CHALMERS
UNIVERSITY OF TECHNOLOGY

Anomaly Detection in Distributed Strain Sensing Data

A case study of machine learning applications of the Götaälv bridge

Master's thesis in Structural Engineering and Building Technology

SIMON FLAGERUP
VIKTOR KJELLSSON

DEPARTMENT OF ARCHITECTURE AND CIVIL ENGINEERING
CHALMERS UNIVERSITY OF TECHNOLOGY
Gothenburg, Sweden 2025
www.chalmers.se

Master's thesis 2025

Anomaly Detection in Distributed Strain Sensing Data

A case study of machine learning applications of the Götaälv bridge

SIMON FLAGERUP
VIKTOR KJELLSSON

CHALMERS
UNIVERSITY OF TECHNOLOGY

Department of Architecture and Civil Engineering
Division of Structural Engineering
Chalmers University of Technology
Gothenburg, Sweden 2025

Anomaly Detection in Distributed Strain Sensing Data
A case study of machine learning applications of the Götaälv bridge
SIMON FLAGERUP
VIKTOR KJELLSSON

Supervisor: Farshid Zamiri, AFRY
Supervisor and examiner: Carlos Gil Berrocal, Department of Architecture and Civil Engineering

Master's Thesis 2025
Department of Architecture and Civil Engineering
Division of Structural Engineering
Chalmers University of Technology
412 96 Gothenburg
Telephone +46 31 772 1000

Cover: Picture of the Götaälv bridge. Photo Farshid Zamiri.

Typeset in LaTeX
Printed by Teknologtryck
Gothenburg, Sweden 2025

Anomaly Detection in Distributed Strain Sensing Data
A case study of machine learning applications of the Götaälv bridge
SIMON FLAGERUP
VIKTOR KJELLSSON
Department of Architecture and Civil Engineering
Chalmers University of Technology

Abstract

The Götaälv Bridge in Gothenburg, Sweden, was a vital part of the region's infrastructure network, and ensuring its operational safety was essential as it approached the end of its service life. To support this, the bridge was instrumented with an advanced distributed fiber optic sensing system to monitor strain and structural behavior during its final years of operation, which contributed to keeping the bridge safely in service until it was replaced and decommissioned in 2021.

This thesis explores unsupervised machine learning methods for anomaly detection in structural health monitoring, using distributed strain sensing data from the Götaälv Bridge. The primary focus is on evaluating the effectiveness of two distinct approaches: time-series anomaly detection using Long Short-Term Memory (LSTM) neural networks, and nonparametric clustering of strain profiles to detect new behaviors in the bridge using a Dirichlet Process Gaussian Mixture Model (DPGMM).

Following an exploratory data analysis (EDA) and a literature review, the LSTM model is trained to reconstruct normal strain patterns over time, enabling the identification of anomalies through deviations in reconstruction error. In parallel, the DPGMM approach applies dimensionality reduction through principal component analysis (PCA) and probabilistic clustering to group similar strain profiles, revealing shifts in the bridges operational behavior and potential damages. Both methods are validated using known events, such as construction activities near the bridge, which provoked displacements of the bridges supports, and through comparison with alarms for abnormally high values raised by the old system.

The results demonstrate one of the main limitations of the LSTM model: it requires high quality, healthy training data to reliably detect anomalies in new data. In this case, such data was not available in sufficient quantities because the system was installed on the bridge late in its service life, resulting in insufficient data for effective training. In contrast, by fully leveraging the distributed nature of the monitoring system, the DPGMM successfully identified changes in the bridge's behavior by detecting shifts in the strain profiles along the monitored beams without requiring any healthy baseline data, making it well suited for applications on existing structures.

Keywords: Structural Health Monitoring, Anomaly Detection, Distributed Fiber Optic Sensing, Long Short-Term Memory (LSTM), Dirichlet Process Gaussian Mixture Models (DPGMM), Götaälv Bridge, Machine Learning, Unsupervised Learning

Acknowledgements

We would like to thank everyone that supported us during our work on this master thesis.

We would like to express our sincere gratitude to our supervisor Farshid Zamiri at AFRY for his commitment and support throughout the project.

We are equally grateful for our supervisor and examiner at Chalmers University of Technology, Carlos Gil Berrocal, for his support, commitment, and fast replies.

We also want to thank everyone at AFRY for allowing us to use their facilities, providing valuable insight into the Götaälv bridge and giving us access to the data necessary to make this thesis possible. An extra big thanks to Ingvar Larsson for his expertise knowledge on the Götaälv bridge and for his selfless involvement in the project.

Finally, we would like to express our deepest gratitude to our families and friends for their unwavering support and encouragement throughout the course of this thesis and our entire time as students at Chalmers University of Technology.

Simon Flagerup & Viktor Kjellsson,
Gothenburg, June 2025

List of Acronyms

Below is the list of acronyms that have been used throughout this thesis listed in alphabetical order:

AI	Artificial Intelligence
AKSC	Adaptive Kernel Spectral Clustering
DPGMM	Dirichlet Process Gaussian Mixture Model
EDA	Exploratory Data Analysis
FE	Finite Element
GMM	Gaussian Mixture Model
IQR	Interquartile Range
LSTM	Long Short Term Memory
ML	Machine Learning
MSE	Mean Squared Error
OC-SVM	One-Class Support Vector Machine
PCA	Principal Component Analysis
RNN	Recurrent Neural Network
SBS	Stimulated Brillouin Scattering
SHM	Structural Health Monitoring

Contents

List of Acronyms **x**

List of Figures **xviii**

List of Tables **xix**

1 Introduction **1**
1.1 Background . . . 1
1.2 Scope . . . 3
1.3 Limitations . . . 5

2 Theory **7**
2.1 Structural Health Monitoring . . . 7
2.2 Distributed Fiber Optic Systems . . . 8
2.2.1 Brillouin Scattering . . . 8
2.2.1.1 Spontaneous Brillouin Scattering . . . 8
2.2.1.2 Stimulated Brillouin Scattering . . . 8
2.3 Anomaly Detection . . . 8
2.4 Machine Learning . . . 9
2.4.1 Data Handling . . . 9
2.4.1.1 Data Cleaning . . . 9
2.4.1.2 Data Scaling . . . 9
2.4.1.3 Feature Engineering . . . 10
2.4.2 Learning Paradigm . . . 10

3 Method **11**
3.1 Exploratory Data Analysis . . . 11
3.1.1 Availability of Data . . . 11
3.1.2 Data Structure . . . 11
3.1.3 Missing Data and Sudden Shifts . . . 12
3.1.4 Temporal Dependencies and Temperature Fluctuations . . . 13
3.1.5 Spatial Dependencies . . . 13
3.1.6 Strain Profiles . . . 14
3.2 Problem Approaches . . . 14
3.2.1 Problem Characteristics . . . 15
3.3 Potential Models . . . 15
3.3.1 Related Works . . . 16

3.3.1.1 Approach 1 - Potential Models . . . 16
3.3.1.2 Approach 2 - Potential Models . . . 17
3.3.2 Model Candidates - Approach 1 . . . 17
3.3.3 Model Candidates - Approach 2 . . . 19
3.3.4 Model Selection . . . 21
3.3.4.1 Model Selection Approach 1 . . . 22
3.3.4.2 Model Selection Approach 2 . . . 22
3.4 LSTM Model . . . 23
3.4.1 Data Preprocessing for LSTM Model . . . 23
3.4.1.1 Raw Data . . . 23
3.4.1.2 Handling Shift in the Time Series . . . 24
3.4.1.3 Data Splitting . . . 25
3.4.1.4 Outliers . . . 25
3.4.1.5 Missing Data . . . 25
3.4.1.6 Feature Engineering . . . 26
3.4.1.7 Data Normalization . . . 26
3.4.1.8 Sequencing the Data . . . 26
3.4.2 LSTM Model Architecture . . . 26
3.4.3 Training . . . 28
3.4.3.1 Input . . . 28
3.4.3.2 Loss Function . . . 29
3.4.3.3 Training Procedure . . . 29
3.4.4 Testing . . . 30
3.4.4.1 Anomaly Detection . . . 30
3.4.5 Validation . . . 30
3.5 Dirichlet Process Gaussian Mixture Model . . . 31
3.5.1 Data Preprocessing for DPGMM . . . 31
3.5.1.1 Data Management . . . 31
3.5.1.2 Outliers . . . 31
3.5.1.3 Missing Data . . . 32
3.5.1.4 Principal Component Analysis . . . 32
3.5.2 Model input . . . 32
3.5.3 Clustering Algorithm . . . 33
3.5.4 Validation . . . 35

4 Results **37**
4.1 LSTM Model Results . . . 37
4.1.1 Model Training . . . 37
4.1.2 Model Validation . . . 38
4.1.3 Test and Anomaly Detection . . . 39
4.2 DPGMM Results . . . 43
4.2.1 Feasibility Assessment of Clustering Strain Profiles . . . 43
4.2.1.1 Dimensionality Reduction with PCA . . . 43
4.2.1.2 Clustering Using Parametric GMM . . . 44
4.2.1.3 Plotting Cluster Assignment over Time . . . 44
4.2.1.4 Cluster Representations . . . 45

4.2.2 Results for Beam C using DPGMM 46
4.2.2.1 Plotting the Clusters 46
4.2.2.2 Cluster Assignment over Time 47
4.2.2.3 Cluster Behavior Representations 48
4.2.3 Results for Beam E Using DPGMM 51
4.2.3.1 Plotting the Clusters 51
4.2.3.2 Cluster Assignment over Time 52
4.2.3.3 Cluster Behavior Representations 52
4.2.4 Comparison of Results in Beam C and E 55
4.2.5 Validation of DPGMM Clustering Model 57
4.2.5.1 Cluster Assignment Over Time 57
4.2.5.2 Stability of Cluster Assignment 58
4.2.5.3 Conclusion on the Validation 59

5 Discussion **61**
5.1 Discussion on Approach 1 - Long Short Term Memory Neural Network 61
5.2 Discussion on Approach 2 - Dirichlet process Gaussian mixture model 63
5.3 Methodological Implications 65
5.4 Ethical, Social, and Privacy Considerations in AI-Based SHM 66

6 Conclusion **67**
6.1 Future Research 69

Bibliography **71**

Appendix **I**

A Appendix A. Results from LSTM model **I**
A.1 Beam B Model Reconstructed Signals and Anomalies II
A.2 Beam C Model Reconstructed Signals and Anomalies III
A.3 Beam D Model Reconstructed Signals and Anomalies IV
A.4 Beam E Model Reconstructed Signals and Anomalies V
A.5 Beam F Model Reconstructed Signals and Anomalies VI
A.6 Support I Model Reconstructed Signals and Anomalies VII
A.7 Support II Model Reconstructed Signals and Anomalies VIII
A.8 Support III Model Reconstructed Signals and Anomalies IX
A.9 Support IV Model Reconstructed Signals and Anomalies X
A.10 Support V Model Reconstructed Signals and Anomalies XI
A.11 All Channels Model Reconstructed Signals and Anomalies XII

B Appendix B. Results from DPGMM **XVII**
B.1 DPGMM Results Beam B XVII
B.2 DPGMM Results Beam D XXI
B.3 DPGMM Results Beam F XXIV

C Appendix C. Sankey Diagrams **XXIX**
C.1 Sankey Diagram Showing the Evolutions of Clusters XXIX

D Appendix D. Correlation Between Channels **XXXI**
D.1 Correlation Between Time Series XXXII

E Appendix E. GitHub Repository **XXXIV**

List of Figures

1.1 Overview of the bridge and its orientation with södra älvbrodelen highlighted. . . . 4
1.2 Cross section view of södra älvbrodelen, with the main longitudinal beams A–G. . . . 4
1.3 Side view of södra älvbrodelen, marked in red, with supports I–V. . . 5

3.1 Strain plot showing a sudden shift along the y-axis following an interruption. . . . 12
3.2 Graph of strain for one loop showing the temporal dependency due to temperature variations. In this figure, the shift seen in Fig. 3.1 for S-F_Close_Comp_II.txt,0.06 has been corrected using the method described later in Section 3.4.1.2. . . . 13
3.3 Strain profiles for beam E for four days in 2017 (top row) and 2018 (bottom row). The first frame on each row marks day 1, the second frame is the next day, followed by one week and one month ahead. . . 14
3.4 Flowchart of data preprocessing strategy. . . . 23
3.5 Showcase of how the data looks before and after shifting 25
3.6 Architecture of the LSTM model. . . . 27
3.7 Flowchart illustrating the steps involved in the model training procedure. . . . 28
3.8 Examples showing the input constellations. The sets of input consisted of for example all supports (I–V) for beam B (marked in blue) or all beams (B–F) at support I (marked in red). . . . 29
3.9 Flowchart of the testing procedure. . . . 30

4.1 T3a alarms over time for beams B-F. . . . 38
4.2 T3b alarms over time for beams B-F. . . . 38
4.3 T4 alarms over time for beams B-F. . . . 39
4.4 True vs. predicted signal for S-B_Close_Comp at support II. The predictions and anomalies come from the model that is trained on support II, according to the constellations in Tab. 3.2. . . . 39
4.5 Sign of reconstruction of abnormal data. . . . 40
4.6 True vs. predicted signal for S-B_Close_Comp at support I (top), II (middle) and IV (bottom). The predictions and anomalies come from the model that is trained on beam B, according to the constellations in 3.2. . . . 41

4.7 True vs. predicted signal for S-C_Close_Comp at support II. The predictions and anomalies come from the model that is trained on beam C, according to the constellations in Tab. 3.2. 41
4.8 True vs. predicted signal for S-D_Close_Comp at support II (top) and V (bottom). The predictions and anomalies come from the model that is trained on beam D, according to the constellations in Tab. 3.2. 42
4.9 True vs. predicted signal for S-F_Close_Comp at support V. The predictions and anomalies come from the model that is trained on beam F, according to the constellations in Tab. 3.2. 42
4.10 The explained variance plotted for a number of principal components. The dataset used for this particular example is from beam C where the original data had 1609 strain components. 43
4.11 Clustering using parametric GMM with the first two principal components plotted. 44
4.12 The cluster assignments over time by the GMM model for beam C. . 45
4.13 Mean strain profiles and average standard deviations for clusters identified by GMM on beam C. Each curve shows a cluster mean and average standard deviation (shaded) indicates how closely individual profiles match their cluster mean for each point x_i. 46
4.14 Mean strain profile and average standard deviation for cluster 3 for beam C. 46
4.15 Clusters prediction by the DPGMM plotted as combination of the first four principal components for beam C. 47
4.16 Cluster assignment over time on beam C using DPGMM clustering with a sliding window approach. 48
4.17 Mean strain distributions (1/2) from DPGMM clustering for beam C 49
4.17 (continued) Mean strain distributions (2/2) from DPGMM clustering for beam C 50
4.18 Comparison between the first and last assigned clusters for beam C. . 51
4.19 Clusters prediction by the DPGMM plotted as combination of the first four principal components for beam E. 51
4.20 Cluster assignment over time on beam E using DPGMM. 52
4.21 Mean strain profiles (1/2) from DPGMM clustering for beam E. . . . 53
4.21 (continued) Mean strain profiles (2/2) from DPGMM clustering for beam E. 54
4.22 Comparison between the first and the last assigned cluster in beam E using DPGMM. 55
4.23 Overview and comparison of cluster formation for beam C (top) and E (bottom). Annotations mark the formation (or return) of the cluster in question. The area marked in green represents the period when vertical and horizontal displacements were detected at support I and 27. 56
4.24 The cluster assignments over time by the DPGMM model for beam C. 57
4.25 The cluster assignments over time by the GMM model for beam C. . 57
4.26 Comparison between cluster 0 and 3 for the GMM (top) and the DPGMM (below). 58

5.1 A simplified flowchart showing a proposed monitoring workflow using the DPGMM model in combination with domain knowledge and integrated decision making. 65

A.1 True vs. predicted signal for the test data for S-B_Close_Comp at support I-V. II
A.2 True vs. predicted signal for the test data for S-C_Close_Comp at support I-V. III
A.3 True vs. predicted signal for the test data for S-D_Close_Comp at support I-V. IV
A.4 True vs. predicted signal for the test data for S-E_Close_Comp at support I-V. V
A.5 True vs. predicted signal for the test data for S-F_Close_Comp at support I-V. VI
A.6 True vs. predicted signal for the test data for beam B-F at support I. VII
A.7 True vs. predicted signal for the test data for beam B-F at support II. VIII
A.8 True vs. predicted signal for the test data for beam B-F at support III. IX
A.9 True vs. predicted signal for the test data for beam B-F at support IV. X
A.10 True vs. predicted signal for the test data for beam B-F at support V. XI
A.11 Reconstructed signals and anomalies for S-B Channels I–IV XII
A.12 Reconstructed signals and anomalies for S-C Channels I–IV XIII
A.13 Reconstructed signals and anomalies for S-D Channels I–IV XIV
A.14 Reconstructed signals and anomalies for S-E Channels I–IV XV
A.15 Reconstructed signals and anomalies for S-F Channels I–IV XVI

B.1 Clusters prediction by the DPGMM plotted as combination of the first four principal components. XVII
B.2 Cluster assignment over time on beam B. XVII
B.3 Mean strain profiles (1/3) from DPGMM clustering for beam B. . . . XVIII
B.3 Mean strain profiles (2/3) from DPGMM clustering for beam B. . . . XIX
B.3 Mean strain profiles (3/3) from DPGMM clustering for beam B. . . . XX
B.4 Clusters prediction by the DPGMM plotted as combination of the first four principal components. XXI
B.5 Cluster assignment over time on beam F. XXI
B.6 Mean strain profiles (1/3) from DPGMM clustering for beam D. . . . XXII
B.6 Mean strain profiles (2/3) from DPGMM clustering for beam D. . . . XXIII
B.6 Mean strain profiles (3/3) from DPGMM clustering for beam D. . . . XXIV
B.7 Clusters prediction by the DPGMM plotted as combination of the first four principal components. XXIV
B.8 Cluster assignment over time on beam F. XXIV
B.9 Mean strain profiles (1/3) from DPGMM clustering for beam F. . . . XXV
B.9 Mean strain profiles (2/3) from DPGMM clustering for beam F. . . . XXVI
B.9 Mean strain profiles (3/3) from DPGMM clustering for beam F. . . . XXVII

C.1 Sankey diagram showing the drift of data points between time steps of twelve week interval for beam C. Each node represents the number of data points within a specific cluster at a certain moment in time. The links represent the flow of data points from one node to another between t_i and t_{i+1}. XXIX
C.2 Sankey diagram showing the drift of data points between time steps of twelve week interval for beam E (bottom). Each node represents the number of data points within a specific cluster at a certain moment in time. The links represent the flow of data points from one node to another between t_i and t_{i+1}. XXX

D.1 Correlation between the timeseries on Älvbrodelen. XXXII

List of Tables

3.1 Outline of the two machine learning approaches. 15
3.2 All points that were chosen to be monitored and modeled. 24

4.1 Training loss for the models after 40 epochs of training. 38

1

Introduction

1.1 Background

The Götaälv bridge was a bascule bridge that connected mainland Gothenburg with the island of Hisingen (shown in Fig. 1.1 and the cover page). It was used for road traffic, trams, bicycles and pedestrians from its inauguration in 1939 until its decommission in 2021 when the newly constructed Hisingsbron was opened up for traffic. The Götaälv bridge was a key part of the road network in the city and the region as it was one of the main road links over the river.

From 2016, three large infrastructure projects and two real estate projects where on-going in close proximity of the Götaälv bridge (Adrian et al., 2021). There were concerns that the nearby piling activities would provoke ground displacements which could negatively affect the old bridge that had low tolerance against displacements to maintain its function. Elements such as the bascule (bride leaf) and its opening mechanisms where critical. Another critical aspect was the steel that was used for the bridge's structure. The steel was of an old type called Thomas steel, which is known to be brittle (Stroetmann & Sieber, 2018). For this reason, the bridge's superstructure could only withstand small displacements in its supports (Adrian et al., 2021).

To secure the function and safety of the bridge, a cooperative task force was set up and consisted of key stakeholders such as city authorities, technical experts as well as representatives from the ongoing projects in the area (Adrian et al., 2021). The group coordinated ground work activities of the nearby construction sites to limit the negative effect they had on the function the Götaälv bridge within acceptable limits. Both vertical and horizontal displacements of the supports were closely monitored and the nearby ground works were coordinated to ensure that their activities did not pose a risk to the integrity of the bridge. The practice of identifying damage to a structure over time is known as structural health monitoring (SHM). SHM measures the performance of a structure and its decreasing function and gives the information back to the user (Farrar & Worden, 2007).

In addition, the bridge had been monitored since 2007 for cracks and high strain values using a distributed optical fiber sensing monitoring system (Glišić et al., 2007). Optical fibers were mounted on five of the bridge's main girders, spanning its full length, totaling to five kilometers of fiber optic sensing cables with a spatial resolution of 100 mm. The bridge was divided into sections and in each section the

fiber optic cable arrangement was further divided into multiple closed loops that consisted of fiber optic cables that monitored specific beams. The beams monitored where beams B–F as shown in Fig. 1.2.

The system performed measurements every four hours, except during the first two years when the frequency was once every two hours (Glišić et al., 2007). During each measurement, the fiber optic cables collected data on both strain and temperature across the bridge. By recording the temperature, the system was able to distinguish between the strain caused by imposed loads and the strain resulting from thermal expansion or contraction. The system was designed to detect anomalies of various degrees, ranging from high strain values to ruptures and it included a warning mechanism that alerted operators whenever predefined thresholds were exceeded. In order to confirm whether the steel beam had suffered a failure or not a physical inspection had to be done following the most servere alarms.

During the period from early 2017 to early or mid-2018, significant vertical and horizontal displacements were recorded at several supports on the south side of the bridge (Adrian et al., 2021). These displacements coincided with intensive piling activities on nearby construction sites, which continued until March 2018. When the piling ceased, the displacements stabilized. Once the displacements had stopped, the settlements were around 15 mm in support I and 30 mm in support 27, while the horizontal displacements were around 45 mm in both support I and 27. The supports are marked in Fig. 1.3.

With a spatial resolution of 100 mm and five kilometers of fiber-optic cables, approximately 50,000 points were being monitored, resulting in large volumes of data to process and analyze. The system's anomaly detection relied on absolute strain thresholds, meaning anomalies were only recognized if strain values exceeded a predefined limit, rather than if they deviated from expected patterns at a given time, which is a clear limitation. This approach also led to a high number of false positives.

By leveraging machine learning, with the ability to efficiently process large datasets and identify complex patterns that may be difficult for humans to detect, there is significant potential to improve the effectiveness and accuracy of structural health monitoring for bridges (Cha et al., 2024).

In this thesis, machine learning is investigated as a complement, with the aim of exploring its potential to handle large datasets form distributed fiber optic systems. The goal is to assess whether machine learning can improve the effectiveness and accuracy of structural health monitoring for bridges monitored using systems similar to the one used on the Götaälv bridge.

1.2 Scope

The scope of this thesis is to investigate the structural condition and behavioral evolution of the Götaälv bridge during the period it was monitored with distributed fiber optic sensing technology, and explore how machine learning could be used to analyze behavior and detect anomalies. The study is based on the extensive dataset collected from 2007 until the bridge's decommissioning, encompassing periods of significant nearby construction activity and observed displacements of the bridge's supports.

Any machine learning model developed in this thesis will serve two main purposes. Firstly, it will be validated against real-world events observed during the monitoring period, with particular focus on the ground movements discussed in Section 1.1 and on alarms triggered by the existing warning system. The aim is to verify that the model can reliably detect these known events while minimizing false positives. However, the validation is challenging because the dataset is unlabeled, which makes it difficult to accurately determine false positive and false negative rates.

Secondly, the model will be used to explore new potential uses of the distributed fiber optic strain data, beyond the original design intentions of the monitoring system. This may include identifying patterns or anomalies that have not previously been recognized or more effectively leveraging the potential of the distributed properties of the fiber optic system.

This thesis focuses on the southern section of the bridge that is closest to the river, also known as "södra älvbrodelen", marked in red in Fig. 1.1. Due to its proximity to other large construction sites in the area and its extensive documentation, södra älvbrodelen represents a particularly valuable subject for this research. While södra älvbrodelen consists of beams A–G spanning over supports I–V, only beams B–F were monitored. A cross section view of södra älvbrodelen showing the longitudinal beams can be seen in Fig. 1.2 and a side view showing the supports can be seen in Fig. 1.3.

Lastly, this thesis aims to provide practical insights for similar future monitoring projects, by evaluating machine learning methods on fiber optic data from the Götaälv bridge. These insights could help guide the design and implementation of more reliable and efficient structural health monitoring systems on similar pieces of infrastructure.

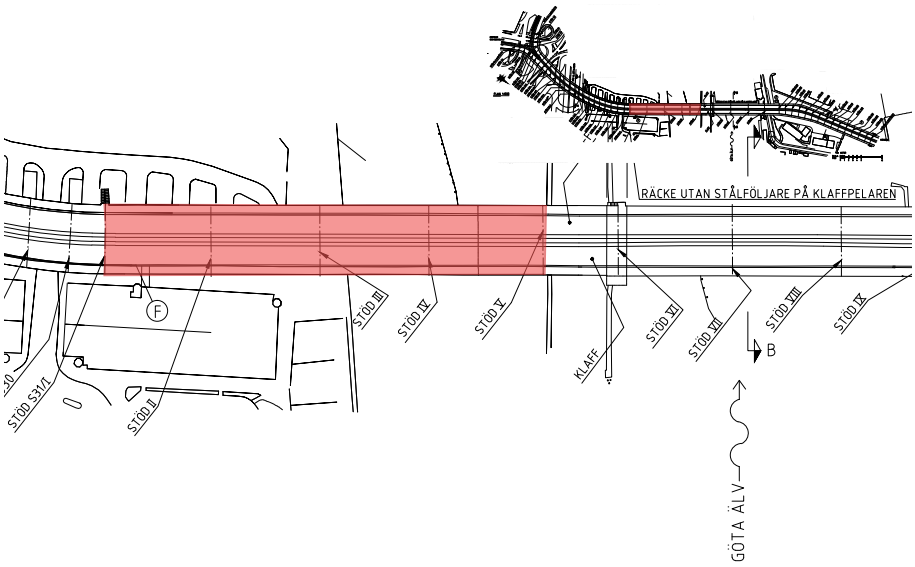

Figure 1.1: Overview of the bridge and its orientation with södra älvbrodelen highlighted.

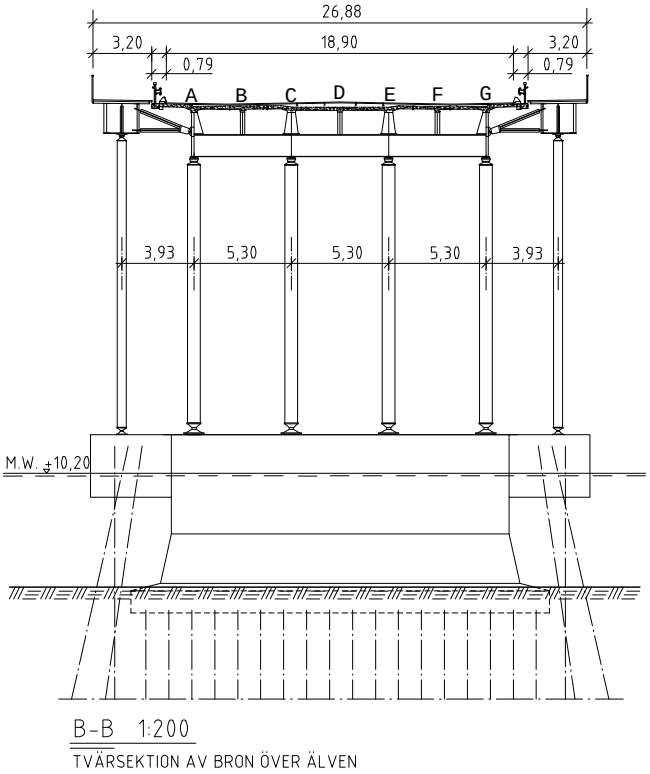

Figure 1.2: Cross section view of södra älvbrodelen, with the main longitudinal beams A–G.

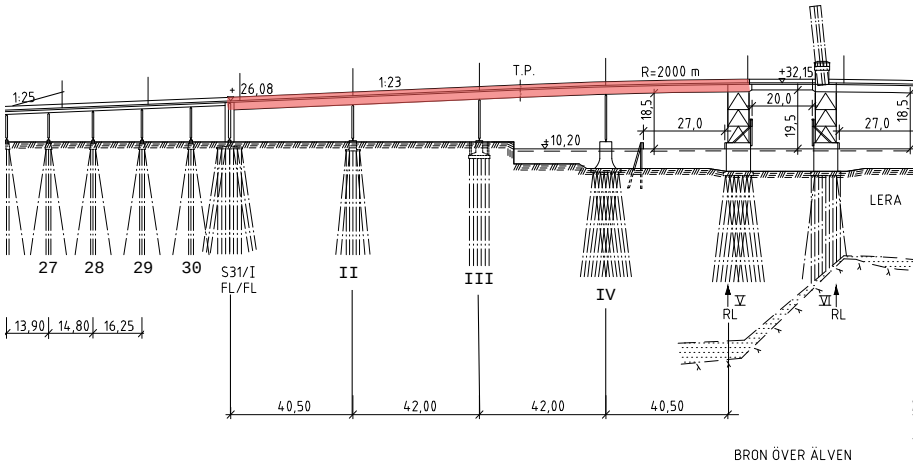

Figure 1.3: Side view of södra älvbrodelen, marked in red, with supports I–V.

1.3 Limitations

The work comes with several limitations. Many of them stem from unavoidable limitations in the data and its structure, while some are active choices to limit the work. The limitations are outlined below.

Absence of labels: The raw dataset comes without any labels about the condition of the bridge. This constrains the model to operate within unsupervised or semi-supervised learning regimes. As mentioned in the scope section, it makes the model validation a challenge as it will need to resort to written documentations about events that happened during the monitoring. This assumes the existence of events that are both well documented and detectable.

Lack of healthy data: Due to the fact that the monitoring system was installed on the bridge after nearly seven decades of operation, it is not guaranteed that a sufficient amount of healthy data to act as baseline for the model training is available. This could be a major hurdle when training models to distinguish abnormal from healthy data.

No information about the load: The data that is available are for strains and temperatures. What is not known, is how the bridge was loaded at each time stamp. This makes it impossible to contextualize the strains, since they make more sense in relation together with the load that produced them.

No dynamic analysis: No dynamic analysis of the bridge will be conducted due to the low temporal resolution of the data. The temporal resolution corresponds to one measurement every four hours, which is too sparse for a dynamic analysis.

Excluding the first years: Prior to June 5, 2009, data was recorded every two hours, whereas after this date, the sampling rate was changed to every four hours. For this reason, all data prior to that will be excluded.

Model selection: There might be a vast array of models that could be employed for the task. However, the goal of this work is not to find the ultimate model but

rather explore the possibilities of the data and the distributed fiber optic technique using ML. For this reason, not all possible models will be considered in this work.

2

Theory

Structural health monitoring systems have been applied to buildings and bridges at an increasing rate over the last decades. In order to handle the huge amount of data from the surveillance the implementation of machine learning has also gained popularity. This chapter aims at providing a brief theoretical background into the monitoring technique used on the Götaälv bridge and machine learning for anomaly detection applications in SHM.

2.1 Structural Health Monitoring

Structural health monitoring refers to the process of implementing a damage identification strategy for civil, aerospace, or mechanical engineering infrastructure (Farrar & Worden, 2013). By observing the response of a structure over time, extracting meaningful features and performing statistical analysis on these features, the current state of a system's health can be estimated. Here, damage will be defined as intentional or unintentional changes to the material and/or geometric properties of the system, including changes to the boundary conditions and system connectivity, which adversely affect the current or future performance of the the system.

In SHM, two main approaches can be distinguished, model-based and data-based (Xu et al., 2023). The model-based methods require a precise finite element (FE) model of the monitored structure. Data collected from the SHM system is used to continuously update the model. Defects can then be identified by comparing the measured response of the bridge to the expected response from the FE model. Implementation is usually limited to experimental structures and small bridges due to the inherent complexity of long-span bridges, which results in time-consuming and impractical analysis of the FE model. This has led to an increasing interest in data-based methods where a digital replica is not needed as the analysis relies purely on data collected by the SHM system.

When monitoring large infrastructures, SHM systems generate massive amounts of data in multiple channels, making it difficult for traditional statistical methods to generalize (Farrar & Worden, 2013). This is the reason why ML has become increasingly deployed in SHM applications. However, it still remains relatively recent for civil engineering structures.

2.2 Distributed Fiber Optic Systems

Distributed fiber optic sensors can measure strain and temperature over a continuous strain field rather than at discrete points (Glišić & Inaudi, 2012). By enabling high spatial resolution measurements, this approach provides a cost-effective method for monitoring structural performance (Delo et al., 2025). What enables the high spacial resolution is a technique called Brillouin scattering, which will be explored in more detail below.

2.2.1 Brillouin Scattering

Brillouin scattering is a nonlinear optical process in which an optical light wave, called the pump, propagates through a fiber and interacts with acoustic waves (phonons) (Lopez-Higuera et al., 2011). The scattered light counter propagates relative to the original wave due to reflection and scattering from the phonons. The presence and characteristics of these acoustic waves depend on the fiber's temperature and strain (Glišić et al., 2007). The light is continuously scattered providing a continuous high spatial resolution in the measured data.

2.2.1.1 Spontaneous Brillouin Scattering

The scattered light from phonons is known as spontaneous brillouin scattering (Inaudi et al., 2001). Since the light is reflected off the waves at a different frequency, the scattered light undergoes a frequency shift that is linearly dependent on the fiber's strain and temperature (Glišić et al., 2007). If the frequency shift is known for a given strain and temperature, the absolute strain and temperature can be determined (Inaudi & del Grosso, 2008).

By applying the radar principle and utilizing the known speed of light, the exact location of the scattering event can be identified. This enables absolute strain and temperature measurements at any point along the fiber (Glišić et al., 2007).

2.2.1.2 Stimulated Brillouin Scattering

In stimulated brillouin scattering (SBS), a second wave, known as the Stokes wave, is introduced into the fiber. This probe wave is generated through energy transfer from the pump (Lopez-Higuera et al., 2011). The Stokes wave matches the frequency of the acoustic waves, increasing the phonon population and enhancing the acoustic wave. As a result, the amount of scattered light increases (Inaudi et al., 2001). This amplification enables accurate measurements over longer distances, making SBS suitable for monitoring large-scale structures. In contrast, Spontaneous Brillouin Scattering, due to its weaker signal, is limited to shorter sensing ranges.

2.3 Anomaly Detection

The article *Dive into Time-Series Anomaly Detection A Decade Review*, by Boniol et al. (2024) presents different definitions of types of anomaly detection.

The traditional definition is that anomalies are observations that considerably differ from the majority of other samples in a distribution.

- **Point anomalies** A single data point that differs from the expected behavior.
- **Contextual anomalies** Anomalies that have anomalous properties in a specific context.
- **Collective anomalies** Group of data points that when studied together show abnormal behavior.

The current anomaly detection implemented on the bridge was a sufficient way to detect point anomalies that had unexpected behavior, indicating a crack or failure of the beam (Glišić et al., 2007), but it missed the two other types of anomalies.

2.4 Machine Learning

This section introduces some of the key concepts in machine learning, tailored to the case study at hand.

2.4.1 Data Handling

SHM systems collect vast amounts of data. The raw data collected from the sensors are not necessarily ready to be analyzed straight away (Xu et al., 2023). Measurement malfunctions due to system errors, such as loosely mounted sensors, or external effects, such as weather or progressive wear of components, can lead to unreliable or missing data (Malekloo et al., 2022). The data might even be in different dimensions, making it impossible to compare. In order to rectify these problems the data must be manipulated and reshaped before it can be used as input for a ML model.

2.4.1.1 Data Cleaning

The first step of the data manipulation is data cleaning, removing common data errors such as missing data points and outliers (Xu et al., 2023).

- **Missing Data** need to be handled, as many algorithms cannot process incomplete data and may produce biased or inaccurate results.
- **Outliers** are data points that significantly differ from other data points, usually exceeding the reasonable range. Outliers that are in the extreme should be removed but some outliers inside certain ranges could indicate abnormal events instead of measurement errors and should be preserved.

2.4.1.2 Data Scaling

Data scaling involves rescaling and standardizing the input data to ensure consistency in dimensions and to reduce the impact of noise. In machine learning, noise refers to random or irrelevant variations that obscure the underlying patterns in the data, often resulting from measurement errors, missing values, or outliers. Since different sensors may operate on varying scales, their output can have disparate

ranges. Without scaling, features with larger ranges may disproportionately influence the machine learning algorithm. Rescaling the data ensures that only the most significant and damage-sensitive features are emphasized (Xu et al., 2023). However, it is crucial that this process does not compromise the model's ability to learn meaningful patterns (Malekloo et al., 2022).

There are different methods to normalize the data and this report has chosen to focus on three different ways (Xu et al., 2023).

- **Min-max normalization** is when the data is rescaled to dimensions between [0, 1]
- **MaxAbsScaler** takes every data feature and rescales it separately so that each maximal absolute value is 1.
- **Standard Scaler** rescales the data to a mean value of 0 and a standard deviation of 1.

2.4.1.3 Feature Engineering

Feature engineering is the process of selecting and extracting features from the raw data. The features are selected based on mathematical procedures or simply by engineering judgment or by a combination of both (Malekloo et al., 2022). It is the last step to make sure that the quality of the raw data is high enough and that it can be used as input for a ML algorithm (Xu et al., 2023).

2.4.2 Learning Paradigm

A learning paradigm in ML refers to the general approach or framework used to train models based on data. The main learning paradigms applicable in SHM are (Xu et al., 2023):

- **Supervised learning:** The training process is guided by labeled data. A supervised model is capable of two kinds of predictions: classification and regression. In the case of SHM, a supervised classification task could be identifying anomalous vibration samples by using pre-labeled training samples. Regression could be the prediction of the structural response.
- **Unsupervised learning:** An unsupervised model lacks the labeled data for training and validation. Therefore, unsupervised models rely on identifying patterns and commonalities among the data, which allow it to e.g. isolate anomalies from normal data in SHM applications.
- **Semi-supervised learning:** These models train on a mix of labeled and unlabeled data, which is useful when labeled data is scarce but there is an abundance of unlabeled data.

3

Method

In this chapter, the methodology for this thesis is outlined. It began with an exploratory data analysis (EDA) to assess the structure and availability of the data. Based on the EDA, two approaches to the problem were identified and defined. The next step involved reviewing and selecting suitable models. Following the initial presentation of candidate models, the most appropriate model for each approach was chosen. These models were then developed and tailored to the specific problem, with the preprocessing and chosen parameters detailed in their respective sections. A full list of the Python libraries used and a link to the GitHub repository containing the code for the project can be found in Appendix E.

3.1 Exploratory Data Analysis

To choose an adequate model and a well-suited data preprocessing method, a thorough EDA was performed on the dataset.

3.1.1 Availability of Data

The data was collected between August 2007 and June 2021, when the bridge was decommissioned. Fiber optic cables installed on the underside of the top flange on five of the main beams registered the strain along the beam, and a temperature sensor cable mounted to the middle of the girder measured the temperature along the beam (Glišić et al., 2007). As mentioned in Section 1.1, the measurements were made with a sampling rate of once every four hours after June 5, 2009. Before that, the sampling rate was once every two hours. For this reason, only data after June 5, 2009 was used.

3.1.2 Data Structure

At the section södra älvbrodelen, data was collected using five fiber optic sensing loops that monitored one beam each. In the dataset, during the investigated time frame, loops that monitor the section södra älvbrodelen are referred to as "Close". For example, the loop that monitor beam B at södra älvbrodelen on the south bank is named S-B_Close_Comp.

The raw data was originally organized into folders labeled with the timestamp of the corresponding measurement. Inside each folder, there were text files, one for each fiber optic loop, containing the strain and temperature measurements recorded at that time. Each file provided spatially distributed measurements along a specific loop.

This structure allowed for two main types of data analysis:

- Time series analysis at fixed points along a beam, by tracking measurements over time.
- Spatial distribution analysis of strain along a beam at a specific timestamps.

Data was extracted for relevant sections of the bridge, as described in Section 1.2.

3.1.3 Missing Data and Sudden Shifts

The strain dataset contained significant amounts of missing data, which were categorized into four distinct types:

1. A few random missing values in one channel independently of the other channels.
2. A few missing values in multiple channels at the same time.
3. Periods of consecutive missing values in one channel independently of the other channels.
4. Periods of consecutive missing values in multiple channels independently at the same time.

Longer periods of missing data is likely due to maintenance work, while shorter periods of missing values likely come as a result of a system malfunctioning of some sort.

It can be observed on several occasions and across different channels that following an interruption, i.e. when there have been missing data, the strain is shifted along the y-axis. Such a shift can be seen in Fig. 3.1. The hypothesis is that this is due to a recalibration of the system following an interruption.

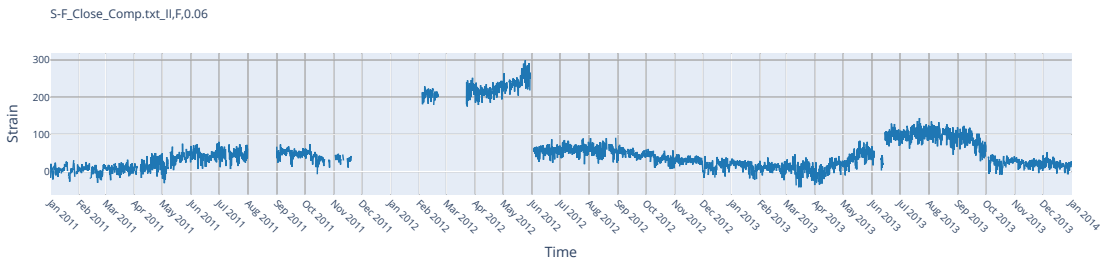

Figure 3.1: Strain plot showing a sudden shift along the y-axis following an interruption.

Given that hypothesis, when working with the data in the form of timeseries, it is necessary to realign the data before and after an interruption to avoid a bimodal distribution, which does not reflect the reality. If there is missing data for any point along a loop at a certain timestamp, like in Fig. 3.1, it implies that data is missing for the entire loop.

3.1.4 Temporal Dependencies and Temperature Fluctuations

The timeseries show a strong temporal dependency; see Fig. 3.2. This dependency is linked to the fluctuations in temperature measured on the beam. There is also a slight but not significant phase shift between the temperature fluctuations and the resulting effect it has on the strain.

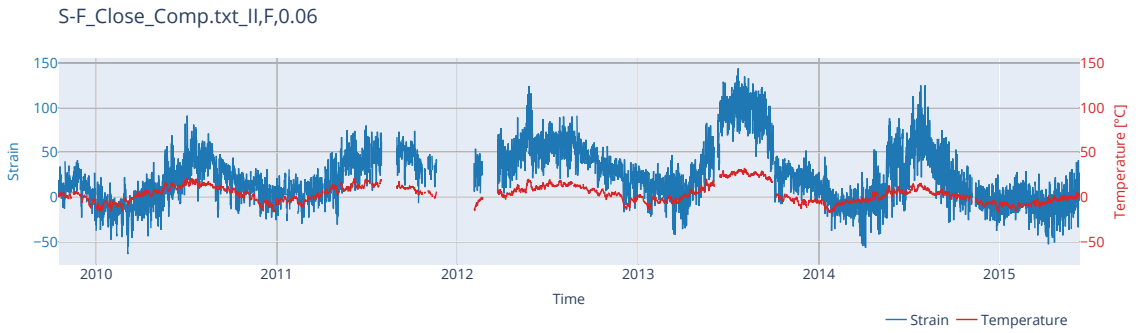

Figure 3.2: Graph of strain for one loop showing the temporal dependency due to temperature variations. In this figure, the shift seen in Fig. 3.1 for S-F_Close_Comp_II.txt,0.06 has been corrected using the method described later in Section 3.4.1.2.

3.1.5 Spatial Dependencies

The correlation heatmap, shown in Fig. D.1 in Appendix D, shows the spatial dependencies between the loops. For example, the correlation value between support IV and V in S-F_Close_Comp is 0.68 indicating a medium strong correlation, while the correlation between support I and V in S-F_Close_Comp is 0.03. This indicates that the correlation between supports that are closer placed together have a higher corelation than with other supports. The corelation between support IV in S-B_Close_Comp and S-F_Close_Comp is significantly lower. This is consistent with the physical layout of the beams: S-B_Close_Comp and S-F_Close_Comp are located on separate ends of the bridge, so they should have a lower correlation.

These patterns could provide valuable insights and context to the model, offering information about the global behavior of the structure. By considering the relationships between neighboring beams, the model could use the highly correlated behavior between parallel beams as a benchmark for detecting anomalies. This would improve anomaly detection by helping the model identify deviations from normal behavior when neighboring beams exhibit unexpected changes.

3.1.6 Strain Profiles

When plotting the strain for a beam at a given timestamp, the strain profile along the beam is obtained. From these plots, it can be observed that within a short time frame, the behavior is quasi constant, i.e. that the strain profile does not change significantly from one day to another. When comparing strain profiles from 2017 and 2018, see Fig. 3.3, it becomes clear that the shapes are close to identical for short time frame, but change when comparing between two years. This could indicate that, as long as the strain profile remains constant, changes in the bridge's behavior are primarily caused by external factors or damage, which could potentially be used to identify changes in behavior.

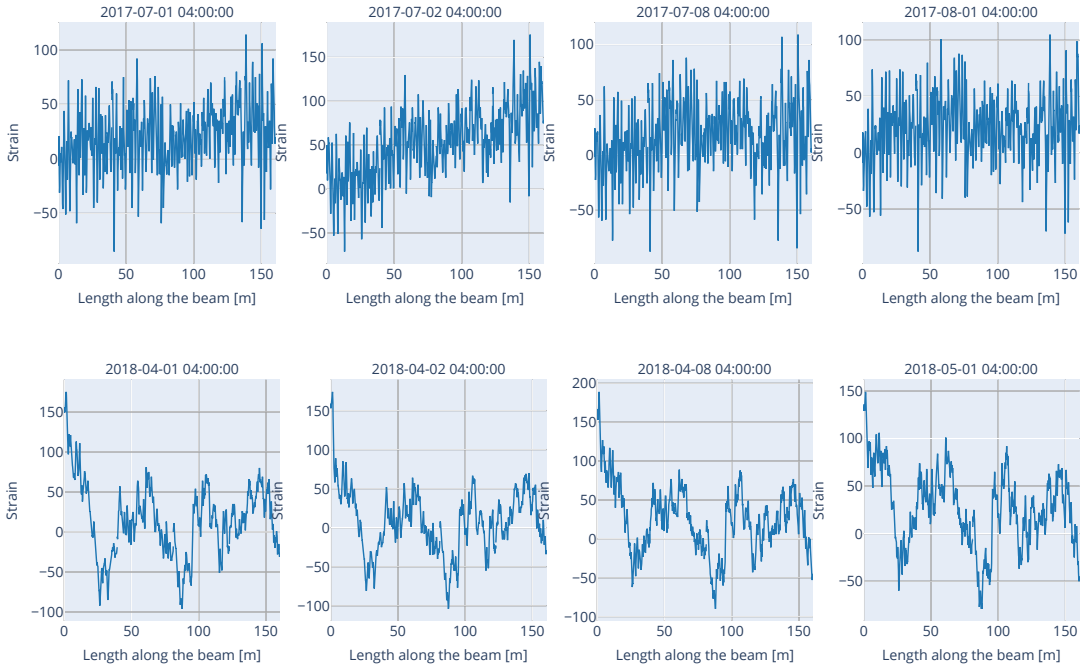

Figure 3.3: Strain profiles for beam E for four days in 2017 (top row) and 2018 (bottom row). The first frame on each row marks day 1, the second frame is the next day, followed by one week and one month ahead.

3.2 Problem Approaches

Based on the findings in the EDA, two approaches to analyze the data were identified.

- **Approach 1** - Detection of anomalies in time series: The model is trained on time series and is used to find anomalies based on reconstruction error between the reconstructed signal and test set.
- **Approach 2** - New behaviors in the bridge are identified by detecting changes in the strain profiles. This approach relies on two key assumptions: (1) the structural behavior of each member can be characterized by its strain profile,

and (2) strain profiles remain relatively constant over time unless a change in the bridge's behavior occurs. A significant change in the strain profile suggests that the member has entered a new regime of behavior, which may indicate an anomaly.

3.2.1 Problem Characteristics

The model selection is based on the characteristics of the problem, i.e. the SHM anomaly detection. The main characteristics for the two approaches are presented in Tab. 3.1:

Table 3.1: Outline of the two machine learning approaches.

	Approach 1	**Approach 2**
Goal	Detect and locate anomalies in time series	Detect anomalous strain profiles and locate anomalies
Learning paradigm	Unsupervised	Unsupervised
Data type	Strain	Strain
Data format	Time series, $\epsilon = f(t)$	Strain profiles, $\epsilon = f(x, t)$
Prediction type	Regression (or classification)	Classification (or regression)
Missing data	Exists missing data (see Section 3.1.3). Will require imputation.	Exists missing data (see Section 3.1.3). Imputation not necessary and impractical since strain profiles are considered*.
Noise & outliers	Noise and outliers exist and need to be treated.	Noise and outliers exist and need to be treated since strain profiles are considered.

*Each strain profile represents the strain measurements for all points along a loop for one measurement cycle. When there are missing values for a loop, all measurements along the loop are missing, making it difficult to impute as that would require the entire loop to be reconstructed.

3.3 Potential Models

A key part of this project is the selection of machine learning models and data processing pipelines tailored to the specific anomaly detection approach being considered. Since each approach, whether focused on time series anomaly detection or on identifying abnormal strain profiles, has distinct data characteristics and objectives, different models and pipelines might be optimal in each case.

The model selection was done by finding the most optimal machine learning models through a literature study: for each approach, suitable machine learning algorithms were identified based on their compatibility with the data format aims of the anomaly detection. The candidate models were implemented and evaluated on the dataset using suitable validation methods and performance metrics. The goal was to identify models that balance accuracy and computational efficiency for each approach, ensuring robust and reliable anomaly detection.

As previously mentioned, the data collected by the fiber optic system on the Götaälv bridge contains no labels. Thus, the anomaly detection task of the project relied purely on finding patterns in the data as labeling would have required deep domain knowledge, and is out of the scope of this work. It would have also been a highly manual task to label the data by hand and not necessarily reliable. The anomaly detection is thereby an unsupervised task, which largely impacted the model selection.

3.3.1 Related Works

In the field of machine learning for SHM applications, a vast amount of literature exists. This section will focus on studies that directly relate to the specific characteristics of the problem at hand described in Tab. 3.1, providing a more targeted review.

There are very few studies that apply machine learning to data from distributed fiber optic sensing systems, particularly when the temporal resolution is low, as is the case with the data used in this thesis. In fact, a review of the literature revealed no previous examples of machine learning applications using data with the same type and structure as was analyzed in this thesis.

Many machine learning applications for structural health monitoring for civil structures face a common problem that is the limited data of real structure faults (Diez et al., 2016). Therefore, unsupervised anomaly detection methods must be used in many cases.

3.3.1.1 Approach 1 - Potential Models

A study by Zhang et al. (2022) proposes the use of a Long Short-Term Memory (LSTM) model to find anomalies in SHM time series. LSTM network is a type of Recurrent Neural Network (RNN) (Zhang et al., 2022) designed to handle long-term dependencies in sequential data. Unlike standard RNNs, LSTMs incorporate memory blocks that allow the model to retain information over extended time periods and thus be able to notice any long term trends in the data

Saha et al. (2024) suggests using a modified version of Long Short Term Memory (LSTM) to find anomalies in time series data. Their method called quantile-Long Short term Memory (q-LSTM) predicts quantiles of the data instead of single data points, like regular LSTM. In this way the method is more flexible to other data types and allows for unsupervised training. If the training data consists of long-term

monitoring data, it might be able to detect long-term effects and defects on SHM systems. This method requires adjusting the different quantile thresholds to find the most optimal anomaly detection range.

Support Vector Machines (SVM) is an anomaly detection method that normally requires supervised training to locate anomalies but Jin et al. (2019) suggests using a One-Class SVM (OC-SVM) to detect the change point, i.e. where the condition transitions from a healthy to a faulty state. OC-SVM is trained on healthy, fault-free data and learns how to recreate it. When the data changes over time, the model flags it as a change point. This way it does not require any labeled anomalies, but the training data must be mostly fault-free.

3.3.1.2 Approach 2 - Potential Models

Diez et al. (2016) introduced a clustering-based approach to group joints with similar behavior on a bridge and detect abnormal or damaged joints separating normal and abnormal events through clustering. The methodology is demonstrated through two case studies involving vibration data monitored from the Sydney Harbour Bridge.

To solve the problem of limited availability of data covering all healthy and damaged states, Rogers et al. (2019) used a Bayesian non-parametric clustering technique, i.e Dirichlet process Gaussian mixture model (DPGMM), for online learning allowing adaptive classification of operational and damage states without extensive pre-collected training data. With the addition of domain knowledge, new regimes of data can be labeled, allowing for a semi-supervised learning.

In another study, Langone et al. (2017) proposed a different technique to address the sensitivity to environmental factors. Their approach unifies data normalization and damage detection by employing an algorithm called Adaptive Kernel Spectral Clustering (AKSC). The algorithm is calibrated on the undamaged structure, and, once calibrated, is capable of detecting new regimes of data.

A study by Aiping Guo et al. (2018) introduces a hybrid clustering approach for structural health monitoring of the ShenNong Creek Bridge. Using data standardization through principal component analysis (PCA) and whitening, it combines ISODATA and hierarchical clustering to improve damage detection. The results show enhanced accuracy compared to ISODATA alone.

3.3.2 Model Candidates - Approach 1

In the following section, two potential models for the task of analyzing time series data are presented. Each model in this section follows the blueprint of approach 1 described in Section 3.2.1 and is described in detail using the following points, which provide a comprehensive overview of its key aspects:

1. **Long Short Term memory (LSTM) model**

   Reference work

   - Long-Short Term Memory Network-Based Monitoring Data Anomaly Detection of a Long-Span Suspension Bridge (Zhang et al., 2022)

- Dive into Time-Series Anomaly Detection: A Decade review (Boniol et al., 2024)
- quantile-Long Short Term Memory: A Robust, Time Series Anomaly Detection Model (Saha et al., 2024)

General idea
LSTM is a type of RNN that contains memory cells, allowing it to keep information over time and detect long-term dependencies in the time-series data. LSTMs are used in an unsupervised learning setup, where the model is trained on healthy data and learns the typical patterns and behavior of said data.

A LSTM model usually consists of an input layer, one or more LSTM layers, a dropout layer and one output layer. The dropout layer randomly chooses some data to be forgotten, according to a probability, to prevent overfitting. Each LSTM layer , mconsists of a number of cells (units) that have their own function. In a LSTM model the memory cells consists of a forget gate, input gate and an output gate. The forget gate decides what to remove from the memory gate while the input gate decides what new information to store in it, and the output gate decides what data should be sent to the next layer.

The model architecture is specified by the parameters input dimension, number of layers and their hidden dimension, i.e. number of LSTM units per layer. The training is further controlled by the number of epochs and the learning rate.

Anomalies are detected by training the model on seen data and is then used to predict future values. The predicted values are compared to actual observations in the test dataset. Anomalies are detected by excessive deviations between the predicted value and the actual ground truth in the test data.

Advantages & Limitations
LSTM models offer several advantages for anomaly detection in SHM. They are capable of capturing complex dependencies and patterns, thanks to its neural network structure, which is useful when analyzing time-series data where anomalies could be a result of a gradually changing behavior. LSTM models have been proven to work well with large volumes of monitoring data and by using the models unsupervised learning framework the need for labels on the data is not necessary.

However, the performance of the analysis depends on the availability of healthy training data. If the training data is unhealthy it raises concerns over the validity of the training and the model could fail to learn healthy behavior and therefore learn to reconstruct anomalies, thus potentially masking real anomalies. Additionally, the threshold values used to identify anomalies in the reconstructed or predicted data must be manually set. This introduces subjectivity into the process, as the choice of threshold can significantly influence the outcome of the anomaly detection.

2. **One-Class Support Vector Machine (OC-SVM)**

Reference work

- A One-Class Support Vector Machine Calibration Method for Time Series Change Point Detection (Jin et al., 2019)
- Dive into Time-Series Anomaly Detection: A Decade review (Boniol et al., 2024)

General idea
One-Class Support Vector Machine (OC-SVM) is an unsupervised variant of the traditional SVM designed for anomaly detection. It is trained only on data representing the "normal" or healthy condition and learns a decision boundary that encloses most of the training data in the feature space. During inference, any data points that fall outside this boundary are considered potential anomalies.

In time series applications, OC-SVM, can differentiate between isolated anomalous spikes and gradual changes in behavior. While the former can be disregarded as noise or faulty measurements, the latter can indicate the system transitioning toward failure. By monitoring the gradual drift, the model can identify *change points* where the behavior changes from normal to anomalous.

Advantages & Limitations
OC-SVM has previously been successfully applied in SHM, including the monitoring of a long-span bridge using acceleration data. The model is capable of handling large datasets efficiently and can be used on a relatively small training sample and still get reliable results. Thus making it suitable for monitoring scenarios with limited monitoring data.

Despite its strengths, OC-SVM has a few limitations one of which is the need for healthy training data. If anomalous data points are included in the training data, the model may fail to detect critical change points rendering it ineffective for anomaly detection. Additionally the anomaly detection requires tuning of the threshold parameters. Improper tuning can lead to false positives or missed anomalies, thereby reducing the reliability of the results.

3.3.3 Model Candidates - Approach 2

In the following sections, two potential models for the task of analyzing strain profiles are presented. Each model in this section follows the blueprint of approach 2 described in Section 3.2.1 and is described in detail using the following points, which provide a comprehensive overview of its key aspects:

1. **Gaussian Mixture Model**

   Reference work

   - A Hybrid Clustering Method for Bridge Structure Health Monitoring (Aiping Guo et al., 2018)

   General idea
   The Gaussian Mixture Model (GMM) is a parametric clustering technique

that models the distribution of data as a combination of multiple Gaussian components, each representing a cluster. In this approach, strain profiles (such as those shown in Fig. 3.3) are first reduced in dimensionality using principal component analysis (PCA), which preserves the maximum amount of variance, thus highlighting the most important features in the data. The GMM is then applied to the reduced data, estimating the probability that each data point belongs to each Gaussian component, thereby enabling soft clustering.

This probabilistic framework allows GMM to capture the underlying statistical structure of the strain profiles, grouping similar behaviors together while isolating those that deviate significantly from the norm. The GMM requires a predefined number of clusters.

Anomalies are detected by identifying clusters that exhibit abnormal behavior or by flagging individual data points that do not fit well into any cluster. These outliers should be carefully evaluated to distinguish between genuine anomalies and artifacts caused by faulty sensors or data errors. Domain expertise is essential for interpreting the underlying behavior of each cluster and labeling them as normal or anomalous, which can subsequently facilitate faster identification of potential issues in future data.

Advantages & Limitations
The GMM-based approach offers practical benefits, including minimal preprocessing and high interpretability. The use of PCA enhances scalability by reducing computational complexity while preserving essential data structure.

Nevertheless, the effectiveness of GMM depends on selecting an appropriate number of clusters and on the quality of the PCA transformation. Although GMMs can be used to evaluate the likelihood of new data points for anomaly detection, its suitability for real-time applications is limited. This is primarily due to the fixed number of clusters, which is set initially and can be restrictive in long-term structural health monitoring scenarios where the optimal number of clusters may evolve over time. This constraint may result in new or rare patterns being forced into existing clusters, potentially masking meaningful anomalies. As a result, GMMs are less flexible for real-time anomaly detection compared to methods that can dynamically adjust to new patterns. Additionally, poorly separated or non-meaningful clusters may occur, and subtle but important variations could be missed if not captured by the principal components.

2. **Dirichlet Process Gaussian Mixture Model**

   Reference work

   - A Bayesian non-parametric clustering approach for semi-supervised Structural Health Monitoring (Rogers et al., 2019)

   General idea
   The DPGMM is a clustering model that incorporates Bayesian non-parametric methods, allowing it to dynamically determine the number of clusters as new data patterns emerge. Unlike traditional clustering approaches that require a

predefined number of clusters, the DPGMM leverages the Dirichlet Process to probabilistically create new clusters when the data cannot be well described by existing ones. This is particularly advantageous in structural health monitoring scenarios where the number of undamaged states is unknown and new patterns may develop over time.

DPGMM extends the standard GMM by introducing Bayesian inference, making the number of clusters a random variable rather than a fixed parameter. Each new data point is assigned to a cluster based on the likelihood that it fits the statistical characteristics of that cluster, with the possibility of forming entirely new clusters as needed. Integrating PCA for dimensionality reduction further enhances computational efficiency, making the method suitable for large datasets and continuous, real-time monitoring.

Additionally, the model can be improved with semi-supervised learning by labeling clusters as they are formed. This allows for automatic anomaly detection: if a data point is assigned to a previously labeled anomalous cluster or triggers the creation of a new cluster, it can be flagged for expert review, enabling real-time diagnosis.

Advantages & Limitations
By dynamically creating new clusters for emerging behaviors, DPGMM eliminates the need to assume the number of clusters in advance, improving generalization and reducing the risk of overfitting or underfitting. It supports automatic anomaly detection, as data points that do not fit into existing clusters naturally lead to the formation of new ones. Since this method detects changes in data behavior rather than relying on a healthy baseline, it can be implemented at any stage of a bridge's lifespan, even after damage has been detected.

However, expert input remains essential to interpret the significance of the underlying behavior represented by newly formed clusters and to distinguish true anomalies from benign changes. The method is also sensitive to the choices of hyperparameters, requiring careful tuning to avoid overfitting or underfitting. Despite these challenges, DPGMM offers increased robustness and adaptability for evolving monitoring scenarios.

3.3.4 Model Selection

After comparing the two approaches proposed, that is to either detect anomalies in strain time series or identifying changes in structural behavior by detecting changes in the strain profiles, it was concluded that both can provide their unique information about the condition of the bridge. The time series approach has the advantage of leveraging the power of deep neural networks to find intricate patterns in the data and by doing so finding anomalies when the measured strain deviates too much from the expected value given the current conditions. The strain profile approach, on the other hand, takes advantage of the high spatial resolution of the fiber optic system, which can give valuable insights on the condition of the bridge, as it considers not just one point but the strains along an entire beam.

This led to the choice of using both approaches. By taking advantage of their individual potential advantages, they could complement each other and reinforce the overall result thanks to the robustness of having two models.

3.3.4.1 Model Selection Approach 1

The Göta älv bridge was already 70 years old when its monitoring system was installed, so the training data may not represent a fully healthy state. This poses challenges for models like OC-SVM, which assume training data is free of anomalies; if change points or damage are present in the training set, the model's effectiveness is compromised and ruled as completely useless. The LSTM is better at handling gradual changes in the data, possibly making it more robust to the issues of not having healthy training data.

Therefore, an LSTM model was chosen for Approach 1.

3.3.4.2 Model Selection Approach 2

Bayesian non-parametric clustering does not require the number of clusters to be specified in advance, making it well-suited for real-life surveillance scenarios where prior knowledge of the behavior of the bridge may be limited. This makes it more flexible and adaptable to changing or unknown data patterns than other clustering methods.

Therefore, the DPGMM was chosen for Approach 2.

3.4 LSTM Model

Developing the LSTM model involved several key steps: preprocessing of the data, designing the model architecture, and then proceeding through training, testing, and validation. Each of these stages were essential to ensure the model's effectiveness and reliability for the task at hand and will be presented in this section.

3.4.1 Data Preprocessing for LSTM Model

To prepare the data for training and testing, the raw strain and temperature data were prepared according to the flowchart in Fig. 3.4 and further described in the following section.

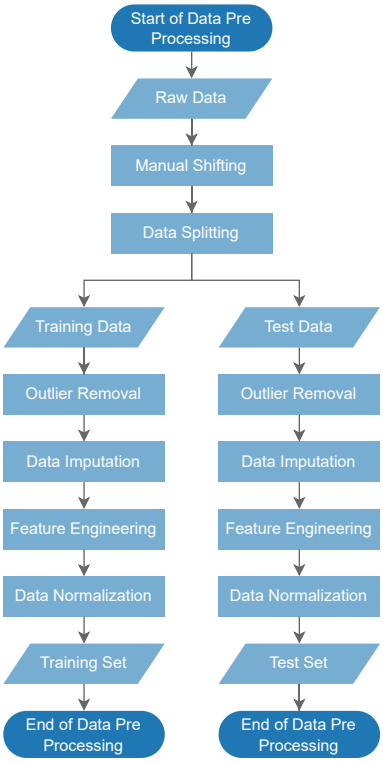

Figure 3.4: Flowchart of data preprocessing strategy.

3.4.1.1 Raw Data

The data preprocessing started with the raw data consisting of strain and temperature for the entire bridge. From this dataset, the relevant data was extracted and restructured to become useful for further analysis. This included extracting time series for specific points along the loops, which corresponds to the points along the loops just over supports I-V for all monitored beams B-F in the part södra

älvbrodelen. All the points that were chosen to be monitored for this project can be found in Tab. 3.2. The naming convention of the measurement loops follows the pattern *direction-beam_part*_Comp.txt_*support_beam_distance*. For example, S-B_Close_Comp.txt_I,B,1.75 refers to a loop located on the south side of the bridge (S), along beam B, and on the part called Close (corresponding to södra älvbrodelen). The measurement point is 1.75 meters from support I, along beam B.

Table 3.2: All points that were chosen to be monitored and modeled.

Point	Beam	Support
S-B_Close_Comp.txt_I,B,1.75	B	I
S-B_Close_Comp.txt_II,B,0.06	B	II
S-B_Close_Comp.txt_III,B,0.06	B	III
S-B_Close_Comp.txt_IV,B,0.06	B	IV
S-B_Close_Comp.txt_IV,B,39.33	B	V
S-C_Close_Comp.txt_I,C,0.55	C	I
S-C_Close_Comp.txt_II,C,0.06	C	II
S-C_Close_Comp.txt_III,C,0.07	C	III
S-C_Close_Comp.txt_IV,C,0.06	C	IV
S-C_Close_Comp.txt_IV,C,39.84	C	V
S-D_Close_Comp.txt_I,D,1.15	D	I
S-D_Close_Comp.txt_II,D,0.06	D	II
S-D_Close_Comp.txt_III,D,0.05	D	III
S-D_Close_Comp.txt_IV,D,0.06	D	IV
S-D_Close_Comp.txt_IV,D,39.43	D	V
S-E_Close_Comp.txt_I,E,0.04	E	I
S-E_Close_Comp.txt_II,E,0.06	E	II
S-E_Close_Comp.txt_III,E,0.06	E	III
S-E_Close_Comp.txt_IV,E,0.06	E	IV
S-E_Close_Comp.txt_IV,E,37.7	E	V
S-F_Close_Comp.txt_I,F,1.6	F	I
S-F_Close_Comp.txt_II,F,0.06	F	II
S-F_Close_Comp.txt_III,F,0.06	F	III
S-F_Close_Comp.txt_IV,F,0.06	F	IV
S-F_Close_Comp.txt_IV,F,38.51	F	V

3.4.1.2 Handling Shift in the Time Series

The first step in the preprocessing was to address the misalignments in the data. To correct this, manual shifting was employed, as this method produced results that most closely resembled the original data, particularly with respect to temporal dependencies, while still allowing for the correction of significant shifts, as illustrated in Fig. 3.5.

The decision of when to use manual adjustment was made subjectively, based on visual inspection of data trends and the identification of gaps that could indicate measurement faults. When preparing the data for training, the data were more

rigorously shifted to correct apparent misalignments, whereas when the test data was being prepared, the data were generally retained in their original form unless there was clear evidence that a realignment was necessary. This careful approach was intended to preserve the integrity of the test data while still addressing obvious errors in the training set.

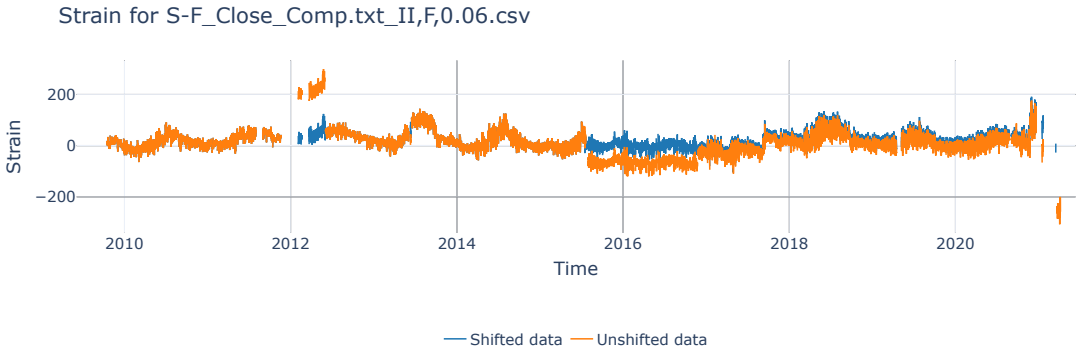

Figure 3.5: Showcase of how the data looks before and after shifting

3.4.1.3 Data Splitting

The dataset was split into 50 % training and 50 % test sets to enable model evaluation. The choice of splitting into equal parts training instead of a comparatively larger portion of training data was made due to the presence of real world observations that are necessary for the validation of the model. This is made clear when the LSTM model validation is explained in Section 3.4.5 and when the model is validated in 4.1.2.

3.4.1.4 Outliers

Outliers were removed using a sliding window approach, with a window size of 100 data points and a step size of 20. For each data frame, the window moved across the time series and flagged strain and temperature values that deviated from the local mean by more than a specified number of standard deviations. Outliers identified in this way were replaced with NaN for later treatment. To ensure robust model training, a stricter threshold of 1.5 standard deviations was used for the training data, while a more lenient threshold of 3 standard deviations was applied to the testing data. This allowed the model to learn from cleaner data while still being evaluated on more realistic, potentially noisy scenarios. The sliding window method was particularly effective in handling the seasonal patterns present in the data, as smaller windows helped to isolate outliers that might otherwise be masked by broader trends, ultimately contributing to both effective anomaly detection and data smoothing.

3.4.1.5 Missing Data

As seen in Section 3.1.3, a significant amount of data is missing. While shorter gaps could be imputed, longer periods, some spanning several months, were not

imputed. A maximum of 20 time steps were imputed using linear interpolation. That corresponds to around three days.

3.4.1.6 Feature Engineering

The raw data consisted of strain and temperature for each time step. To provide the model with more context and to enhance the model's generalization capabilities, new features were created from the existing strain time series and added as new features used to train the model. These features were rolling mean and rolling standard deviation, which were derived from each of the original strain input features. Temperature was also added as a feature, however, no new features were derived from the temperature as it is the strains that are of interest to the model. For beam B, for example, this resulted in 20 input features instead of the 5 features in the original data.

3.4.1.7 Data Normalization

Prior to model training, the features were normalized using min-max scaling to normalize the data to the range $[0, 1]$ to facilitate stable and efficient learning. Due to the varying ranges across feature categories (e.g., strain and temperature), the scaler was fitted on each individual feature category. This preserved the relative structure within each feature category while preventing that feature categories with higher order values dominate over the others. The normalization was fit on the training data and then applied separately to the test data to prevent implicitly leaking information about the range of the test data to the training data.

3.4.1.8 Sequencing the Data

The training and test data were split into sequences of 64 time steps. These sequences were then loaded into a PyTorch DataLoader with a batch size of 16.

3.4.2 LSTM Model Architecture

The LSTM model was constructed using the PyTorch framework. The network consisted of the components shown in Fig. 3.6.

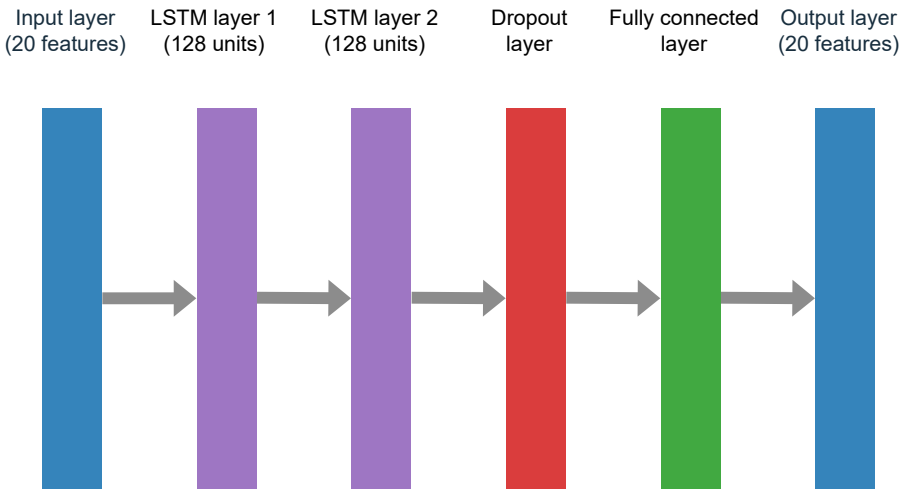

Figure 3.6: Architecture of the LSTM model.

- **Input layer:** The input dimension was set equal to the number of features in the time series, i.e. 20, except for the model that was trained on all features across all beams and supports. In that case, the input dimension was 100 features.
- **LSTM layers:** Two stacked LSTM layers were used, each with a hidden state size of 128. For the model using all 100 features, the hidden dimension was 512, to compensate for the increased complexity.
- **Dropout layer:** A dropout rate of 0.3 was applied after the last LSTM layer to prevent overfitting.
- **Fully connected layer:** The output from the final time step of the LSTM was passed through a fully connected (linear) layer to reconstruct the original input dimension.
- **Output layer:** The final output was a vector of size 20 (except for the model trained on all 100 features), which matches the dimensionality of the input features.

3.4.3 Training

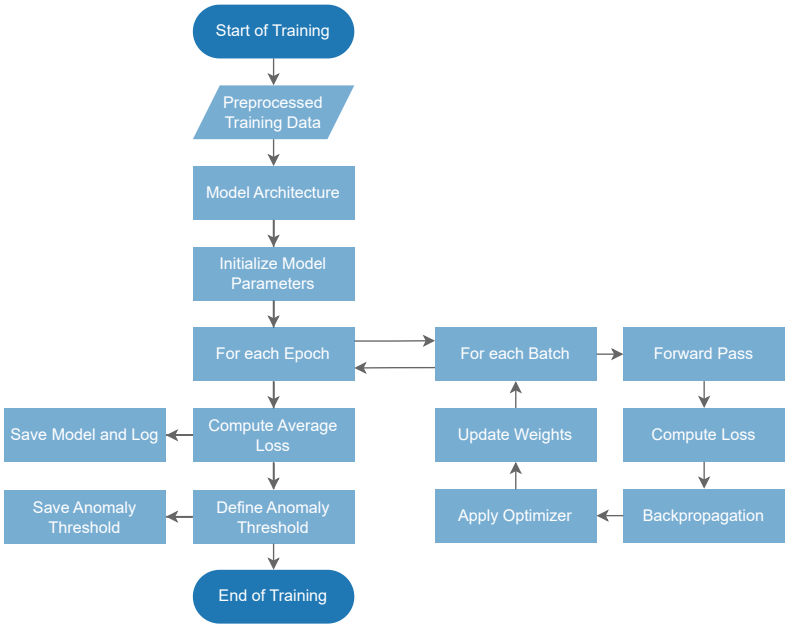

Figure 3.7: Flowchart illustrating the steps involved in the model training procedure.

3.4.3.1 Input

The architecture shown in Fig. 3.6 was used to train a number of different models, each with a different set of input features. These feature sets consisted of various combinations of measurement points from beams B–F and supports I–V, as detailed in Tab. 3.2. In Fig. 3.8, examples of the input constellations are exemplified. Additionally, a comprehensive model was trained using all available features. In total, eleven models were trained.

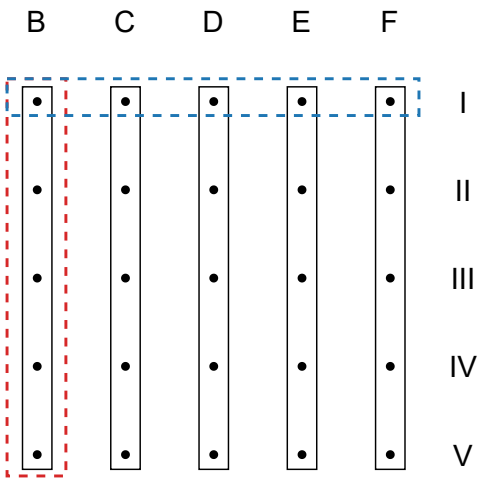

Figure 3.8: Examples showing the input constellations. The sets of input consisted of for example all supports (I–V) for beam B (marked in blue) or all beams (B–F) at support I (marked in red).

3.4.3.2 Loss Function

The model was trained using the mean squared error (MSE) loss function. This regression-based loss measures the squared differences between predicted values and ground truth. Since the aim was to find anomalies in strain, the loss function was minimized only with respect to the strain output. The reason for this was so that the loss function would not be diluted with the losses from features other than strain. The remaining input features were part of the output too, but had no contribution to the loss function. Instead, they served as context to predict the strain. For example, it was known that there was a relation between strain and temperature, seen in Fig. 3.2, and by using temperature as a feature, additional context was provided about its environment and surroundings to the model. Strain input channels from neighboring beams also serve a contextual purpose. If two neighboring beams that normally correlate suddenly deviate, it can be perceived as anomalous to the model.

3.4.3.3 Training Procedure

The model was trained using batches of size 16 for 40 epochs. The PyTorch Dataloader was employed for efficient data batching and loading. A dropout rate of 0.3 was applied to the LSTM layers to mitigate overfitting. The weights were updated via backpropagation using the Adam optimizer and a learning rate of 0.001.

3.4.4 Testing

The testing phase started by loading the model weights obtained during training. The preprocessed test data was then passed through the network in sequences using the test DataLoader, which produced the reconstructed signal as output.

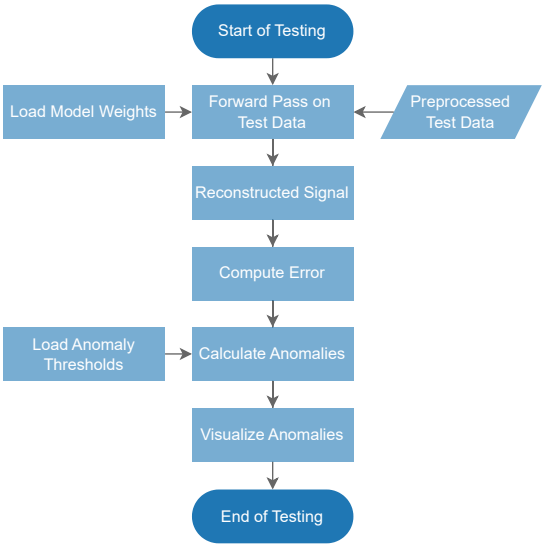

Figure 3.9: Flowchart of the testing procedure.

3.4.4.1 Anomaly Detection

The anomaly classification had two parts: setting the threshold and checking when it was exceeded. During training, mean and standard deviation of the reconstruction error were obtained. These were loaded during the test phase and used to set the anomaly threshold for identifying potential anomalies. To avoid flagging every single time step where the prediction error briefly exceeded the threshold, a rule was added: the rolling mean error had to stay above the threshold for a minimum number of consecutive time steps before being classified as an anomaly. The limit was set to 18 consecutive time steps for robustness, equivalent to three days.

3.4.5 Validation

The validation of the LSTM model posed a challenge due to the absence of labeled data which could have been used to tune the parameters during testing as well as help defining an appropriate anomaly threshold.

To overcome this problem, the alarms that were raised by the old system were exploited. The alarm system was designed to raise alarms to the operators of the bridge when a high value was detected (Glišić et al., 2007). By mapping the alarms

that were raised by the system for the relevant parts of the bridge (see Fig. 3.2), the anomalies found by the LSTM model could be compared against to validate the model and facilitate the parameter tuning. By using 50% of the data for testing some alarms were featured in the test data which enabled the validation of the model.

3.5 Dirichlet Process Gaussian Mixture Model

The clustering model works by first fitting a fixed number of data points to clusters to initialize the model. Thereafter, it uses a sliding window approach to assign new data to clusters or creating new ones if a new data point deviates too much from the prior distributions. A detailed explanation is given in the following section.

3.5.1 Data Preprocessing for DPGMM

Before clustering, the data needed to be preprocessed to enhance the quality and robustness of the clustering algorithm. The goal was to compare the different strain profiles based on the shape and patterns of the strain profile rather than their absolute magnitudes. The different steps of the preprocessing are described below.

3.5.1.1 Data Management

Data from beams C and E were selected for analysis, as their positions directly above the supports make them more susceptible to the effects of displacements, which were described in Section 1.1. Additionally, only data collected at 04:00 were utilized, since traffic volume is minimal at this time, providing measurements that most closely approximate the unloaded state of the structure. By doing so, the long-term remaining changes on the bridge could be captured. Assuming the bridge is unloaded allows the load to be treated as constant between consecutive samples. This assumption facilitates meaningful comparisons that would otherwise be hindered by the lack of information regarding the actual loading conditions on the bridge.

3.5.1.2 Outliers

Outliers were identified and removed from the dataset using a two-step method. The removed outliers were not replaced; instead, the dataset was cleaned by excluding these rows entirely to ensure the integrity of the analysis. The two-step method is presented below:

- **Mean-based row filtering**
  Every row containing missing values were discarded and then the mean value of every row in the dataset was calculated. By comparing the mean value of every row to the interquartile range (IQR) of the entire dataset the rows

outside the bounds were eliminated. The limits are defined as:

$$\text{lower\_bound} = Q_1 - \alpha \cdot IQR \tag{3.1}$$

$$\text{upper\_bound} = Q_3 + \alpha \cdot IQR \tag{3.2}$$

Where α is a multiplier set to 7 in this case. This outlier removal strategy finds entire rows that have values exceeding the normal range.

- **Row-wise thresholding based in deviation from row mean**
  For each row, the mean and standard deviation of absolute values are computed. If any individual sensor reading within a row exceeds a threshold defined as a deviation of α standard deviations from the mean, in this case 7, the entire row is then defined as an outlier and is removed. This strategy finds rows that have singular values that are outliers and then removes the entire row.

3.5.1.3 Missing Data

No imputation of missing data was necessary because any gaps data meant that the entire data loop was missing for that timestamp. Since this approach involves analyzing the entire loop, imputing the missing data would mean creating entire loops made of fictional data, which would not represent anything meaningful.

3.5.1.4 Principal Component Analysis

After the outlier removal, PCA was applied to each strain profile to reduce the dimensionality while still preserving a sufficient amount of variance. The number of principal components was chosen so that at least 90 % of the variance was preserved. To ensure uniform contribution of each component in the subsequent clustering process, the resulting principal component scores were standardized using a z-score transformation (zero mean and unit variance) via the StandardScaler.

3.5.2 Model input

To initialize the clustering the first 1095 data points, equivalent to 3 years, were clustered. The time of 3 years were chosen as it gave the model enough points to create a robust distribution for continued clustering.

A sliding window was designed to simulate new data being added as would in a real monitoring scenario. Once a data point was outside the window it could no longer be reassigned. This so that information about previous clusters would be preserved. The window size was set to 180 days and the step length to 7 days. The choices were made to balance accuracy and computational efficiency.

To emulate a continuous stream of incoming data, a step size for the sliding window was introduced. The smaller the step size the more computational heavy the model would be but but the results would be more reliable. The final choice was 7 data points, one week, as a compromise between computational efficiency and result accuracy.

3.5.3 Clustering Algorithm

In the following section, the algorithm for the DPGMM with a cluster cleaning step is explained. First, a detailed explanation is provided. Thereafter, in Algorithm 1, the algorithm is presented as pseudo code.

Let the dataset be represented by a sequence of data points$\{x_1, x_2, \dots x_t\}$, where $x_t \in \mathbb{R}^d$ is a data point at time t with d principal components.

1. **Initialization:**
   The first n data points are clustered to initialize the clustering structure and to broaden the distribution to better fit future data points.

   $$C_0 = DPGMM(\{x_1, x_2, \dots x_n\}) \tag{3.3}$$

   where, $C_0 = \{c_1, c_2, \dots c_k\}$ is the initial set of clusters based on the distribution of the first n data points and where k is the number of clusters after performing the initial clustering.

2. **Sliding window:**
   A sliding window of size w is defined. The window contains the most recent data points at time t.

   $$W_t = \{x_{t-w+1}, x_{t-w+2}, x_t\} \tag{3.4}$$

3. **Fitting:**
   At each timestep $t > n$, the clustering model is fitted to the entire dataset observed up to time t:

   $$C_t = DPGMM(\{x_1, x_2, \dots, x_t\}) \tag{3.5}$$

   The DPGMM is a nonparametric Bayesian model that defines a prior over the number of clusters. Each data point x_i is assigned to either an existing cluster $c_k \in C_t$ or to a new cluster c_{new} based on the following conditional probability:

   $$p(c_i = k \mid \mathbf{z}_{-i}, \alpha) \propto \begin{cases} n_k \cdot \mathcal{N}(x_i \mid \mu_k, \Sigma_k), & \text{for existing } c_k \\ \alpha \cdot \mathcal{N}(x_i \mid \mu_0, \Sigma_0), & \text{for new cluster} \end{cases} \tag{3.6}$$

   Here:

   - n_k is the number of data points already assigned to cluster c_k.
   - α is the concentration parameter controlling the tendency to create new clusters.
   - $\mathcal{N}(x_i \mid \mu_k, \Sigma_k)$ is the likelihood of x_i under the Gaussian associated with cluster c_k.
   - μ_0, Σ_0 are prior mean and covariance for new clusters.

   The cluster with the highest probability is selected based on the posterior probabilities:

   $$c_i = \arg\max \{p_1, \dots, p_K, p_{\text{new}}\} \tag{3.7}$$

4. **Cluster Assigning:**

   Only the points within the current sliding window

$$W_t = x_{t-w+1}, \ldots, x_t \tag{3.8}$$

   are updated with new cluster assignments:

$$\text{AssignCluster}_t(x_i) = \begin{cases} \text{UpdateCluster}_{C_t}(x_i), & \text{if } x_i \in W_t \\ \text{RetainCluster}_{t-1}(x_i), & \text{otherwise} \end{cases} \tag{3.9}$$

5. **Cluster Cleaning:**
   As the model receives more data, the distribution expands and adapts. Two data points that were initially assigned to different clusters may, at a later time step t, be better represented by the same cluster due to the broadened dataset. To account for this, a cleaning step that merges similar clusters is performed at the end of each time step. This process is described as follows:

$$C_t^{\text{cleaned}} = \text{MergeSimilarClusters}(C_t, \text{distance\_threshold}) \tag{3.10}$$

   where C_t^{cleaned} is the updated set of clusters after cleaning and where the function MergeSimilarClusters() checks the similarity between clusters by comparing their means and covariances. The similarity is assessed using the Mahalanobis distance between the means of two clusters c_i and c_j with covariances Σ_i and Σ_j:

$$\text{dist}(c_i, c_j) = \sqrt{(\mu_i - \mu_j)^\top \left(\frac{\Sigma_i + \Sigma_j}{2}\right)^{-1} (\mu_i - \mu_j)} \tag{3.11}$$

   If the distance between two clusters is smaller than the predefined threshold distance_threshold, the clusters are merged into a single cluster. Otherwise, they remain separate. The resulting cleaned clusters are then returned:

$$C_t^{\text{cleaned}} = \{c_1, c_2, \ldots, c_k\} \tag{3.12}$$

6. **Step:** The window moves a step forward and the processes 3–6 are repeated.

Algorithm 1 Dirichlet Process Gaussian Mixture Model with Cleaning Step

Require: $\{x_1, x_2, \ldots x_T\}$, initial number of points $n > 0$, window step *step*, sliding window size w, distance threshold $d_{\text{threshold}}$

```
C_0 ← InitialClustering({x_1, ..., x_n})        ▷ Initialize clusters with first n points
for t = n + step to T do                          ▷ For each new time step t
    W_t ← {x_{t−w+1}, ..., x_t}                   ▷ Define sliding window
    Initialize C_t with current cluster assignments up to time t
    for each data point x_i in {x_1, ..., x_t} do
        Remove x_i from its current cluster (temporarily)
        for each existing cluster c_k ∈C_t do
            Compute p_k ∝ n_k · N(x_i | μ_k, Σ_k)
        end for
        Compute p_new ∝ α · N(x_i | μ_0, Σ_0)
        Normalize {p_1, ..., p_K, p_new} so that Σ_{k=1}^{K+1} p_k = 1
        Assign cluster c_i = arg max {p_1, ..., p_K, p_new}
    end for
    for each data point x_i in W_t do
        if x_i ∈ W_t then
            AssignCluster_t(x_i) ← UpdateCluster(x_i, C_t)      ▷ Assign new cluster
        else
            AssignCluster_t(x_i) ← RetainCluster_{t−1}(x_i)     ▷ Keep previous cluster
        end if
    end for
    C_t^cleaned ← MergeSimilarClusters(C_t, d_threshold)        ▷ Clean clusters by merging
similar clusters
end for
```

3.5.4 Validation

To assess the feasibility of a clustering approach for anomaly detection based on changes in strain profiles, it was necessary to first determine whether it is possible to make meaningful clusters from the strain profile data. This was done in several steps, which included:

- Verification of the PCA by plotting the explained variance and finding a suitable number of principal components to keep,
- plotting and visual inspection of the clusters,
- plotting the assignment to clusters over time using both a GMM and a DPGMM and compare,
- plotting the cluster means and standard deviation for both models and compare.

These steps were applied to a standard parametric GMM, and the resulting outputs were then used to validate the performance of the non-parametric DPGMM with a sliding window approach. If similar clustering had been done with both models then the validity of the DPGMM model could be deemed as promising. The results from the validation is presented in the following chapter.

4

Results

In this chapter, the results from both the LSTM models and the DPGMM are presented. The performance of the LSTM models in reconstructing the long-term fiber optic strain signals is shown alongside their ability to highlight anomalous behavior. The output of the DPGMM clustering is also included to illustrate how patterns in the strain data are grouped over time. Comparisons between normal and anomalous signal patterns are provided, and key observations related to the structural behavior of the Götaälv bridge are discussed to support the overall analysis. The findings demonstrate that the DPGMM was able to successfully identify new behaviors in the bridge while the LSTM models lack ways of validating the results and because of unhealthy training data they missed potential anomalies.

the complementary strengths of these methods for improving anomaly detection in structural health monitoring

4.1 LSTM Model Results

In the following section, the results from Approach 1 using the LSTM-model for time series anomaly detection are presented. To evaluate the algorithm's performance and ability to find anomalous patterns in the data from södra älvbrodelen, several models with different input setups were created, trained and evaluated.

4.1.1 Model Training

The models were trained as described in Section 3.4.3. Different models were created for several constellations of input signals, grouped by loop, support and with all signals combined to one large model according to Tab. 3.2 and Fig. 3.8. The training losses calculated using MSE after 40 epochs are presented below in Tab. 4.1.

Table 4.1: Training loss for the models after 40 epochs of training.

Model	Training Loss
Beam B	0.001049
Beam C	0.001291
Beam D	0.001000
Beam E	0.000788
Beam F	0.001216
Support I	0.001191
Support II	0.001210
Support III	0.001191
Support IV	0.000953
Support V	0.000910
All	0.001372

4.1.2 Model Validation

The goal of the validation was to adjust the model to detect documented events on relevant parts of the bridge. These included high strain values which triggered alarms and displacements observed at support I. Between the years 2009 and 2021, the monitoring system had raised alarms for high values a number of times. The time and location of these alarms have been summarized in Fig. 4.1–4.3. A marked box means that at that specific date and time, an alarm was raised somewhere along the giving loop. The different alarms were T3a, T3b and T4, where T4 indicates a failure in the structure and T3a and T3b indicates high elongation and increased elongation respectively.

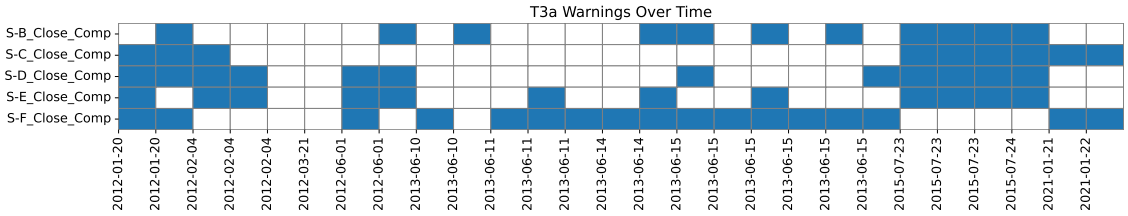

Figure 4.1: T3a alarms over time for beams B-F.

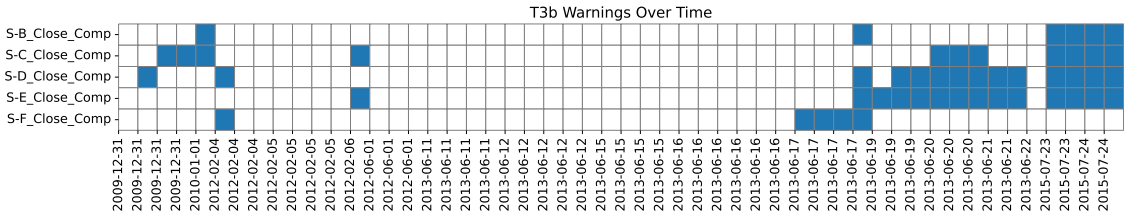

Figure 4.2: T3b alarms over time for beams B-F.

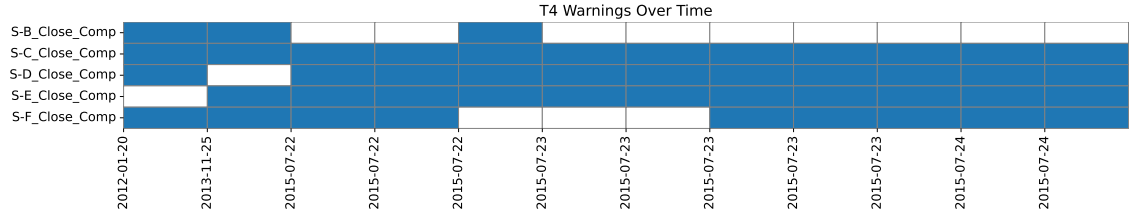

Figure 4.3: T4 alarms over time for beams B-F.

Since the split between training and test data was done chronologically, the alarms toward the end of life were of most interest for the validation and calibration of an anomaly threshold. According to Fig. 4.1–4.3, 22–24th of July 2015 stand out as dates of high interest for detecting high values. It was also the last time this type of alarm was triggered. This was the reason behind the 50/50 split which preserved timestamps test set that had triggered alarms.

However, none of these dates were flagged as anomalies by the models and the alarms for high values could thus not be used as a method of validation and threshold calibration. The reconstructed test signal and the corresponding anomalies can be seen in Appendix A.

Some of the models did, however, flag anomalies around the time when displacements were observed at support I. This suggests that the models may be sensitive to the strain changes caused by these displacements and thus identified them as anomalies. However, since not all models detected this potential anomaly, it is difficult to assess the reliability and consistency of this prediction, and thereby also the method of using it for validation. The model trained on support II is one example of anomaly flags around that time and it can be seen in Fig. 4.4.

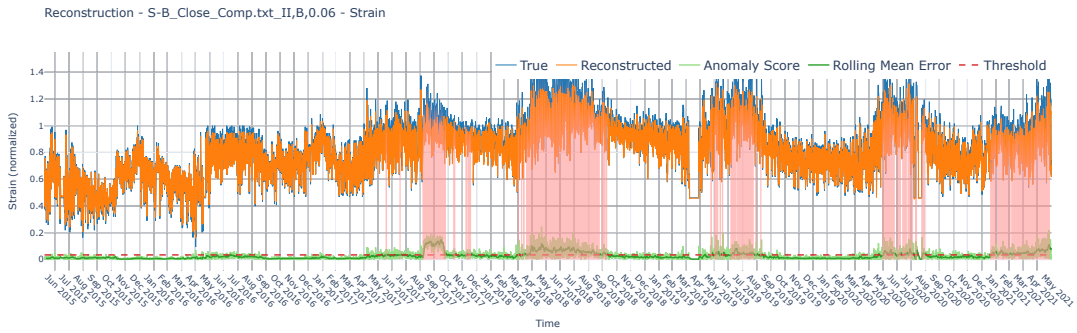

Figure 4.4: True vs. predicted signal for S-B_Close_Comp at support II. The predictions and anomalies come from the model that is trained on support II, according to the constellations in Tab. 3.2.

4.1.3 Test and Anomaly Detection

When evaluated on the test set, the model demonstrated good generalization to unseen data, achieving a consistently low prediction loss. In addition to successfully reconstructing visibly normal sequences of data, the model also successfully

predicted sequences that were visibly abnormal with sharp drops and other irregularities (seen in Fig. 4.5, in April of 2019), which may have masked some of the potential anomalies.

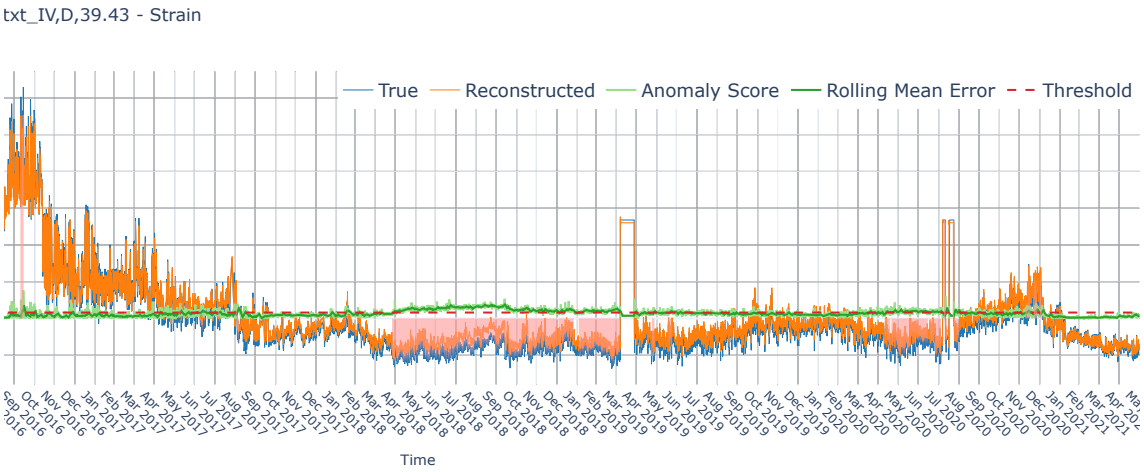

Figure 4.5: Sign of reconstruction of abnormal data.

Some of the models trained on the data along each beam B-F detected anomalies at a high rate in early-mid 2018. This was observed on beam B at support I, II and IV (Fig. 4.6), for beam C at support II (Fig. 4.7), for beam D at support II and V (Fig. 4.8), and for beam F at support V (Fig. 4.9). In the case with beam D at support II and support V and beam B at support IV, the potentially anomalous behavior remained into 2019.

Figure 4.6: True vs. predicted signal for S-B_Close_Comp at support I (top), II (middle) and IV (bottom). The predictions and anomalies come from the model that is trained on beam B, according to the constellations in 3.2.

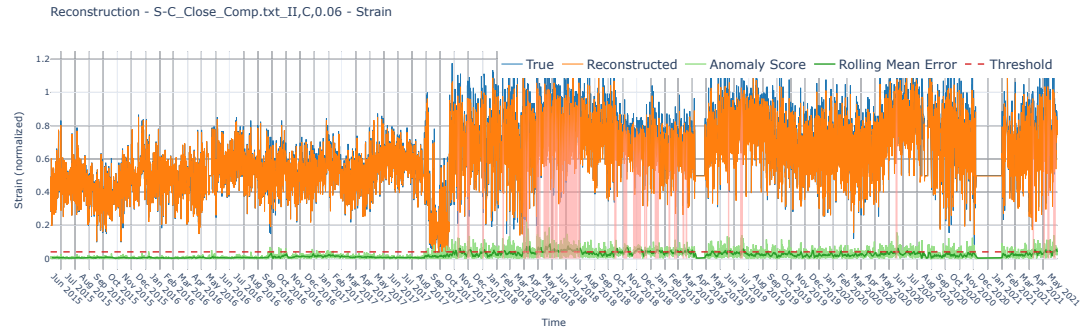

Figure 4.7: True vs. predicted signal for S-C_Close_Comp at support II. The predictions and anomalies come from the model that is trained on beam C, according to the constellations in Tab. 3.2.

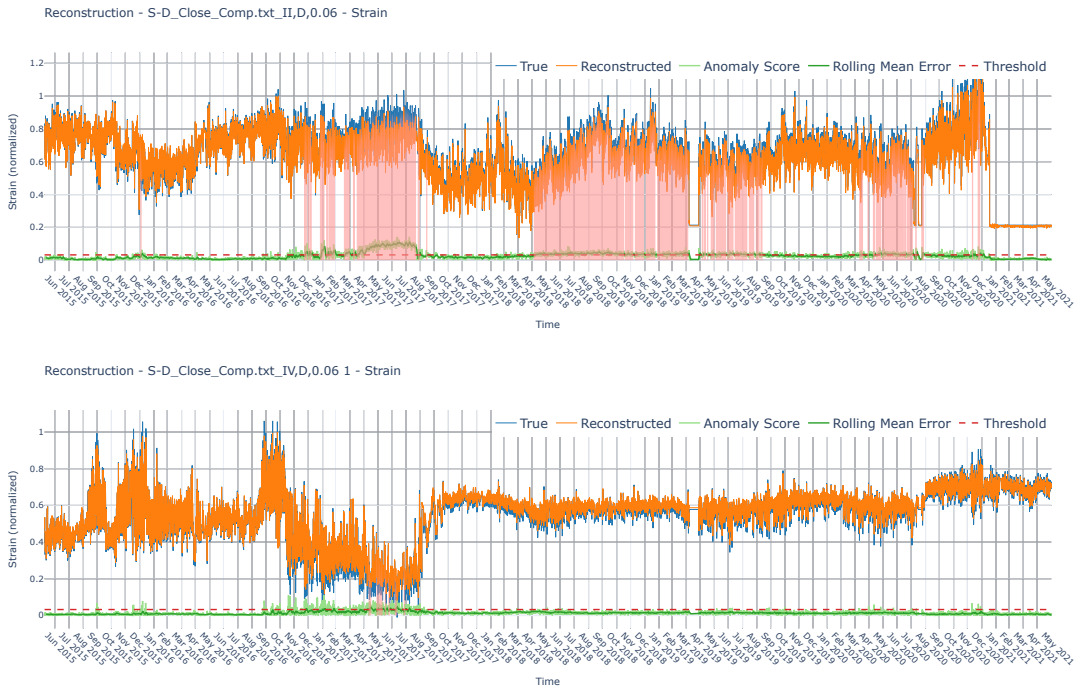

Figure 4.8: True vs. predicted signal for S-D_Close_Comp at support II (top) and V (bottom). The predictions and anomalies come from the model that is trained on beam D, according to the constellations in Tab. 3.2.

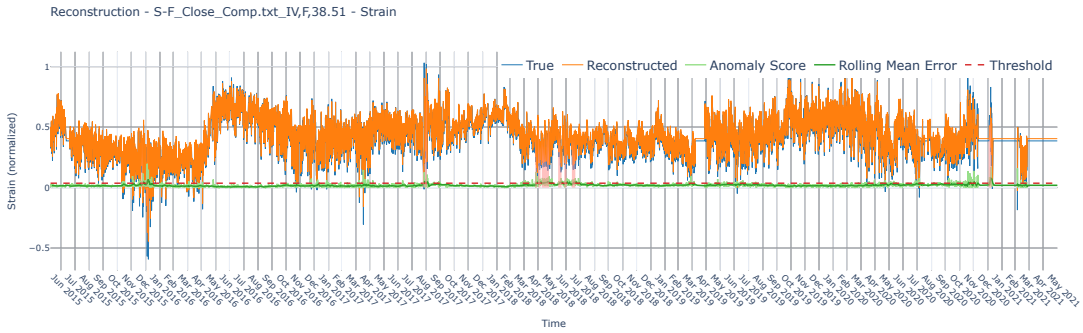

Figure 4.9: True vs. predicted signal for S-F_Close_Comp at support V. The predictions and anomalies come from the model that is trained on beam F, according to the constellations in Tab. 3.2.

Between April and August 2017, strains larger than expected were observed at support II along beam D, see Fig. 4.8. While it is difficult to derive the exact cause of this anomalous period, it does coincide with a period of large ground movements due to groundwork at nearby sites, which had to be halted due to the risk of damage to the Götaälv bridge (Adrian et al., 2021).

4.2 DPGMM Results

In the following section, the results from Approach 2 using the the DPGMM clustering model are summarized. First a validation of the method itself is presented followed by the results with a focus on beam C and E. Results from the remaining beams have been extracted and plots are available in Appendix B.

4.2.1 Feasibility Assessment of Clustering Strain Profiles

To assess the feasibility of a clustering approach inspired by the work of (Rogers et al., 2019) for detecting changes in strain profiles, it was necessary to first determine whether it is possible to make meaningful clusters from the strain profile data. This was done through the validation technique described in Section 3.5.4. The idea behind the method is to identify changes in structural behavior by detecting changes in strain profiles. The feasibility study aims at confirming the initial assumptions that structural behavior is characterized by strain profile and that the strain profiles do not change randomly but remain relatively constant unless a change occurs to the structural behavior.

These steps were applied to a standard parametric GMM clustering model to then be used to validate the results from the DPGMM clustering in a later step.

4.2.1.1 Dimensionality Reduction with PCA

PCA was applied to the strain profile dataset to assess the potential for dimensionality reduction. As shown in Fig. 4.10, the first eight principal components together explain over 90 % of the total variance in the data. This high proportion indicates that most of the information in the original dataset is preserved. Therefore, reducing the dataset to these principal components before performing clustering is likely to be effective.

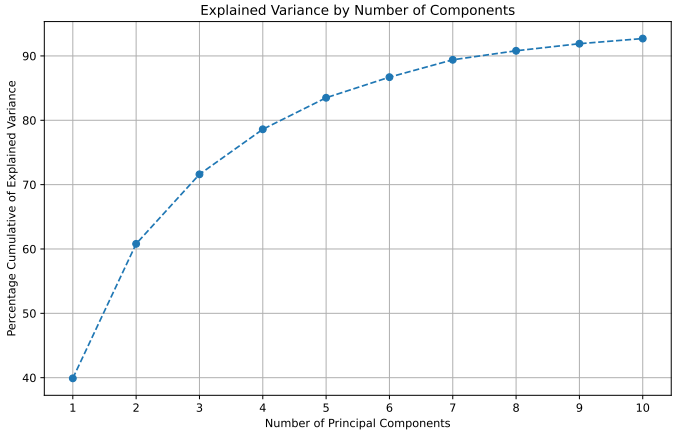

Figure 4.10: The explained variance plotted for a number of principal components. The dataset used for this particular example is from beam C where the original data had 1609 strain components.

4.2.1.2 Clustering Using Parametric GMM

Next, a clustering using a standard parametric GMM was performed using the eight normalized principal components from beam C. From Fig. 4.11 it can be seen that the clusters seem to have a decent separation in their two first principal components. Having a separation between the clusters is necessary for a meaningful behavior identification and for the initial hypothesis to hold true for this model.

Each data point in Fig. 4.11 corresponds to a single strain profile at a specific timestamp, represented in the space of its principal components.

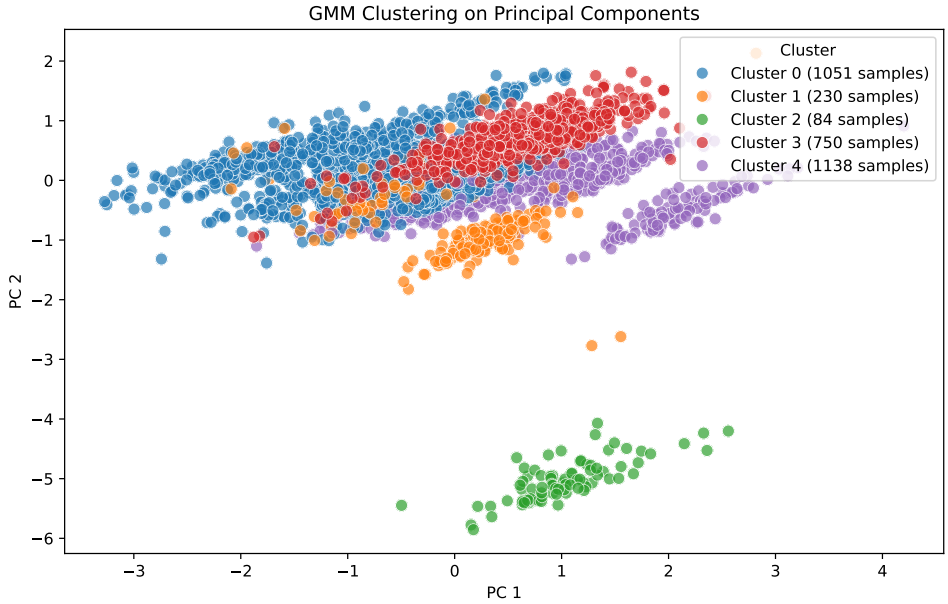

Figure 4.11: Clustering using parametric GMM with the first two principal components plotted.

4.2.1.3 Plotting Cluster Assignment over Time

To assess the robustness of the method and ensure that the models are not assigning data points to random clusters, the cluster assignments over time were plotted. If the theory illustrated in Fig. 3.3 holds—that strain profiles remain relatively stable over time—then a desirable outcome would be consistent assignment of data points to the same cluster during stable periods, with transitions to new clusters occurring gradually or abruptly. Such behavior would suggest that the model forms meaningful clusters and that the cluster evolution over time reflects the underlying structural behavior, supporting the initial hypothesis.

Fig. 4.12 from beam C shows a continuous cluster assignment over time. As hypothesized, there are both clear cuts between two periods of different clusters being assigned to as well as periods of transition where the model is alternating between two clusters. It is worth mentioning that Fig. 4.12 presents the results from the model after being provided with the entire dataset for all years, while the aim of the

final DPGMM model is to continuously feed the model with data as new data are being collected and by doing so simulate a real world scenario.

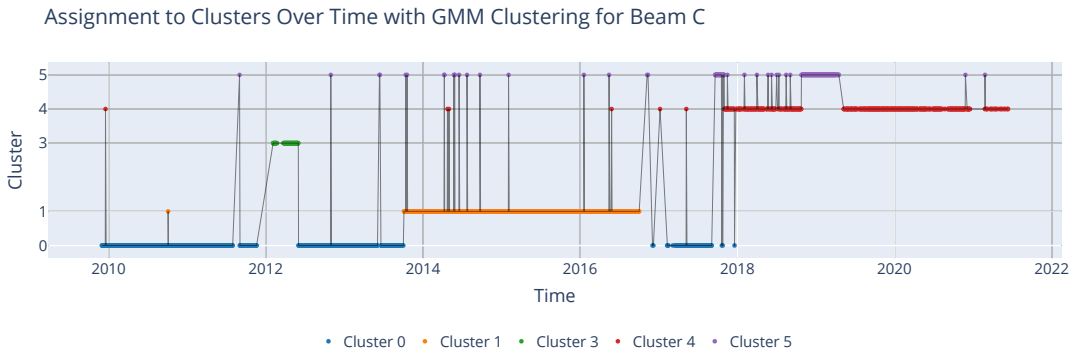

Figure 4.12: The cluster assignments over time by the GMM model for beam C.

4.2.1.4 Cluster Representations

To further evaluate whether it is possible to make meaningful clusters of the data and if the clusters represent a common behavior, the underlying behaviors represented by each cluster were visualized. To do this, the strain profiles were grouped by cluster assignment and the mean profile for each cluster was computed using all strain profiles within that cluster. The mean strain profile representations of the different clusters were then compared.

To characterize the strain behavior within each cluster, the mean strain profile was defined as shown in Eq. 4.1.

$$\bar{\epsilon} = (\bar{\epsilon}(x_1), \bar{\epsilon}(x_2), \ldots, \bar{\epsilon}(x_n)), \tag{4.1}$$

where each $\bar{\epsilon}(x_i)$, as calculated in Eq. 4.2, represents the average strain at spatial position x_i across all k samples in the cluster:

$$\bar{\epsilon}(x_i) = \frac{1}{k} \sum_{m=0}^{k} \epsilon(x_{i,m}). \tag{4.2}$$

Here, $\epsilon(x_{i,m})$ denotes the strain at position x_i in the m-th sample of the cluster.

In addition, the standard deviation σ_i was calculated for each $\bar{\epsilon}(x_i)$ along the length of the profile to quantify the variation in strain and assess the internal similarity of profiles within each cluster. The results obtained using GMM clustering are shown in Fig. 4.13, where the positions of each support are labeled using roman numerals I–V.

Clusters 0, 1, and 4 exhibit mean profiles that closely represent their respective individual strain profiles, as indicated by their relatively low standard deviations. In contrast, cluster 5 shows more variation around its mean, suggesting a higher degree of dissimilarity among its strain profiles. Consequently, the mean profile of cluster 5 is a less reliable representation of the underlying behavior.

Cluster 3 has a very large range and has been removed in this figure to highlight the variance in the other clusters. Cluster 3 can be seen in Fig. 4.14 separately.

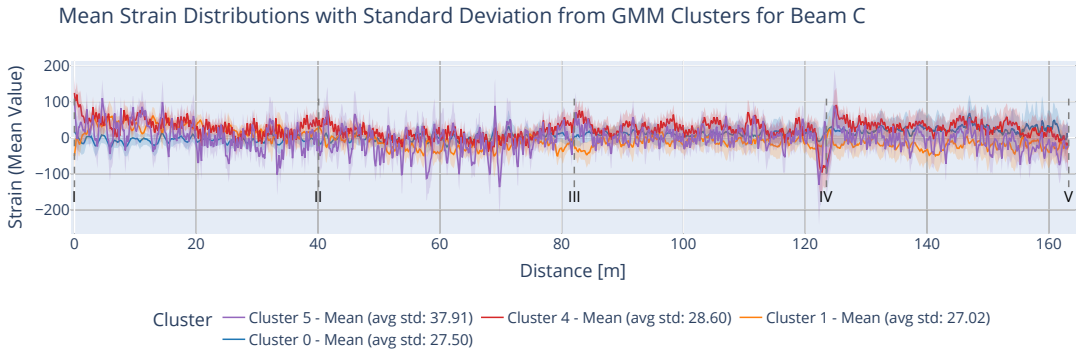

Figure 4.13: Mean strain profiles and average standard deviations for clusters identified by GMM on beam C. Each curve shows a cluster mean and average standard deviation (shaded) indicates how closely individual profiles match their cluster mean for each point x_i.

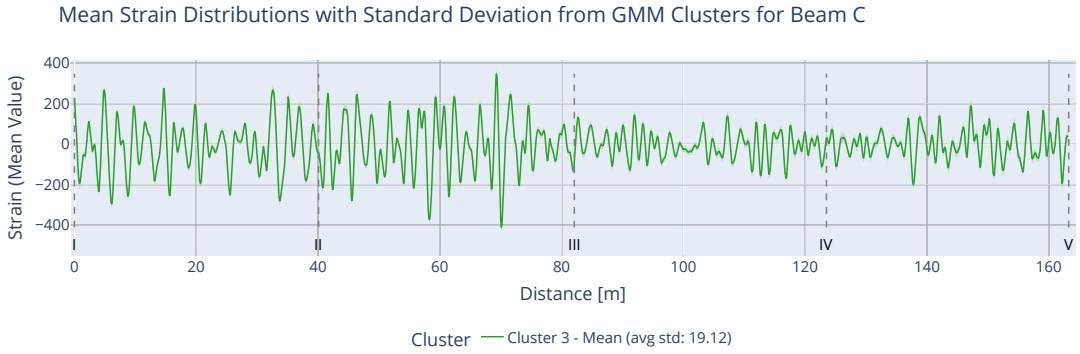

Figure 4.14: Mean strain profile and average standard deviation for cluster 3 for beam C.

4.2.2 Results for Beam C using DPGMM

In the following section, the results using the DPGMM on the dataset for beam C are presented (beam C can be seen in the cross sectional view in Fig. 1.2, followed by the corresponding results for beam E in Section 4.2.3.

4.2.2.1 Plotting the Clusters

When running the DPGMM over the dataset, it resulted in the cluster assignments seen in Fig. 4.15. Eight principal components were used for the clustering but only combinations of the first four components are displayed in Fig. 4.15 since they represent the majority of the variance.

The model finally yielded five active clusters. The number of clusters is controlled by the concentration parameter α that controls the DPGMM's tendency to create new clusters.

The clusters shown in Fig. 4.15 are well separated within the component space used for visualization. This suggest that the model was able to isolate different behaviors. It is also worth pointing out that since there are parts of cluster 0 that are far away from the other, the perspective may be distorted which may lead to the impression that the other clusters are less separated than they in fact are, due to the scale.

One can note the absence of a cluster 2 in the plot. The reason lies in the way the cluster merging function operates. After having updated the data points within the current window, the model merges clusters according to Alg. 1. In this process, locally indexed clusters receive global indices. In a case where 2 has been used as a local index and where that cluster has been merged with another cluster, index 2 is not reused, which explains why it does not reappear.

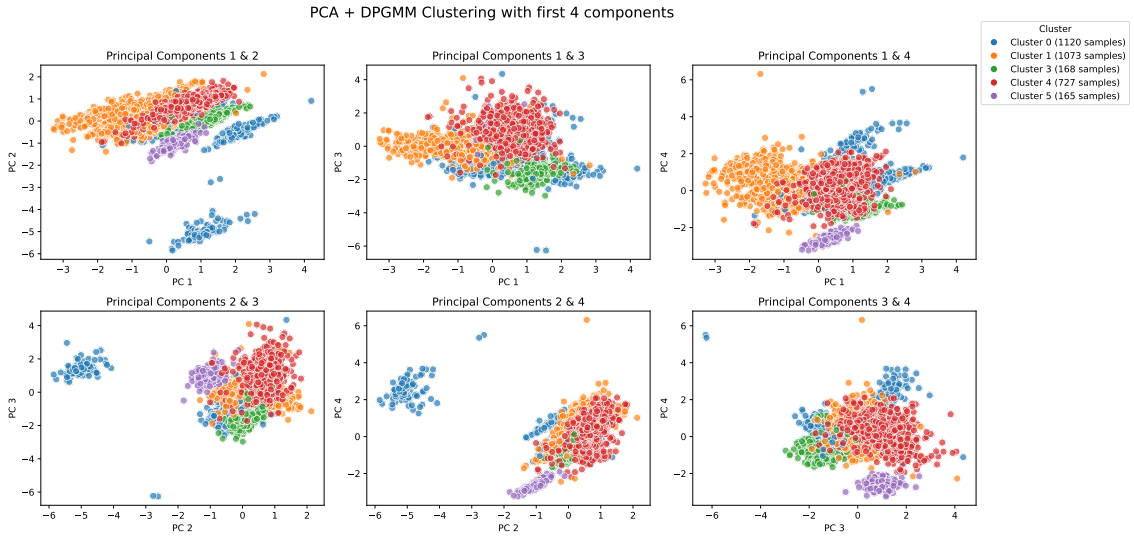

Figure 4.15: Clusters prediction by the DPGMM plotted as combination of the first four principal components for beam C.

4.2.2.2 Cluster Assignment over Time

Similarly as was done for the parametric GMM, the cluster assignments over time were plotted for the DPGMM on beam C. As seen in Fig. 4.16, there are clean cuts between periods of different clusters. There is also a transition period from late 2017 until late 2018, where the model prediction alternates between different states. The representation of the respective clusters is presented in the next section. It can also be observed a less noisy plot for the DPGMM model in Fig. 4.16 compared to the GMM in Fig. 4.25.

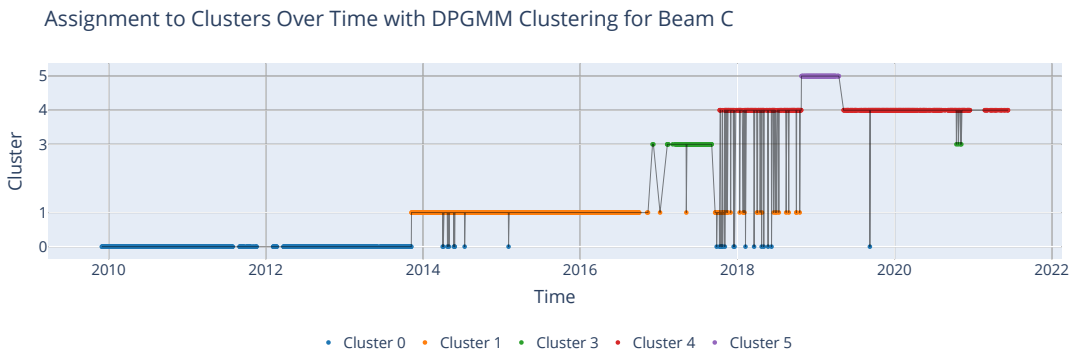

Figure 4.16: Cluster assignment over time on beam C using DPGMM clustering with a sliding window approach.

4.2.2.3 Cluster Behavior Representations

The mean strain profiles and standard deviations, calculated as described in Section 4.2.1.4 for the clusters identified in Fig. 4.15 are presented in Fig. 4.17 below. The DPGMM model has distinguished five different behaviors for beam C, represented by the mean strain distribution for each strain profile within each cluster. Some of the profiles exhibit notable similarities. By examining the strain profiles representing each clusters in order, corresponding to the chronological order of the clusters appearing in Fig. 4.16, a clear transient change in behavior can be observed.

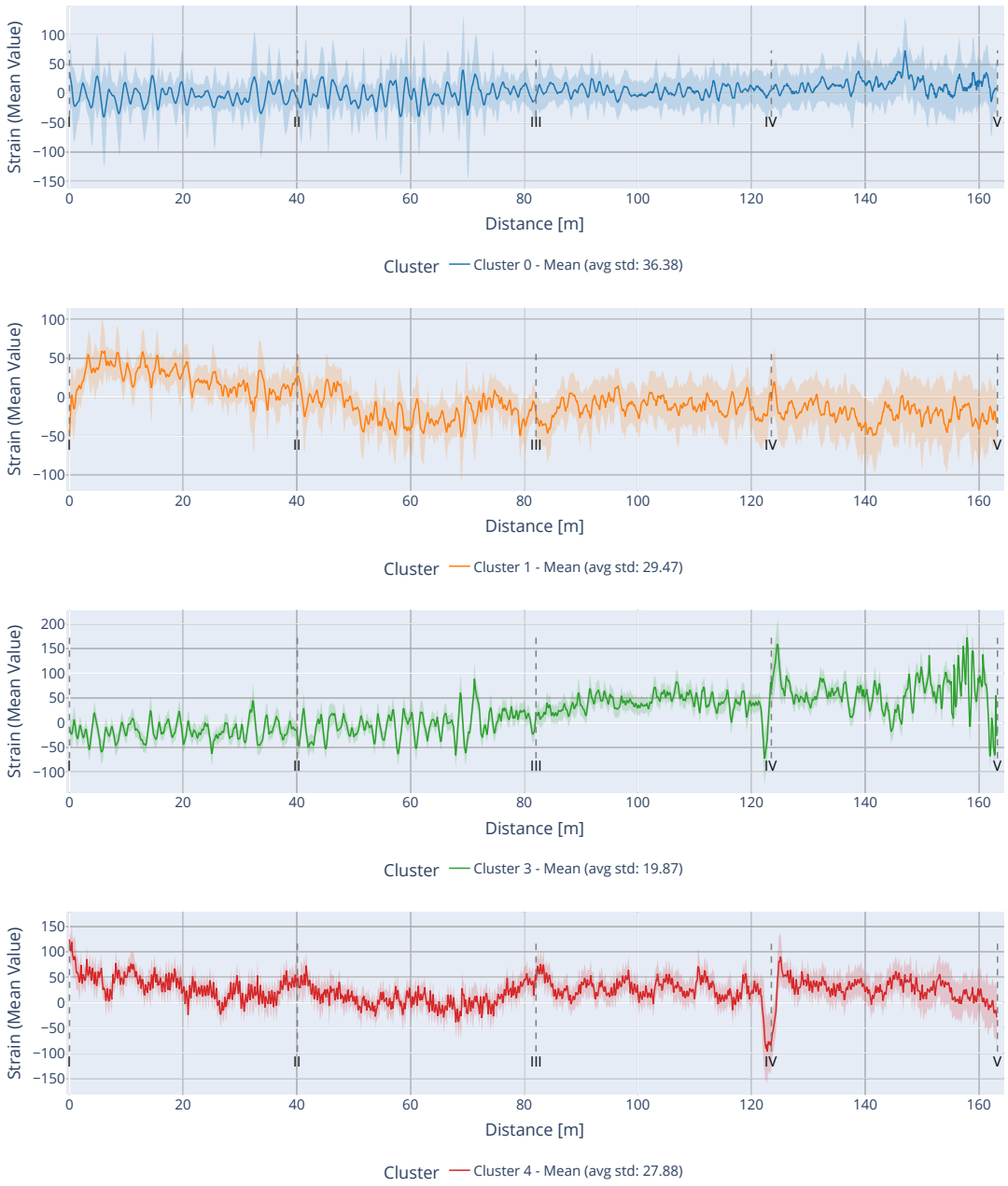

Figure 4.17: Mean strain distributions (1/2) from DPGMM clustering for beam C

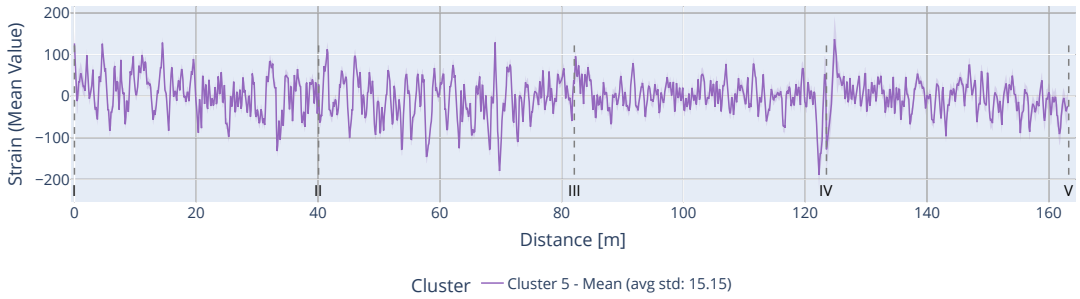

Figure 4.17: (continued) Mean strain distributions (2/2) from DPGMM clustering for beam C

The mean strain representation for cluster 0 and 1 are similar with the difference that cluster 1 is experiencing higher strains in field between support IV and V, while cluster 0 has a more uniform profile along its entire length.

A notable change starts to appear with the formation of cluster 3 in late 2016. With the new behavior that is represented by cluster 3, a sharp spike appeared at support IV. Coming from support III, the strain plunges close to the support and rises rapidly again over the support. At support V some disturbance also seem to have appeared.

In late 2017, a new cluster formed. This time it is cluster 4 that represents a new behavior. The feature with the sharp drop at support IV remains, although the shape is slightly different. What sets the overall behavior of cluster 4 apart from cluster 3 are the spikes around supports I–III and the drop at support V. Given that the drop in support IV remains, even though the overall behavior has changed, could be an indication that an irreversible change has occurred at support IV.

Cluster 5 appears for a brief period of time before data points continue being reassigned to cluster 4. In fact, cluster 5 can be seen as a noisier variety of cluster 4. It still features the sharp drop at support IV and the light peaks around supports I–III, although hidden by the surrounding noise.

The transient change becomes more apparent when plotting the strain profiles from clusters 0 and 4 in the same frame, like in Fig. 4.18. From the relatively uniform strain profile in blue that represents the behavior of beam C at the time when the model was assigning points to cluster 0, to the red line that represents the behavior at a later moment in time, a significant change has taken place.

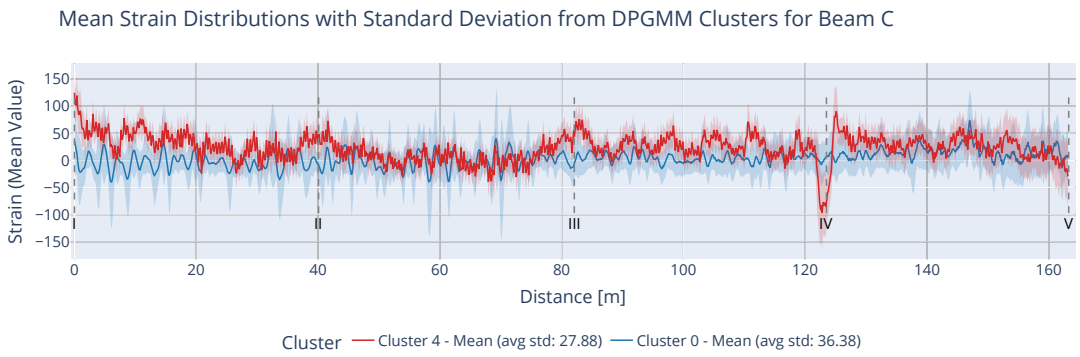

Figure 4.18: Comparison between the first and last assigned clusters for beam C.

4.2.3 Results for Beam E Using DPGMM

Results for all the beams were extracted but in this section the results from Beam E will be presented, as seen in the cross-sectional view in Fig. 1.2.

4.2.3.1 Plotting the Clusters

When running the DPGMM over the dataset, it resulted in the cluster assignments seen in Fig. 4.19. Eight principal components where used for the clustering but only combinations for the first four components are displayed in Fig. 4.19 since they represent the majority of the variance.

Similarly to beam C, beam E is also missing a cluster index. In this case it is cluster 6 and the reason for its absence is the same as explained in Section 4.2.2.1.

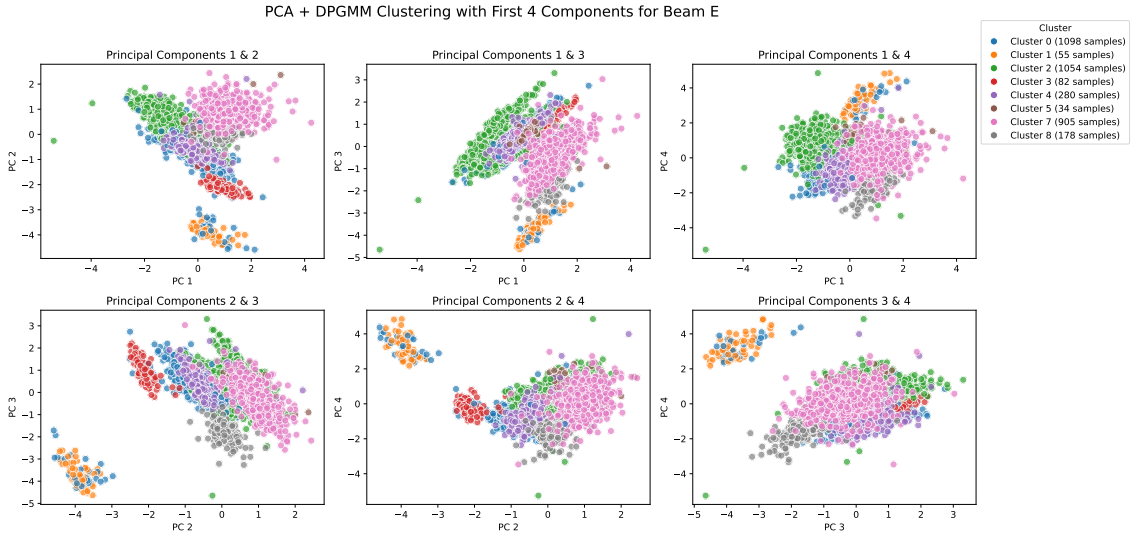

Figure 4.19: Clusters prediction by the DPGMM plotted as combination of the first four principal components for beam E.

4.2.3.2 Cluster Assignment over Time

Cluster assignments over time was plotted for the DPGMM on beam E. As seen in Fig. 4.20, there are some clean cuts between periods of different clusters. Here too, there is a transition period from early 2017 until late 2018, similar to the cluster assignment over time for beam C, where the model prediction alternates between different states.

The representation of the respective clusters is presented in the next section.

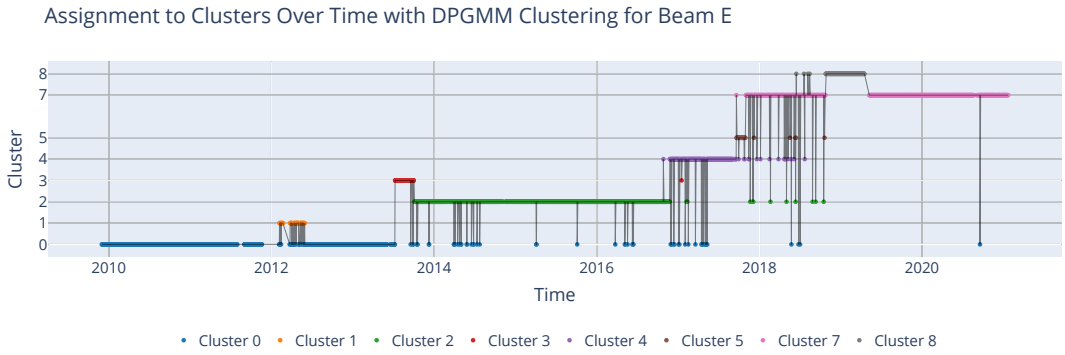

Figure 4.20: Cluster assignment over time on beam E using DPGMM.

4.2.3.3 Cluster Behavior Representations

In Fig. 4.21 below, the mean strain profiles for the clusters showed in Fig. 4.19 are presented. While eight distinct behaviors can be observed, several of the profiles exhibit notable similarities. By examining the strain profiles representing each clusters in order, corresponding to the chronological order of the clusters appearing in Fig. 4.20, a transient change in behavior can be observed for beam E just like for beam C.

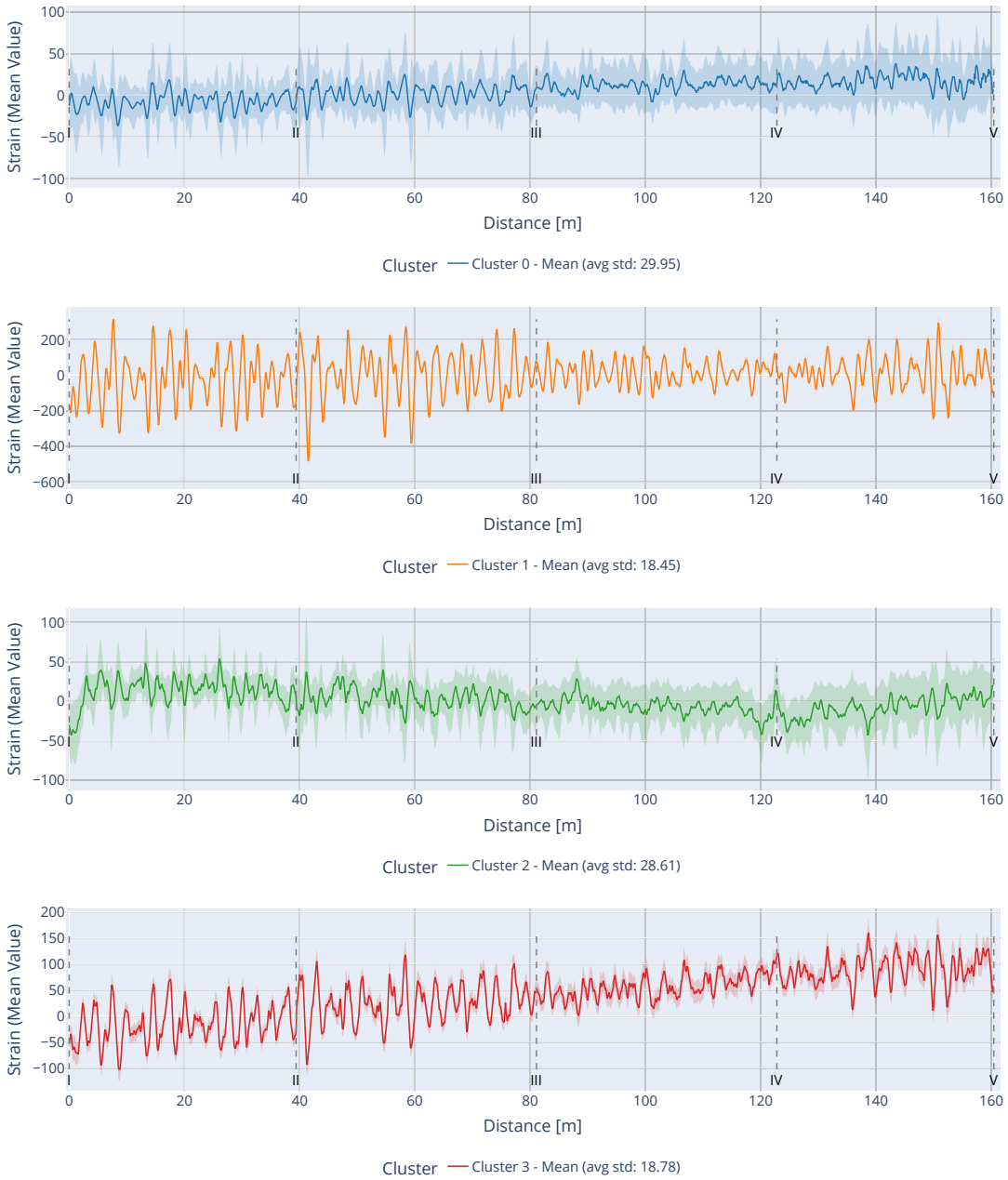

Figure 4.21: Mean strain profiles (1/2) from DPGMM clustering for beam E.

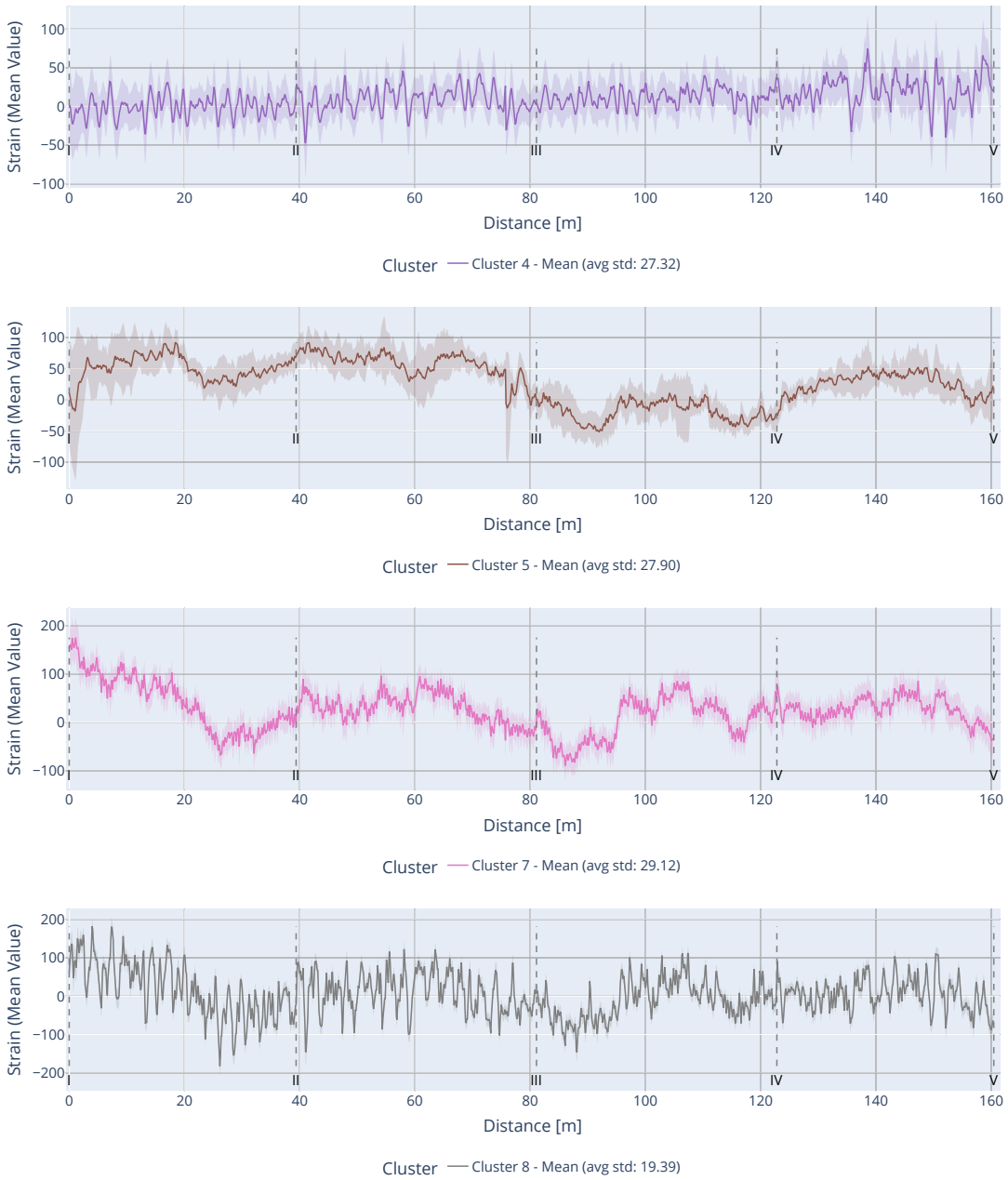

Figure 4.21: (continued) Mean strain profiles (2/2) from DPGMM clustering for beam E.

Cluster 0 has a relatively uniformly distributed mean strain and its graph is slightly tilted, indicated by a higher mean strain in the span between supports IV-V than between I-II.

The mean strain profile for cluster 1 can be seen as a noisier variety to cluster 0. It has local peaks at the same places. This would seem accurate as in Fig. 4.20 cluster 1 is temporary before returning to the behavior represented by cluster 0. The main difference is that cluster 1 does not have the same tilt as cluster 0.

The noticeable differences between cluster 0 and cluster 2 is that the tilt is reversed

between supports I-IV but changes between IV-V, making support IV a global minimum despite it having a local peak. A local negative peak at support I has also formed.

In cluster 3 a more prominent tilt than in cluster 0 has emerged and support IV is no longer a global minimum. The local negative peak at support I from cluster 2 has disappeared.

Cluster 4 is relatively similar to cluster 0 but has noisier spans between support II-III and IV-V. Up until cluster 4, the overall behavior has not gone through any significant changes.

Cluster 5 has more of an oscillating pattern than the previous clusters. The spans between supports I-III has increased strain values while the strain at support III is approximately 0.

For cluster 7 the strain values of support II is also approximately 0 but the oscillating pattern is still apparent. The strain values for support I have drastically increased from cluster 5. Observe the overall increase in range for the representation of cluster 7 compared to that of cluster 5.

Cluster 8 can be seen as a noisier variety of cluster 7. As seen in Fig. 4.20, cluster 8 only appears briefly after a transition period between clusters 4, 5 and 7.

The difference between clusters 0 and 7 becomes clear when their strain profiles are plotted together, as shown in Fig. 4.22. The blue line, representing beam E during the period associated with cluster 0, appears nearly straight. In contrast, the pink line for cluster 7, recorded at a later time, shows a pronounced change in the strain profile. This shift is especially evident at support I and V and in the spans between supports, highlighting a significant alteration in the beam's structural response. Like it was observed for beam C, the change in behavior that appeared with the formation of cluster 5 persists.

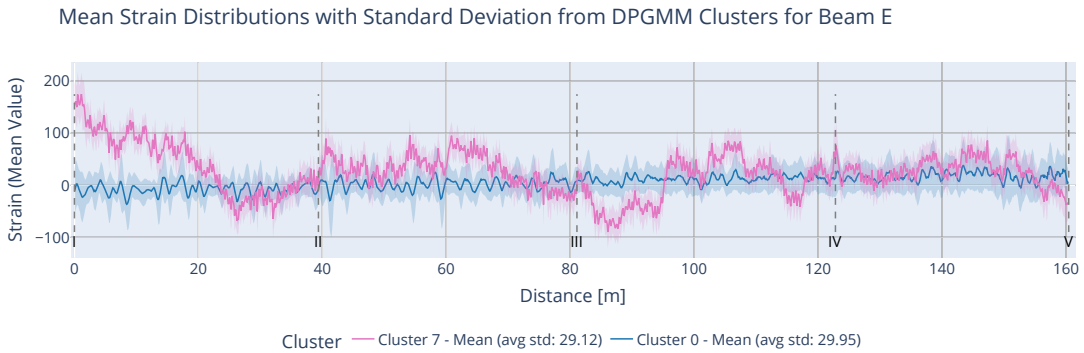

Figure 4.22: Comparison between the first and the last assigned cluster in beam E using DPGMM.

4.2.4 Comparison of Results in Beam C and E

The dates marking the formation of new clusters, indicating changes in behavior, are shown in Fig. 4.23. The dates come from the figures showing the cluster assignments

over time, Fig. 4.16 for beam C and 4.20 for beam E. In some cases, e.g. with cluster 4 for beam E, a single point was assigned to the cluster about a month before the model began to continuously assign data points to the cluster. In that and other similar cases, the point from which the model is continuously assigning to the new cluster has been chosen to mark the formation of the new cluster.

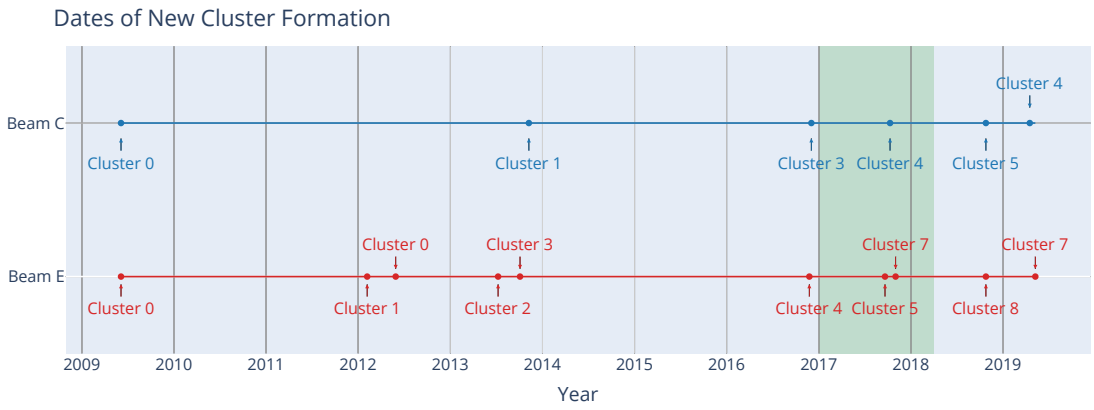

Figure 4.23: Overview and comparison of cluster formation for beam C (top) and E (bottom). Annotations mark the formation (or return) of the cluster in question. The area marked in green represents the period when vertical and horizontal displacements were detected at support I and 27.

It can be seen that several of the formations of new clusters coincide for both beam C and E. This means that they both detect changes in behavior around the same time. The first instance is when cluster 1 is formed for beam C around the same time as cluster 3 is formed for beam E. From 2017 and onward, the formation of new clusters, always seem to coincide for the two beams. The formation of cluster 3 for beam C and cluster 4 for beam E also coincides with the beginning of the period when large vertical and horizontal displacements were measured at support I, which is described in Section 1.1. During the same period, another formation took place when cluster 4 formed for beam C and cluster 7 shortly after cluster 5 formed for beam E.

After a closer examination of Fig. 4.17 and Fig. 4.21, it can be determined that the underlying behavior of the clusters that form during the period of vertical and horizontal displacements, differ greatly from the initial behaviors of beam C and E. This is most apparent in the transition from cluster 1 to 3 for beam C and from cluster 4 to 5 for beam E. The period also overlaps partly with the transitions periods in Fig. 4.16 and Fig. 4.20 respectively, where the model is alternating between two clusters, indicating a potential ongoing change. After the period ends, i.e. when the displacements cease, the new behaviors persist, which is proven by the representations for cluster 4 and 5 for beam C in Fig. 4.17 and cluster 8 for beam E in Fig. 4.21.

4.2.5 Validation of DPGMM Clustering Model

It was shown in Section 4.2.1 that a parametric clustering approach is feasible for detecting new behaviors in a beam by separating distinct patterns in strain distribution along the beam. Then, in Sections 4.2.2 and 4.2.3, the results from the non-parametric sliding window approach using a DPGMM were presented for beam C and E. To validate the non-parametric aspect of the model and the sliding window approach, the results of beam C in Section 4.2.2 are, in this section, compared to the corresponding results from the parametric GMM from Section 4.2.1.

4.2.5.1 Cluster Assignment Over Time

In Fig. 4.24 and Fig. 4.25, it can be observed that the DPGMM makes very similar predictions as the GMM. Note that, to maintain the comparability of results between the two models, the number of clusters for the parametric GMM has been predefined to match the number predicted by the non parametric DPGMM. The key difference between the two models is that the GMM has done clustering using the entire dataset at once, while the DPGMM has performed clustering incrementally using a sliding window, meaning that the sliding window has passed cannot be reassigned.

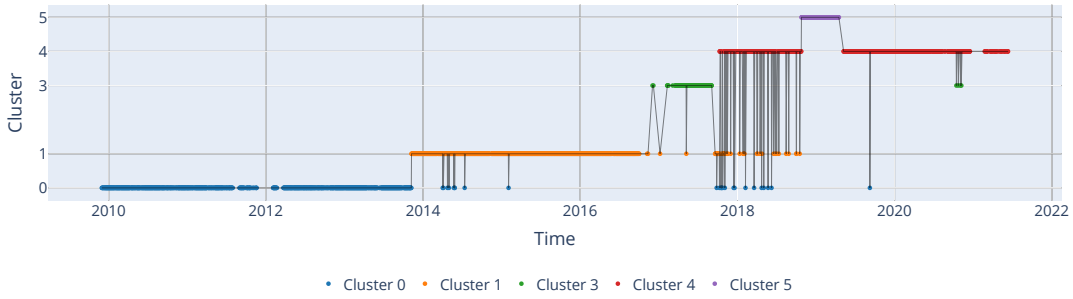

Figure 4.24: The cluster assignments over time by the DPGMM model for beam C.

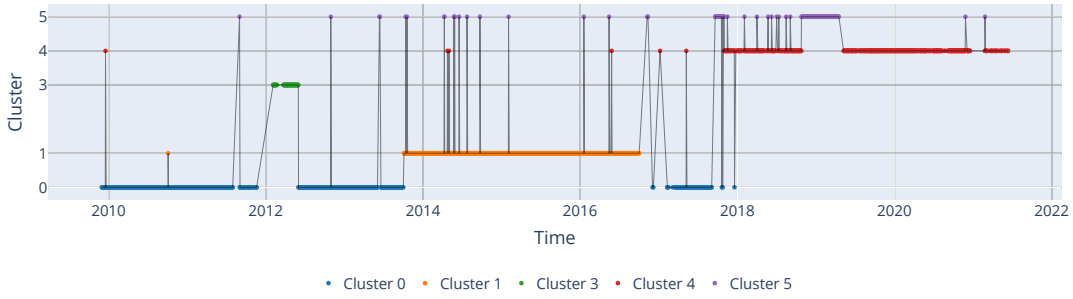

Figure 4.25: The cluster assignments over time by the GMM model for beam C.

The first main difference between Fig. 4.24 and Fig. 4.25 is that GMM assigned a set of points in early 2012 to cluster 3 while the DPGMM continues to assign the data to the same cluster during that period. The second main difference between the two

is that during the first half of 2017, the GMM returns to assigning to cluster 0, while the DPGMM is assigning to cluster 3 for the same period. The representations of clusters 0 and 3 are compared in Fig. 4.26. For the GMM, cluster 0 and 3 are visually very similar in terms of shape but differ in range. For the DPGMM however, cluster 0 and 3 show a clear difference. For the DPGMM, the the representation of cluster 3 more closely resembles the behavior that beam C is transitioning towards, i.e. with the sharp peaks at the support. It is reasonable that the GMM has differentiated between cluster 0 and 3 in 2012, however it is probably less representative that it returns to cluster 0 in 2017, despite the clear indications that the behavior is going through a significant change during that period. The DPGMM, on the other hand, is detecting the change that has started in 2017, which is more representative of the reality.

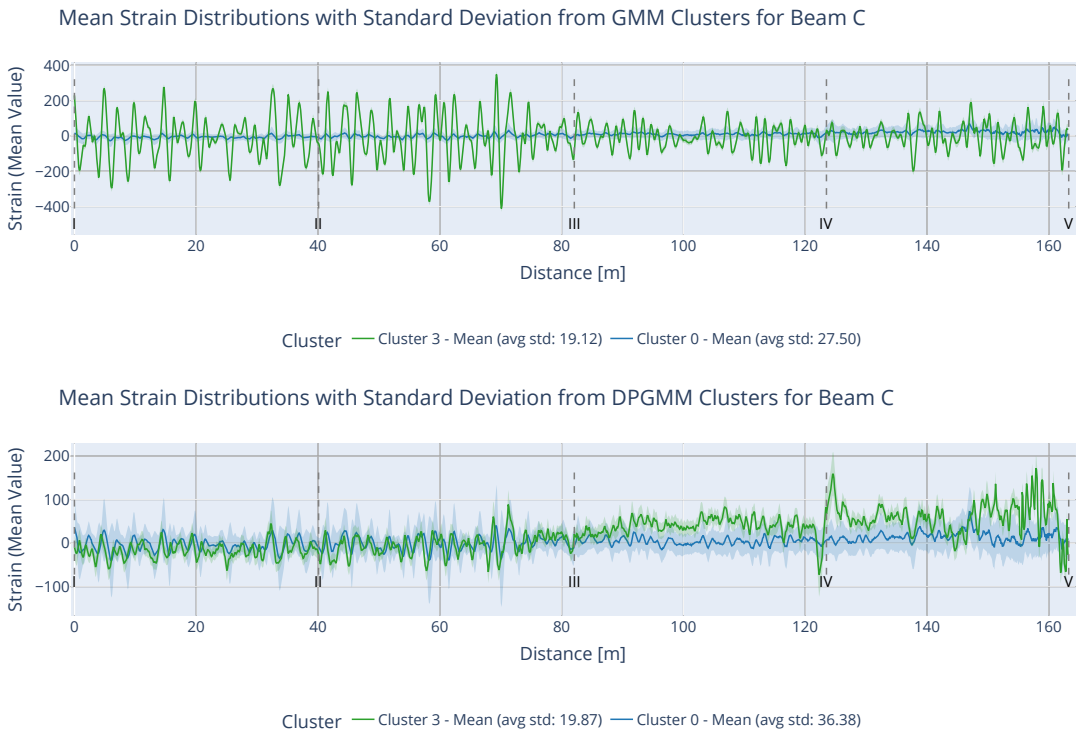

Figure 4.26: Comparison between cluster 0 and 3 for the GMM (top) and the DPGMM (below).

4.2.5.2 Stability of Cluster Assignment

Finally, it is also of interest to understand the consistency of clusters and points over time, not only the final cluster assignments over time as Fig. 4.24 and Fig. 4.25 show. This is important as the sliding window included a feature that merged clusters that were similar according a threshold using Mahalanobis distance. It is therefore of interest to understand the merging sensitivity as well as the tendency of data points to be reassigned to new clusters in the future as the sliding window moves across the data.

To do this, the data was restructured to reflect source and target destinations (clus-

ters) between time steps as well as the number of data points that drifted from one cluster to another between each given pair of timesteps. Using the restructured data, a sankey diagram was created to visualize the drift of data points between clusters. For practical reasons, the changes were accumulated into nodes with a twelve week interval to make the plot less crowded. Sankey diagrams for beams C and E can be found in Appendix C.1 and C.2. The diagrams show consistency in the cluster assignments with a limited drift over time. This indicates that the model makes stable predictions. This stability helps preserve the integrity of the clusters, enabling previously observed patterns to be retained and referenced in the future, should similar strain profiles reappear.

4.2.5.3 Conclusion on the Validation

This validation chapter has demonstrated the effectiveness and robustness of the DPGMM with a sliding window approach for clustering strain profile data. By comparing the DPGMM results with those of a traditional parametric GMM on beam C, it was shown that both models produce highly similar cluster assignments, despite some differences in how clusters are formed and updated over time.

The analysis highlighted minor differences in cluster assignments during certain periods, which can be attributed to the incremental nature of the DPGMM and the inherent similarity between some clusters. If not, the cluster assignment over time for the DPGMM made more sense than that of the GMM, as it seemed to better capture ongoing changes that were taking place.

More importantly, the DPGMM approach was shown to maintain stable and consistent cluster assignments over time, as confirmed by the limited drift observed in the sankey diagram analysis.

Overall, these findings validate the non-parametric, sliding window-based DPGMM as a reliable method for detecting and tracking evolving behaviors in structural monitoring data, offering flexibility and adaptability without sacrificing clustering performance. Moreover, its ability to retain information about previously observed clusters supports long-term monitoring by enabling the recognition of recurring structural patterns.

5

Discussion

This thesis has investigated the possibilities of using machine learning for anomaly detection in structural health monitoring (SHM) based on data from a distributed fiber optic monitoring system. An early choice was made to view the data from two different perspectives – as time series and as strain profiles. This lead to the development of two distinct approaches: a Long Short Term Memory (LSTM) neural network was set up for the time series approach while a Dirichlet process Gaussian mixture model (DPGMM) was developed to identify changes in structural behavior by separating different behaviors represented by strain profile.

As seen in the previous section, the second approach using the DPGMM to find anomalous strain profile samples proved to be more effective in doing so than the LSTM model was in finding anomalies in time series. The two approaches differ significantly, therefore they will be discussed separately in the following sections.

5.1 Discussion on Approach 1 - Long Short Term Memory Neural Network

As seen in Section 4.1, the LSTM model was in fact capable of accurately predicting the signal in the test set. It was also able to capture the long term cyclic trends that were a result of the variations in ambient temperature. However, it was suspected that the model closely captured and reconstructed anomalous patterns in the data, such as sudden and in a physical sense unnatural spikes in strain, as well as gaps in the data that the model still was able to reconstruct to a varying degree of accuracy. Since anomalous behaviors were being reconstructed as if they were normal, it resulted in a low prediction error and patterns that were visibly potentially anomalous were not flagged as such. Consequently, this make the model unreliable as more potential anomalies could be masked, as was demonstrated with the unnatural patterns.

The likely reason the model successfully reconstructed potentially anomalous patterns is that it had encountered similar patterns during training. The monitoring system was installed on the bridge in 2007, by which time the structure was nearly seventy years old and already showing significant damage. It is therefore plausible that anomalies were present throughout the dataset, including the training portion. Since LSTM models learn from normal patterns, their reconstruction performance

is best for data they have seen or that resembles the training distribution. If the training set is contaminated with anomalous data, likely in this case, the model may learn to interpret such patterns as normal. Consequently, it may reconstruct both normal and anomalous patterns with high accuracy, reducing its sensitivity to true anomalies during inference.

Although efforts were made to improve the quality of the training data—such as temporal realignment (Section 3.4.1.2), thorough outlier removal, imputation of missing values, and the addition of smoothing features like moving averages—these issues persisted. This underscores the importance of careful data selection and preprocessing in structural health monitoring, where reliable anomaly detection depends on the availability of truly representative normal data.

For future machine learning applications to be effective on low temporal resolution time series data, such as the data gathered by the fiber optic system, it will require one of two things - labels or normal data. From the standpoint of this project, the task is purely unsupervised due to the absence of labels. Labeling the data continuously during the first years of monitoring, by marking anomalous data points as labeled anomalies, could have made it possible to tune the model using those labels and thus making the model more reliable. The downside of this is that the manual labeling will be labor intensive and would require domain knowledge. The second option would be to retrospectively process the data, while the conditions during collection are still known, to correct errors that have occurred during collection in an attempt to make the data more normal. Another way of acquiring normal data would be to monitor the structure as soon as it is taken into operation, which would guarantee as baseline of healthy data to be used for model training. While this sounds like an appealing option, it would likely not make economical sense to closely monitor a structure during its first years of operation.

A key difference between the anomaly detection strategy described in this work compared to the one used during monitoring of the bridge is that they based the anomaly threshold for the alarms on absolute strain values while the model in this thesis is based on reconstruction error. What this means is that a timestamp that was flagged as anomalous by the old system due to a high value might not be flagged as an anomaly by the LSTM model given that the reconstruction error is below the threshold. Likewise, a timestamp with a strain value within the acceptable range in terms of absolute values of strain would have not been flagged as an anomaly by the old system while it might have been so by the LSTM model if the reconstruction error is high enough, which would mean that the value was not expected at that time. Consequently, the LSTM model provides more flexibility as anomalies can be detected also within the acceptable range of strain, in the form of contextual anomalies. The old system, on the other hand, only recognized anomalies if they exceeded the absolute limits. A key challenge for both the LSTM model and the previous system lies in the selection of appropriate anomaly thresholds, either in terms of absolute strain values or reconstruction error. The latter often requires detailed subject–specific knowledge to define what constitutes an anomalously high strain, while the former allows for a more probabilistic and generalizable approach to anomaly detection.

5.2 Discussion on Approach 2 - Dirichlet process Gaussian mixture model

As demonstrated in Section 4.2, the DPGMM effectively identified the emergence of new structural behaviors by detecting changes in the strain profiles. Over time, it detected the effects of displacements in the supports beneath beams C and E, dividing the observed behaviors into distinct clusters. The fact that DPGMM detected changes in the strain profiles for both beam C and E simultaneously, during a time that coincided with the period of ground displacements, strongly suggests that the model was capable of capturing the resulting changes in structural behavior provoked by the displacements.

The method is inspired by the works of Rogers et al. (2019), who used a non parametric Bayesian clustering model for damage detection by finding changes in the frequency spectrum. The main assumption for approach 2 in this thesis is that a similar method could be used to identify new behaviors in the bridge by detecting changes in the strain profile, just like Rogers et al. (2019) did but for frequency spectrum. This thesis shows that it is indeed possible to use non-parametric clustering for this purpose.

The model does not require high temporal resolution but does require high spatial resolution if it is to be effective. This utilizes the strengths of the fiber optic monitoring system, unlike the time series approach that only tracks one point along the beam over time. In this thesis, for this type of SHM system the DPGMM and its anomaly detection strategy was found to be the most suitable.

Despite the bridge being approximately 70 years old at the time of monitoring system installation, the model successfully detected and differentiated various behavioral regimes. Crucially, it does not require healthy baseline data; it only needs an initial set of data points to initialize clustering. This characteristic enhances its applicability for deployment on other existing bridges, supporting effective SHM in real-world scenarios. Since it was tested with a sliding window it further demonstrates its suitability for ongoing monitoring. However, expert supervision would be needed to adapt the model to new structures and to label clusters for eventual semi-supervised surveillance.

To capture only the permanent changes in the bridge, measurements were taken at 4 a.m., when traffic on the bridge was minimal and the bridge was assumed to be unloaded. This approach allowed the model to detect both gradual and sudden changes in the bridge's behavior by analyzing variations in the strain profiles independently of the load. However, if measurements from the entire day were used, the model would likely identify a larger number of clusters, as it would interpret the resulting structural response from caused by the varying load configurations as different behaviors. As a result, it could struggle to generalize and to distinguish between true structural changes and variations caused by alternating loading conditions. The term *anomalous*, as defined within the current DPGMM model used in

this work, is only meaningful relative to a specific load configuration. Its interpretation may lose validity if multiple load configurations are introduced. This effect would need to be further investigated if the model were to be applied to a structure that is continuously loaded and subject to varying load conditions.

One of the objectives of this thesis was to develop a method for anomaly detection on the bridge, using the settlements on supports I and 27 as reference. In both beam C & E, potential signs of settlements can be seen (see Fig. 4.18 & Fig. 4.22) beginning in early 2018. However, it is impossible to determine whether these changes in the strain profile are caused by changes in vertical or horizontal strain. This is due to the fact that the measurement only measured the raw strain indifferent of the direction.

As illustrated in Fig. 4.22, the strain at support I is significantly higher than at the other supports. Since support I connects södra älvbrodelen with the remainder of the south side of the bridge, this may indicate that södra älvbrodelen was more affected by settlement than other sections. Consequently, the area between södra älvbrodelen and the rest of the bridge could be interpreted as a critical point.

In Section 4.2.4, the cluster assignments over time for beams C and E are compared. Fig. 4.23 shows that both beams experience changes in cluster assignments at roughly the same time. This is expected since both beams are longitudinal and rest directly on supports that share a common piled foundation. As a result, external factors such as foundation settlements could trigger new behaviors in both beams simultaneously, although the specific behaviors may differ. Notably, the most significant behavioral changes in both beam, when compared to their initial states, occur during the same period. This period coincides with the ground displacements discussed in Section 1.1, suggesting a possible link between the observed changes and external ground movement. This strongly indicates that the bridge's behavior is affected by the ground displacements and more importantly that the model is capable of detecting the changes. It is important to point out that the alarm system of the distributed fiber optic system was not capable to detect these changes at the time. The ground displacements which moved the supports were being measured using another system, specifically designed for that purpose. This shows that by using distributed strain data, it is possible to detect changes in behavior that in this case likely stem from displacements of the supports.

Although the model detected changes around the same time for both beams, it separated the clusters into a larger number of clusters for beam E than it did for C. Firstly, when examining the strain profile representations of the cluster, it becomes clear that the changes between each behavior is less pronounced for beam E compared to beam C. This could suggest that the behavioral changes in beam E are more gradual or subtle, leading the model to distinguish more clusters to capture these finer variations, whereas beam C exhibits more distinct shifts that are captured using fewer, more clearly separated clusters. Another likely reason is that the differences stem from the DPGMM's sensitivity to hyperparameters, that control the likelihood of forming new clusters. For the model to work optimally, it is essential with a robust hyperparameters tuning.

5.3 Methodological Implications

Beyond the scope of this research, there are some potential methodological implications of the findings in this study. DPGMM showed promising results in identifying new behaviors in the bridge without the need of healthy data to train the model, which is a major advantage compared to the LSTM-model that relies on the assumption that the training data is normal and contains no or few anomalies. Another strength of the clustering model is the possibility of semi-supervised learning, by the incorporation of domain knowledge to the otherwise unsupervised clustering model. The implication of this is that when new clusters are formed, they can be analyzed and documented by encoding information about already observed clusters for future knowledge about old clusters if they were to appear again. Based on this, a proposed workflow for structural health monitoring using the DPGMM model in this thesis is presented in Fig. 5.1.

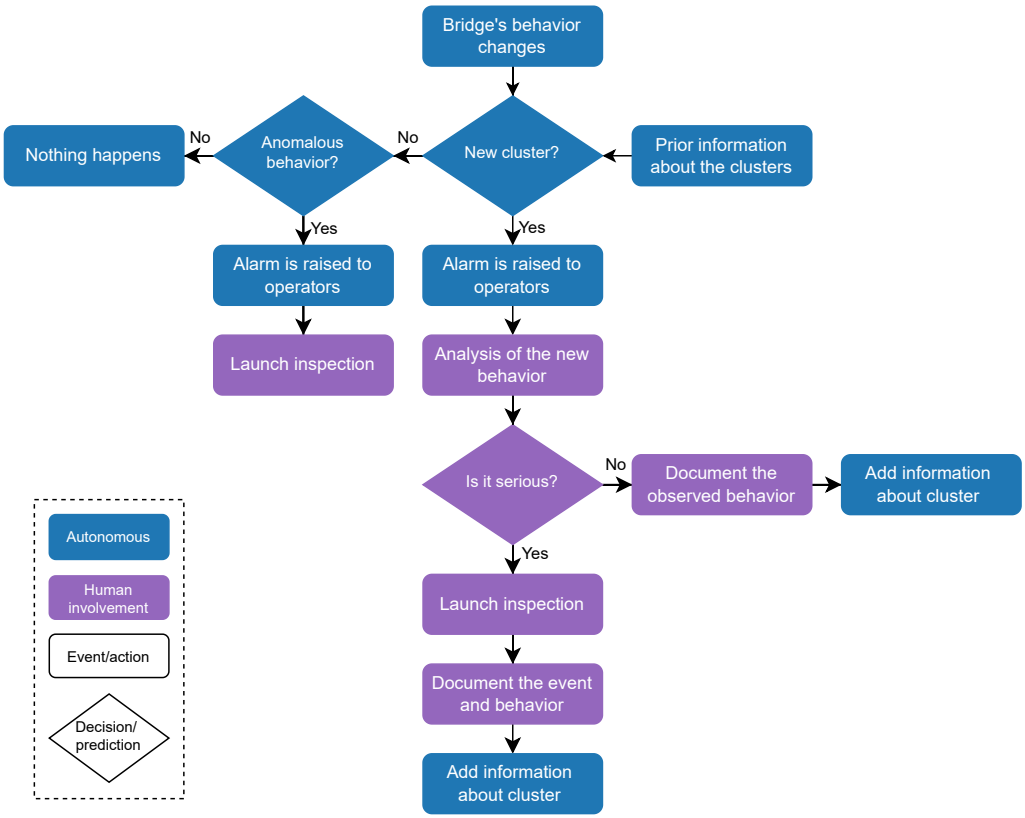

Figure 5.1: A simplified flowchart showing a proposed monitoring workflow using the DPGMM model in combination with domain knowledge and integrated decision making.

The DPGMM also allows for continuous tuning of its hyperparameters during opera-

tion. After a period of monitoring, operators may find the clustering unsatisfactory. For instance, the model might produce clusters that are too broad, leading to mean strain profiles that poorly represent the cluster. Conversely, it might split clusters too early, triggering an excessive number of alerts due to overly similar clusters being treated as separate. To address this, operators could simulate the model on already seen data using various hyperparameter combinations—similar to the approach in this thesis, where the entire dataset was available but processed with a sliding window to mimic real-time conditions. This simulation enables the identification of hyperparameters that yield a reasonable number of clusters on previously seen data. Once a suitable configuration is found, the model's parameters can be updated, allowing for more controlled future behavior without needing to wait and react retrospectively.

5.4 Ethical, Social, and Privacy Considerations in AI-Based SHM

The use of artificial intelligence (AI) and ML in SHM introduces several ethical and social considerations. Traditionally, the monitoring and maintenance of infrastructure have been the responsibility of engineers and other domain experts, with accountability for decisions resting on their professional expertise and ethical standards. Given the critical importance of infrastructure to society, it is essential that maintenance decisions are made with sound judgment and responsibility.

When AI systems are employed as decision-makers, questions regarding accountability become more complex. If an AI system fails to detect a defect or issue, resulting in an incorrect assessment of structural health, it may be unclear where responsibility lies. The fault could rest with the technology itself, the engineers or experts overseeing the AI, or the organization that deployed the system.

In addition, the use of AI in SHM raises important concerns regarding data privacy. As previously discussed, SHM systems generate and process large volumes of data that could be sensitive. Ensuring that this data is collected, stored, and used responsibly is critical to prevent misuse or unauthorized access. The integration of AI for data management introduces further challenges related to privacy and data security, necessitating robust safeguards to protect sensitive information.

6

Conclusion

The aim of this thesis was to investigate the potential applications of machine learning for anomaly detection in structural health monitoring, particularly for structures monitored using distributed fiber optic sensing techniques. Such a system was installed to monitor the Götaälv bridge in Gothenburg, and the data collected during approximately fifteen years of monitoring was used as a case study. The monitoring system was equipped with an alarm system that was limited to detecting high strain values. Additionally, this work aimed at identifying anomalies and patterns in the data that the original alarm system could overlook.

An early decision was made to move forward with two different approaches – one that regarded the data as time series and another that made use of continuous strain profiles. For the first approach, a model based on a long short term memory neural network was developed, trained and tested on the extracted time series data. For the second approach, a non parametric clustering model, a Dirichlet process Gaussian mixture model was created.

The result shows that it is possible to detect anomalies and to reduce the amounts of false positives through the use of machine learning for distributed fiber optic sensing data with low temporal resolution. The LSTM model was able to accurately reconstruct the strain data, but it was also able to reconstruct anomalies due to the the existence of anomalous behavior in the training data. The efforts to smoothen and clean the data did not solve the issue. For future applications either a labeled or healthy baseline of data is needed. Options like continuous labeling of the data or starting to monitor the structure from the start of its lifespan would possibly solve the issues met in this thesis.

The non parametric clustering model for the second approach using strain profiles proved effective in detecting changes in behavior in the monitored beams by forming new clusters whenever an unseen strain profile deviated too much from previously seen behaviors, which was made possible thanks to the non parametric nature of the DPGMM. The sliding window approach used to simulate a real-world monitoring scenario was validated against a regular parametric Gaussian mixture model, which yielded similar results. Significant changes in behavior of the monitored beams were detected by the model during a time that coincided with large soil movements, which lead to the displacements of supports. By detecting those changes that likely come as a result of the displacement of the supports, this thesis suggests that a DPGMM is a promising model to detect anomalies in structures, especially in the absence of labeled data and on existing structures. Already seen clusters have the

possibility of acting as archive of information for the future, with the addition of domain knowledge, and by doing so making the method more semi-supervised. This thesis also shows that by exploiting the spatial context of the distributed strain data, it is possible to gain insightful knowledge about the bridge, with the possibility of accurately detecting new behaviors, localizing them and potentially diagnosing the bridge by analyzing abnormal strain profiles using a finite element model.

6.1 Future Research

One promising direction for future research is the prediction of when a new cluster is about to form in order to detect changes in structural behavior before a new cluster has fully emerged. In the current model a new behavior is only identified once the model has accumulated enough data points to justify a new cluster forming. However, the possibility to anticipate whenever a new cluster is about to emerge based on the evolving profile could prove useful in proactively sending out warnings before the structural behavior has changed to become possibly anomalous. This would improve the models usefulness in SHM by enabling preemptive maintenance, reducing risk and increasing safety.

Even though the DPGMM could detect new behaviors in strain profiles, there is still considerable room for model improvement. A potential improvement is to find a way of more robustly deciding its hyperparameters for improved generalization across different sets of data, without forming too many or too few clusters than necessary for efficient interpretation.

Another interesting aspect about the DPGMM is the requirement of spatial resolution but lack of requirement of temporal resolution. Future research could try to determine how low the spatial resolution could be while still keeping the effectiveness of the model. Similarly, with the temporal resolution, a study in how frequent the measurements must be made could be conducted. By determining the requirements of resolutions for the model other types of measurement systems could be developed that could be cheaper in its implementation or be more suited to the anomaly detection model.

Bibliography

Adrian, A., Björklund, B., Kullingsjö, A., Larsson, I., Linné, C., & Tremblay, M. (2021, March). *Erfarenhetsrapport projektsamarbete och teknik vid arbeten i anslutning till Götaälvbron* (tech. rep.). Göteborgs stad.

Aiping Guo, Ajuan Jiang, & Zhangyu Cheng (Eds.). (2018). *A Hybrid Clustering Method for Bridge Structure Health Monitoring* [OCLC: 1126580353]. Newswood Limited, International Association of Engineers.

Boniol, P., Liu, Q., Huang, M., Palpanas, T., & Paparrizos, J. (2024). *Dive into Time-Series Anomaly Detection A Decade Review* (tech. rep.).

Cha, Y.-J., Ali, R., Lewis, J., & Büyükztürk, O. (2024). Deep learning-based structural health monitoring. *Automation in Construction*, *161*, 105328. https://doi.org/10.1016/j.autcon.2024.105328

Delo, G., Roy, R., Worden, K., & Surace, C. (2025). Using the inverse finite-element method to harmonise classical modal analysis with fibre-optic strain data for robust population-based structural health monitoring [_eprint: https://onlinelibrary.wiley.com/doi/pdf/10.1111/str.12481]. *Strain*, *61*(1), e12481. https://doi.org/10.1111/str.12481

Diez, A., Khoa, N. L. D., Makki Alamdari, M., Wang, Y., Chen, F., & Runcie, P. (2016). A clustering approach for structural health monitoring on bridges. *Journal of Civil Structural Health Monitoring*, *6*(3), 429–445. https://doi.org/10.1007/s13349-016-0160-0

Farrar, C. R., & Worden, K. (2013). *Structural health monitoring: A machine learning perspective* (Online-Ausg). Wiley.

Farrar, C. R., & Worden, K. (2007). An introduction to structural health monitoring. *Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences*, *365*(1851), 303–315. https://doi.org/10.1098/rsta.2006.1928

Glišić, B., & Inaudi, D. (2012). Development of method for in-service crack detection based on distributed fiber optic sensors [00023]. *Structural Health Monitoring*, *11*(2), 161–171. https://doi.org/10.1177/1475921711414233

Glišić, B., Posenato, D., & Inaudi, D. (2007). Integrity monitoring of old steel bridge using fiber optic distributed sensors based on Brillouin scattering [00024]. *3rd International Conference on Structural Health Monitoring of Intelligent Infrastructure*. https://doi.org/10.1117/12.716055

Inaudi, D., Casanova, N., Glišić, B., Vurpillot, S., Kronenberg, P., & LLoret, S. (2001). Lessons learned in the use of fiber optic sensor for civil structural monitoring [00019]. *International Journal for Restoration of Buildings*

and Monuments, *7*(3-4), 301–320. Retrieved November 16, 2015, from http://www.roctest.com/fr/content/download/772/5957/file/P21.PDF

Inaudi, D., & del Grosso, A. (2008). Fiber optic sensors for structural control [00008]. *Proc. 14th World Conf. Earthquake Engineering*. Retrieved December 30, 2016, from http://www.telemac.fr/en/content/download/677/5007/file/c196.pdf

Jin, B., Chen, Y., Li, D., Poolla, K., & Sangiovanni-Vincentelli, A. (2019). A One-Class Support Vector Machine Calibration Method for Time Series Change Point Detection. *2019 IEEE International Conference on Prognostics and Health Management (ICPHM)*, 1–5. https://doi.org/10.1109/ICPHM.2019.8819385

Langone, R., Reynders, E., Mehrkanoon, S., & Suykens, J. A. (2017). Automated structural health monitoring based on adaptive kernel spectral clustering. *Mechanical Systems and Signal Processing*, *90*, 64–78. https://doi.org/10.1016/j.ymssp.2016.12.002

Lopez-Higuera, J. M., Rodriguez Cobo, L., Quintela Incera, A., & Cobo, A. (2011). Fiber Optic Sensors in Structural Health Monitoring. *Journal of Lightwave Technology*, *29*(4), 587–608. https://doi.org/10.1109/JLT.2011.2106479

Malekloo, A., Ozer, E., AlHamaydeh, M., & Girolami, M. (2022). Machine learning and structural health monitoring overview with emerging technology and high-dimensional data source highlights. *Structural Health Monitoring*, *21*(4), 1906–1955. https://doi.org/10.1177/14759217211036880

Rogers, T., Worden, K., Fuentes, R., Dervilis, N., Tygesen, U., & Cross, E. (2019). A Bayesian non-parametric clustering approach for semi-supervised Structural Health Monitoring. *Mechanical Systems and Signal Processing*, *119*, 100–119. https://doi.org/10.1016/j.ymssp.2018.09.013

Saha, S., Sarkar, J., Dhavala, S. S., Mota, P., & Sarkar, S. (2024). Quantile-Long Short Term Memory: A Robust, Time Series Anomaly Detection Method. *IEEE Transactions on Artificial Intelligence*, *5*(8), 3939–3950. https://doi.org/10.1109/TAI.2024.3353163

Stroetmann, R. M., & Sieber, L. (2018). Toughness of old mild steels. *Proceedings 12th international conference on Advances in Steel-Concrete Composite Structures - ASCCS 2018*. https://doi.org/10.4995/ASCCS2018.2018.7029

Xu, D., Xu, X., Forde, M. C., & Caballero, A. (2023). Concrete and steel bridge Structural Health Monitoring—Insight into choices for machine learning applications. *Construction and Building Materials*, *402*, 132596. https://doi.org/10.1016/j.conbuildmat.2023.132596

Zhang, J., Zhang, J., & Wu, Z. (2022). Long-Short Term Memory Network-Based Monitoring Data Anomaly Detection of a Long-Span Suspension Bridge. *Sensors*, *22*(16), 6045. https://doi.org/10.3390/s22166045

Appendix A

Appendix A. Results from LSTM model

A.1 Beam B Model Reconstructed Signals and Anomalies

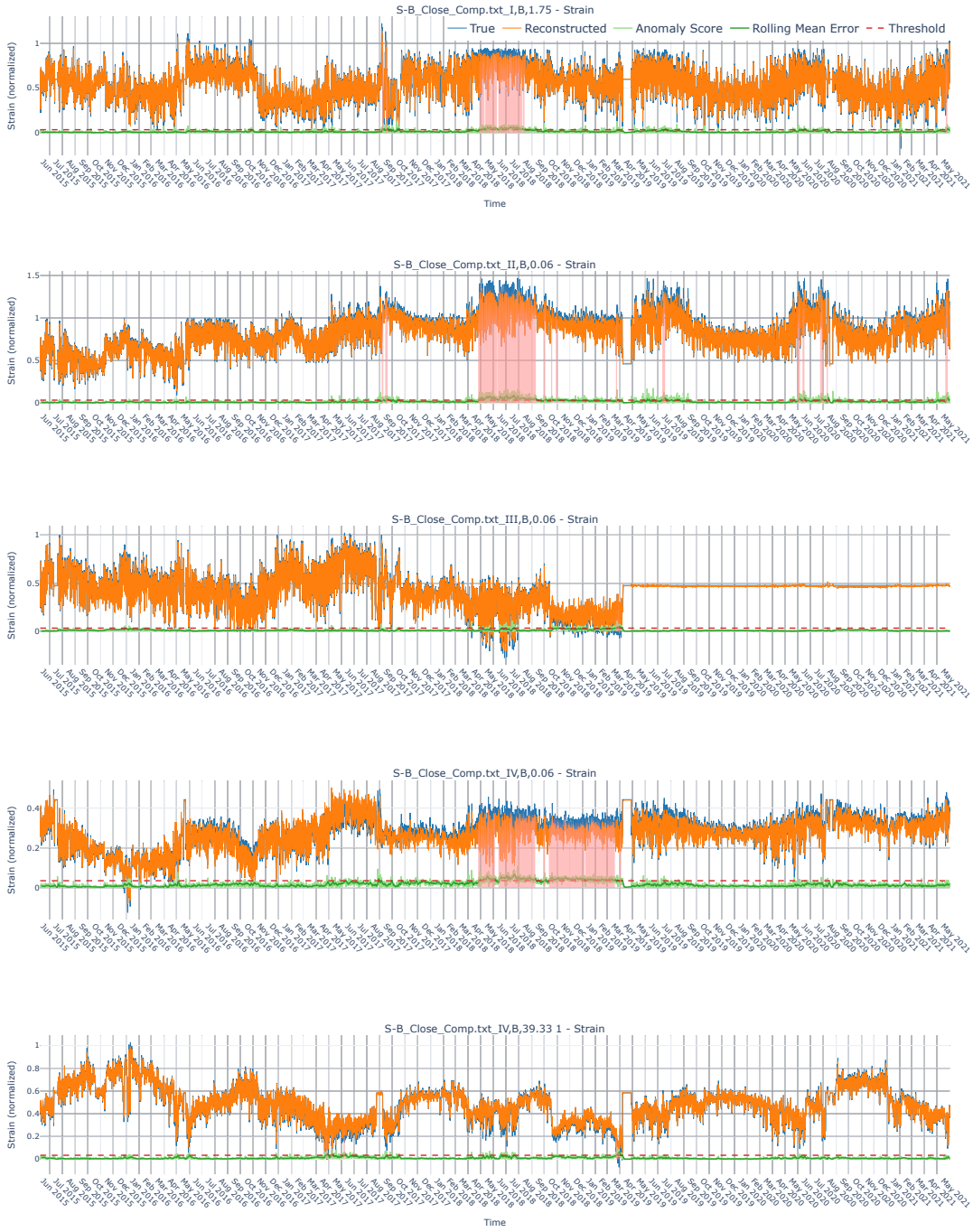

Figure A.1: True vs. predicted signal for the test data for S-B_Close_Comp at support I-V.

A.2 Beam C Model Reconstructed Signals and Anomalies

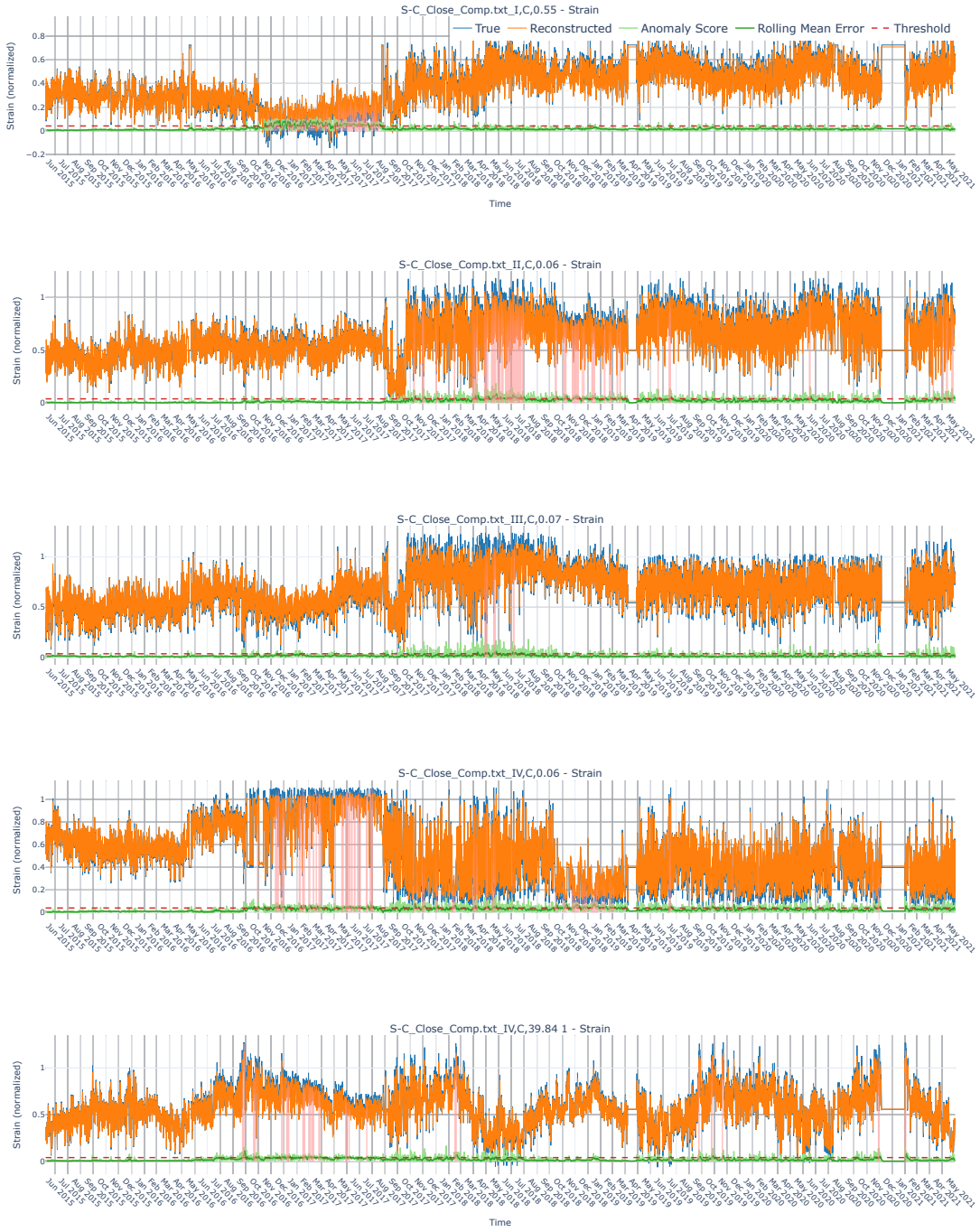

Figure A.2: True vs. predicted signal for the test data for S-C_Close_Comp at support I-V.

A.3 Beam D Model Reconstructed Signals and Anomalies

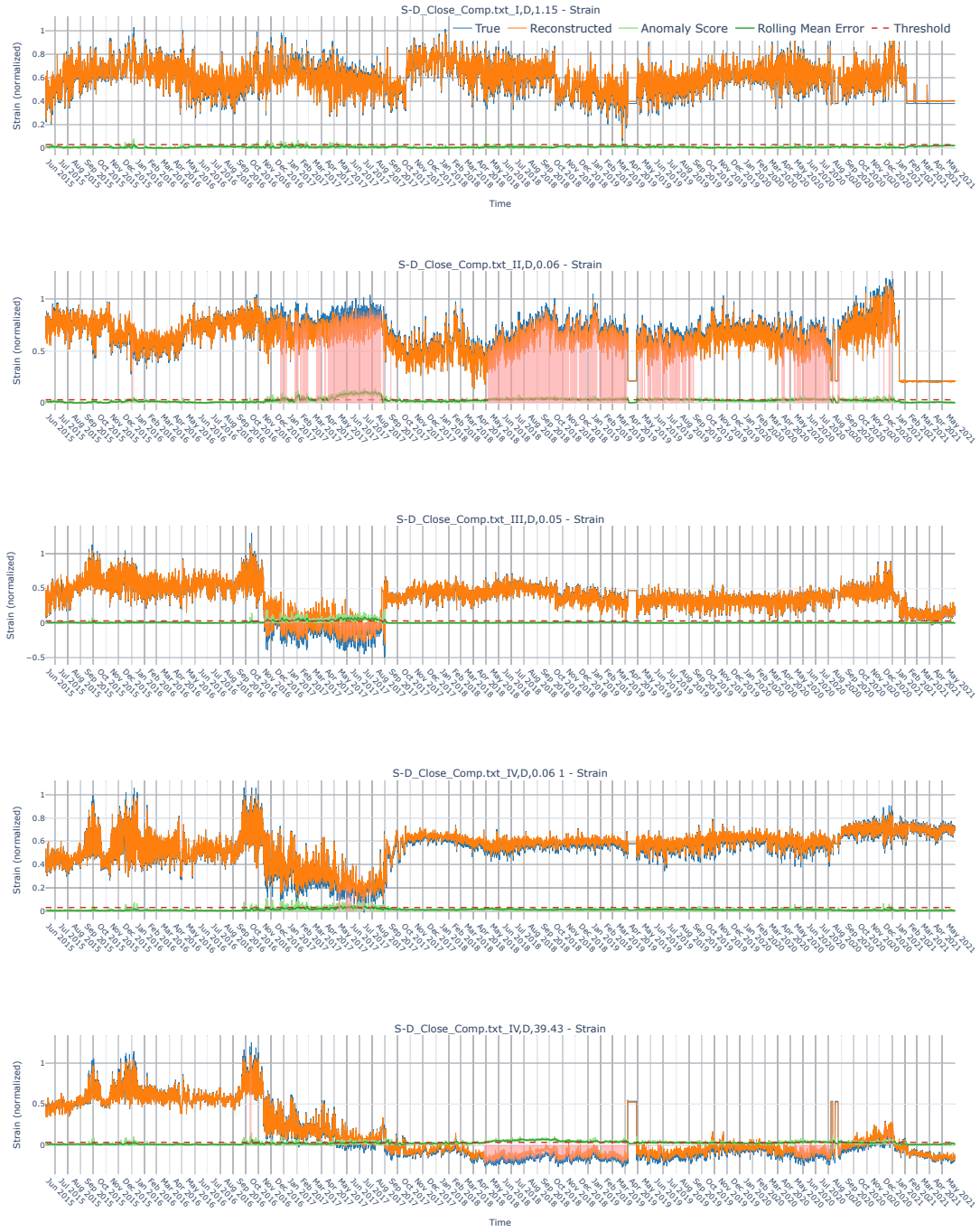

Figure A.3: True vs. predicted signal for the test data for S-D_Close_Comp at support I-V.

A.4 Beam E Model Reconstructed Signals and Anomalies

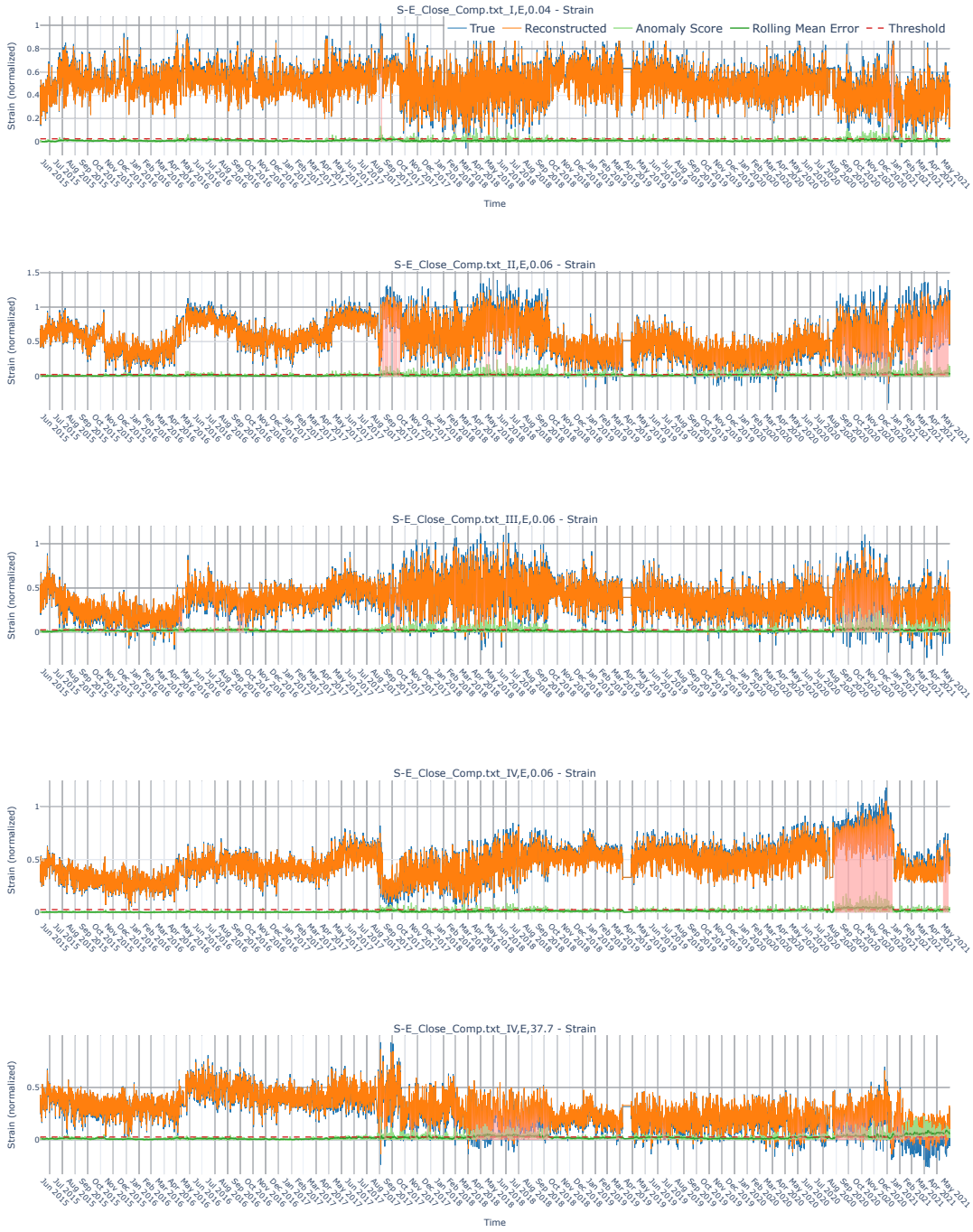

Figure A.4: True vs. predicted signal for the test data for S-E_Close_Comp at support I-V.

A.5 Beam F Model Reconstructed Signals and Anomalies

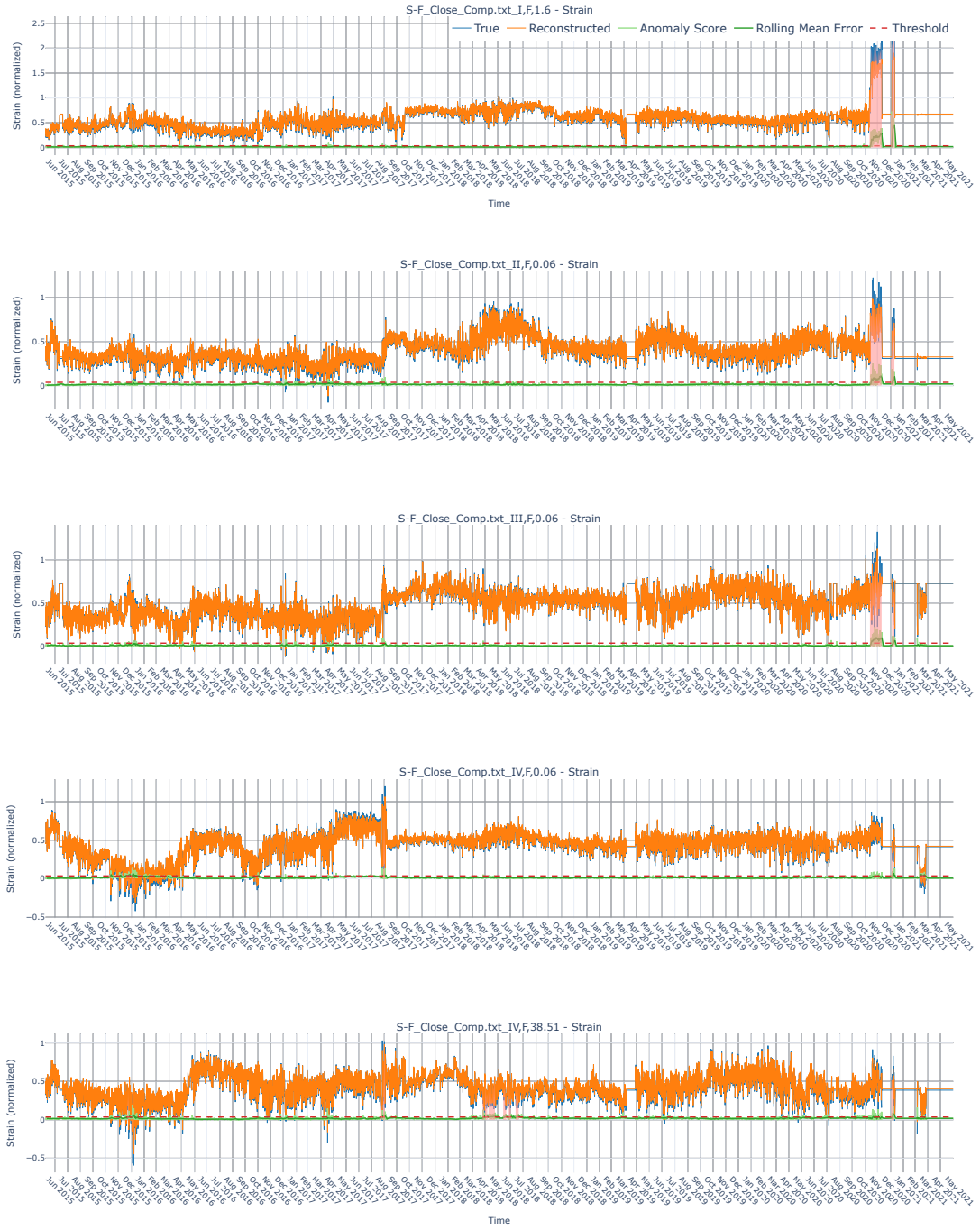

Figure A.5: True vs. predicted signal for the test data for S-F_Close_Comp at support I-V.

A.6 Support I Model Reconstructed Signals and Anomalies

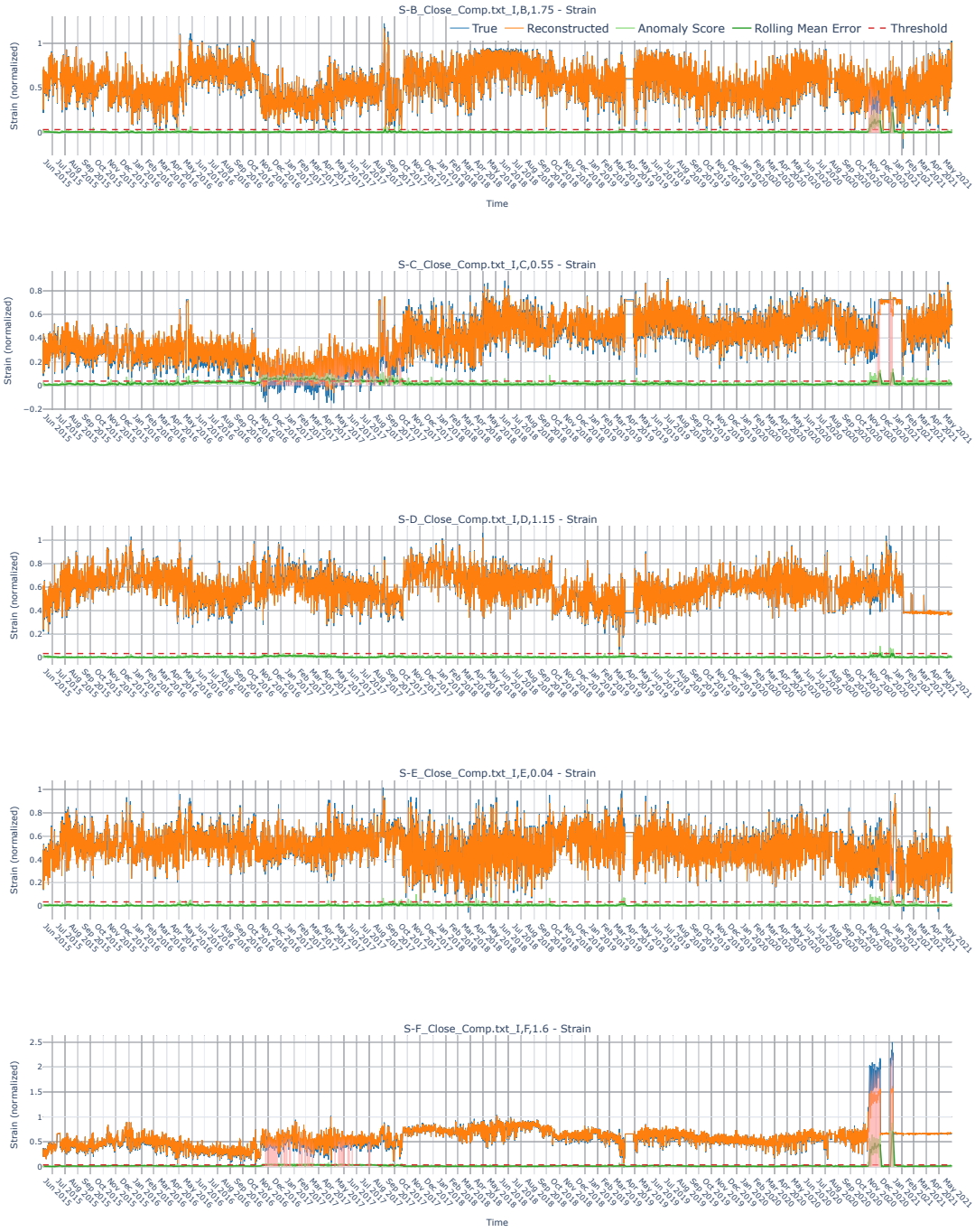

Figure A.6: True vs. predicted signal for the test data for beam B-F at support I.

A.7 Support II Model Reconstructed Signals and Anomalies

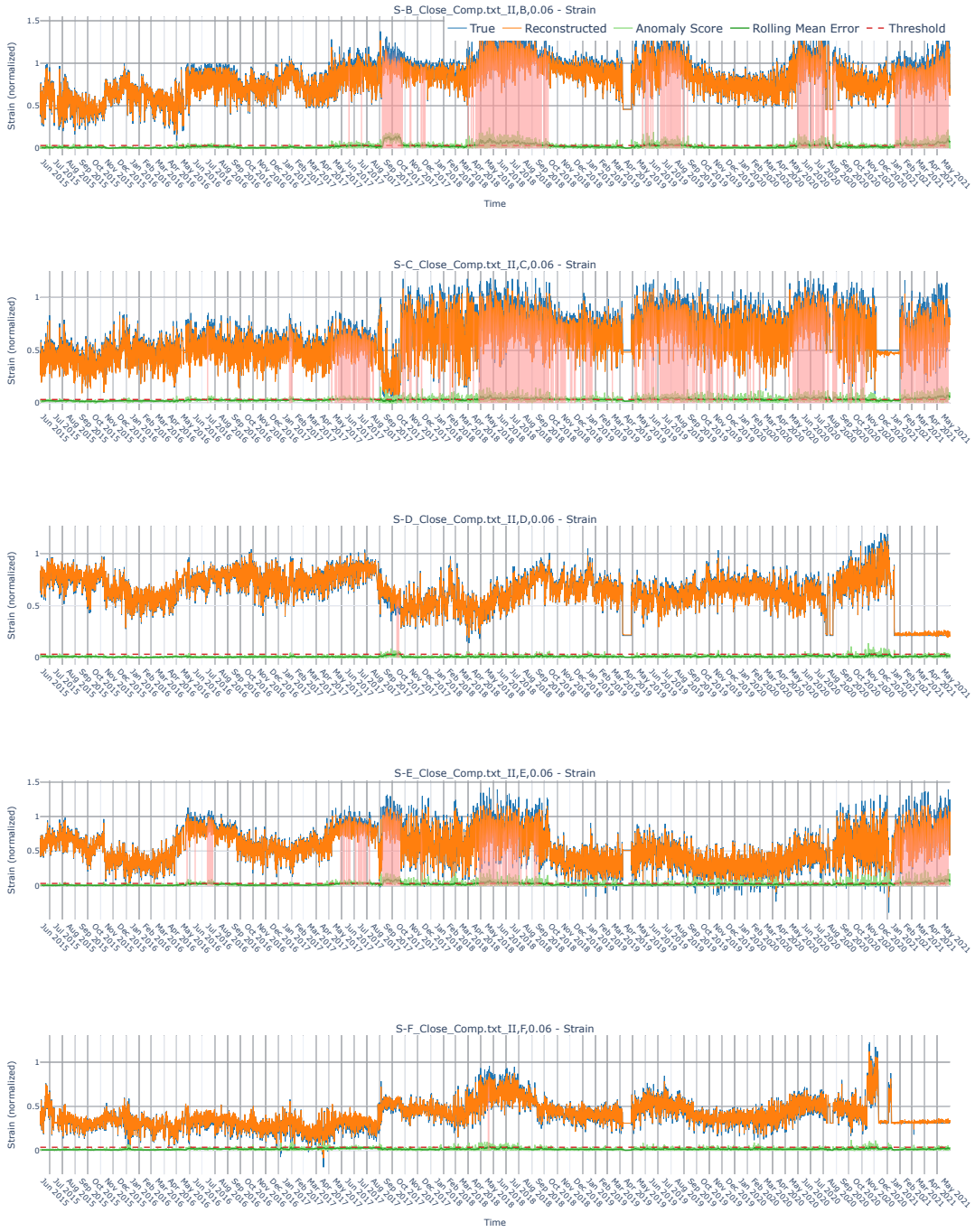

Figure A.7: True vs. predicted signal for the test data for beam B-F at support II.

A.8 Support III Model Reconstructed Signals and Anomalies

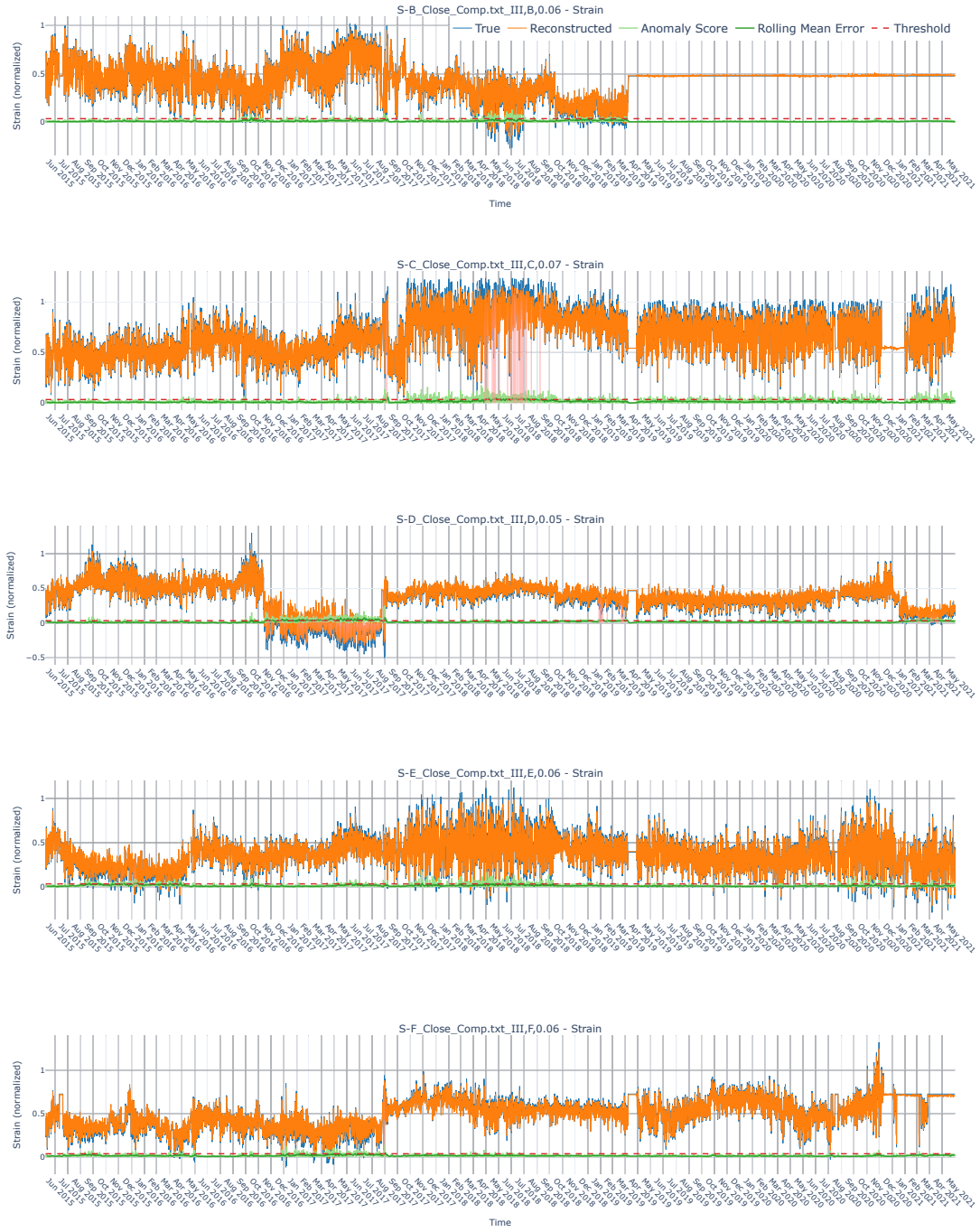

Figure A.8: True vs. predicted signal for the test data for beam B-F at support III.

A.9 Support IV Model Reconstructed Signals and Anomalies

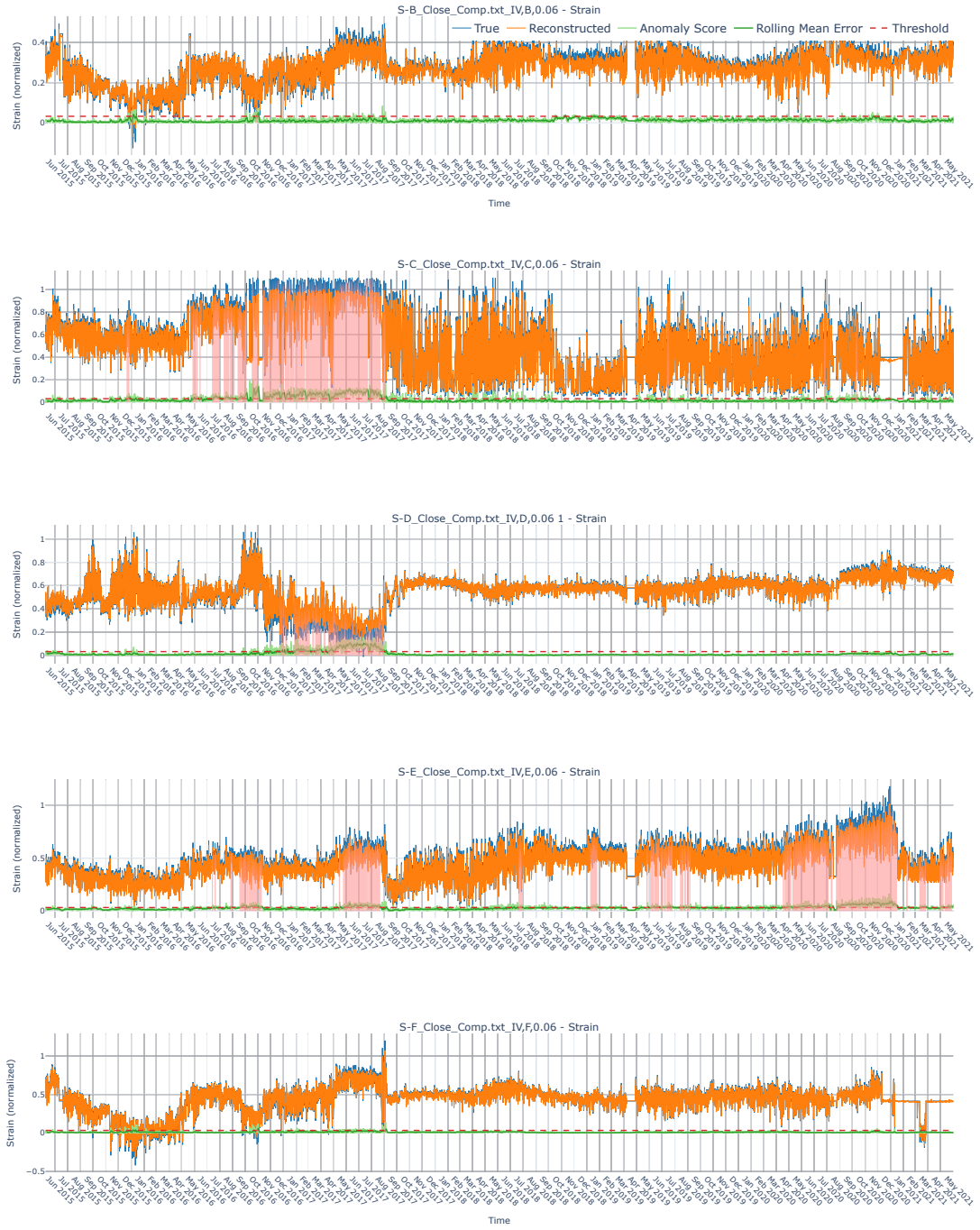

Figure A.9: True vs. predicted signal for the test data for beam B-F at support IV.

A.10 Support V Model Reconstructed Signals and Anomalies

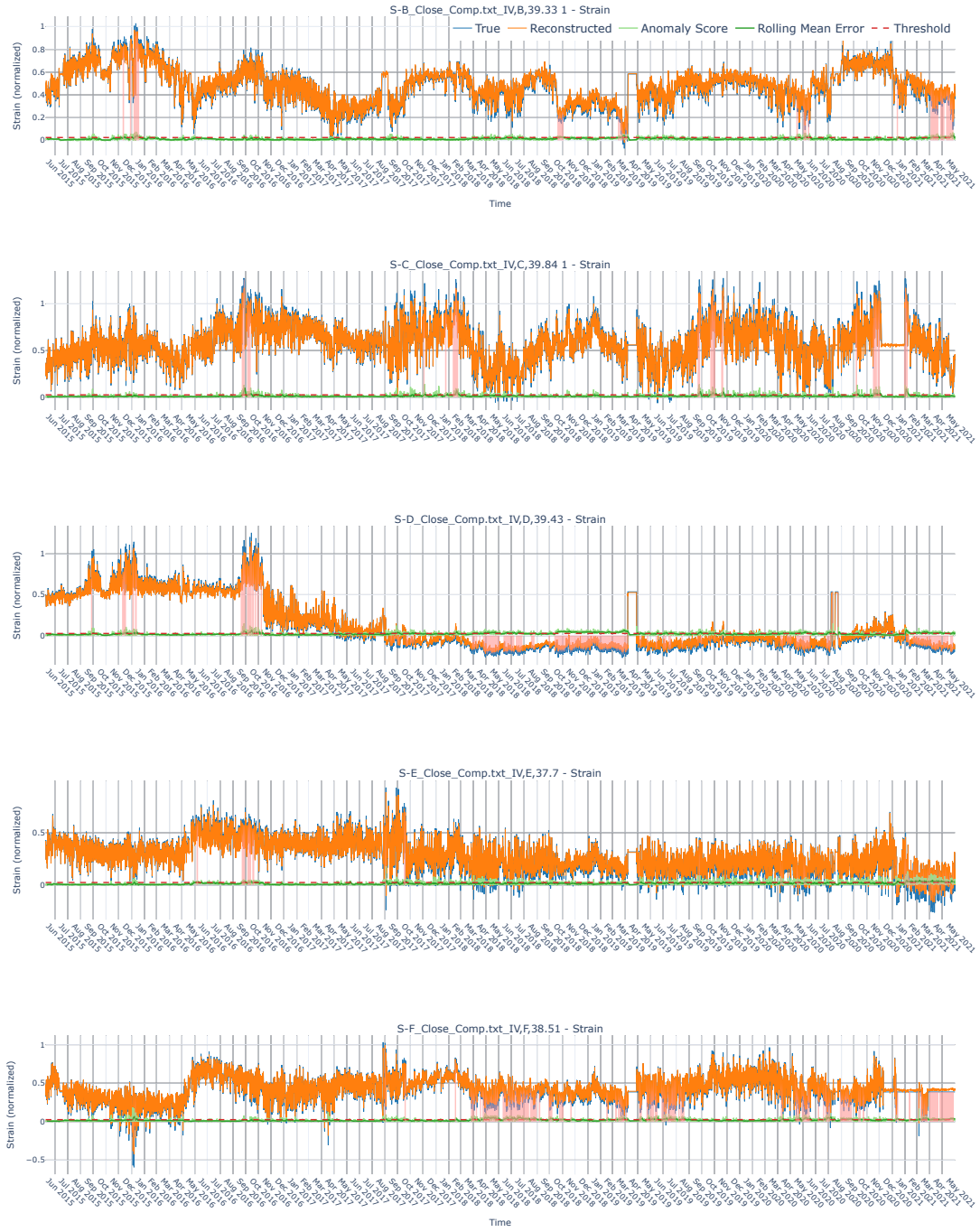

Figure A.10: True vs. predicted signal for the test data for beam B-F at support V.

A.11 All Channels Model Reconstructed Signals and Anomalies

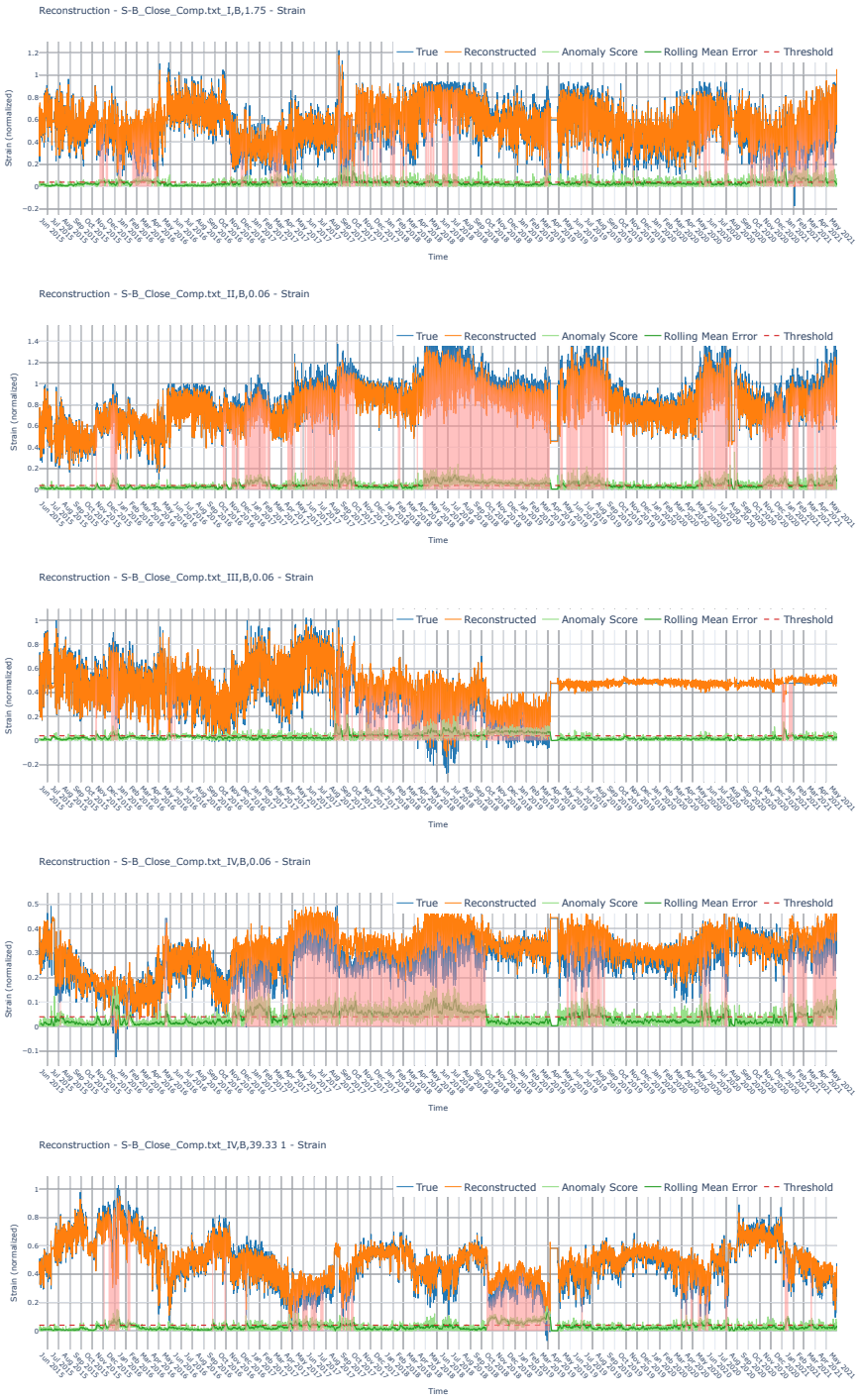

Figure A.11: Reconstructed signals and anomalies for S-B Channels I–IV

Figure A.12: Reconstructed signals and anomalies for S-C Channels I–IV

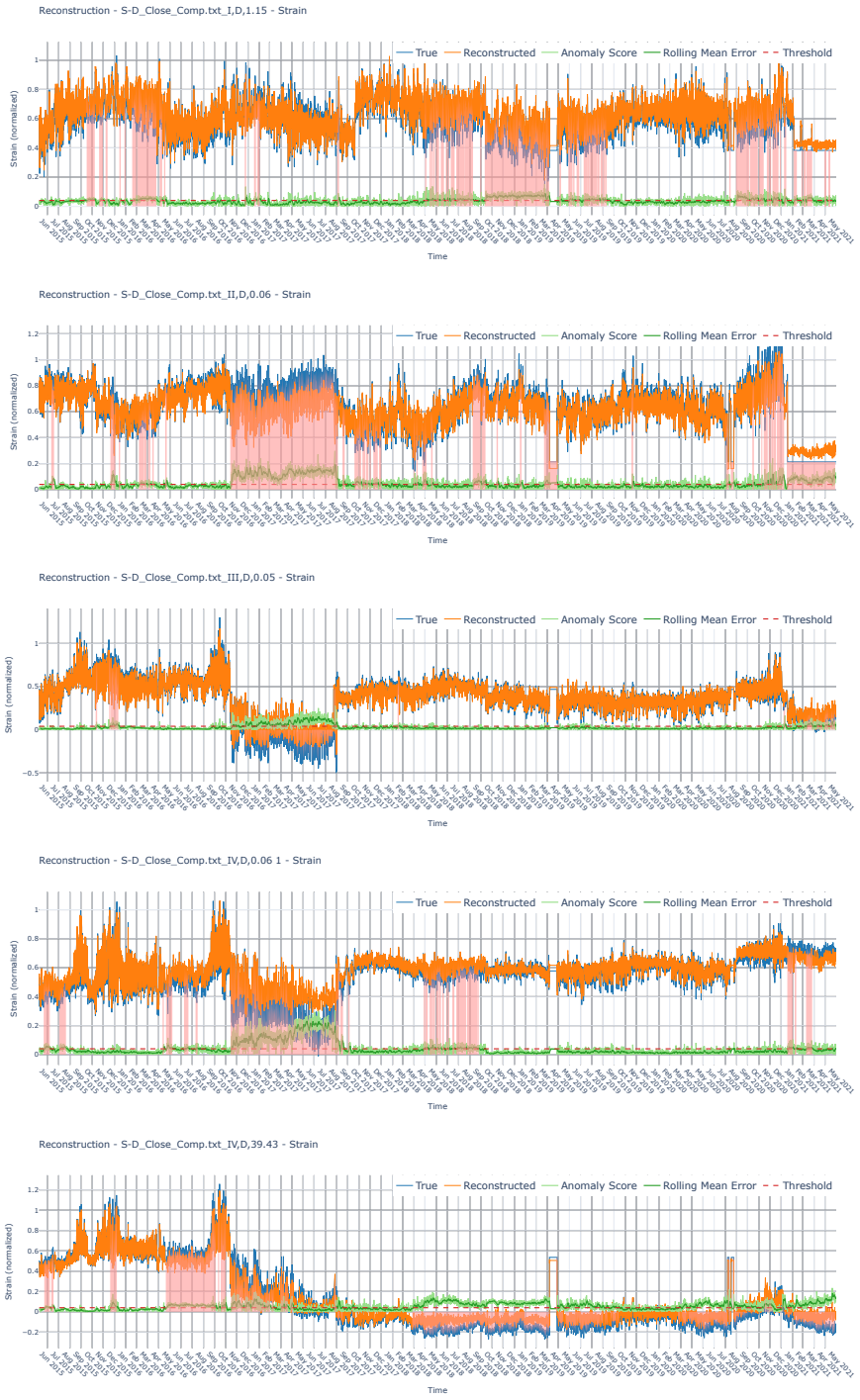

Figure A.13: Reconstructed signals and anomalies for S-D Channels I–IV

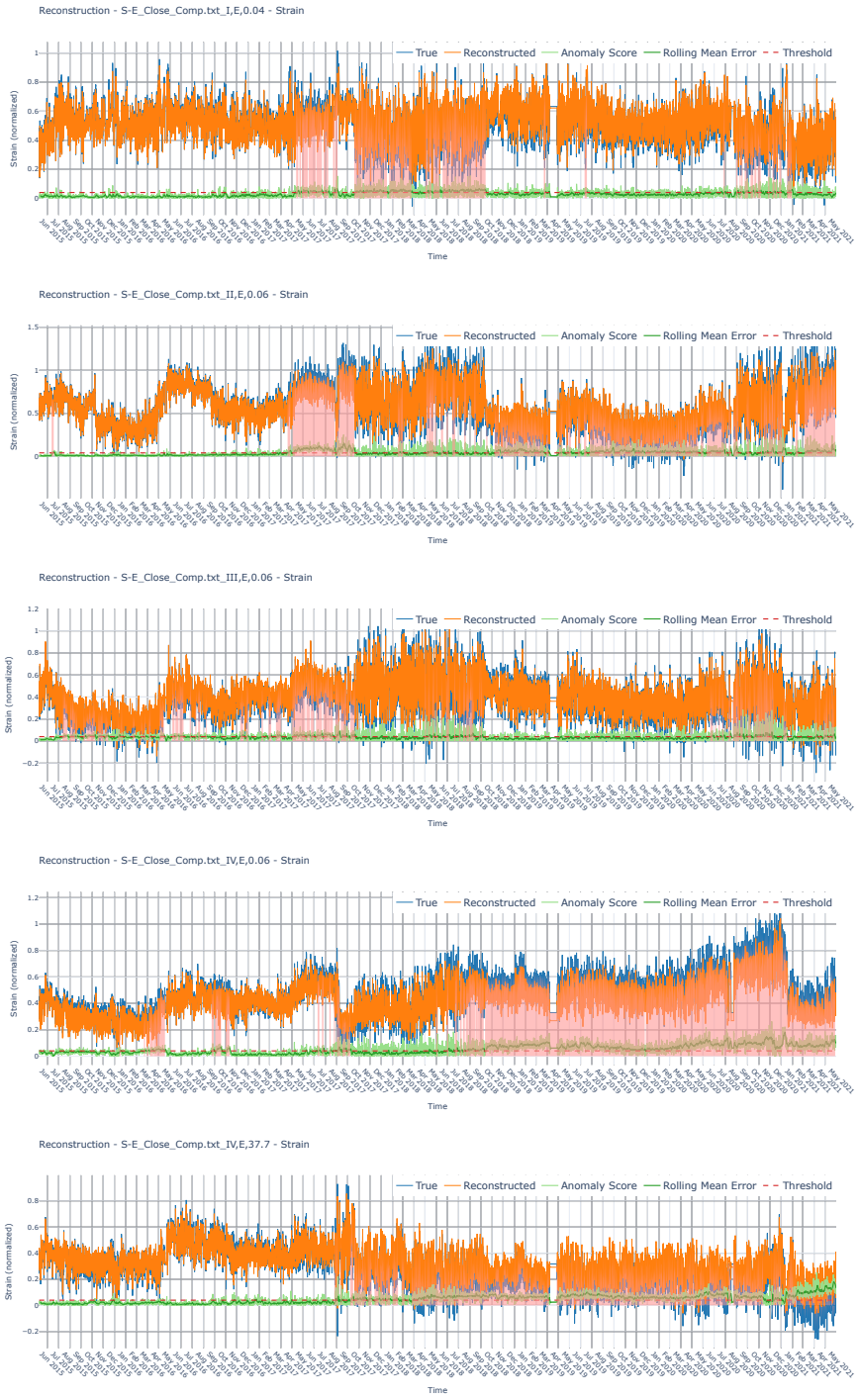

Figure A.14: Reconstructed signals and anomalies for S-E Channels I–IV

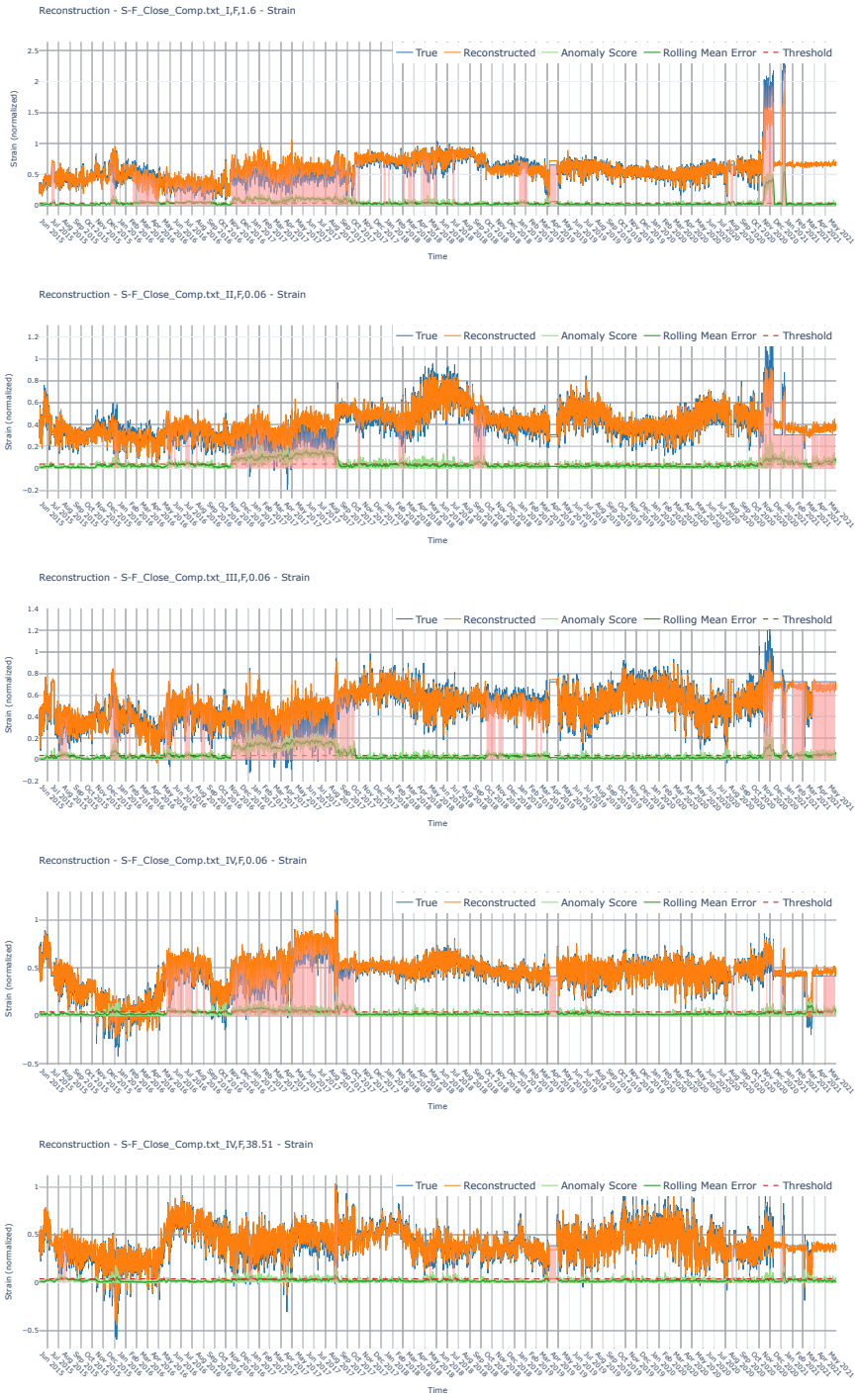

Figure A.15: Reconstructed signals and anomalies for S-F Channels I–IV

Appendix B

Appendix B. Results from DPGMM

B.1 DPGMM Results Beam B

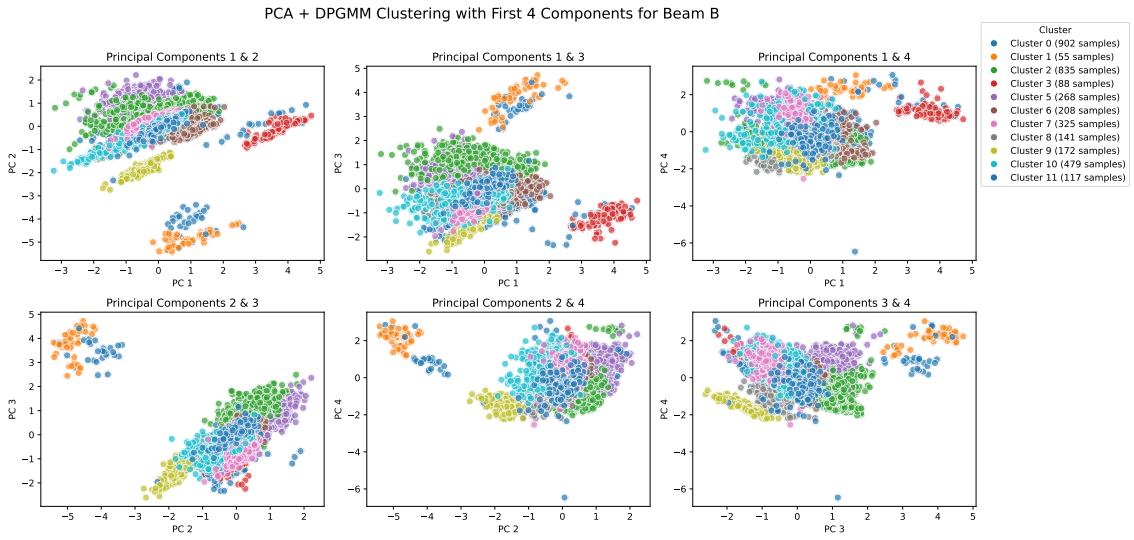

Figure B.1: Clusters prediction by the DPGMM plotted as combination of the first four principal components.

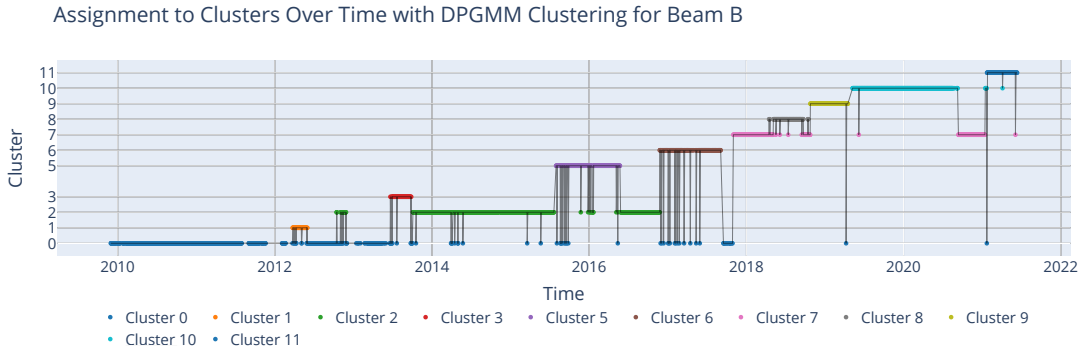

Figure B.2: Cluster assignment over time on beam B.

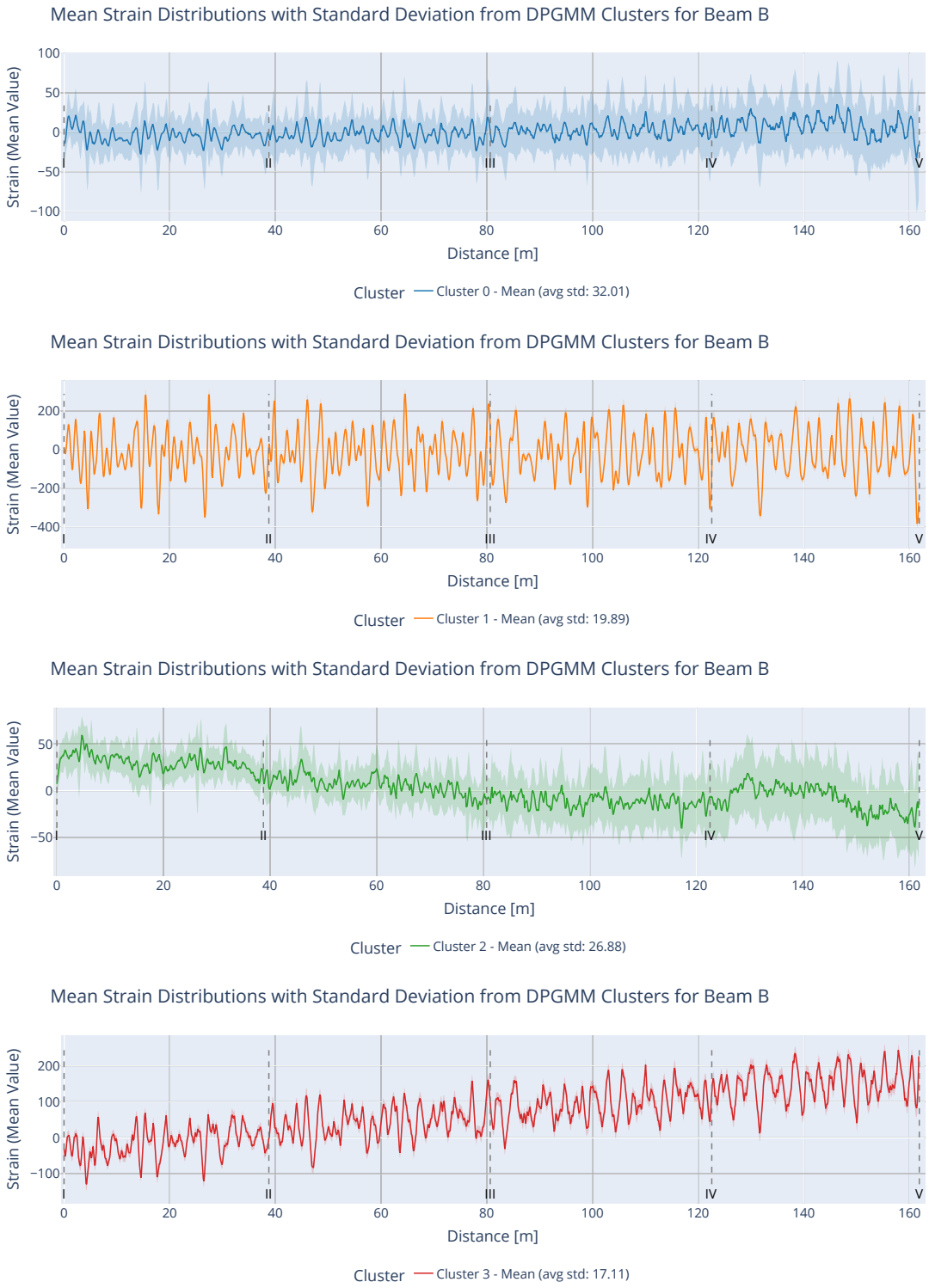

Figure B.3: Mean strain profiles (1/3) from DPGMM clustering for beam B.

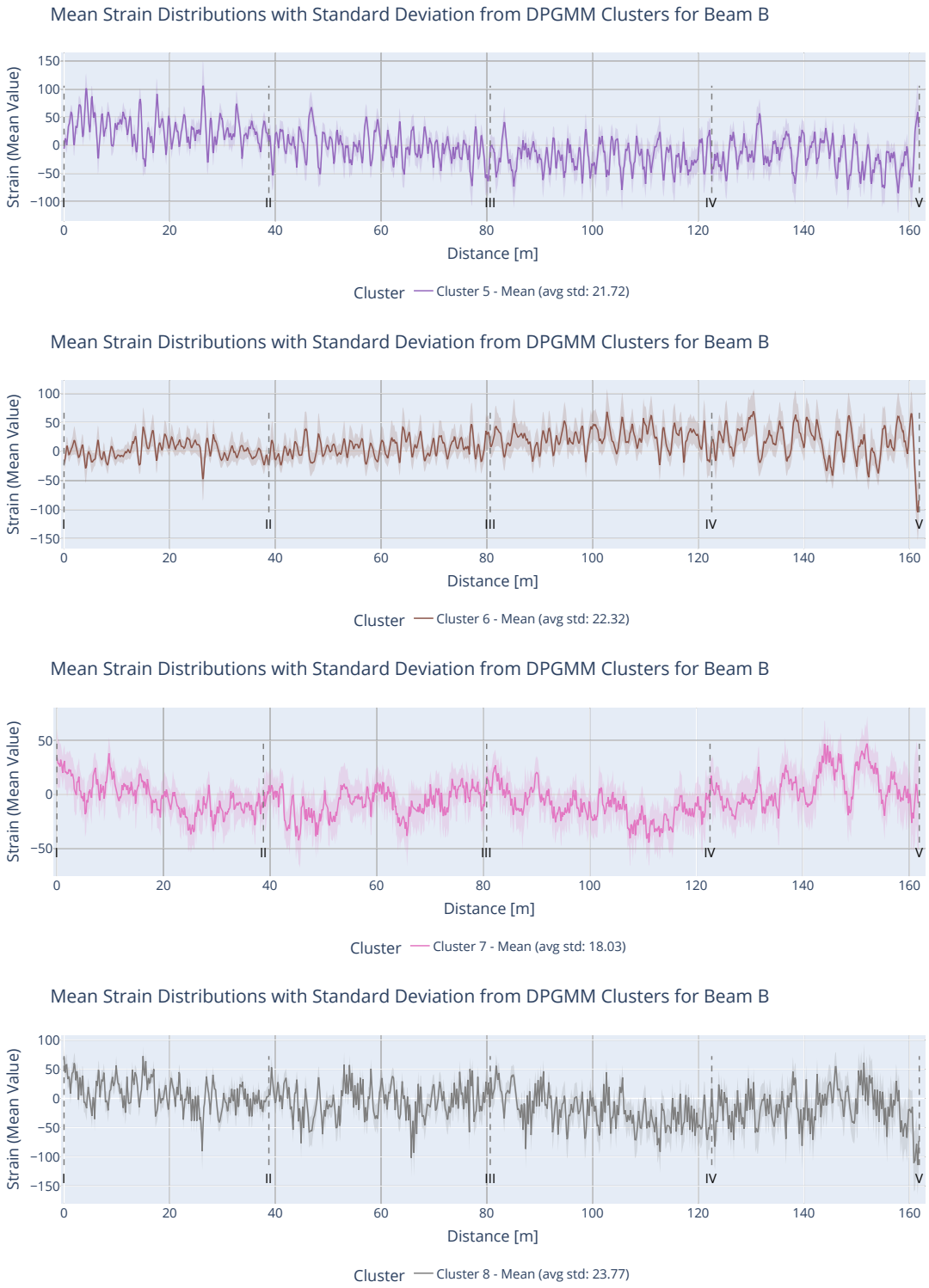

Figure B.3: Mean strain profiles (2/3) from DPGMM clustering for beam B.

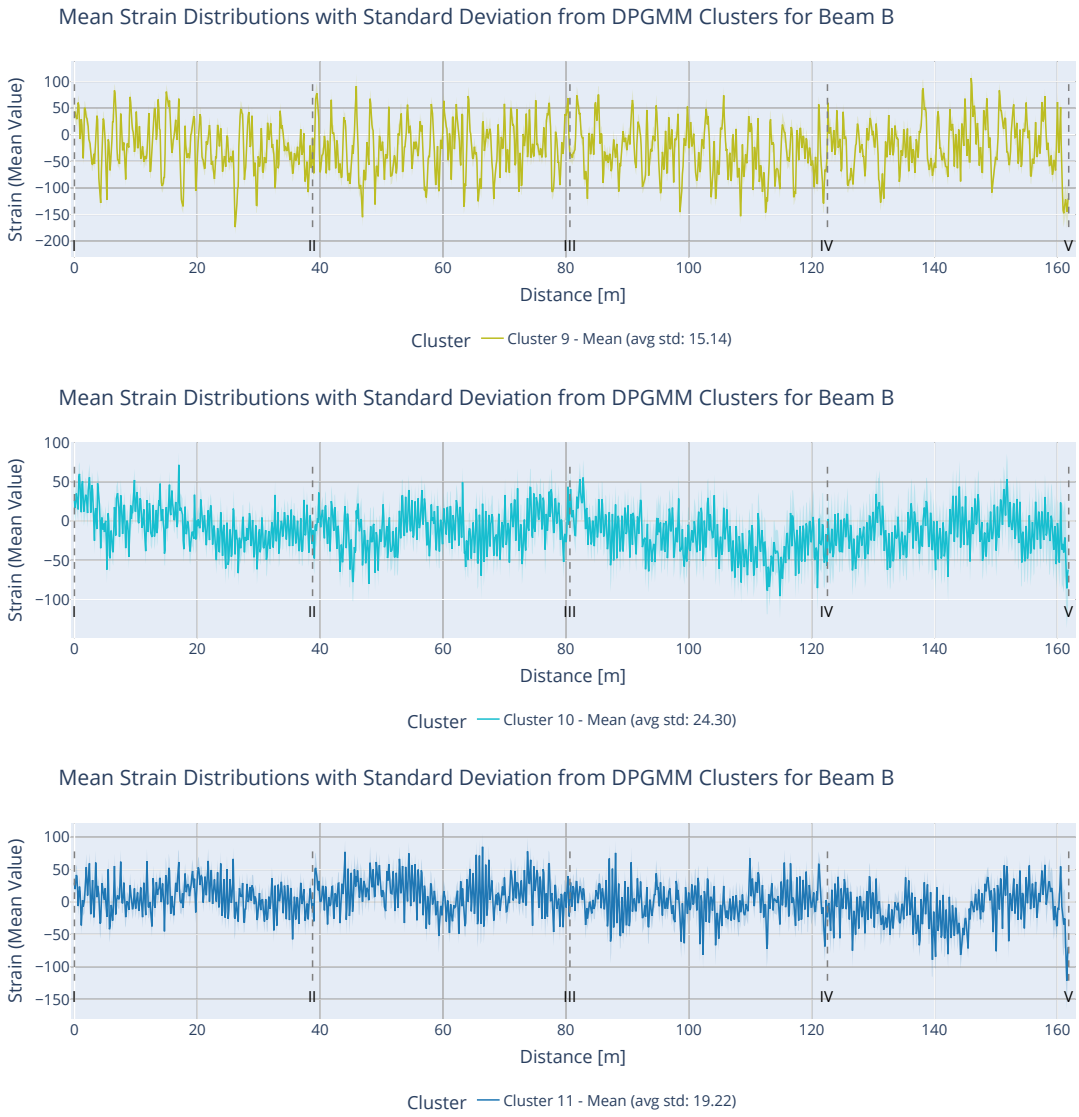

Figure B.3: Mean strain profiles (3/3) from DPGMM clustering for beam B.

B.2 DPGMM Results Beam D

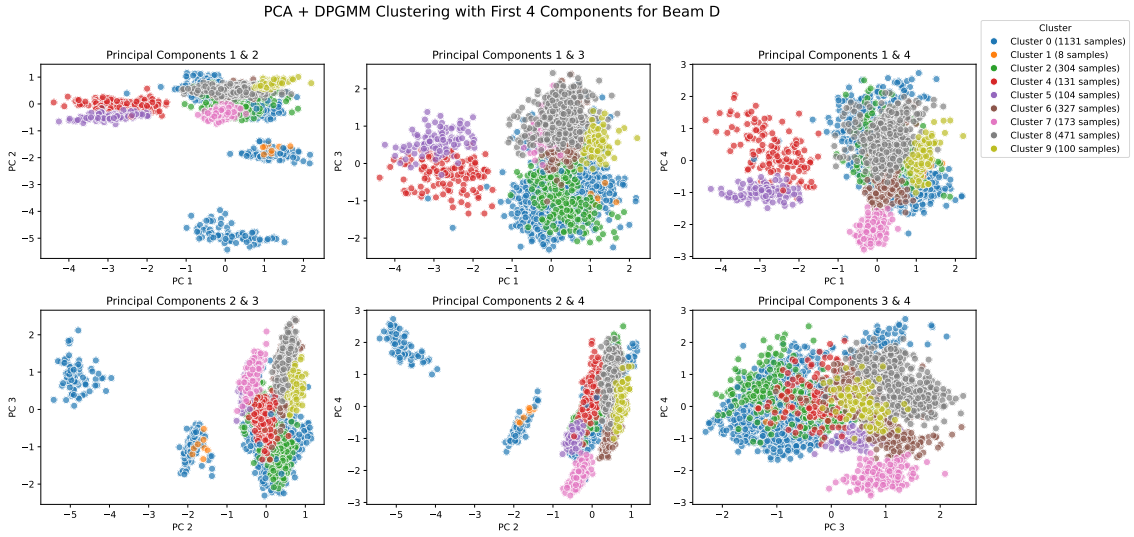

Figure B.4: Clusters prediction by the DPGMM plotted as combination of the first four principal components.

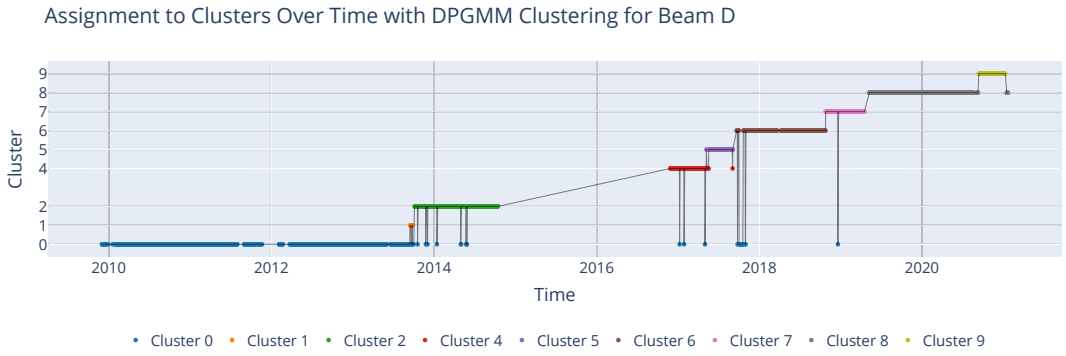

Figure B.5: Cluster assignment over time on beam F.

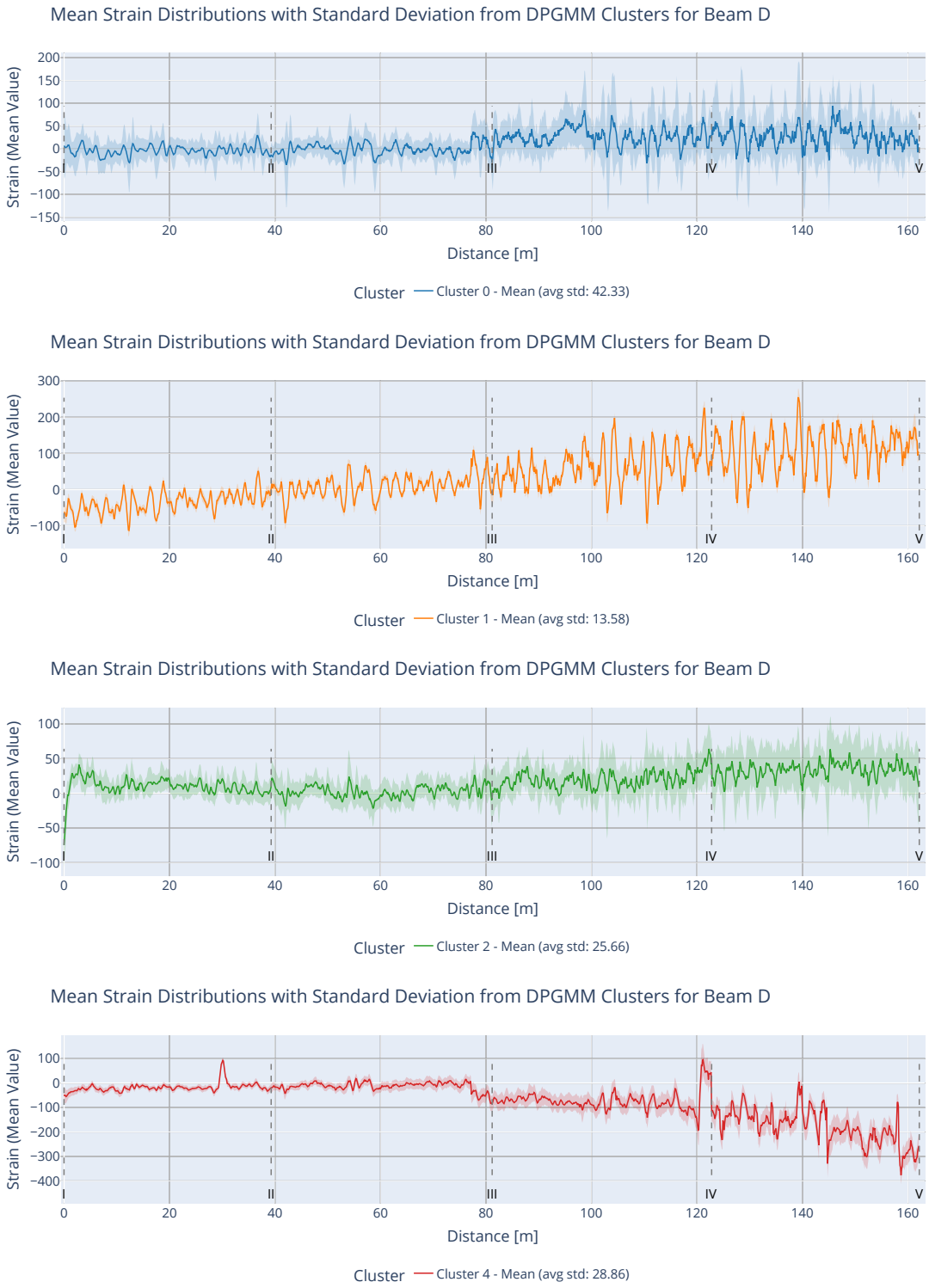

Figure B.6: Mean strain profiles (1/3) from DPGMM clustering for beam D.

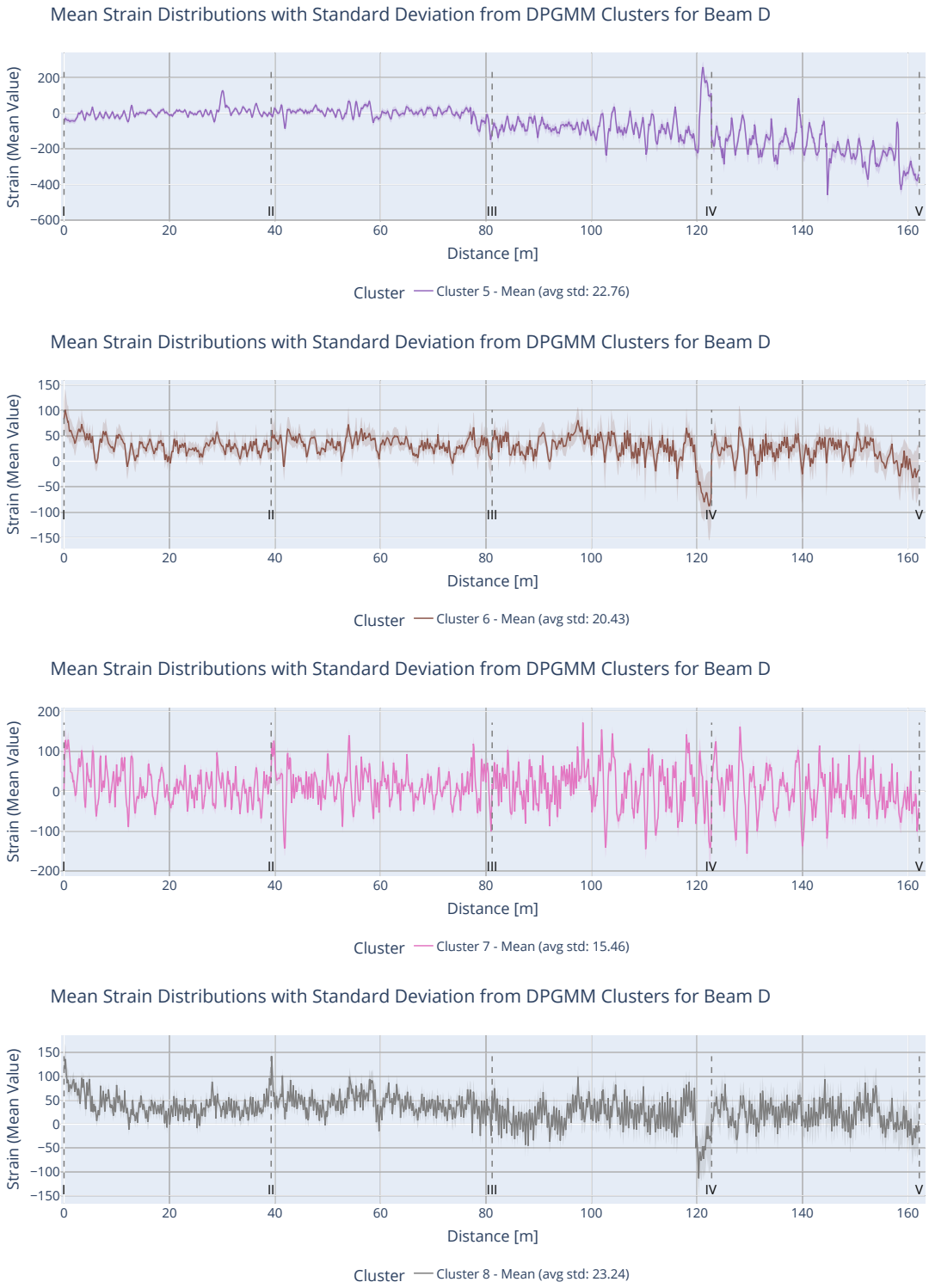

Figure B.6: Mean strain profiles (2/3) from DPGMM clustering for beam D.

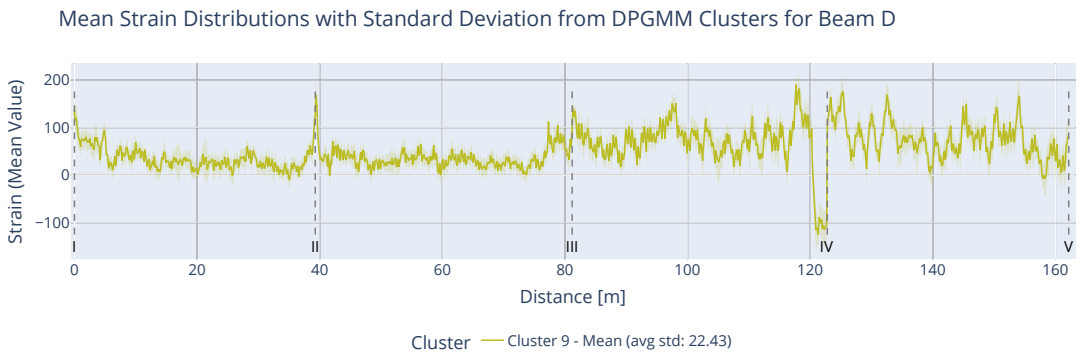

Figure B.6: Mean strain profiles (3/3) from DPGMM clustering for beam D.

B.3 DPGMM Results Beam F

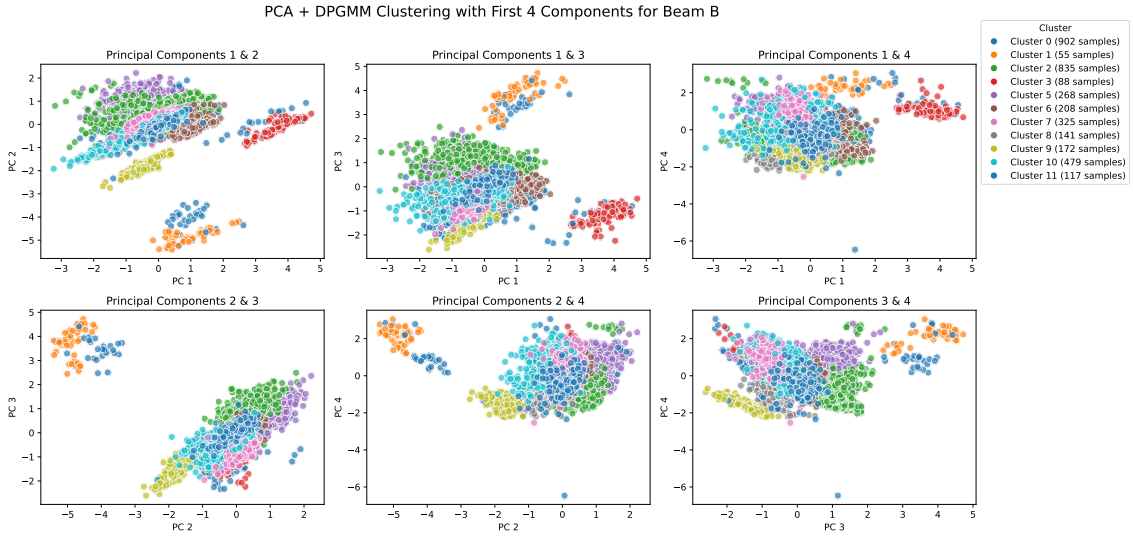

Figure B.7: Clusters prediction by the DPGMM plotted as combination of the first four principal components.

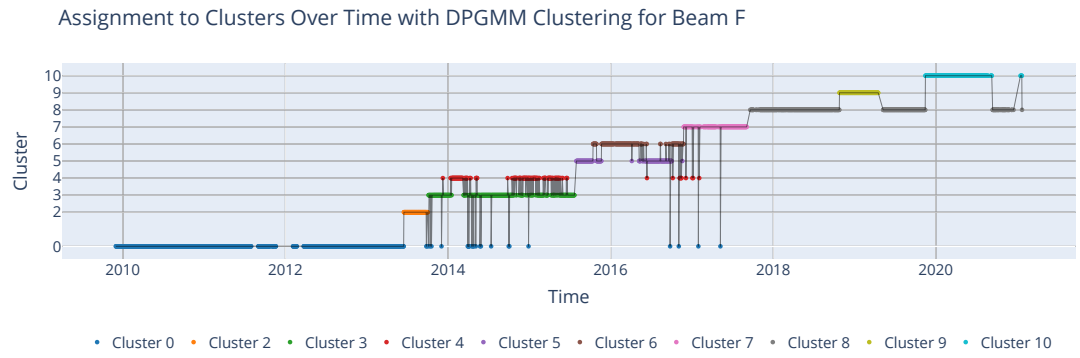

Figure B.8: Cluster assignment over time on beam F.

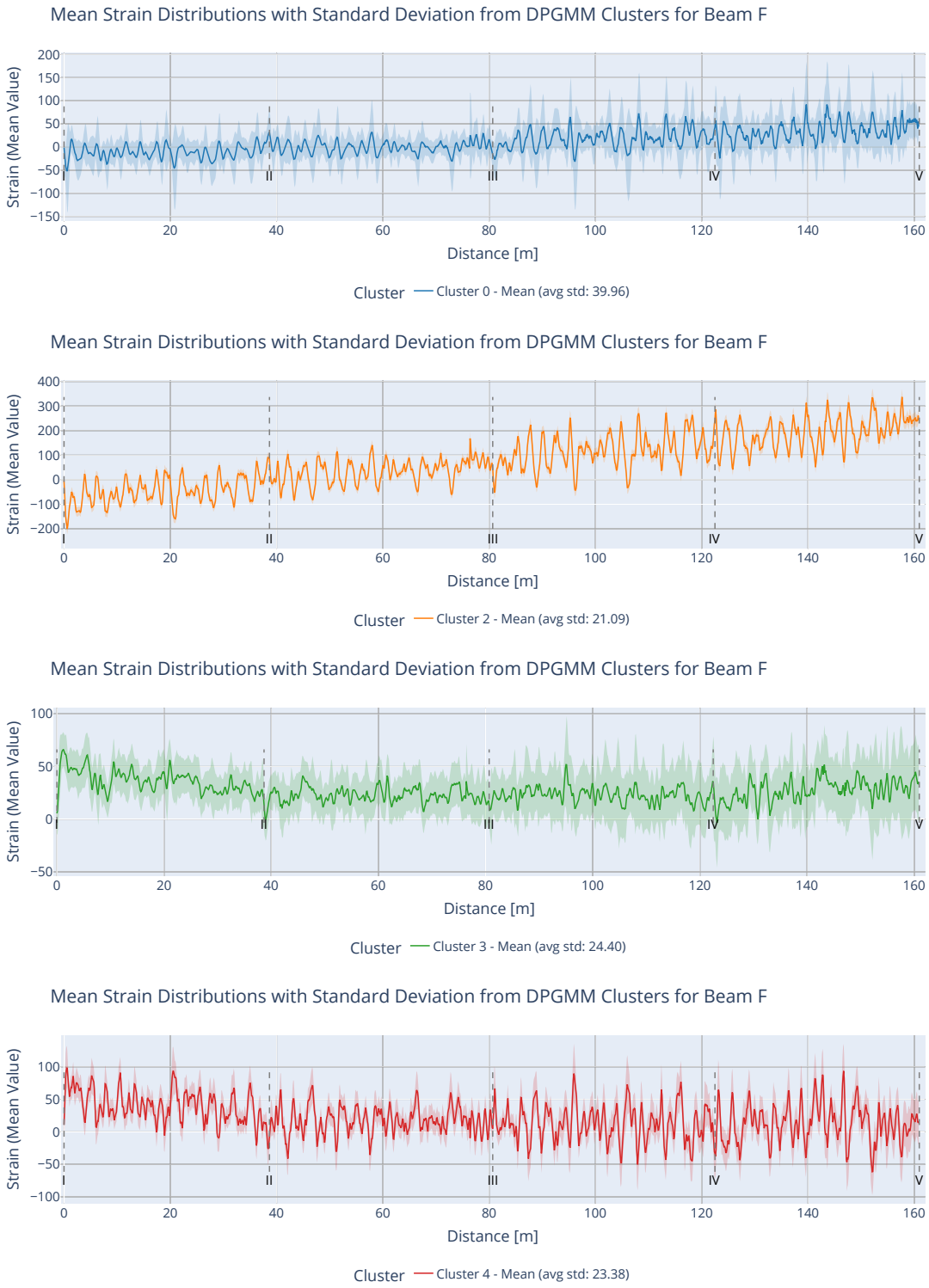

Figure B.9: Mean strain profiles (1/3) from DPGMM clustering for beam F.

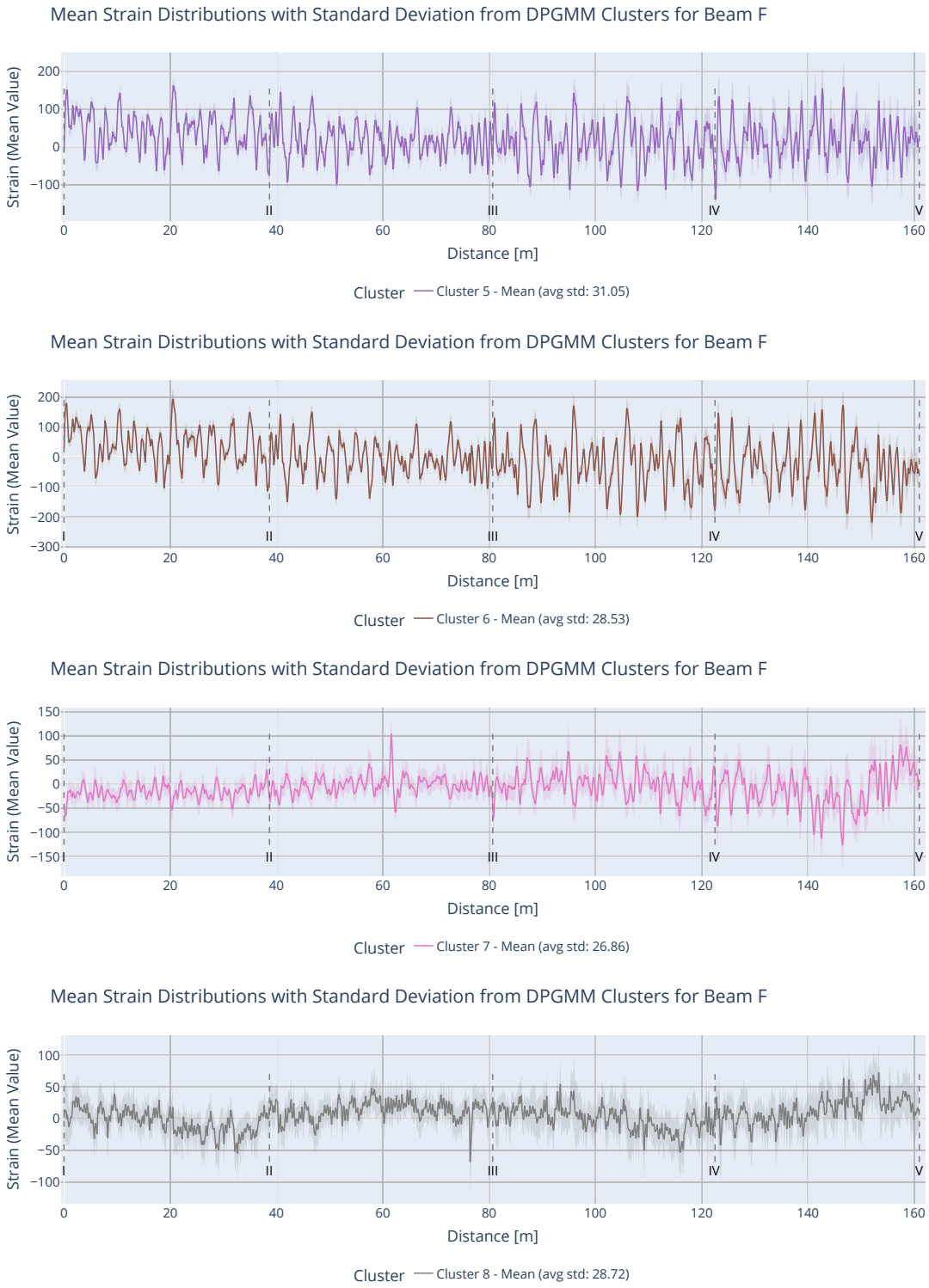

Figure B.9: Mean strain profiles (2/3) from DPGMM clustering for beam F.

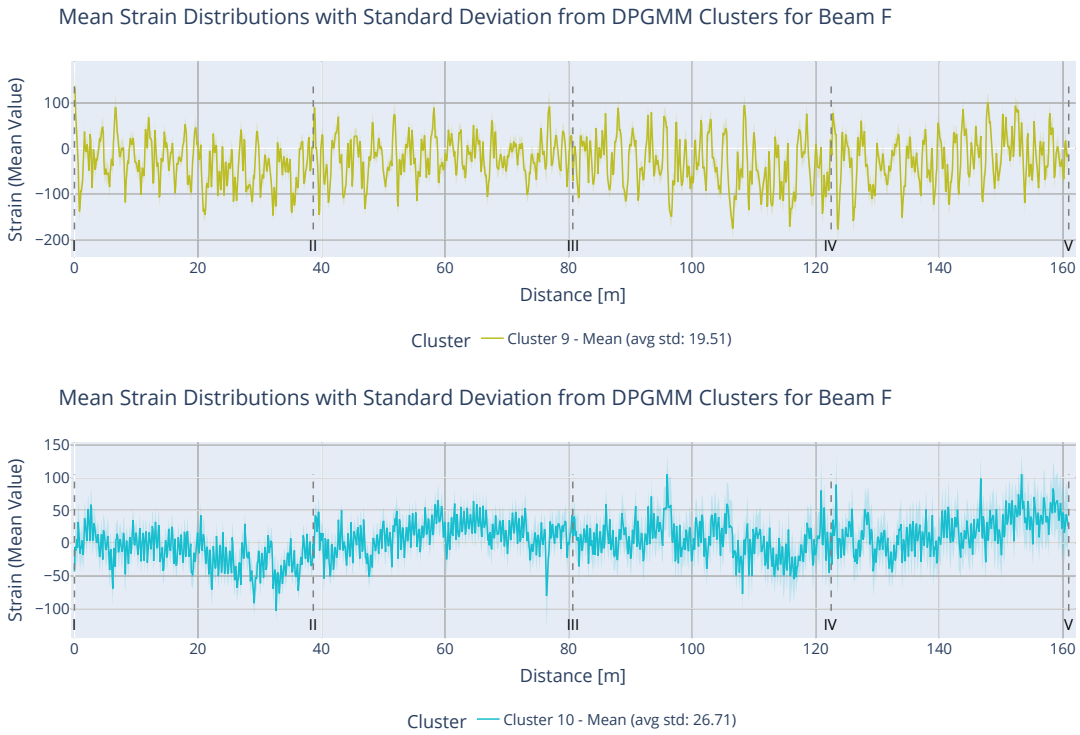

Figure B.9: Mean strain profiles (3/3) from DPGMM clustering for beam F.

Appendix C

Appendix C. Sankey Diagrams

C.1 Sankey Diagram Showing the Evolutions of Clusters

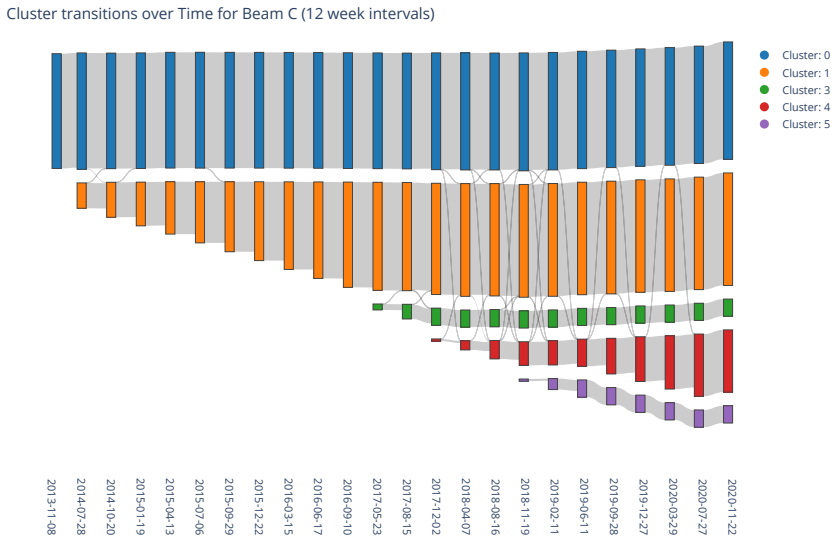

Figure C.1: Sankey diagram showing the drift of data points between time steps of twelve week interval for beam C. Each node represents the number of data points within a specific cluster at a certain moment in time. The links represent the flow of data points from one node to another between t_i and t_{i+1}.

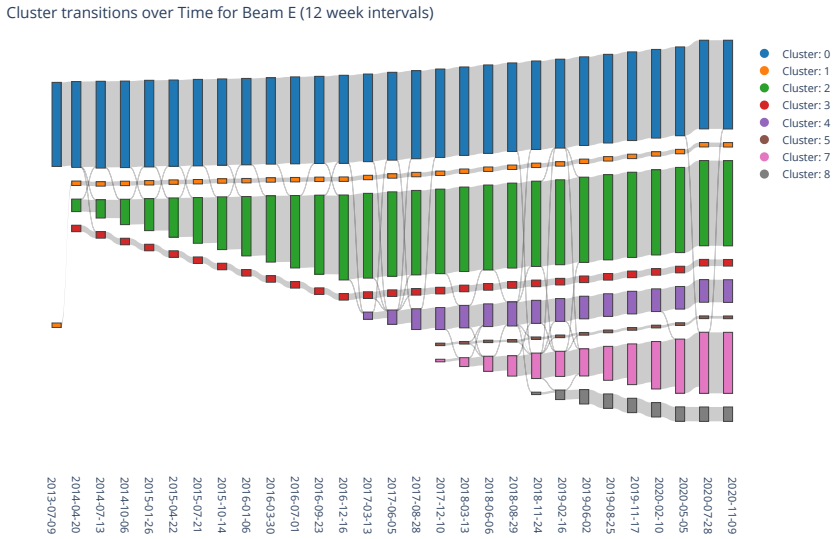

Figure C.2: Sankey diagram showing the drift of data points between time steps of twelve week interval for beam E (bottom). Each node represents the number of data points within a specific cluster at a certain moment in time. The links represent the flow of data points from one node to another between t_i and t_{i+1}.

Appendix D

Appendix D. Correlation Between Channels

D.1 Correlation Between Time Series

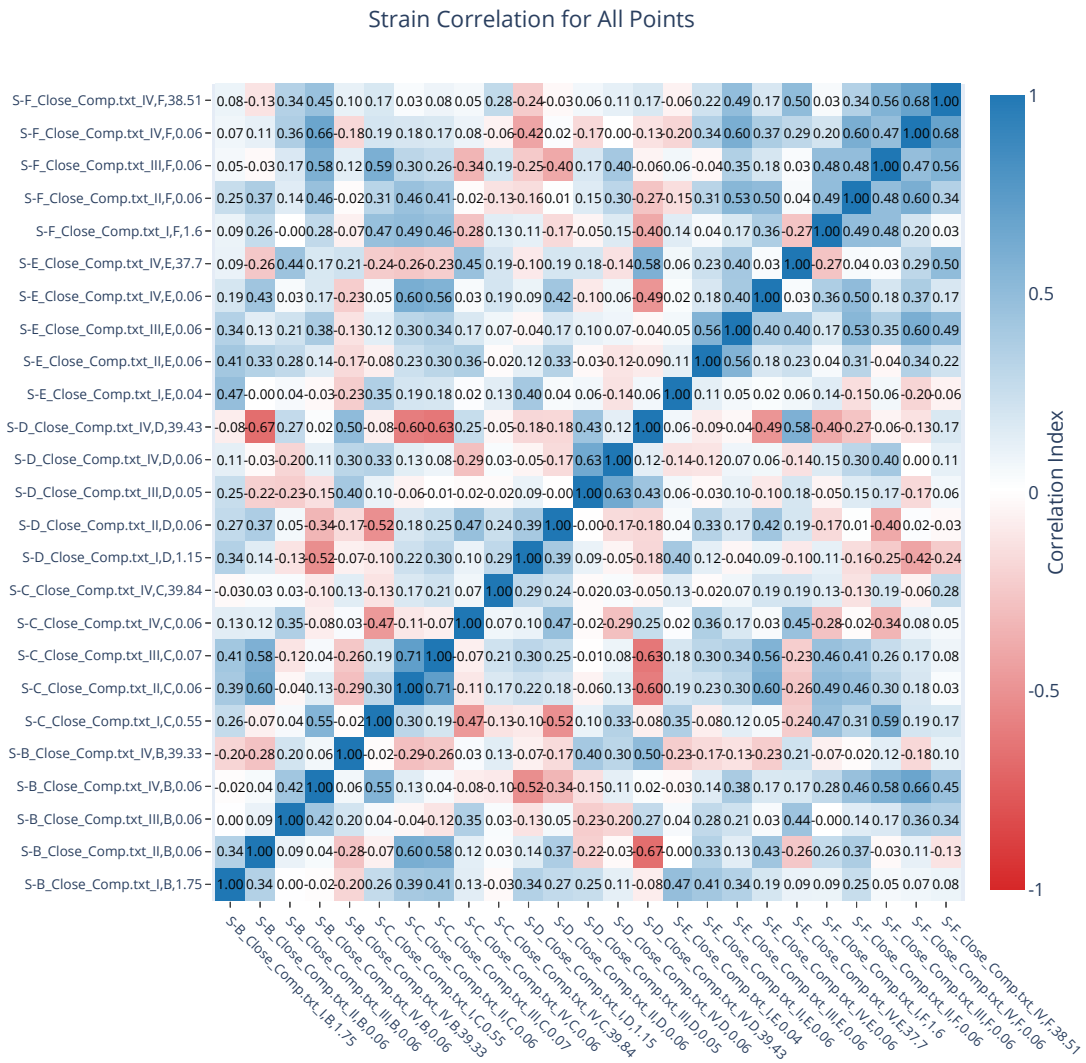

Figure D.1: Correlation between the timeseries on Älvbrodelen.

Appendix E

Appendix E. GitHub Repository

The project repository is available at:
https://github.com/viktorkjellsson/Master-Thesis-Anomaly-Detection

Python version: 3.12.9

Preinstalled Standard Libraries Used

pathlib
tqdm
sys
numpy
json
pickle
csv
collections
datetime
tkinter
time
itertools
operator

External Libraries Used

Library	Version
json	0.10.0
numpy	2.1.3
pandas	2.2.3
plotly	5.24.1
pytorch	2.5.1
scikit-learn	1.6.1
scipy	1.15.3
seaborn	0.13.2
statsmodels	0.14.4
tqdm	4.67.1

DEPARTMENT OF SOME SUBJECT OR TECHNOLOGY
CHALMERS UNIVERSITY OF TECHNOLOGY
Gothenburg, Sweden
www.chalmers.se

CHALMERS
UNIVERSITY OF TECHNOLOGY