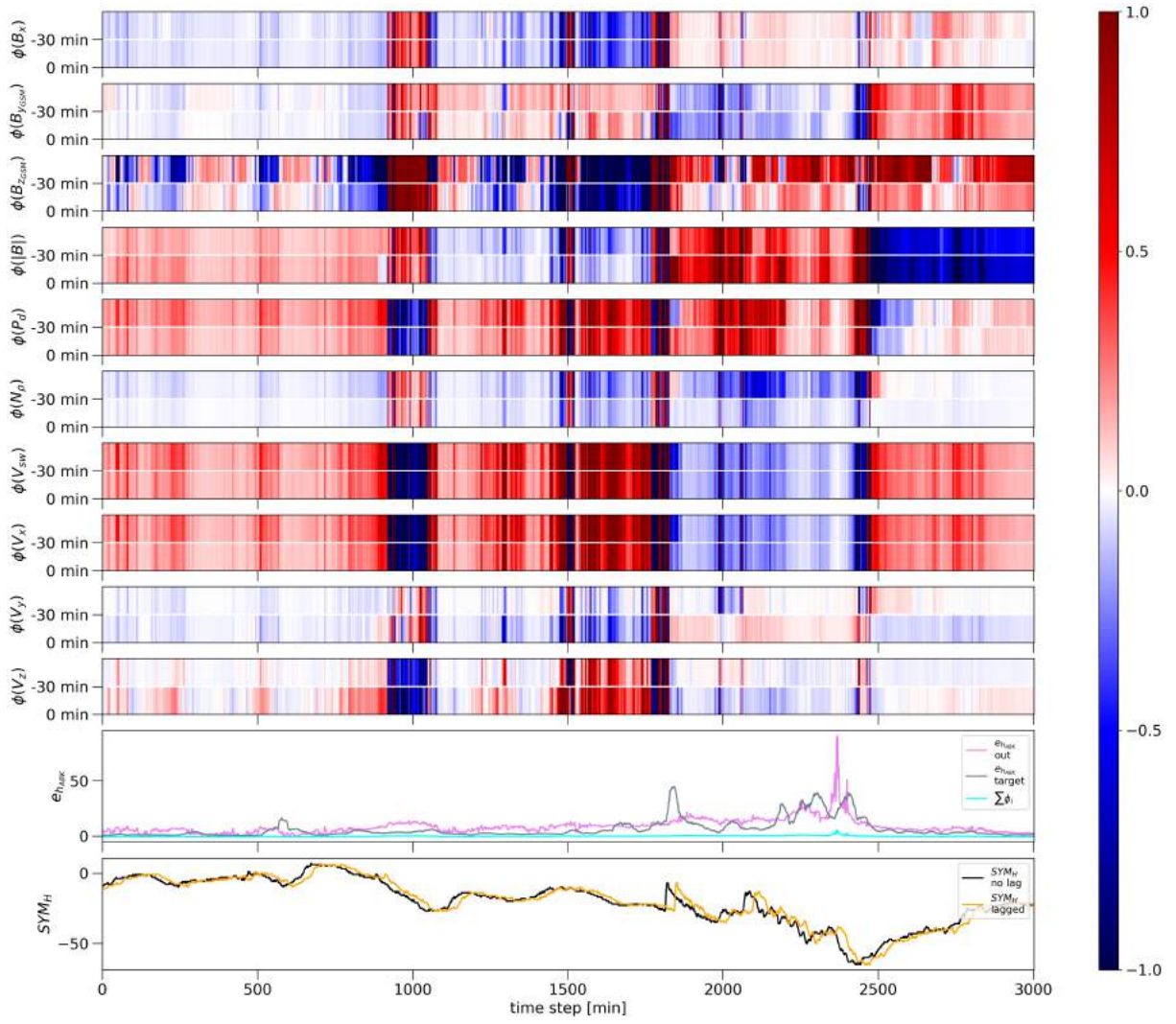
Attribution plot of all non auto-regressive input features for e_{hABK} predicted during 2000_mild_009 with attributions generated using an ‘opposite type storms’ reference set selection scheme.



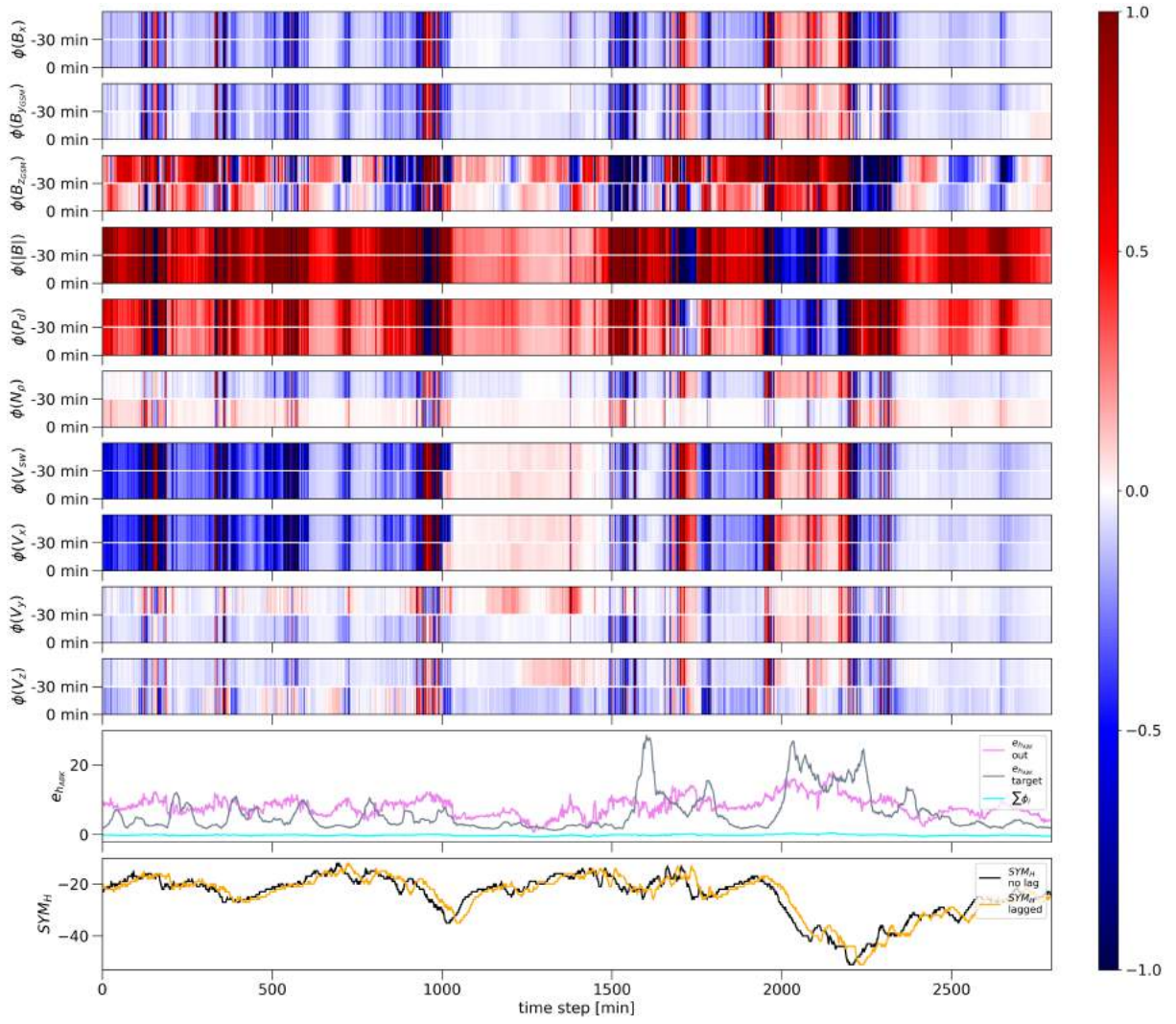
Attribution plot of all non auto-regressive input features for $e_{h_{ABK}}$ predicted during 2014_mild_010 with attributions generated using an ‘all storms’ reference set selection scheme.



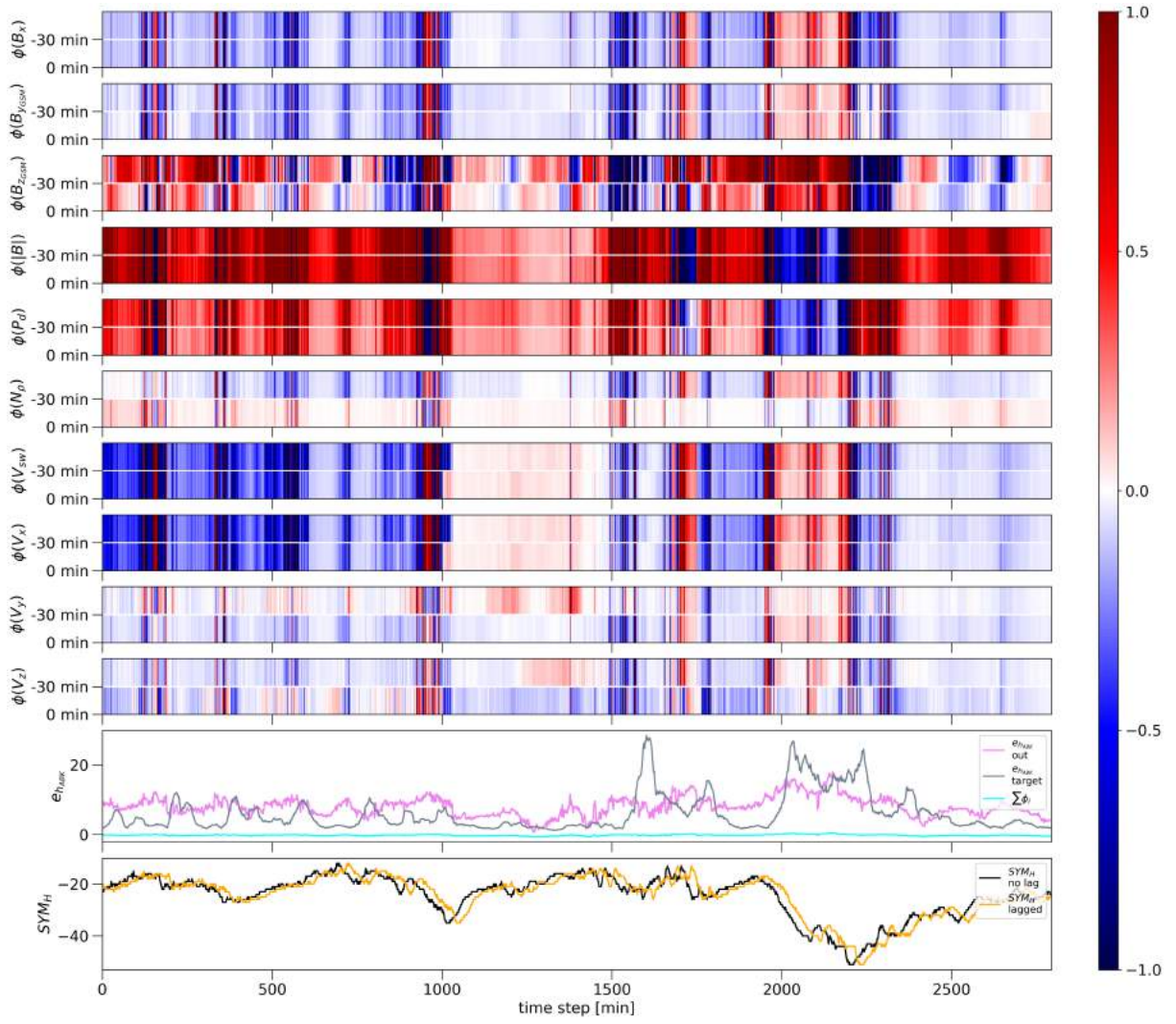
Attribution plot of all non auto-regressive input features for e_{hABK} predicted during 2014_mild_010 with attributions generated using a ‘same type storms’ reference set selection scheme.



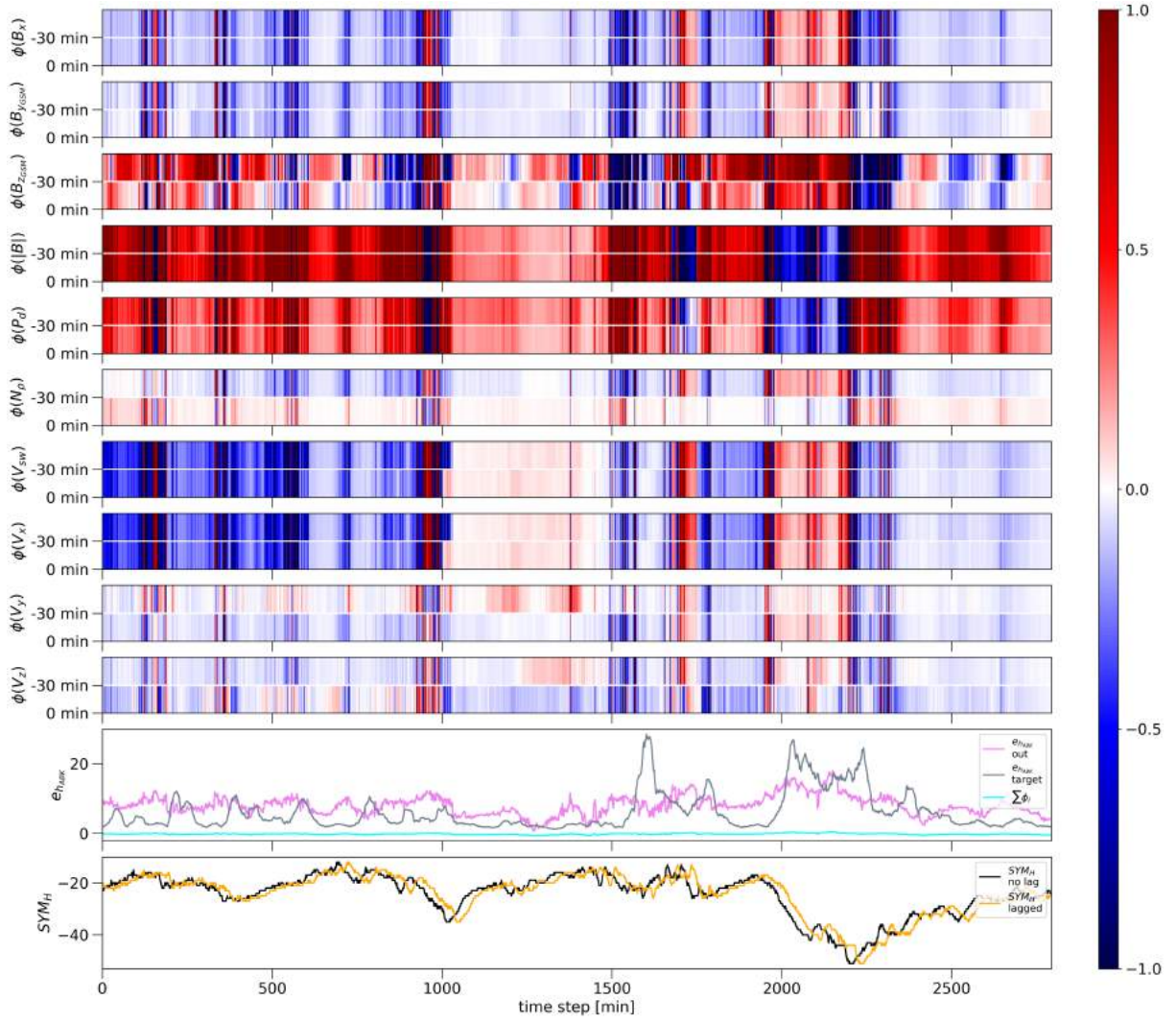
Attribution plot of all non auto-regressive input features for e_{hABK} predicted during 2014_mild.010 with attributions generated using an ‘opposite type storms’ reference set selection scheme.



Attribution plot of all non auto-regressive input features for e_{hABK} predicted during 2019_mild.003 with attributions generated using an ‘all storms’ reference set selection scheme.



Attribution plot of all non auto-regressive input features for e_{hABK} predicted during 2019_mild.003 with attributions generated using a 'same type storms' reference set selection scheme.

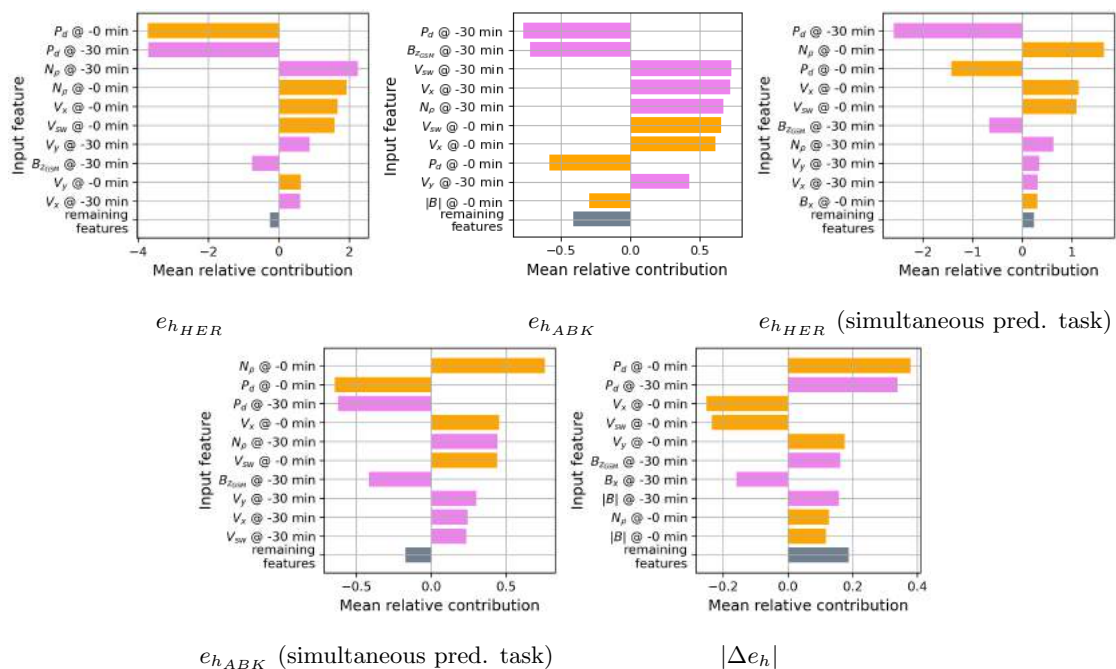
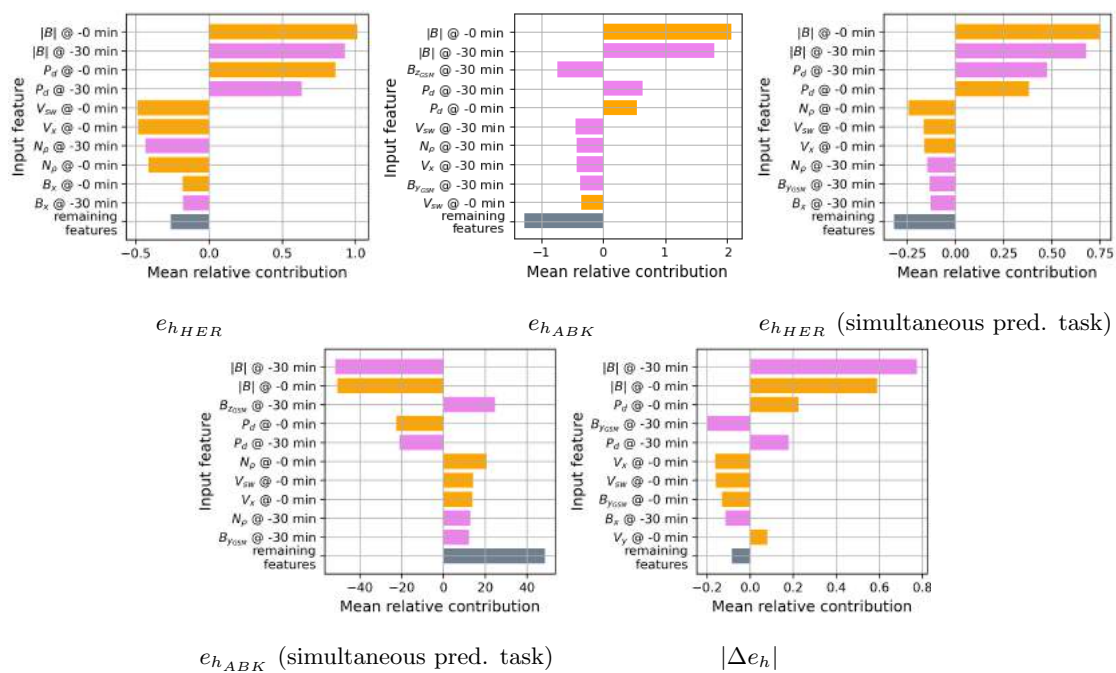


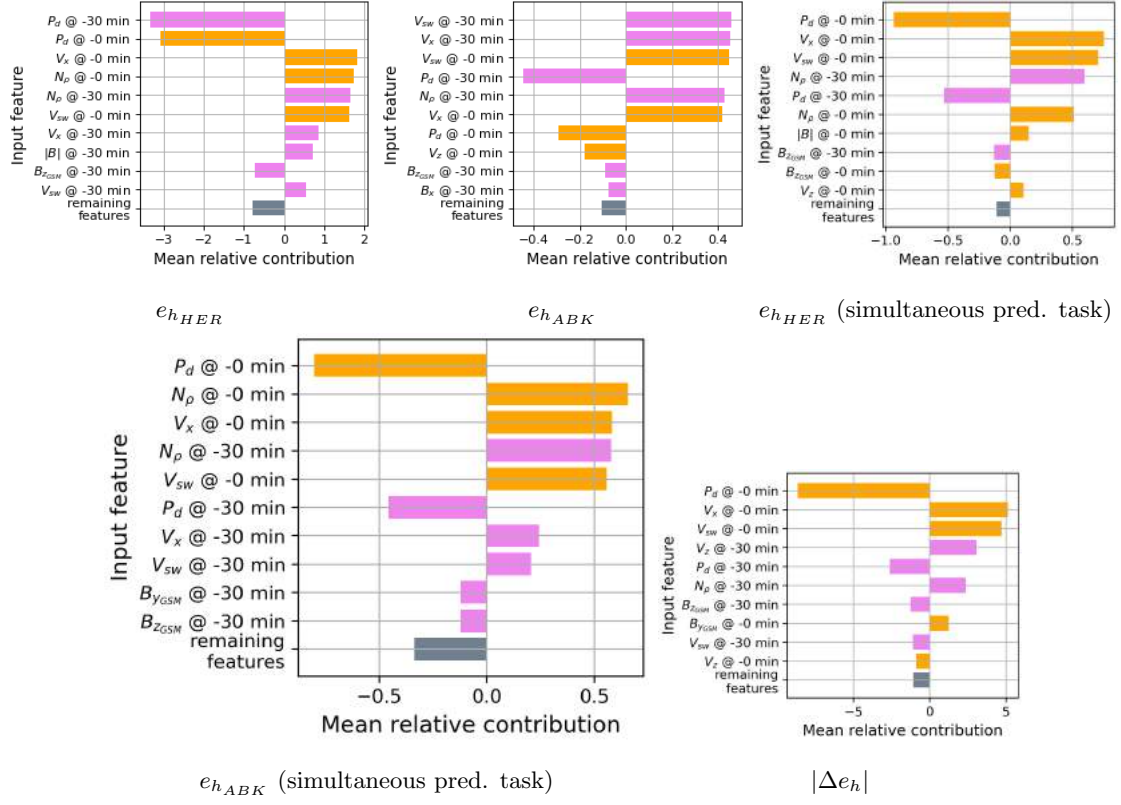
Attribution plot of all non auto-regressive input features for e_{hABK} predicted during 2019_mild.003 with attributions generated using an ‘opposite type storms’ reference set selection scheme.

A.4.2 Supplemental CIR-driven storm attributions

Here we display the feature ranking plots and attribution plots for each of the individual selected CIR-driven storms.

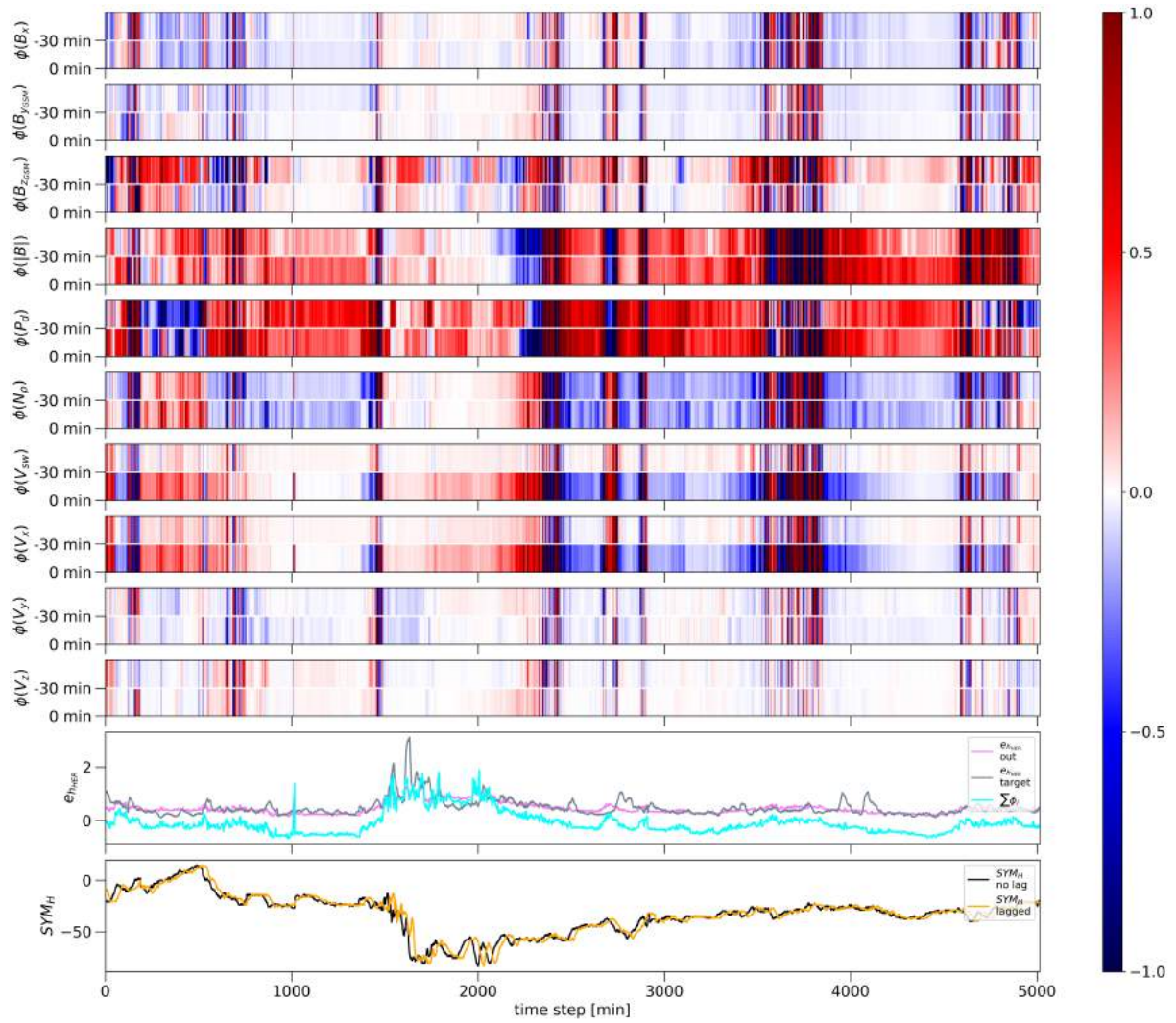
Feature ranking plots



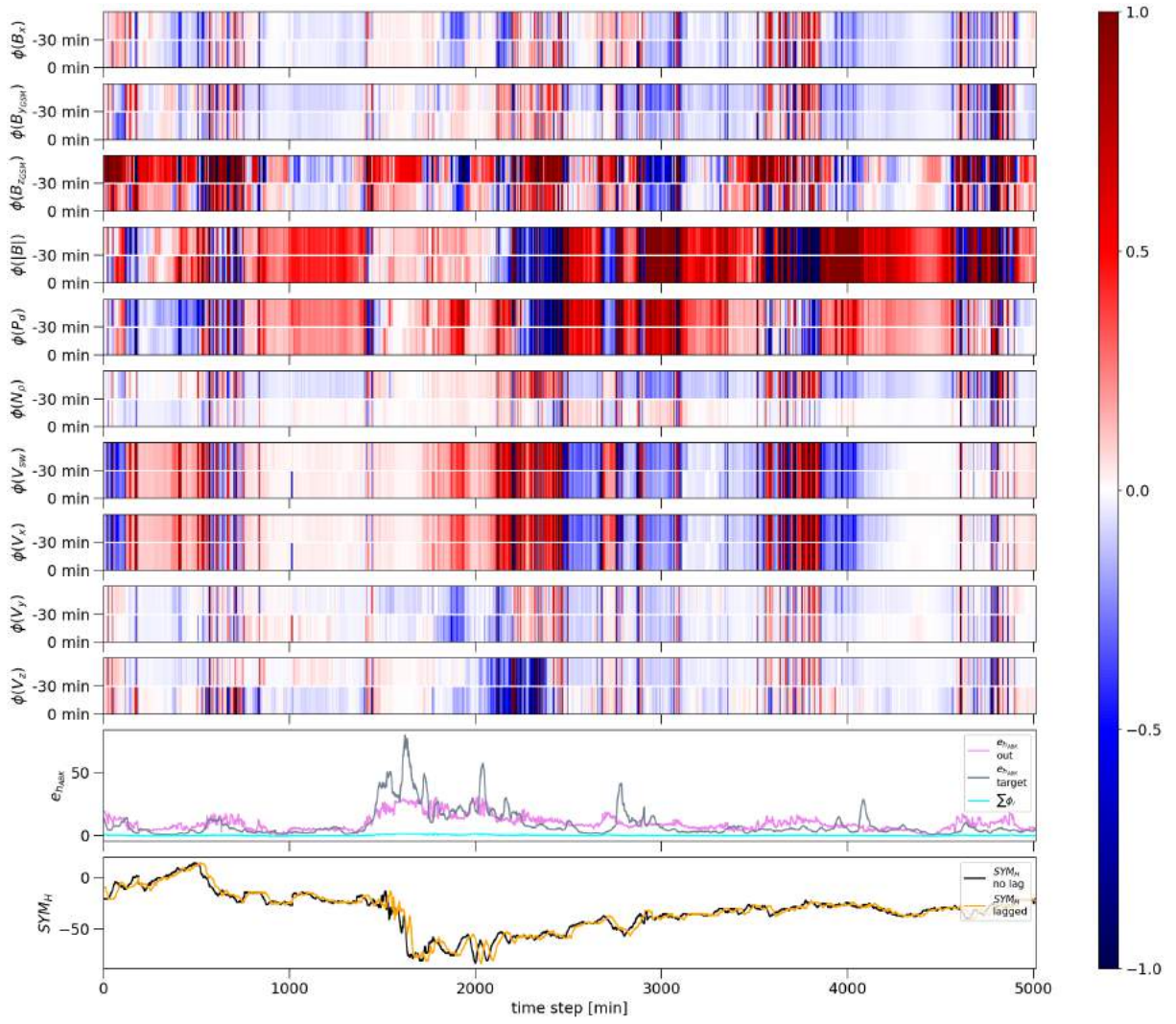


Feature ranking plots for all tasks during storm 2005_severe.010.

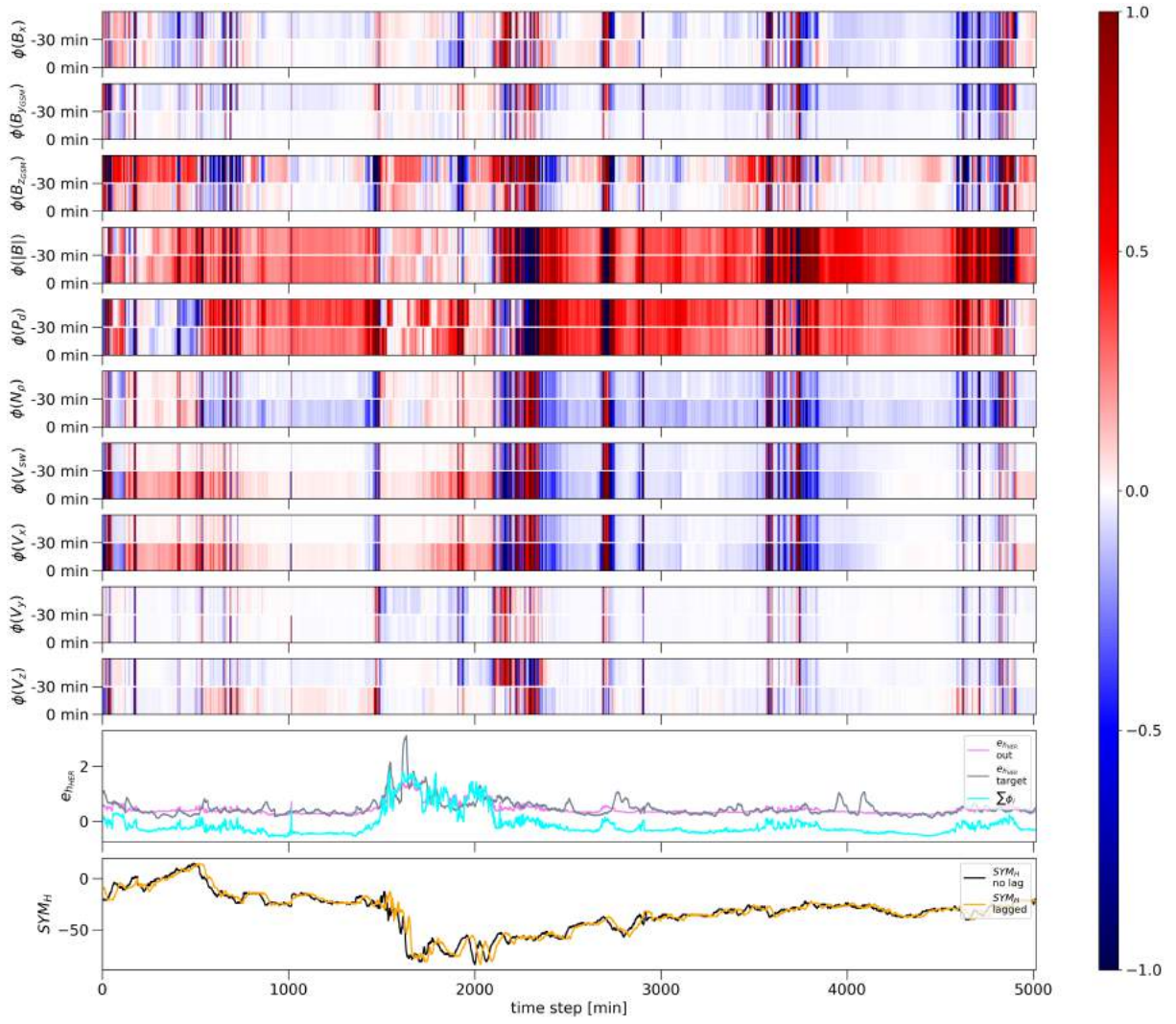
Attribution plots



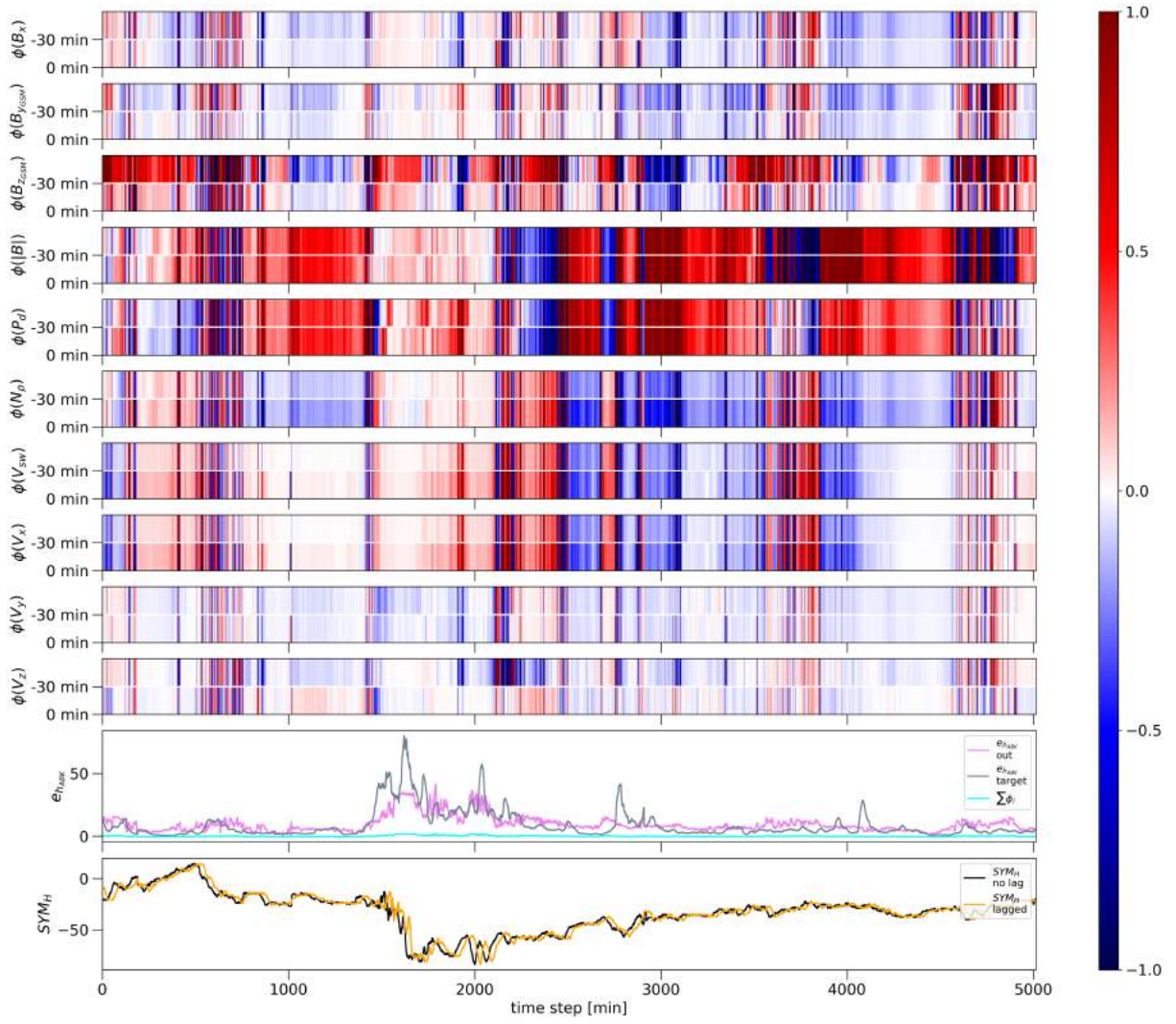
Attribution plot of all non auto-regressive input features for $e_{h_{HER}}$ predicted during 1997_moderate.001 in the individual e_h prediction task.



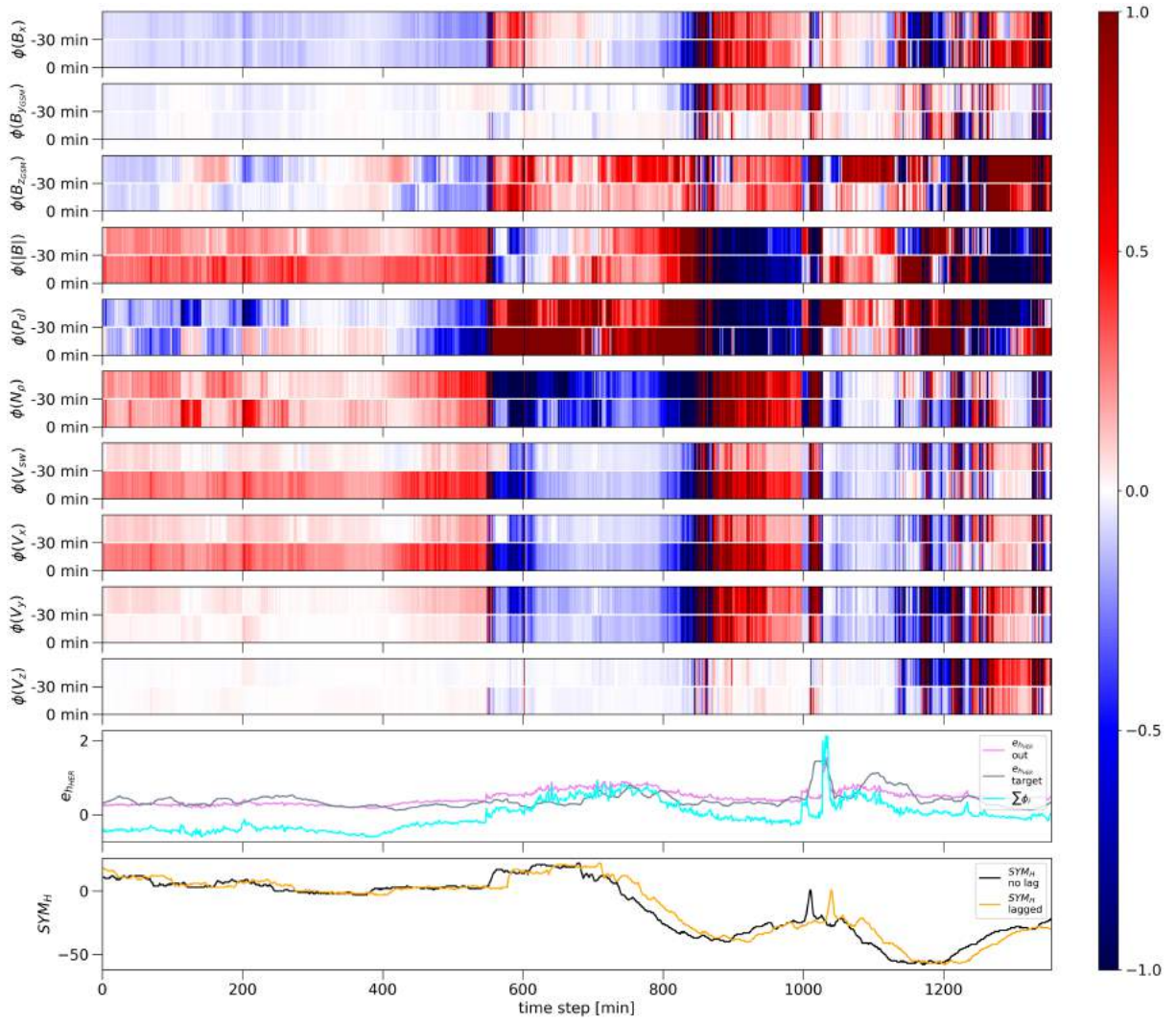
Attribution plot of all non auto-regressive input features for $e_{h_{ABK}}$ predicted during 1997_moderate.001 in the individual e_h prediction task.



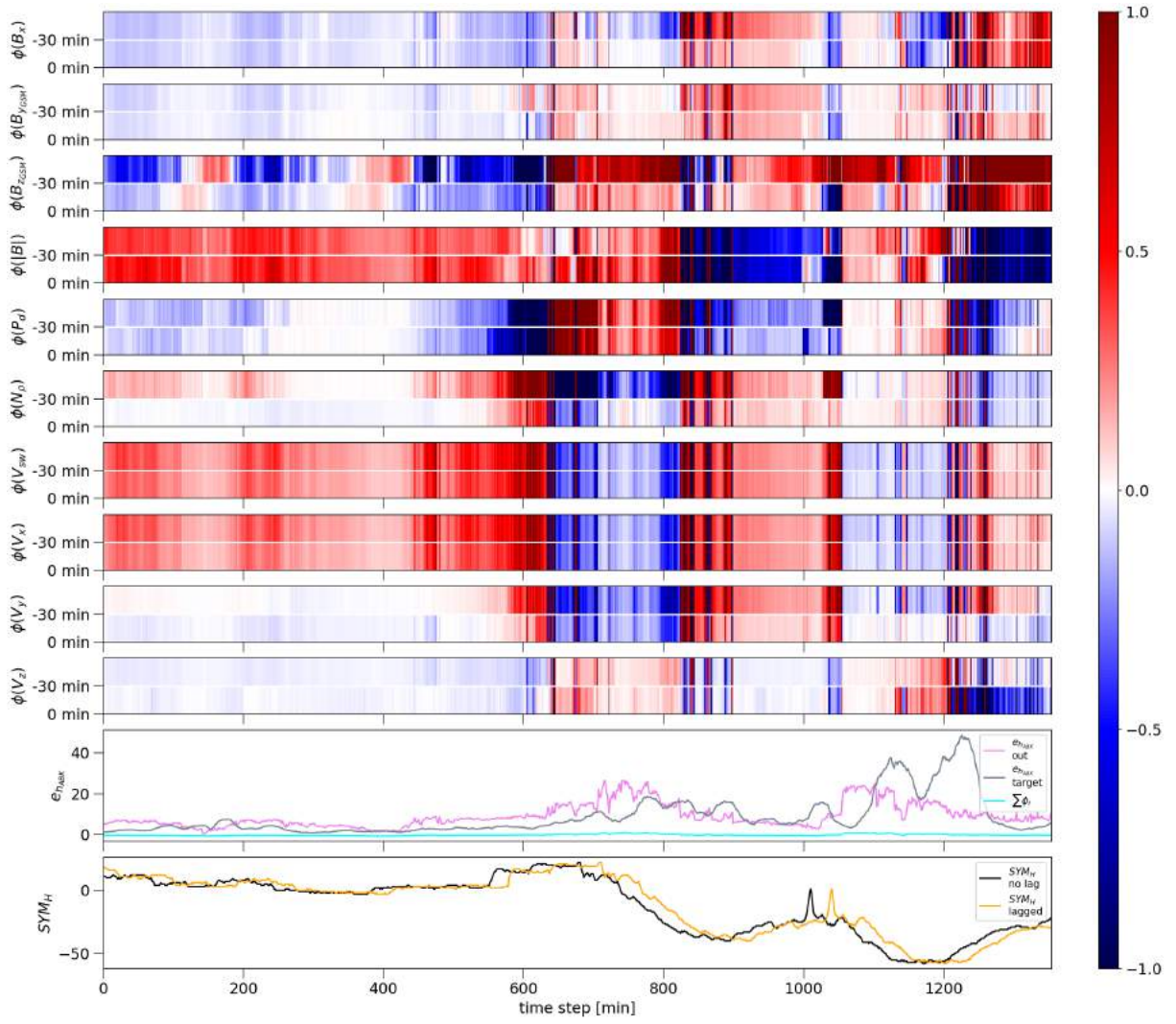
Attribution plot of all non auto-regressive input features for $e_{h_{HER}}$ predicted during 1997.moderate.001 in the simultaneous e_h prediction task.



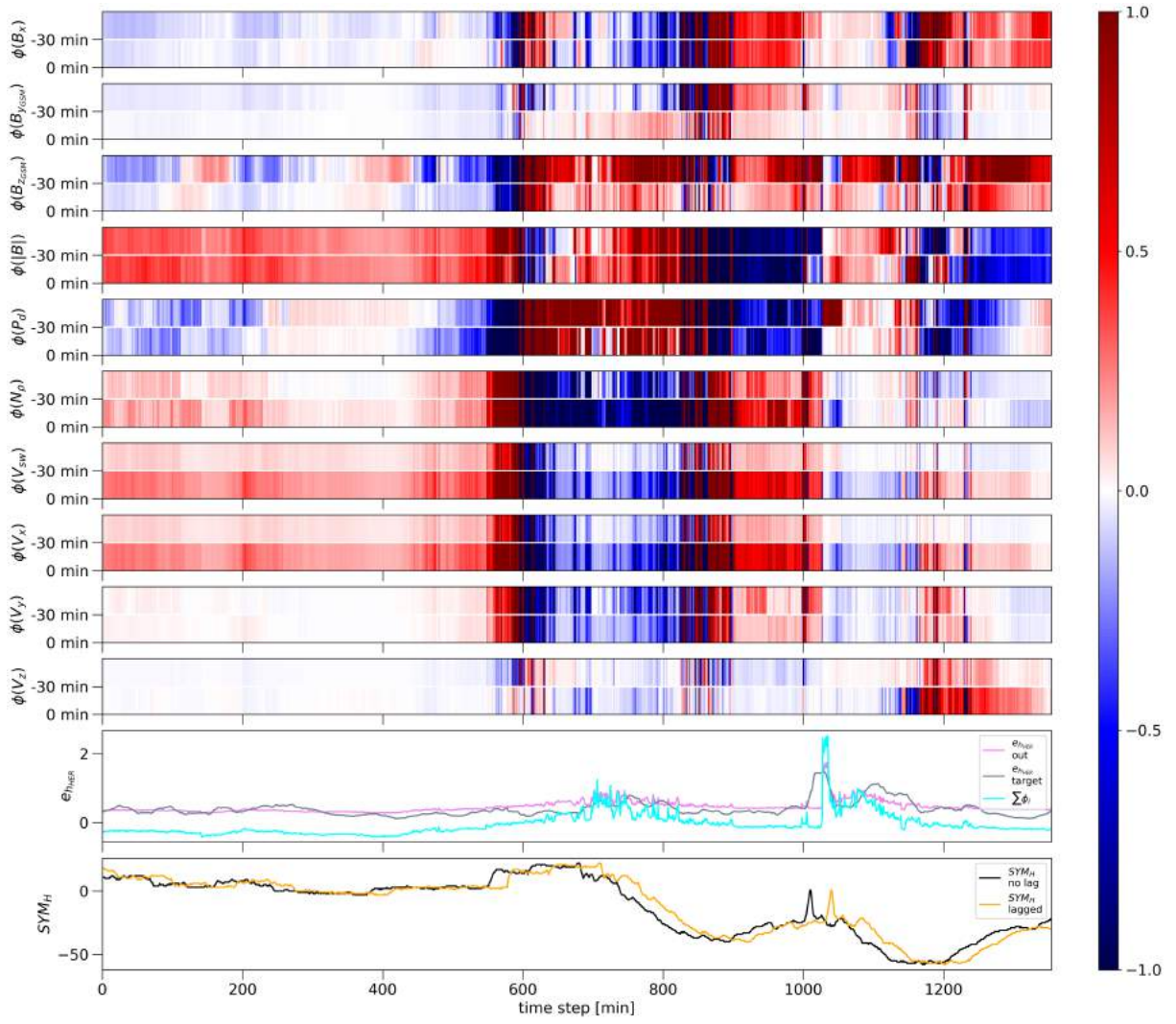
Attribution plot of all non auto-regressive input features for $e_{h_{ABK}}$ predicted during 1997.moderate.001 in the simultaneous e_h prediction task.



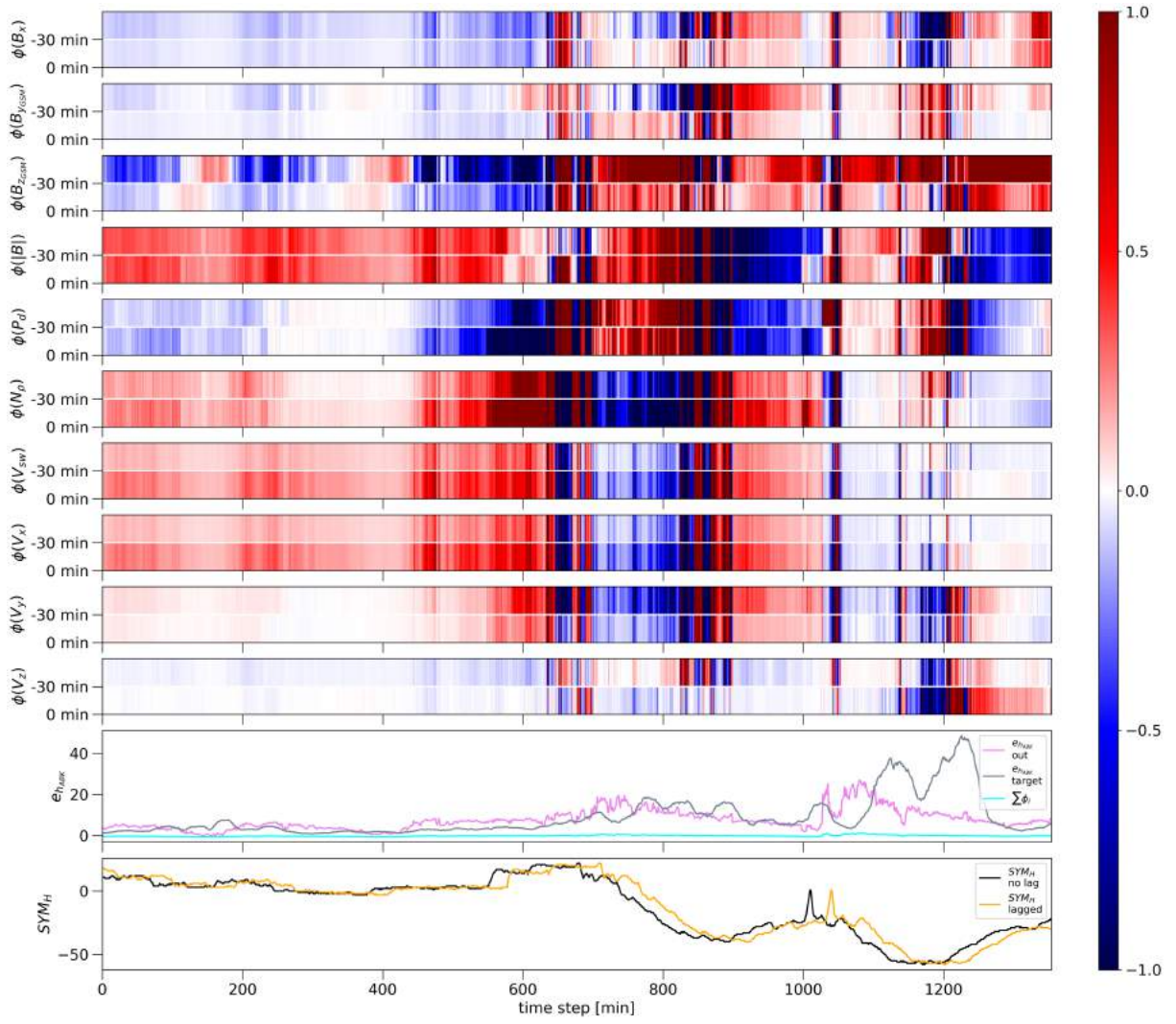
Attribution plot of all non auto-regressive input features for $e_{h_{HER}}$ predicted during 2005_mild.004 in the individual e_h prediction task.



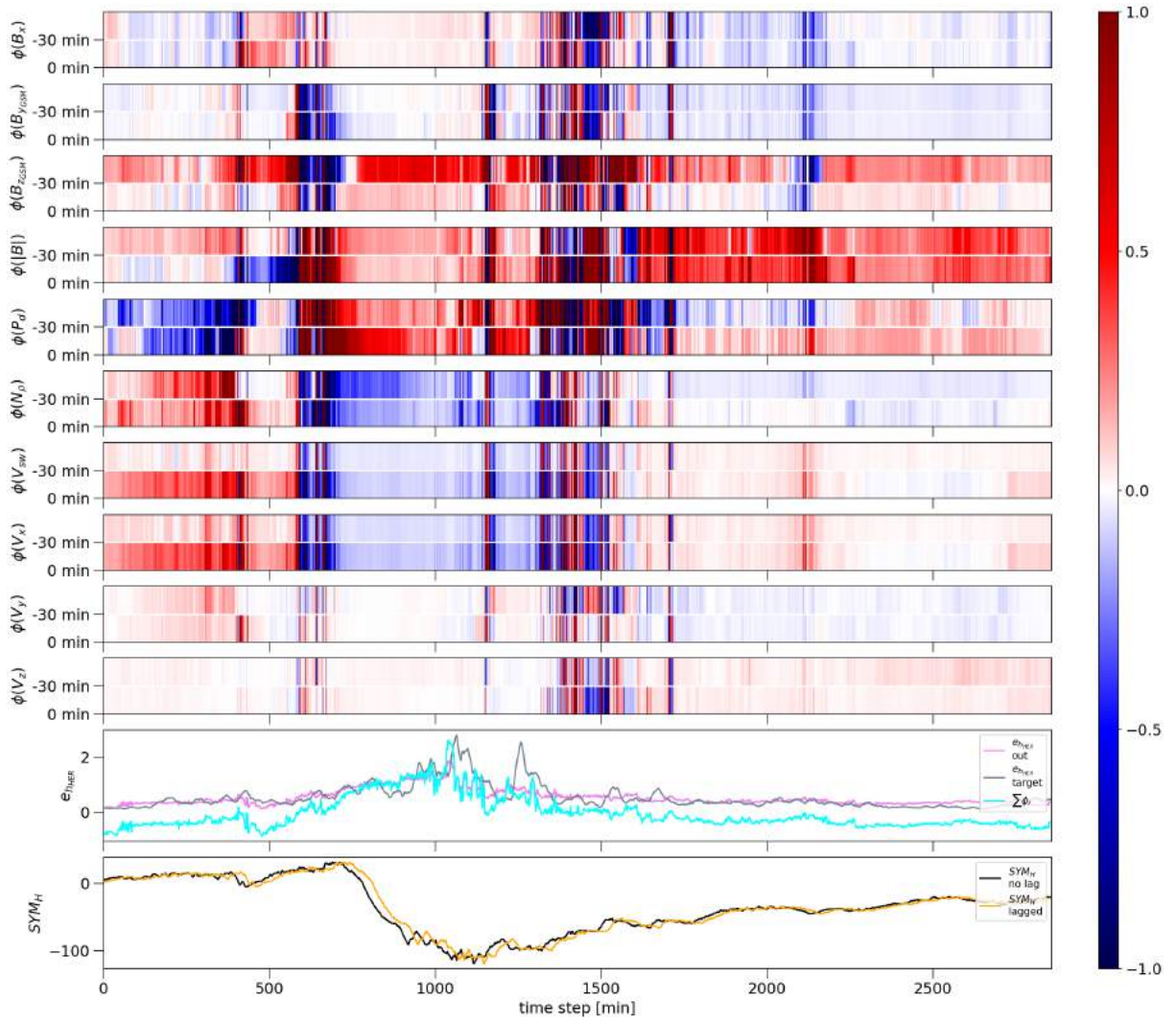
Attribution plot of all non auto-regressive input features for $e_{h_{ABK}}$ predicted during 2005_mild.004 in the individual e_h prediction task.



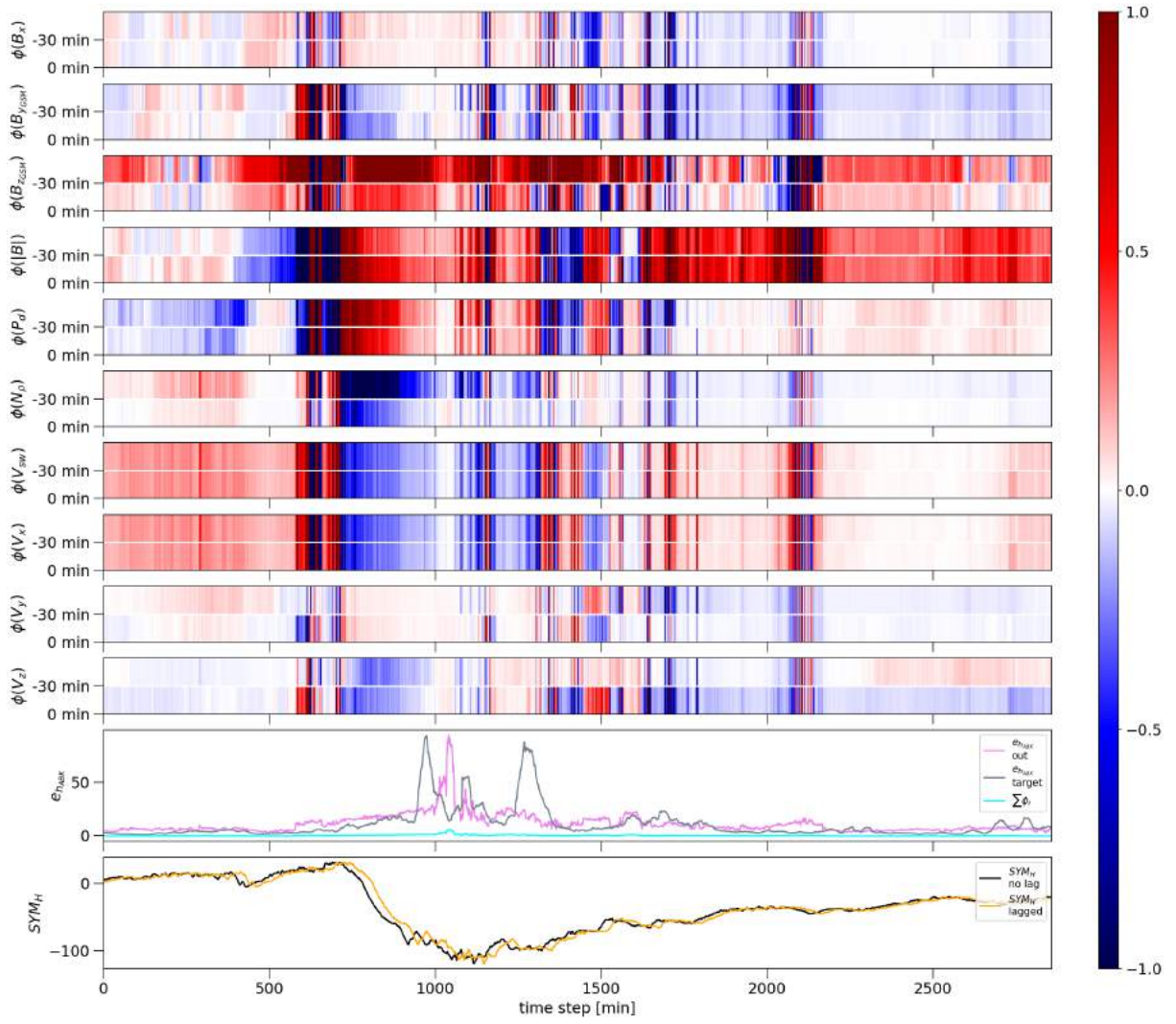
Attribution plot of all non auto-regressive input features for $e_{h_{HER}}$ predicted during 2005_mild_004 in the simultaneous e_h prediction task.



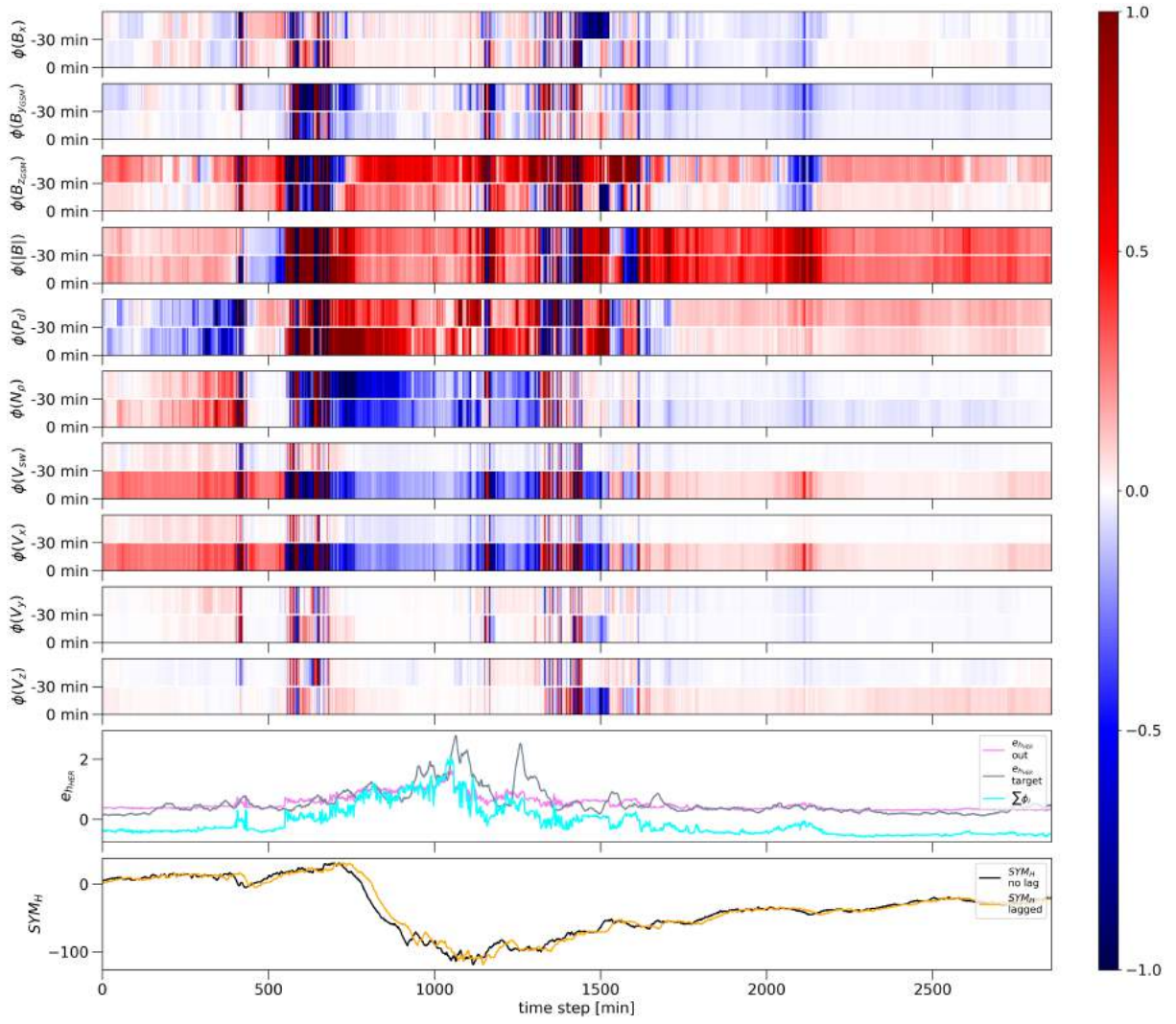
Attribution plot of all non auto-regressive input features for $e_{h_{ABK}}$ predicted during 2005_mild_004 in the simultaneous e_h prediction task.



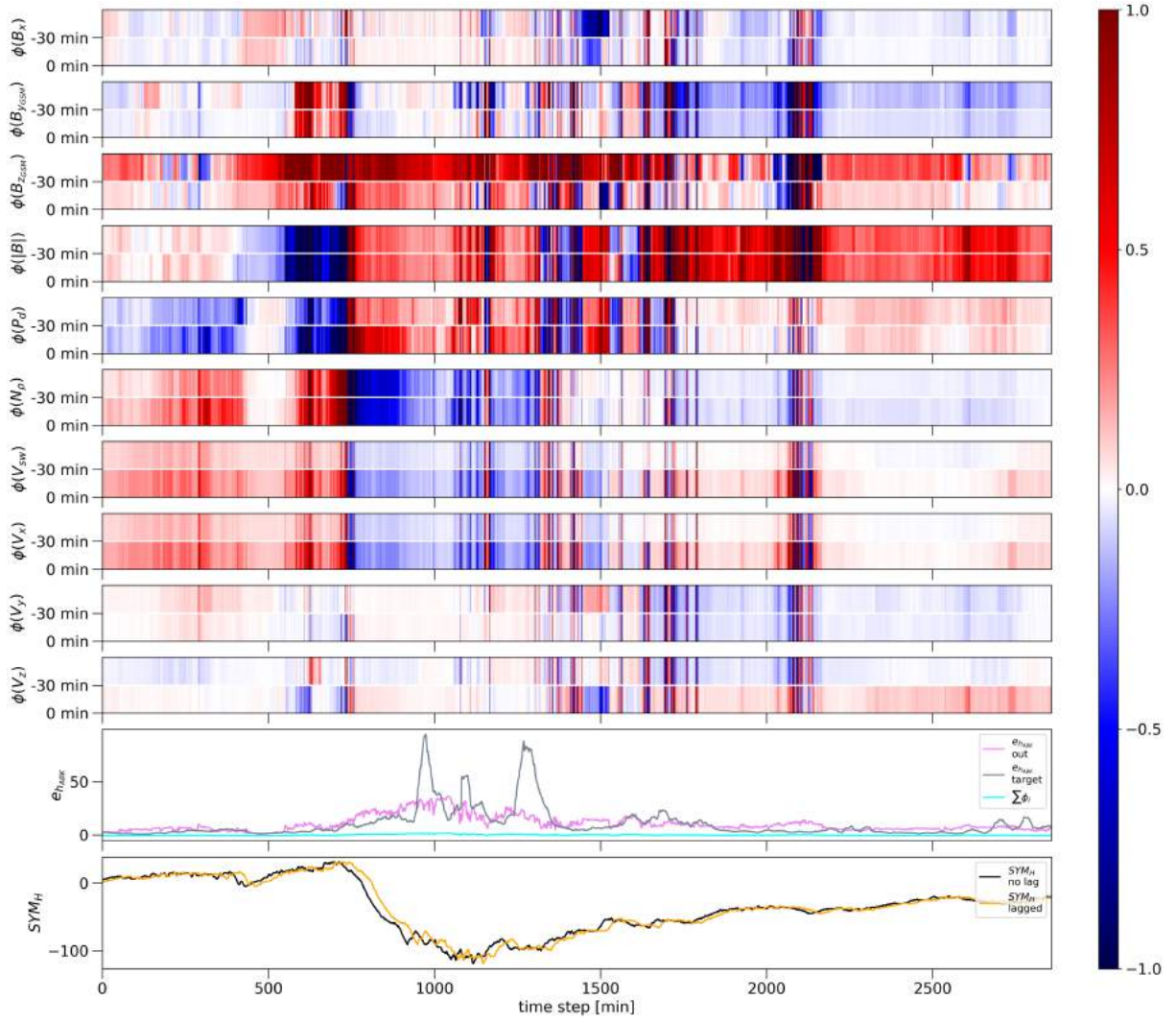
Attribution plot of all non auto-regressive input features for $e_{h_{HER}}$ predicted during 2005.severe.010 in the individual e_h prediction task.



Attribution plot of all non auto-regressive input features for e_{hABK} predicted during 2005.severe.010 in the individual e_h prediction task.



Attribution plot of all non auto-regressive input features for $e_{h_{HER}}$ predicted during 2005_severe_010 in the simultaneous e_h prediction task.

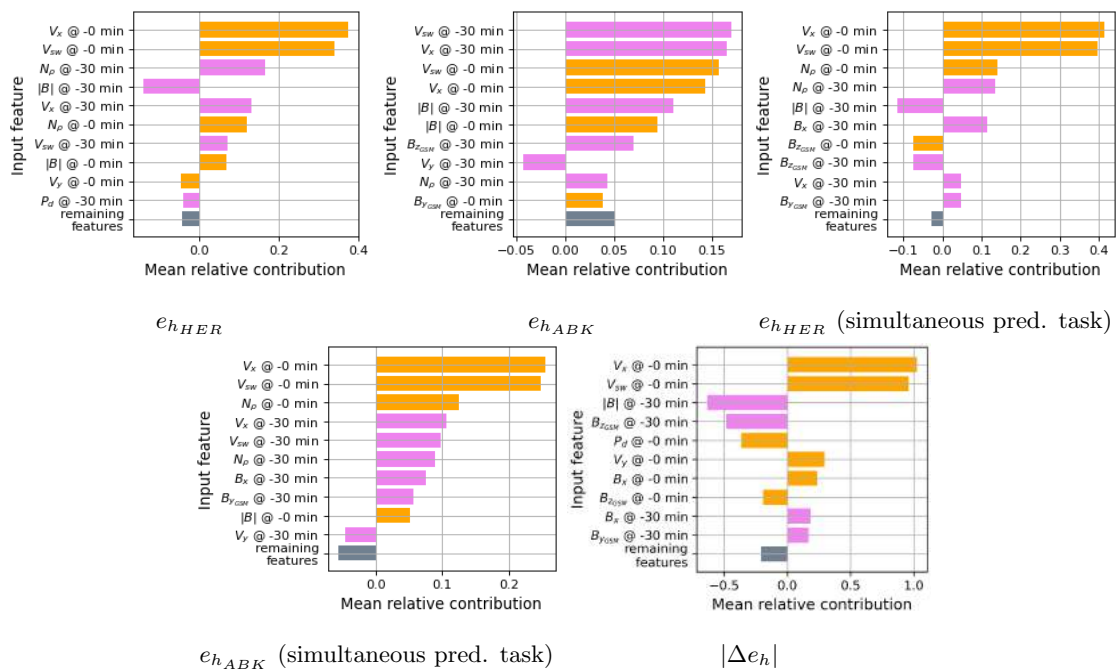


Attribution plot of all non auto-regressive input features for e_{hABK} predicted during 2005.severe.010 in the simultaneous e_h prediction task.

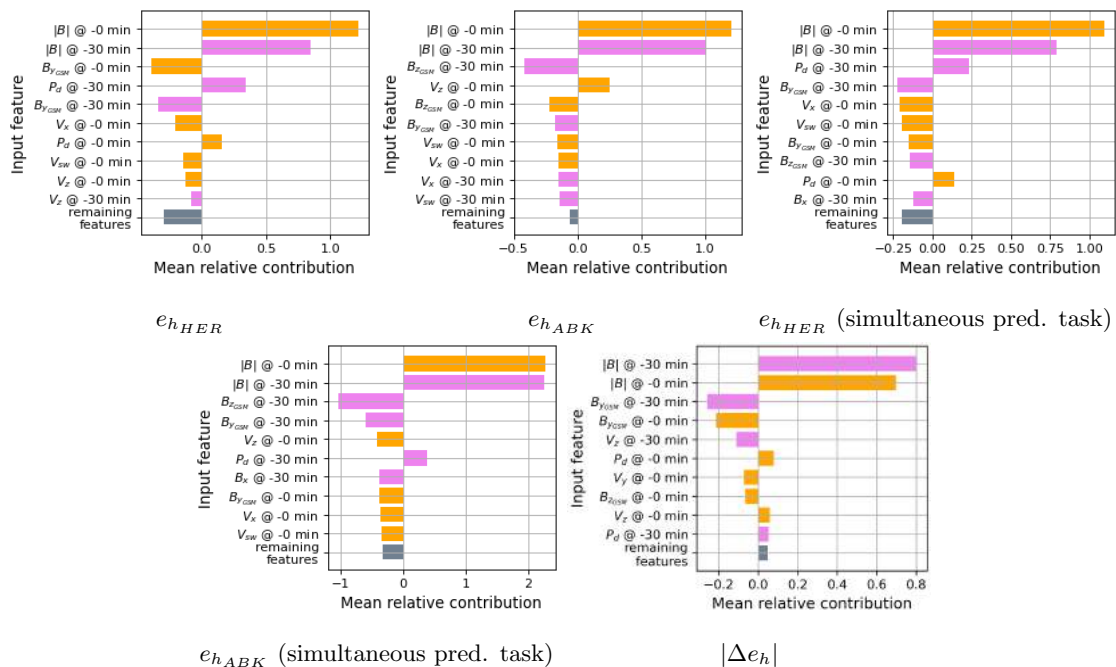
A.4.3 Supplemental CME-driven storm attributions

Here we display feature ranking plots and attribution plots for additional CME-driven storm examples.

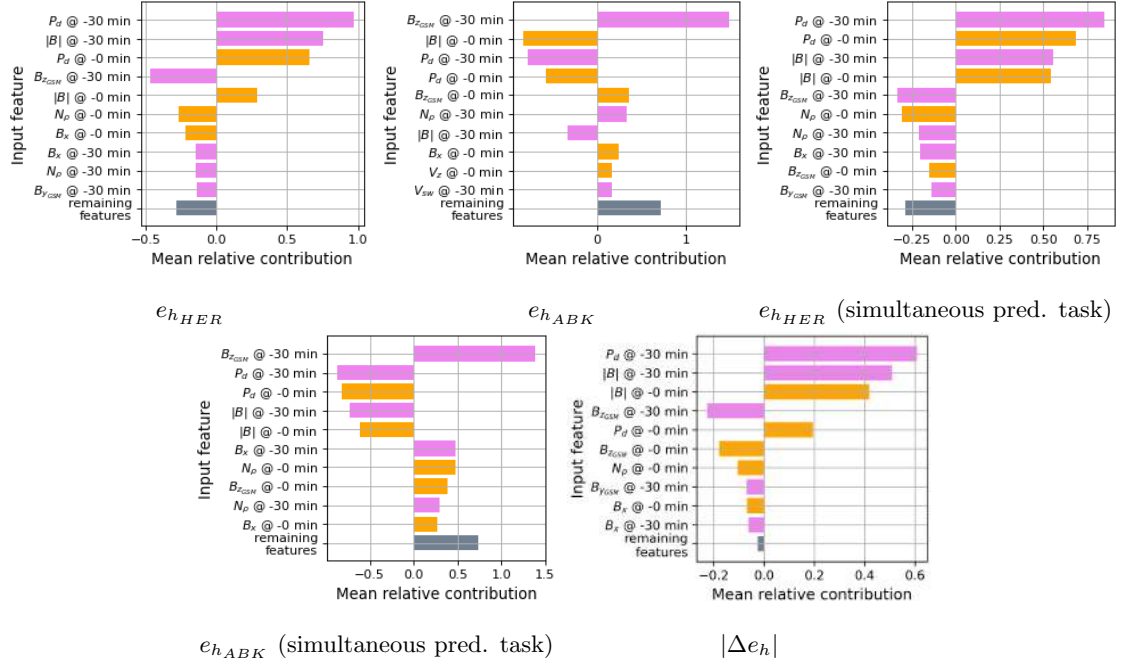
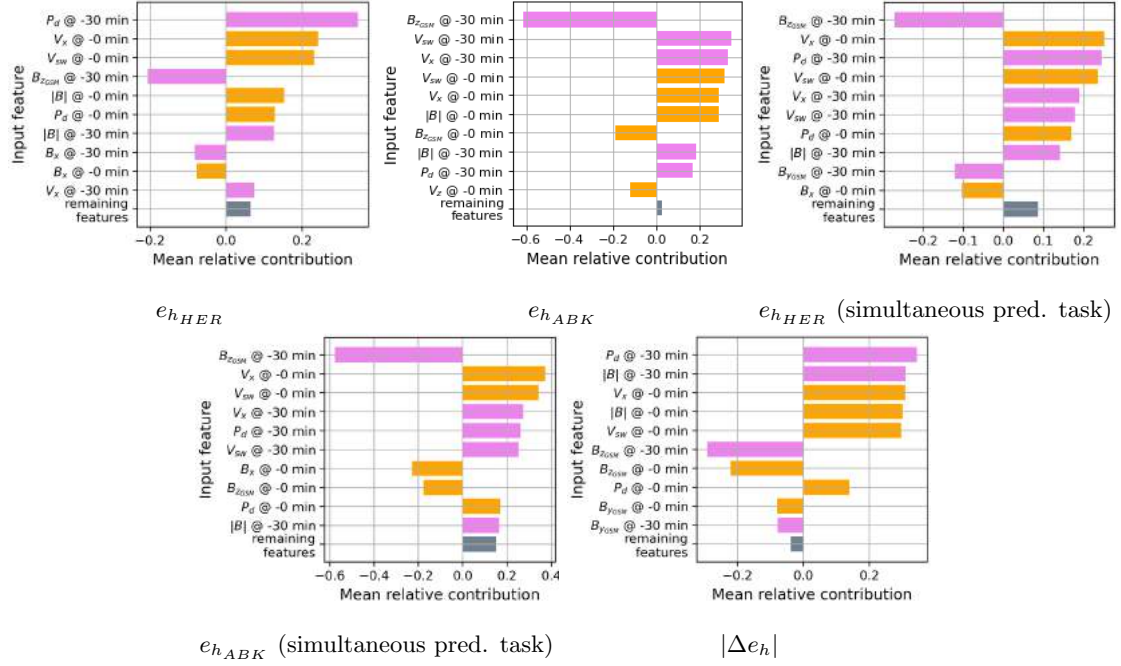
Feature ranking plots for supplemental CME-driven storms

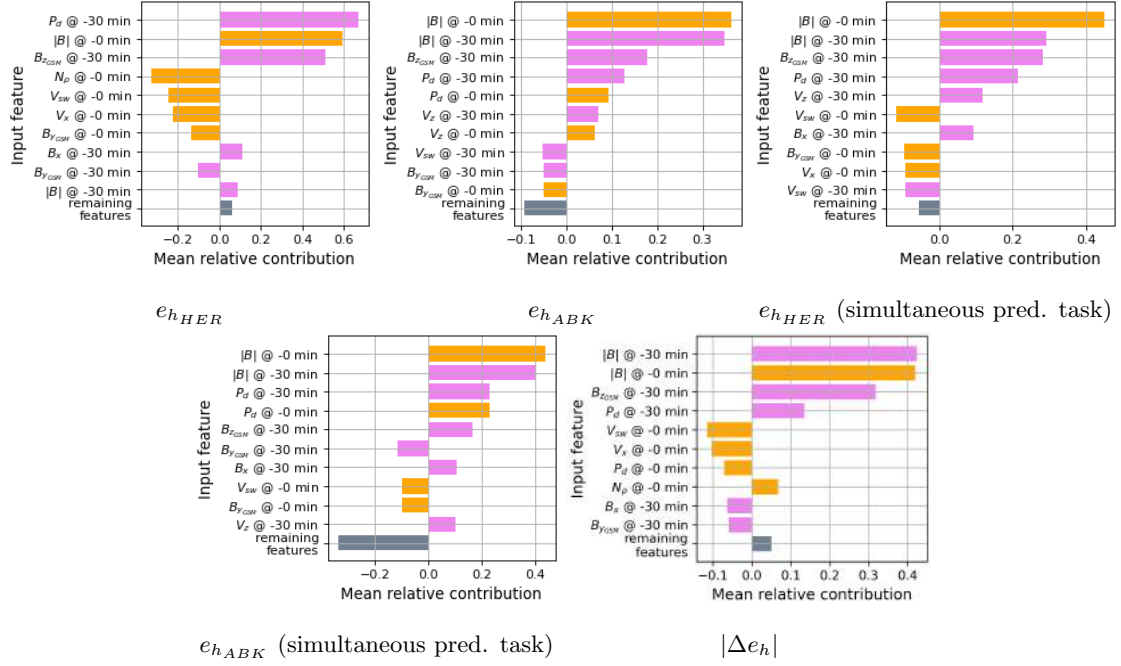
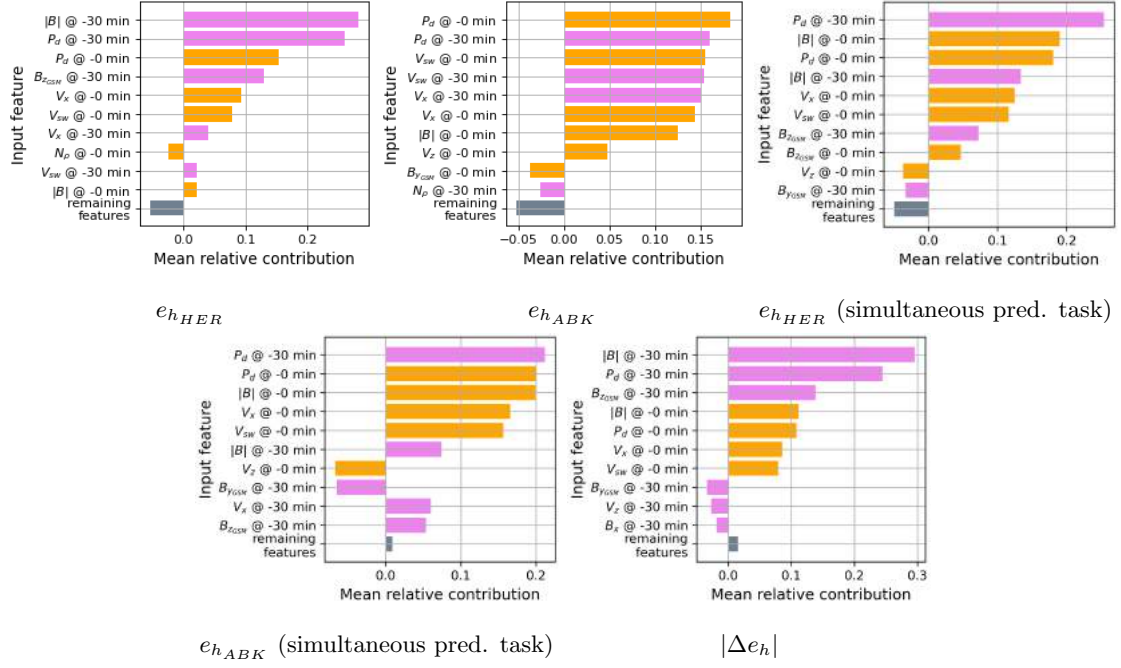


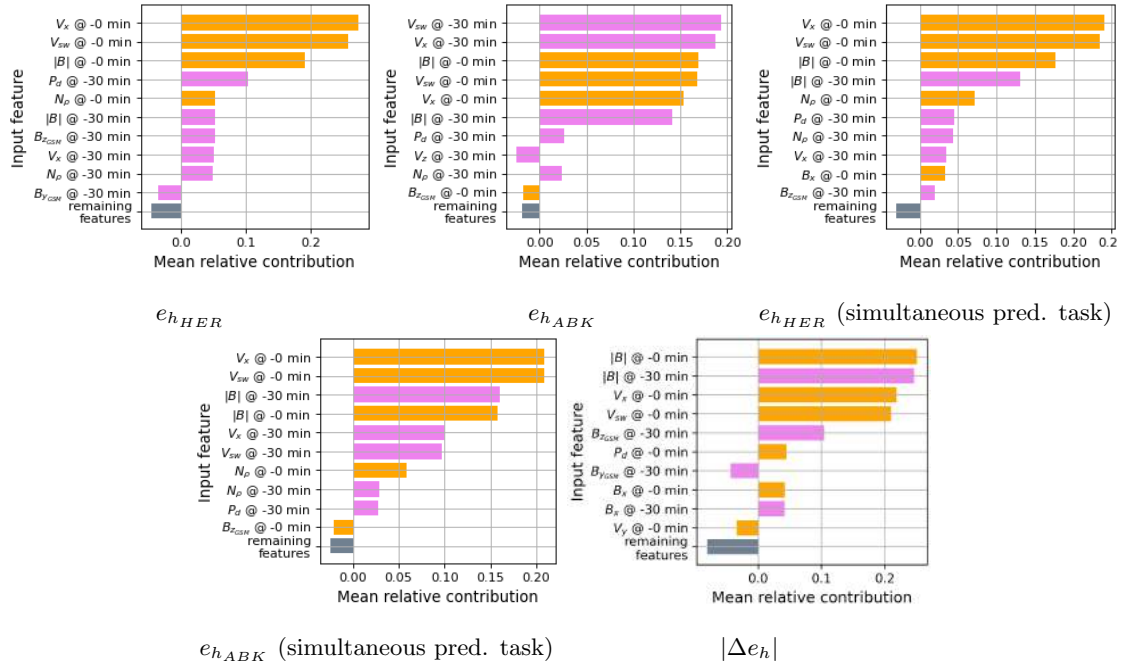
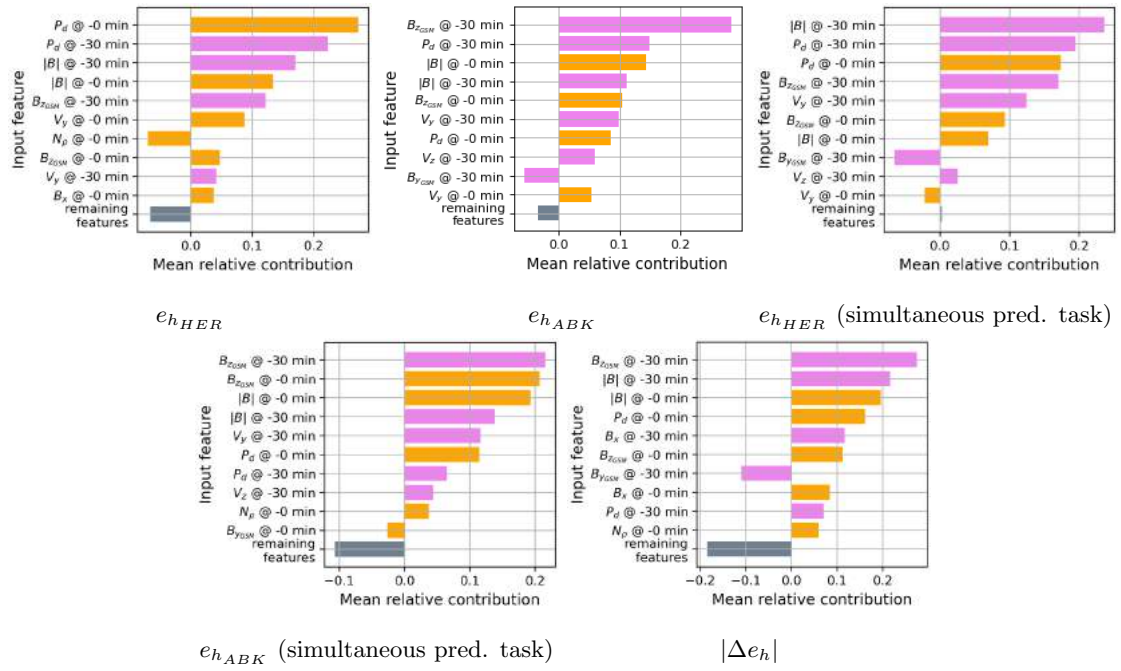
Feature ranking plots for all tasks during storm 1997_severe.000.



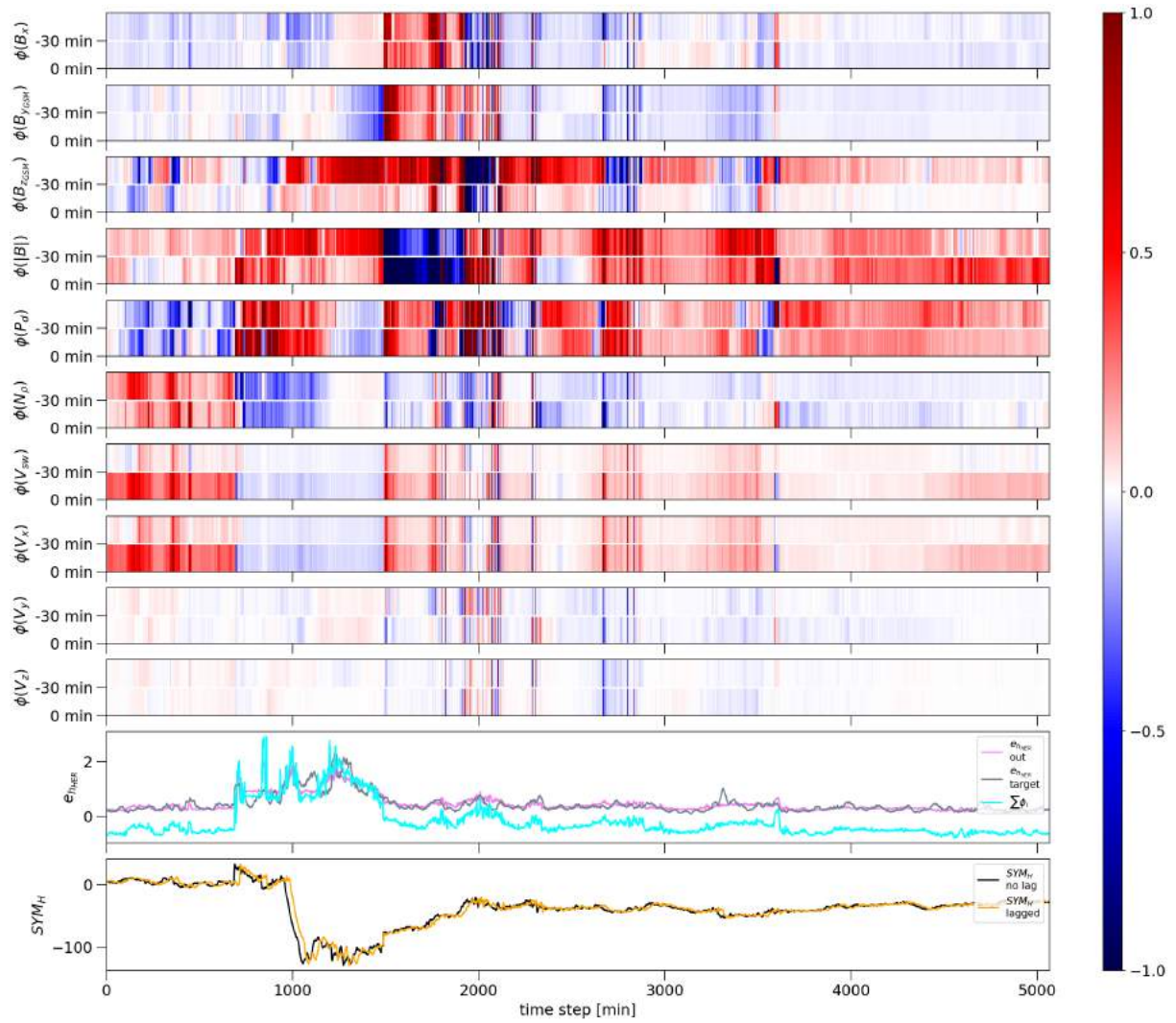
Feature ranking plots for all tasks during storm 2000_severe.010.



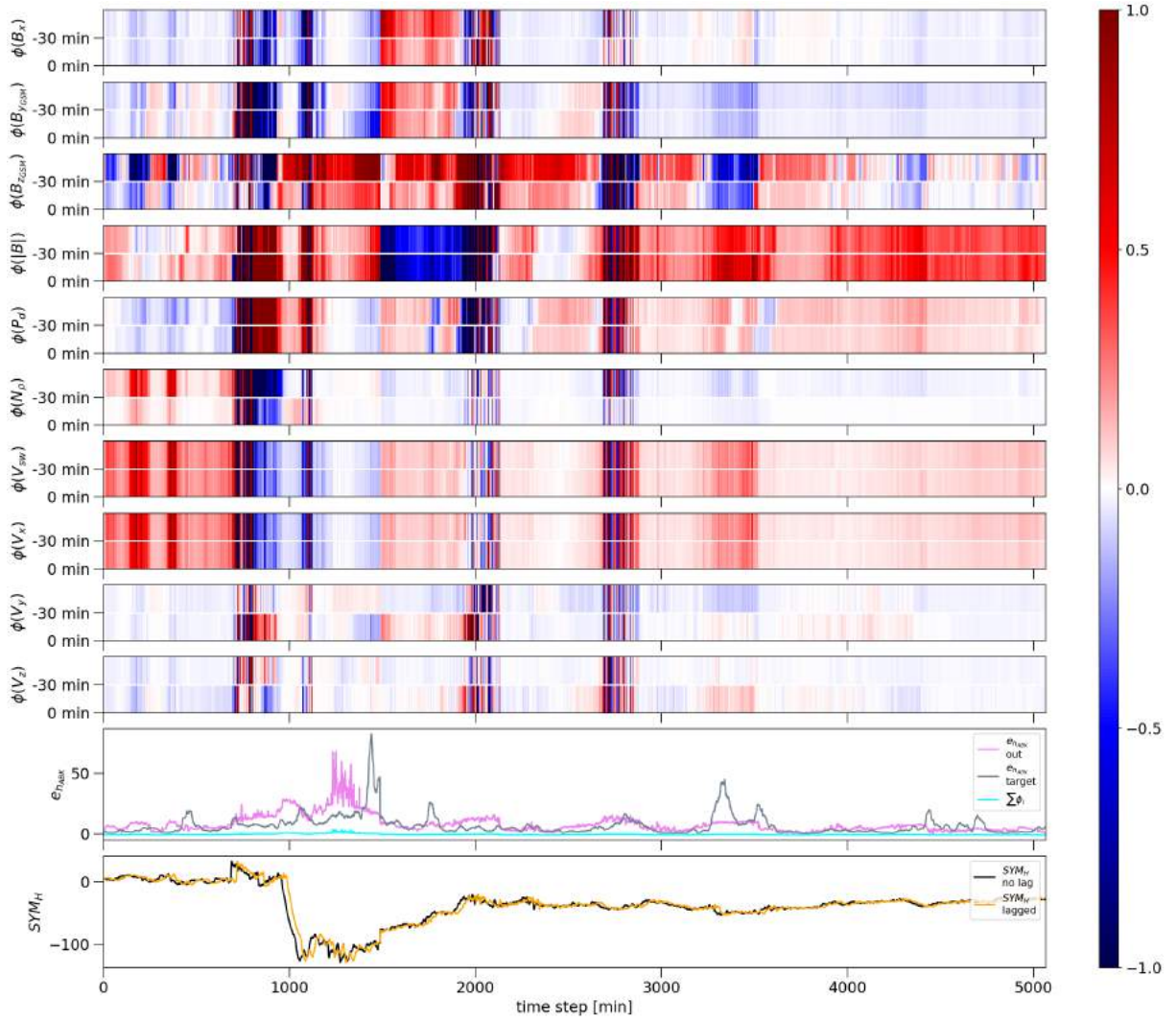




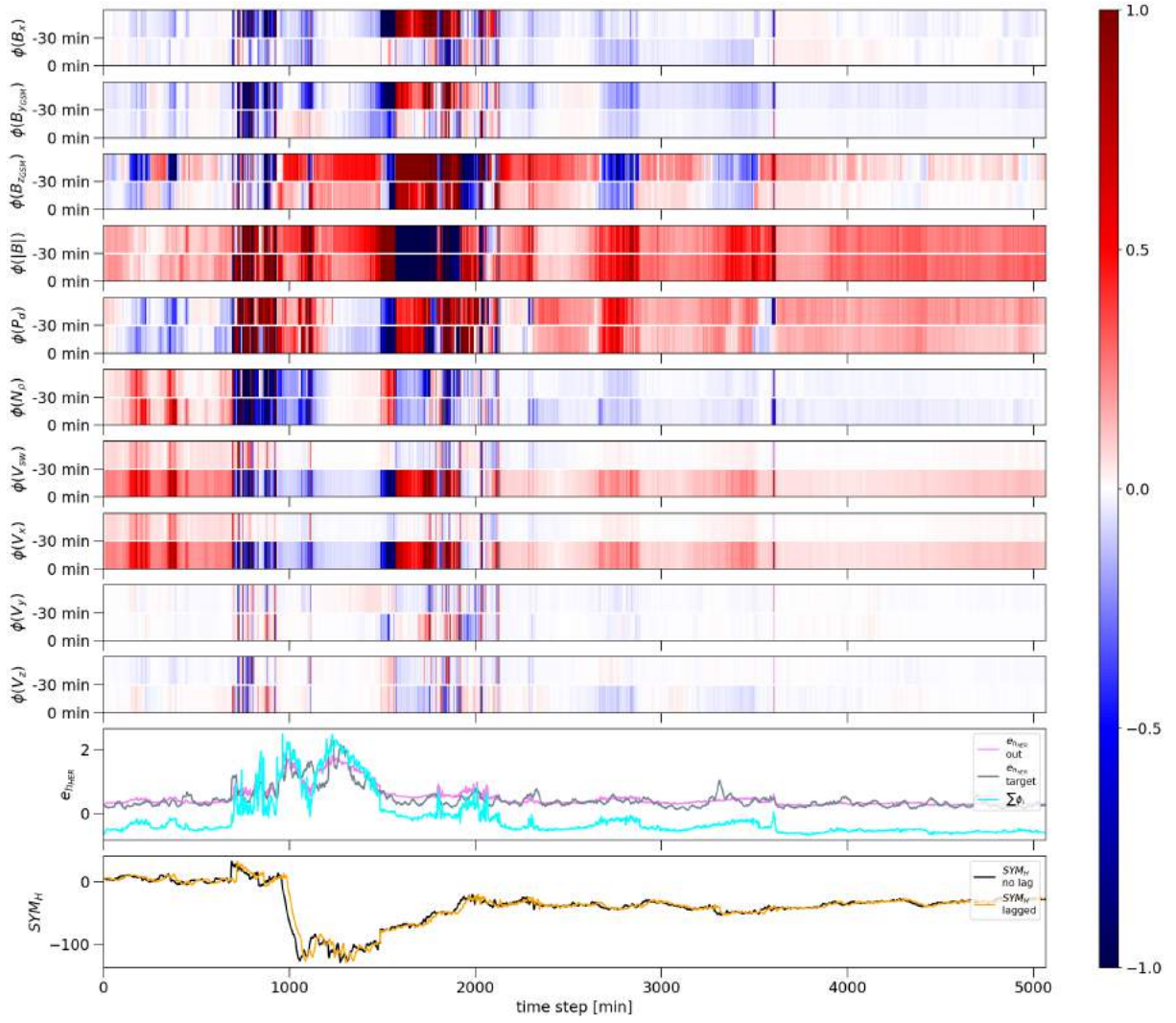
Attribution plots



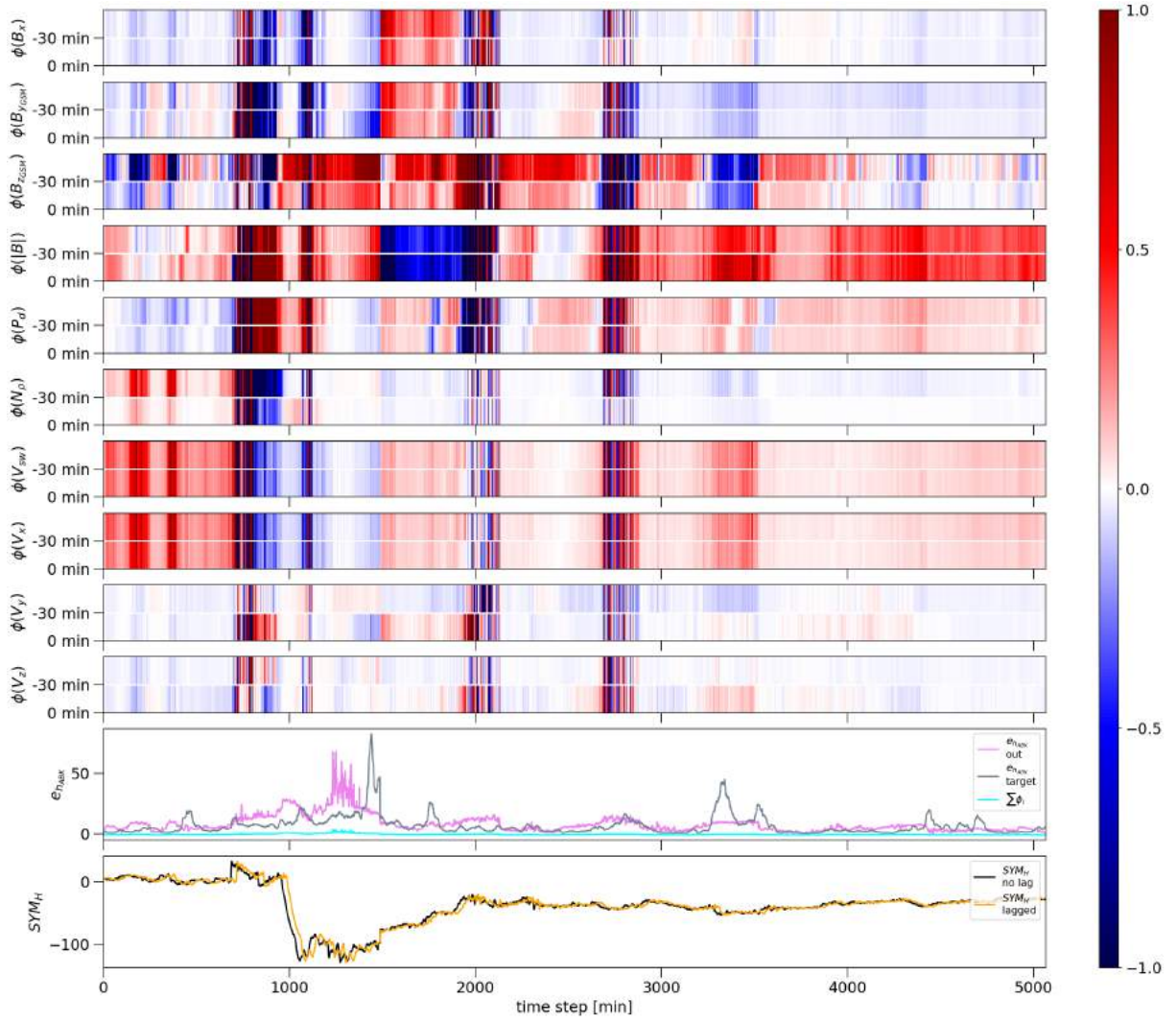
Attribution plot of all non auto-regressive input features for $e_{h_{HER}}$ predicted during 1997_severe_000 in the individual e_h prediction task.



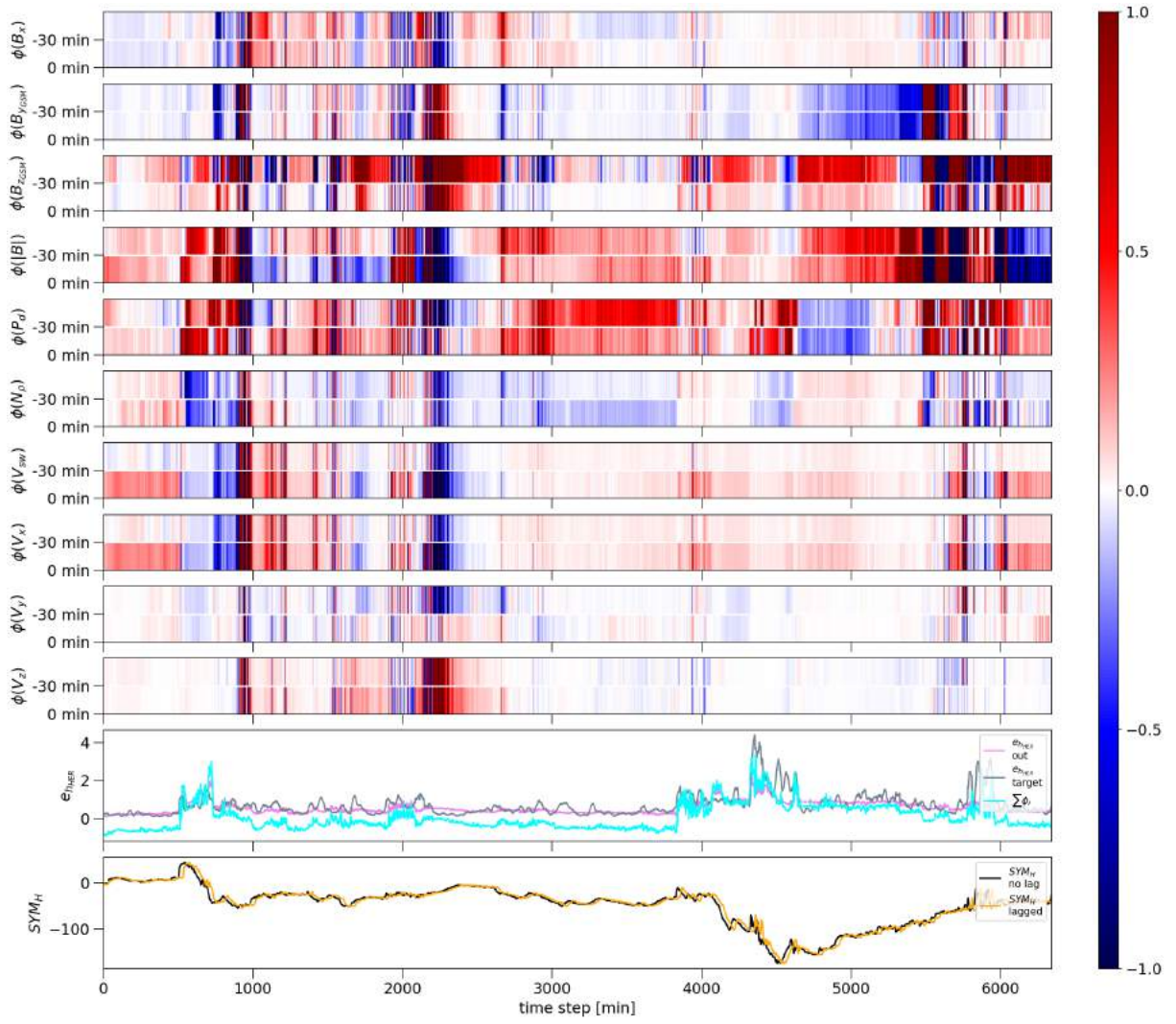
Attribution plot of all non auto-regressive input features for $e_{h_{ABK}}$ predicted during 1997_severe_000 in the individual e_h prediction task.



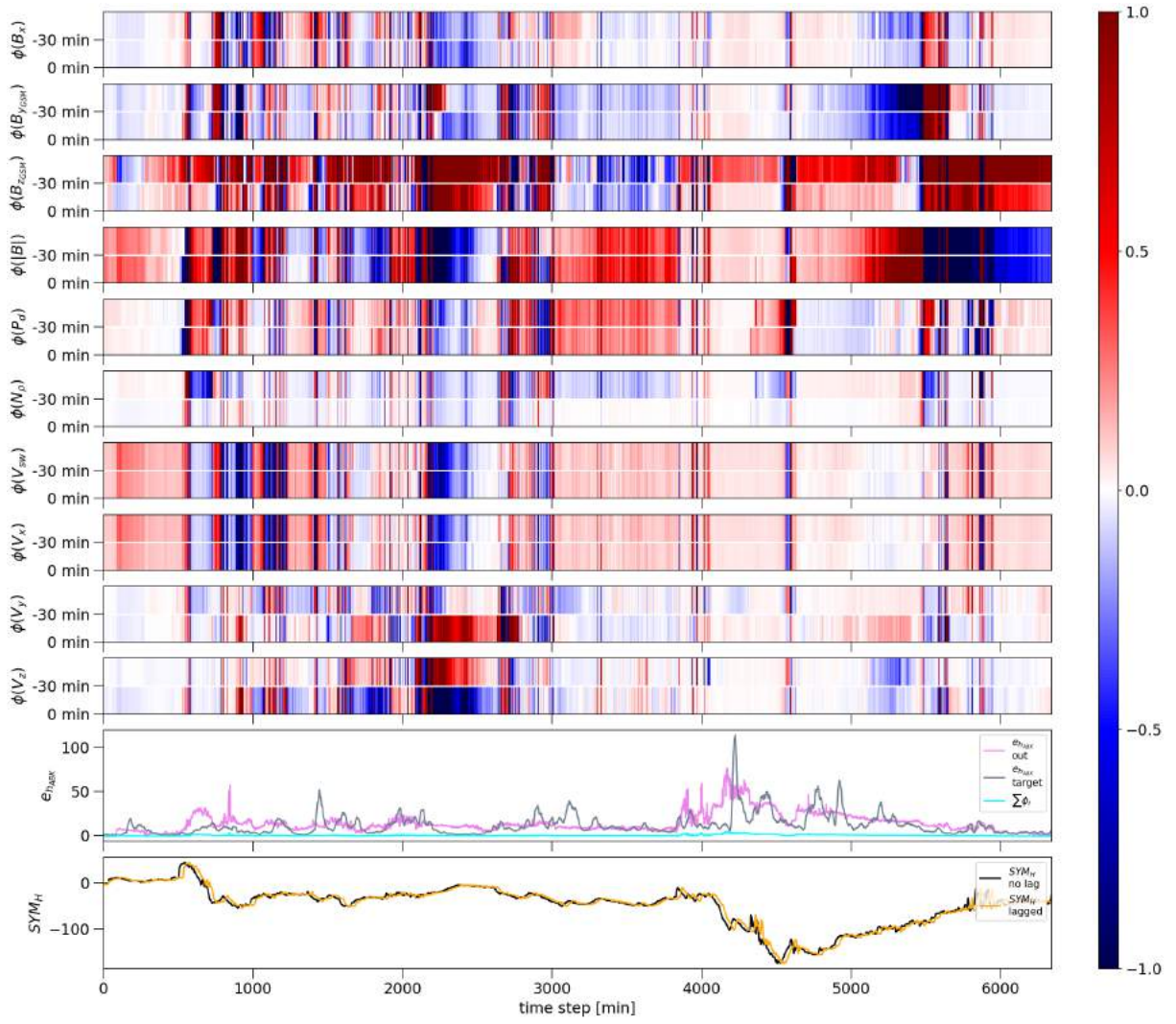
Attribution plot of all non auto-regressive input features for $e_{h_{HER}}$ predicted during 1997_severe_000 in the simultaneous e_h prediction task.



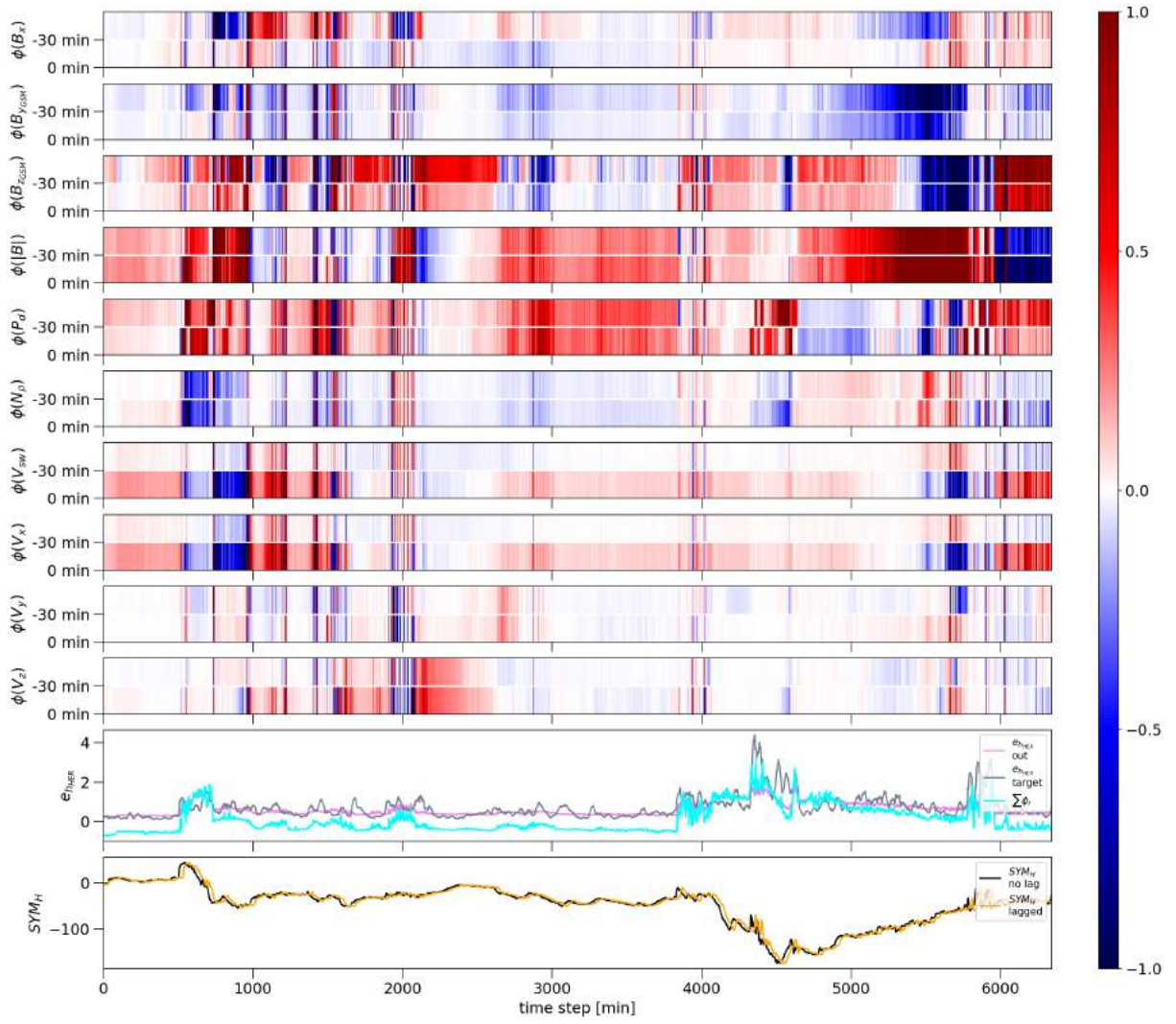
Attribution plot of all non auto-regressive input features for e_{hABK} predicted during 1997_severe_000 in the simultaneous e_h prediction task.



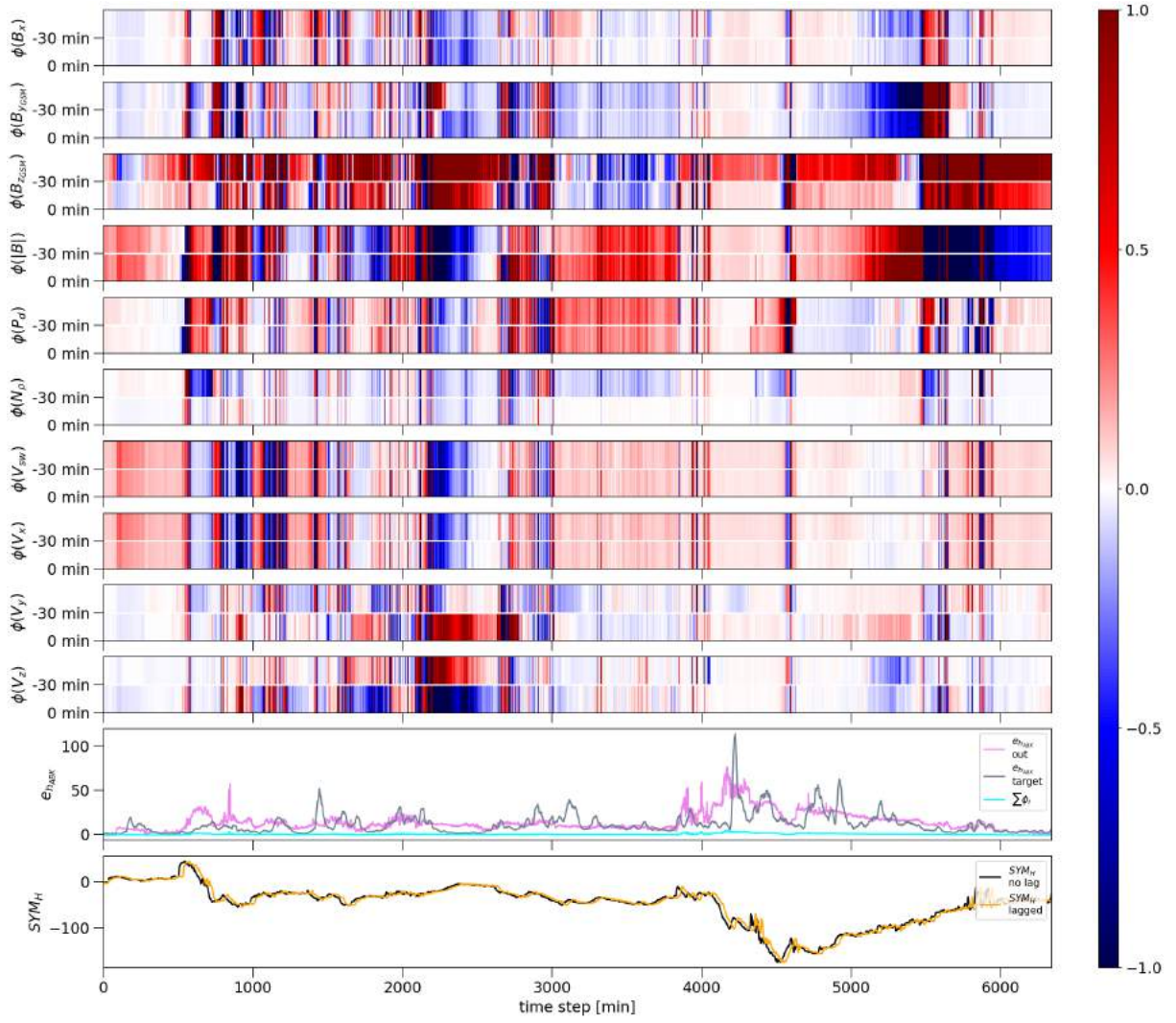
Attribution plot of all non auto-regressive input features for $e_{h_{HER}}$ predicted during 2000_severe_010 in the individual e_h prediction task.



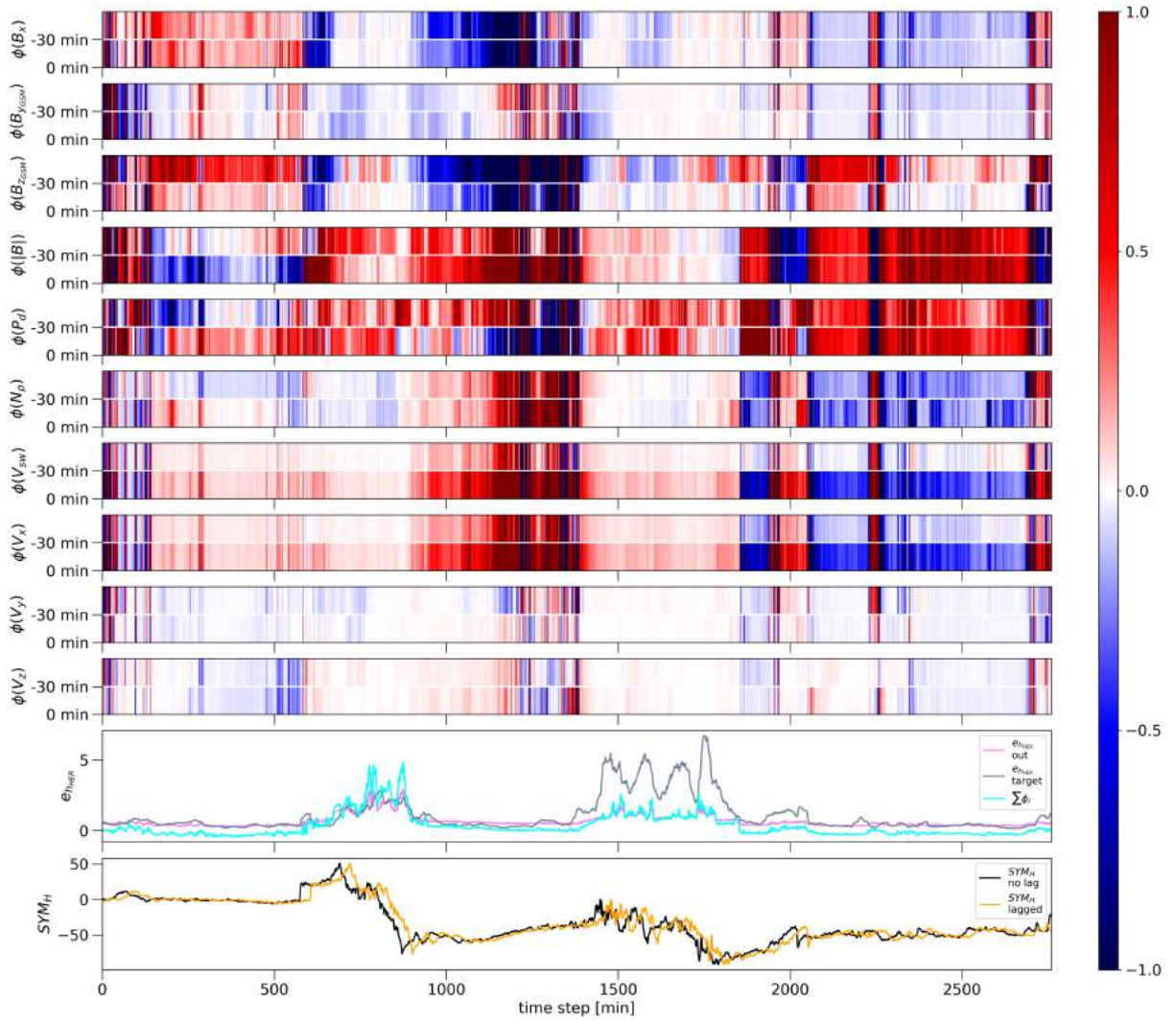
Attribution plot of all non auto-regressive input features for e_{hABK} predicted during 2000_severe_010 in the individual e_h prediction task.



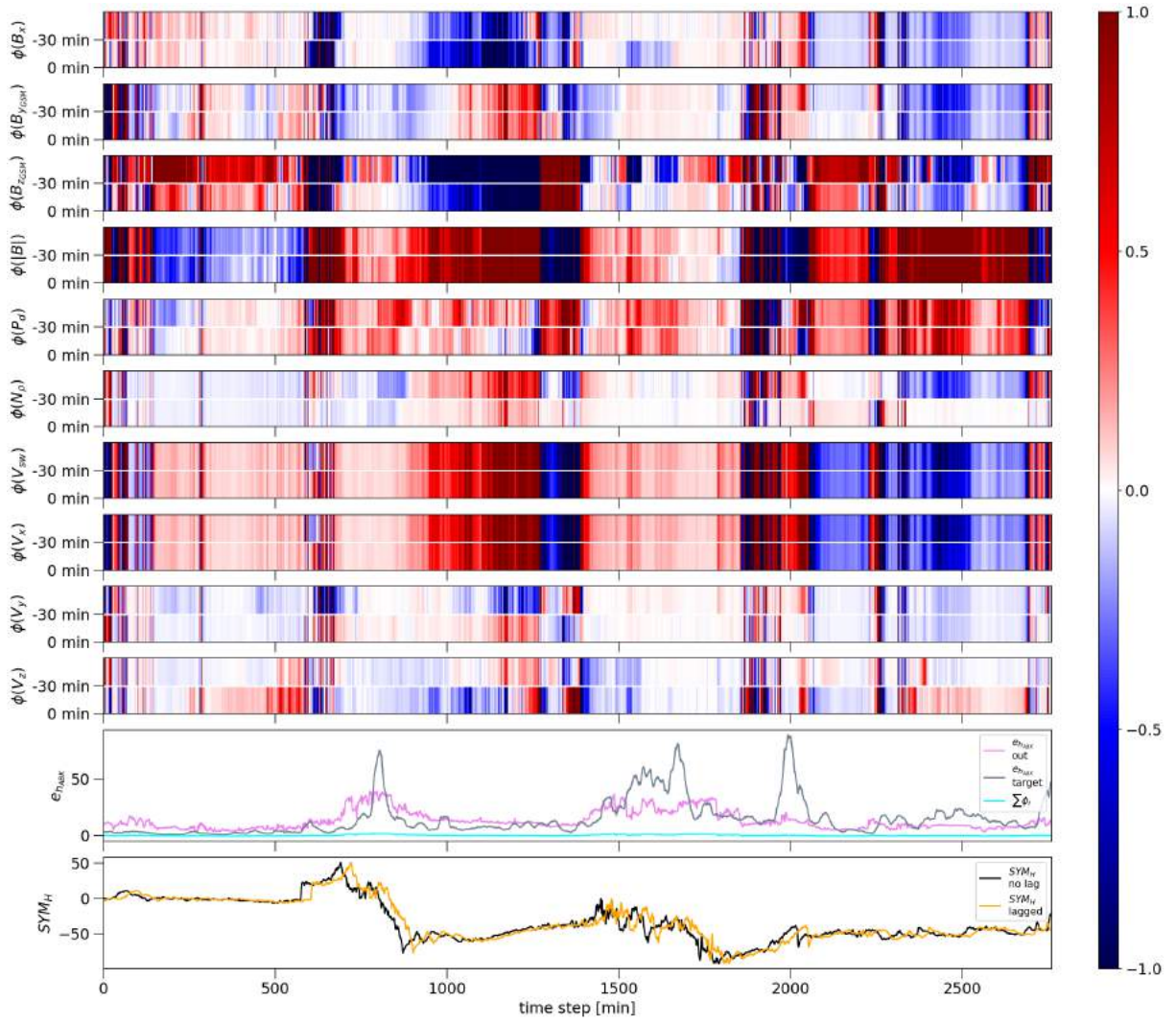
Attribution plot of all non auto-regressive input features for $e_{h_{HER}}$ predicted during 2000_severe_010 in the simultaneous e_h prediction task.



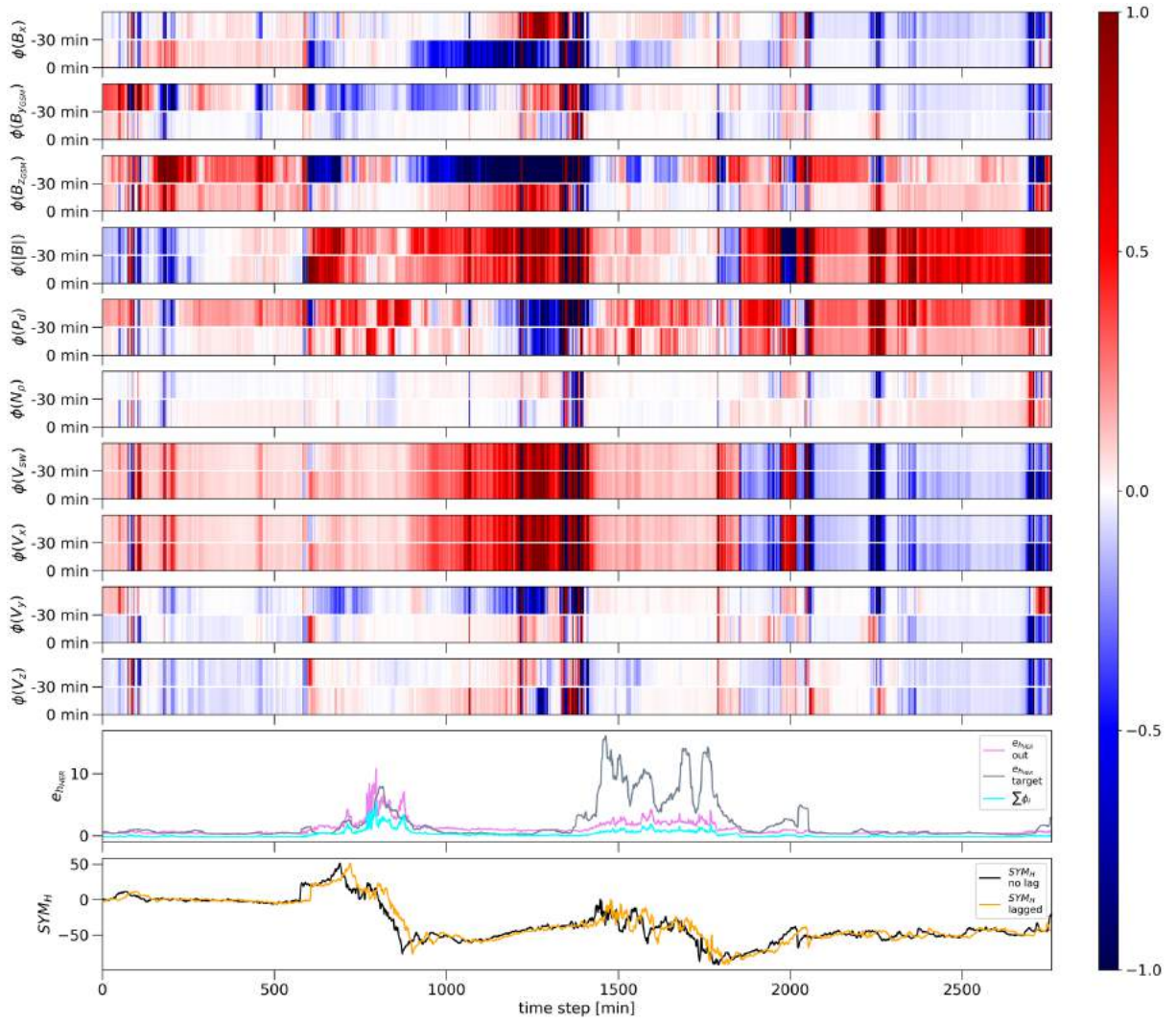
Attribution plot of all non auto-regressive input features for e_{hABK} predicted during 2000_severe_010 in the simultaneous e_h prediction task.



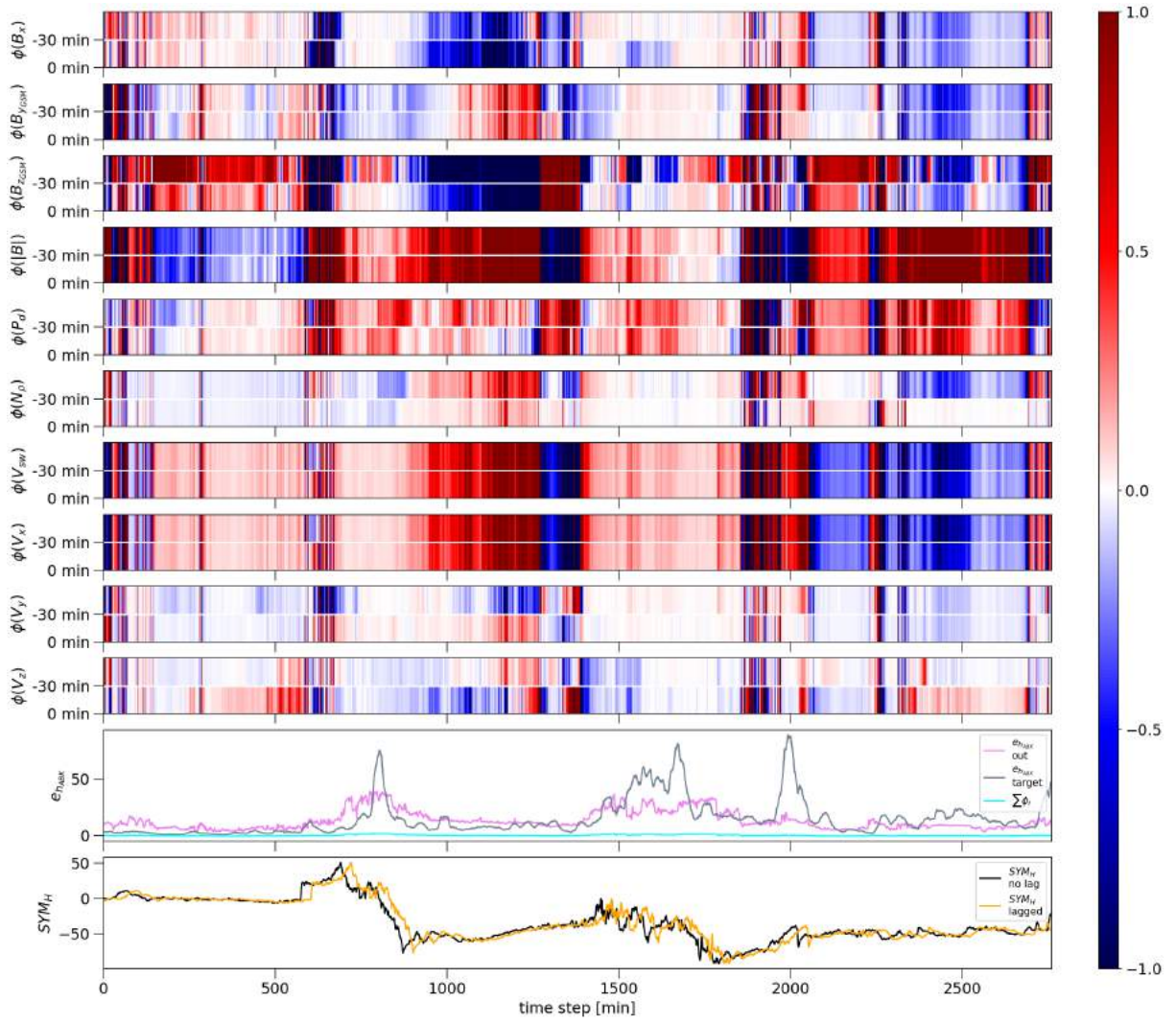
Attribution plot of all non auto-regressive input features for $e_{h_{HER}}$ predicted during 2001.moderate.001 in the individual e_h prediction task.



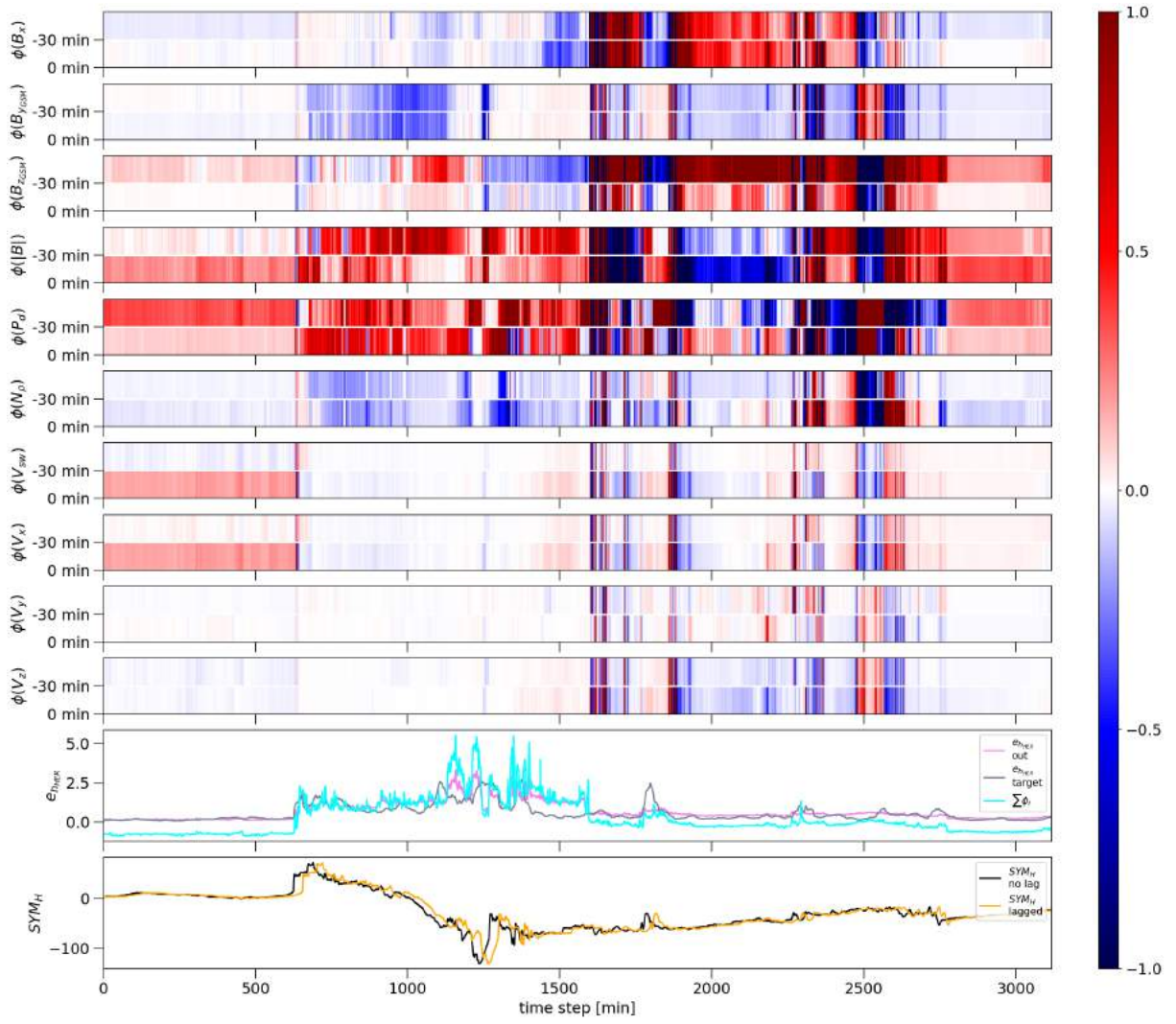
Attribution plot of all non auto-regressive input features for e_{hABK} predicted during 2001.moderate.001 in the individual e_h prediction task.



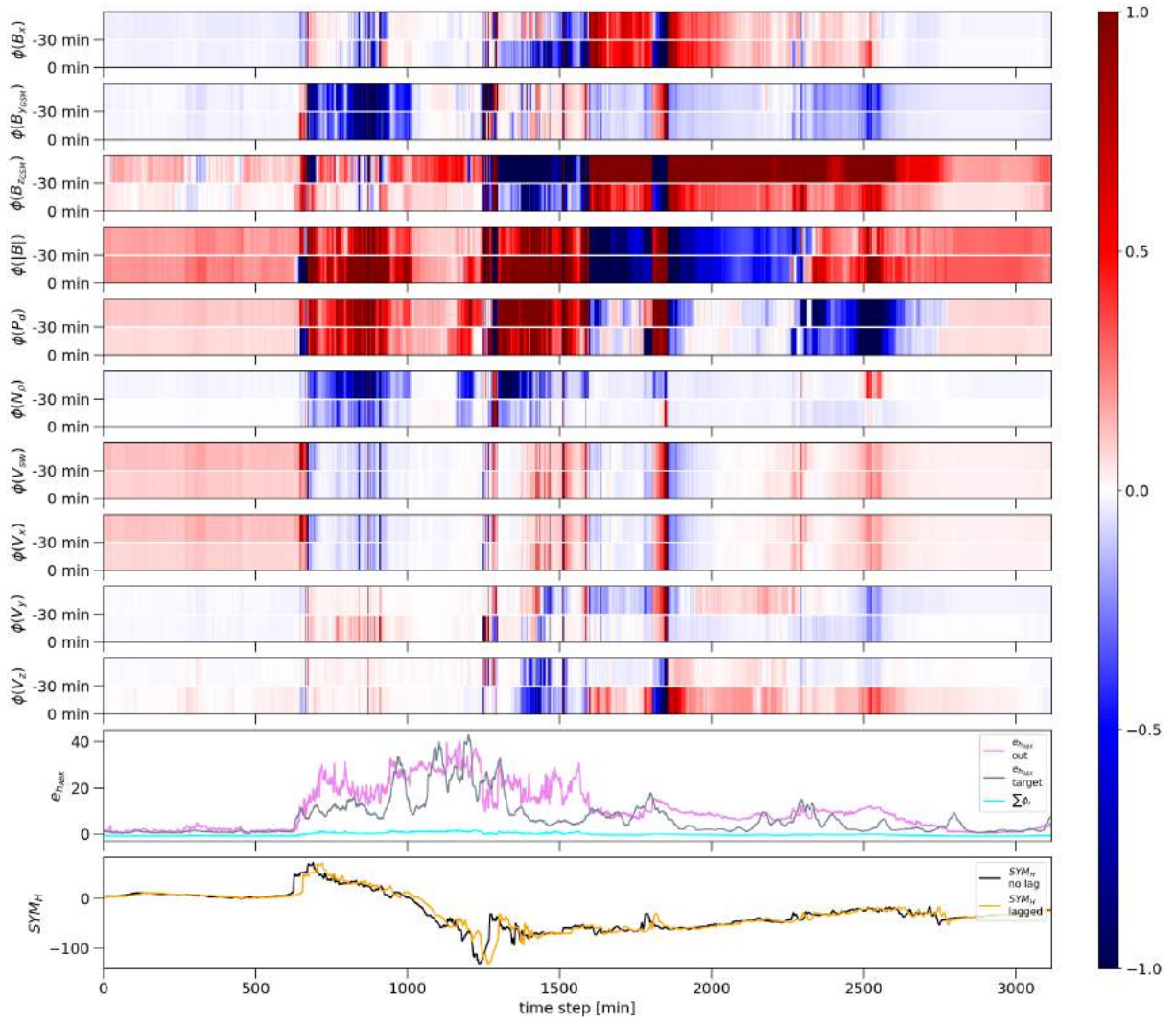
Attribution plot of all non auto-regressive input features for $e_{h_{HER}}$ predicted during 2001.moderate.001 in the simultaneous e_h prediction task.



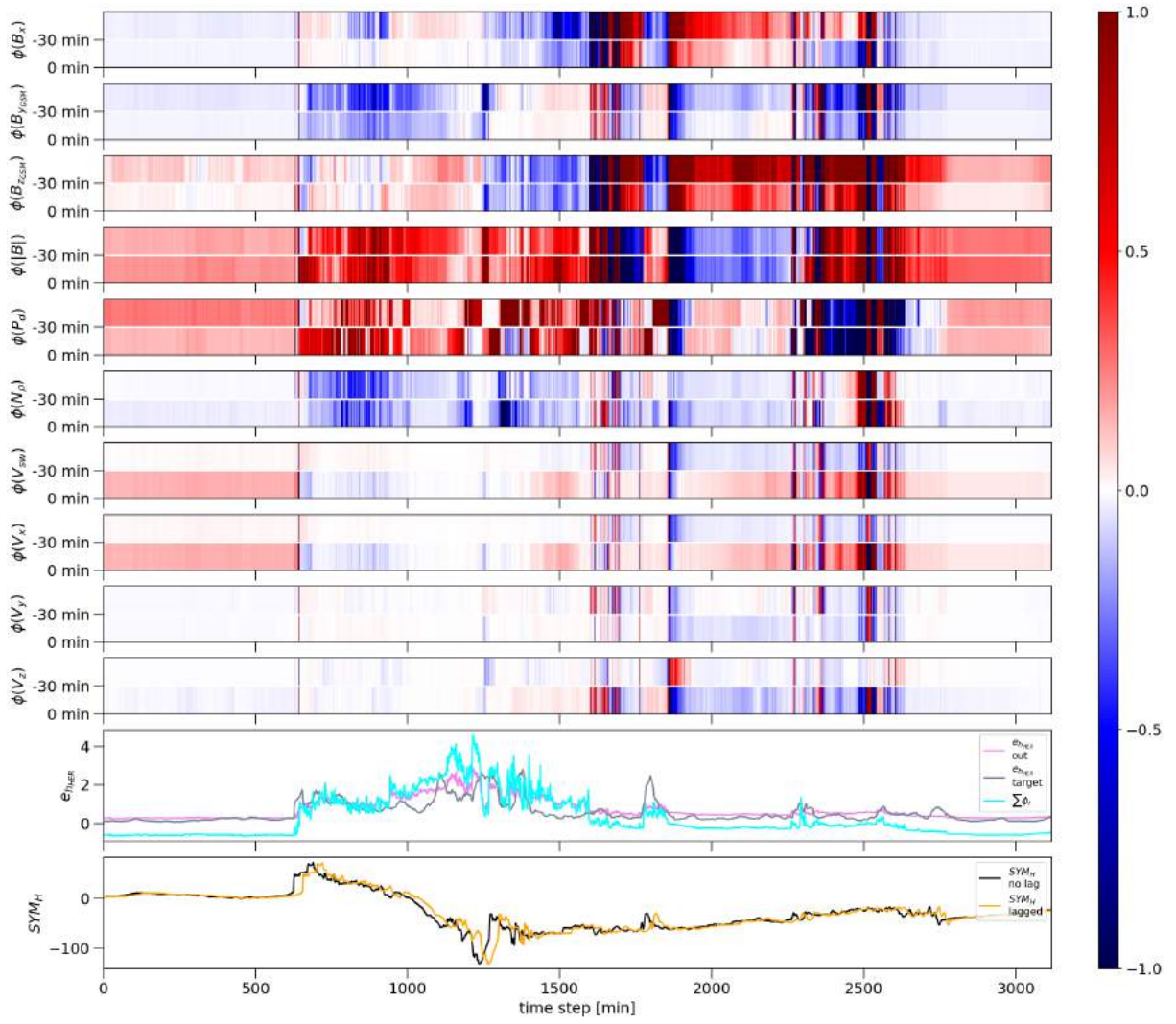
Attribution plot of all non auto-regressive input features for $e_{h_{ABK}}$ predicted during 2001.moderate.001 in the simultaneous e_h prediction task.



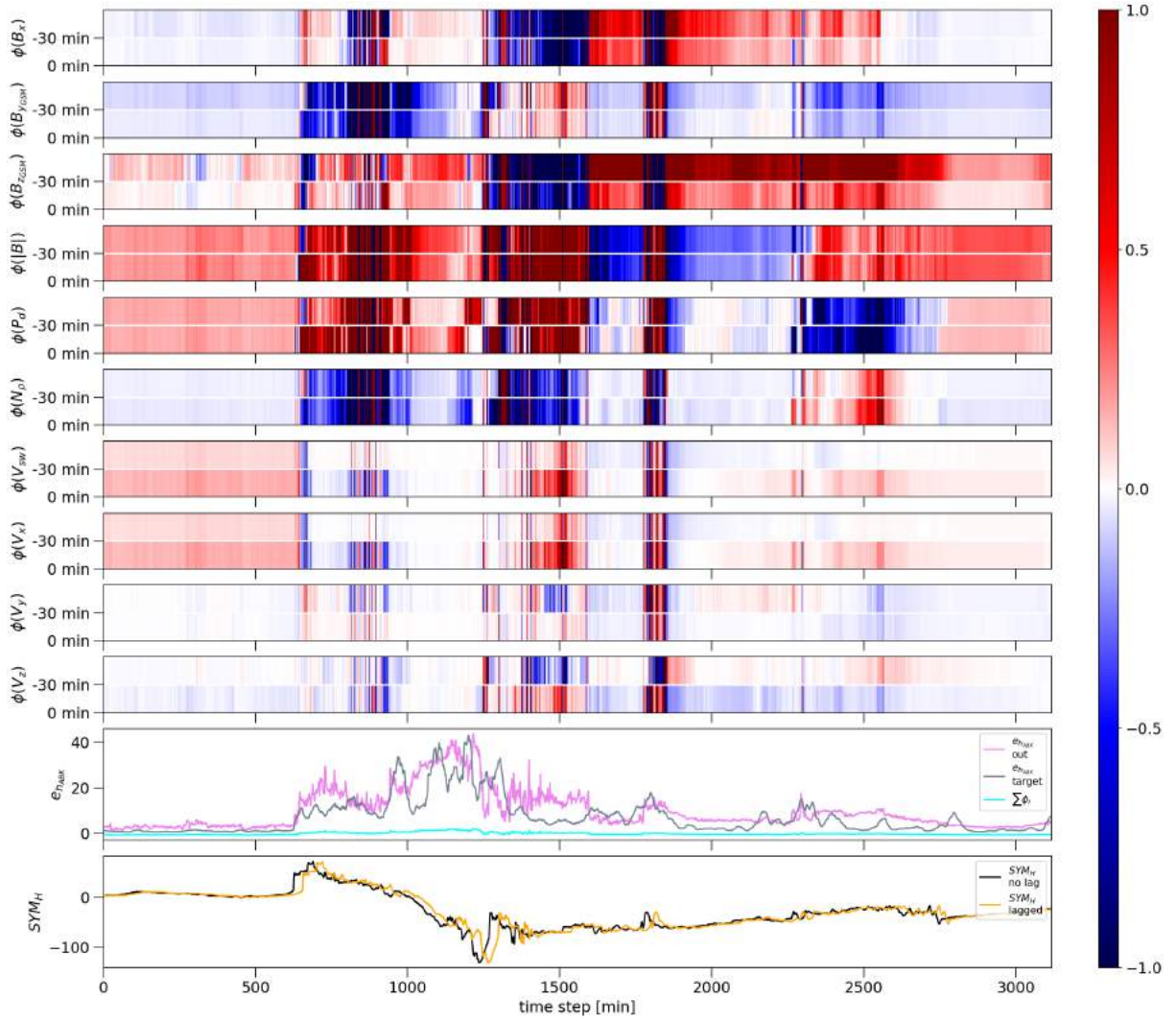
Attribution plot of all non auto-regressive input features for $e_{h_{HER}}$ predicted during 2001.severe.005 in the individual e_h prediction task.



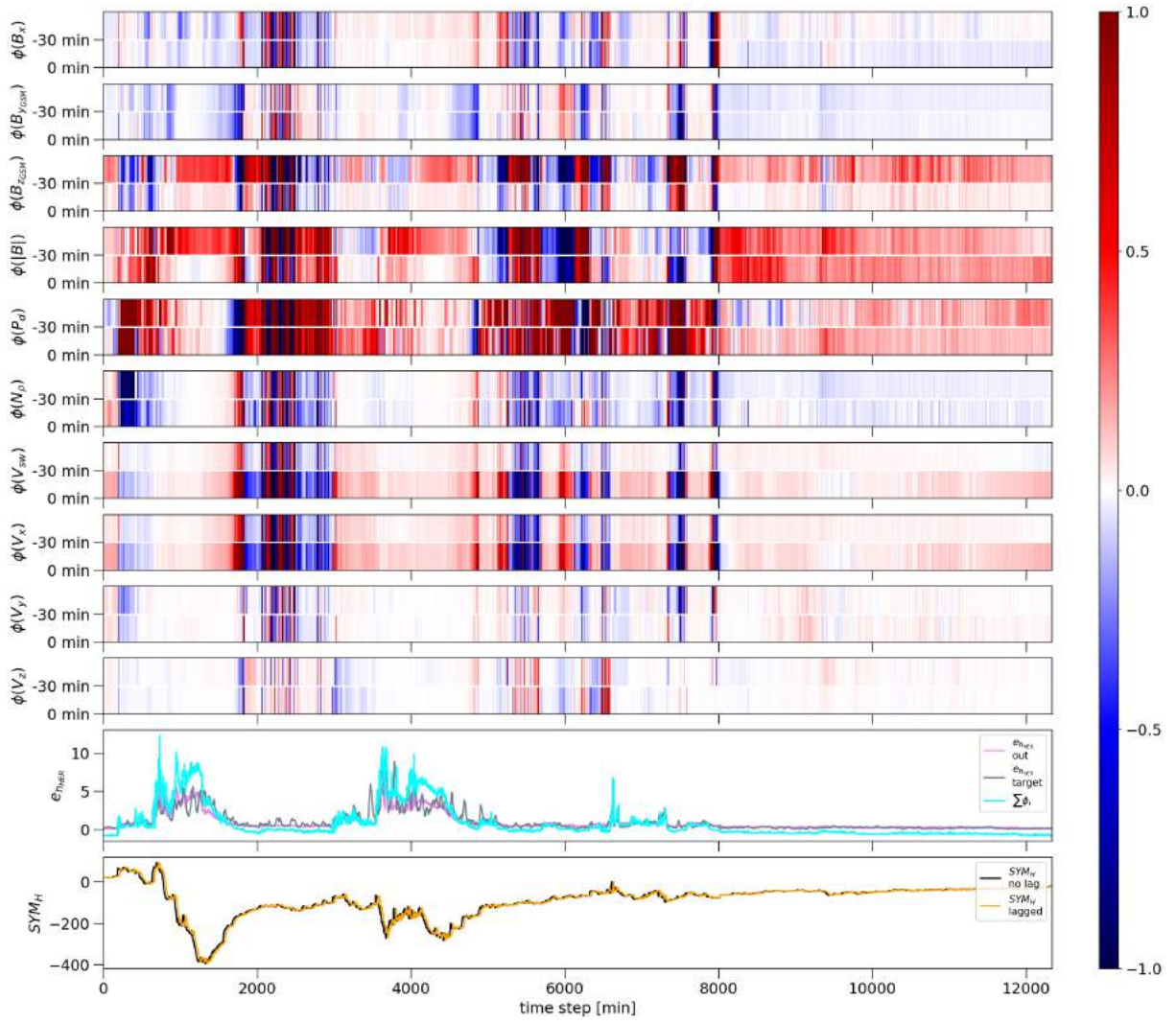
Attribution plot of all non auto-regressive input features for e_{hABK} predicted during 2001.severe.005 in the individual e_h prediction task.



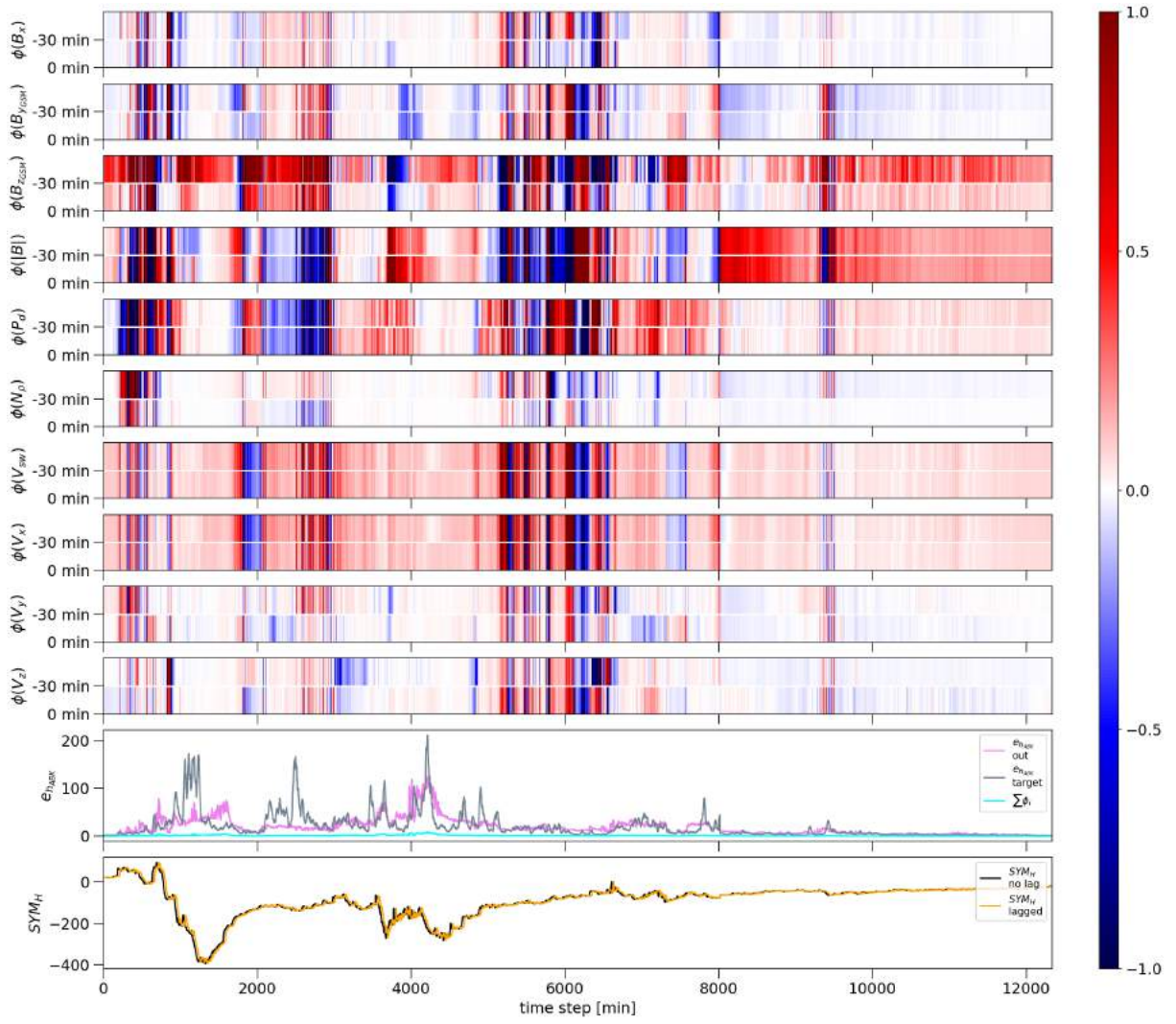
Attribution plot of all non auto-regressive input features for $e_{h_{HER}}$ predicted during 2001.severe.005 in the simultaneous e_h prediction task.



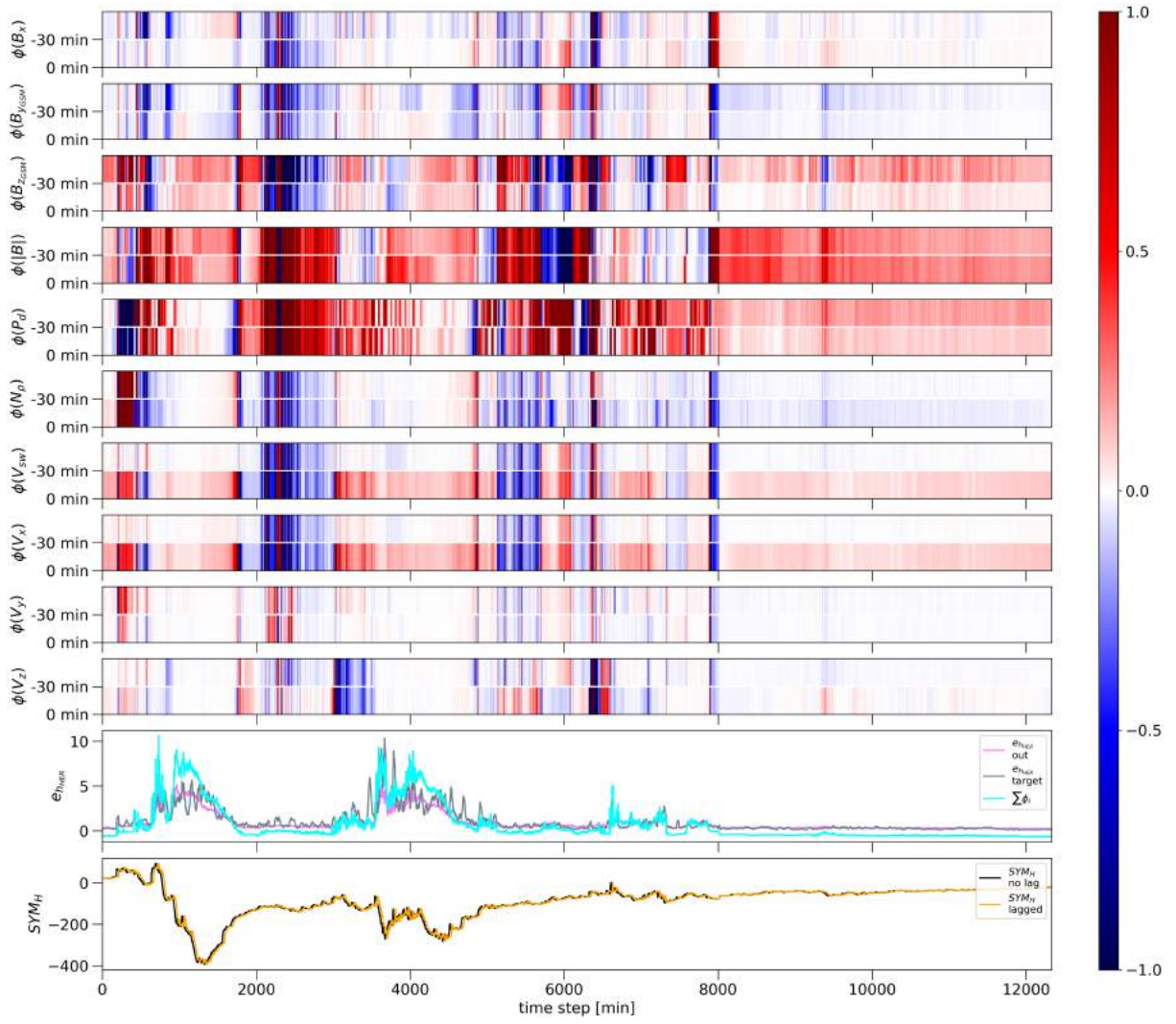
Attribution plot of all non auto-regressive input features for e_{hABK} predicted during 2001.severe.005 in the simultaneous e_h prediction task.



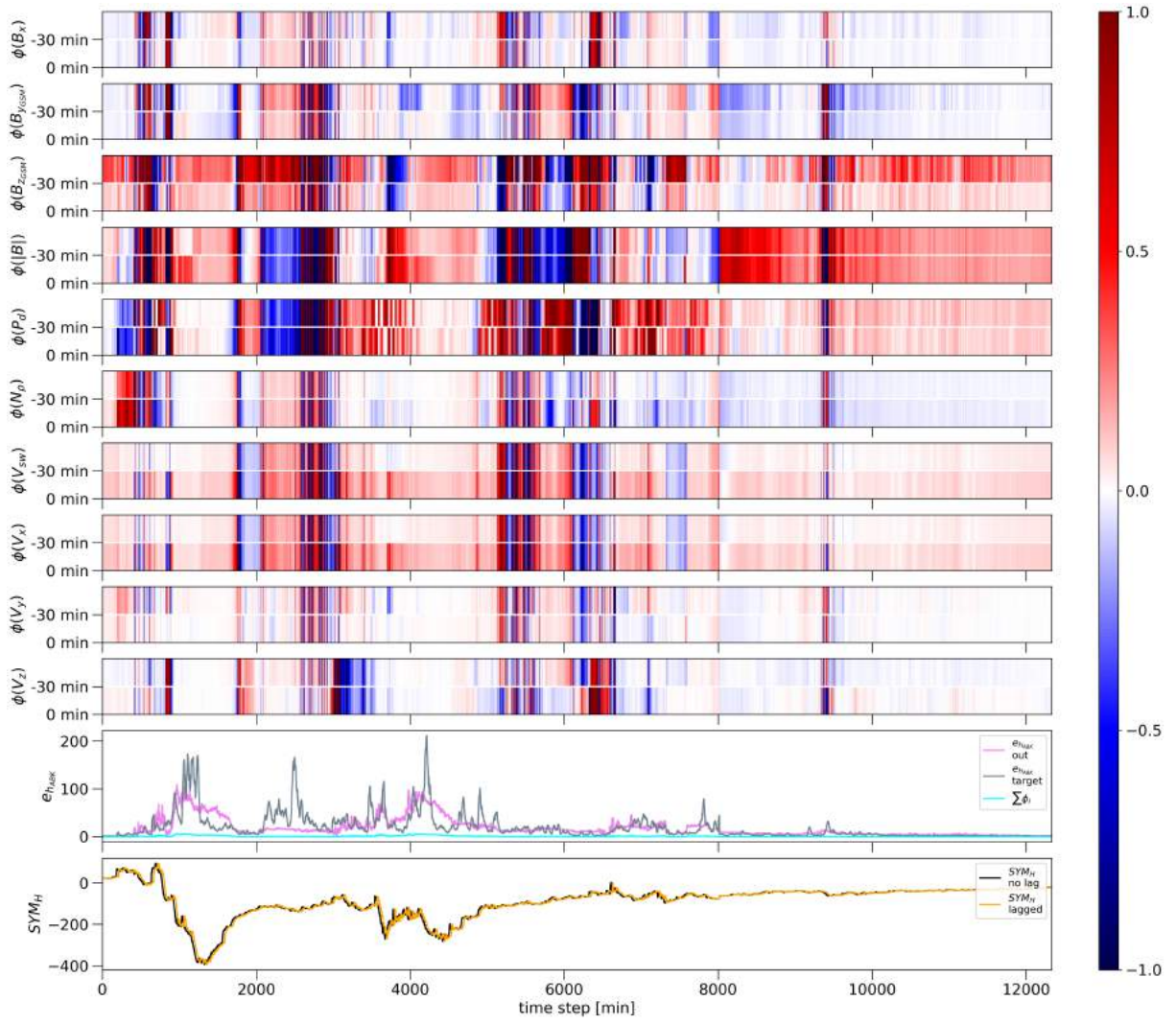
Attribution plot of all non auto-regressive input features for $e_{h_{HER}}$ predicted during 2004.severe.006 in the individual e_h prediction task.



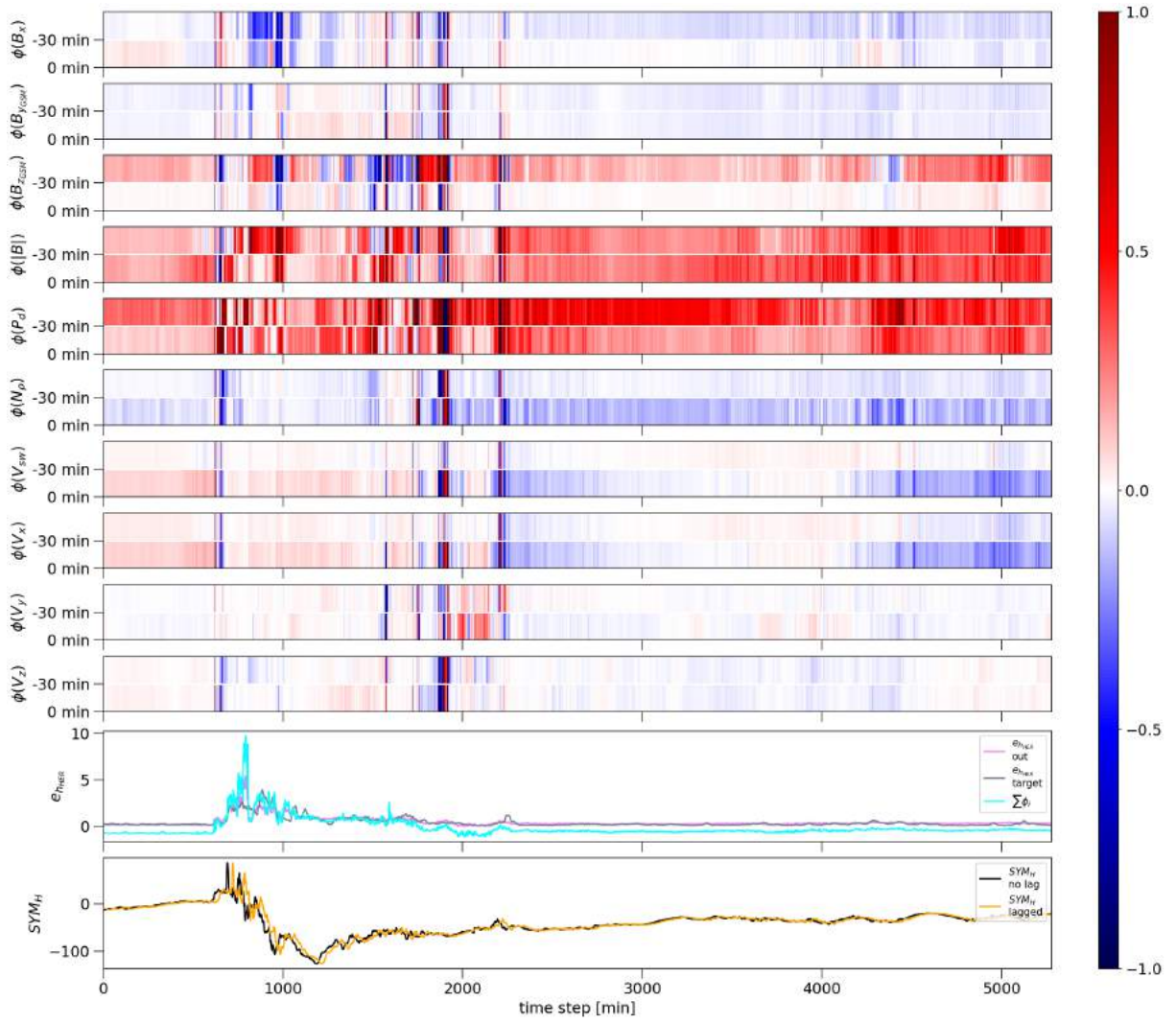
Attribution plot of all non auto-regressive input features for $e_{h_{ABK}}$ predicted during 2004.severe.006 in the individual e_h prediction task.



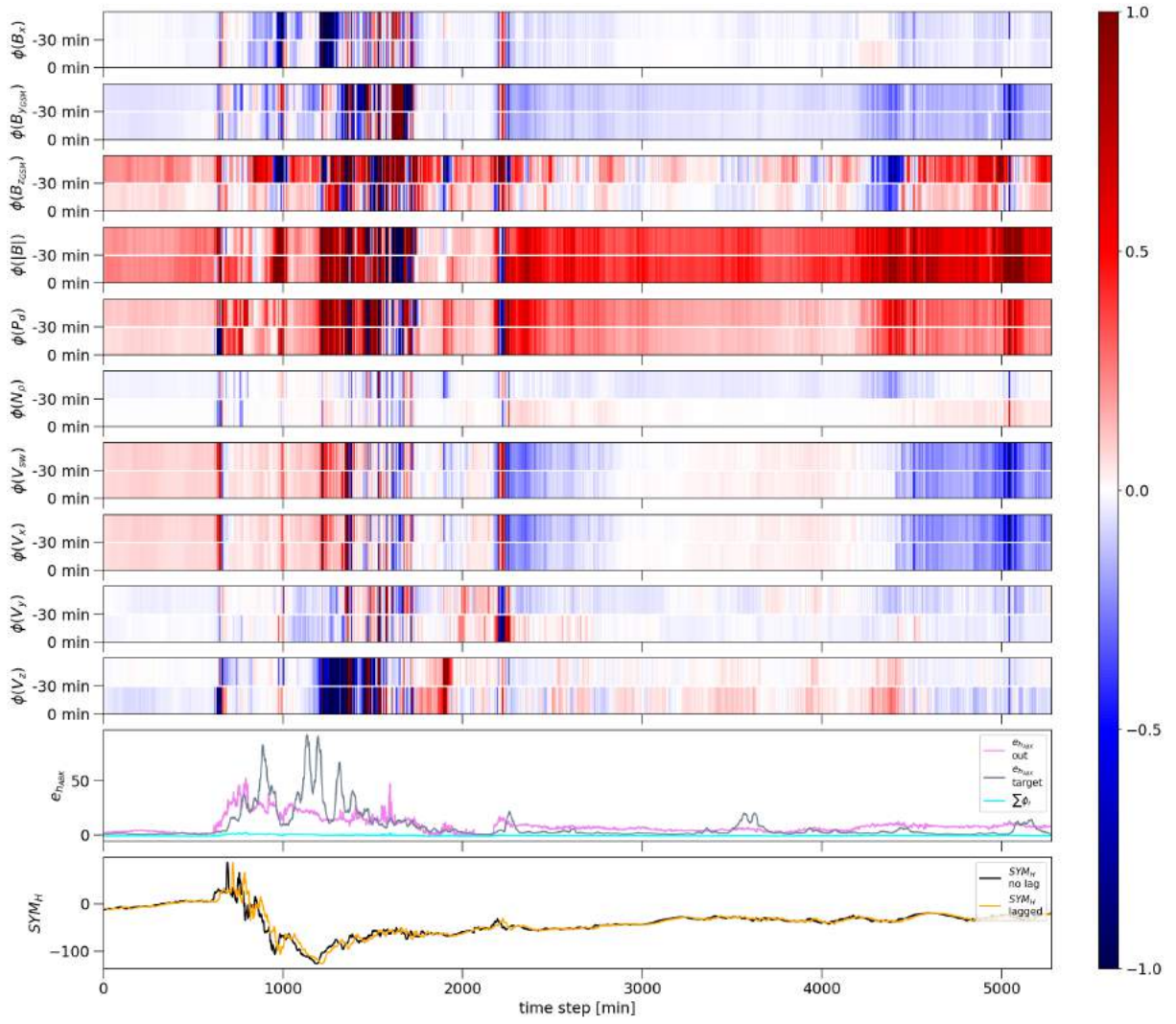
Attribution plot of all non auto-regressive input features for $e_{h_{HER}}$ predicted during 2004.severe.006 in the simultaneous e_h prediction task.



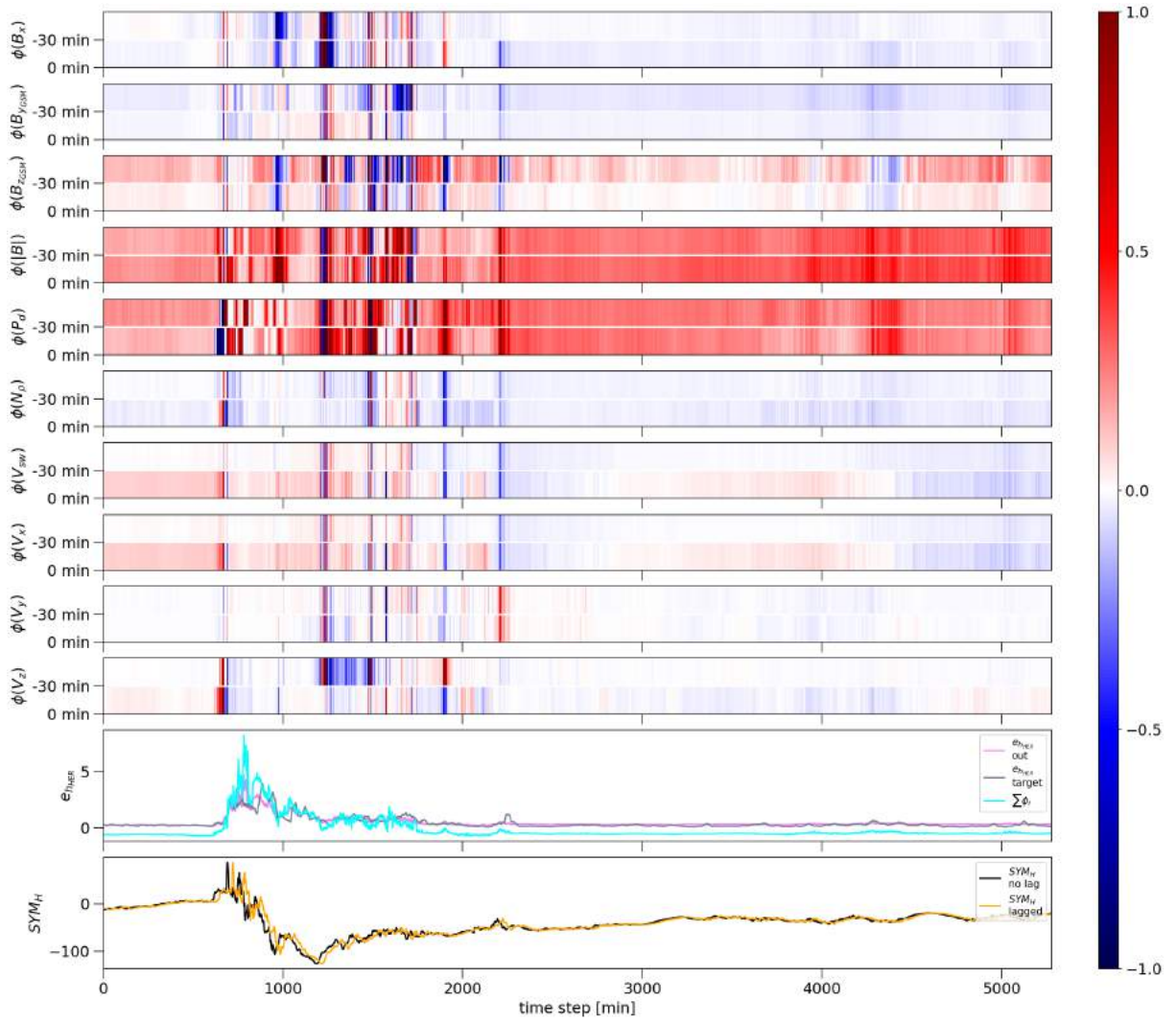
Attribution plot of all non auto-regressive input features for e_{hABK} predicted during 2004.severe.006 in the simultaneous e_h prediction task.



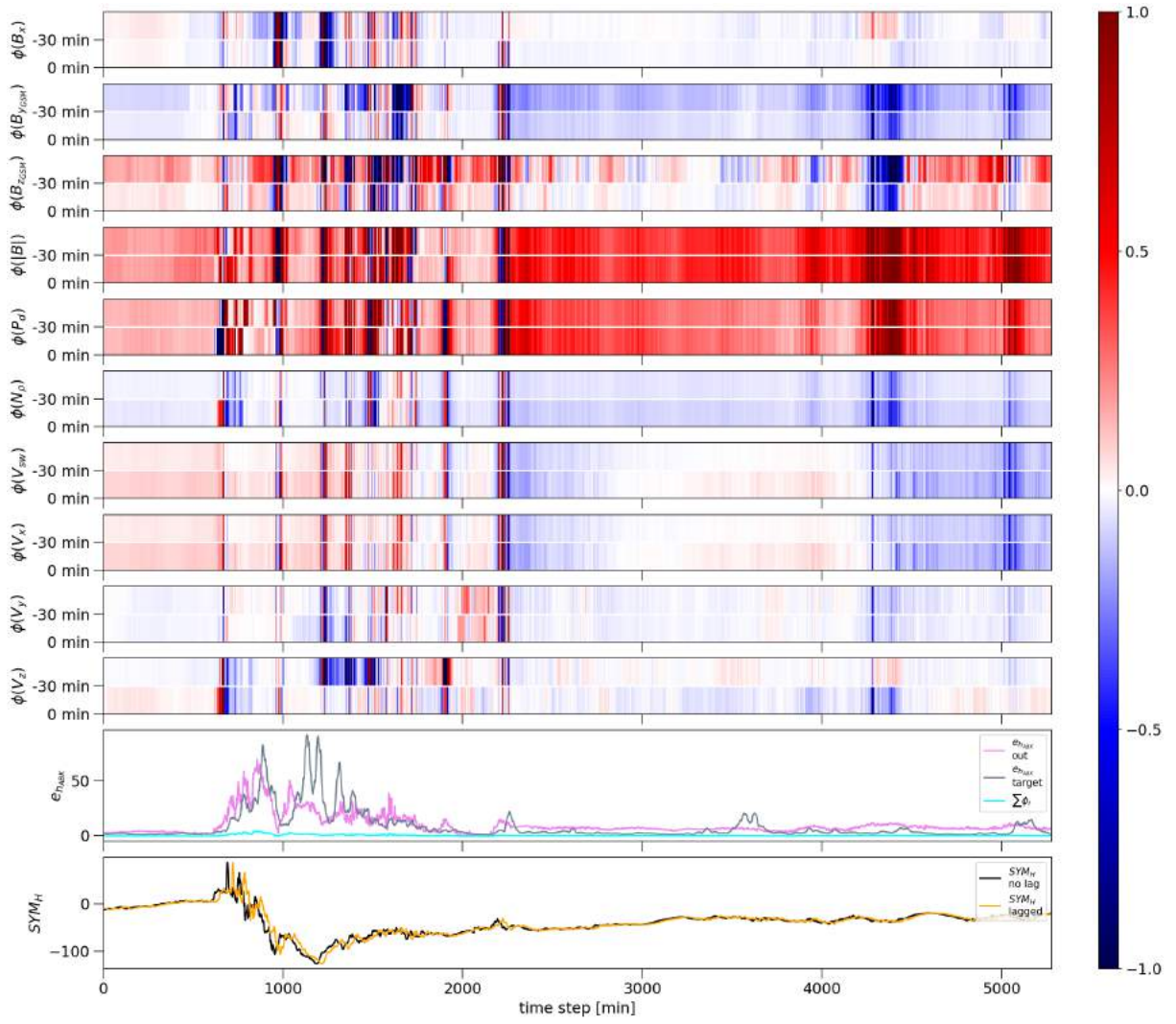
Attribution plot of all non auto-regressive input features for $e_{h_{HER}}$ predicted during 2011.severe.000 in the individual e_h prediction task.



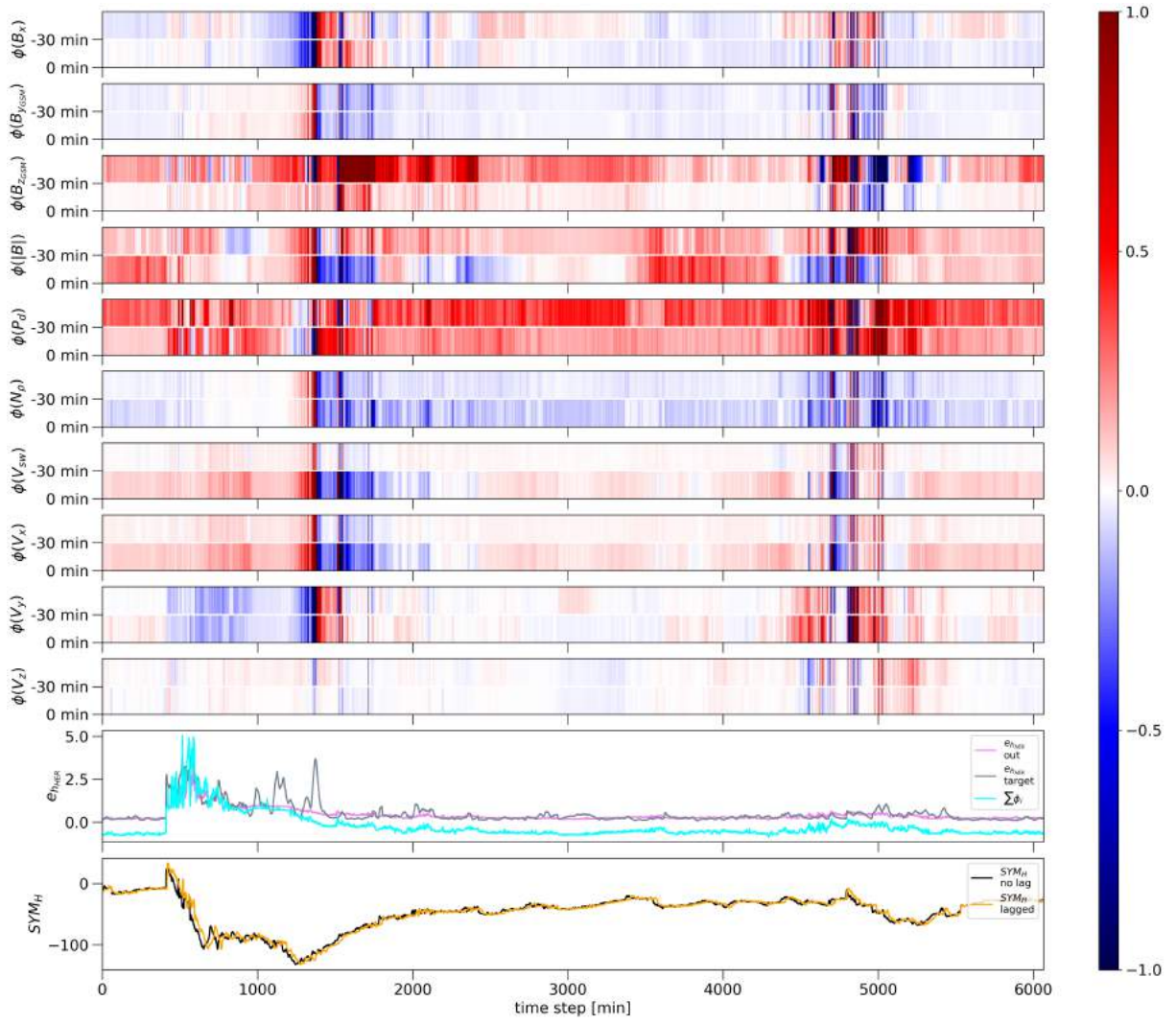
Attribution plot of all non auto-regressive input features for $e_{h_{ABK}}$ predicted during 2011.severe.000 in the individual e_h prediction task.



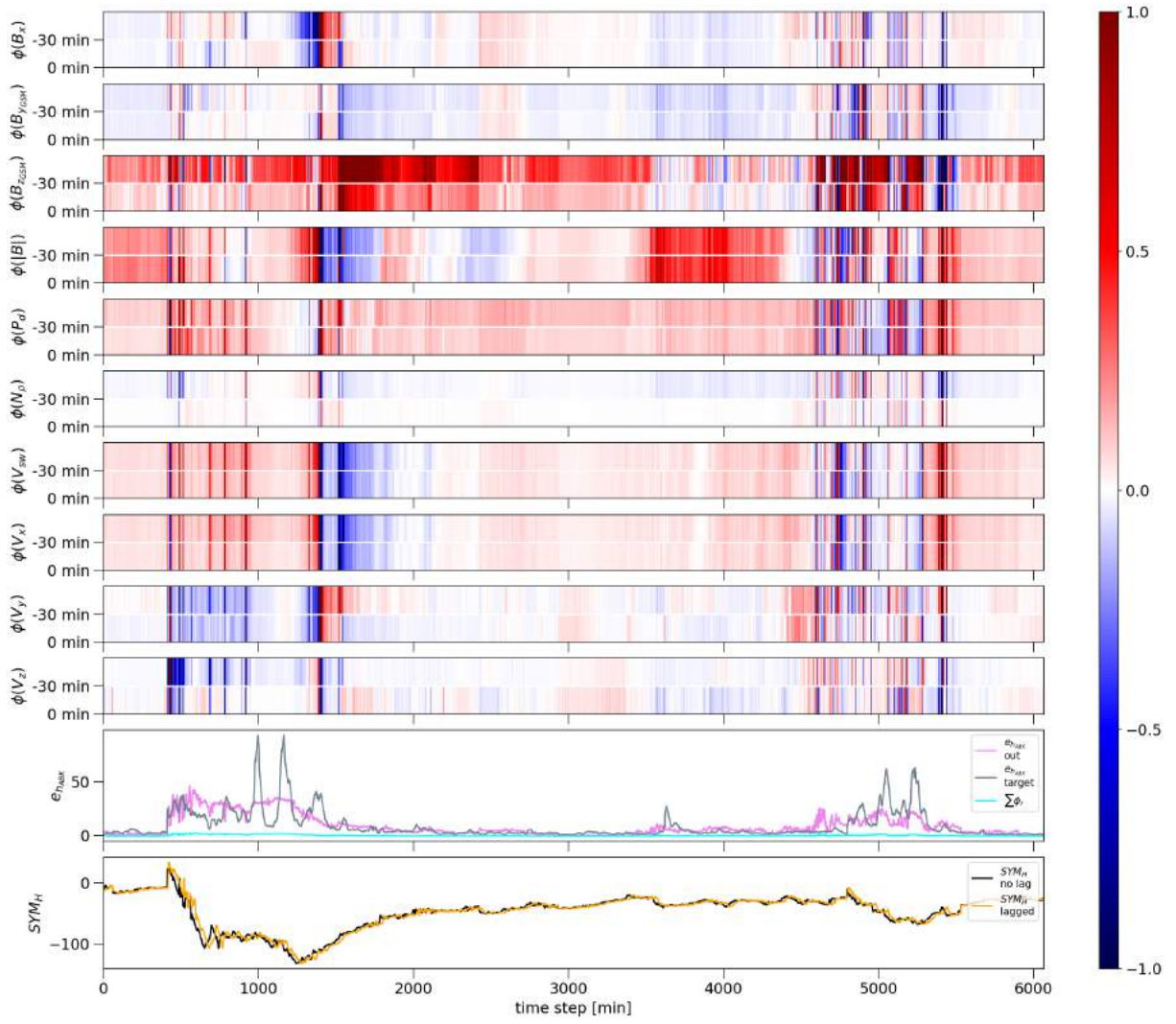
Attribution plot of all non auto-regressive input features for $e_{h_{HER}}$ predicted during 2011.severe.000 in the simultaneous e_h prediction task.



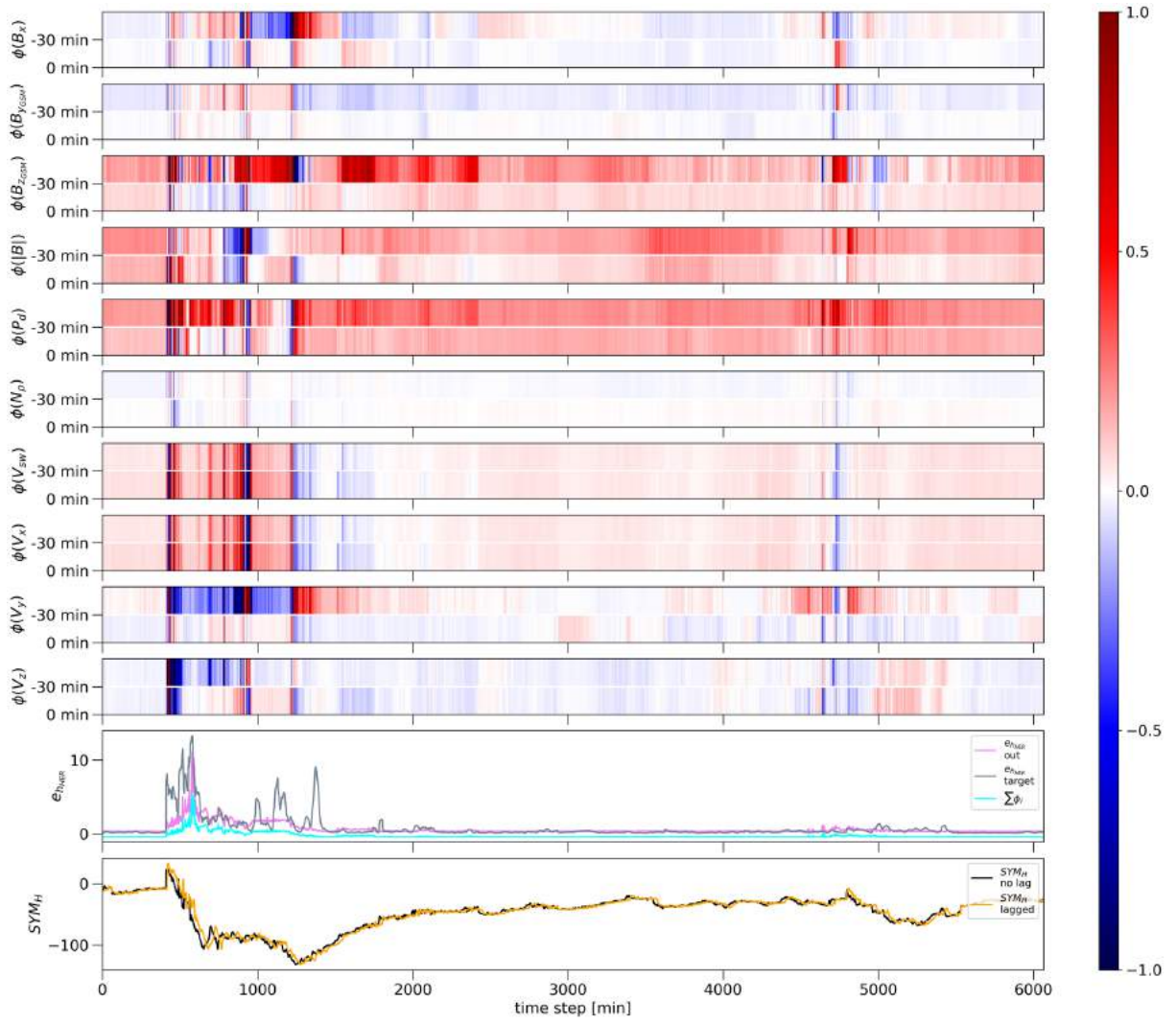
Attribution plot of all non auto-regressive input features for e_{hABK} predicted during 2011.severe.000 in the simultaneous e_h prediction task.



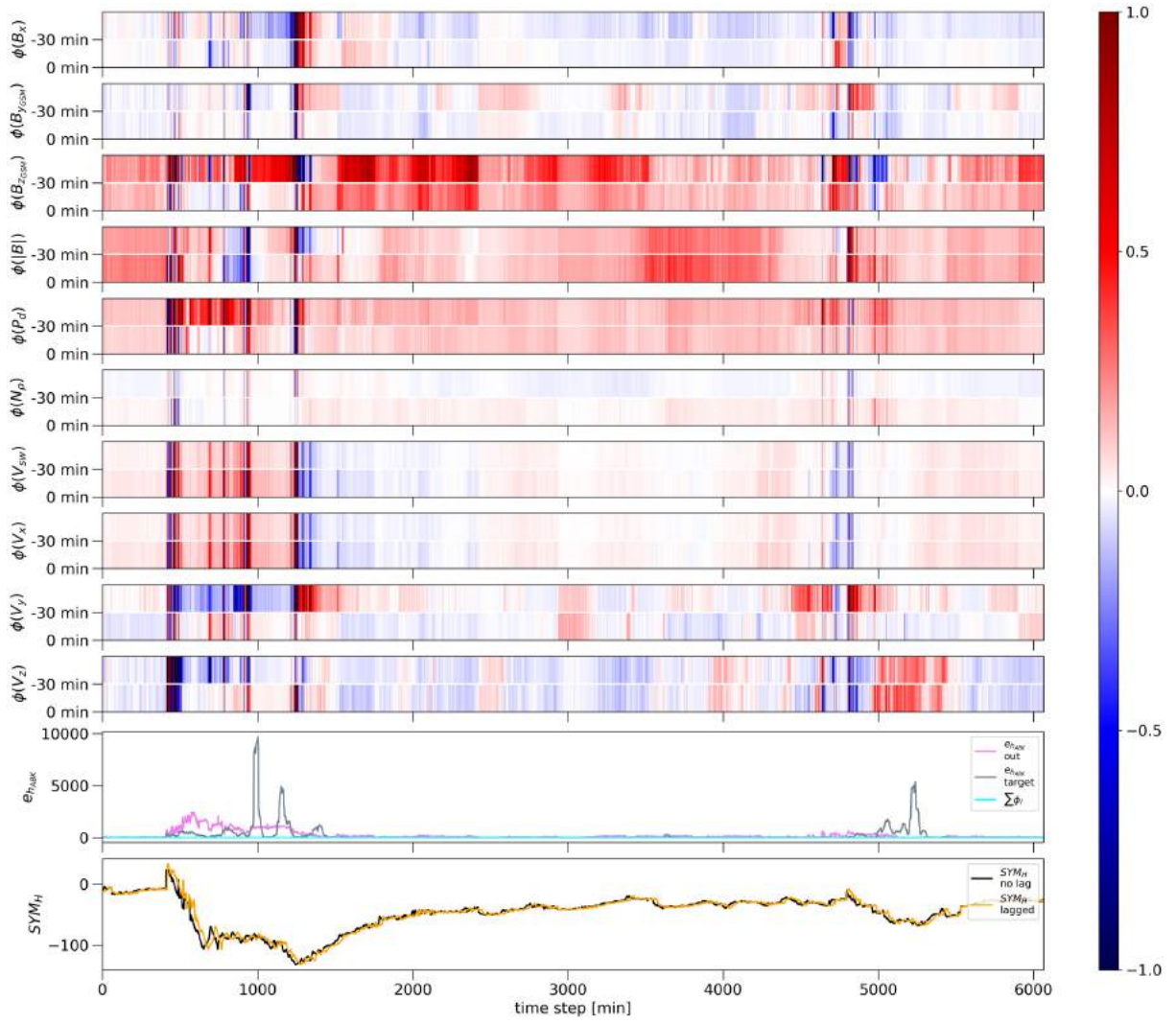
Attribution plot of all non auto-regressive input features for $e_{h_{HER}}$ predicted during 2013.severe.000 in the individual e_h prediction task.



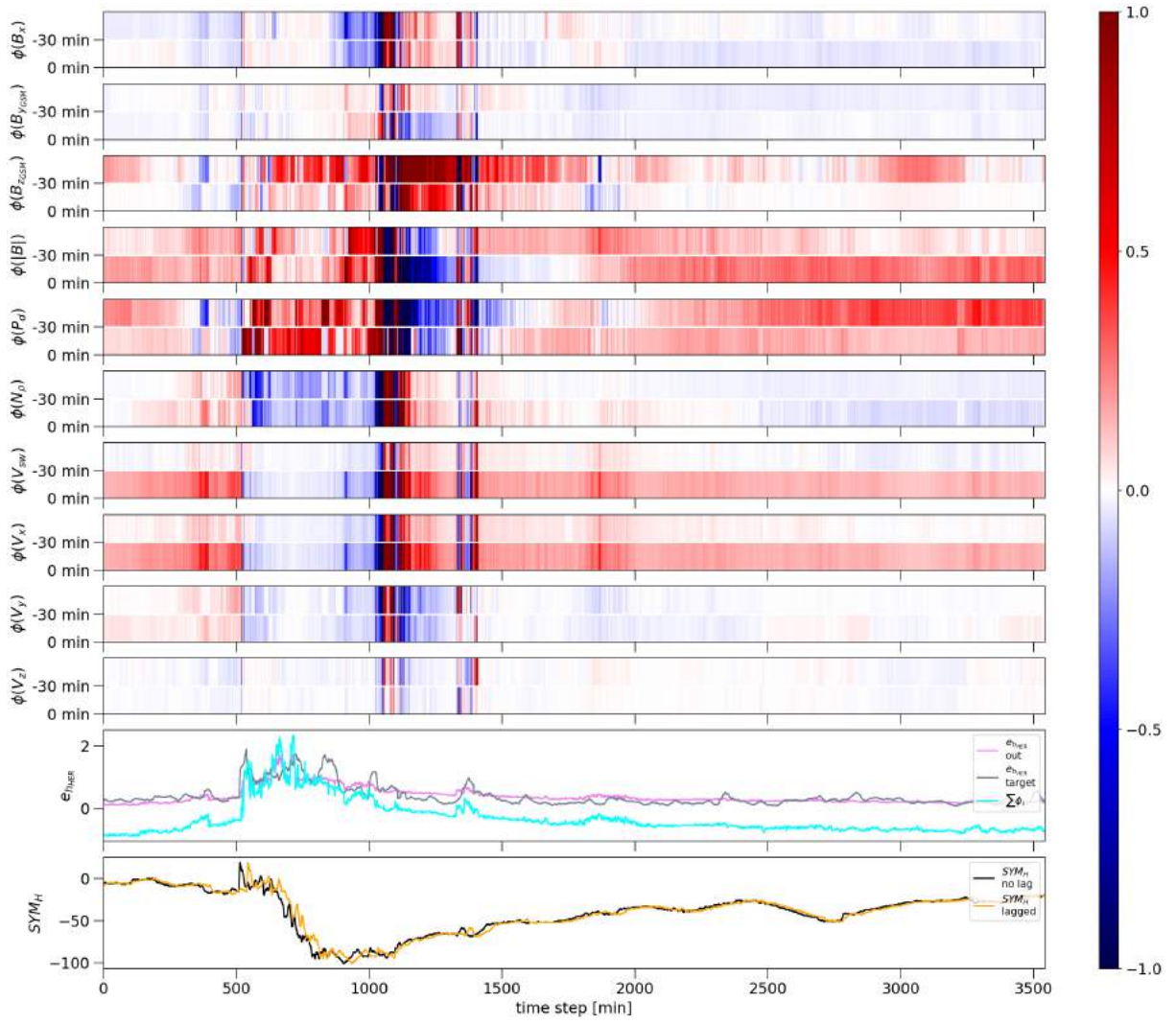
Attribution plot of all non auto-regressive input features for $e_{h_{ABK}}$ predicted during 2013.severe.000 in the individual e_h prediction task.



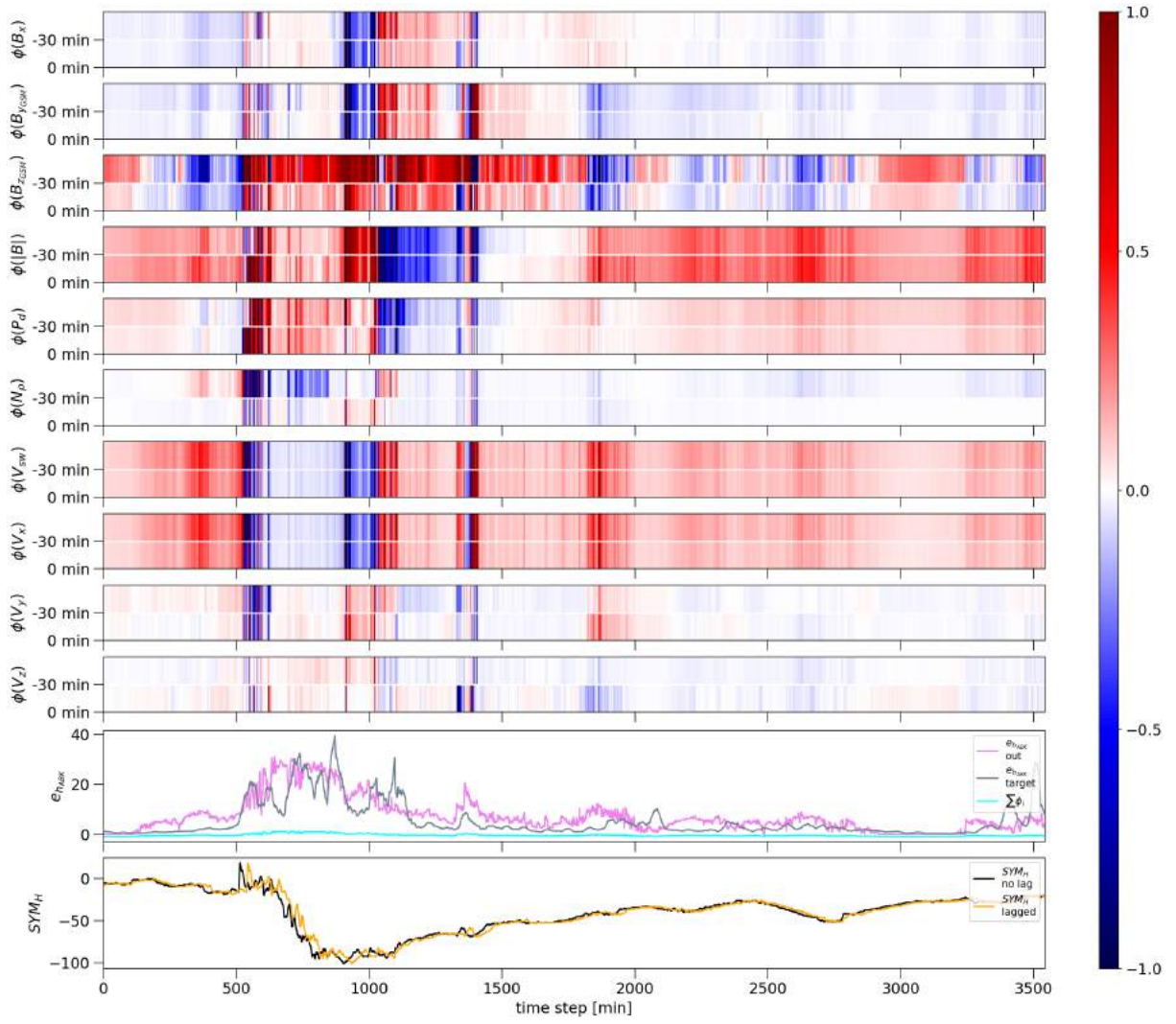
Attribution plot of all non auto-regressive input features for $e_{h_{HER}}$ predicted during 2013.severe.000 in the simultaneous e_h prediction task.



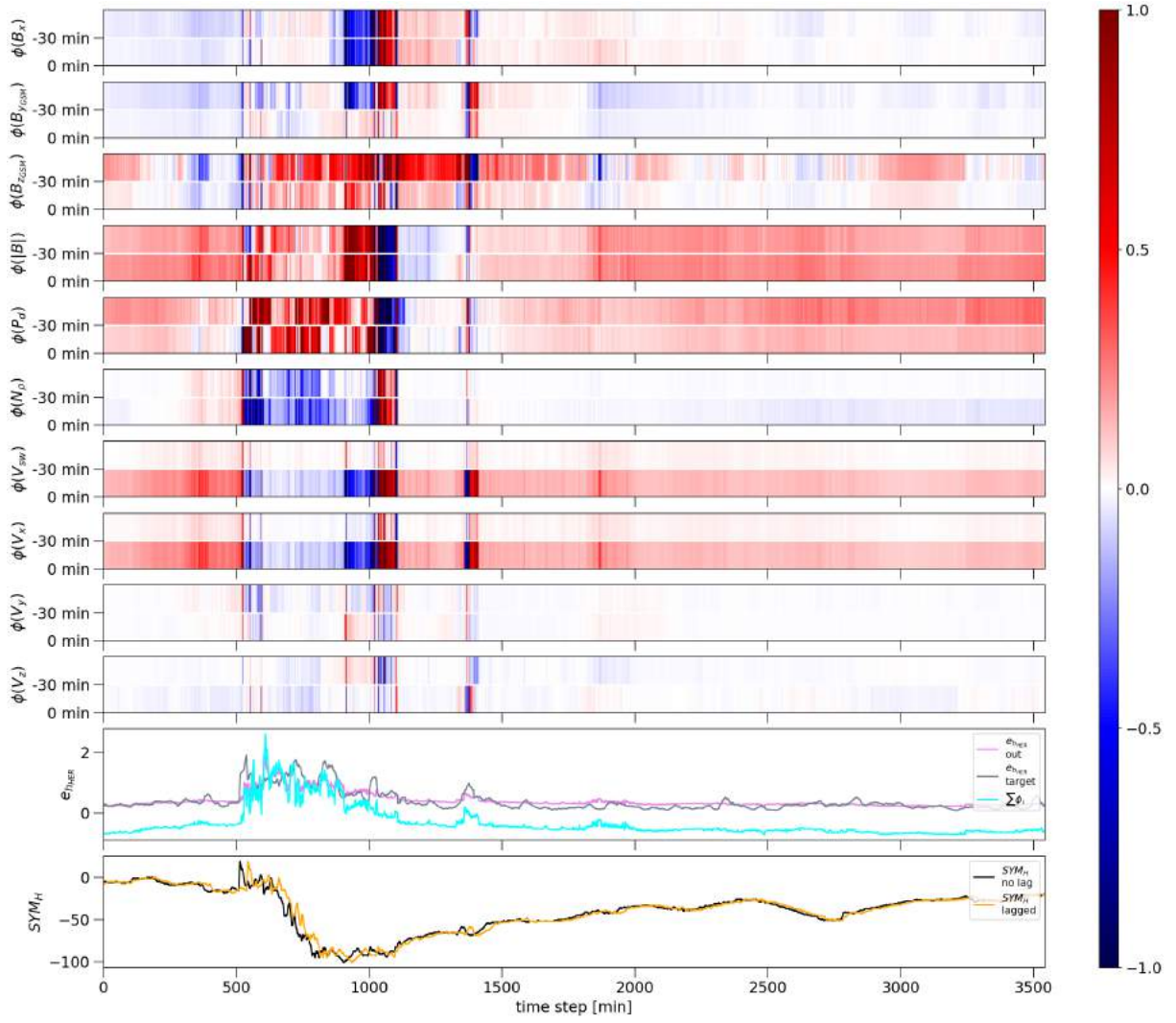
Attribution plot of all non auto-regressive input features for e_{hABK} predicted during 2013.severe.000 in the simultaneous e_h prediction task.



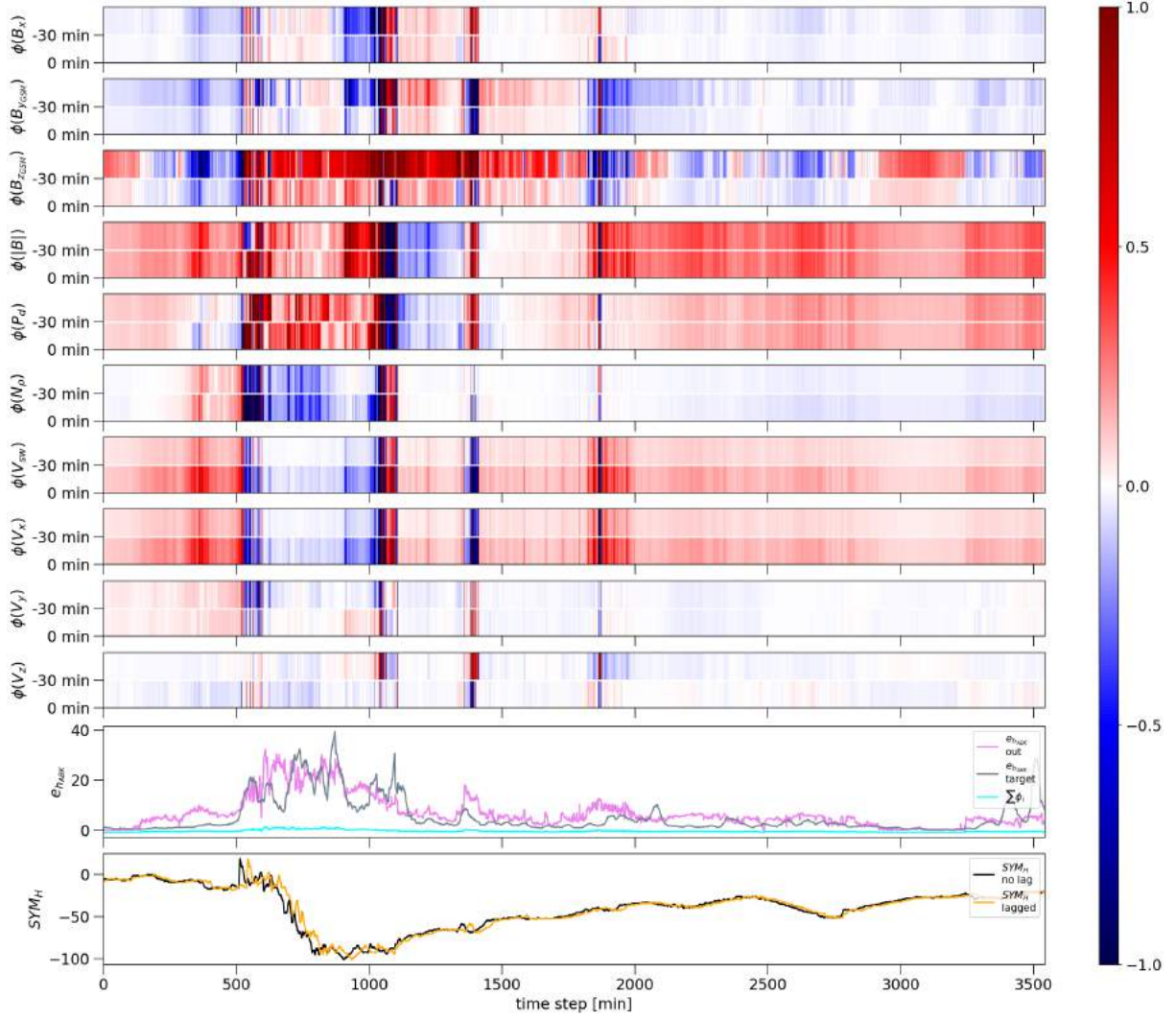
Attribution plot of all non auto-regressive input features for $e_{h_{HER}}$ predicted during 2014.severe.001 in the individual e_h prediction task.



Attribution plot of all non auto-regressive input features for e_{hABK} predicted during 2014_severe_001 in the individual e_h prediction task.



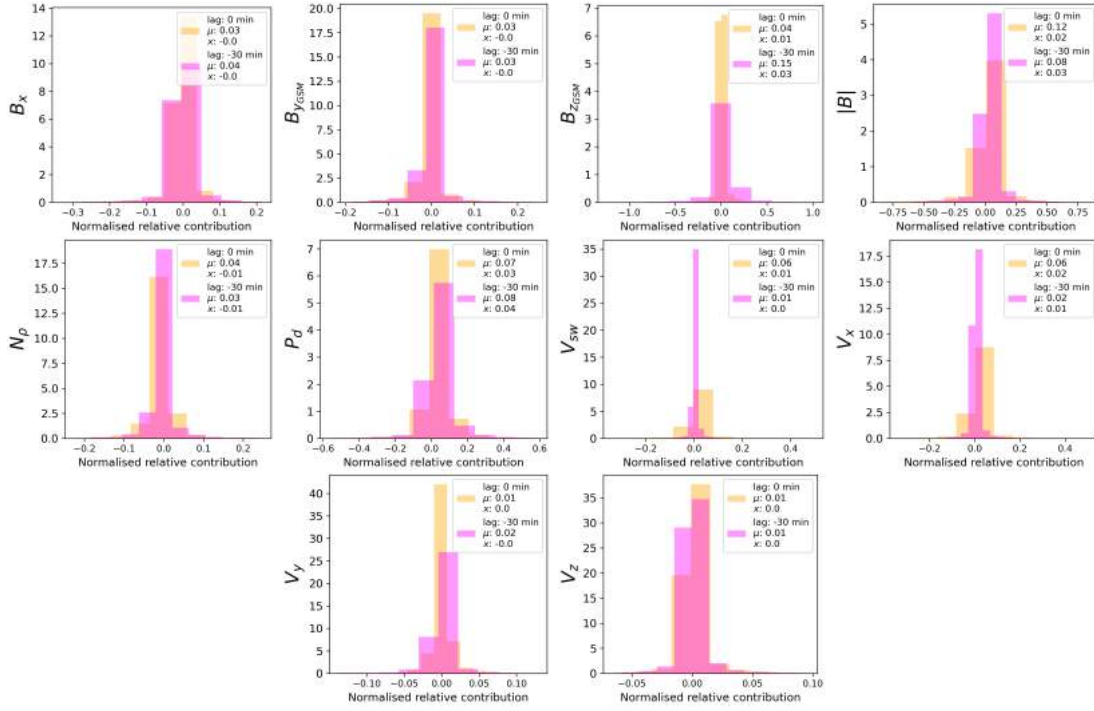
Attribution plot of all non auto-regressive input features for $e_{h_{HER}}$ predicted during 2014_severe_001 in the simultaneous e_h prediction task.



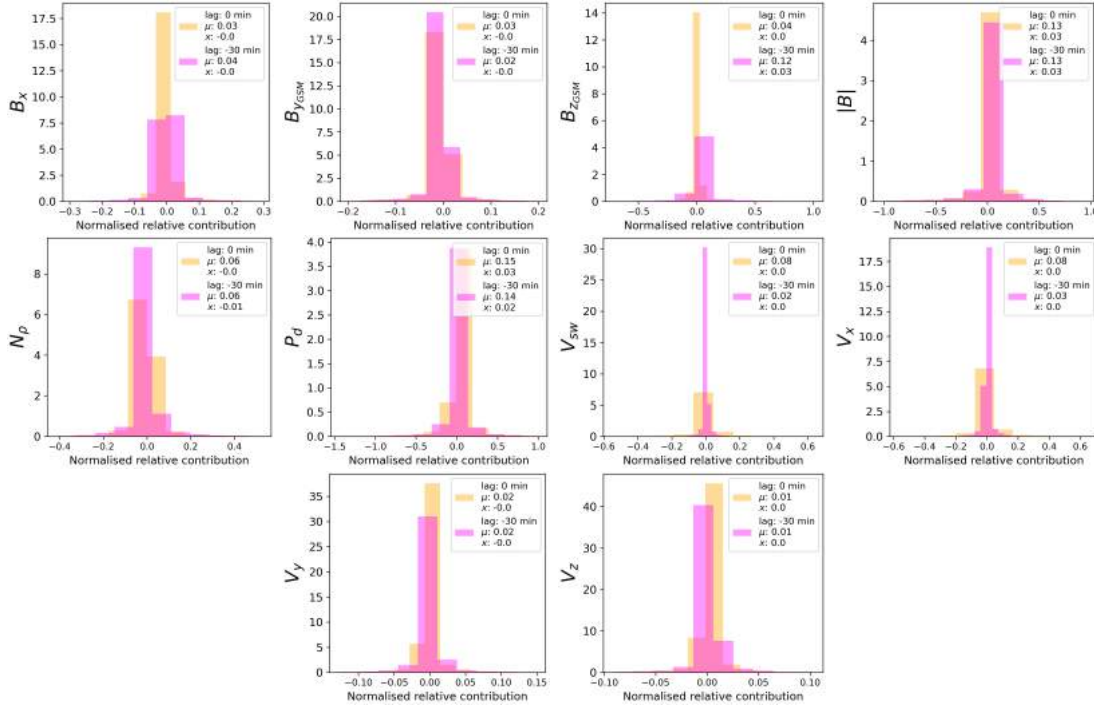
Attribution plot of all non auto-regressive input features for $e_{h_{ABK}}$ predicted during 2014_severe_001 in the simultaneous e_h prediction task.

A.4.4 Aggregate relative contribution distributions

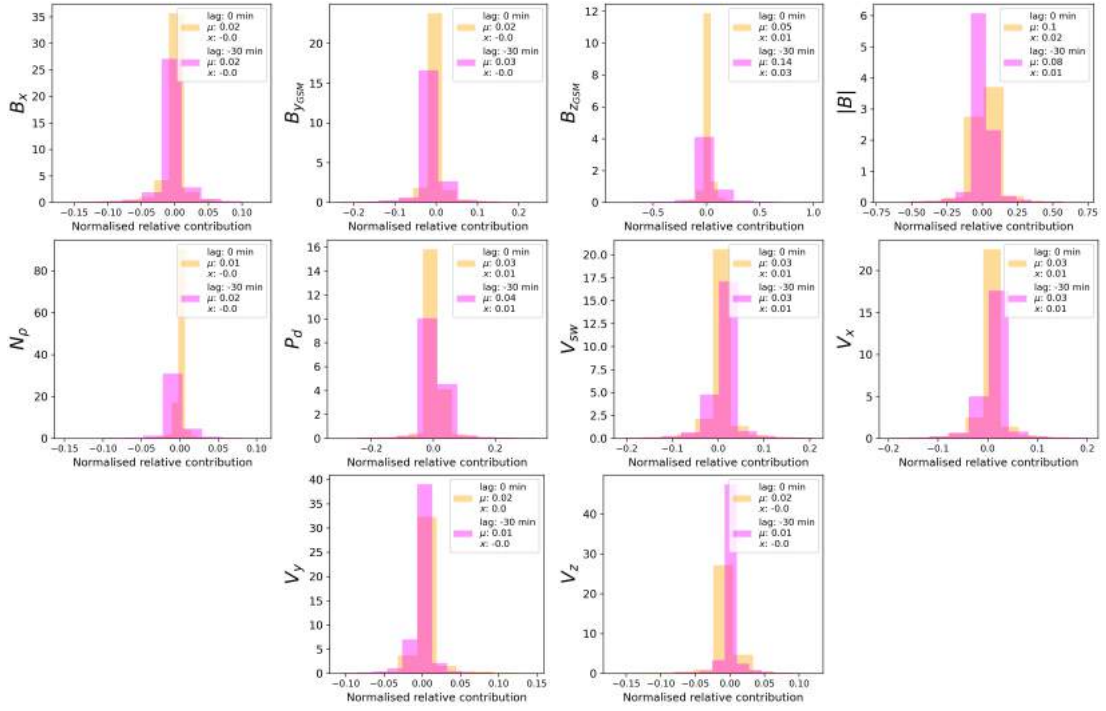
Here the mean relative contribution of each input feature to the output for all four tasks (with auto-regressive input features excluded) are displayed, alongside the *distributions* of the relative contributions of the input features in each case. This is done separately for the aggregation of selected CME-driven storms and the aggregation of selected CIR-driven storms.



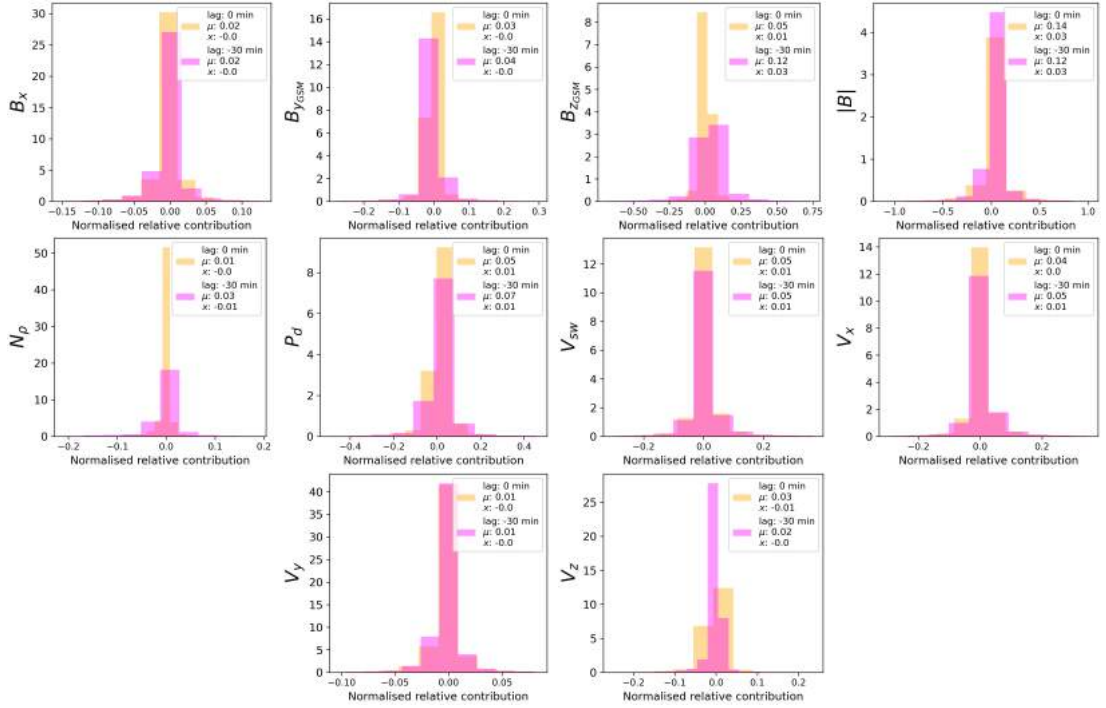
Distributions of relative contribution to the output of each feature on the e_{hHER} prediction task for aggregated CME-driven storms.



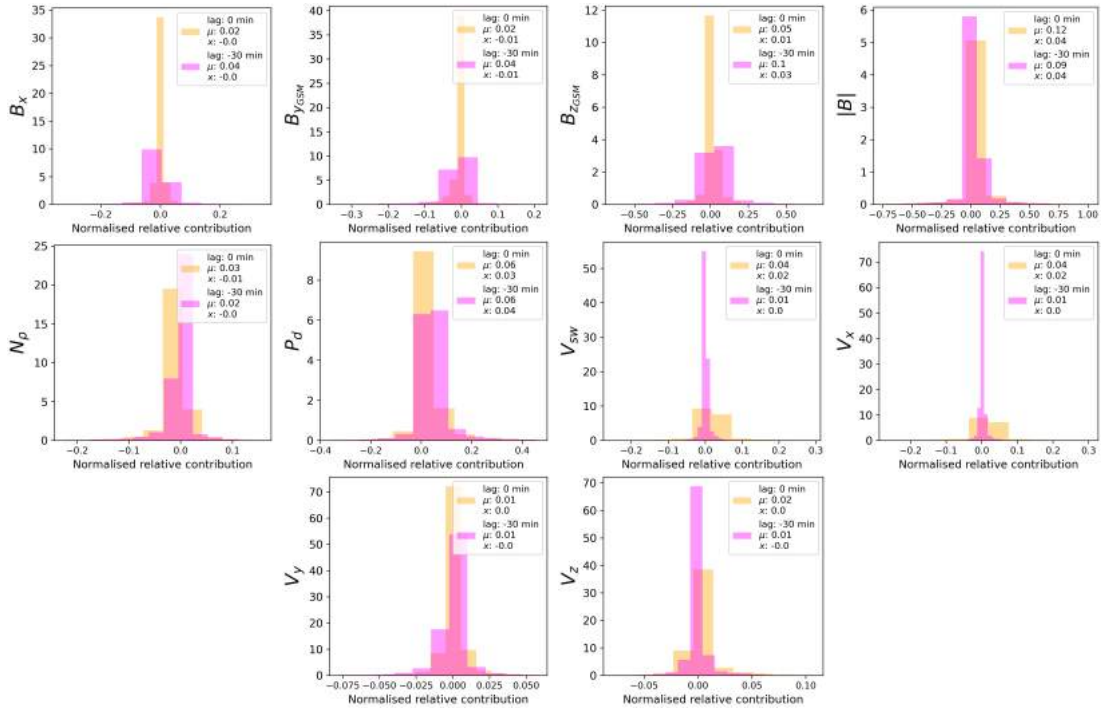
Distributions of relative contribution to the output of each feature on the e_{hHER} prediction task for aggregated CIR-driven storms.



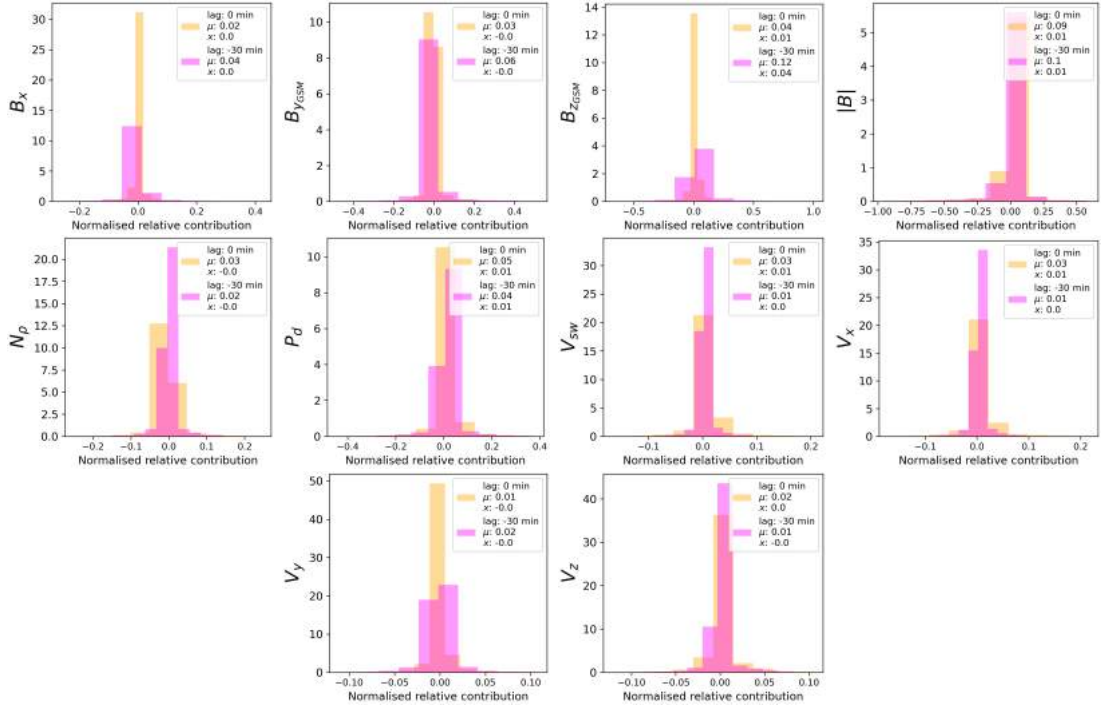
Distributions of relative contribution to the output of each feature on the $e_{h_{ABK}}$ prediction task for aggregated CME-driven storms.



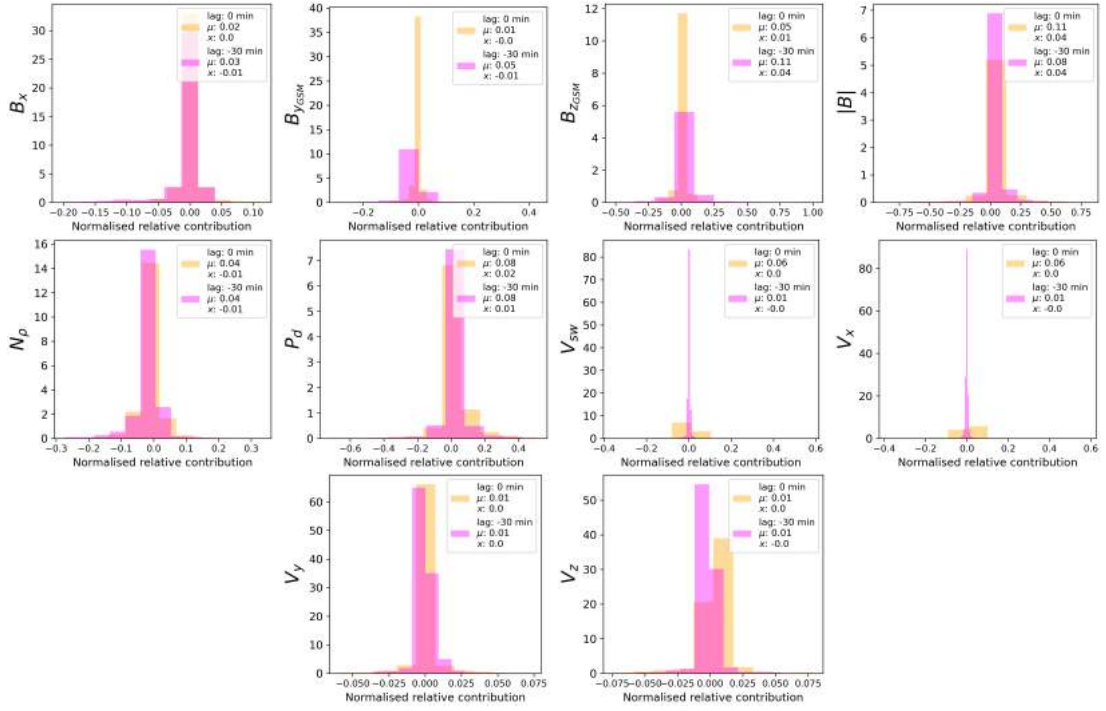
Distributions of relative contribution to the output of each feature on the $e_{h_{ABK}}$ prediction task for aggregated CIR-driven storms.



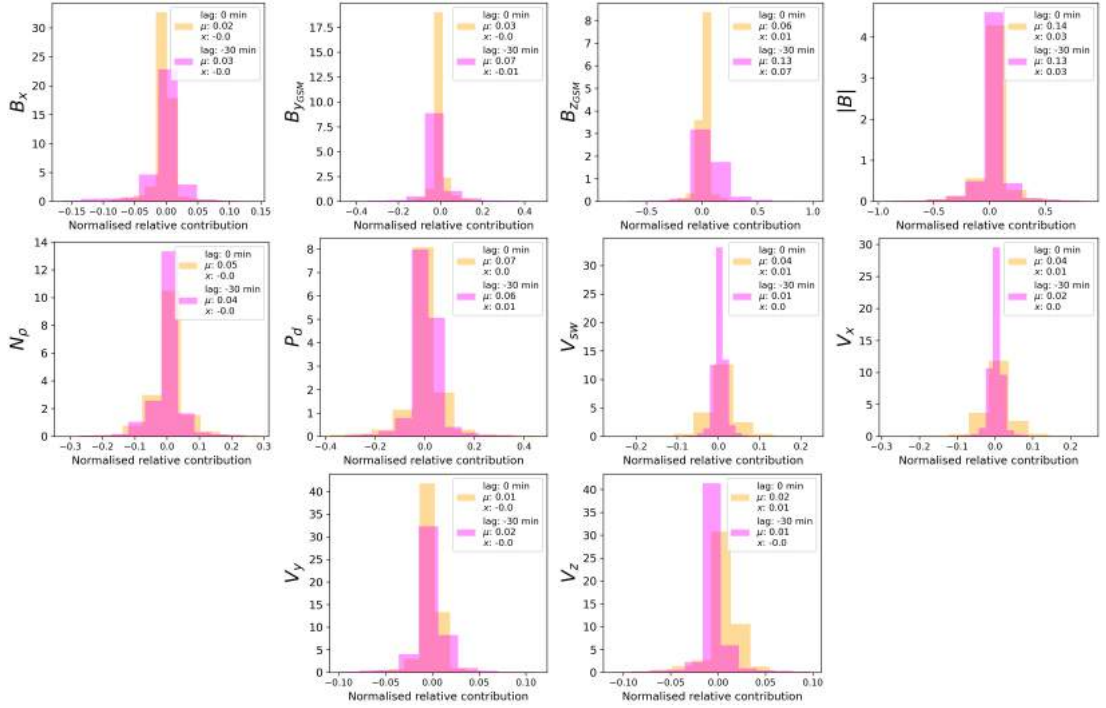
Distributions of relative contribution to the output of each feature on the simultaneous e_{hHER} and e_{hABK} prediction task for aggregated CME-driven storms, for the output e_{hHER} .



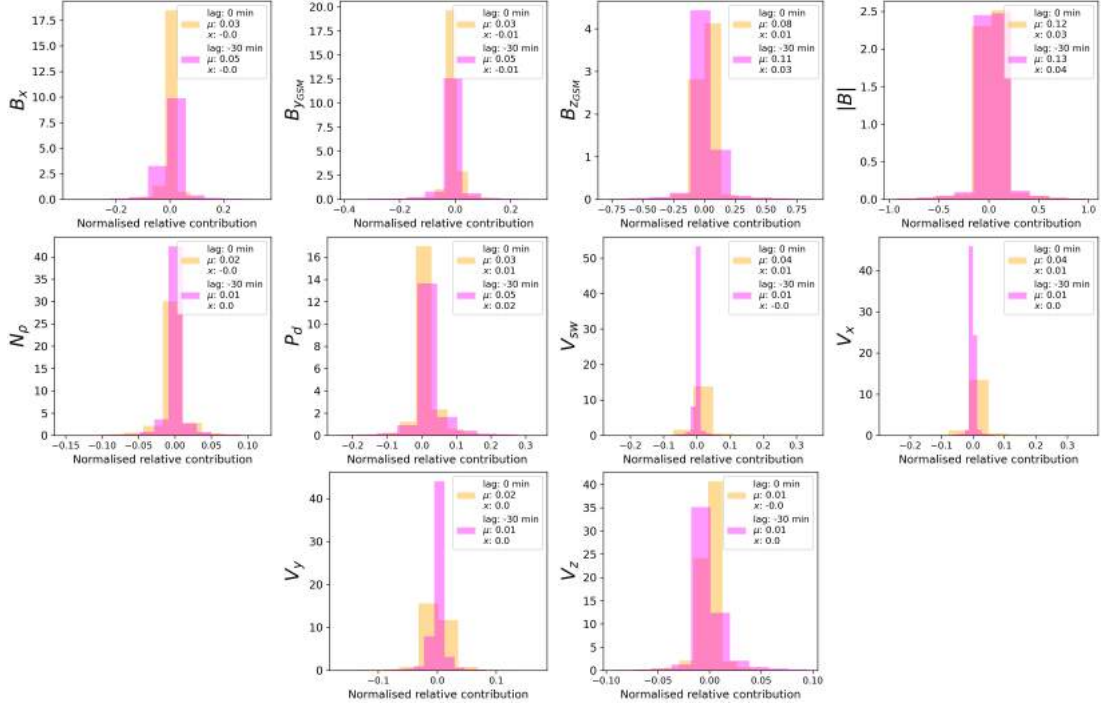
Distributions of relative contribution to the output of each feature on the simultaneous e_{hHER} and e_{hABK} prediction task for aggregated CME-driven storms, for the output e_{hABK} .



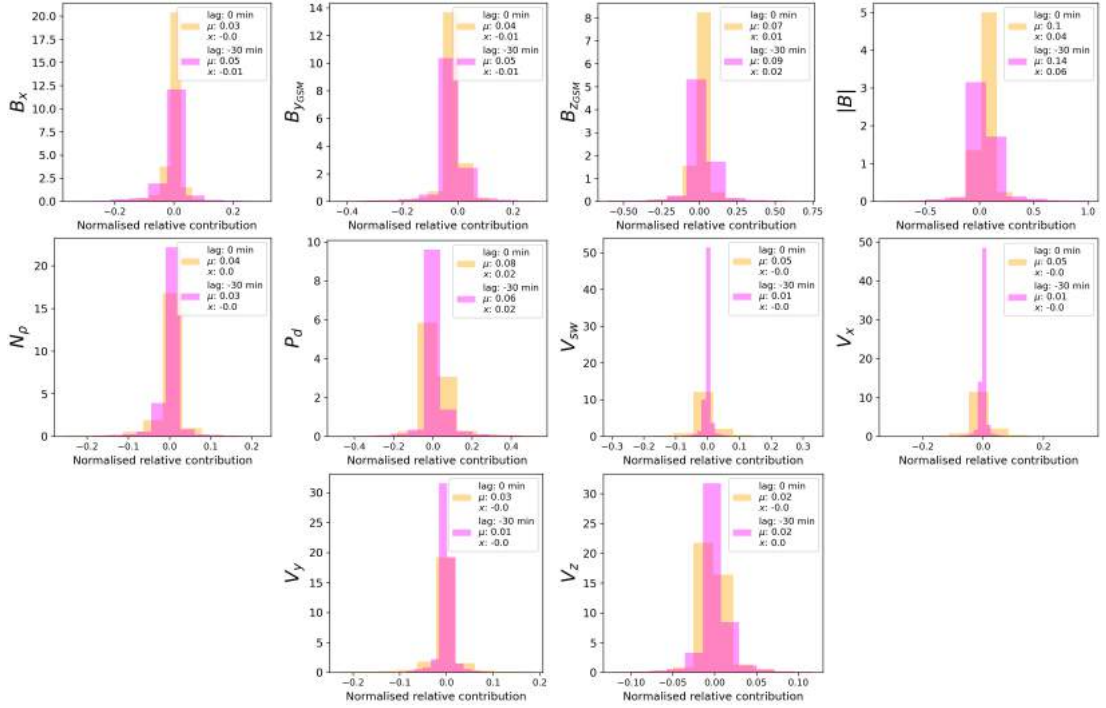
Distributions of relative contribution to the output of each feature on the simultaneous e_{hHER} and e_{hABK} prediction task for aggregated CIR-driven storms, for the output e_{hHER} .



Distributions of relative contribution to the output of each feature on the simultaneous e_{hHER} and e_{hABK} prediction task for aggregated CIR-driven storms, for the output e_{hABK} .



Distributions of relative contribution to the output of each feature on the $|\Delta e_h|$ prediction task for aggregated CME-driven storms.



Distributions of relative contribution to the output of each feature on the $|\Delta e_h|$ prediction task for aggregated CIR-driven storms.