

博士論文

A Multifidelity Simulation Framework for Digital Twin

Modelling of Spacecraft

(宇宙機のデジタルツインモデル化のための
複数忠実度のシミュレーションフレームワーク)

コンタクソグル アナスタシオス

KONTAXOGLOU ANASTASIOS

A MULTIFIDELITY SIMULATION FRAMEWORK FOR
DIGITAL TWIN MODELLING OF SPACECRAFT

by

Kontaxoglou Anastasios

A Doctor Thesis

Submitted to
the Graduate School of the University of Tokyo
on August 05, 2022
in Partial Fulfillment of the Requirements
for the Degree of Doctor in Aeronautics and Astronautics

Thesis Supervisor: Nakasuka Shinichi
Professor of Aeronautics and Astronautics

ABSTRACT

In space, where human intervention is not possible, faults must be autonomously detected. The state of the satellite should be continuously monitored, to both react and notify the on-board health management system. However, inaccuracies, delays, or disturbances can cause component breakdowns that can lead to catastrophic failures. In light of this, a dynamic system simulation can greatly enhance the operational phase of a satellite.

A major cause of inaccuracies are unmodelled physics. A multitude of factors are necessary to understand the thermal profile of the satellite. For example, incoming radiation from a thermal radiation source, e.g., the Sun, heats in a larger degree the surface facing it, compared to the dark side, with the hot side losing more momentum than the cool side, therefore the local temperature is affected by attitude. Additionally, power components release heat, increasing the temperature of surrounding components.

This research presents a multifidelity framework for the simulation of small satellites. It applies the concept of dynamic Digital Twins which includes bidirectional communication and mapping to the physical system. Emphasis is placed on handling a constant stream of live data and cokriging estimators. These estimators leverage multiple separate estimations, of varying fidelities, for a quantity of interest, and provide a better approximation compared to traditional regression methods that often treat data as spatially independent observations.

Current multifidelity modelling methods are surveyed. In particular, two approaches are used: a multifidelity approach that combines a computationally cheap, low fidelity surrogate model, with an accurate high fidelity model. In the case of the former, recurrent neural networks, particularly a Gated Recurrent Unit is considered. For the latter, a finite element model is used to produce sparse high fidelity data describing the satellite's state.

Finite element data is expensive to generate. However, abundant low fidelity data could be taken advantage of to speed up the process. By means of cokriging, low fidelity data is corrected by high fidelity data through a comprehensive correction, where the parameters are given by the use of Gaussian processes to provide uncertainty quantification. When new data arrives, the model can be refitted for a minimal computation cost.

The multifidelity framework is demonstrated using thermal data from two satellites. A) The ISSL6U-1 satellite, a non-Japanese 6U cube satellite, co-developed by the Intelligent Space Systems Laboratory (ISSL) of the University of Tokyo, intended to orbit in Low Earth Orbit (LEO) and is intended to provide a convenient validation platform to test optical sensors. B) The SDS-4 (Small Demonstration Satellite 4), a small 5U satellite developed by JAXA. It was launched as a secondary payload on the Shizuku mission on 2012 and its purpose was to conduct experiments on-board.

This research aims to improve accuracy by supplying information from data-driven methods, which include any unmodelled physics in the finite element simulator. In this work, thermal and simulated power data is used to demonstrate the multimodality of the approach. The multifidelity framework is benchmarked against traditional approaches and the ability for real time simulations is investigated by using the sliding window technique. The experimentation in this study produced promising results.

Acknowledgements

In space no one can hear you scream.

A genius marketing team in the late 70s.

I would like to start by thanking the Intelligent Space Systems Laboratory (ISSL) and my professor S. Nakasuka for his supervision and support. Additionally, I would like to thank my supervisors S. Tsutsumi and S. Khan for their day-to-day guidance, supervision, and encouragement. Last but not least, I want to thank my mother Panagiota and my uncle Vasilis, without whom I would not be where I am today.

Contents

1	Introduction	2
1.1	Research Motivation	2
1.2	Issues and Requirements	3
1.3	Objectives	4
1.4	Outline	5
2	Background Concepts	6
2.1	Multifidelity Modelling	6
2.2	Gaussian Processes	10
2.2.1	Definition and Methodology of Gaussian Processes	10
2.2.2	Gaussian Processes' Complexity	12
2.3	Kriging	13
2.3.1	Kriging Definition	13
2.4	Cokriging	14
2.4.1	Definition and Methodology of Cokriging	14
2.4.2	The Scaling Factor ρ	16
2.4.3	Cokriging Complexity	18
2.4.4	Literature on Cokriging	19
2.5	Autoregression	21
2.5.1	Modelling with ARIMAX	21
2.5.2	ARIMAX Complexity	22
2.6	Artificial Neural Networks	23
2.6.1	Feedforward Networks	24
2.6.2	Recurrent Networks	24
2.6.3	The Problem of Long-Term Dependencies	25
2.6.4	LSTMs and GRUs	26
2.6.5	CNN LSTM	27
2.6.6	Neural Network Training	28
2.6.7	GRU Complexity	28
2.7	Finite Element Analysis	31
2.7.1	Thermal Desktop/SINDA	31
2.7.2	The Heat Equation	31
2.8	Applications	32
3	Digital Twin	35
3.1	Requirements of Digital Twin	35
3.2	Model Based Simulations	38
3.3	Data Driven Models	38
3.4	Comparison	39
3.5	Hybrid Approaches	40
3.6	Combining Information	40
3.7	Digital Twin in Aerospace	42

3.8	Reduced Order Modelling	43
3.9	Research Gaps	45
4	Proposed Methodology	47
4.1	Framework	47
4.2	Multimodality	50
4.3	Unaccounted Interdependencies	51
4.4	Framework Assumptions	51
4.5	Framework Application	52
4.6	Robustness	53
5	Experiments & Simulations	59
5.1	Experiment Introduction	59
5.1.1	Disclaimer	59
5.1.2	PC Specifications	59
5.1.3	Traditional Techniques	59
5.1.4	Error and Time Metrics Definition	60
5.1.5	The Sliding Window	61
5.2	ISSL6U-1	62
5.2.1	Thermal Environment	63
5.2.2	Scenarios	65
5.2.3	GRU Training	66
5.2.4	Multifidelity Framework Testing	67
5.2.5	Uniform HFM Data Points	70
5.2.6	Random and Noisy HFM Data Points	71
5.2.7	Experiment Summary	74
5.3	Multimodality Evaluation Using ISSL6U-1 with Simulated Power Data	75
5.3.1	Simulating Power Components	76
5.3.2	Multifidelity Framework Testing with Simulated Power Data	78
5.3.3	Results	81
5.3.4	Results with Sliding Window	88
5.3.5	ARIMAX	92
5.3.6	Robustness Study	95
5.3.7	Time Allocation	98
5.3.8	Experiment Summary	98
5.4	SDS-4	99
5.4.1	The SDS-4 Satellite	99
5.4.2	Experiment Setup	100
5.4.3	Multifidelity Framework Training Data	100
5.4.4	Multifidelity Framework Testing with SDS-4 Data	102
5.4.5	Results	104
5.4.6	Results with Sliding Windows	112
5.4.7	ARIMAX	113
5.4.8	Robustness Study	117
5.4.9	Experiment Summary	119
5.5	SDS-4 Using the Entire Dataset	119
5.5.1	Multifidelity Framework Testing - Entire SDS-4 Dataset	119
5.5.2	Results GRU	120
5.5.3	Results Uniform HF Data Points	120
5.5.4	Results Random HF Data Points	124
5.5.5	Experiment Summary	128

6	Discussion & Conclusion	129
6.1	Discussion	129
6.2	Limitations	132
6.3	Contributions	132
6.4	Future Research	133
6.5	Conclusion	135
	Publications	137
	List of Tables	137
	List of Figures	139
	Appendix A	154
A.1	Multivariate Normal Distribution Theorem	154
A.2	GRU SDS-4	154
A.3	Uniform SDS-4 HF Point Placement	162
A.4	Random SDS-4 HF Points Placement	178
A.5	SDS-4 Overall With Windows	192
A.6	ARIMAX SDS-4	208
A.7	GRU SDS-4 HF Points Placement - Entire Dataset	215
A.8	Uniform SDS-4 HF Points Placement - Entire Dataset	223
A.9	Random SDS-4 HF Points Placement - Entire Dataset	239

Nomenclature

AI	: Artificial Intelligence
ANN	: Artificial Neural Network
AOCS	: Attitude & Orbit Control System
AR	: Autoregression
ARIMAX	: Autoregressive Integrated Moving Average with Explanatory Variables
CNN	: Convolutional Neural Network
CPU	: Central Processing Unit
CVE	: Cross-Validation Error
DA	: Deterministic Approach
DT	: Digital Twin
FD	: Finite Differences
FE	: Finite Element
FPGA	: Field-Programmable Gate Array
FPI	: Fixed Point Iteration
GP	: Gaussian Process
GPU	: Graphics Processing Unit
GRU	: Gated Recurrent Unit
HF	: High Fidelity
HFM	: High Fidelity Model
JAXA	: Japan Aerospace Exploration Agency
LF	: Low Fidelity
LFM	: Low Fidelity Model
LSTM	: Long-Short Term Memory Network
LTF	: Linear Transfer Function
MA	: Mean Average
MAE	: Mean Absolute Error
MAPE	: Mean Absolute Percentage Error
MC	: Markov Chain
MCMC	: Markov Chain Monte Carlo
MF	: Multifidelity
MFM	: Multifidelity Model
ML	: Machine Learning
MSE	: Mean Square Error
NDA	: Non-Deterministic Approach
PE	: Physical Environment
PHM	: Prognostics and Health Management
RBF	: Radial Basis Function
RNN	: Recurrent Neural Network
ROM	: Reduced Order Model(ling)
NSR	: Northern Sky Research
SBC	: Schwarz Bayesian Criterion
SINDA	: Systems Improved Numerical Differential Analyzer
VE	: Virtual Environment
stdv	: Standard Deviation

Chapter 1

Introduction

Both industry and research, rely progressively more on simulations, to reduce the cost of development from design to operation. This is particularly important in the space industry where costs are high and the timetables strict. Additionally, the nature of the space environment introduces high risk, so monitoring the spacecraft continuously to detect issues and tackle them early is of paramount importance. The state of the art in simulation is the Digital Twin, which functions as a virtual representation of a physical system. While a multitude of definitions have been proposed for the Digital Twin, it can be thought of as a digital model that exchanges information with a physical model and mimics its behaviour. Digital Twin is a promising approach for aerospace applications. However, since the exchange of information implies online modelling, the computational load should be minimized if possible. Therefore, multifidelity modelling methods should be considering to reduce cost while keeping accuracy high.

This paper is an expansion of our works [1, 2], which discuss a multifidelity framework in a Digital Twin architecture and a preliminary comparison of cokriging with more traditional regression methods for small satellite thermal data.

1.1 Research Motivation

In recent years, small satellites have become increasingly popular in the space industry. In this work, the term "small satellites" is not restricted to 100kg-500kg satellites but also includes micro/nano/pico-satellites with weight ranging from 1kg to 100kg. This is due to their lower costs and shorter development times, allowing both governments and academic institutions or private companies with limited resources to design and launch their own spacecraft. In particular, the CubeSat is a form of nano-satellite composed of box-shaped units (U) of dimensions 10 cm x 10 cm x 10 cm and unit weight of approximately 1.5kg [3]. According to the satellite industry market research consultant company Northern Sky Research (NSR) [4], the market has been steadily increasing and projected to increase even further in the future. While North America currently dominates in the number of launches, Europe and Asia have also started to participate and are expected to expand their markets. This is illustrated in Figure 1.1.

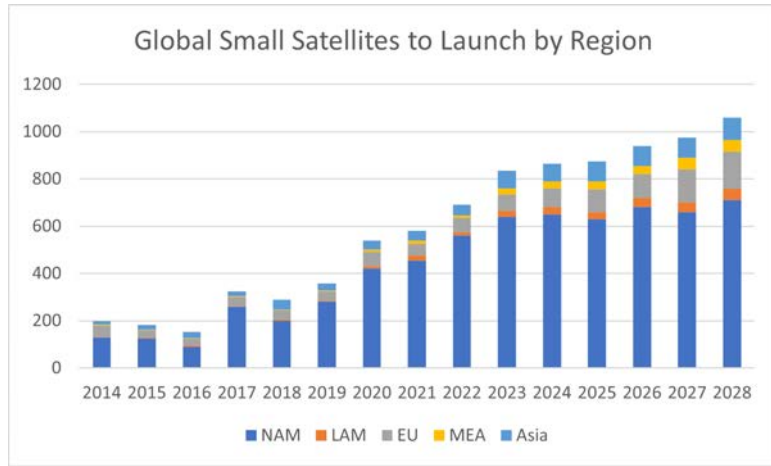


Figure 1.1: Small satellite launches, based on numbers found in NSR [4].

With the emerging interest, research on small satellites has increased. New tools are expected to satisfy ever stricter requirements for ever more specialized applications. For example, considering thermal modelling, a substantial body of work has been done on large satellites. The methods used may not necessarily be applicable to their smaller counterparts.

The common approach when simulating complex systems is to break them down into manageable constituent subsystems e.g., power subsystem, thermal subsystem, AOCS and others. This perspective encourages treating these subsystems as silos, or isolated parts in the system where data is contained and isolated, without communicating with other parts of the overall architecture. However, treating subsystem as silos does not always reflect reality as they affect each other through both modelled and unmodelled means and exchange information either as data or physically as heat, mechanics etc. The need arises for a framework that brings together an integrated, comprehensive simulation of the entire system.

The Digital Twin is a body of methodologies that attempt to accurately reflect the behaviour of a complex system, often in real time. Unfortunately, assembling those subsystems together is not a straightforward, streamlined process. Simulations of a system's individual components have different fidelities, usually high but limited by requirements of computational power. To produce an as accurate as possible approximation of the system, these fidelities would need to be adjusted.

My motivation is to transcend the silo mentality in modelling, which has high accuracy requirements, and put together a framework that can potentially assist in the construction of a Digital Twin and particularly by introducing multifidelity modelling. The focus is especially on spacecraft due to the unique challenges introduced by the space environment, but this research is by no means restricted to one particular application.

1.2 Issues and Requirements

What sets spacecraft apart from other applications is that spacecraft must operate within the harsh space environment. It can cause anomalies and unexpected faults. For example, in space there are two main sources of heat, heat generated from said environment and heat generated by internal components e.g. for communication, maneuvering, batteries etc. If the thermal state exceeds tolerance,

it can have serious effects on the system such as large thermal gradients lessening optical pointing accuracy, complicating sensor calibration, and destroying electrical components such as batteries.

With this in mind, multiple methodologies and approaches have been developed and applied to spacecraft, e.g., Prognostics and Health Management (PHM), Digital Twin, etc., to tackle these issues. Online monitoring is necessary to detect early deviant system behaviour and take appropriate steps to rectify the situation. Since the system should reflect the state of the physical object in space, an online simulation is required, with a state update rate adequately high to capture any events early but not too high, which would use computational power without providing useful information. For example, temperatures in a vacuum change slowly as there is no medium to transfer heat other than black body radiation. In that case, updating the estimation too often will not provide any benefits. Additionally, the accuracy should be as high as possible to reflect the actual state of the physical object. Finally, since spacecraft is operating in a high-risk environment, any estimation must be accompanied by an uncertainty measure to help the automatic system or operators to assess risk before acting. Therefore, to summarize, simulating satellites in space requires the following four characteristics,

1. an estimation update online,
2. manageable computational complexity,
3. high accuracy,
4. uncertainty quantification, and
5. not be application specific.

My work aims to assist both designers and operators of satellites. The design phase by providing a tool for fast and reliable offline simulation and the operation phase by estimating online a satellite's quantities of interest. Usually the simulation needs of these groups are traditionally satisfied with either

1. Numerical Analysis,
2. Knowledge-Based, or
3. Data Driven

approaches, all of which have advantages and disadvantages and have been extensively studied.

Multifidelity modelling is a popular set of methodologies that can potentially reconcile the usually competing requirements of accuracy and computational complexity. To substitute a purely high fidelity but expensive simulation, cheaper estimations can be utilized, along with a few high fidelity data values to achieve near high fidelity accuracy at a reduced cost.

1.3 Objectives

The main objectives of this thesis are summarised as,

1. Propose a framework that can potentially be used as an alternative to reduced order models in a Digital Twin architecture.

2. Test the proposed framework with simulated satellite thermal data.
3. Test the multimodality aspect of the proposed framework with simulated and obtained through testing satellite thermal and power data.
4. Evaluate the robustness of the framework to high fidelity data point placement.
5. Discuss what different sources of information offer to multifidelity modelling.
6. Conduct a review of the Digital Twin architecture in the context of space engineering.
7. Conduct a review of multifidelity approaches in the context of space engineering.

1.4 Outline

It is organized as follows. In Chapter 1, the research motivation is stated, in Chapter 2 necessary concepts, including multifidelity modelling, are explained. In Chapter 3, the concept and applications of the Digital Twin are presented. In Chapter 4 discusses the multifidelity framework and its characteristics. Chapter 5 presents simulations and the comparison between the multifidelity framework with other "traditional" approaches. Finally, Section 6 summarizes the work and discusses its limitations.

Chapter 2

Background Concepts

2.1 Multifidelity Modelling

Typically, data is either too expensive to obtain through experiments or unavailable. For this reason, multiple models must be developed to describe the same process or output quantities. These models attempt to establish a relationship between sets of input and output data. However, they often differ in the quality of the approximation and computation cost. These models can be distinguished into two categories.

- High-fidelity models (HFMs) estimate the output with the accuracy that is necessary for the current task [5].
- Low-fidelity models (LFMs) are models that estimate the output with lower accuracy than the HFM, typically in favour of lower costs [5].

Lower fidelities are generally constructed by:

1. Simplifying the mathematical model, e.g., simplifying the geometry / boundary conditions;
2. Varying the spatial and temporal resolution;
3. Using real-time distorted data.

and other methods a few examples of which can be found in Fig. 2.1. In thermal modelling an expensive finite element simulation can be considered a HFM. If the number of nodes used to solve the heat equation is reduced by the user, in order to save computational resources, then the simulation can be considered a LFM. Another example is considering structures rigid when simulating physical systems. Rigid structures could be considered a LFM through simplified physics. Including deformations in the modelling would be the HFM. Empirical data, data obtained through physical tests or received through sensors can be considered high fidelity. A cheap simulation based on simplified physics can be used as the LFM, for example not including structure deformations, and sensor measurements as the HFM supplementing data that includes information on the neglected deformations.

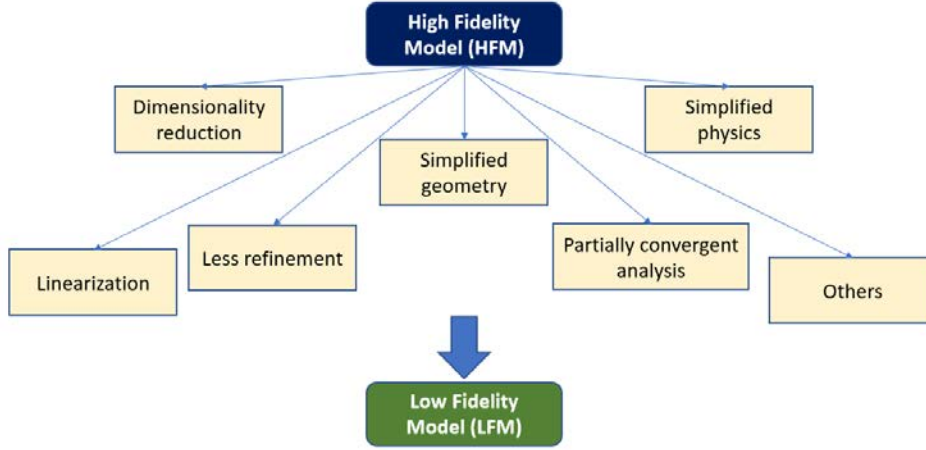


Figure 2.1: Examples of fidelity reduction methods.

While LFMs and HFMs can describe the same process, they may not produce the same estimation, e.g., the LFM might not capture some underlying processes successfully compared to the HFM. However, if one attempts to analyze a complex problem using only a HFM, the cost will quickly become impractical.

Multifidelity modelling attempts to construct a multifidelity model (MFM) by combining the strengths of both models, particularly the accuracy of the HFM and the low computational cost of the LFM. The principal challenge of multifidelity modelling in multidisciplinary problems, is the mitigation of computational cost, which scales with the increase in available models.

Peherstorfer et al. [5] surveyed multifidelity approaches for outer loop applications, namely optimization, uncertainty propagation and inference. They categorize the approaches to combine fidelities in three: adaptation, fusion and filtering.

- **Adaptation:** Adaptation strategy uses adaptation to enhance the LFM with information from the HFM.
- **Fusion:** Approaches based on fusion evaluate LFMs and HFMs and then combine information from all outputs.
- **Filtering:** Filtering approaches invoke the HFM following the evaluation of a LFM as filter. The HFM is used only if the LFM is inaccurate, or when the candidate point meets some criterion based on the LFM evaluation.

However, not all approaches can be strictly placed into one of the three categories. Multifidelity approaches can additionally be separated into two other broad categories, deterministic approaches (DAs) and non-deterministic approaches (NDAs), based on the properties of the model used to make a prediction. In DAs, the MFM is constructed by fitting the sampling points using distance based metrics such as the minimum root mean square error (RMSE) or the minimum cross-validation error (CVE). NDAs are constructed using the maximum likelihood criterion instead. DAs can be applied to any surrogate model. NDAs require an uncertainty structure. In addition, NDAs were found to be more accurate than DAs [6, 7].

In satellite development where managing uncertainty is of paramount importance, NDA methods, and in particular Bayesian theory, are well suited for their ability to not only produce an estimation for the quantity of interest but also

estimate the uncertainty. The difference between Bayesian methods and classical statistics is that the former takes advantage of any prior information available. Both approaches can be applied on the same model, e.g., a Kriging surrogate can be constructed with both Bayesian and non-Bayesian methods. Popular MFM approaches use Gaussian processes (GP) to model each fidelity response. A GP is a collection of random variables with the property that the joint distribution of any finite subset is Gaussian.

A number of multifidelity methods were considered. The approach of choice would have to be implemented in a Digital Twin architecture, therefore, it should have appropriate characteristics that conforms to the requirements of the Digital Twin specified in Section 3. Table 2.1 summarizes a few representative methods per category outlined in Peherstorfer et al. [5]. The multifidelity methods that were considered are popular in closed applications, namely, optimization, inference and uncertainty propagation.

Multifidelity Methods				
<i>Surrogate</i>	<i>Description</i>	<i>Advantages</i>	<i>Disadvantages</i>	<i>References</i>
Fusion				
Cokriging	Cokriging is the multifidelity version of Kriging, which constructs a Gaussian Process of a partially observed process.	<ul style="list-style-type: none"> - Excellent Accuracy - Robust - Few high fidelity data points required 	<ul style="list-style-type: none"> - Potentially slow due to matrix inversions. 	[8]
Control Variates	The control variate framework aims to reduce the variance of a random variable by exploiting the correlation with an auxiliary random variable with known statistics.	<ul style="list-style-type: none"> - Good Variance Reduction 	<ul style="list-style-type: none"> - Slow at higher dimensions 	[9]
Filtering				
Importance Sampling	Importance sampling uses a biasing distribution to sample the model more efficiently.	<ul style="list-style-type: none"> - Isolates extreme phenomena of interest 	<ul style="list-style-type: none"> - Sensitive to noise. - If the system complexity increases, method becomes complicated 	[10]
Adaptation				
Online Adaptive Model Correction	Projection-based reduced models are updated online as high fidelity data become available. Have both an offline and an online phase.	<ul style="list-style-type: none"> - Online Computation 	<ul style="list-style-type: none"> - Long offline phase - Unsuitable when model characteristics are not static 	[11]

Table 2.1: Representative examples of multifidelity methods.

Importance sampling uses a biasing distribution to sample the model efficiently. This is useful when we are trying to focus on specific model behaviours e.g., rare or extreme phenomena. When the filter detects these phenomena the estimation will lean more towards expensive HFM data. This method is sensitive to noise and it becomes more complicated as system complexity increases [10]. Moreover, Control Variates is a method for reducing model variance using a number of high fidelity data by exploiting the correlation with an auxiliary random variable with known statistics [9]. These methods are not suitable for my application, since they cannot provide an accurate estimation throughout the satellite's mission in an online manner.

More suitable methods are Cokriging and Online Adaptive Model Correction. Cokriging is the multifidelity version of Kriging, which uses data from a partially observed process to construct an estimation using a Gaussian process. It can update when more high fidelity data become available. On the other hand, an Online Adaptive Model Correction, builds projection - based reduced models which are again updated when high fidelity data become available. An example is Proper Orthogonal Decomposition (POD).

2.2 Gaussian Processes

2.2.1 Definition and Methodology of Gaussian Processes

A Gaussian processes model describes a probability distribution over a set of possible functions that fit a set of data points. Since we have the probability distribution over all possible function and there the means can be calculated as a function along with the variances for each prediction.

- The posteriors update with new data.
- Gaussian process model is a probability distribution over possible functions, and any finite samples of functions are jointly Gaussian distributed.
- The mean function used for regression, is calculated by the posterior distribution of possible functions.

The regression function modelled by a multivariate Gaussian is given as

$$P(\mathbf{f}|\mathbf{X}) = N(\mathbf{f}|\boldsymbol{\mu}, \mathbf{K}) \quad (2.1)$$

where $\mathbf{X} = [x_1, \dots, x_n]$, $\mathbf{f} = [f(\mathbf{x}_1), \dots, f(\mathbf{x}_n)]$, $\boldsymbol{\mu} = [m(\mathbf{x}_1), \dots, m(\mathbf{x}_n)]$ and $K_{ij} = k(\mathbf{x}_i, \mathbf{x}_j)$ [12]. \mathbf{X} are the observed data points, m represents the mean and k represents a kernel, which is a positive definite function. Given no observations the mean can be considered zero. K defines the smoothness of the functions considered and it expresses that if x_i and x_j are "similar" then it can be expected that the function outputs $f(x_i)$ and $f(x_j)$ are also "similar". Fig. 2.2 illustrates the mean function estimated by observed data.

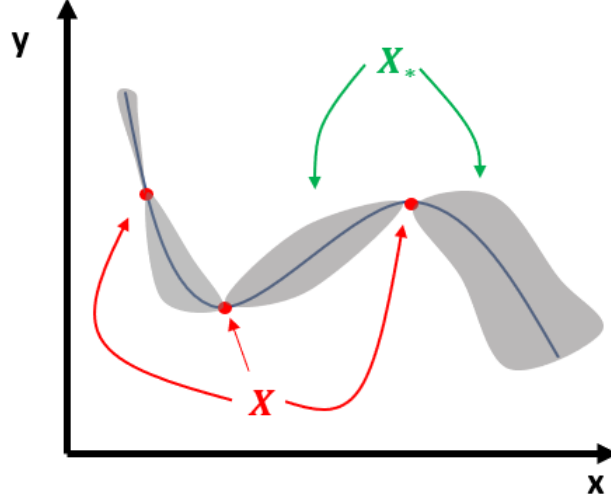


Figure 2.2: Gaussian process regression. \bullet : observed data \mathbf{X} , $-$: mean function estimated by the observed data points, \blacksquare : the confidence interval and green points \mathbf{X}_* are new points to be predicted.

The joint distribution between \mathbf{f} and \mathbf{f}_* is given by

$$\begin{bmatrix} \mathbf{f} \\ \mathbf{f}_* \end{bmatrix} \sim \mathcal{N} \left(\begin{bmatrix} m(\mathbf{X}) \\ m(\mathbf{X}_*) \end{bmatrix}, \begin{bmatrix} \mathbf{K} & \mathbf{K}_* \\ \mathbf{K}_*^\top & \mathbf{K}_{**} \end{bmatrix} \right), \quad (2.2)$$

where $\mathbf{K} = K(\mathbf{X}, \mathbf{X})$, $\mathbf{K}_* = K(\mathbf{X}, \mathbf{X}_*)$ and $\mathbf{K}_{**} = K(\mathbf{X}_*, \mathbf{X}_*)$. And the mean $(m(\mathbf{X}), m(\mathbf{X}_*)) = \mathbf{0}$. Eq. 2.2 expresses the joint probability distribution $P(\mathbf{f}, \mathbf{f}_* | \mathbf{X}, \mathbf{X}_*)$. However, to calculate regressions, the conditional distribution $P(\mathbf{f}_* | \mathbf{f}, \mathbf{X}, \mathbf{X}_*)$ over \mathbf{f}_* only is required instead. The proof can be found in Appendix A.1. The conditional distribution is

$$\mathbf{f}_* | \mathbf{f}, \mathbf{X}, \mathbf{X}_* \sim \mathcal{N} \left(\mathbf{K}_*^\top \mathbf{K} \mathbf{f}, \mathbf{K}_{**} - \mathbf{K}_*^\top \mathbf{K}^{-1} \mathbf{K}_* \right). \quad (2.3)$$

Usually the true function values are not available and noise must be taken into account. The function must be adjusted for noise as

$$y = f(x) + \epsilon. \quad (2.4)$$

If we assume Gaussian noise with variance σ_n^2 , additive independent and identically distributed, the prior is

$$\text{cov}(y) = \mathbf{K} + \sigma_n^2 I. \quad (2.5)$$

The joint distribution of observed values and the function values at new observations are

$$\begin{pmatrix} \mathbf{y} \\ \mathbf{f}_* \end{pmatrix} \sim \mathcal{N} \left(\mathbf{0}, \begin{bmatrix} \mathbf{K} + \sigma_n^2 I & \mathbf{K}_* \\ \mathbf{K}_*^\top & \mathbf{K}_{**} \end{bmatrix} \right) \quad (2.6)$$

Deriving the conditional distribution, the predictive equations for Gaussian processes regression are

$$\bar{\mathbf{f}}_* | \mathbf{X}, \mathbf{y}, \mathbf{X}_* \sim \mathcal{N}(\bar{\mathbf{f}}_*, \text{cov}(\mathbf{f}_*)) \quad (2.7)$$

where,

$$\begin{aligned}\bar{\mathbf{f}}_* &\triangleq \mathbb{E} [\bar{\mathbf{f}}_* \mid \mathbf{X}, \mathbf{y}, \mathbf{X}_*] \\ &= \mathbf{K}_*^\top [\mathbf{K} + \sigma_n^2 I]^{-1} \mathbf{y}\end{aligned}\tag{2.8}$$

$$\text{cov}(\mathbf{f}_*) = \mathbf{K}_{**} - \mathbf{K}_*^\top [\mathbf{K} + \sigma_n^2 I]^{-1} \mathbf{K}_*\tag{2.9}$$

The variance function $\text{cov}(\mathbf{f}_*)$, is not depended on the observed output \mathbf{y} . It is only depended on the inputs \mathbf{X} and \mathbf{X}_* . This is a property of the Gaussian distribution [13].

Choosing appropriate hyperparameters is very important for successful regression. The choice of kernel is particularly important. The kernel is a covariance function that determines most of the generalization properties of a Gaussian process model. A collection of kernels can be found in Duvenaud [14]. Additionally you can make your own kernel or use a combination of popular kernels. Kernels are sensitive to their hyperparameters. One of the most popular kernels and the kernel of choice for my work is the Radial Basis Function or RBF kernel. The RBF function is

$$k(\mathbf{x}_i, \mathbf{x}_j) = \sigma_f^2 \exp\left(-\frac{1}{2l} (\mathbf{x}_i - \mathbf{x}_j)^\top (\mathbf{x}_i - \mathbf{x}_j)\right).\tag{2.10}$$

where, σ_f is the vertical scale or how much the function can span vertically and l is the horizontal scale or how much the correlation between two points decreases as their distance increases. The optimal hyperparameters Θ^* can be calculated by the log marginal likelihood [12],

$$\Theta^* = \arg \max_{\Theta} \log p(\mathbf{y} \mid \mathbf{X}, \Theta).\tag{2.11}$$

Taking hyperparameters into account Eq. 2.7 for predicting new testing points becomes

$$\bar{\mathbf{f}}_* \mid \mathbf{X}, \mathbf{y}, \mathbf{X}_*, \Theta \sim \mathcal{N}(\bar{\mathbf{f}}_*, \text{cov}(\mathbf{f}_*)).\tag{2.12}$$

More information and tutorials can be found in the works by Schulz et al. [15] and Wang [16]. Gaussian process regression is analogous to simple Kriging.

2.2.2 Gaussian Processes' Complexity

In Eq. 2.3, the term K^{-1} , where K is the kernel. Inverting an $n \times n$ matrix is a computationally difficult task with $\mathcal{O}(n^3)$ complexity and $\mathcal{O}(n^2)$ for storage. This inhibits the use of Gaussian processes in many applications. In Gaussian processes a large set is defined to have at least a few thousands of data and these are usually, video, population scale medical datasets etc. However, rarely in Machine Learning such an inversion is calculated, instead approximate methods are used to speed up computation. Looking at Eq. 2.8, the predictor including noise,

$$\bar{\mathbf{f}}_* = \mathbf{K}_*^\top [\mathbf{K} + \sigma_n^2 I]^{-1} \mathbf{y} \Rightarrow \mathbf{y} = \mathbf{K}_*^\top [\mathbf{K} + \sigma_n^2 I] \bar{\mathbf{f}}_*.\tag{2.13}$$

Using an iterative method [17], a sequence of approximate solutions can be generated that will closely approach the value for $\bar{\mathbf{f}}_*$. A suitable iterative method is the conjugate-gradient² [18] method because $[\mathbf{K} + \sigma_n^2 I]$ is positive definite and symmetric. The stopping criterion for the iterative method is

$$\|\mathbf{y} - (\mathbf{K} + \lambda\mathbf{I})\xi_k\|_2 \leq \eta \|\mathbf{y} - (\mathbf{K} + \lambda\mathbf{I})\xi_0\|_2 \quad (2.14)$$

for a given tolerance $0 < \eta < 1$. An estimation of the iterations required is

$$k \geq \frac{\ln \left[\frac{2\sqrt{\kappa}}{\eta} \right]}{2 \ln \left[\frac{\sqrt{\kappa}+1}{\sqrt{\kappa}-1} \right]}. \quad (2.15)$$

where $\kappa = \lambda_{\max}/\lambda_{\min}$ is the spectral condition number. This way, every iterations only requires one matrix-vector multiplication and $5N$ flops per iteration. Additionally, 4 vectors of N length are required for storage. The new complexity is now $\mathcal{O}(kn^2)$ [19].

2.3 Kriging

2.3.1 Kriging Definition

Kriging is based on the idea that if two points x_i, x_j are close to each other then the random variables $Y(x_i), Y(x_j)$ will be similar. This is expressed statistically through their correlation

$$\begin{aligned} R &= \text{Corr}[Y(\chi_i), Y(\chi_j)] \\ &= \exp \left(-\sum_{l=1}^d 10^{\theta_l} \|\chi_{il} - \chi_{jl}\|^{p_l} \right) \end{aligned} \quad (2.16)$$

where θ_l and p_l represent the hyperparameters of the l th variable. These hyperparameters are chosen via a maximization of the concentrated log likelihood function [20]. If n is the number of data points, the log likelihood ϕ is

$$\phi = \frac{n}{2} \ln(\hat{\sigma}^2) - \frac{1}{2} \ln(|\mathbf{R}|) \quad (2.17)$$

where the optimal variance σ^2 and mean μ are estimated by

$$\hat{\sigma}^2 = \frac{1}{n} (\mathbf{y} - \mathbf{1}\hat{\mu})^T \mathbf{R}^{-1} (\mathbf{y} - \mathbf{1}\hat{\mu}) \quad (2.18)$$

and

$$\hat{\mu} = \frac{\mathbf{1}^T \mathbf{R}^{-1} \mathbf{y}}{\mathbf{1}^T \mathbf{R}^{-1} \mathbf{1}} \quad (2.19)$$

The concentrated likelihood function only depends on the hyperparameters through the correlation matrix. The main computation cost of this method arises from the optimization problem of maximizing the likelihood. Attempts have been made to improve computation, such as in Toal et al. [21] and Giles [22], in which the adjoint of the correlation matrix can be used to reduce computation time. Based on the algebraic calculations by Giles [22], the adjoint of the correlation matrix is

$$\overline{\mathbf{R}} = \frac{1}{2\hat{\sigma}^2} \mathbf{R}^{-T} (\mathbf{y} - \mathbf{1}\hat{\mu})^T (\mathbf{y} - \mathbf{1}\hat{\mu})^T \mathbf{R}^{-T} - \frac{1}{2} \mathbf{R}^{-T} \quad (2.20)$$

and by deriving the likelihood,

$$\frac{\partial \phi}{\partial \theta_l} = \ln 10 \sum_{ij} -10^{\theta_l} \|\mathbf{x}_{i\cdot} - \mathbf{x}_{j\cdot}\|^{p_l} \mathbf{R}_{ij} \overline{\mathbf{R}}_{ij} \quad (2.21)$$

and

$$\frac{\partial \phi}{\partial p_l} = \sum_{ij} -10^{\theta_l} \|\mathbf{x}_{i_l} - \mathbf{x}_{j_l}\|^{p_l} \ln \|\mathbf{x}_{i_l} - \mathbf{x}_{j_l}\| \mathbf{R}_{ij} \bar{\mathbf{R}}_{ij} \quad (2.22)$$

The kriging predictor, for a particular set of hyperparameters, is

$$\mathbf{y}(\mathbf{x}^*) = \hat{\mu} + \mathbf{r}^T \mathbf{R}^{-1}(\mathbf{y} - 1\hat{\mu}) \quad (2.23)$$

where \mathbf{r} is a vector of correlations between the unknown point \mathbf{x}^* and previous sample points \mathbf{x}_i . For more information, a comprehensive guide on kriging can be found in Jones [20].

2.4 Cokriging

2.4.1 Definition and Methodology of Cokriging

Cokriging, also known as Multifidelity Gaussian process regression is a popular surrogate model for geospatial applications [23, 24]. Fundamentally, cokriging is a weighted combination of a low fidelity kriging model and the difference between low and high fidelity models.

The seminal papers [25, 26], proposed a Bayesian framework for simulating using models for which faster approximations are available. In particular the correlation between a low fidelity model with input-output pairs (X_L, Y_L) and high fidelity model with input-output pairs (X_H, Y_H) . A base kriging surrogate model is constructed using the LFM and then a second kriging surrogate is constructed using the discrepancy between the HFM and the LFM. If the discrepancy function is defined as

$$\delta = Y_H - \rho Y_L, \quad (2.24)$$

then the estimation of δ is given by replacing the models, with their estimation as

$$\hat{\delta} = \hat{Y}_H - \rho Z_L \quad (2.25)$$

where, Z_L , a Gaussian process fitted on LFM data or

$$\hat{Y}_H = \rho Z_L(x) + Z_\delta \quad (2.26)$$

which can be viewed as a comprehensive correction with a scaling factor ρ . Z_δ is a Gaussian Process approximation for the discrepancy δ . \hat{Y}_H is an estimation for the HFM. The scale factor ρ is estimated using the maximum likelihood criterion during the computation of Z_δ .

An important assumption is that in points where high fidelity estimation exists, a low fidelity estimation also exists. Its covariance matrix \mathbf{C} can be given as

$$\mathbf{C} = \begin{pmatrix} C_{11} & C_{12} \\ C_{21} & C_{22} \end{pmatrix}, \quad (2.27)$$

$$\begin{aligned} C_{11} &= \sigma_L^2 \mathbf{R}_L(\mathbf{X}_L, \mathbf{X}_L), \\ C_{12} &= \rho \sigma_L^2 \mathbf{R}_L(\mathbf{X}_L, \mathbf{X}_H), \\ C_{21} &= \rho \sigma_L^2 \mathbf{R}_L(\mathbf{X}_H, \mathbf{X}_L) \text{ and} \\ C_{22} &= \rho^2 \sigma_L^2 \mathbf{R}_L(\mathbf{X}_H, \mathbf{X}_H) + \sigma_d^2 \mathbf{R}_\delta(\mathbf{X}_H, \mathbf{X}_H), \end{aligned}$$

where x_H and x_L are predicted high and low fidelity data points, respectively. R_L and R_δ are correlation matrices for high-low fidelity models and the discrepancy

δ respectively. σ signifies the standard deviation. While the correlations are the same as in standard kriging, the number of hyperparameters doubles. If more fidelities are used, the number increases even further.

While the correlations are the same as in the standard kriging, the number of hyperparameters doubles. If more fidelities are used, then the number increases even further. The hyperparameters of the LFM, assuming the LFM data is independent of the HFM data, can be calculated as in the standard kriging. However to calculate the hyperparameters corresponding to the HFM, the difference \mathbf{d} between the expensive and cheap data at known points should be calculated as

$$\mathbf{d} = \mathbf{y}_H - \rho \mathbf{y}_L(\mathbf{X}_H). \quad (2.28)$$

Therefore, the concentrated log likelihood for the HFM data is

$$\phi_d = -\frac{n_H}{2} \ln(\hat{\sigma}_d^2) + \frac{1}{2} \ln(|\mathbf{R}_d(\mathbf{X}_H, \mathbf{X}_H)|). \quad (2.29)$$

The variance and the mean are given by the same Eq. (2.19), (2.18), however the parameters are replaced by their equivalents obtained through the difference model $\mathbf{R}_d(\mathbf{X}_H, \mathbf{X}_H)$, \mathbf{d} and n_H . Methods from standard kriging, such as Giles [22], can again be used to reduce computation time.

To accelerate the likelihood maximisation an adjoint matrix of the hyperparameters can be used. Again as in kriging, Eq. 2.20 is used for calculation by replacing the terms with those of the differencing model. Derivatives can be calculated as in Eq. 2.21 and Eq. 2.22. By taking the adjoint of the cheap-expensive differences,

$$\bar{\mathbf{d}} = -\mathbf{R}_d^{-1}(\mathbf{d} - \mathbf{1}\hat{\mu}_d) \left(\frac{1}{\hat{\sigma}_d^2} \right) \quad (2.30)$$

and the derivative of the likelihood with respect to the scaling factor ρ ,

$$\frac{\partial \phi}{\partial \rho} = -\sum_{i=1}^{n_e} \mathbf{y}_{c_i} \bar{\mathbf{d}}_i. \quad (2.31)$$

The final cokriging predictor is similar to the one for kriging. The augmented covariance matrix is produced by the data set augmented by an unknown point as

$$\tilde{\mathbf{C}} = \begin{pmatrix} \mathbf{C} & \mathbf{c}^T \\ \mathbf{c} & \rho^2 \hat{\sigma}_c^2 + \hat{\sigma}_d^2 \end{pmatrix} \quad (2.32)$$

where, \mathbf{c} is a column vector of the covariances between the unknown point and the known points. The predictor for an expensive function at a new point is

$$y_e(\mathbf{x}^*) = \hat{\mu} + \mathbf{c}^T \mathbf{C}^{-1}(\mathbf{y} - \mathbf{1}\hat{\mu}) \quad (2.33)$$

where, $\hat{\mu}$ is

$$\hat{\mu} = \frac{\mathbf{1}^T \mathbf{C}^{-1} \mathbf{Y}}{\mathbf{1}^T \mathbf{C}^{-1} \mathbf{1}} \quad (2.34)$$

and \mathbf{Y} is a vector containing the known cheap and expensive data.

2.4.2 The Scaling Factor ρ

The scaling factor, in the initial work by Kennedy and O'Hagan [25] was considered constant. However, it can be replaced by a linear, nonlinear or non-deterministic function. The role of the scaling factor is to map between the low and high-fidelity functions. If that relationship is linear, a constant ρ suffices. If the relationship is not linear, then a nonlinear scaling factor is required. Considering the following example [27], Eq. 2.35, 2.36, describe the low and high fidelity models as

- LFM:

$$y_{LFM} = \sin(8\pi x) \quad (2.35)$$

- HFM:

$$y_{HFM} = (x - \sqrt{2}) \sin^2(8\pi x) \quad (2.36)$$

Fig. 2.3 contains the plots for Eq. 2.35, 2.36.

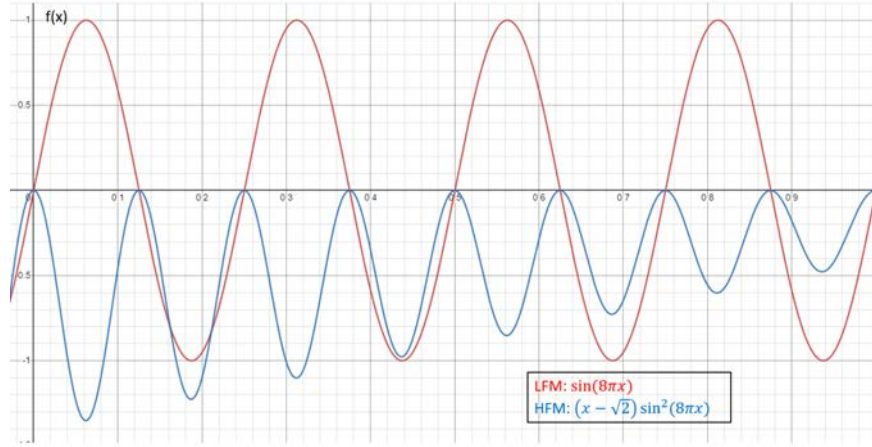


Figure 2.3: An example of LFM and HFM. —: Low fidelity model \mathbf{X} , —: High fidelity model.

If cokriging is applied to the data, the multifidelity model, in Fig. 2.4, fails to capture accurately the shape of the high fidelity model. This is due to the relationship between the models being nonlinear as in Fig. 2.5.

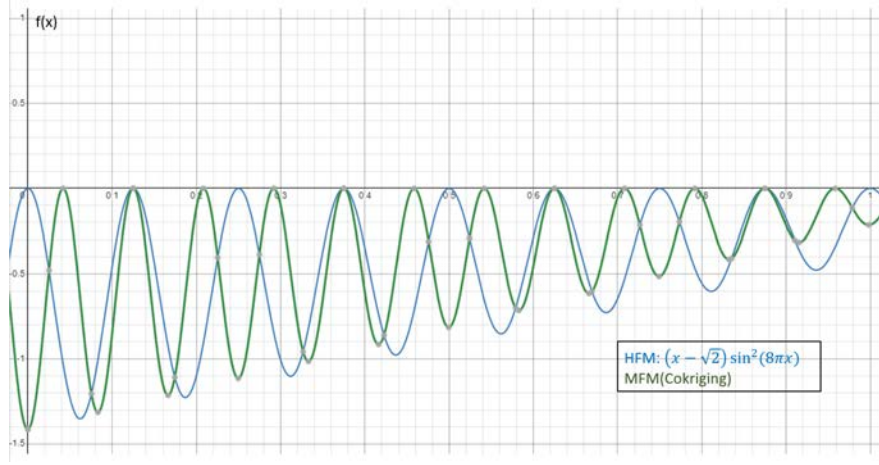


Figure 2.4: Cokriging with constant ρ . —: Low fidelity model \mathbf{X} , —: High fidelity model, —: Multifidelity model.

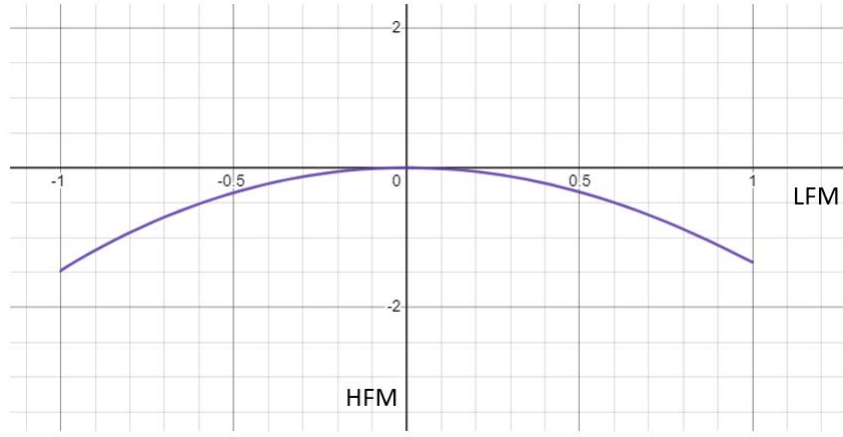


Figure 2.5: Relationship between the LFM and HFM.

If a non deterministic ρ is used, the multifidelity model captures the high fidelity model successfully in Fig. 2.6.

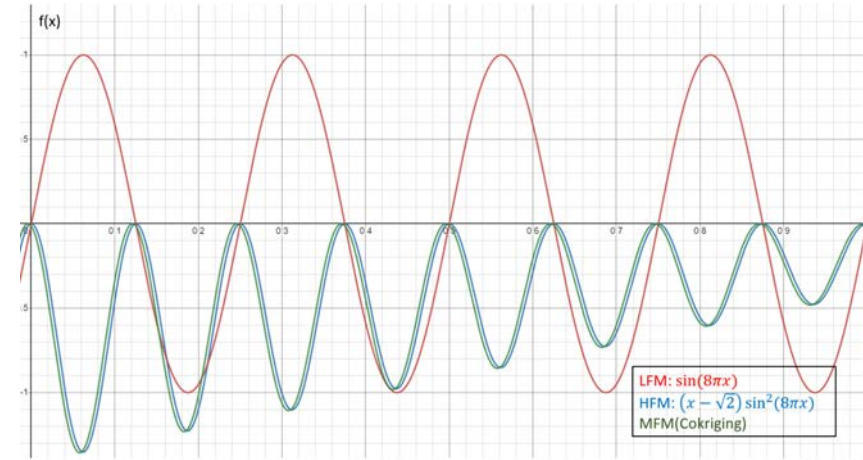


Figure 2.6: Cokriging with nonlinear ρ . —: Low fidelity model \mathbf{X} , —: High fidelity model, —: Multifidelity model.

2.4.3 Cokriging Complexity

Le Gratiet and Cannamela [28] proposed that cokriging with s -levels of fidelity can be decoupled to s independent kriging problems. Rewriting the cokriging Eq. 2.26 in a more mathematically strict form,

$$\begin{cases} Z_t(x) = \rho_{t-1}(x)Z_{t-1}(x) + \delta_t(x) \\ Z_{t-1}(x) \perp \delta_t(x) \\ \rho_{t-1}(x) = g_{t-1}^T(x)\beta_{\rho_{t-1}} \end{cases} \quad (2.37)$$

where,

$$\delta_t(x) \sim \mathcal{GP}(f_t^T(x)\beta_t, \sigma_t^2 r_t(x, x')) \quad (2.38)$$

and

$$Z_1(x) \sim \mathcal{GP}(f_1^T(x)\beta_1, \sigma_1^2 r_1(x, x')). \quad (2.39)$$

In this case, T stands for the transpose, \perp denotes the independence relationship, \mathcal{GP} stands for Gaussian Process, $g_{t-1}(x)$ is a vector of q_{t-1} regression functions, $f_t(x)$ is a vector of p_t regression functions, $r_t(x, x')$ is a correlation function, β_t is a p_t -dimensional vector, $\beta_{\rho_{t-1}}$ is a q_{t-1} -dimensional vector, and σ_t^2 is a positive real number.

Le Gratiet and Cannamela [28] expressed Eq. 2.37 as,

$$\begin{cases} Z_t(x) = \rho_{t-1}(x)\tilde{Z}_{t-1}(x) + \delta_t(x) \\ \tilde{Z}_{t-1}(x) \perp \delta_t(x) \\ \rho_{t-1}(x) = g_{t-1}^T(x)\beta_{\rho_{t-1}} \end{cases} \quad (2.40)$$

where, $\tilde{Z}_{t-1}(x)$ is a Gaussian process with distribution

$$\left[Z_{t-1}(x) \mid \mathcal{Z}^{(t-1)} = z^{(t-1)}, \beta_{t-1}, \beta_{\rho_{t-2}}, \sigma_{t-1}^2 \right], \quad (2.41)$$

with $\delta_t(x)$ being a Gaussian process with distribution as in Eq. 2.38 and

$$D_s \subseteq D_{s-1} \subseteq \dots \subseteq D_1. \quad (2.42)$$

The only difference from Eq. 2.37 is the way Z_t , or the Gaussian process at fidelity level t is represented as a function of the Gaussian process Z_{t-1} at points in the experimental design sets $(D_i)_{i=1, \dots, t-1}$.

If the nested property for the experimental design sets is assumed, defined as a class of experimental design in which every level of a given factor appears with only a single level of any other factor, then for fidelities $t = 2, \dots, s$ and $x \in U$:

$$\left[Z_t(x) \mid \mathcal{Z}^{(t)} = z^{(t)}, \beta_t, \beta_{\rho_{t-1}}, \sigma_t^2 \right] \sim \mathcal{N}(\mu_{Z_t}(x), s_{Z_t}^2(x)), \quad (2.43)$$

where,

$$\begin{aligned} \mu_{Z_t}(x) = & \rho_{t-1}(x)\mu_{Z_{t-1}}(x) + f_t^T(x)\beta_t \\ & + r_t^T(x)R_t^{-1}(z_t - \rho_{t-1}(D_t) \odot z_{t-1}(D_t) - F_t\beta_t), \end{aligned} \quad (2.44)$$

and

$$\sigma_{Z_t}^2(x) = \rho_{t-1}^2(x)\sigma_{Z_{t-1}}^2(x) + \sigma_t^2(1 - r_t^T(x)R_t^{-1}r_t(x)) \quad (2.45)$$

R_t is a correlation matrix $R_t = (r_t(x, x'))_{x, x' \in D_t}$, $r_t^T(x)$ is a correlation vector $r_t^T(x) = (r_t(x, x'))_{x' \in D_t}$, $z_t(D_{t-1})$ the vector of known values of $Z_t(x)$ at points in D_{t-1} , and F_t is a matrix with the values of $f_t(x)^T$ on D_t .

- $\mu_{Z_t}(x)$: is the mean of the surrogate model of the response at level t , $1 \leq t \leq s$.
- $\sigma_{Z_t}^2(x)$: the variance represents the mean squared error.

Since the mean and variance have successfully been written as functions of fidelities of a previous level $t-1$, a recursive multifidelity metamodel has been constructed. In covariance matrices, or kernels, with the form $k(x, x') = \sigma^2 r(x, x')$, the mean of the surrogate model is independent of σ^2 . This still holds true as the mean of the predictive distribution is independent of $(\sigma_t^2)_{t=1, \dots, s}$.

The cost of cokriging is mostly due to expensive covariance matrix inversions. If the fidelities are not decoupled, then the inversion of a matrix with size of $\sum_{i=1}^s n_i \times \sum_{i=1}^s n_i$ where $n_i = |D_i|$ the number of observations at level $i = 1, \dots, s$. With the new formulation in Eq. 2.40, s independent krigings can be calculated instead, requiring the inversion of s matrices $(R_t)_{t=1, \dots, s}$ of size $(n_t \times n_t)_{t=1, \dots, s}$ where n_t corresponds to the size of the vector z_t at level $t = 1, \dots, s$. Solving the kriging problems independently reduces the computational burden significantly and additionally reduces the need for storage as storing s matrices requires less space than storing the large matrix.

Finally, expressing cokriging as in Eq. 2.40 provides the following advantages [28]:

- Lower computational complexity.
- Provides closed-form expressions for both the cokriging mean and variance.
- Allows for a fast cross-validation procedure.
- It is more interpretable as the impact of each fidelity can be assessed separately.

2.4.4 Literature on Cokriging

The Bayesian framework proposed by Kennedy and O'Hagan [25] and its follow-up [26], has become increasingly popular producing good results [7]. Multiple works have employed it [29, 30, 31]. However it also has drawbacks. Hyperparameter estimation is the bottleneck in the computation. It requires finding the optimum solution in nonlinear likelihood functions, which themselves require inversions of covariance matrices and are prone to numerical instabilities. Notable expansions of this methodology are

- Forrester et al. [32] introduced multiple fidelities, increasing efficiency with an exchange algorithm.
- Qian and Wu [31], tackled the hyperparameter estimation problem, with a non-constant ρ , using Markov Chain Monte Carlo (MCMC).
- Gratiet and Garnier [33], suggested that emerging cokriging schemes with s -levels of fidelity can be decoupled and formulated recursively as s -independent kriging problems. This eases computation by solving a sequence of simpler kriging problems with covariance matrices of smaller dimensions.

Toal [34] outlined the requirements for successful cokriging to be:

1. The correlation between the low and high fidelity function should be reasonably high, $R^2 > 0.9$.

2. No more than 80% of the total evaluation budget should be converted to cheap evaluations, $f_r < 0.8$, where f_r is the fraction of the expensive functions replaced by cheap functions.
3. More than 10% of the total evaluation budget should be converted to cheap evaluations, $f_r > 0.1$.
4. There should always be slightly more cheap data points than expensive with the inequality, $f_r > \frac{1.75}{1+1/C_r}$, giving a conservative bound for this condition, where C_r is the ration between the cost of the cheap model over the cost of the expensive model.

Many other popular MFM approaches use GPs to model each fidelity response. In particular, Qian and Wu [31] expanded on the work by Kennedy and O’Hagan [25]. The main idea was a two-step approach in which again 1) uses low fidelity data to provide a base surrogate model and 2) adjusts the model accordingly by utilizing HFM data. The application required two assumptions a) the prediction of uncertainty of one fidelity is independent of the other fidelity and b) the HFM data is considered ground truth and therefore the error between it and the true process can be neglected. In more detail, in the first step, the LFM was estimated through a Gaussian process using LFM data. Subsequently, assuming both HFM and LFM predictions for the same input x are available, the discrepancy could be modelled as $\hat{y}_{HF} = \rho \cdot y_{LF} + \delta$, where both ρ and δ are constants. In most applications however, simple constants are not enough to estimate the discrepancy. Qian and Wu [31] replaced the scalar ρ of Eq. (2.26) with a linear regression. For data with noise, GP based surrogates tend to underestimate the noise by overly smoothing the prediction [35].

A more complete formulation of cokriging can be found in Gratiet and Garnier [33]. Le Gratiet and Cannamela [28] discusses sequential design for cokriging. Both works [28, 33] used a nested space filling design for the inputs in which the inputs at higher levels are incorporated in the lower level ones. Gratiet and Garnier [33] was a special case with constant ρ . Combining both the error of the evaluation and observed errors by a leave-one-out cross-validation procedure a sequential design method was proposed. An important drawback is that both do not take into account the difference between the true values and their estimation and therefore no guarantee of accuracy. Gaussian process regression’s hyperparameters are updated recursively as data becomes available. The maximum squared error, the entropy criterion and others was used in more classical approaches [36].

Yang et al. [37] discussed physics based cokriging. Initially a low fidelity model is modelled with a Gaussian process using data by a partial physics based model. Then the discrepancy between accurate observations and that model is calculated. This way the Gaussian process estimating the output additionally satisfies the partial differential equations that were used to generate the low fidelity data. It was shown that physical constraints are satisfied within a certain error.

In Zhou et al. [38], a generalized hierarchical Gaussian cokriging model for multifidelity data fusion was compared with other methods using two engineering problems, for nested and non-nested data. Particularly, against an autoregressive model, Gaussian cokriging had similar performance for nested sampling data, but for non-nested sampling data cokriging performed better. Additionally, Gaussian cokriging outperformed the autoregressive model when few low fidelity data is available. Finally, hierarchical cokriging outperforms Gaussian cokriging for few data but if a lot of data is available, Gaussian cokriging is more accurate.

Thelen et al. [39] used cokriging with adaptive sampling to calculate intermediate values instead of the quantities of interest to speed up computation.

2.5 Autoregression

2.5.1 Modelling with ARIMAX

Before employing more complicated machine learning techniques, such as neural networks, classical linear techniques for time series forecasting need to be exhausted. Since these focus on linear relationships, they are not expected to have the performance of more advanced methods. However if they are well calibrated and the data is suitably pre-processed, they can tackle a wide variety of problems. Their main strengths are their fairly easy implementation and fast computation. A multitude of methods have been developed, however they are heavily dependent on the type of data available. A crucial assumption when handling time series is that they incorporate some form of statistical equilibrium, expressed by the concept of stationarity. A stationary time series is one whose properties do not depend on the time at which the series is observed. This implies the absence of a trend or seasonality. A stationary timeseries can be adequately described by its mean, variance, and either autocorrelation function or spectral density function. Autoregression methods work well on stationary data, however even if the series is non-stationary, the data could be transformed from non-stationary to stationary, through proper operations [40].

Timeseries describing the thermal profile of a spinning satellite are expected to have seasonality since it is orbiting the Earth and the position in relation to the Sun changes during every orbital period. In addition, the temperature will stay within an upper and lower bound, having a more or less stable mean during every revolution, meaning that there will be no trend component. However, if it is required to estimate temperatures multiple times within an orbital period, a local approximation at each estimation timestep should be taken. If the timesteps are small enough, the data points that will be fitted with autoregression will not have any seasonality but will exhibit a local trend that needs to be eliminated before applying an autoregressive method.

The AutoRegression Integrated Moving Average (ARIMA) model, an extension of the basic AutoRegression (AR) model, augmented with the errors from previous steps, for timeseries with trends, is one of the most successful, flexible, and easy to use models for the analysis of univariate time series. If the reader is unfamiliar with autoregression, Box et al. [40] is a highly regarded book on timeseries analysis.

In ARIMA (p,d,q) model, the p lags of each variable are used as regression predictors for each variable, 'q' is the order of the 'Moving Average' (MA) term, referring to the number of lagged forecast errors that should go into the ARIMA Model and 'd' is the minimum number of differencing needed to remove the trend and make the series stationary. Ho et al. [41] described ARIMA(p,d,q) mathematically as:

$$Y_t = \phi_1 Y_{t-1} + \dots + \phi_p Y_{t-p} + e_t - \theta_1 e_{t-1} - \dots - \theta_q e_{t-q} + e_t \quad (2.46)$$

Coefficients ϕ_i , θ_i capture the influence of the i_{th} lag of variable Y on itself and the errors e_i .

Priyamvada and R. Wadhvani [42] summarize neatly the process for calculating the coefficients. The AR coefficients can be calculated using ordinary least

squares, maximum likelihood estimation or the Yule-Walker method. The MA coefficients are found using the ad-hoc method which can be outlined as

1. Convert the MA (q) to an AR(∞).
2. Cut off the AR (∞) term to some suitable order, making it a finite term.
3. Use any of the three methods mentioned above to estimate AR parameters.
4. After finding the AR parameters, the MA parameters can be obtained from the relationship between them.

These techniques however work well with univariate data. My data is multivariate, since besides the temperature, satellites have other relevant features, including position, attitude etc, and instead have to be treated with an Autoregressive Integrated Moving Average with exogenous input (ARIMAX), which is related to ARIMA but is suitable for analyses where there are additional explanatory variables. ARIMAX can be viewed as a multiple regression model with one or more autoregressive terms and one or more moving average (MA) terms. This makes it suited for multivariate, non-stationary data. It is described mathematically simply by adding the exogenous X scaled by a parameter β to the ARIMA model as,

$$Y_t = ARIMA(p, d, q) + \beta X_t \quad (2.47)$$

The effect of parameter β is not as straight forward as it looks at first glance. If Eq. (2.47) is rewritten, assuming the data is already stationary, ignoring integration and using the lag or backshift operator $B^k z_t = z_{t-k}$, which expresses the value of a variable k time steps in the past, it becomes,

$$\phi(B)Y_t = \beta X_t + \theta(B)e_t \Rightarrow Y_t = \frac{\beta}{\phi(B)}X_t + \frac{\theta(B)}{\phi(B)}e_t \quad (2.48)$$

where $\phi(B) = 1 - \phi_1 B - \dots - \phi_p B^p$ and $\theta(B) = 1 - \theta_1 B - \dots - \theta_q B^q$. An increase of X_t by 1 would not increase Y_t by β since it is conditional on the past values of Y . If integration is to be taken into account, $\phi(B)$ can be replaced with $\nabla^d \phi(B)$ where $\nabla = (1 - B)$ denotes the differencing operator. Rewriting the equation this way reduces the number of parameters to be estimated and also makes use of data more efficiently.

The parameters can be estimated with two methods, the method proposed by Box and Jenkins [40], which is difficult to implement when there are more than one exogenous variable, and a method called Linear Transfer Function (LTF) described in detail by Pankratz [43]. Going into details is beyond the scope of this review, however the reader is encouraged to refer to Pankratz [43] for ARIMAX, which in the book is indicated as dynamic regression.

ARIMAX has been applied to many problems such as epidemiology [44], macroeconomics [45] and urban planning [46]. ARIMAX is often juxtaposed with neural networks [47].

2.5.2 ARIMAX Complexity

ARIMAX models offer better accuracy compared to their single variable counterparts but they require more parameters to run. A trade-off between accuracy and complexity is required. When more than one explanatory variable is used, the accuracy is expected to increase even further. Lee [48] used the Schwarz Bayesian Criterion (SBC) to quantify the relationship between accuracy and complexity. A

lower SBC value indicates a more balanced trade off. It was found that including an explanatory variable is always beneficial and worth the extra computational cost.

Reiterating Eq. 2.47,

$$Y_t = ARIMA(p, d, q) + \beta X_t \quad (2.49)$$

The parameter *beta* introduces a large number, $m \times n$, where n is the number of values of a variable and m the number of variables, of coefficients to be estimated. In Box et al. [40] the coefficients are estimated by looking at the cross-correlation function and various auto-correlation functions. Subjective decisions are taken about the orders and lags. It only works well for a single explanatory variable.

For multiple explanatory variables, coefficients are estimated by the Linear Transfer Function methodology proposed by Pankratz [43]. In short, it can be summarised with the following steps:

- Fit lagged regressions with AR errors and determine the appropriate rational lag structure from the fitted coefficients.
- Refit the model with the lag structures in order to extract the residuals.
- The order of the ARMA error process is determined from the residuals.
- The final model is re-estimated.

It is evident that the increase in parameters to be estimated, along with the need to re-estimate the model makes ARIMAX computationally expensive. A single $n \times n$ matrix inversion has a complexity of $O(n^3)$. To fit ARIMAX multiple inversions are required for matrices with large dimensions.

2.6 Artificial Neural Networks

Neural networks approach problems in a different way than ordinary computers. The computer follows predefined steps and commands to solve problems. This limits performance significantly, because if these commands are not strictly defined, the computer is unable to answer. This is not a problem for neural networks. Artificial neural networks deal with problems in a similar way human brain does and is based on its architecture. However, biological neurons conduct signals slowly compared to electronic components. For this reason the network consists of a large number of interconnected neurons that work in parallel. Neural networks can not be programmed for specific functions, but learn with examples. These examples must be chosen carefully otherwise we risk wasting time or even the neural network not working properly. The main disadvantage is that because the network itself approaches the solution to the problem, its operation can be unpredictable. On the other hand, shared computers use a cognitive approach to the problem, that is, break it down simplest known problems whose solution is implemented by a sequence of simple, strictly certain steps. These commands are converted to machine language and executed. For this reason their behaviour is predictable. Errors are due to either the programmer or to the computer hardware alone. Neural networks and algorithmic based computers do not compete but complement each other. Some tasks are offered for algorithmic solving, such as arithmetic operations and tasks that are solved more efficiently by neural networks. A lot of tasks require combining both approaches, mainly

with a conventional computer monitoring the neural network so that it functions satisfactorily.

ANN are universal approximators of nonlinear dynamic systems. Using appropriate networks, mapping between, large enough, coupled sets of input and output data can be achieved.

2.6.1 Feedforward Networks

An artificial neuron is a mathematical model that describes the behaviour of a biological neuron. In real neurons dendrites receive electricity signals from axons and other neurons, in the artificial neuron these signals are described by numerical values. In the connections between dendrites and axes, electrically signals are amplified. Something similar happens to artificial neurons. The values of each input signal is multiplied by some constant w_i weights. Real neurons neurons are triggered when the sum of the input signals exceeds a threshold. To model this phenomenon we enter the sum, multiplied by the input weights, in an activation function which decides whether or not to activate the neuron. The result is propagated to other neurons. A widely used activation function $f(u)$ is the sigmoid function,

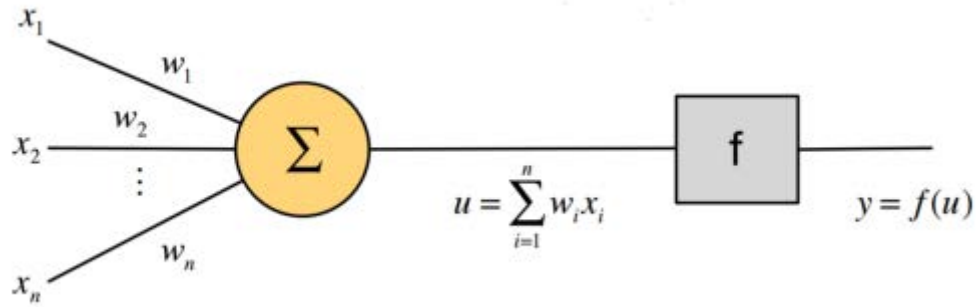


Figure 2.7: Artificial neuron, where x_i : inputs, w_i : weights, f : activation function and y : output [49].

$$f(u) = \alpha \frac{1}{1 + e^{-\beta u}} - \gamma \quad (2.50)$$

because it simulates adequately the behaviour of real firing neurons. Fig. 2.7 illustrates an artificial neuron. More complicated models have been developed for demanding applications. Values 0 and 1 are convenient for simulating the state of inaction and firing respectively. As the sum u approaches changing sign, the output tends to change state. A simpler version is the Heaviside function. [50, 51]

2.6.2 Recurrent Networks

Unlike feedforward networks in which neurons connect only to the following layers, in recurrent networks each neuron can be connected to previous layers or to other neurons in the same layer. This enables the neural network to access previous information relevant to the task at hand. Fig. 2.8 illustrates a typical recurrent neural network. Arrows not only move forward but the previous output is also fed back to the nodes for the calculation of the next output. Recurrent neural networks, instead of a loop, can be thought of as a repeating chain of identical cells as in Fig. 2.9.

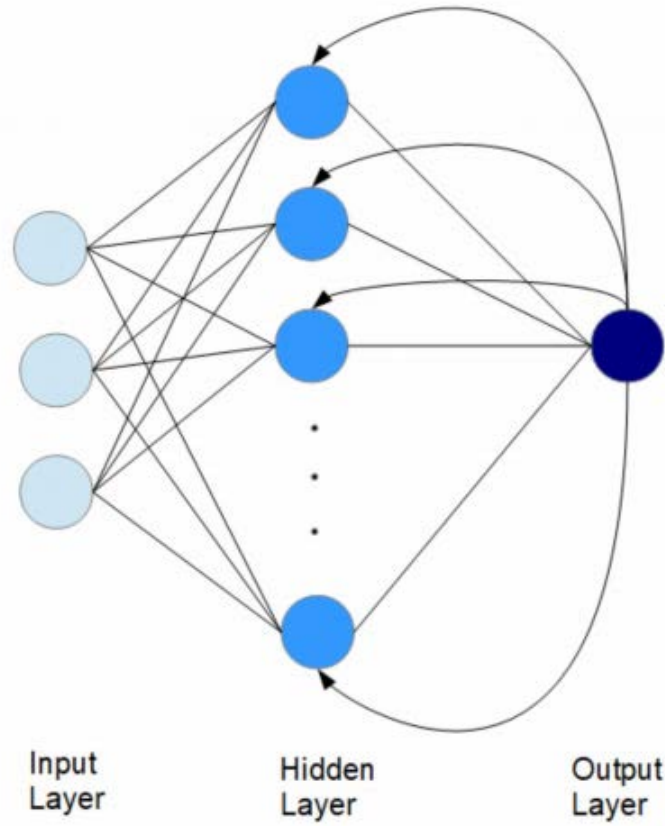


Figure 2.8: An abstract schematic of a recurrent neural network[49].

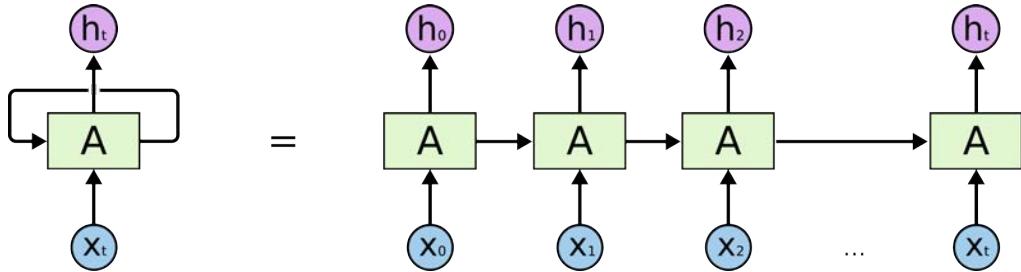


Figure 2.9: A recurrent neural network in chain form [52].

2.6.3 The Problem of Long-Term Dependencies

As mentioned, RNN's retain previous information. For example, in natural language processing, in the sentences *The sky is blue. It also has clouds*, the word *It* refers to the sky but to know this the network must have access to the previous words. Unfortunately the relevant information can be far apart in the text. The protagonist of the novel might be introduced in the first chapter, yet his name might not come up anywhere else. As the distance grows the network becomes increasingly unable to retain that information. While in theory, RNNs could remember information indefinitely, in practice this is not possible. This is due to expanding gradients and it was investigated by Bengio et al. in [53]. Long Short Term Memory networks or LSTM's were introduced in 1997 [54] to solve this problem. LSTMs remember information as their default behaviour and they do

not struggle to learn.

2.6.4 LSTMs and GRUs

In our work [1], the application of LSTMs as a surrogate model for timeseries was discussed. While their training is costly, their forward operations are fast, can handle multivariate inputs, and capture inter-dependencies between data and nonlinear trends. LSTMs have been successful for short-term forecasting e.g., Zhao et al. [55] applied LSTMs for short-term forecasting of traffic conditions, taking advantage of spatial and temporal correlations and shows strong performance against other methods for nonlinear spatial-temporal data forecasting [56, 57].

LSTMs constitute a strong alternative to auto-regressive methods. LSTMs can capture temporal inter-dependencies in addition to inter-dependencies between features and work well for short term forecasting and nonlinear data. A series of studies have compared LSTMs to auto-regressive methods. Siami-Namini et al. [58] showed 85% improvement over ARIMA for univariate, nonlinear financial data. Li et al. [59], LSTM and ARIMAX performance was directly compared in predicting tuberculosis incidents in eastern China with the inclusion of meteorological factors as additional variables. ARIMAX performed better than the neural network. In another example, Serafini et al. [60] applied both forecasting models to predict the behavior of the BITCOIN market through financial and sentiment features extracted from economic and crowdsourced data. ARIMAX again outperformed the LSTM.

ARIMAX seems to hold an edge over LSTM in the aforementioned applications. This is because tuning the hyperparameters of a neural network is a difficult task. In contrast ARIMAX can achieve good results with minimum input from the analyst. However, autoregression models are linear models and there is always the risk of them not capturing important information in a particular application. For this reason my own comparison of the two methods will be performed on satellite data with a trade off between computation time and model accuracy. The accuracy will be evaluated with three metrics, Root Mean Square Error (RMSE), Mean Absolute Percentage Error (MAPE) and Mean Absolute Error (MAE). The computation time will be evaluated as the input vector arrives to the time a prediction is generated.

A variant of the LSTM architecture is the Gated Recurrent Unit (GRU). It was first proposed by Cho et al. [61] and simplifies the cell by providing combined gates and merging the cell states among other minor changes. The GRU has been benchmarked against the standard LSTM [62, 63], with the GRU found to be slightly more efficient and accurate. Fig. 2.10 presents a layout for the GRU.

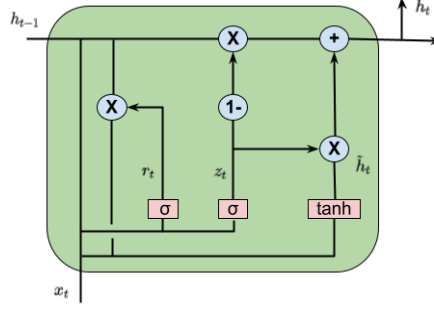


Figure 2.10: GRU layout. x_t , h_t are input and output vectors, \hat{h}_t is the candidate activation vector and z_t, r_t are the update and reset gate vectors.

The states and gates of which are described mathematically as:

- Update gate:

$$z_t = \sigma(W_z \cdot [h_{t-1}, x_t]) \quad (2.51)$$

- Reset gate:

$$r_t = \sigma(W_r \cdot [h_{t-1}, x_t]) \quad (2.52)$$

- Cell state:

$$\tilde{h}_t = \tanh(W + (r_t * h_{t-1}, x_t)) \quad (2.53)$$

- New state:

$$h_t = (1 - z_t) * h_{t-1} + z_t * \tilde{h}_t \quad (2.54)$$

where x_t , h_t are the input and output vectors, \hat{h}_t is the candidate activation vector, z_t is the update gate vector, r_t is the reset gate vector and W , W_z , W_r are parameter matrices.

2.6.5 CNN LSTM

Another candidate architecture was briefly considered, the CNN-LSTM, where CNN indicates a Convolutional Neural Network. It stacks CNN layers for feature extraction on input data with LSTM layers to support sequence prediction as in Fig. 2.11. The data needs to have two characteristics for the CNN-LSTM to be suitable.

1. Have spatial structure in their input. This could mean a 2D images or any 2D structure. 1D structures such as sentences or timeseries can also be used.
2. Have a temporal structure in their input. The order of the data plays a role, e.g., the order images are presented or words are spoken.



Figure 2.11: Stacked CNN and LSTM layers.

However, this framework was rejected in this study as it did not manage to achieve the accuracy of a simple GRU with out data set. Fig. 2.12 represents the best estimation obtained by tuning the hyperparameters of the CNN-LSTM. It has approximately 54% larger mean square error than the GRU.

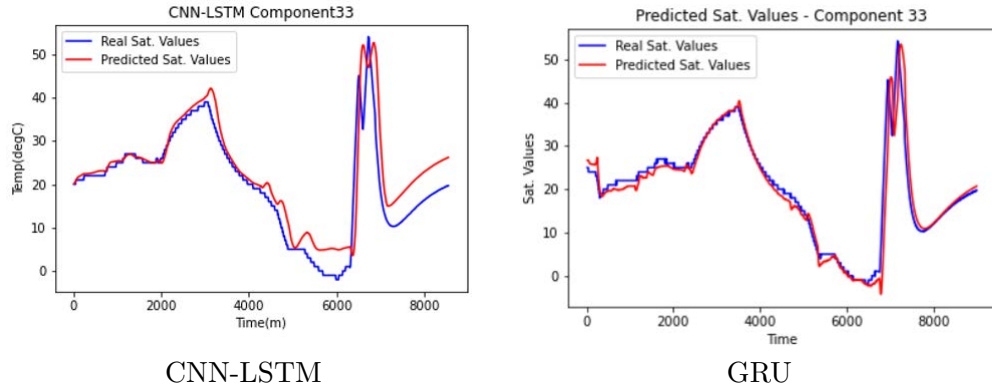


Figure 2.12: CNN-LSTM vs GRU results, SDS-4 data. Blue lines (—) indicate the real satellite values and red lines (—) the predicted data points for component 33.

2.6.6 Neural Network Training

Neural network training can be done with one of the following approaches [50, 51, 64].

- **Supervised Learning:** In this approach, a supervisor supplies the network with examples, of which he knows the answer. E.g. in pattern recognition, the network gives some estimates of the solution of the problems and then compares its answers with the correct answers are supervised by making adjustments.
- **Unsupervised Learning:** Used when no known examples are available. An example is the problem of clustering, in which the network must be based on the given data only.
- **In Reinforcement Learning:** The neural network is based on the observation of its environment and depending on whether it is positive or not the observation adapts the his weights to change his response to the next repetition.

2.6.7 GRU Complexity

Neural networks are often represented as graphs, with arrows connecting nodes through linear transformations using weights and biases. Nodes are arranged in layers. Circles represent nonlinear activation functions e.g., tanh. Layers are distinguished in the input and output layers and with hidden layers in between. The computational complexity of a network is the total number of arrows or "edges" in the network.

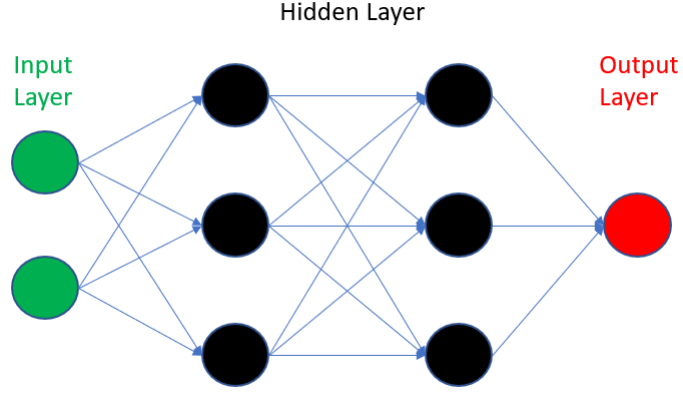


Figure 2.13: An abstract schematic of a feedforward neural network.

For a feedforward network (FNN), as in Fig. 2.13, this is as simple as

$$W = IH + HK \quad (2.55)$$

where I is the number of inputs, K be the number of outputs and H be the number of cells in the hidden layer. For example, in Japanese character recognition [65],

- There is a total of 85,000 Kanji characters.
- Only 13,000 of them are not archaic.
- Only 5,000 are regularly used. $K=5,000$
- Max 30 features extracted at every timestep.
- Number of hidden cells used are of the order of the total classes.

In most applications, $I \ll H \sim K$. In this case the computational complexity is $O(KH)$.

Expanding Eqs. 2.51, 2.52, 2.53 and 2.54,

- Reset gate:

$$r_t = \sigma(W_{ir}x_t + b_{ir} + W_{hr}h_{(t-1)} + b_{hr}) \quad (2.56)$$

- Update gate:

$$z_t = \sigma(W_{iz}x_t + b_{iz} + W_{hz}h_{(t-1)} + b_{hz}) \quad (2.57)$$

- Cell state:

$$\tilde{h}_t = \tanh(W_{in}x_t + b_{in} + r_t * (W_{hn}h_{(t-1)} + b_{hn})) \quad (2.58)$$

- New state:

$$h_t = (1 - z_t) * h_{(t-1)} + z_t * \tilde{h}_t \quad (2.59)$$

where W_{**} are weight matrices, b_{**} are bias vectors, x_t is the input vector and h_t, h_{t-1} are the current and previous states. $*$ is the Hadamard product. The computational complexity of the r_t (Reset gate), z_t (Update gate) and \tilde{h}_t (Cell state) is the same there for by using the reset gate for calculations,

$$r_t = \sigma(W_{ir}x_t + b_{ir} + W_{hr}h_{(t-1)} + b_{hr}) \quad (2.60)$$

where,

- $W_{ir} \in \mathbb{R}^{h \times d}$ is a matrix that represents the forward connections for the reset gate.
- $x_t \in \mathbb{R}^d$ is an input vector with d entries.
- $b_{ir} \in \mathbb{R}^h$ is a bias vector for forward operations
- $W_{hr} \in \mathbb{R}^{h \times h}$ is a matrix for the recurrent connections and has different dimensions from W_{ir}
- $h_{t-1} \in \mathbb{R}^h$ is the hidden state.
- $b_{hr} \in \mathbb{R}^h$ is a bias vector for recurrent operations.

Looking at the complexity of each term individually,

- $W_{ir}x_t$ is a matrix-vector multiplication with complexity $\mathcal{O}(hd)$.
- $W_{ir}x_t + b_{ir}$ is a sum of two vectors of size h with complexity $\mathcal{O}(h)$.
- $W_{hr}h_{t-1}$ is a matrix-vector multiplication with complexity $\mathcal{O}(h^2)$
- $W_{hr}h_{t-1} + b_{hr}$ is a sum of two vectors of size h with complexity $\mathcal{O}(h)$

If $N = \max(d, h)$, then the time complexity of the operations inside the nonlinearity is $\mathcal{O}(hN)$ or

$$\mathcal{O}(hd + h + h^2 + h) = \mathcal{O}(hd + 2h + h^2) = \mathcal{O}(h(d + 2 + h)) \quad (2.61)$$

The computational complexity of σ depends on the activation function chosen. If the ReLU function is chosen,

$$\sigma(x) = \max(0, x) \quad (2.62)$$

then the complexity is constant and equal to h . Since r_t, z_t and \tilde{h}_t have the same complexity then by multiplying by 3 and ignoring tanh as having the same complexity as ReLU, you can get the overall complexity, including constant terms as $\mathcal{O}(3h(d + 2 + h))$. For the final term, the new state from Eq. 2.59, $h_t = (1 - z_t) * h_{(t-1)} + z_t * \tilde{h}_t$, where $c_t \in \mathbb{R}^h$ is the cell vector and $*$ is the element-wise multiplication or Hadamard product. The complexity of this operation is $\mathcal{O}(2h)$. The overall complexity for a single GRU layer is

$$\mathcal{O}(GRU) = \mathcal{O}(3h(d + 2 + h) + 2h) \quad (2.63)$$

For multiple stacked layers the input vector x_t would be the output of the previous layer so this must be taken into account when scaling the complexity. Additionally the calculation of the loss function must be added to the complexity.

2.7 Finite Element Analysis

Finite Element (FE) and Finite Differences (FD) analyses are numerical methods for solving differential equations in engineering problems. They remain the workhorse of computer simulations and have been applied to many aspects of satellite design e.g., in thermal simulation [66] or structures [67]. In particular, Akita et al. [66] employed finite element analysis of a small satellite's thermal mathematical model, with the status of the central point, or node, of each element representing its status and using a sensitivity analysis and a Gaussian process to reduce the necessary finite element calculations.

The popularity of FE/FD analysis is due to its ability to handle complex geometries with any kind of boundary conditions and anisotropy in the properties of the system. However a major disadvantage is that the accuracy highly depends on the degree of discretization. This implies that accuracy is directly connected to computer memory and time. FE/FD analysis is costly [68].

2.7.1 Thermal Desktop/SINDA

Thermal Desktop/Systems Improved Numerical Differential Analyzer (SINDA) is a PC based design environment for thermal modelling in vehicles, satellites and electronics. Thermal Desktop uses parameter based (TRASYS like) finite difference surfaces with finite elements with CAD technology for thermal modelling. Thermal Desktop develops the capacitance and conductance network for input to SINDA/FLUINT. Thermal Desktop has thermal analysis specific type of capabilities such as applying contact conductance, insulation, heat loads, and heaters [69].

According to Aertia software company's site [69], " *Thermal Desktop provides full design parameterization using spreadsheet-like variables and arbitrarily complex expressions as input, rather than hardwired numbers. This allows complex models to be rapidly manipulated using a few keystrokes, meaning that updating or maintaining a model is trivially easy, as is performing sensitivity studies and investigating what-if scenarios. This provides access to SINDA/FLUINT's Optimization and Reliability Engineering modules, which can be used to size components, maximize performance, find worst-case operating conditions (design cases), correlate models to test data, and even treat uncertainties statistically.*"

2.7.2 The Heat Equation

Model based approaches for thermal simulations are based on solving Eq. 2.64, also known as the heat equation:

$$C_i \frac{dT_i}{dt} = Q_i - \sum_{j=1}^N C_{ij} (T_i - T_j) - \sum_{j=1}^N R_{ij} \sigma (T_i^4 - T_j^4) \quad (2.64)$$

where N is a total number of nodes, T_i and C_i are temperature and a capacitance of node i , respectively, Q_i is an external or internal thermal load to node i , C_{ij} and R_{ij} is a linear conductor and a radiation factor between node i and j , respectively, and σ is the Stephan-Boltzmann constant.



Figure 2.14: Example of FE nodes.

Eq. 2.64 is solved for all nodes, e.g., Fig. 2.14, simultaneously to calculate temperature changes. The heat load Q_i can be obtained through either the orbital environment or instrument power level, while C_{ij} and R_{ij} have to be identified through thermal-vacuum tests.

2.8 Applications

ANNs have been successfully applied in many engineering applications, including failure detection, thermal modelling [70], and aerodynamics [71, 72]. In the work by Junior et al. [73] the use of neural networks for real time simulation for the thermal modelling of satellites in orbit.

In the work by Junior et al. [73] the use of neural networks for real time simulation for the thermal modelling of satellites in orbit. In particular, [73] used a recurrent multilayer perceptron to simulate the thermal behaviour of Amazonia-1, a remote sensing satellite which was successfully launched on February 2021 with the objective to monitor the Amazon rainforest and Brazilian agricultural areas. Amazonia-1 is a medium sized satellite weighting 650kg in Sun-synchronous orbit, much larger and more complex than the 6U satellite currently under consideration. The data used for the simulation were generated by SINDA(Thermal Desktop), a high precision finite differences analysis tool for thermal modelling. It was shown that ANNs can be used for the real time simulation of satellites in orbit, with a good accuracy benchmarked against SINDA and with only some localized errors.

Additionally, Tipaldi et al. [74] investigated the possibility of on-board implementation of machine learning algorithms on satellites and proposed a general framework. With the increase of satellite's computational power in the future, it will be possible to reduce dependency on unreliable telemetry data.

Shi et al. [75] applied a cokriging framework to a real world all electric geostationary satellite's low thrust transfer. Particularly, a system dynamics and radiation damage optimization problem. The HFM used was a numerical approach of the motion equations taking into account perturbations and eclipses. As a LFM the orbital averaging technique was used for the trajectories and a linear regression model for the total dose. Multifidelity metamodels were constructed and improved by the infill sampling process for multi-objective optimization. As a result, Pareto front trajectories are obtained with limited computational budget, with both the transfer time and damage dose significantly reduced compared to the standard design. Finally, since multifidelity methods use various sources of information, model management strategies should be further developed to address which information should be utilized when but at what inputs should be used for each source.

Shi et al. [76] applied a framework based on multifidelity cokriging on an all - electric satellite in geostationary orbit. The goal was to enable a Multidiscipline Design Optimization (MDO) using cokriging to speed up computation. In

particular, simultaneously optimizing for orbit transfer dynamics and structural disciplines. Concerning the orbit transfer, for both the high and low fidelity models, finite element analysis with 500s and 1000s timestep respectively was used. Similarly, for structure analysis, a finite element analysis with twice the nodes was used in the high fidelity model versus the low fidelity model. A cokriging scheme with comprehensive correction was used to combine both models. With 15 design variables and 10 constraints, significant computational gains were made, establishing cokriging to be effective in thermal applications.

Ueda and Ogawa [77] applied a multifidelity approach on global trajectory optimization. As the LFM, a parallelized trajectory propagation performed on GPU cores, with single precision, is used to identify feasible trajectories. For the HFM parallelized gradient-based trajectory optimization on CPU cores with double precision, using the LFM as an initial guess.

Minisci et al. [78] applied multifidelity modelling to optimize the design characteristics of a re-entry unmanned space vehicle, to minimize the heat flux, the mean value of the maximum internal temperature, and the weighted sum of their variance. An evolutionary multi-objective algorithm was coupled with a direct transcription method for optimal control problems. The evolutionary algorithm determined the shape parameters and the transcription method calculated optimal control profiles. ANNs were used as a surrogate model of the aerodynamic forces required by the control solver and a multifidelity approach was employed to train them. As an LFM a low-fidelity analytical model of a waverider type vehicle is used to train the network. The network is updated through high-fidelity data by analytical and computational fluid dynamics expensive computations. The process produced realistic shapes and to generate good approximations for incidences over the 20 deg, the use of even higher fidelity models was proposed. The authors expanded their work in [79], by including a test case of a single-stage vertically ascending rocket.

Chaudhuri et al. [80] used multifidelity techniques for uncertainty propagation in two applications, a fire detection satellite problem and a fixed wing aerostuctural analysis. Fixed point iteration (FPI) with Monte-Carlo was avoided by reducing the number of required realizations for which FPI is used. LFM surrogates approximate coupling variables and the HFM were sampled to improve the LFM surrogates to maintain a high level of accuracy. The number of realizations requiring FPI was reduced by 99%.

Santos et al. [81] employed cokriging to predict the turbulent convective heating on hypersonic inflatable aerodynamic decelerators with smooth and scalloped surfaces. As a HFM, a FE simulator is used and for the LFM, a simplified physics model. The multifidelity approach showed strong performance with 7% and 10% error compared to high fidelity CFD solutions for smooth and scalloped surfaces respectively. Building a cokriging model was computationally expensive but still was approximately 5 times cheaper than a purely high fidelity model. Compared to standard, single fidelity kriging, cokriging required only 1/4 of the HFM data points.

Multifidelity Modelling in Spacecraft Applications			
<i>Reference</i>	<i>Topic</i>	<i>Application</i>	<i>Date</i>
[77]	Trajectory Planning	Optimization	2021
[75]	Orbit Determination	Optimization	2020
[76]	Thermal Modelling	Optimization	2020
[78]	Thermal Modelling	Optimization	2011
[80]	Fire Detection, Aerostructures	Uncertainty Propagation	2018
[81]	Thermal Modelling	Prediction	2018
[79]	Thermal Modelling	Optimization	2013

Table 2.2: List of Digital Twin review papers per application.

Chapter 3

Digital Twin

3.1 Requirements of Digital Twin

In the previous decades significant gains have been made in the amount and fidelity of available information concerning physical and virtual systems. The amount of information was increased by adding behavioural characteristics considered in the models but also by increasing our testing capabilities to assess their performance.

At the same time, our ability to build lighter models, both in terms of cost and time has improved. Usually through stripping down the models to the necessary characteristics or attributes and ignoring others that do not contribute as much to the quantities of interest. This reduces the size of the models. Moreover, more data is collected from the physical model and its physical behaviour, through e.g., sensors, in real time. With these lighter models, multiple subsystems can be simulated simultaneously and together compose an entire system or process.

Due to the remote nature of space, space agencies recognized early the benefits of simulation, which eventually led to the development of the concept of Digital Twin. NASA coined the term "twin" during the Apollo Program, referring to two identical space vehicles built so that the space vehicle on earth can mirror, simulate, and predict the conditions of the other in space. Since its introduction, multiple definitions have been attached to the concept of the Digital Twin.

In the last 5 years, interest in the Digital Twin has renewed as indicated by searching for the term on Google trends, illustrated in Figure 3.1. Additionally, a significant number of review papers on the Digital Twin have been published in the last few years. Due to the high number of reviews, I focus on the most recent ones, published in 2020 and 2021 and presented in Table 3.1.

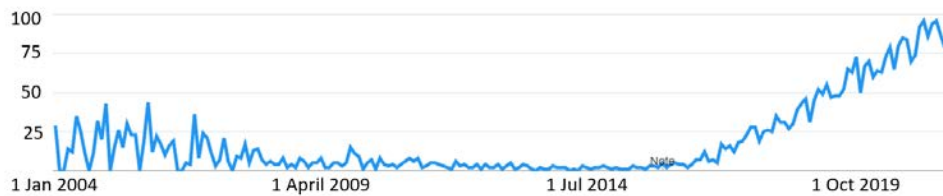


Figure 3.1: The Digital Twin on Google trends.

Digital Twin Review		
<i>Reference</i>	<i>Topic</i>	<i>Date</i>
<i>Journals</i>		
[82]	Production/Industry	2021
[83]	Aerospace/Industry/Robotics	2021
[84]	Urban Planning	2021
[85]	Construction	2021
[86]	General	2021
[87]	Material Science	2021
[88]	Manufacturing	2021
[89]	Computer Science	2021
[90]	Electric Vehicles	2021
[91]	Construction	2021
[92]	Manufacturing	2021
[93]	General	2020
[94]	General	2020
[95]	Maintenance	2020
[96]	Manufacturing	2020
[97]	Manufacturing	2020
<i>Conferences</i>		
[98]	Computer Science	2021
[99]	Aerospace	2020
[100]	Industry - Manufacturing	2020
[101]	General	2020
[102]	Aerospace	2020

Table 3.1: List of Digital Twin review papers by topic.

Initially, Digital Twin was treated as a high fidelity model or multidisciplinary simulation, and the real time component had not yet been under consideration [103]. However, the definition has evolved to encompass dynamic modelling and bidirectional communication and mapping to the physical system. In essence, the Digital Twin is based on the simple idea of linking a physical object with its digital counterpart accurately and in real time. However, a fit-all concept architecture has not been developed, as discussed in the recent review on the Digital Twin [93]. In particular, Xu et al. [104] defined Digital Twin to be a dynamic representation of physical entities with their functions, behaviours, and rules. Fig. 3.2 describes our conception of the Digital Twin. The framework consists of three components;

1. The digital model describes the physical object.
2. A knowledge base used to build the framework.
3. An analytics component used to assess its performance.

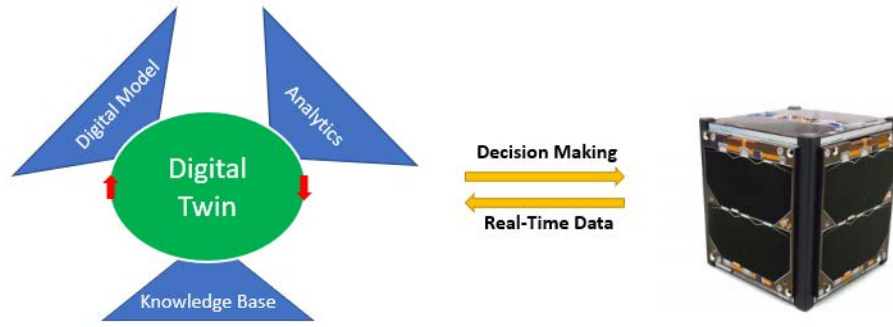


Figure 3.2: The Digital Twin's main components.

Using real time data, the system should communicate accurately, predict its state and react. Through our proposed framework, the dynamic Digital Twin will act as a living entity or a true representation of the system.

Jones et al. [93] recently reviewed the current state in the literature on the Digital Twin. They described the Digital Twin as a collection of themes depicted in Figure 3.3. Particularly, they described the Digital Twin as Physical (PE) and Virtual (VE) entities encompassing physical and virtual processes and operating in their respective physical and virtual environments. The models communicate through physical-to-virtual and virtual-to-physical connections. These connections have two main phases. The metrology, which refers to physical sensor readings measuring relating quantities of interests and the realisation, during which, the difference between the values of the virtual and physical models is calculated and the models are adjusted accordingly. This general process is called Twinning or Synchronization and happens at a Twinning Rate, depending on what is necessary for the application.

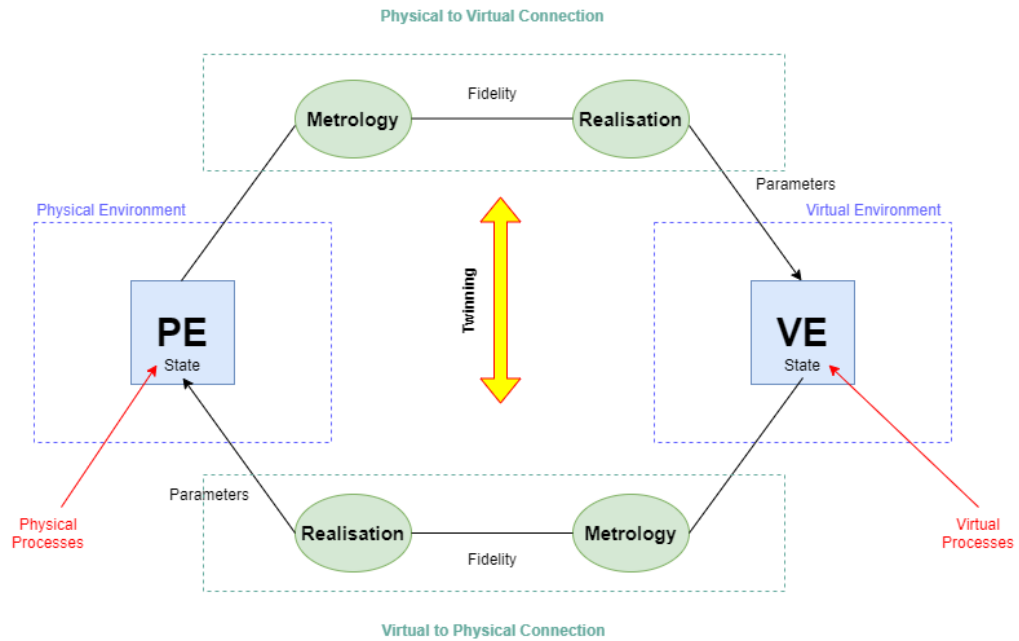


Figure 3.3: The Digital Twin diagram with the connections between the virtual and the physical entities.

An important aspect of the Digital Twin is fidelity. Most definitions treat the Digital Twin as a model of the highest fidelity e.g., "one definition with

fully mirroring its (physical twin) characteristics and functionalities”. No journal papers in our review discussed task specific fidelity.

Literature indicates two main approaches when building a Digital Twin architecture. A designer can either create a model based or physics driven model, of the physical entity or use sensor data to construct relationships between available information and quantities of interest.

3.2 Model Based Simulations

Until recently, model based simulations have been popular with designers and engineers. They follow the same traditional approach of observing a physical phenomenon, develop an understanding of it, usually through experimentation, and develop a mathematical model describing the observed behaviour to the best of our abilities. Naturally, assumptions and simplifications have to be made, both during the derivation of the mathematical model and during numerical implementation. There are two main approaches for the development of model based simulations, experimental modelling and numerical simulations.

- **Experimental Modelling** refers to conducting actual physical experiments with the goal of understanding the underlying processes and laws governing a physical phenomenon. It can generate high quality data but it can be expensive. Scaling down the model saves on costs.
- **Numerical Simulations** are used to solve the equation derived through experimental modelling. Model based numerical simulations are highly interpretable and tend to be less biased than their data driven counterparts [105]. On the downside, they are limited by computational power, uncertainties in the inputs and in the modelling, and are vulnerable to instabilities during numerical solving.

3.3 Data Driven Models

During the design phase, model based simulations are more popular as the amount of data available is limited. However, with the improvement in data management technologies, open source libraries and computation infrastructure, e.g., FPGAs, data driven modelling has become increasingly popular.

The way these models are usually build is by breaking the entire model to smaller parts with distinct functionalities. This is beneficial during root-cause analyses performed in health management studies. Subsequently, data resources are allocated to each division. These can be sensors, historical data or any other data source available. From that point on, traditional data processing techniques, methods, and algorithms are implemented. One of the most popular choices for data processing is Machine Learning (ML) and Artificial Intelligence (AI). ML models are generally categorized as supervised, unsupervised and reinforcement learning.

- **Supervised:** Focusing on identifying correlations between independent variables to dependent variables. The main tasks are regression and classification. Popular methods include, Linear Regression, Neural Networks and Support Vector Machines. Requires labelled data.
- **Unsupervised:** These methods use unlabelled data. They are popular for clustering tasks.

- **Reinforcement Learning:** In Reinforcement Learning, the model does not learn directly from the data, but an agent searches for an optimal policy to maximize cumulative gain.

3.4 Comparison

One important advantage of data driven approaches is that data contains information about both known and unknown physics. Thus data driven models can be seen as filling gaps in model based simulations and it can be said that they account for the full physics. Particularly, the full physics can be subdivided into three subcategories as seen in Fig. 3.4 [106]. The definition of these classifications are as follows.

- **Full Physics:** Physics that completely describe every aspect of the behaviour of the model.
- **Observed Physics:** Understood physics that might not completely describe the behaviour of the model.
- **Modelled Physics:** The governing equations devised to describe the observed physics.
- **Resolved Physics:** The implementation of the modelled physics limited by numerical methods and computation constraints.

Model based simulations are restricted to resolved physics only and alone may not reflect reality. The advantages and disadvantages are summarised in Table 3.2.

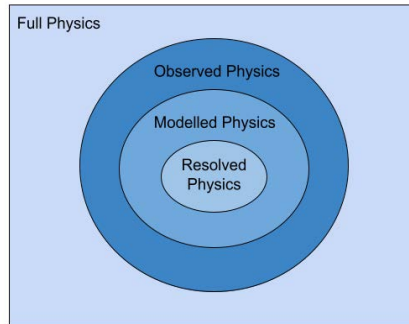


Figure 3.4: Hierarchy of physics models in simulations.

Approach Comparison		
-	<i>Model Based</i>	<i>Data Driven</i>
Data Reliance	Small Data	Big Data
Domain Reliance	Deep Domain Knowledge	Little Domain Knowledge
Fidelity/Robustness	Highly nonlinear and complex relations	Limited to data and model complexity
Adaptability	Complex and time consuming setup	Rapidly adapt to a specific problem
Interpretability	Consistent physical meaning	Physics agnostic surrogate

Table 3.2: Model based vs data driven approaches comparison.

3.5 Hybrid Approaches

Hybrid approaches refer to frameworks and methodologies leveraging both data driven models and model based simulations. This topic has been studied extensively in the previous years. Review papers [107, 108, 109, 110] considered the complementary effect of combining data driven and model based methods. Generally, the model based simulation is used to amend the limitations of the data driven approach such as, the lack of interpretability, without having to completely rely on them, either due to our limited computational capabilities or the lack of a complete physics model. However, all reviews agree that so far in the state of research, the methodologies proposed are tied to a particular field of study. Particularly, Rai and Sahu [108] classified data - physics model interactions in 3 categories.

- Approaches using data from the physics based model during preprocessing or as inputs to an ML model,
- architectures that insert data from the physics based model to the output of the ML model, and
- using physics informed loss functions that enforce in a deep learning model.

A large number of papers have combined data and physics models on various applications [111, 112, 113, 114, 115, 116], with positive results.

3.6 Combining Information

An important advantage of multifidelity modelling is its ability to combine information from different sources. In the context of spacecraft, I consider three sources of information, a) on-board sensor measurements, b) data driven surrogate models and c) model based simulations. As described in Section 2, either type of surrogate model comes with advantages and disadvantages. Multifidelity modelling allows for combining sources of data, and depending on the sources of choice, has different advantages and disadvantages. The data used can come from different sources. It can be collected either from previous missions of the same type of satellite or from ground tests. In the case of the latter, extra care must be taken to ensure that the data is as close to operating conditions as possible, e.g., using a thermal vacuum chamber for thermal tests. On the other hand, a

FE model can be used as a model based HFM or, if possible, live data being received through telemetry.

Depending on what is used as LFM and HFM, different information configurations can be distinguished. If historical data is used to train the neural network, then the best accuracy can be expected. The data contains information on unmodelled physics both relating to the satellite itself and the operating environment. However, this data is hard to obtain. Especially when considering the design and development of new systems, training with historical data is impossible. Ground data for training on the other hand are available but may not reflect the operating environment accurately, potentially missing environment related unmodelled physics. Moreover, concerning the HF model, a finite element simulator can be used. This can complement the shortcomings of data driven approaches e.g., extreme phenomena not included in the data used to train the ML model. However, live data received through telemetry can also be used to receive information on unexpected phenomena. There is however a major issue in the latter approach. Telemetry distorts the signal, both by adding noise and through missing values. Communication between spacecraft and ground control is a bottleneck for Digital Twin applications. While the sparsity of data is not an issue for the multifidelity framework [2], the framework depends on the accuracy of the HFM. These are summarised in Tables 3.3 and 3.4.

In order not to restrict the framework to any application, the choice of configuration is left to designers working on a particular subsystem. As discussed in Subsection 4.2, the simulation of subsystems can be, at least partially, decoupled. Therefore, different configuration can be chosen for each subsystem.

Data Combinations		
<i>HF — LF</i>	<i>Test(Ground)</i>	<i>Historical Data</i>
Finite Element Analysis	- Pot. Least Trustworthy + Data Avail. Independent	- Data Limited Availability + Trustworthy
Live Data(Telemetry)	- Noise + Trustworthy + Can be implemented on-board	- Noise - Limited Availability + Pot. Most Trustworthy + Can be implemented on-board

Table 3.3: Data combinations, advantages and disadvantages.

Information Combinations		
<i>Source</i>	<i>Test(Ground)</i>	<i>Historical Data</i>
Finite Element Analysis	HF - Solves Diff. Equations LF - Picks up interdependencies between features and temperature from ground test data.	HF - Solves Diff. Equations LF - Picks up interdependencies between features and temperature from historical data.
Live Data(Telemetry)	HF – Supplies data about the current situation in space. LF – Estimates a state based on tests. Testing condition might differ from the space environment.	HF – Checks if estimation is correct. If not, it corrects. LF – Estimates a state based on data from the space environment.

Table 3.4: What LFM and HFM information sources have to offer.

3.7 Digital Twin in Aerospace

The American Institute of Aeronautics and Astronautics (AIAA) and Aerospace Industries Association (AIA) published a report concerning their position on the Digital Twin [117]. In this report, the definition accepted the most by their respective members is

“A set of virtual information constructs that mimic the structure, context and behaviour of an individual / unique physical asset, or a group of physical assets, is dynamically updated with data from its physical twin throughout its life cycle and informs decisions that realize value. ”

Moreover, the report also lists what they consider as essential elements of a Digital Twin architecture, namely a virtual representation (model), a physical realization (asset) and an exchange of information between the two.

Digital Twin is popular in Prognostic and Health Management (PHM) applications, especially in industrial applications. However, there are some examples from the field of aerospace.

Kapteyn et al. [118] constructed a Digital Twin for the PHM of an Unmanned Aerial Vehicle (UAV). It was demonstrated that by using online data, a probabilistic classification problem can be solved. In particular, reduced order modelling (Proper Orthogonal Decomposition basis) was used to assess damage on the UAV and, through classification, pick the appropriate damage model from a pre-defined library of damaged components. However, the extreme cost of high fidelity models is a bottleneck for implementing a wide variety of damage models and the robustness of the method has not been studied yet.

Ye et al. [119] proposed a two stage Digital Twin framework to track the structure in real time, in order to reduce uncertainty. As in Kapteyn et al. [118], the offline stage is composed of a knowledge library and simulation models, and the online stage is composed of diagnoses, updates and evaluations. This framework can be applied in a variety of tasks.

Shangguan et al. [120] introduced a framework for fault diagnosis that at-

tempts to integrate both model based and data driven approaches. The resulting framework is more accurate and more reliable than a data driven only approach. Additionally, the model is more interpretable and easier to maintain. However, the framework is overly reliant on the high fidelity model for accuracy and not adequately comprehensive. Finally, more efficient machine learning methods can be used.

3.8 Reduced Order Modelling

Reduced Order Modelling (ROM) is a set of techniques employed to reduce computational complexity of mathematical models in numerical simulations. Proper Orthogonal Decomposition (POD) is a method for producing reduced order models that often appears in the literature as an approach to build the Digital Twin. POD follows these steps:

I Sample (snapshot) the vector field over a period of time.

$$U = \begin{pmatrix} u(x_1, t_1) & \dots & u(x_n, t_1) \\ \dots & & \dots \\ u(x_1, t_p) & \dots & u(x_n, t_p) \end{pmatrix} \quad (3.1)$$

where, \mathbf{n} are spatial elements, and \mathbf{p} are time samples.

II Compute the covariance matrix C ,

$$C = \frac{1}{(p-1)} U^T U. \quad (3.2)$$

III Compute the eigenvalues and eigenvectors of C .

IV Order eigenvalues and eigenvectors from the largest to the smallest.

V Computer matrix Φ ,

$$\phi = \begin{pmatrix} \phi_{1,1} & \dots & \phi_{1,n} \\ \dots & & \dots \\ \phi_{n,1} & \dots & \phi_{n,n} \end{pmatrix} \quad (3.3)$$

where, $\lambda_1 \dots \lambda_n$ are the eigenvalues, and ϕ are the eigenvectors. n is the number of obtained eigenvalues. Matrix Φ has $n \times n$ dimensions.

VI The decomposed vector $u(x, t)$ is expressed through a combination of deterministic spatial functions $\Phi_k(x)$, the eigenvectors, and random time coefficients $a_k(t)$:

$$u(x, t) = \sum_{k=1}^{\infty} a_k(t) \phi_k(x) \quad (3.4)$$

POD is popular because it is,

- application independent and
- reduces computational cost.

A disadvantage of the POD is that it is searching for the direction of maximum variance. The variance is represented by the eigenvalues, which contains information on how much variance exists in a particular direction (the eigenvector). The largest eigenvalue describes the maximum variance. In a lot of cases, the direction of maximum variance is clear. In Fig. 3.5 the direction is marked by a red line.

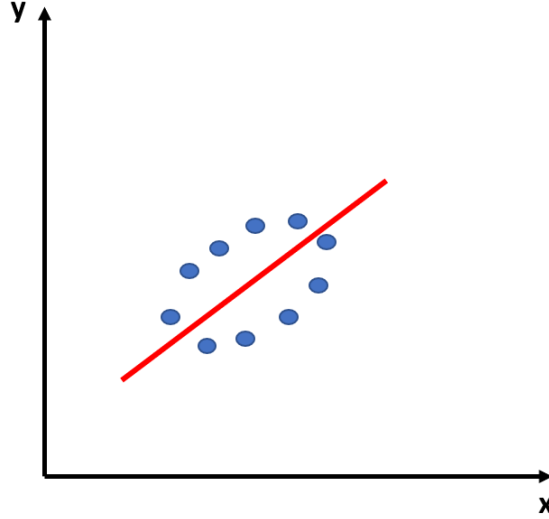


Figure 3.5: An example of the direction of maximum variance. - is the direction of maximum variance.

On the other hand, if the data are arranged in a circle-like pattern there is no clear direction of maximum variance as in Fig. 3.6. In this case the eigenvalues are close or equal.

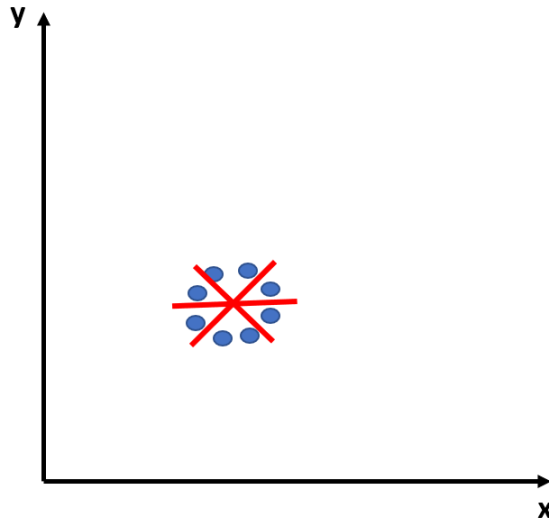


Figure 3.6: An example where the direction of maximum variance is not clear. - is the direction of maximum variance.

In cases where there is no clear direction of maximum variance the effectiveness of the POD drops. Examples of problematic eigenvalues appear in a variety of subsystems e.g., structural modelling [121], turbulence [122], fluid dynamics [123].

3.9 Research Gaps

Our research illustrated some gaps in the literature which can be summarised as

1. Research on the Digital Twin in aerospace is limited and mostly restricted to structures and aerodynamics. Particularly, research is even more limited when you consider only spacecraft.

The Digital Twin is discussed in Section 3. While a respectable number of papers have been written on the Digital Twin, they focus mostly on defining its characteristics. Additionally, the review papers authored in 2020 and 2021 summarizing the work of previous years, are application specific and heavily skewed towards industry and manufacturing. No reviews found focusing on aerospace were published in a journal.

2. Most applications of the Digital Twin are on Prognostics and Health Management (PHM), e.g., Fault Detection.

An additional conclusion that could be drawn from the literature in Section 3 is that if papers demonstrate an application, it is usually on PHM. While PHM is an important area of research, it does not necessitate a full model analysis, something implied by most definitions of the Digital Twin which state that it should act as a mirror of the physical model.

3. Reduced Order Modelling method is often used to speed up computation.

While no applications on spacecraft were found, papers with other areas of interest, e.g., drones in Section 3, relied on reduced order modelling to ease the computational load. Specifically, the basis used to construct the reduced order model was Proper Orthogonal Decomposition (POD). As described in this Section, POD has important drawbacks and cannot be used as a universal approach for a full analysis. The applications in the eligible papers did not require a full analysis. An estimate of the physical model's state was used to make decisions on how to act. E.g., picking a predetermined response procedure from a library.

4. Multifidelity approaches on the Digital Twin are limited.

While multifidelity modelling in general, and cokriging in particular, appears in the literature in the context of space applications, see Section 2, it is not discussed from the prism of the Digital Twin. It is used purely to decrease computational cost in offline simulations and more complex or online configurations are not considered. Additionally, no particular attention is given to the sources of information used for the low and higher fidelity models, with most applications relying on different discretizations of the same model to produce different fidelities.

5. No multi-subsystem implementation of Digital Twin in spacecraft.

No work was found that proposed a Digital Twin for an entire spacecraft. While some work in Section 3 does include satellites, it is focused on particular subsystems e.g., structures, missing a key element of the Digital Twin, multimodality, as outlined in Section 3. Subsystems in a satellite are interdependent, especially when considering the unmodelled physics. Information will be missing and the estimation may be lacking the accuracy required to ideally mirror the physical model.

Due to the inability of reduced order modelling to be used dependably for analysis, our aim is to introduce an alternative framework that could be used for analysis and assist designers and operators. Cross-communication between the digital and physical models should be included to function in a Digital Twin architecture, as well as the ability to detect interdependencies. The framework should use the predictive power of a universal approximator, such as an ANN, to include unmodelled physics, detect subsystem interdependency and to be applied on any type of system.

To apply the Digital Twin in space engineering, gap 1 needs to be addressed, meaning that further research is required. Additionally, it must be expanded to more spacecraft subsystems. On a similar note, gap 5 must be considered. While a lot of papers discuss the definition and characteristics of the Digital Twin, system-wide implementations on spacecraft are rare, both due to satellites being hard to access and the costs involved. Gap 2 indicates that there is a lot of interest in PHM and fault detection, however it should be expanded in other areas as well. I consider gaps 3 and 4 important since research should not be restricted to reduced order modelling due to its limitations and multifidelity modelling could provide an alternative approach that will enable building the Digital Twin in the future. Therefore, in this subsection, a framework that could potentially function with a Digital Twin architecture is proposed, addressing gaps 3 and 4. Moreover, I contribute to gap 1 by adding to the body of literature.

Chapter 4

Proposed Methodology

Our research investigates how different topics can be integrated to build a Digital Twin. This thesis is looking into live data, finite element analysis and multifidelity modelling to produce a potential general framework for small spacecraft. Broadly speaking, a Digital Twin simulation should constitute of an input vector of quantities of interest, usually through sensors, appropriately preprocessed, the main framework that produces an estimation and a reasoner that adjusts the framework, according to a set of criteria, to satisfy the requirements at the least possible cost. Figure 4.1 illustrates this general approach.

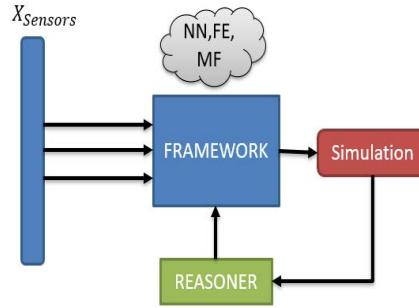


Figure 4.1: The general approach for a multifidelity framework.

4.1 Framework

In this framework, a multifidelity approach is used to take advantage of both the accuracy of high fidelity data and the availability of cheap low fidelity data. Both the low and high fidelity models independently produce estimations of the same quantity of interest. This work considers different sources of information are used for the LFM and the HFM. As in Fig. 4.2, the LFM is based on collected empirical data e.g., used in ANN training, and the HFM is based on the physical model and numerically solving equations.

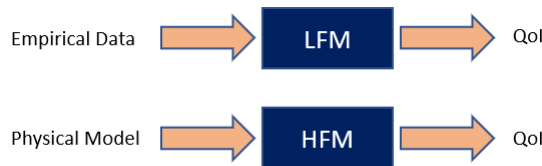


Figure 4.2: LFM and HFM and their inputs. QoI is the quantity of interest.

A HFM produces a high fidelity estimation describing a satellite's component state. In parallel, an ANN predicts, in real time, all quantities of interest, and is used as the LFM to speed up computation. The ANN should be trained with input-output pairs from all components. In complex systems, components are not insulated from each other but interact with each other often through unmodelled phenomena. The ANN will capture interdependencies missed by a physics based model.

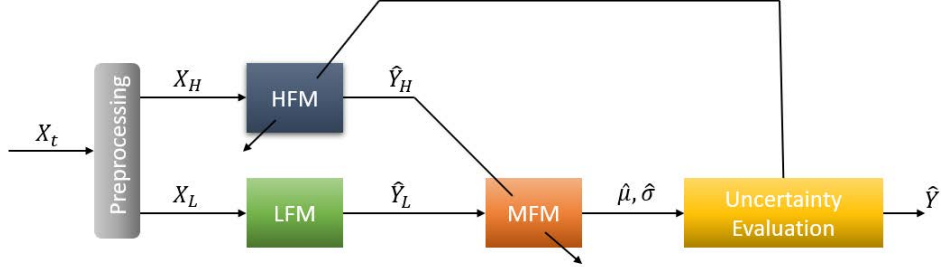


Figure 4.3: Schematic of the proposed multifidelity framework.

As illustrated in Fig. 4.1, X_t , which includes the current state and a pre-determined number of previous timesteps stored in memory, is fed as input to two separate systems. A distinction between X_L and X_H is drawn because since the LFM runs much faster than the HFM, each input is fed a sequence with a different sampling rate, with the rate of the LFM being much higher than the HFM's. Each system produces a set of predictions, particularly a high fidelity model and a low fidelity surrogate model, as two estimates for the next thermal state \hat{Y} . Subsequently, a meta predictor combines both predictions appropriately through a multifidelity scheme to produce a multifidelity surrogate model for X_{t+1} . X_{t+1} is estimated by the mean value μ :

$$\hat{\mu} \equiv \hat{X}_{t+1} \quad (4.1)$$

This surrogate will follow the LFM's sampling rate. The produced estimated mean $\hat{\mu}$ and standard deviation $\hat{\sigma}$ are compared with a desired value for uncertainty and fed back to the HFM. The equation estimating $\hat{\mu}$ is

$$\hat{\mu} = \rho Z_{\hat{Y}_L} + Z_{(\hat{Y}_H - \hat{Y}_L)} \quad (4.2)$$

where, ρ is a non deterministic scaling function, $Z_{\hat{Y}_L}$ is a Gaussian process fitted on the LFM estimation and $Z_{(\hat{Y}_H - \hat{Y}_L)}$ a Gaussian process fitted on the difference between the HFM estimation and the LFM estimation.

The time component is structured as follows. A window of time is defined as an appropriate number n of values. The exact number will directly affect the model's refresh time and is to be determined by mission requirements and computational power. LFM data points will be generated for the next n moments, requiring time t_n . At the same time, the HFM will generate as many values m , $m < n$, as possible in the time frame t_n . If real time values are used, then m values will be collected in that timeframe. The cokriging scheme is applied to this window, producing an estimate for the system's behavior for t_n time steps in the future. The process repeats continuously during the system's operation. The time needed for the system to refresh the model t_{ref} is

$$t_{ref} = t_{LFM} + t_{ck}, \quad (4.3)$$

where t_{LFM} is the time required to generate n data values with the LFM and T_{ck} is the time required to compute the cokriging scheme. The time window is shown in Fig. 4.4. Red marks indicate low fidelity estimations. These arrive at even intervals through the GRU, which can use past values at any time to produce a future estimation. Occasionally, a green high fidelity simulation data point arrives, these do not need to arrive uniformly and therefore $T_1 \neq T_2$. For a set amount of time, defined by t_{window} , both green and red values are collected. After a sufficient values have been collected then cokriging estimates future values. The process repeats for the next future window.

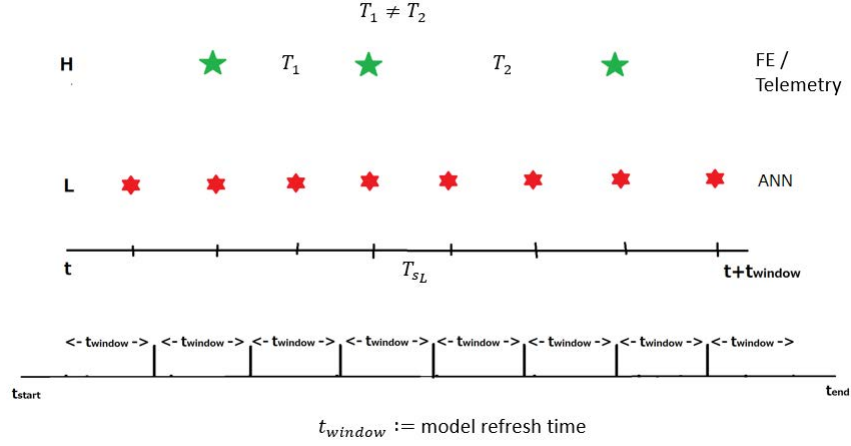


Figure 4.4: Timekeeping in the multifidelity framework. H, L are high and low fidelity data points, T_1 , T_2 are intervals between high fidelity data, T_{sL} is the sampling interval of the low fidelity model and t_{window} is the pre defined length of the update window.

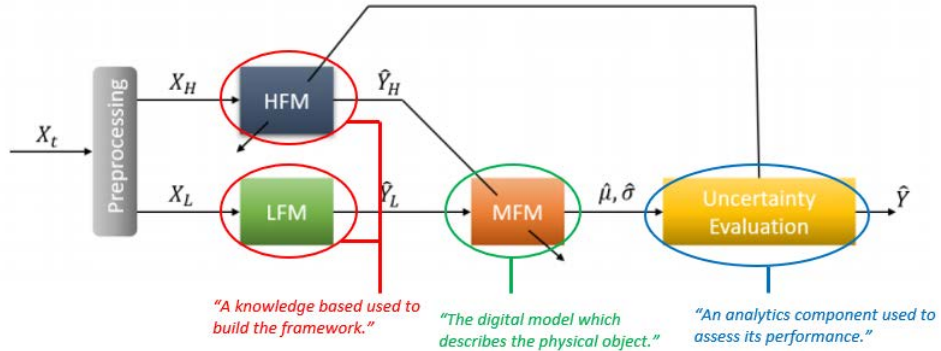


Figure 4.5: How the proposed framework fits the conception of the Digital Twin.

The proposed framework reflects the Digital Twin framework as described in Section 3. This is illustrated in Fig. 4.5. The MFM corresponds to the digital model. An established knowledge base is utilized to construct the low fidelity model. In this case the knowledge based refers to the empirical data and the our understanding of physics used to build the FE model. Analytics are performed to decide whether to increase the number of data taken into account by the framework.

4.2 Multimodality

An important aspect of a Digital Twin architecture is multimodality. In this case, multimodality refers to the interaction between different, distinct, subsystems within a single system. A multimodal framework must detect interdependencies between subsystems and generate an estimation that represents the entire system at a large degree.

The framework in Subsection 4.1 can work in a multimodal setting. The GRU can accept as inputs timeseries data from multiple components, detect interdependencies and produce an estimation. This is because in the hidden layers all inputs are combined, shown in Fig. 4.6, and all contribute together to the activation function.

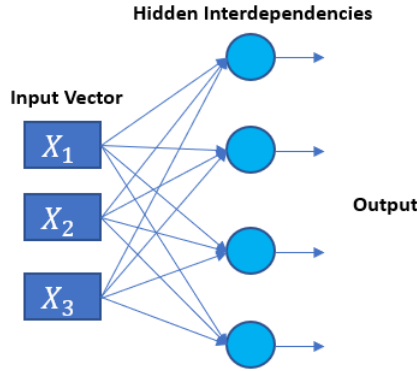


Figure 4.6: Interdependencies in the hidden layers.

For example, the output of a power component is expected to affect the temperature of a thermal component. Both the finite element simulator and the ANN support multiple outputs. The GRU's output is a vector of estimations for every quantity of interest. However, to keep things simple, cokriging for each individual component can be decoupled and be calculated separately. Implementing parallel computing techniques for this step can increase efficiency. By using key components and a moderate amount of data points, along with parallel processing, the computation cost remains manageable. This is illustrated in Figure 4.7 showing only power and thermal components for simplicity.

Advantages of this framework, include robustness and estimation of uncertainty, both important in space applications. In Kontaxoglou et al. [2] it was shown that cokriging is robust to high fidelity data placement, especially against a standard Gaussian process. Moreover, it was robust to noise, reducing its effects by using information from the other model, correcting and keeping the final estimation close to the desired curve.

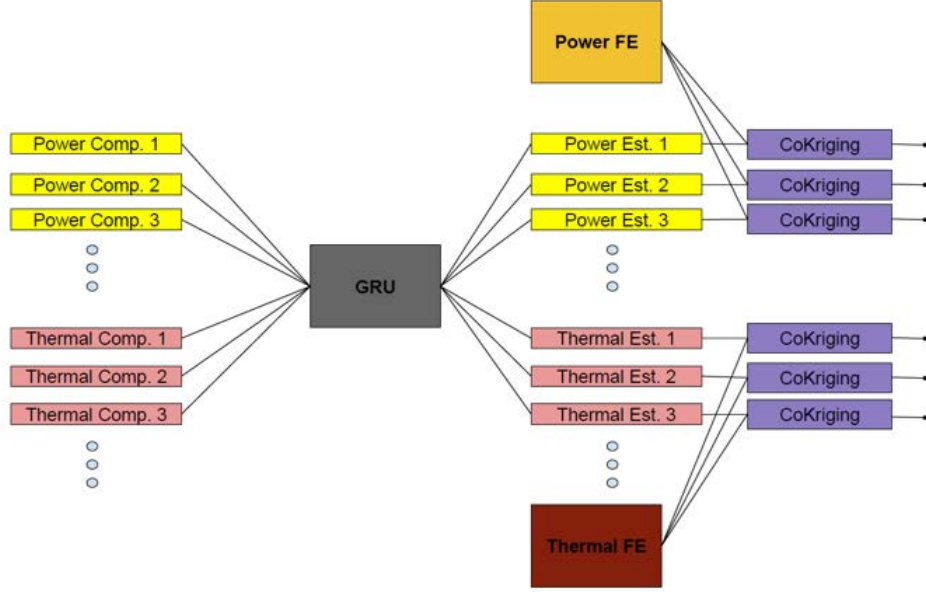


Figure 4.7: The multimodal framework configuration.

4.3 Unaccounted Interdependencies

During the operation of the system unexpected interdependencies appear. These interdependencies do not appear in the training data and therefore are impossible to be accounted for by data driven techniques such as the GRU. While the multifidelity framework is not a purely data driven technique, it heavily depends on the the GRU to detect all interdependencies. In the case of unaccounted interdependencies, the responsibility falls on the high fidelity model to correct the estimation. If the high fidelity model is a finite element or finite differences model then it is impossible for these interdependencies to be corrected. On the other hand, if a trustworthy sensor measurement is used as the high fidelity model, the true state of the satellite is reflected so any and all interdependencies are already included. Therefore, transforming the LFM to sensor-level accuracy should fix the issue.

Other than sensors, there is no other high fidelity model that can counter this source of error. When using this framework for operation however, communication with the spacecraft must be included as it is also one of the characteristic components of the Digital Twin. Using the information about the current state of the satellite and its subsystems, the low fidelity model can be corrected sufficiently to any unmodelled or unexpected disturbances.

4.4 Framework Assumptions

This subsection aims to discuss the assumptions the multifidelity framework requires to work, especially concerning the high fidelity points. An important assumption of cokriging is that

- Assumption 1: At locations where a high fidelity data exists, a low fidelity data point also exists.

This assumption is required since to estimate the high fidelity model, cokriging

requires the calculation of the difference between high and low fidelity models, to calculate this difference $d = Y_{HF} - Y_{LF}$, the estimations Y_{HF} and Y_{LF} must refer to the same QoI at the same moment in time.

- Assumption 2: High fidelity points are not clustered together.

While the distribution of high fidelity points does not have a large impact to the multifidelity framework as it is robust to high fidelity point placement and number, some minimas must be observed. For example, if the high fidelity points are sampled to be clustered together high fidelity information will not be available for most of the length of the experiment. Fig. 4.8 illustrates such clustering.

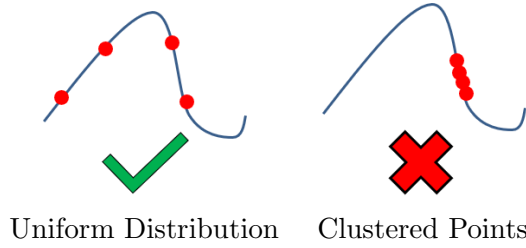


Figure 4.8: Example and counter example of high fidelity point distributions.

- Assumption 3: The accuracy of the high fidelity model is sufficient for the task at hand.

As cokriging transforms the low fidelity to the high fidelity model, it cannot exceed the accuracy of the high fidelity model.

4.5 Framework Application

This framework is intended to assist operators and engineers with their work on small satellites. The framework can be utilised in three main ways:

1. To be used to accelerate satellite development: Speed up computation when being run offline, using expensive and accurate simulator data, e.g., Thermal Desktop.
2. To be used for onboard prediction of satellite states: Using sensor data from satellite located in space through telemetry to act as HFM data while the framework works on Earth.
3. To reduce the computational load so that the calculation can be made on-board: Using sensor measurements and using on-board computers for the calculations, e.g., FPGAs.

Speeding up computation was the main idea driving the development of multifidelity modelling. With this framework, a designer can obtain high fidelity model accuracy at reduced computational times with the help of low fidelity data. This can speed up the development phase and reduce the resources required e.g., computer hardware. In this case the sliding window technique would not be used and the framework would be applied to the entirety of the available data. The HFM can be a FE/FD simulator or other computationally expensive methods.

An application scenario is using data provided by telemetry. This is especially important if the framework is to be incorporated in a Digital Twin architecture. On-board measurements, received by telemetry, can replace the HFM. The main drawback of telemetry is that data arrives sparsely. However, this is not an issue for the framework because it can update if and when new data arrives. Noise however remains an issue.

The framework is robust to moderate noise [2], but the overall accuracy is dependent on the accuracy of the HFM. In the case that telemetry noise is high, the framework can reduce the distortion, however the accuracy achieved might not be sufficient for a Digital Twin. Even in this case, a small number of quality measurements is enough to improve accuracy compared to using only LFM data and will sync the simulation with the true state of the physical entity.

Besides the development of another framework for satellite simulation, the main impact of our approach is to compliment efforts by other researchers to tackle the unreliability of telemetry data. This will enable fault detection systems on the ground to have accurate information on the satellite.

Considering on-board implementation, the sensor measurements will again act as HFM data. This eliminates a lot of telemetry issues like sparsity and noise. Looking at the necessary components, the GRU could be potentially run on-board using FPGAs. Recent advances on FPGAs demonstrate that it is possible to accelerate the LSTM using FPGA architectures [124, 125, 126]. Gaussian processes can also be implemented on FPGAs [127]. With future advances in FPGA technology, it can be expected that this framework can be fully implemented on-board to support decision making and PHM applications.

4.6 Robustness

Robustness is a fuzzy term used in everyday life with a multitude of definitions and metrics. These are usually very depended on the context, application or field of study. However, generally speaking, all definitions regardless of context are based on the same ideas or principles. A strict mathematical definition found in the seminal paper [128] defines robustness as

Definition 1 *A sequence of estimators $\{T_n\}$ is robust at a probability measure F if*

$$\forall \varepsilon > 0 \exists \delta > 0 \forall G \forall n : \\ \{\pi(F, G) < \delta \Rightarrow \pi(\mathcal{L}_F(T_n), \mathcal{L}_G(T_n)) < \varepsilon\}$$

(ii) *A sequence $\{T_n\}$ is robust in a neighborhood of F if there is an $\eta > 0$ such that $\pi(F, G) < \eta$ implies $\{T_n\}$ is robust at G . A sequence $\{T_n\}$ is robust at a class $\mathcal{E} \subset \mathcal{F}$ if it is robust at all $F \in \mathcal{E}$. A sequence $\{T_n\}$ is robust (everywhere) if it is robust at all $F \in \mathcal{F}$.*

According to [129], to calculate robustness, the performance of the system needs to be tested across a, preferably large, number of scenarios $S = s_1, s_2, \dots, s_n$ for a particular performance metric $f(\bullet)$ e.g., error. Additionally, when designers have to choose between multiple decision alternatives, x_i , the process must be repeated for each of these decision alternatives.

Different performance metrics focus on different aspects of performance. Some categories of robustness used are:

- *Expected value metrics* [130], express the expected performance over the tested scenarios.

- *Metrics of higher-order moments* [131], e.g., variance, skew .These express information concerning how the expected level of performance varies in the tested scenarios.
- *Regret-based metrics* [132]. In game theory, regret is defined as the difference between an option under consideration and the best possible option. This class of metrics can be used to reduce the maximum loss or to increase the minimum reward.
- *Satisfying metrics* [133]. After the designers decide on a threshold, satisfying metrics calculate which subset of scenarios have an acceptable performance according to the predefined requirements.

Thus, for direct comparisons, the performance of different decision alternatives, across different scenarios $f(x_i, S) = f(x_i, s_1), f(x_i, s_2), \dots, f(x_i, s_n)$ must be transformed to a single robustness value $R(x_i, S)$ per decision alternative. While different robustness metrics, focus on different aspects, a single framework for the calculation of different metrics must be defined. The transformation from $f(x_i, S)$ to $R(x_i, S)$ can be achieved through three transformations applied one after the other. Particularly, a performance value transformation (T1), scenario subset selection (T2), and robustness metric calculation (T3). This is demonstrated in Fig. 4.9. Taking a more detailed look at each of the three transformations:

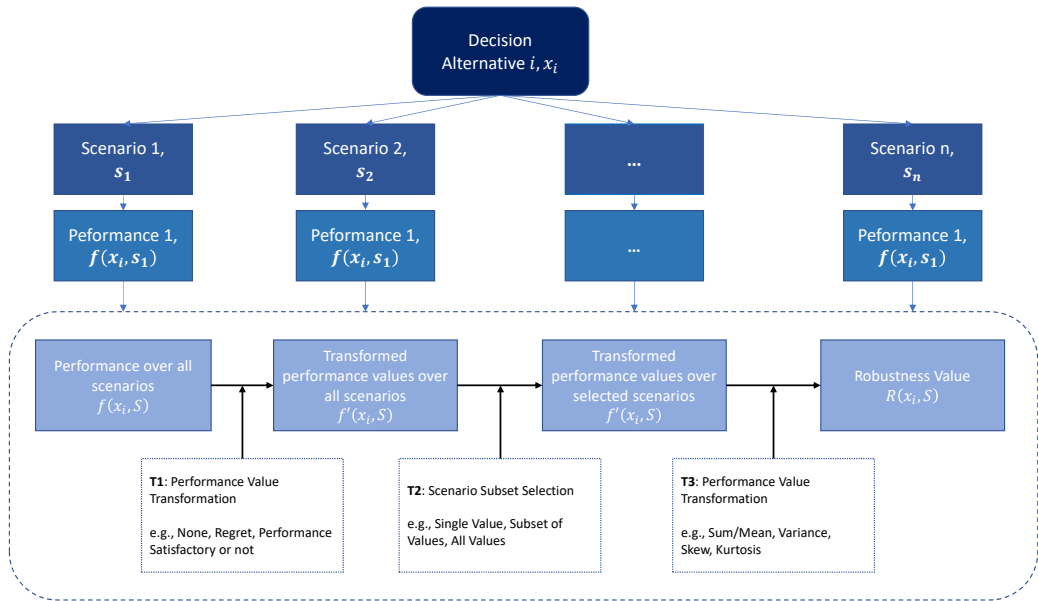


Figure 4.9: Process for extracting a robustness value.

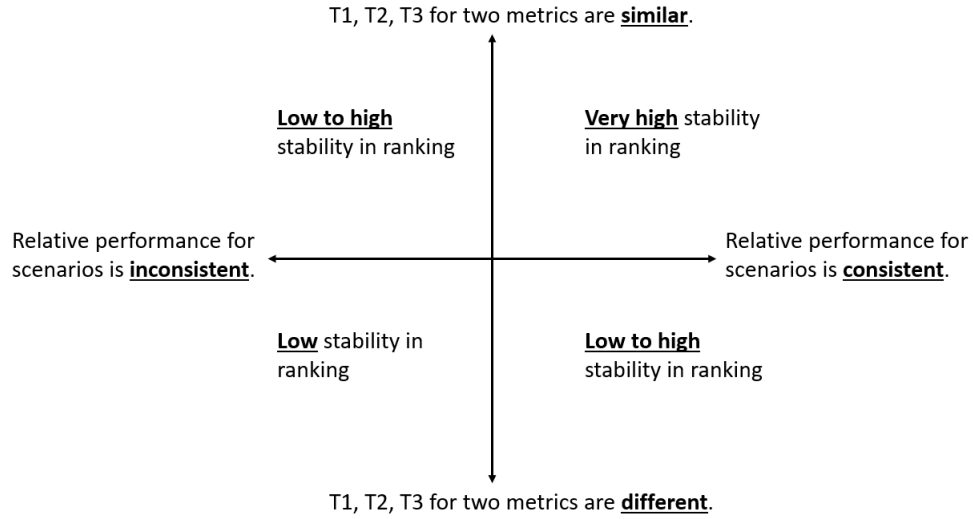


Figure 4.10: Conditions affecting ranking stability. A high stability in ranking indicates that two metrics will rank the decision alternatives the same, whereas a low stability indicates that two metrics will rank the decision alternatives differently.

- The performance value transformation (T1) transforms performance values of a particular metric $f(x_i, S)$ to the "type" of information $f'(x_i, S)$ that will be used for the calculation of the robustness metric $R(x_i, S)$. The choice is often between examining the actual performance of the system or whether the system performance is satisfactory.
- The scenario subset selection transformation (T2) determines which values of $f'(x_i, S)$ will be used when calculating the robustness value. The transformation can keep all scenarios, or a subset of scenarios $f'(x_i, S') \subseteq f'(x_i, S)$ or even a single scenario, e.g., the worse-case. The overall performance of the system will be evaluated on these scenarios only.
- The final transformation (T3) calculates the actual robustness value. It converts $f'(x_i, S')$ obtained through the system performance values (T1) for the selected scenarios (T2), to a single robustness value, $R(x_i, S)$, e.g., by taking the expected value.

Risk Aversion

There are many metrics that measure performance in the literature, but they do so in a different manner. It is therefore difficult to assess how robust the actual performance of a system. One way to categorize these metrics is through their risk aversion.

Risk is a vague term, dependent on the context of the application. Taking for example, the case where the model is tested for the robustness of the error in a thermal simulation. Risk aversion in this case looks to avoid the worst case in which the error will be the highest. If keeping the errors low is very important then a very risk averse method of robustness calculation should be picked. For instance, if there are temperature sensitive components and the temperature must remain under a threshold, then the error should not be too high in order to have

precision in your temperature calculations and operate the system safely or closer to its limits.

The metrics reflect different aspects of robustness and not the overall robustness of a decision alternative. On top of that it is particularly difficult to determine if one alternative is better than the other since their relative ranking might be different depending on what metric it was based on. A list of metrics starting from the least to the most risk averse is the following:

1. Maximax - least risk averse
2. Hruwicz optimism-pessimism rule
3. Percentile-based skewness
4. Percentile-based peakedness
5. Laplace's principle of insufficient reason
6. Mean-variance
7. Starr's domain criterion
8. Undesirable deviations
9. 90th percentile minimax regret
10. Minimax regret
11. Maximin - most risk averse

Their respective transformations T1, T2 and T3, along with their original reference, can be found in Table 4.1.

Robustness Metrics Disagreement

After calculating two robustness values for two decision alternatives x_1 and x_2 , notated as $(R_a(x_1, S), R_a(x_2, S))$ and $(R_b(x_1, S), R_b(x_2, S))$, the decisions must be ranked. The relative ranking of two decision alternatives will be the same, or stable, if the following three conditions are true:

$$R_a(x_1) > R_a(x_2) \text{ and } R_b(x_1) > R_b(x_2) \quad (4.4)$$

or

$$R_a(x_1) < R_a(x_2) \text{ and } R_b(x_1) < R_b(x_2) \quad (4.5)$$

or

$$R_a(x_1) = R_a(x_2) \text{ and } R_b(x_1) = R_b(x_2) \quad (4.6)$$

The relative rankings will be different if the following two conditions are true:

$$R_a(x_1) > R_a(x_2) \text{ and } R_b(x_1) < R_b(x_2) \quad (4.7)$$

or

$$R_a(x_1) < R_a(x_2) \text{ and } R_b(x_1) > R_b(x_2) \quad (4.8)$$

If the transformations that led to the calculation of R_a and R_b are similar, then the performance of the two decision alternatives is consistent across all scenarios. In this case the ranking stability is considered very high. "Similarity" in this context is not a strictly defined term. For example, if R_a and R_b use minimax

regret and 90th percentile minimax regret, there is a high degree of similarity in the transformation T1, subset selection T2 and the robustness calculation T3 because both metrics use regret and only a single scenario is used and therefore there is no need to combine any values for the final robustness calculation T3. On top of that, those metrics are also consistent as they are both looking at the worst scenario and at the 90th percentile value which almost reflects the worst scenario. We can expect those two metrics to agree and the ranking they produce to be stable.

A counter example would a set of R_a and R_b corresponding to minimax regret and percentile based peakedness. For these two, transformations T1, T2 and T3 are quite different. For minimax regret, performance values are transformed to regret, consisting of one, worse-case, scenario. However for percentile based peakedness, the actual values are used and consist of four scenarios. The final robustness value is the kurtosis of the four regret values. It is apparent that these metrics have fundamental differences that make a direct comparison difficult. Additionally, these metrics are also inconsistent since one looks at the worse case scenario while the other looks at four scenarios spanning across the distribution of S . In this case the relative ranking stability is "low". Fig. 4.10 illustrates this relationship between similarity and consistency.

Scenarios				
<i>Metric</i>	<i>Original reference</i>	<i>T1: Performance value transformation</i>	<i>T2: Scenario subset selection</i>	<i>T3: Robustness metric calculation</i>
Maximin	[130]	Identity	Worst-case	Identity
Maximax	[130]	Identity	Best-case	Identity
Hurwicz optimism-pessimism rule	[134]	Identity	Worst-and Best-case scenarios	Weighted mean
Laplace's principle of insufficient reason	[135]	Identity	All	Mean
Minimax regret	[132, 136]	Regret from best decision alternative	Worst-case	Identity
90th percentile minimax regret	[132]	Regret from best decision alternative	90th percentile	Identity
Mean-variance	[137]	Identity	All	Mean-variance
Undesirable deviations	[131]	Regret from median performance	Worst-half	Sum
Percentile-based skewness	[138, 131]	Identity	10th, 50th, and 90th percentiles	Skew
Percentile-based peakedness	[138, 131]	Identity	10th, 25th, 75th and 90th percentiles 0	Kurtosis
Starr's domain criterion	[139, 140]	Satisfaction of constraints	All	Mean

Table 4.1: Metrics and their T_1 , T_2 and T_3 transformations.

The authors in [136] performed extensive tests on three environmental scenarios to assess which metrics are similar and consistent enough to agree. Three combinations that give stable results are

- (a) Laplace's principle of insufficient reason & Mean-Variance, which describe the average actual performance across scenarios.
- (b) Maximin & Laplace's principle of insufficient reason, which describe the worst case scenario.
- (c) Maximax & Laplace's principle of insufficient reason, which describe the best case scenario.

Chapter 5

Experiments & Simulations

This section contains a modified version of a conference paper titled "Use of cokriging for thermal analysis in small satellites" published in the 2022 Aerospace Conference of the IEEE [2].

In this section simulation results using Python will be presented. The framework described in Section 4 will be tested with three sets of data, based on two satellites ISSL6U-1 and SDS-4. The results will be compared with other traditional techniques for errors and computational complexity.

5.1 Experiment Introduction

5.1.1 Disclaimer

All the data used in this research belong to the Japan Aerospace Exploration Agency (JAXA) and are classified. Only approximate models of the satellites and a general description of the data will be presented.

5.1.2 PC Specifications

The laptop used for all simulations has the following technical characteristics.

- Processor: Intel Core i7 10750H 2.60 GHz
- RAM: 32 GB DDR4
- Graphics Card: NVIDIA GTX 1660 Ti 6 GB

5.1.3 Traditional Techniques

The framework will be benchmarked against three other methods, usually employed by designers to solve similar problems. All three have been described in Section 2.

Gaussian Processes

Gaussian processes are the single fidelity counterpart of cokriging. The comparison serves to illustrate the benefits of introducing both high and low fidelity data, despite the computational cost increasing.

Gated Recurrent Unit

Neural networks, and in particular recurrent neural networks, are a staple of system modelling due to their low cost and good results. However, in order to

have high accuracy, they need to be trained with a high number of data which may not be available. In multifidelity modelling, high accuracy by the neural network is not absolutely required as the estimation will be corrected by the high fidelity data. LSTMs have consistently shown good performance with timeseries data, like temperature which can be expressed as $T(t)$. The Gated Recurrent Unit (GRU) is a simplified, and faster, version of the LSTM.

ARIMAX

Autoregressive methods use past values to statistically forecast future values. For example, using previous temperature measurements, the future trends can be estimated. I investigate if they can accurately predict satellite temperatures and how they perform against the multifidelity framework. Concerning the choice of autoregressive method see Section 2.

5.1.4 Error and Time Metrics Definition

Three errors are measured, mean squared error (MSE), mean absolute percentage error (MAPE) and mean absolute error (MAE). These are defined as

- MSE:

$$e_{MSE} = \frac{1}{n} \sum_{i=1}^n (x_i - y_i)^2 \quad (5.1)$$

- MAPE:

$$e_{MAPE} = \frac{1}{n} \sum_{i=1}^n \left| \frac{x_i - y_i}{y_i} \right| \quad (5.2)$$

- MAE:

$$e_{MAE} = \frac{1}{n} \sum_{i=1}^n |x_i - y_i| \quad (5.3)$$

The average errors across components, are defined as

$$\bar{e} = \frac{\sum_{k=1}^n e_k}{n} \quad (5.4)$$

where e is the error metric, between the tested framework and the HFM, and n the number of components, is presented. Additionally, as a general metric for standard deviation the average can be used defined as

$$\bar{\sigma} = \frac{\sum_{k=1}^n \frac{\sum_{l=1}^{n_p} \sigma_{k,l}}{n_p}}{n} \quad (5.5)$$

where $\sigma_{k,l}$ is the standard deviation at a particular point, for a particular component and n_p the number of points. It is averaged across all points in a prediction and across all components. However, this metric is limited since it does not reflect the distribution of the standard deviation and even if uncertainty is low overall, locally it might be prohibitively high.

In order to quantify the change in the magnitude of an error between two methods, I calculate the difference as a percentage as:

$$e.c. = \frac{e_{method_1} - e_{method_2}}{e_{method_2}} \cdot 100\% \quad (5.6)$$

where $e.c.$ is the error change, e_{method_1} is the error estimated by method 1 and e_{method_2} is the error estimated by method 2. A negative value indicates the percentage the error was decreased and a positive value the percentage it was increased. For example, if $e_{method_1} = 0.2443$ and $e_{method_2} = 2.0344$ then the error change is

$$e.c. = \frac{0.2443 - 2.0344}{2.0344} \cdot 100\% \approx -88\%, \quad (5.7)$$

and therefore can be said that *"The error of method 2 was reduced by 88% by method 1."*

Time to compute is estimated from the point the first input enters the framework to the point when an estimation is output. Moreover, the confidence interval for predictions is defined as

$$C.I. = 2 \cdot \sigma \quad (5.8)$$

where σ is the standard deviation.

5.1.5 The Sliding Window

Window Sliding Technique is a computational technique that reduces the use of nested loop and replaces it with a single loop. In our terms, the sliding window is used to utilize a window of past values to predict a set of future values. This way, computational time is allocated at each repetition to generate the data used by the multifidelity framework. Concerning the framework, the following process will be followed to predict the thermal and power values for an entire orbit, one window length at the time.

1. Define 1 slot of width w .
2. Use n previous slots to estimate the next slot. (3 previous slots is a good rule of thumb.)
3. Slide the window, calculate the estimation, repeat.

A large window with the width of n times the slot of width w will move forward in time, evaluating w future values per repetition. Width w of the window is equivalent to the update rate. Smaller width implies that new values will be calculated more often. However, w should not be chosen to be too small as

- (a) Information may be lost when the total number of values used for the framework are too few.
- (b) Some subsystems like thermal evolve slowly over time. Updating too often confers no benefits.

Fig. 5.1 illustrates how the sliding window moves along 1 orbit or 1 period T .

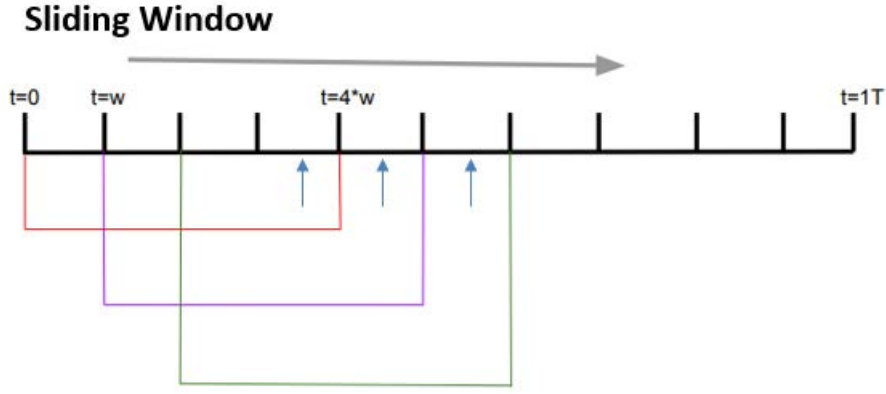


Figure 5.1: The sliding window. w is the width of the window and T is one period.

To assess the performance of the algorithm, the result per window could be compared with the actual values for that window. However it would be more useful to compare an entire orbit, generated step by step by sliding the window. In this way the errors and standard deviation obtained will reflect the realistic numbers received by other subsystems.

5.2 ISSL6U-1

ISSL6U-1 is a 6U satellite intended to orbit at LEO. The project provides a fast validation platform to test optical sensors. Among other components there are 3 telecommunication components, components 1-3, and 3 Printed Circuit Boards(PCBs), components 4-6. A simplified ISSL6U-1 satellite layout is described in Figure 5.2.

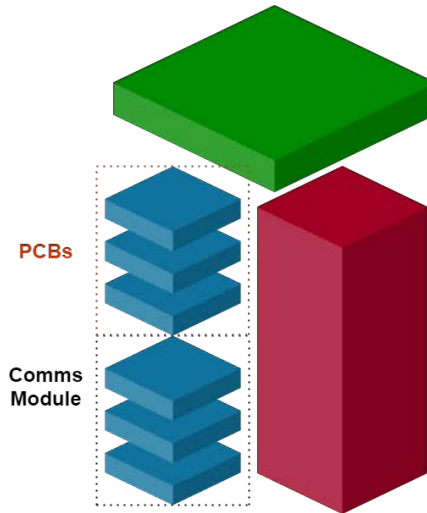


Figure 5.2: Simplified ISSL6U-1 layout.

The communication components have a large heat path and exchange a substantial amount of heat with the satellite panel. Also, components 2 and 3 are screwed together, and exchange some heat. The electrical boards are also screwed to the satellite panel, but do not exchange much heat compared to the communication components. The electrical boards are not thermally connected to each

other directly. The power consumption, in the form of heat, for each component is summarised:

- Component 1: 15W during data downlink
- Component 2: 8W during data downlink
- Component 3: 1.44W always
- Component 4: No heat consumption
- Component 5: 3.3W during data downlink
- Component 6: 1.09W always

5.2.1 Thermal Environment

A small spinning satellite in orbit around the Earth is subject to multiple sources of thermal radiation. As a result, the satellite's behavior can be severely affected, especially in the areas of orbital evolution, attitude and instrumentation. The most relevant thermal loads for our study are in the low earth environment. As illustrated in Fig. 5.3, a satellite in LEO is subject to several radiation sources.

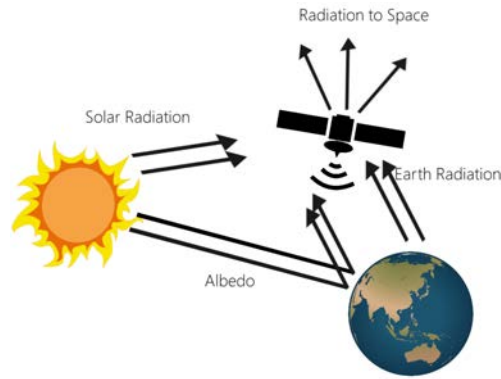


Figure 5.3: Sources of radiation for a satellite.

The thermal forces acting on satellites are generated when the incoming radiation from a thermal radiation source, e.g. the Sun, heats in a larger degree the surface facing it, compared to the dark side of the satellite. The uneven distribution of incident radiation results in asymmetric temperature distribution on the surface of the satellite. Consequently, the balance in photon re-emission is compromised, with the hot side losing more momentum than the cool side. The effect of this anisotropic behavior is observed as a total linear momentum, creating a recoil force on the satellite. This force is time-variate, due to the varying amount of incidence radiation from the source to the surface, which depends on coordinates controlled by the orbital characteristics of the satellite (spin axis, rotational motion, orbital motion). In the relative system of the satellite the thermal distribution is described by asymmetries along the spin axis (summer-winter effect) and along the equatorial direction (day-night effect) [141]. Furthermore, sudden heating and cooling on the satellite's surface may create temperature gradients and thus bending moments due to thermal stresses. These deformations in the structure affect the energy efficiency and the reliability of the satellite. Over the last decades, instrument failures are thought to have been caused by excessive thermal deformation. In a study on a 3D CubeSat, it was shown that

the thermal deformation of the satellite structure, which was in orbit of 450 km high and angle $B = 90^\circ$, caused a deviation of about 0.03° from the normals to the opposite small sides of the satellite [142]. Such a deviation is commensurate with the required satellite pointing accuracy on the order of 0.1° necessary for laser communication.

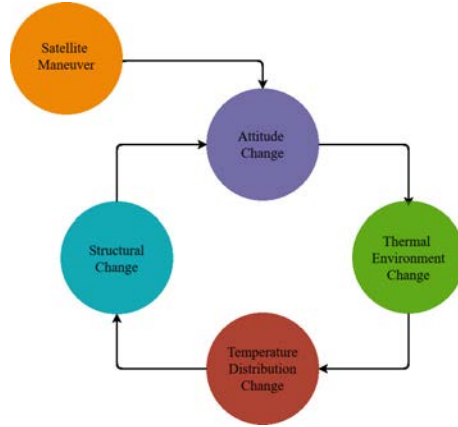


Figure 5.4: Thermal effects on satellites.

A spinning satellite, having no control capabilities, cannot correct for external forces influencing its behaviour. In the context of thermal effects, as illustrated in Fig. 5.4, a maneuver performed by the spacecraft would inevitably change its attitude, changing the thermal environment. In turn, the temperature distribution also changes, altering the structure through stresses and strains along the surface. The additional Summer-Winter and Day-Night effects will further distort the satellite's attitude and repeat the vicious cycle.

Downlink is the communication originating from the satellite and towards the ground. Communication in the opposite direction is called an uplink. When the communication components are activated for the downlink, heat is released from their electronics, increasing temperature temporarily. Subsequently, the components cool off slowly while the satellite moves along its orbit.

At this point the concept of the beta angle must be introduced. Beta angle is the angle between the orbital plane of the satellite and the vector to the Sun, effectively expressing the percentage of time a satellite in LEO spends in direct sunlight. It is given by

$$\beta = \sin^{-1}[\cos(\Gamma) \sin(\Omega) \sin(i) - \sin(\Gamma) \cos(\epsilon) \cos(\Omega) \sin(i) + \sin(\Gamma) \sin(\epsilon) \cos(i)] \quad (5.9)$$

where Γ is the Ecliptic True Solar Longitude, Ω is the Right Ascension of Ascending Node, i is the orbit's inclination, and ϵ is the Obliquity of the Ecliptic (approximately 23.45 degrees for Earth at present [143]).

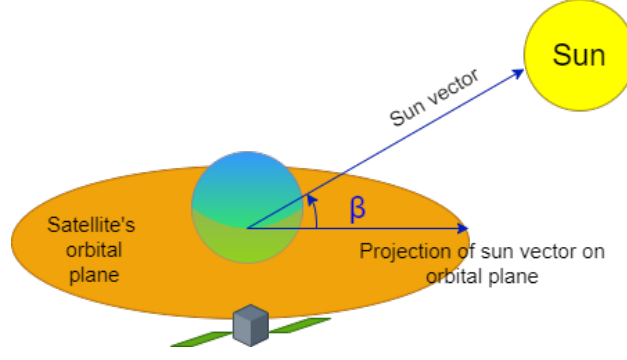


Figure 5.5: The beta angle β .

Additionally, when unloading, the satellite is releasing the momentum of the reaction wheel, using the magnetic torquer. The whole power consumption of the satellite is slightly higher than 3-axis control. In summary, affecting the power consumption, and therefore the thermal state of the satellite, along with their respective impact, are

- Data Downlink: Very High Impact,
- Sun Exposure: High Impact,
- Unloading: Low Impact,
- 3-Axis Control: Minimum Impact.

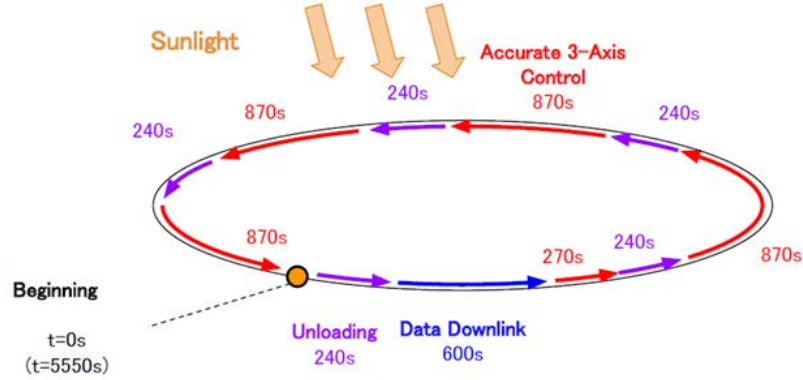


Figure 5.6: An example orbit with data downlink.

An example orbit is illustrated in Figure 5.6. Similar orbits are used to simulate data.

5.2.2 Scenarios

Our simulations will use data generated through Thermal Desktop. Our training data consists of orbits, simulated with different orbital altitudes and at different beta angles using Thermal Desktop. Additionally, the downlink scenario is initiated at different points in the orbit, for different beta angles and different orbit altitudes. The generated scenarios and their characteristics are summarised in Table 5.1.

The data is timeseries $T_i(t)$, $1 \leq i \leq 6$ describing the evolution over time of the 6 communication components and PCBs. The input vector is shown in Table

5.2. Each cell contains the temperature for the 3 communication components (T_1 - T_3) and 3 PCBs (T_4 - T_6) of Fig. 5.2.

Scenarios			
<i>Scenario Name</i>	<i>Beta Angle [°]</i>	<i>Orbit Altitude [km]</i>	<i>Data Downlink Start Time [s]</i>
Cold	0	400	0
Hot	72	400	240
A	30	400	1200
B	50	500	600
C	10	400	2400
D	70	400	4200
E	0	500	900
F	70	400	2100
G	0	500	1800
H	45	400	2700

Table 5.1: Generated scenarios for framework training and testing.

ISSL6U-1 Data Vector						
$t(s)$	$T_1(C^\circ)$	$T_2(C^\circ)$	$T_3(C^\circ)$	$T_4(C^\circ)$	$T_5(C^\circ)$	$T_6(C^\circ)$

Table 5.2: ISSL6U-1 data timeseries. Each cell contains the temperature for an ISSL6U-1 component.

5.2.3 GRU Training

In the work by Reis et al [73], a RNN is shown to have good performance for real time prediction of satellite thermal data. This work uses a subclass of RNNs, a GRU trained with Thermal Desktop data. The data covers orbits with multiple beta angles, including the orbits with minimum and maximum temperatures, and with downlink taking place at different points in time. As expected, the GRU shows good performance, approximating the curves well with a small error. For a lot of applications this performance might be satisfactory, however, this work investigates possible improvements and at what cost they come. The GRU is used as a base model for benchmarking.

Data generated by SINDA and transformed to resemble noisy sensor data were used to train the GRU. Particularly, the training process that was followed is:

1. Initialize the weights randomly.
2. Input the feature variables in the input layer.
3. Propagate the values through the network.
4. Compare the output layer with the Thermal Desktop data.
5. The error is backpropagated and the weights are adjusted appropriately.
6. After n_e number of epochs validation is performed.

7. If the validation error is higher than a pre-determined tolerance the training process is restarted, otherwise the training process is terminated. It is also terminated if an upper number of epochs is exceeded.

The GRU is comprised of an input layer, 3 hidden layers, with 50 nodes each and a Dropout regularization of 0.2, and an output layer. The optimizer used was Adam with parameters, learning rate $lr = 0.001$, $\beta_1 = 0.9$, $\beta_2 = 0.999$ and $\epsilon = 1e - 07$ and the mean squared error as the loss. The network was trained with 500 epochs. Scenarios Cold, Hot, A, B, C, D and E are used for training.

Validation

Validation is employed to avoid the problem of overfitting. Every n_e epochs, a scenario not used in training is tested to ensure that the ANN retains its ability to generalize. If the validation error is lower than previously then the weights are stored. Scenarios F and G are used for validation.

The Data

Fig. 5.7 represents the way data was used in training. Each scenario contains $N = 100$ rows of temperatures sampled at even intervals. \mathbf{T} represents a vector of temperatures for a particular moment in time, in this case the vector's dimensions is 1×6 , a temperature for each component of ISSL6U-1. Scenarios are split for training and validation.

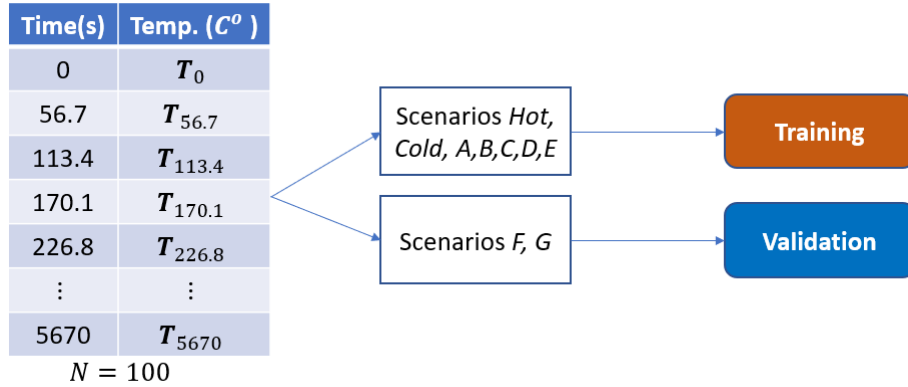


Figure 5.7: Training approach summary for ISSL6U-1 data. \mathbf{T} : vector of temperatures, N : number of rows.

5.2.4 Multifidelity Framework Testing

Scenario H is used for testing. Particularly, the GRU - Cokriging framework will be benchmarked against a) a purely high fidelity simulation using thermal desktop, b) the output of a GRU trained with noisy thermal desktop data and c) a GP fitted on high fidelity data points. The comparison will focus on time and accuracy. The multifidelity network is implemented with a GRU and a finite element simulator as a LFM and HFM respectively. It is compared to two traditional regression techniques for timeseries, a standalone GRU and a GP fitted on high fidelity data generated by the finite element simulator. This work aims to achieve accuracy close to that of the HFM but at decreased computational cost through the utilization of a LFM to produce estimations for the thermal state

of a 6U satellite. A set of simulations were conducted to assess the performance of a multi-fidelity framework based on cokriging, using a GRU as a LFM and a finite element simulator for thermal analysis, Thermal Desktop, as a HFM. The estimation is compared for computation time and accuracy against ANNs and Gaussian processes. For the HFM both noisy and noiseless data points from Thermal Desktop is used to assess robustness.

Three comparisons will be conducted depending on the number and the quality of the high-fidelity data, particularly using:

- 6 HFM data points,
- 16 HFM data points,
- 16 HFM data points with uniform noise and random missing values.

In Tables 5.3, 5.4 and 5.5 the results are summarised. The graphs for 6 HFM points can be seen in Figure 5.11, for 16 HFM points in Figure 5.12 and for 16 noisy and sparse HFM data the results can be seen separately for each component in Fig. 5.13 and 5.14.

Fig 5.8 summarizes how data is used to test the multifidelity framework. Testing data, $N = 100$ points in time from scenario H, is used as input to the GRU to produce a low fidelity estimation. Simultaneously, N_s random points are sampled from the testing data to be used as HFM. Cokriging produces multifidelity estimations for the 100 points.

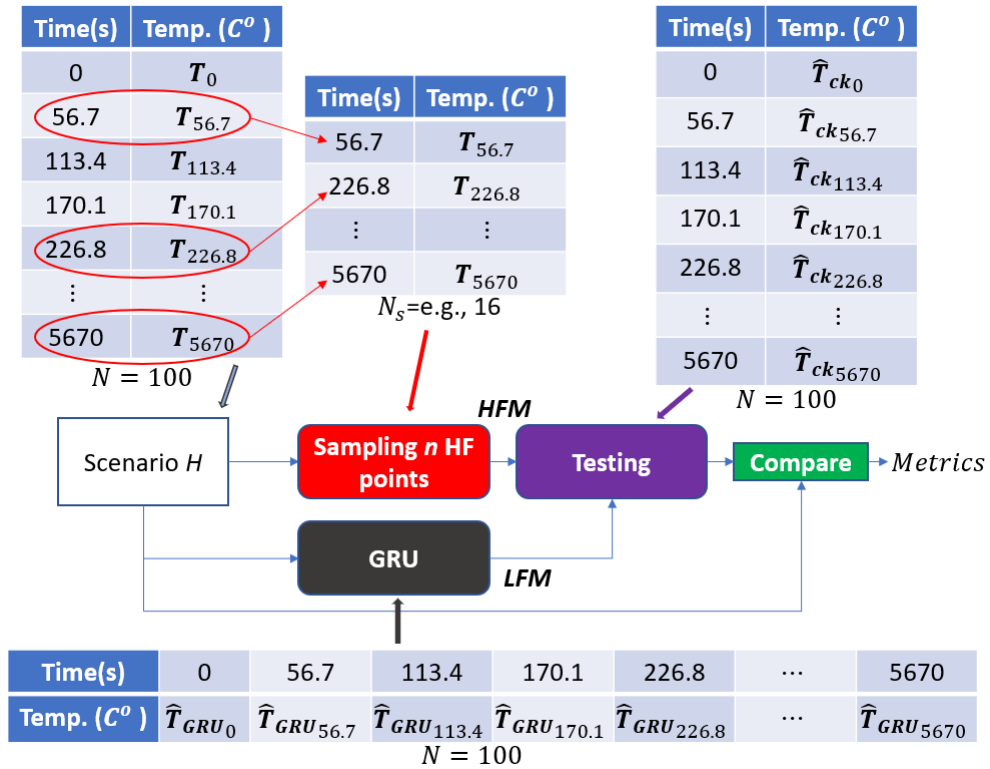


Figure 5.8: Testing approach summary for ISSL6U-1 data. \mathbf{T} : vector of temperatures, $\hat{\mathbf{T}}_{GRU}$: vector of GRU estimated temperatures, $\hat{\mathbf{T}}_{ck}$: vector of cokriging estimated temperatures, N : number of rows.

GP/ARIMAX

The testing strategy for the Gaussian process and ARIMAX skips the GRU and uses the sampled high fidelity points to estimate values as in Fig. 5.9.

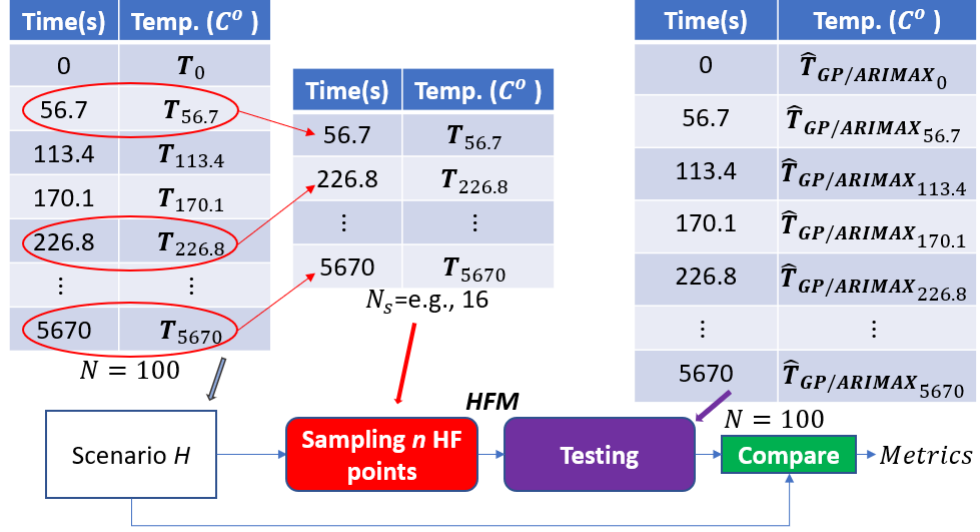


Figure 5.9: Testing approach summary for ISSL6U-1 data for GP and ARIMAX. \mathbf{T} : vector of temperatures, $\hat{\mathbf{T}}_{\text{GRU}}$: vector of GRU estimated temperatures, $\hat{\mathbf{T}}_{\text{ck}}$: vector of cokriging estimated temperatures, N : number of rows.

GRU

In Fig. 5.10 the result of the GRU for the 6 communication components and PCBs is demonstrated.

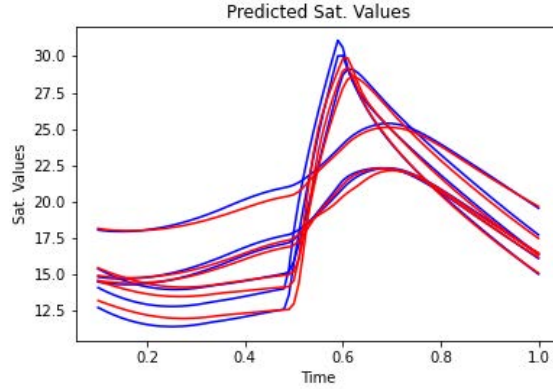


Figure 5.10: GRU results for ISSL6U-1 data. Red lines (-) indicate the predicted temperatures in the y-axis in C° and blue lines (-) indicate the real temperatures for the 6 components. Time indicates percentage of the orbit with 1 being a complete orbit.

The error and time metrics of the GRU output are used in Tables 5.11, 5.12, 5.13, 5.14 for comparisons against the multifidelity framework and Gaussian processes.

5.2.5 Uniform HFM Data Points

In the following simulations, 6 and 16 high fidelity points were used. No noise was added and are considered ground truth. Additionally, the points are evenly spaced along the orbit. The graphs for the 6 communication components and PCBs for 6 points are found in Fig. 5.11 and for 16 high fidelity points in Fig. 5.12.

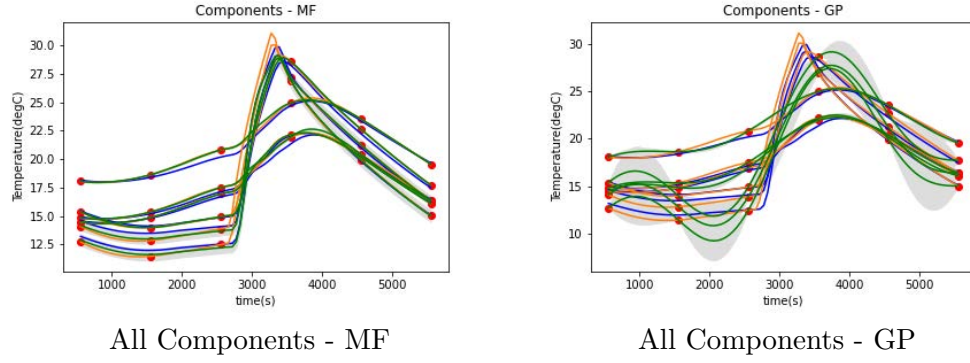


Figure 5.11: Using 6 noiseless and uniformly placed HFM points for ISSL6U-1 data. Blue lines (-) indicate the LFM, orange lines (-) the HFM, green lines (-) the multifidelity estimation and red dots (•) the HFM data points used and the gray area (■) the confidence interval for the 6 components.

6 HF Points - Noiseless			
<i>Metrics</i>	<i>MF</i>	<i>GRU</i>	<i>GP</i>
\bar{e}_{MSE}	0.2443	0.294	2.0344
\bar{e}_{MAPE}	0.009	0.0173	0.0548
\bar{e}_{MAE}	0.1748	0.3105	0.973
$\bar{\sigma}$	0.0894	n/a	0.3838
time(s)	5.8225	3.0332	3.0853
\bar{e}_{MSE} change	n/a	-53.7444 %	-90.8392 %
\bar{e}_{MAPE} change	n/a	-54.3159 %	-82.6199 %
\bar{e}_{MAE} change	n/a	-50.5476 %	-80.7603 %
time increase	n/a	192 %	189 %

Table 5.3: Error & time metrics. 6 HF points - noiseless and uniform.

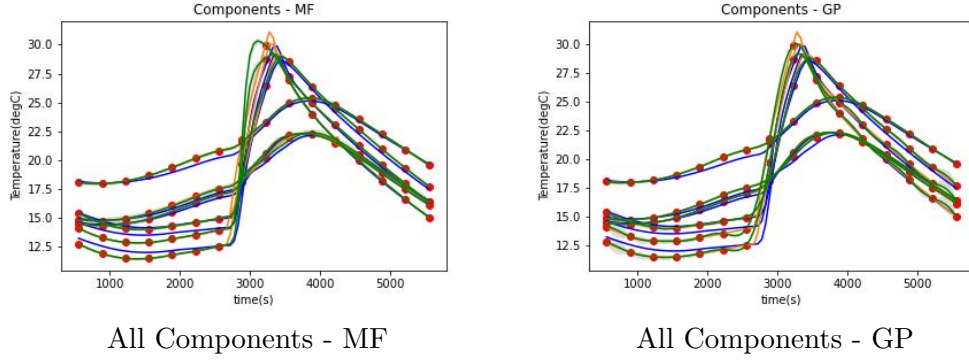


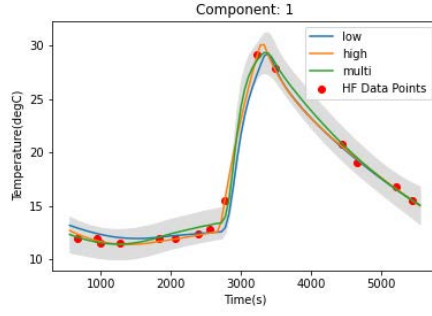
Figure 5.12: Using 16 noiseless and uniformly placed HFM points for ISSL6U-1 data. Blue lines (-) indicate the LFM, orange lines (-) the HFM, green lines (-) the multifidelity estimation and red dots (•) the HFM data points used and the gray area (■) the confidence interval for the 6 components.

16 HF Points - Noiseless			
<i>Metrics</i>	<i>MF</i>	<i>GRU</i>	<i>GP</i>
\bar{e}_{MSE}	0.2334	0.294	0.0374
\bar{e}_{MAPE}	0.0055	0.0173	0.0044
\bar{e}_{MAE}	0.1189	0.3105	0.0826
$\bar{\sigma}$	0.0508	n/a	0.0321
time(s)	6.7249	3.0332	3.2836
\bar{e}_{MSE} change	n/a	-60.5296 %	695.3114 %
\bar{e}_{MAPE} change	n/a	-71.6379 %	89.4774 %
\bar{e}_{MAE} change	n/a	-66.7395 %	108.1508 %
time increase	n/a	208 %	205 %

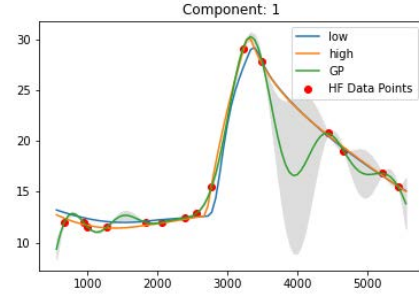
Table 5.4: Error & time metrics. 16 HF points - noiseless and uniform.

5.2.6 Random and Noisy HFM Data Points

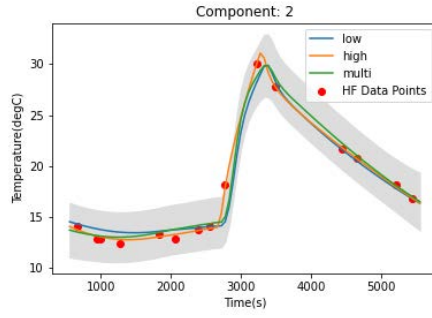
In the following simulations, 16 high fidelity points were used. Uniform noise, up to 10% of the current value, was added to the high fidelity data points to erode their accuracy so they don't reflect the ground truth exactly. Additionally, the points are not evenly spaced along the orbit but are chosen at random. This aims to model missing values and provide insight on the robustness of the framework against Gaussian processes. In this case, the graphs for each component are presented separately for a better demonstration. The graphs for the 16 high fidelity points are found in Fig. 5.13, 5.14 and the results in Table 5.5.



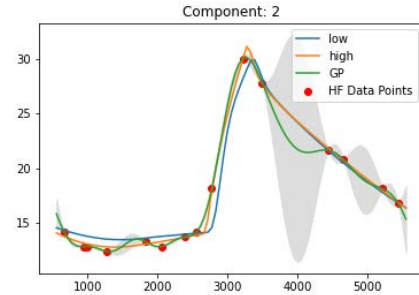
MF - Component 1



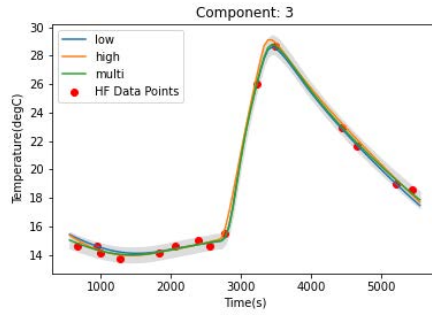
GP - Component 1



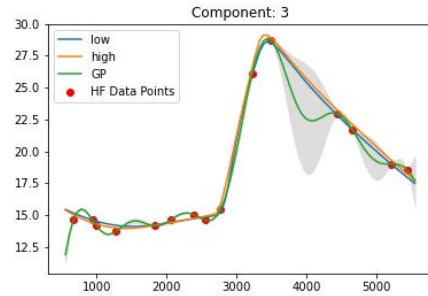
MF - Component 2



GP - Component 2

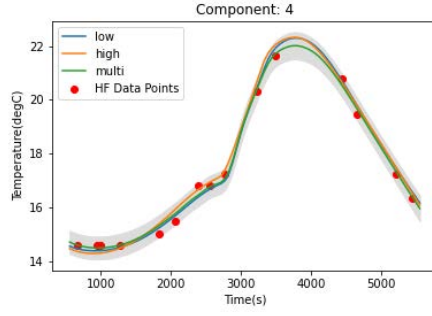


MF - Component 3

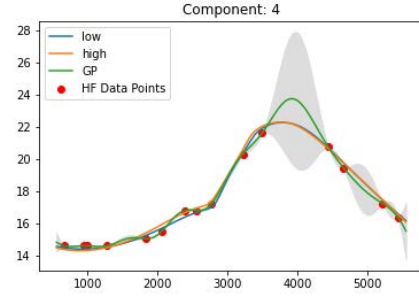


GP - Component 3

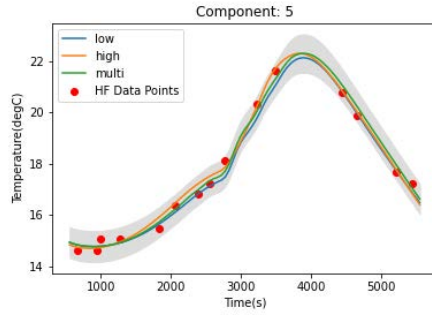
Figure 5.13: Using 16 noisy and randomly placed HFM points for ISSL6U-1 data. Blue lines (-) indicate the LFM, orange lines (-) the HFM, green lines (-) the multifidelity estimation and red dots (•) the HFM data points used and the gray area (■) the confidence interval for components 1-3.



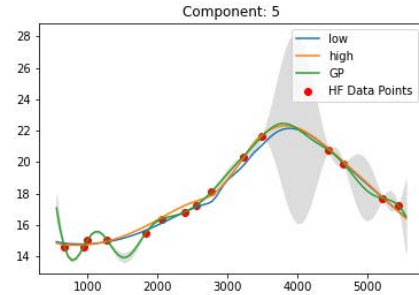
MF - Component 4



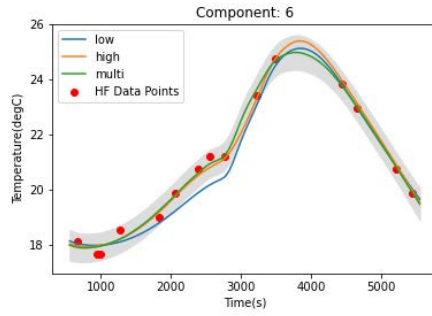
GP - Component 4



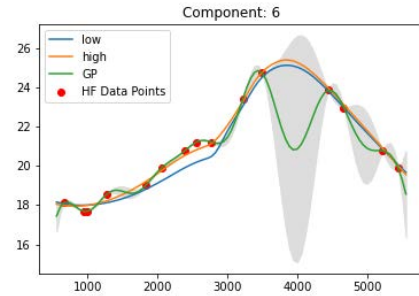
MF - Component 5



GP - Component 5



MF - Component 6



GP - Component 6

Figure 5.14: Using 16 noisy and randomly placed HFM points for ISS6U-1 data. Blue lines (-) indicate the LFM, orange lines (-) the HFM, green lines (-) the multifidelity estimation and red dots (•) the HFM data points used and the gray area (■) the confidence interval for components 4-6.

16 HF Points - Noisy - Sparse			
<i>Metrics</i>	<i>MF</i>	<i>GRU</i>	<i>GP</i>
\bar{e}_{MSE}	0.1781	0.294	1.4203
\bar{e}_{MAPE}	0.0143	0.0173	0.0336
\bar{e}_{MAE}	0.2677	0.3105	0.6617
$\bar{\sigma}$	0.3212	n/a	0.4703
time(s)	7.0702	3.0332	3.1260
\bar{e}_{MSE} change	n/a	-27.4030 %	-81.3163 %
\bar{e}_{MAPE} change	n/a	-13.3312 %	-55.9412 %
\bar{e}_{MAE} change	n/a	-9.4036 %	-56.6584 %
time increase	n/a	211 %	209 %

Table 5.5: Error & time metrics. 16 HF points - noisy and random.

5.2.7 Experiment Summary

GRU Comparison

Concerning the comparison with the GRU, in terms of accuracy, the multi-fidelity framework, with only 6 HF points, performed well at over a 50% reduction across error metrics in comparison to the GRU. With 16 HFM data points, as expected, since the GRU is independent of the HFM, the multi-fidelity framework performs even better with 60%-61% error reduction. At the final setup with 16 noisy and random values as HFM, the multi-fidelity framework still outperforms the GRU in accuracy, however, the effect is less pronounced at 9%-27%. The GRU's computational time is shorter in all scenarios.

GP Comparison

The main advantage of the MFM against a Gaussian process fitted on the high fidelity data points is that it can provide good approximations with a limited number of data points. This advantage diminishes when using more data points. However, Gaussian processes are faster than cokriging, requiring approximately only a third of the run time.

In terms of accuracy, the multi-fidelity framework, with only 6 HF points, performed well at over a 80%-90% reduction across error metrics in comparison to a Gaussian process. If the simulation is run with 16 HF points, without noise, the accuracy of the multi-fidelity is lower than the GPs. However, both produce comparably low errors.

Even though the multi-fidelity framework seemingly requires about twice the computation time than the GP, the GP requires three times the high-fidelity points the multi-fidelity does. The computational bottleneck in applications is the finite element simulator. To generate three times more high-fidelity points in half the time for the GP, the nodes, or accuracy, of the finite element model will have to be reduced in order to keep up, sacrificing fidelity. On the other hand, the multi-fidelity framework allows for a higher number of nodes, needing fewer points and allowing for more time to be computed.

In realistic applications however, if a real time component is to be implemented, there is no guarantee that the values will come at even intervals, or that some will not be missing all together. Additionally, the values might be noisy due to e.g., telemetry. In this case, the Gaussian process results in high uncertainty

and larger errors around the area of the missing value. On the other hand, the multi-fidelity framework does not suffer from occasional missing values because it is using the LFM as baseline and whenever high fidelity information is lacking, it tends to follow the shape of the LFM. This behaviour can be observed in Fig. 5.13, 5.14 and Table 5.5. Using 16 data points, distorted by noise and random values, the multi-fidelity framework outperforms the Gaussian process between 55%-81% in error reduction. To the cost of the MFM must also be taken into account the time required for training and fine tuning, which can take tens of hours to conclude, depending on the amount of data available.

Overall, the performance of Gaussian processes is comparable to the multi-fidelity framework. The approach of choice should depend on the application and whether it focuses on computational speed and simplicity against accuracy. The main advantage of the multi-fidelity framework is the robustness of its performance.

Concerning potential applications, the multifidelity framework can straightforwardly be used as a tool to speed up simulations during the design phase compared to a strictly finite element simulator. It is not restricted to thermal modelling, but can be used with any system that has multiple sources of information describing the same quantity or models of fidelity. When considering applications like small satellites, even though ideally data from previous missions or vacuum tests could be used they are hard to obtain beforehand. However, before vacuum tests are conducted, the framework can be used to supply thermal information to preliminary tests in atmosphere in which thermal sensor readings don't reflect reality. This can provide an estimate, e.g., on how long can components be operated without overheating. Also, operating in space, satellite state values received through telemetry are distorted and/or sparse. The multi-fidelity framework is robust to missing values and moderately noisy high fidelity data points. This enables the framework to use values directly from the physical system as a high fidelity estimation of the quantity of interest and combine it with a real time estimation generated by the neural network. Moreover, this allows for near real-time simulation since cokriging allows for the multi-fidelity estimation to be updated as high fidelity values become available. This will be the object of our future study.

To sum up, since GRU and Gaussian processes are simpler techniques, they have faster computation times. On the other hand, the multi-fidelity framework adds moderate to significant accuracy gains, as well as robustness. These traits are desirable for a high accuracy Digital Twin simulation. While computation times are approximately double than the other two regression techniques, they remain low enough to be implemented in a near real-time framework.

5.3 Multimodality Evaluation Using ISSL6U-1 with Simulated Power Data

As the data from ISSL6U-1 contain only temperatures, there are no interdependencies with other subsystems modelled. This makes the data unusable for demonstrating multimodality. To demonstrate multimodality we need to augment the available ISSL6U-1 data with simulated data describing a separate, fundamentally different, system that has interdependencies with the temperature. An excellent candidate are power components. Data describing power components are introduced in the form of a timeseries $P(t)$.

ISSL6U-1 and Power Data Vector									
$t(s)$	$T_1(C^\circ)$	$T_2(C^\circ)$	$T_3(C^\circ)$	$T_4(C^\circ)$	$T_5(C^\circ)$	$T_6(C^\circ)$	$P_1(W)$	$P_2(W)$	$P_3(W)$

Table 5.6: ISSL6U-1 and power data timeseries. Cells contain a temperature value for each component and three power measurements.

5.3.1 Simulating Power Components

Power components interact with temperature in 3 major ways.

1. As power output increases, components increase in temperature.
2. As temperature rises battery voltage drops.
3. As temperature rises solar panel output decreases.

When components are activated power from the batteries is demanded. This leads to an increase in temperature with the battery as a heat source. It is expected that a rise in power should be reflected with a near-simultaneous rise in nearby components' temperatures.

Batteries experience a decrease in voltage as their temperature rises. For a lead-acid battery the relationship between voltage/cell and temperature is given in Fig. 5.15. ISSL6U-1 data temperatures range between $8 - 35^\circ$. If the graph is treated as almost linear and if we assume that at 8° we have 100% of the voltage then at 35° we have 97.37%. Then the relationship between temperature T and voltage V can be expressed as a percentage

$$V = -0.1547 \cdot T + 102.32 \quad (5.10)$$

It can be expressed as power P through the equation

$$P = \frac{V^2}{R} \quad (5.11)$$

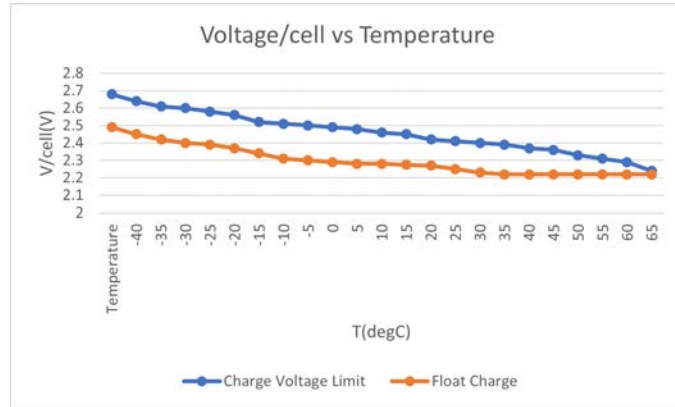


Figure 5.15: Voltage per cell - temperature for acid-lead batteries [144].

Concerning the solar panels, the efficiency factor is dependent on the temperature through the equation

$$\eta = \frac{P_{max}}{H \cdot A} = \eta_r - \mu(T_c - T_r) \quad (5.12)$$

where η is the efficiency factor, P_{max} is the maximum power output, H and A are the radiation intensity and solar panel surface respectively. Additionally, T_r

and η_r are the reference temperature and the efficiency factor at the reference temperature. μ is a coefficient measured in $\frac{1}{^\circ\text{C}}$ and T_c is the cell temperature. According to the literature, for a serial combination of panels $mu = 0.007\%$, $T_r = 25^\circ$ and $\eta_r = 0.125$ [145]. If H and A are considered fixed then the maximum power can be given as

$$P_{max} = 0.125 - 0.007/100(T_c - 25) = 0.12675 - 7e^{-5} \cdot T_c \quad (5.13)$$

Taking into account the three effects we can construct a relationship between temperature and power in the form

$$\vec{P}_a = F(\vec{P}, \vec{T}) \quad (5.14)$$

where \vec{P}_a is the vector of adjusted power output, \vec{P} is the vector of nominal power components and \vec{T} the temperature of the dependent components.

The Battery Arrays

A battery has three modes of operation:

1. Discharge, defined as the mode where power is released.
2. Float, defined as the mode in which the voltage is maintained.
3. Recharge, defined as the current / time required to return the capacity removed for the previous discharge.

There are two methods to calculate the power supplied by a battery,

1. The voltage method, $P = \frac{V^2}{R}$ and
2. the current method, $P = I^2 \cdot R$

Three battery assemblies were introduced to the data, supplying power required by the communication components and other mechanical components in the following configurations:

Configuration 1:

- (a) Supplying 50W for 15min at 20°.
- (b) 3 parallel strings each comprising 4 x 12V monoblocs; i.e. 24 cells.
- (c) Float voltage 2.27Vpc
- (d) Nominal capacity for each string 11Ah
- (e) Monoblock resistance 3.8mΩ

Configuration 2:

- (a) Supplying 50W for 15min at 20°.
- (b) 3 parallel strings each comprising 8 x 12V monoblocs; i.e. 48 cells.
- (c) Float voltage 2.27 Vpc
- (d) Nominal capacity for each string 11Ah
- (e) Monoblock resistance 3.8mΩ

Configuration 3:

- (a) Supplying $20W$ for $15min$ at 20° .
- (b) 2 parallel strings each comprising $4 \times 12V$ monoblocs; i.e. 24 cells.
- (c) Float voltage $2.27V_{pc}$
- (d) Nominal capacity for each string $11Ah$
- (e) Monoblock resistance $3.8m\Omega$

Using the appropriate current the power in Watts is calculated and then adjusted using corresponding temperatures so vectors of interdependent data is created. Those simulated numbers are not meant to reflect the exact reality of the situation but instead to introduce realistic power-temperature (P-T) relationships so the ability of the framework to detect interdependencies in data can be demonstrated. Fig. 5.16 demonstrates an example orbit that will be used for testing.

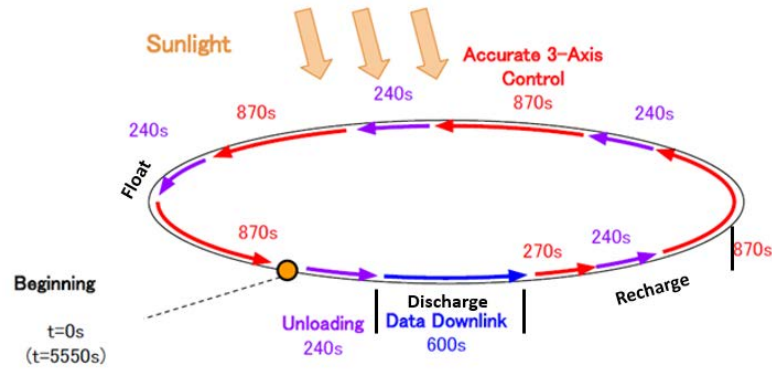


Figure 5.16: An example orbit including the battery states.

5.3.2 Multifidelity Framework Testing with Simulated Power Data

The training of the GRU was conducted with exactly the same process as for the ISS6U-1 data in Subsection 5.2. The difference in this subsection is that simulated power data is included in the input vector as in Fig. 5.17.

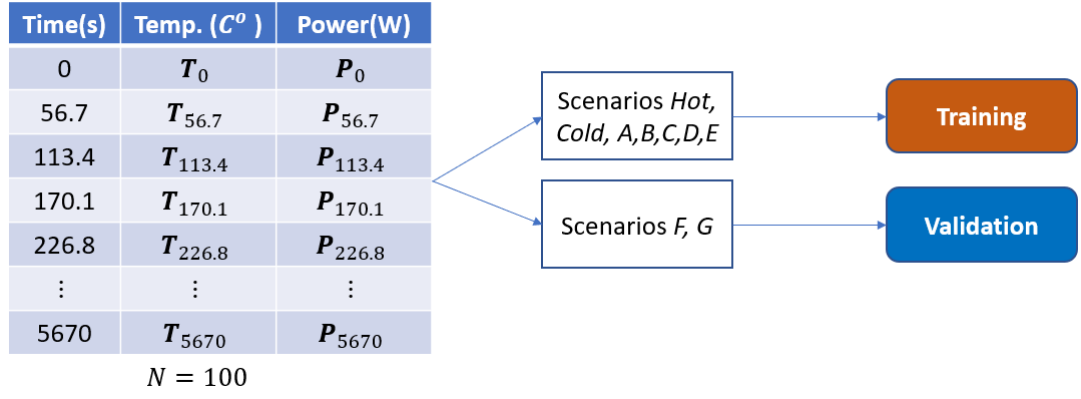


Figure 5.17: Training approach summary for ISSL6U-1 data with power data. \mathbf{T} : vector of temperatures, \mathbf{P} : vector of power measurements, N : number of rows.

In Fig. 5.18, the way data is used in the framework is summarised. This time, $N = 100$ rows of both temperature and power measurements are inputs to the GRU in order to detect interdependencies between them. 16 random values from both the temperature and power data are sampled to act as the HFM. Cokriging is applied independently to each component for the final estimation vectors in Fig. 5.19.

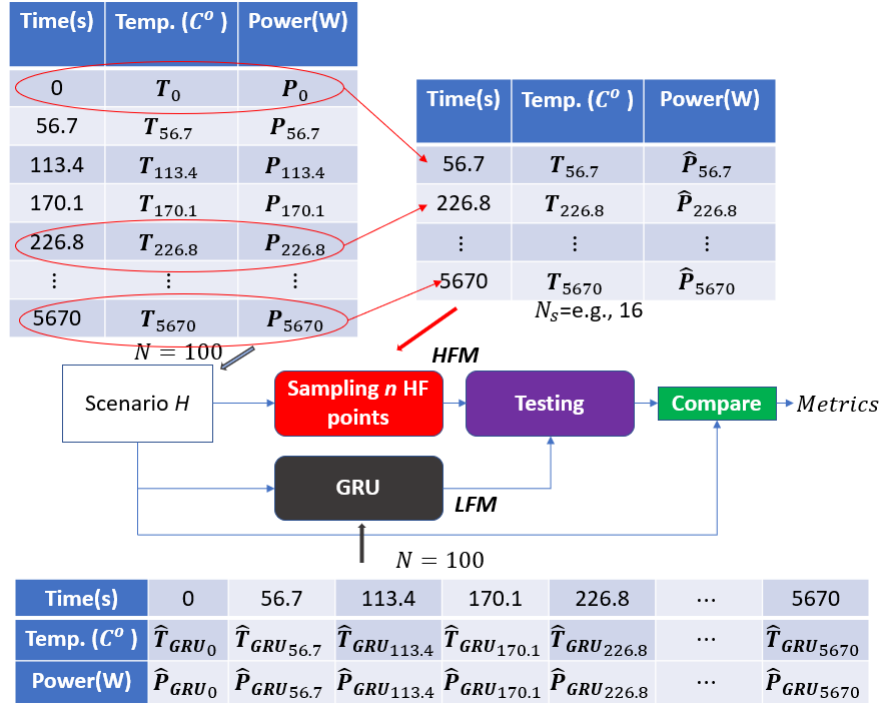


Figure 5.18: Testing approach summary for ISSL6U-1 data with power data. \mathbf{T} : vector of temperatures, $\hat{\mathbf{T}}_{GRU}$: vector of GRU estimated temperatures, $\hat{\mathbf{T}}_{ck}$: vector of cokriging, $\hat{\mathbf{P}}_{GRU}$: vector of GRU estimated power measurements, $\hat{\mathbf{P}}_{ck}$: vector of cokriging estimated power measurements, N : number of rows.

Time(s)	Temp. (C^o)	Power(W)
0	\hat{T}_{ck_0}	\hat{P}_{ck_0}
56.7	$\hat{T}_{ck_{56.7}}$	$\hat{P}_{ck_{56.7}}$
113.4	$\hat{T}_{ck_{113.4}}$	$\hat{P}_{ck_{113.4}}$
170.1	$\hat{T}_{ck_{170.1}}$	$\hat{P}_{ck_{170.1}}$
226.8	$\hat{T}_{ck_{226.8}}$	$\hat{P}_{ck_{226.8}}$
\vdots	\vdots	\vdots
5670	$\hat{T}_{ck_{5670}}$	$\hat{P}_{ck_{5670}}$

$N = 100$

Figure 5.19: Multifidelity framework's output format.

GP/ARIMAX

As with the ISSL6U-1 data, the testing strategy for the Gaussian process and ARIMAX again skips the GRU and uses the sampled high fidelity points, including power measurements, to estimate values as in Fig. 5.9.

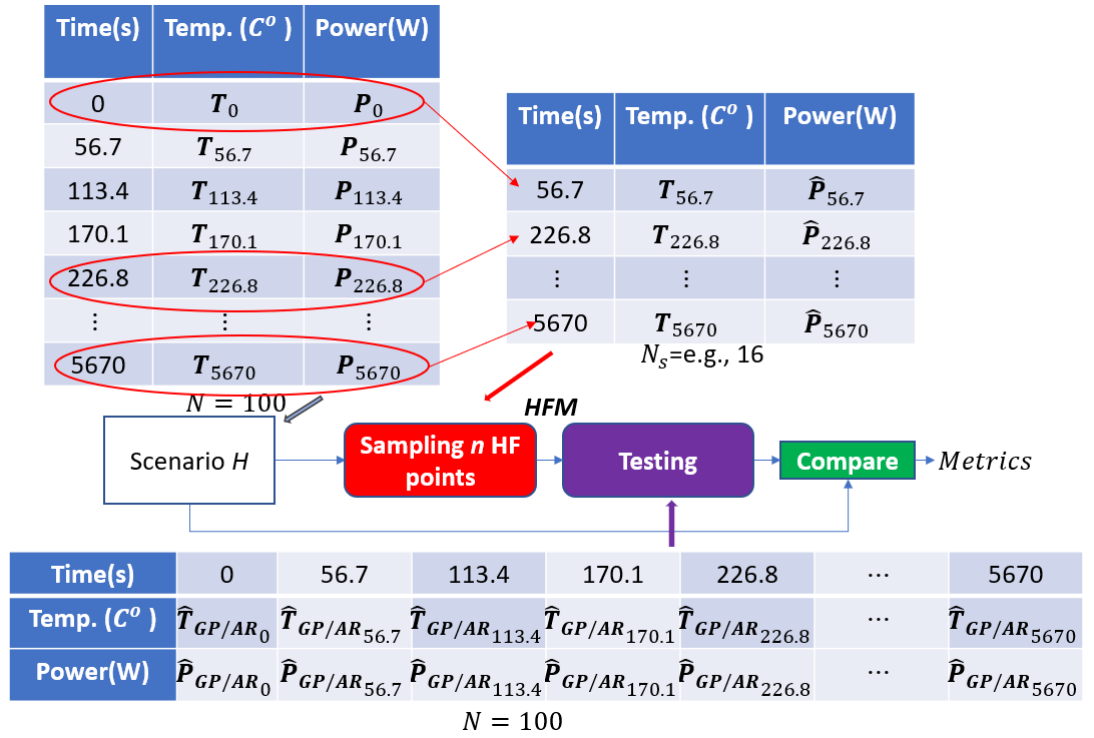


Figure 5.20: Testing approach summary for ISSL6U-1 data with power for GP and ARIMAX. \mathbf{T} : vector of temperatures, $\hat{\mathbf{T}}_{GRU}$: vector of GRU estimated temperatures, $\hat{\mathbf{T}}_{ck}$: vector of cokriging estimated temperatures, N : number of rows.

5.3.3 Results

GRU

In Fig. 5.21 the result of the GRU for the 6 communication and PCBs and 3 power components demonstrated. The y-axis for the communication components reflects temperature(C°) and for the power components reflects Watts(W).

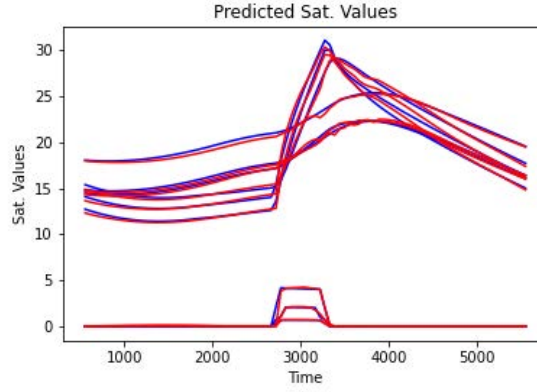
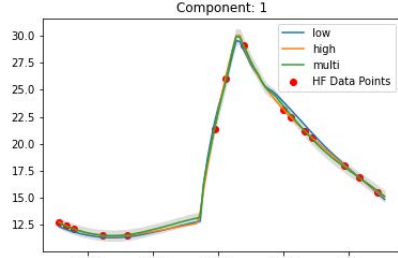


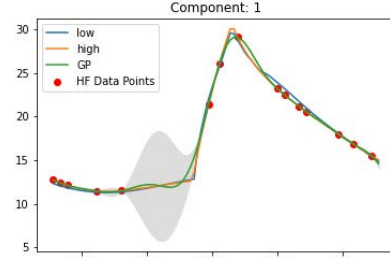
Figure 5.21: GRU results for ISSL6U-1 data with power. Red lines (-) indicate the predicted temperatures and blue lines (-) indicate the real temperatures for the 9 components.

15 Noisy Data Points

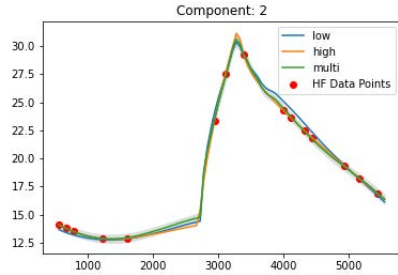
As with the previous ISSL6U-1 simulation without power, the simulation is run with 15, noisy, high fidelity data points to construct a more realistic situation. The results are summarised in Fig. A.101, 5.23, 5.24 and Tables 5.7, 5.8, 5.9. The graphs for each component are plotted separately.



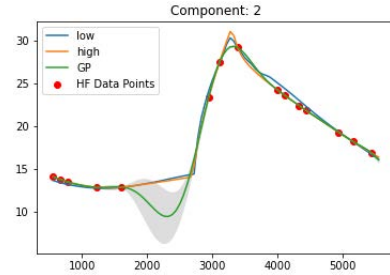
MF - Component 1



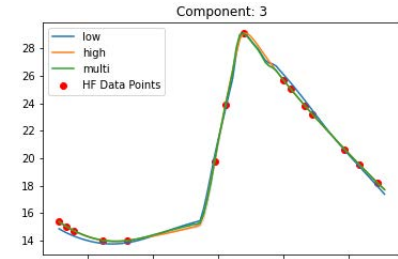
GP - Component 1



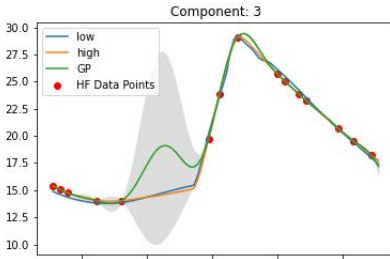
MF - Component 2



GP - Component 2



MF - Component 3

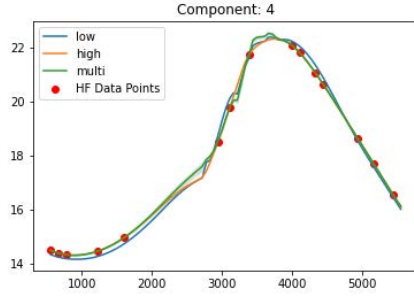


GP - Component 3

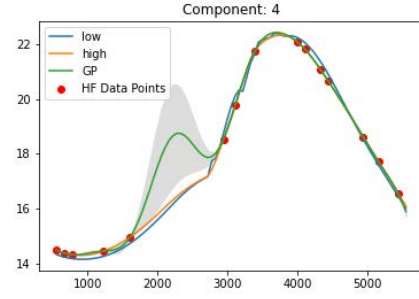
Figure 5.22: Multimodal simulation, including power data, with 15 HF points randomly placed. Blue lines (-) indicate the LFM, orange lines (-) the HFM, green lines (-) the multifidelity estimation and red dots (•) the HFM data points used and the gray area (■) the confidence interval for components 1-3.

ISSL6U-1 with Simulated Power Data 1-3			
Component 1			
<i>Metrics</i>	<i>MF</i>	<i>GRU</i>	<i>GP</i>
e_{MSE}	0.0461	0.1108	0.1612
e_{MAPE}	0.0104	0.01416	0.0124
e_{MAE}	0.1662	0.2525	0.2095
$\bar{\sigma}$	0.2382	n/a	0.4463
e_{MSE} change	n/a	-58 %	-71 %
e_{MAPE} change	n/a	-27 %	-16 %
e_{MAE} change	n/a	-34 %	-21 %
$\bar{\sigma}$ change	n/a	n/a	-47 %
Component 2			
e_{MSE}	0.0684	0.1596	1.5452
e_{MAPE}	0.0104	0.0153	0.0391
e_{MAE}	0.1796	0.302	0.5849
$\bar{\sigma}$	0.1817	n/a	0.2304
e_{MSE} change	n/a	-57 %	-96 %
e_{MAPE} change	n/a	-32 %	-74 %
e_{MAE} change	n/a	-41 %	-69 %
$\bar{\sigma}$ change	n/a	n/a	-21 %
Component 3			
e_{MSE}	0.0164	0.0789	2.1453
e_{MAPE}	0.0032	0.0133	0.0473
e_{MAE}	0.0669	0.2445	0.7222
$\bar{\sigma}$	0.0346	n/a	0.6262
e_{MSE} change	n/a	-79 %	-99 %
e_{MAPE} change	n/a	-75 %	-93 %
e_{MAE} change	n/a	-73 %	-91 %
$\bar{\sigma}$ change	n/a	n/a	-94 %

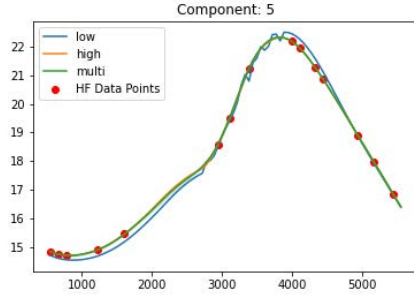
Table 5.7: Error & time metrics. 15 HF points - randomly placed for components 1-3.



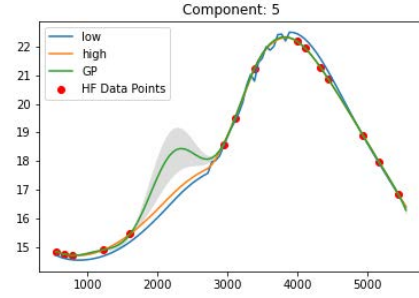
MF - Component 4



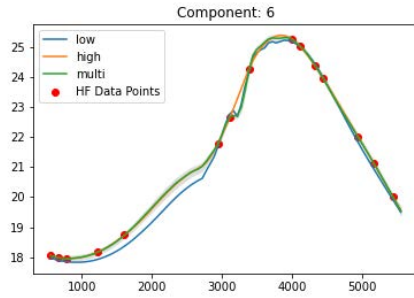
GP - Component 4



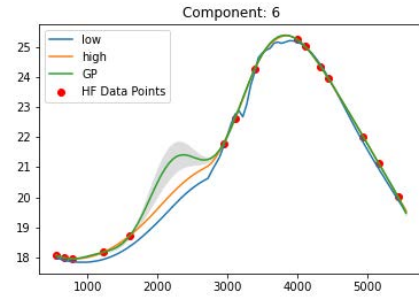
MF - Component 5



GP - Component 5



MF - Component 6

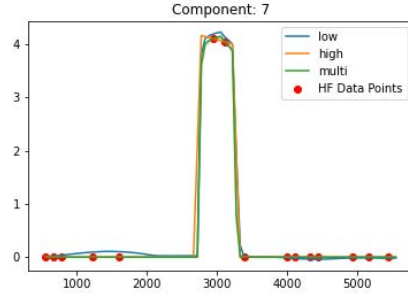


GP - Component 6

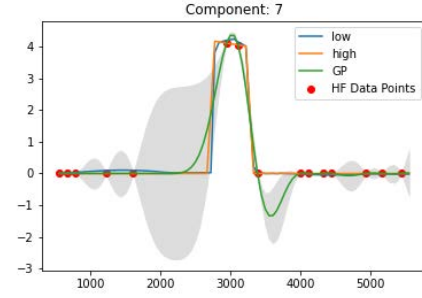
Figure 5.23: Multimodal simulation, including power data, with 15 HF points randomly placed. Blue lines (-) indicate the LFM, orange lines (-) the HFM, green lines (-) the multifidelity estimation and red dots (•) the HFM data points used and the gray area (■) the confidence interval for components 4-6.

ISSL6U-1 with Simulated Power Data 4-6			
Component 4			
<i>Metrics</i>	<i>MF</i>	<i>GRU</i>	<i>GP</i>
e_{MSE}	0.0197	0.0308	0.5551
e_{MAPE}	0.0036	0.0088	0.02123
e_{MAE}	0.0675	0.1531	0.3483
$\bar{\sigma}$	0.0202	n/a	0.1335
e_{MSE} change	n/a	-36 %	-96 %
e_{MAPE} change	n/a	-59 %	-83%
e_{MAE} change	n/a	-56 %	-81 %
$\bar{\sigma}$ change	n/a	n/a	-85 %
Component 5			
e_{MSE}	0.0006	0.0397	0.2107
e_{MAPE}	0.0008	0.0098	0.01252
e_{MAE}	0.0149	0.1736	0.2107
$\bar{\sigma}$	0.0134	n/a	0.0582
e_{MSE} change	n/a	-98 %	-100 %
e_{MAPE} change	n/a	-92 %	-93 %
e_{MAE} change	n/a	-91 %	-93 %
$\bar{\sigma}$ change	n/a	n/a	-77 %
Component 6			
e_{MSE}	0.0115	0.0677	0.1224
e_{MAPE}	0.0019	0.0103	0.0079
e_{MAE}	0.045	0.2116	0.1607
$\bar{\sigma}$	0.0379	n/a	0.0376
e_{MSE} change	n/a	-83 %	-91 %
e_{MAPE} change	n/a	-81 %	-75 %
e_{MAE} change	n/a	-79 %	-72 %
$\bar{\sigma}$ change	n/a	n/a	0.8 %

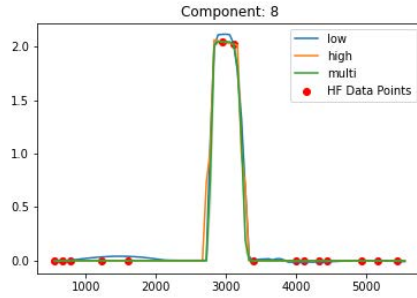
Table 5.8: Error & time metrics. 15 HF points - randomly placed for components 4-6.



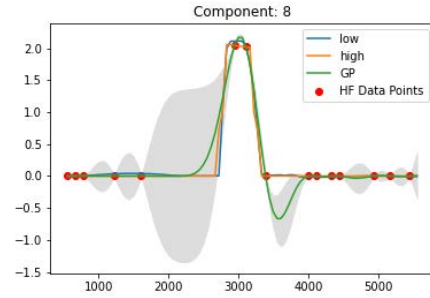
MF - Component 7



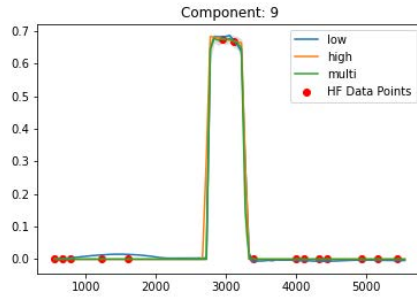
GP - Component 7



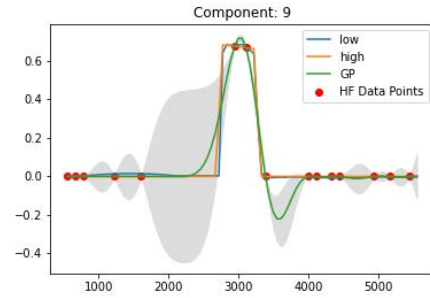
MF - Component 8



GP - Component 8



MF - Component 9



GP - Component 9

Figure 5.24: Multimodal simulation, including power data, with 15 HF points randomly placed. Blue lines (-) indicate the LFM, orange lines (-) the HFM, green lines (-) the multifidelity estimation and red dots (•) the HFM data points used and the gray area (■) the confidence interval for components 7-9.

ISSL6U-1 with Simulated Power Data 7-9			
Component 7			
<i>Metrics</i>	<i>MF</i>	<i>GRU</i>	<i>GP</i>
e_{MSE}	0.06836	0.0517	0.2018
e_{MAPE}	0.02	0.4562	0.187
e_{MAE}	0.04799	0.0689	0.2059
$\bar{\sigma}$	0.0028	n/a	0.3314
e_{MSE} change	n/a	32 %	-66 %
e_{MAPE} change	n/a	-100 %	-100%
e_{MAE} change	n/a	-30 %	-77 %
$\bar{\sigma}$ change	n/a	n/a	-99 %
Component 8			
e_{MSE}	0.0103	0.0082	0.0399
e_{MAPE}	0.9441	0.1588	0.9317
e_{MAE}	0.0214	0.0314	0.0939
$\bar{\sigma}$	0.0134	n/a	0.0582
e_{MSE} change	n/a	24 %	-74 %
e_{MAPE} change	n/a	-100 %	-100 %
e_{MAE} change	n/a	-32 %	-77 %
$\bar{\sigma}$ change	n/a	n/a	-98 %
Component 9			
e_{MSE}	0.0013	0.0011	0.0054
e_{MAPE}	0.02	0.436	0.19
e_{MAE}	0.0059	0.0099	0.0331
$\bar{\sigma}$	0.0015	n/a	0.0548
e_{MSE} change	n/a	19 %	-76 %
e_{MAPE} change	n/a	-99 %	-100 %
e_{MAE} change	n/a	-41 %	-82 %
$\bar{\sigma}$ change	n/a	n/a	-97 %

Table 5.9: Error & time metrics. 15 HF points - randomly placed for components 7-9.

ISSL6U-1 with Simulated Power Data Averages			
<i>Metrics</i>	<i>MF</i>	<i>GRU</i>	<i>GP</i>
\bar{e}_{MSE}	0.0269	0.0609	0.5541
\bar{e}_{MAPE}	0.1127	0.1247	0.161
\bar{e}_{MAE}	0.0684	0.1608	0.2854
$\bar{\sigma}$	0.0604	n/a	0.2196

Table 5.10: Metrics' averages for ISSL6U-1 data with power. 15 HF points - randomly placed.

GRU Comparison

In the multimodal case, the multifidelity framework, in terms of accuracy, performs better than the GRU for the communication components. This holds true for all three error metrics with the improvement ranging from 36% – 98% for the mean squared error. However, for the power components which have a pulse-like

response the GRU performs slightly better than the multifidelity network. The GRU performance is better at around 19% – 32% error reduction. The GRU’s computational time is shorter in all scenarios as expected so it is not included in the table.

GP Comparison

The discussion around the Gaussian process is very similar to the experiment without power data. The multifidelity framework’s performance dominates that of the Gaussian process’. This is true for the power components as well. In Fig. 5.24, where the waveforms resemble a pulse, if a strategically placed data point is missing then the Gaussian process cannot follow the high fidelity model.

5.3.4 Results with Sliding Window

In this simulation, the sliding window is used with the ISSL6U-1 thermal and power data. Fig. 5.25 illustrates an example window from component 1. The multifidelity model follows the high fidelity model closely, while the Gaussian process fails immediately as soon as a high fidelity data point is missing. While a discrepancy in a single window is not an issue by itself, these errors aggregate along the orbit causing the overall performance to degrade.

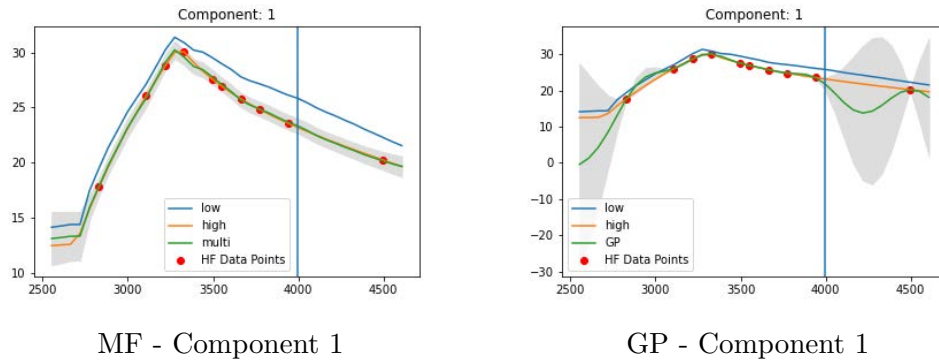
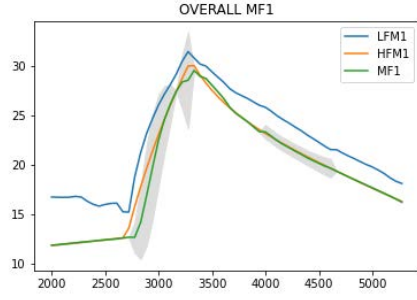
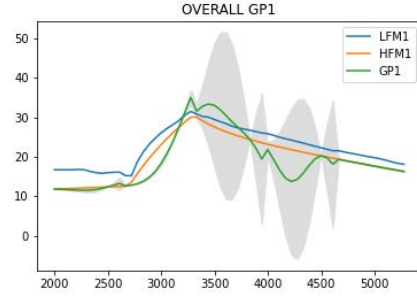


Figure 5.25: An example of a window for ISSL6U-1 data with power. The vertical blue lines indicate the point from which future values are calculated. Horizontal blue lines (—) indicate the LFM, orange lines (—) the HFM, green lines (—) the multifidelity estimation and the gray area (■) the confidence interval for components 1-3.

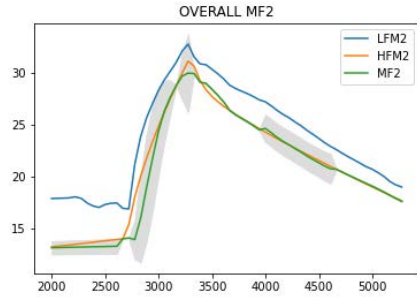
Running the window for an entire period and summing the results the overall values can be obtained as seen in Fig. 5.26, 5.27, 5.28. All the estimations together are plotted in Fig. 5.29 and the average metrics are presented in Table 5.11.



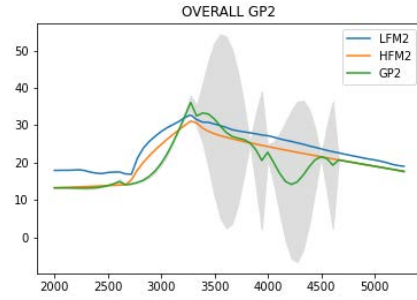
MF - Component 1



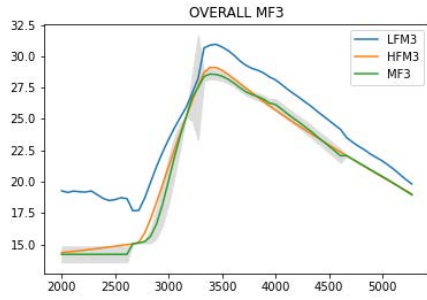
GP - Component 1



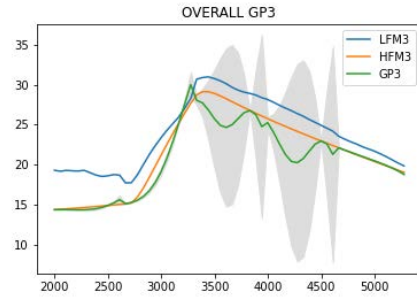
MF - Component 2



GP - Component 2

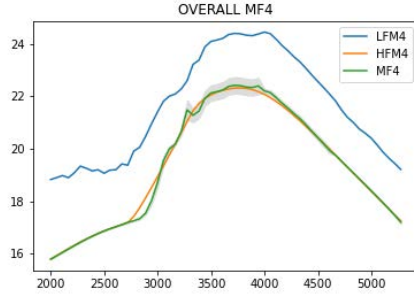


MF - Component 3

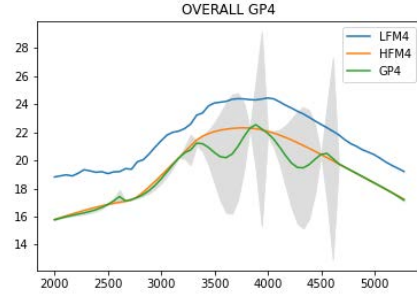


GP - Component 3

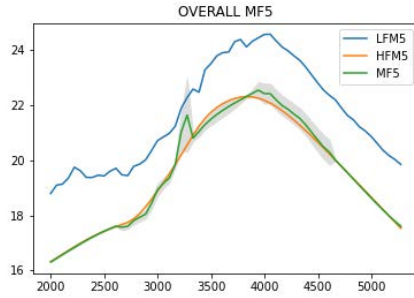
Figure 5.26: The overall result after sliding windows for ISSL6U-1 with power data. Blue lines (-) indicate the LFM, orange lines (-) the HFM, green lines (-) the multifidelity estimation and the gray area (■) the confidence interval for components 1-3.



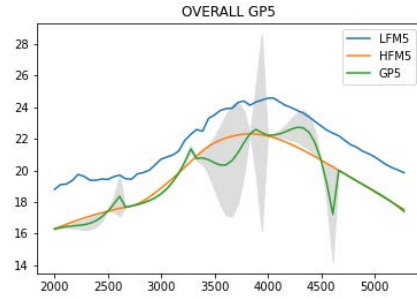
MF - Component 4



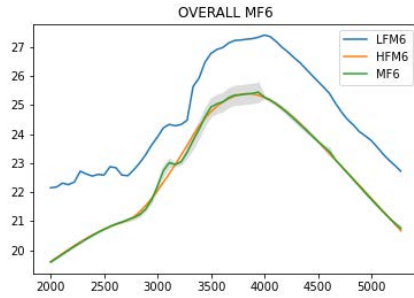
GP - Component 4



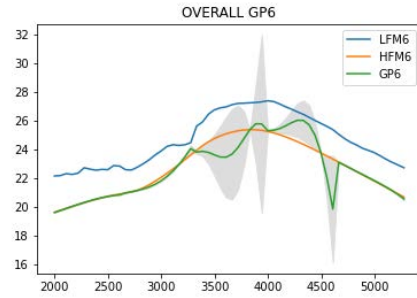
MF - Component 5



GP - Component 5

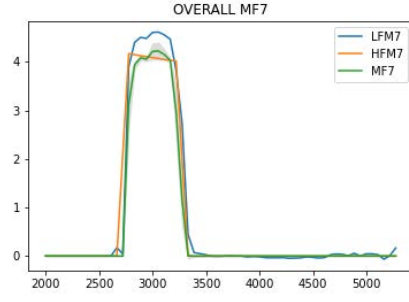


MF - Component 6

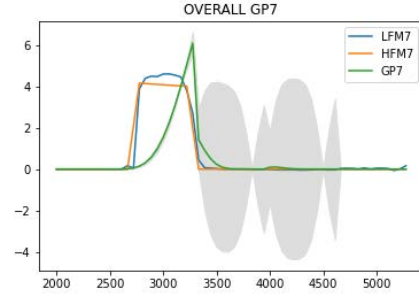


GP - Component 6

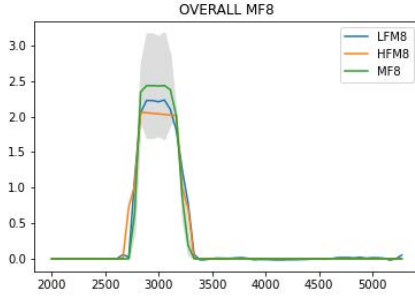
Figure 5.27: The overall result after sliding windows for ISSL6U-1 with power data. Blue lines (-) indicate the LFM, orange lines (-) the HFM, green lines (-) the multifidelity estimation and the gray area (■) the confidence interval for components 4-6.



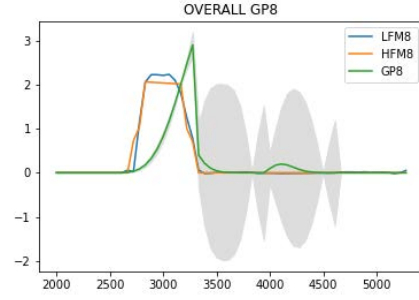
MF - Component 7



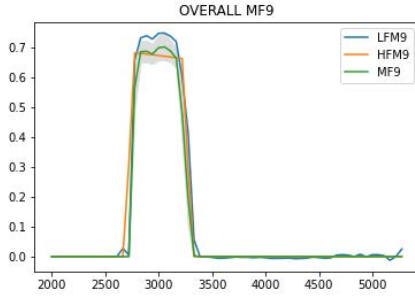
GP - Component 7



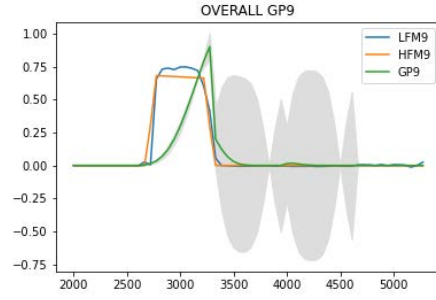
MF - Component 8



GP - Component 8



MF - Component 9



GP - Component 9

Figure 5.28: The overall result after sliding windows for ISSL6U-1 with power data. Blue lines (-) indicate the LFM, orange lines (-) the HFM, green lines (-) the multifidelity estimation and the gray area (■) the confidence interval for components 7-9.

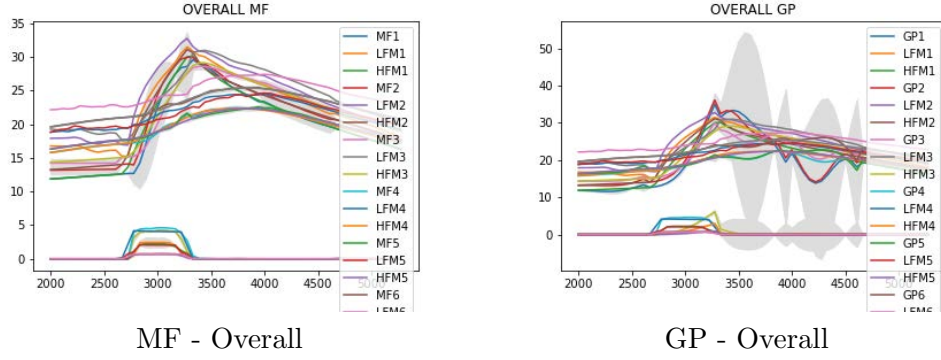


Figure 5.29: The overall result for all components after sliding windows for ISSL6U-1 with power data.. Blue lines (-) indicate the LFM, orange lines (-) the HFM, green lines (-) the multifidelity estimation and the gray area (■) the confidence interval for components 7-9.

Overall Average Performance			
<i>Metrics</i>	<i>MF</i>	<i>GRU</i>	<i>GP</i>
\bar{e}_{MSE}	0.6538	11.8789	1.4203
\bar{e}_{MAPE}	0.02	0.1818	0.4188
\bar{e}_{MAE}	0.50964	4.6623	2.3721
$\bar{\sigma}$	0.1508	n/a	0.9649
time(s)	47.4	33.3	2.1
\bar{e}_{MSE} change	n/a	-94 %	-30 %
\bar{e}_{MAPE} change	n/a	-88 %	-95 %
\bar{e}_{MAE} change	n/a	-89 %	-49 %
$\bar{\sigma}$ change	n/a	n/a %	-84 %

Table 5.11: Error & time metrics for the overall performance of the sliding window with ISSL6U-1 data with power.

Looking at the overall performance, the multifidelity framework shows consistently better performance than both the GRU and the Gaussian process. The average standard deviation remains 84% smaller for the framework against the Gaussian process with error improvements ranging from 30% to 95%.

5.3.5 ARIMAX

Another method used to predict ISSL6U-1 thermal data with power components is the ARIMAX, described in Section 2. Noisy data, similar to noisy data used with the other methods, were used to train the autoregressive method and a prediction for a future orbit was used. The results are summarised in Fig. 5.30, 5.31.

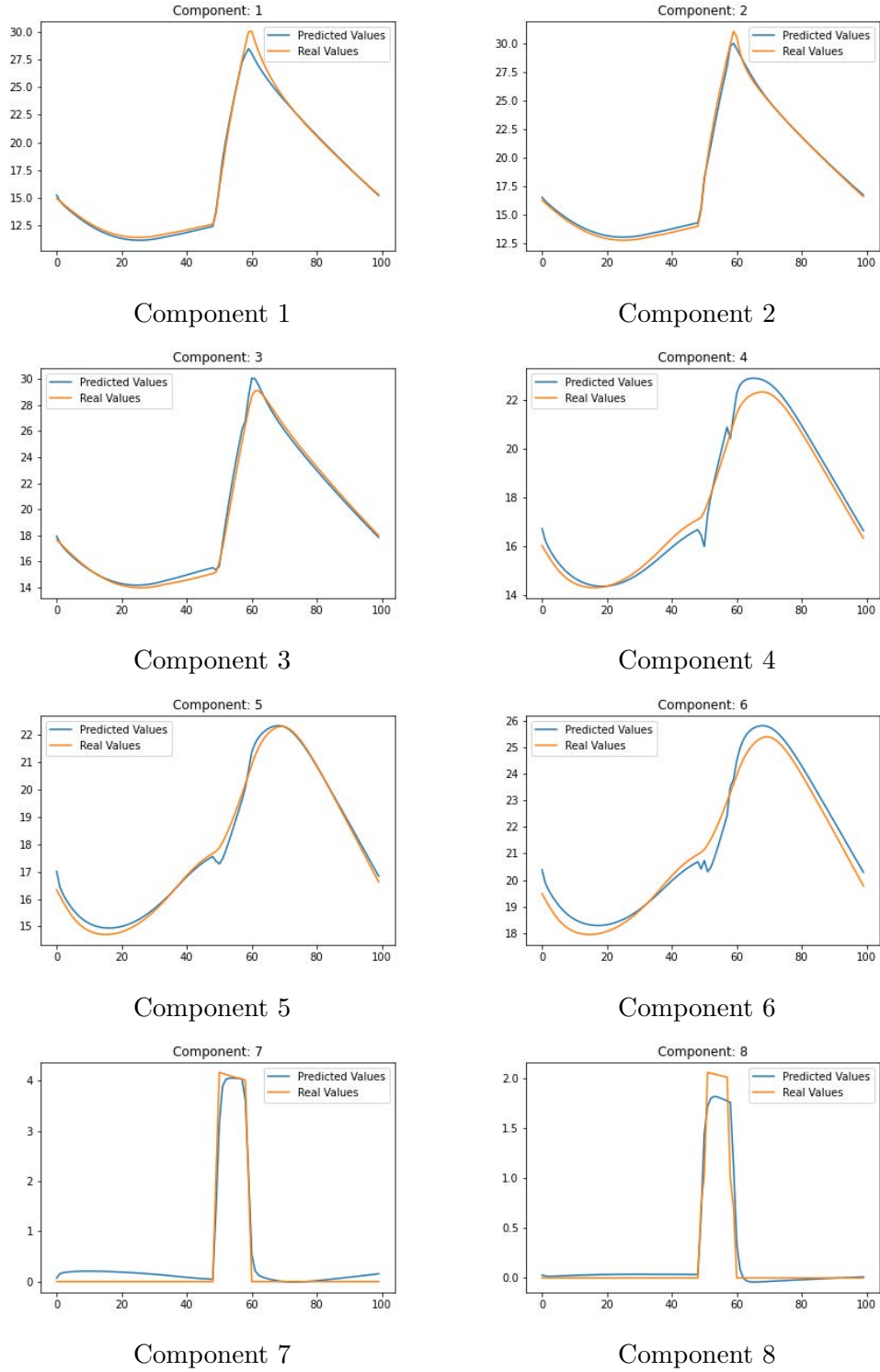


Figure 5.30: ARIMAX for components 1-8, ISSL6U-1 data with power. Blue lines (-) indicate the LFM, orange lines (-) the HFM, green lines (-) the multifidelity estimation for components 1-8. The x-axis represents the percentage of the orbit between 0 – 100%.

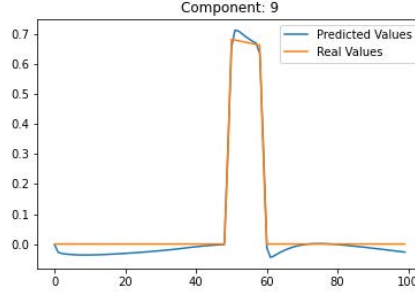


Figure 5.31: ARIMAX for component 9, ISSL6U-1 data with power. Blue lines (-) indicate the LFM, orange lines (-) the HFM, green lines (-) the multifidelity estimation for components 1-8. The x-axis represents the percentage of the orbit between 0 – 100%.

ARIMAX			
Components	MSE	MAPE	MAE
Component 1	0.364519	0.0229105	0.41391
Component 2	0.54562	0.0267743	0.51688
Component 3	0.111309	0.0130202	0.232922
Component 4	0.242182	0.024171	0.43191
Component 5	0.105697	0.0147827	0.257318
Component 6	0.305196	0.0226401	0.478764
Component 7	0.216641	0.16134	0.201385
Component 8	0.0456953	0.1007	0.0876811
Component 9	0.00499777	0.1596	0.0316515
Average	0.215762	0.1669	0.294714

Table 5.12: ARIMAX Errors per component, ISSL6U-1 data with power.

Comparison

Using the average metrics from Table 5.10 and combining them with the average metrics from Table 5.12 to construct Fig. 5.32. Comparing the errors with the previously discussed methods it is plain to see that ARIMAX has comparable errors to the Gaussian process, with the exception of the mean squared error, in which, ARIMAX still performs worse than the multifidelity model and the GRU. MAPE is similar for all methods and in MAE, the multifidelity framework and the GRU have slightly better performance.

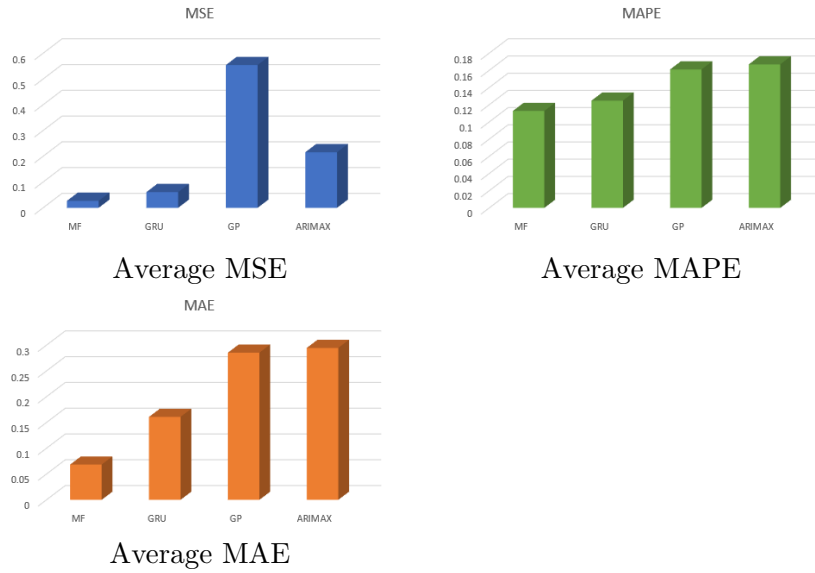


Figure 5.32: Average metric comparison, including ARIMAX. ISSL6U-1 data with power.

5.3.6 Robustness Study

Following the procedure outlined in Section 2, the multifidelity framework was tested for robustness against the placement of the high fidelity data points. The methods were applied to $N = 200$ scenarios differing in both the number and the placement of the high fidelity data points. The number of high fidelity points ranged from 5 to 20. Four robustness metrics were used, expressing different characteristics.

- Minimax: Expressing the minimum possible worst-case scenario.
- Laplace's Principle of Insufficient Reason: Describes average behaviour across all scenarios.
- Maximin: Expressing the minimum possible best-case scenario.
- Mean-Variance: Describes how the behaviour varies across scenarios.

Results of robustness values are summarised in Table 5.13. They have been transformed by $\frac{1}{R}$ if necessary so that the higher value corresponds to better performance.

Robustness Metrics			
Maximax			
<i>Method</i>	\bar{e}_{MSE}	<i>stdv</i>	\bar{e}_{MAE}
MF	0.074	1.755	0.0001
GP	0.058	0.276	0.00005
Laplace's Principle of Insufficient Reason			
MF	0.9	9.32	0.0027
GP	0.11	1.32	0.00035
Maximin			
MF	52.852	52.46	2.976
GP	12.049	38.599	3.71
Mean-Variance			
MF	0.0104	8.6244	$6 \cdot 10^{-7}$
GP	0.0027	0.767	$8 \cdot 10^{-8}$

Table 5.13: Robustness to HFM point placement, ISSL6U-1 data with power.

The multifidelity framework is consistently more robust across the board with the exception of the mean squared error in which both methods have similar robustness values. It would be interesting to look at how many scenarios the methods did not produce good results. The MSE errors per scenario can be seen in Fig. 5.33, 5.34.

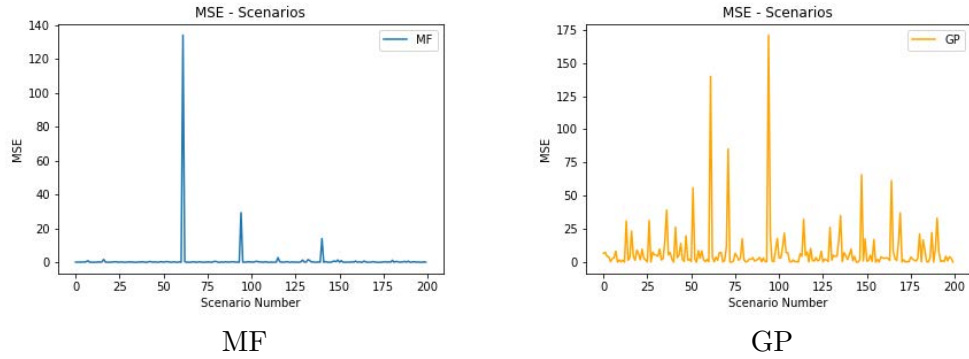


Figure 5.33: MSE per scenario, ISSL6U-1 data with power.

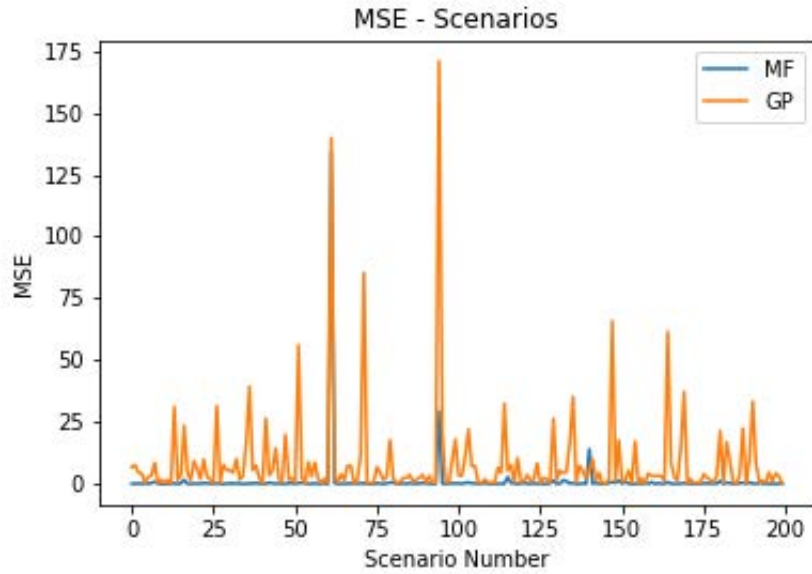


Figure 5.34: MF and GP MSE errors per scenario juxtaposed, ISSL6U-1 data with power.

In Fig. 5.33 it is evident that the multifidelity framework only failed in a handful of cases, in particular 3/200 scenarios while the Gaussian process failed, or at least has large errors, in at least 50% of the cases. Fig. 5.34 illustrates this by juxtaposing the errors. The three cases the multifidelity failed were cases in which the high fidelity data were randomly placed in a cluster, tightly bunched together, offering no useful information. In scenario 62 where the error was the highest points were indeed clustered as shown in Fig. 5.35 which shows the high fidelity on Component 9 for scenario 62.

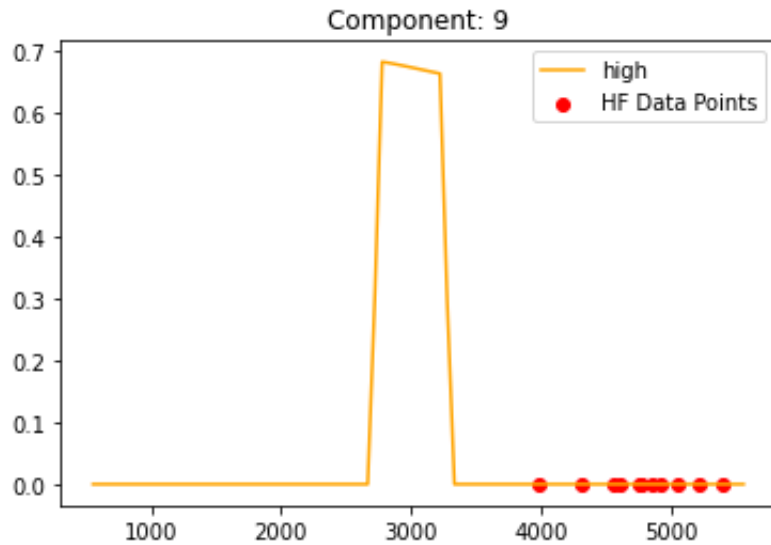


Figure 5.35: Point distribution in failed scenario 62.

5.3.7 Time Allocation

Concerning the runtime, the GRU required approximately 33 seconds. The multifidelity required approximately 47 seconds. This timeframe includes the time required for the GRU, the time for cokriging, while in parallel a finite element model would be producing a small number of expensive estimations. On the other hand, the gaussian process only takes 2 seconds to run. However it required high fidelity data which were not yet calculated. Therefore, you would have to wait for the high fidelity data points and then add the two seconds for the Gaussian process. In this last case, if you want the quality of your high fidelity points to be as good as in the proposed multifidelity framework, you would have to allocate comparable time to the high fidelity calculation. This is illustrated in Fig. 5.36.

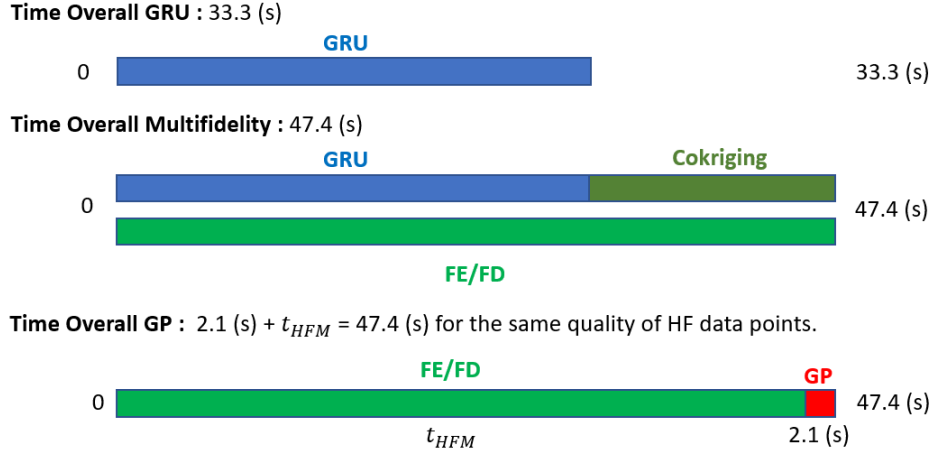


Figure 5.36: Time allocation for ISSL6U-1 data with power components.

5.3.8 Experiment Summary

The results in this section support the use of the framework for simulating multiple heterogeneous subsystems. The approach followed was described in Fig. 4.7. The GRU was used to detect interdependencies between component temperatures and simulated power components the values of which are functions of the temperatures $f(T)$. After a vector of predictions containing predictions for temperatures and power is obtained, each component is combined with its high fidelity counterpart through cokriging to produce a final estimation.

The simulation results consistently show that the multifidelity framework outperforms other methods. Error metrics are reduced in every simulation by the multifidelity network. Similarly, the uncertainty is also reduced. Concerning the Digital Twin, which is interested in real-time simulation, the sliding window produces improved results and does not suffer from missing values like its single fidelity counterpart, the Gaussian process. The standard deviation, even though slightly higher than when computing the values at the same time in their entirety, remains lower than the Gaussian process.

5.4 SDS-4

5.4.1 The SDS-4 Satellite

SDS-4 (Small Demonstration Satellite 4) is a small satellite (50 cm cube with mass of 50 kg) developed by JAXA. It was launched as a secondary payload on the Shizuku mission on 17 May 2012 UTC [146]. Its purpose was to test four experimental systems on-board.

- Space based Automatic Identification System Experiment (SPAISE)
- In-orbit performance evaluation of flat plate heat pipe (FOX: FHP On-orbit Experiment)
- In-flight experiment of Space materials using THERME (IST)
- Quartz Crystal Microbalance (QCM: Quartz Crystal Microbalance)

Its purpose is two-fold:

1. The realization of a 50 kg class small triaxial satellite standard bus.
2. Human resource development and research on efficient satellite development methods in a short period of time.

Fig. 5.37 illustrates SDS-4. The satellite's characteristics are summarised in Table 5.14.

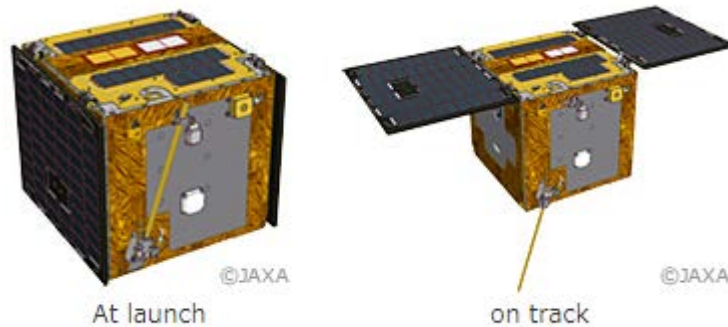


Figure 5.37: SDS-4 satellite model [146].

SDS-4 Characteristics	
Mass	48kg
Size	5U(50cm x 50cm x 50cm)
Attitude control	Triaxial zero momentum, Sun-oriented (steady time) Earth-oriented (during experiment)
Generated Power	120W (2 solar panels)
Communication	S-Band, command: 4kbps, HK telemetry: 16 kbps, mission telemetry: 1 kbps
Orbit	Altitude 677km

Table 5.14: Summary of SDS-4 characteristics.

5.4.2 Experiment Setup

The data used for the following simulations were taken through a thermal vacuum test on an SDS-4 model. The test lasted 6 days and measured logged temperature values every minute through an arrangement of 60 sensors placed strategically around the structure of the satellite. The results are presented in the following subsections for the same types of comparisons and simulations as with the ISSL6U-1 data.

As with the previous datasets the multifidelity framework will be compared against

- GRU
- Gaussian process
- ARIMAX

To simulate different modes of operation, at certain times during the week of testing, different subsystems were turned on and off generating different thermal responses to different components.

5.4.3 Multifidelity Framework Training Data

SDS-4 data resemble Table 5.15. At time moment t measurements T are obtained from each of the 60 placed sensors.

SDS-4 Data Vector					
$t(s)$	$T_1(C^\circ)$	$T_2(C^\circ)$	$T_3(C^\circ)$	\dots	$T_{60}(C^\circ)$

Table 5.15: SDS-4 data timeseries. Each temperature T corresponds to a sensor placed on the satellite’s body.

As the origin of the data was a week-long test, there is only a single set of data without repeating the test with different parameters. This creates an issue when deciding if and how to split the data. Usually when a single dataset is considered, data are split 80%-10%-10% or 70%-20%-10% between training, validation and

testing. However in this case this is not possible. At different times throughout tests, systems were turned on and off. The nature of these systems will remain vague for reasons of confidentiality. Splitting the data at any point would leave out engaged subsystems and valuable information for the GRU. A neural network is only as good as the data it was trained with. To create appropriate training data, augmented data must be created.

In practice, we may have a dataset taken in a limited set of conditions. But satellites may operate in a variety of conditions. To account for this, the network can be trained with additional synthetic data. Data augmentation can be done offline, by augmenting the data beforehand, or online, by augmenting the data in batches. In image processing, rotation, flipping and Gaussian noise are very popular augmentation techniques. Wen et al. [147] and Iwana and Uchida [148] surveyed methods for data augmentation in timeseries. In particular, three techniques were used in this work.

- *Jittering*: A simple yet effective technique is to simply add noise to the model.

$$\mathbf{x}' = x_1 + \epsilon_1, \dots, x_t + \epsilon_t, \dots, x_T + \epsilon_T \quad (5.15)$$

where, ϵ is often Gaussian noise added to each time step t , $\epsilon \sim \mathcal{N}(0, \sigma^2)$, σ^2 being the variance.

Adding noise to the inputs increases the generalization of neural networks under the assumption that unseen test patterns are only different from the training patterns by a factor of noise. Jittering also mitigates time series drift. Jittering is often used with sensor data [149] and has been used successfully in a variety of applications [150, 151].

- *Scaling*: This method alters the global magnitude of a random value using a scalar value.

$$\mathbf{x}' = \alpha x_1, \dots, \alpha x_t, \dots, \alpha x_T \quad (5.16)$$

where α is a scaling parameter than can be obtained either by a Gaussian distribution $\epsilon \sim \mathcal{N}(1, \sigma^2)$, σ^2 being the variance, [152] or from a random value from a pre-defined set [149].

- *Magnitude Warping*:

$$\mathbf{x}' = \alpha_1 x_1, \dots, \alpha_t x_t, \dots, \alpha_T x_T \quad (5.17)$$

where $\alpha_1, \dots, \alpha_t, \dots, \alpha_T$ is a set of scalars, obtained by interpolating a cubic spline $S(\mathbf{u})$ with knots $\mathbf{u} = u_1, \dots, u_i, \dots, u_1$. Each knot u_i is taken from a distribution $\mathcal{N}(1, \sigma^2)$ where the number of knots and the standard deviation σ are hyperparameters.

Um et al. [152] used the magnitude warping technique to introduce fluctuations in the data by increasing or decreasing random regions in the timeseries. However, magnitude warping assumes that the transformation is realistic and it depends on two, not one as with other methods, hyperparameters.

In this work two extra datasets for training and validation were created through two augmentations:

Data Augmentation 1:

- Jittering (Gaussian noise)
- Magnitude Warping (Sinusoidal)

Data Augmentation 2:

- Jittering (Gaussian noise)
- Scaling (Gaussian distribution)

Fig. A.71 illustrates our training approach. In this case, the vector of temperatures \mathbf{T} has dimensions 1×60 as the measurements of 60 sensors, placed on the model are used.

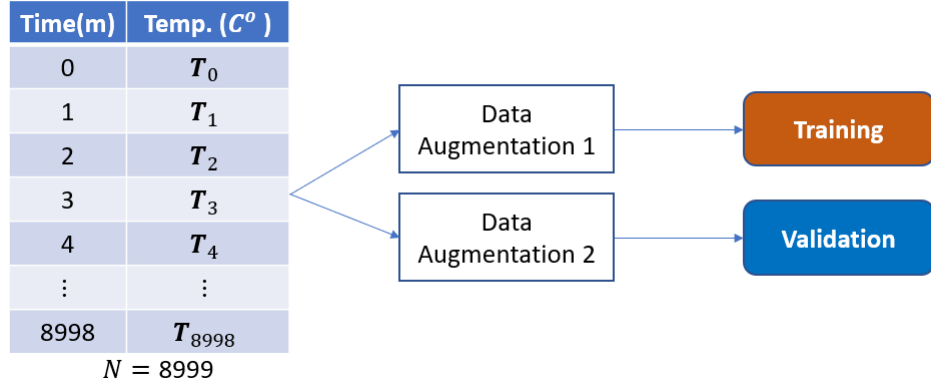


Figure 5.38: Training approach summary for SDS-4 data. \mathbf{T} : vector of temperatures, N : number of rows.

5.4.4 Multifidelity Framework Testing with SDS-4 Data

For the following tests, the testing dataset was downsampled uniformly to 1000 points for convenience in both presentation and development without loss of generality. For simulations using the entire dataset see Section 5.5. Fig. 5.39 demonstrates the testing process, the undistorted and downsampled dataset was used as input to the GRU to generate the LFM model. Moreover, 150, or around 16% out of all the measurements were sampled and taken as the high fidelity model.

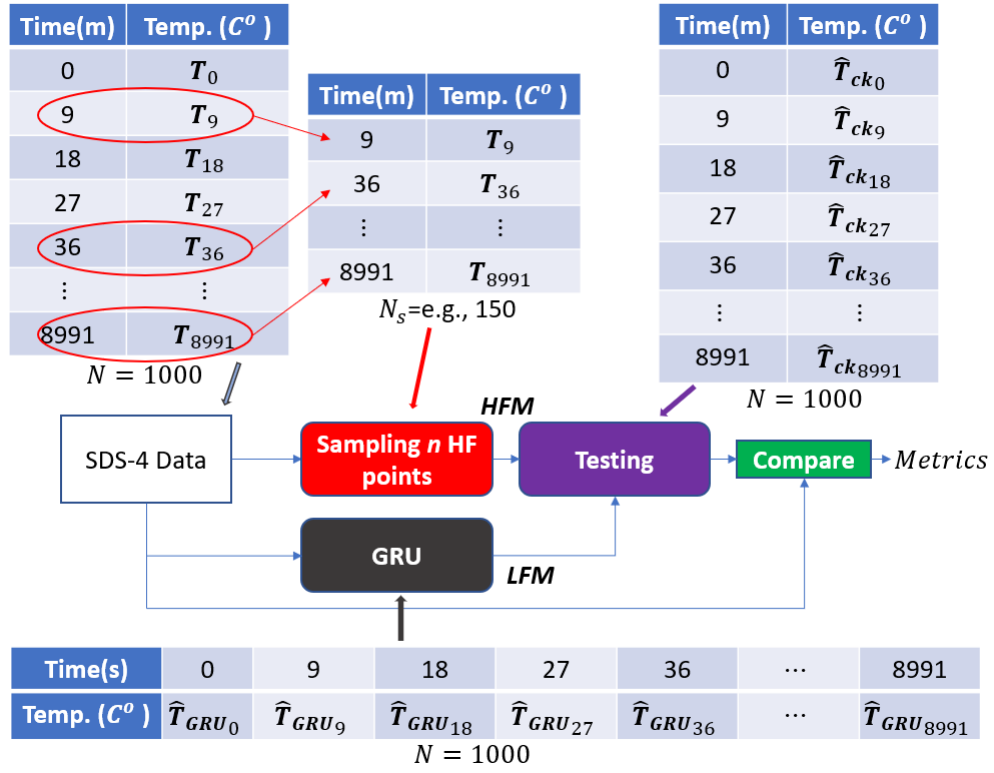


Figure 5.39: Testing approach summary for SDS-4 data. \mathbf{T} : vector of temperatures, $\hat{\mathbf{T}}_{\text{GRU}}$: vector of GRU estimated temperatures, $\hat{\mathbf{T}}_{\text{ck}}$: vector of cokriging estimated temperatures, N : number of rows.

GP/ARIMAX

For SDS-4 data, the testing strategy for the Gaussian process and ARIMAX is the same as with the ISSL6U-1 data. It skips the GRU and uses the sampled high fidelity points to estimate values as in Fig. 5.40. The difference is that temperature vector \mathbf{T} is of dimensions 1×60 .

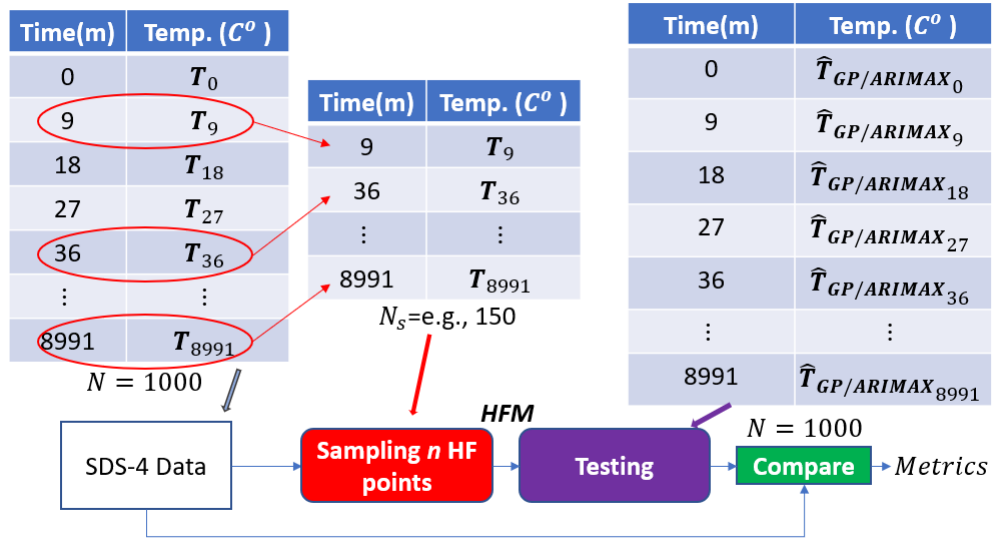


Figure 5.40: Testing approach summary for SDS-4 data for GP and ARIMAX. \mathbf{T} : vector of temperatures, $\hat{\mathbf{T}}_{\text{GRU}}$: vector of GRU estimated temperatures, $\hat{\mathbf{T}}_{\text{ck}}$: vector of cokriging estimated temperatures, N : number of rows.

5.4.5 Results

Due to the high number of plots, only a few will be showcased in this chapter. The rest can be found in Appendix ??.

GRU

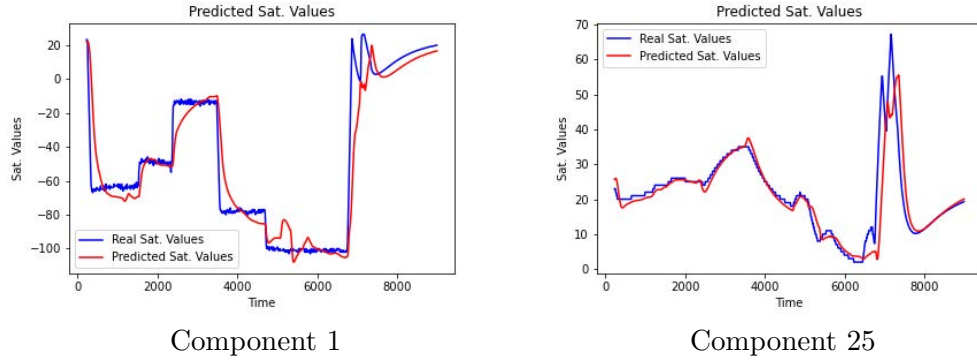


Figure 5.41: Examples of the GRU estimation for SDS-4 data. y-axis is temperature $T(C^\circ)$ and x-axis is time $t(s)$.

Since the behaviour of components across different actions varies, it is difficult for the GRU to predict the behaviour. For training, a heavily noisy version of the data was used. As previously said, at points along the testing week, different events took place. It would not be good practice to benchmark the GRU against events it has never seen. The average metrics can be found in Tables 5.18,??.

Even though the SDS-4 data have a peculiar shape, the GRU has mixed effectiveness, showcasing two components in Fig. 5.41. However, it is able to capture the general form of the data it is trying to predict, something very important for applying cokriging through the multifidelity framework. The rest of the figures can be found in Appendix A.2.

Uniform Data Points

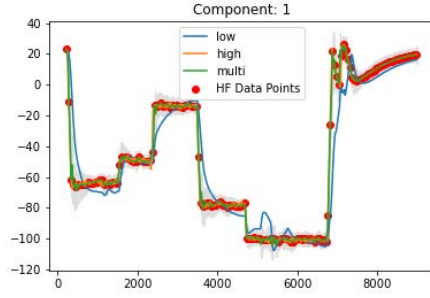
In this simulation, 16% of the data were added as high fidelity data points. This conforms to the requirements by Toal [34], outlined in Chapter 2. The GRU prediction previously shown was used as the LFM model and noiseless data points were used as the HFM values. In this subsection, the HFM values were taken at even intervals. The errors and standard deviation for each component are presented in Tables 5.16, 5.17.

SDS with Uniform High Fidelity Data 1-30				
Components	MSE	MAPE	MAE	sdtv
Component 1	6.7472	0.1627	1.125	2.5026
Component 2	7.2359	0.1353	1.1331	2.5421
Component 3	13.3107	0.1565	1.2978	3.9175
Component 4	10.9333	0.0852	1.1616	3.9868
Component 5	0.2056	0.1601	0.2265	1.0584
Component 6	0.3235	0.237	0.2558	0.9732
Component 7	0.6171	0.4962	0.3262	1.326
Component 8	12.38	0.1409	1.0386	2.2307
Component 9	0.1965	0.406	0.2186	0.9358
Component 10	3.3689	0.0261	0.6979	2.3184
Component 11	0.2706	0.2551	0.2513	1.0118
Component 12	0.2137	0.0225	0.2355	0.8281
Component 13	0.2733	0.3144	0.2502	0.8566
Component 14	0.1872	0.2583	0.2079	1.0071
Component 15	0.2262	0.5079	0.2283	1.0177
Component 16	0.2374	0.2566	0.2198	1.0541
Component 17	0.554	0.0219	0.2723	1.1568
Component 18	0.2614	0.1461	0.2287	0.9793
Component 19	0.2118	0.1111	0.2197	0.9915
Component 20	0.275	0.758	0.2232	0.8779
Component 21	0.1934	0.0124	0.2116	0.8312
Component 22	0.2944	0.2191	0.2453	0.9411
Component 23	0.3639	0.1726	0.2295	1.1552
Component 24	0.2824	0.1576	0.2278	0.9817
Component 25	1.3917	0.0292	0.4072	1.2387
Component 26	0.2073	0.0141	0.1953	1.1442
Component 27	0.4472	0.8006	0.2619	1.3006
Component 28	2.3746	0.2169	0.477	1.8017
Component 29	0.1616	0.9714	0.2055	1.0039
Component 30	0.1857	0.015	0.2095	0.9841

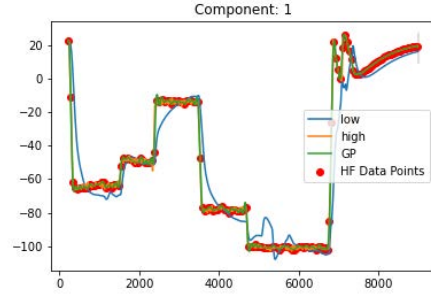
Table 5.16: SDS-4 Errors per component 1-30. Uniform data.

SDS with Uniform High Fidelity Data 31-60				
Components	MSE	MAPE	MAE	sdtv
Component 31	0.1621	0.1512	0.2001	1.0441
Component 32	0.1819	0.2359	0.2139	1.0525
Component 33	0.3147	0.1625	0.2444	0.9556
Component 34	0.2117	0.0147	0.2247	0.8546
Component 35	0.3473	0.0157	0.2645	0.8253
Component 36	0.3254	0.0148	0.249	0.9661
Component 37	0.2942	0.514	0.2531	0.8959
Component 38	0.2928	0.0159	0.2383	0.9469
Component 39	0.4298	0.0194	0.2667	0.8409
Component 40	0.4819	0.1464	0.273	0.8562
Component 41	19.0259	0.6989	1.357	3.709
Component 42	20.8349	0.3334	1.3467	3.8172
Component 43	25.6015	0.1507	1.5338	4.0707
Component 44	25.2526	0.2204	1.3982	4.032
Component 45	158.8609	0.1291	3.6078	5.6496
Component 46	154.9722	0.1454	3.1606	5.8536
Component 47	155.1367	0.1728	3.3094	5.7025
Component 48	18.8077	0.3741	1.2959	3.8599
Component 49	20.5748	0.0689	1.3558	4.2203
Component 50	21.0951	0.0626	1.357	4.1897
Component 51	19.883	0.418	1.2426	3.7153
Component 52	18.4187	0.8087	1.209	3.4107
Component 53	24.6543	0.8313	1.3466	3.88
Component 54	47.8983	0.1979	2.0709	6.9572
Component 55	58.4307	0.2576	2.5395	6.9104
Component 56	41.9387	0.3381	2.0584	7.0709
Component 57	1.8817	0.0371	0.4971	2.0058
Component 58	0.3105	0.9091	0.2555	0.6143
Component 59	1.9964	0.9506	0.5113	1.9544
Component 60	7.9632	0.1146	0.8317	2.4593
Average	15.1751	0.3348	0.7866	2.2713

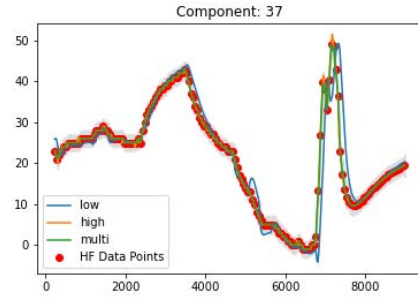
Table 5.17: SDS-4 Errors per component 31-60 and averages. Uniform data.



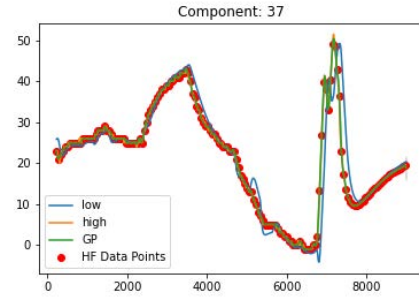
MF - Component 1



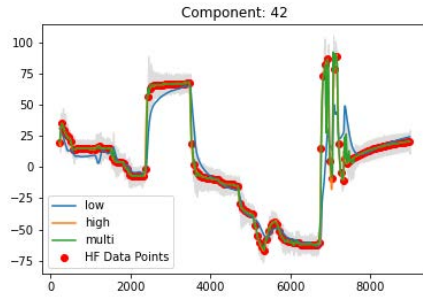
GP - Component 1



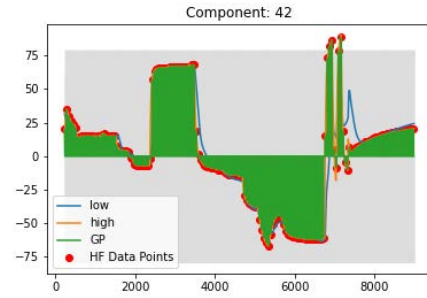
MF - Component 37



GP - Component 37



MF - Component 42



GP - Component 42

Figure 5.42: SDS-4 data, multifidelity vs Gaussian process. Uniform high fidelity data points. Blue lines (-) indicate the LFM, orange lines (-) the HFM, green lines (-) the multifidelity estimation and red dots (●) the HFM data points used and the gray area (■) the confidence interval for components 1 and 42.

Overall Average Performance SDS-4			
<i>Metrics</i>	<i>MF</i>	<i>GRU</i>	<i>GP</i>
\bar{e}_{MSE}	15.1751	134.66	76.7946
\bar{e}_{MAPE}	0.3348	0.5818	0.548
\bar{e}_{MAE}	0.7866	5.357	1.8643
$\bar{\sigma}$	2.2713	n/a	1.893
time(s)	484.07	5.77	18.92
\bar{e}_{MSE} change	n/a	-89 %	-80 %
\bar{e}_{MAPE} change	n/a	-42 %	-39 %
\bar{e}_{MAE} change	n/a	-85 %	-58 %
$\bar{\sigma}$ change	n/a	n/a %	20 %

Table 5.18: Error & Time metrics overall average performance SDS-4 with uniform data.

With uniform data points, the performance difference between the multifidelity framework and the Gaussian process is minimal. An example is component 1 in Fig. 5.42. The large error improvements in Table 5.18 are due to a number of Gaussian processes failing to fit the data as in component 42 in Fig. 5.42. The multifidelity framework does not suffer of this kind of problems as it can always fall back on the LFM for guidance. The Gaussian process could be fixed by tuning its parameters, however this would defeat the purpose as the designer would have to spend time on trial and error. The rest of the figures can be found in Appendix A.3.

Random Data Points

In this simulation, 16% of the data were added as high fidelity data points. The location of the high fidelity data points were chosen at random. The GRU prediction previously shown is again the same as in the previous test and noiseless data points were used as the HFM values. The errors and standard deviation for each component are presented in Tables 5.19, 5.20.

SDS-4				
Components	MSE	MAPE	MAE	sdtv
Component 1	13.9535	0.2294	1.6056	3.0624
Component 2	14.2209	0.1815	1.5784	3.1207
Component 3	14.7697	0.2127	1.5347	4.9202
Component 4	15.19	0.1461	1.6035	4.9053
Component 5	1.2722	0.7795	0.4953	1.3298
Component 6	0.8672	0.1158	0.3825	1.2612
Component 7	1.7635	0.2234	0.548	1.6383
Component 8	20.0857	0.1859	1.3598	1.9845
Component 9	0.8382	0.1059	0.3564	1.224
Component 10	6.9475	0.0329	0.9814	2.6553
Component 11	0.8537	0.4156	0.3784	1.3224
Component 12	0.749	0.0309	0.3888	1.0741
Component 13	0.8001	0.4898	0.3845	1.1149
Component 14	1.066	0.2705	0.3678	1.3032
Component 15	1.0976	0.1686	0.4101	1.3163
Component 16	0.9759	0.4591	0.3949	1.3598
Component 17	1.7196	0.0616	0.5275	1.4435
Component 18	1.1612	0.2876	0.4207	1.2324
Component 19	1.0693	0.4799	0.4331	1.2714
Component 20	0.6506	0.2426	0.3672	1.1948
Component 21	0.7506	0.0202	0.3828	1.0582
Component 22	0.8815	0.3482	0.3897	1.2204
Component 23	1.5999	0.9106	0.4669	1.4428
Component 24	1.0374	0.4979	0.3583	1.2719
Component 25	1.4965	0.0525	0.4672	1.5741
Component 26	1.661	0.0487	0.51	1.4774
Component 27	1.0877	0.1001	0.4152	1.7263
Component 28	1.6144	0.4595	0.4515	2.5632
Component 29	1.0768	0.2363	0.3998	1.2795
Component 30	1.016	0.0244	0.3836	1.2757

Table 5.19: SDS-4 Errors per component, components 1-30.

SDS				
Components	MSE	MAPE	MAE	sdtv
Component 31	1.1813	0.2969	0.4552	1.3059
Component 32	1.1536	0.2383	0.4205	1.3449
Component 33	1.114	0.4008	0.3706	1.2388
Component 34	0.9518	0.0238	0.4051	1.0878
Component 35	1.0465	0.023	0.4035	1.0785
Component 36	0.7344	0.0211	0.3577	1.2721
Component 37	0.8253	0.4146	0.3929	1.1543
Component 38	0.8817	0.023	0.3777	1.2258
Component 39	0.9915	0.0323	0.3765	1.1018
Component 40	1.048	0.1264	0.3966	1.1177
Component 41	18.1006	0.1426	1.6287	5.2129
Component 42	17.3224	0.681	1.5471	5.4341
Component 43	21.0952	0.1653	1.695	5.8682
Component 44	18.9748	0.2183	1.6209	5.6591
Component 45	66.7939	0.7508	2.6815	7.1275
Component 46	211.3309	0.4059	5.4862	7.1998
Component 47	64.5278	0.1945	2.6473	7.286
Component 48	22.1335	0.3592	1.8692	5.3258
Component 49	34.0137	0.1592	2.2589	5.5622
Component 50	26.3905	0.1674	1.9393	5.6639
Component 51	30.2098	0.4849	1.7247	4.7713
Component 52	22.594	0.1051	1.5059	4.8039
Component 53	27.3127	0.2239	1.6068	5.1369
Component 54	64.1415	0.3252	2.9629	8.7182
Component 55	90.152	0.2133	3.5691	8.488
Component 56	89.1831	0.5135	3.5382	8.4284
Component 57	8.7354	0.149	1.1362	2.5003
Component 58	0.6693	0.1268	0.3826	0.7887
Component 59	5.878	0.7424	0.9142	2.5685
Component 60	21.4409	0.7977	1.5285	2.8505
Average	16.386	0.4544	1.082	5.8315

Table 5.20: SDS-4 Errors per component, components 31-60, and averages.

With randomly placed high fidelity points, the multifidelity framework performs 88% and 92% better than the GRU and the Gaussian process respectively. Additionally, there is a 37% reduction in the standard deviation against the Gaussian process. The average metrics are summarised in Table 5.21. Fig. 5.43 showcases examples for components 1, 37 and 47. At locations without a high fidelity point the Gaussian process either fails to predict accurately, or has large standard deviation. The behaviour is similar to the ISS6U-1 data for the same simulation setup. The framework shows consistent performance, irregardless of the data. The rest of the figures can be found in Appendix A.4.

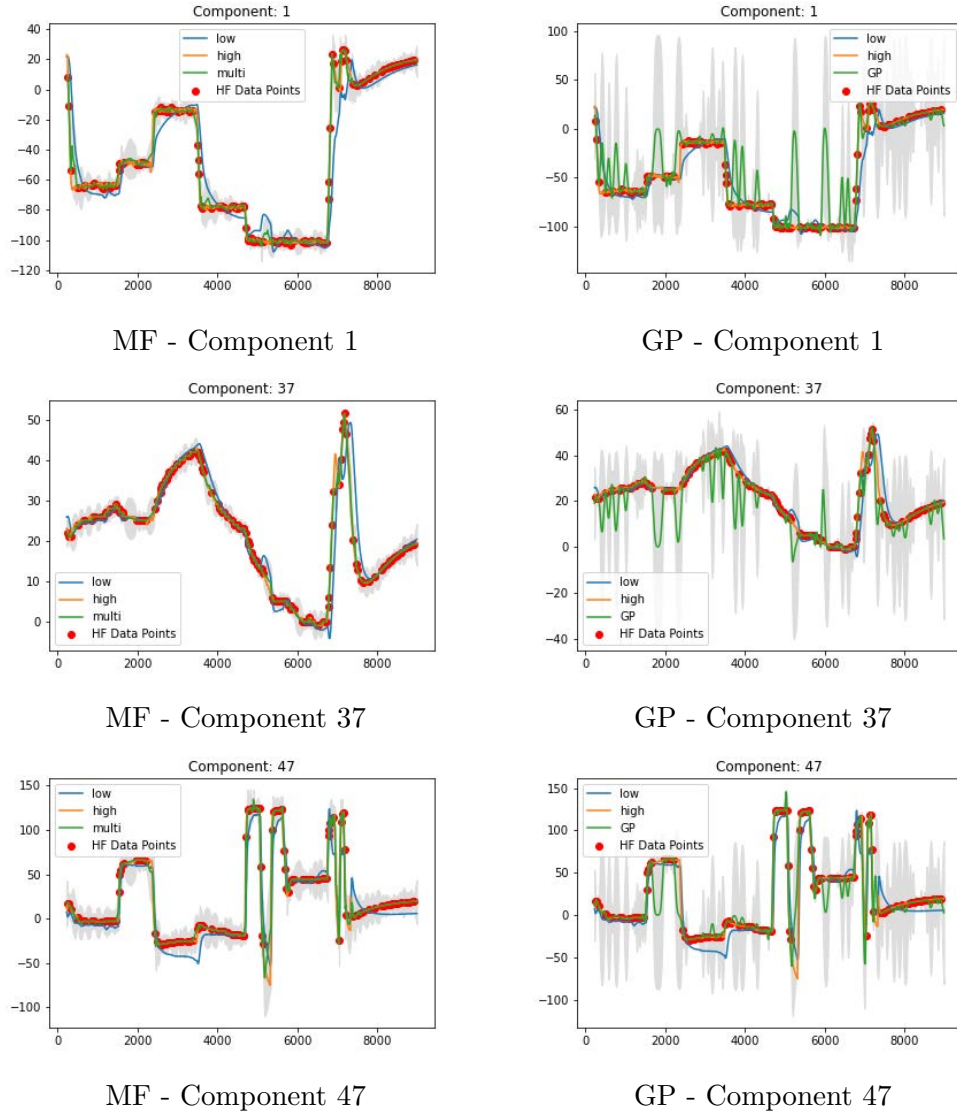


Figure 5.43: SDS-4 data, multifidelity vs Gaussian process. Randomly placed high fidelity data points. Blue lines (-) indicate the LFM, orange lines (-) the HFM, green lines (-) the multifidelity estimation and red dots (•) the HFM data points used and the gray area (■) the confidence interval for components 1, 37 and 47.

Overall Average Performance SDS-4			
<i>Metrics</i>	<i>MF</i>	<i>GRU</i>	<i>GP</i>
\bar{e}_{MSE}	16.386	134.66	210.52
\bar{e}_{MAPE}	0.4544	0.5818	0.648
\bar{e}_{MAE}	1.082	5.357	6.33
$\bar{\sigma}$	5.8315	n/a	9.37
time(s)	478.84	5.88	18.66
\bar{e}_{MSE} change	n/a	-88 %	-92 %
\bar{e}_{MAPE} change	n/a	-22 %	-30 %
\bar{e}_{MAE} change	n/a	-79 %	-83 %
$\bar{\sigma}$ change	n/a	n/a %	-37 %

Table 5.21: Error & time metrics overall performance for random HF data placement, SDS-4 data.

5.4.6 Results with Sliding Windows

In this simulation, the sliding window is used with the SDS-4 thermal data. Fig. 5.44 illustrates an example window from component 43. The multifidelity framework resembles the high fidelity model closely. An example of this is Fig. 5.45 for component 57. The Gaussian process performs 31% - 86% worse according to the error metrics in Table 5.22. Moreover the multifidelity framework has 62% smaller standard deviation. The rest of the figures can be found in Appendix A.5.

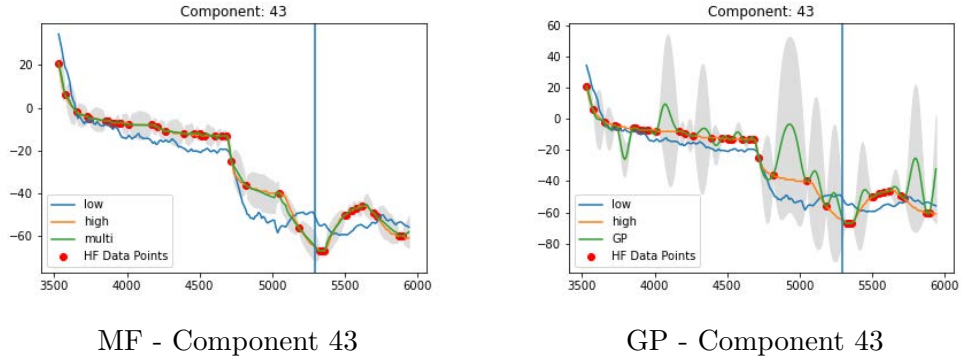


Figure 5.44: An example of a window for component 43 of SDS-4. The vertical blue lines indicate the point from which future values are calculated. Horizontal blue lines (-) indicate the LFM, orange lines (-) the HFM, green lines (-) the multifidelity estimation and the gray area (■) the confidence interval.

Overall Average Performance			
<i>Metrics</i>	<i>MF</i>	<i>GRU</i>	<i>GP</i>
\bar{e}_{MSE}	35.7	199.33	264.18
\bar{e}_{MAPE}	0.489	0.611	0.714
\bar{e}_{MAE}	2.6114	9.216	13.43
$\bar{\sigma}$	7.3643	n/a	12.301
time(s)	701	23.1	42.7
\bar{e}_{MSE} change	n/a	-82 %	-86 %
\bar{e}_{MAPE} change	n/a	-20 %	-31 %
\bar{e}_{MAE} change	n/a	-71 %	-80 %
$\bar{\sigma}$ change	n/a	n/a	-40 %

Table 5.22: Error & time metrics overall performance for SDS-4 data with the sliding window.

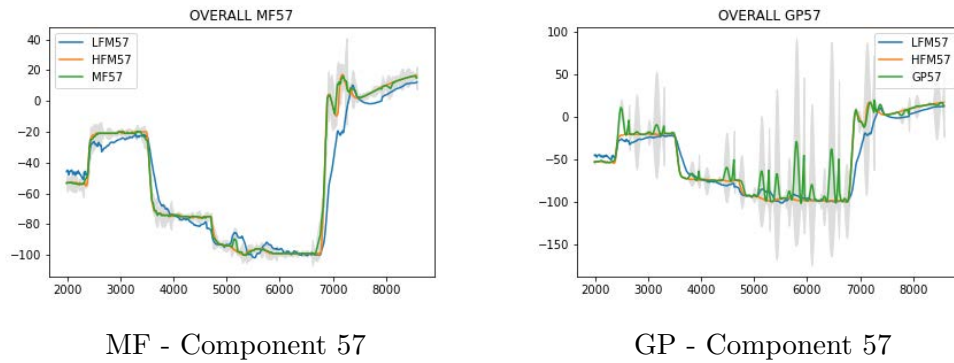
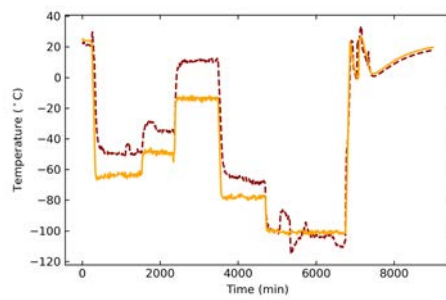


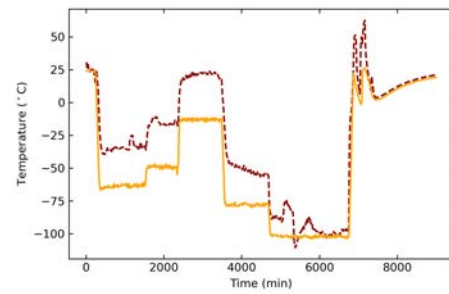
Figure 5.45: An example of a window for component 57 of SDS-4 data. The vertical blue lines indicate the point from which future values are calculated. Horizontal blue lines (-) indicate the LFM, orange lines (-) the HFM, green lines (-) the multifidelity estimation and the gray area (■) the confidence interval.

5.4.7 ARIMAX

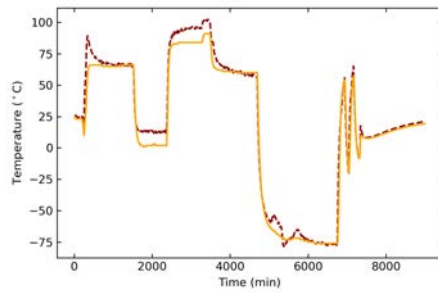
As with the ISSL6U-1 thermal data, ARIMAX is applied on SDS-4 data as well. Noisy data were used to predict future values. Examples are presented in Fig. 5.46 for components 1-4. The prediction of all the components can be found in the Appendix. Tables 5.23, 5.24 contain the error metrics for all SDS-4 components. The rest of the figures can be found in Appendix A.6.



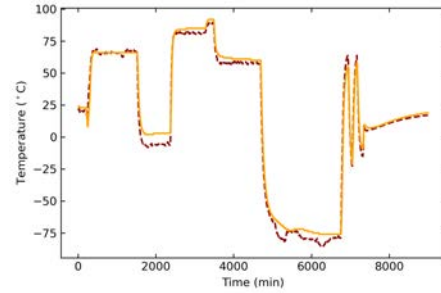
Component 1



Component 2



Component 3



Component 4

Figure 5.46: ARIMAX SDS-4 data. Blue lines (-) indicate the LFM, orange lines (-) the ARIMAX prediction for components 1-4.

SDS-4			
Components	MSE	MAPE	MAE
Component 1	205.1563	0.8492	12.1539
Component 2	136.8604	0.66	8.6084
Component 3	66.814	0.5231	6.0199
Component 4	132.7513	0.7554	9.0196
Component 5	74.7555	0.3412	6.5079
Component 6	20.9458	0.15266	3.7171
Component 7	6.9055	0.1459	2.0726
Component 8	141.8519	0.4996	8.7529
Component 9	3.0986	0.5068	1.4088
Component 10	256.6241	0.8691	11.9928
Component 11	58.3973	0.3935	6.2202
Component 12	4.2618	0.1594	1.346
Component 13	5.5722	0.7785	1.7377
Component 14	23.6887	0.5802	4.0045
Component 15	10.2232	0.3325	2.4808
Component 16	41.4733	0.533	4.9783
Component 17	23.0856	0.5939	4.0258
Component 18	22.2677	0.8922	3.7512
Component 19	21.9017	0.3945	3.9201
Component 20	8.237	0.7949	2.1401
Component 21	40.8073	0.3464	5.3113
Component 22	97.532	0.7455	7.8859
Component 23	9.2095	0.3643	2.2343
Component 24	6.2302	0.3566	1.8857
Component 25	13.0842	0.3455	2.998
Component 26	67.3693	0.3849	6.3228
Component 27	8.7006	0.74911	2.2964
Component 28	4.6045	0.1863	1.5515
Component 29	20.3605	0.7747	4.102
Component 30	43.9475	0.4588	5.2739

Table 5.23: SDS-4 ARIMAX errors per component, components 1-30.

SDS-4			
Components	MSE	MAPE	MAE
Component 31	15.1662	0.4801	2.8548
Component 32	7.1772	0.5982	1.9974
Component 33	21.6286	0.7191	4.0382
Component 34	4.1297	0.3018	1.511
Component 35	26.4413	0.4783	4.6967
Component 36	31.0534	0.5882	4.3471
Component 37	12.1588	0.4719	2.775
Component 38	7.8798	0.2221	2.1146
Component 39	38.3512	0.3233	4.8293
Component 40	13.7412	0.3641	3.202
Component 41	67.4687	0.2266	6.627
Component 42	95.4623	0.9337	7.8054
Component 43	9.2017	0.4994	2.5275
Component 44	4.773	0.2023	1.4458
Component 45	55.2443	0.5589	6.2689
Component 46	62.5792	0.6164	6.9172
Component 47	98.422	0.9146	8.0089
Component 48	36.8209	0.481	5.3544
Component 49	71.2181	0.846	6.3854
Component 50	54.7276	0.6747	6.1687
Component 51	44.7344	0.882	5.324
Component 52	98.577	0.7879	7.9353
Component 53	17.8802	0.4953	3.1026
Component 54	533.552	0.7055	19.1752
Component 55	201.5543	0.4031	10.4089
Component 56	130.9305	0.6677	7.3363
Component 57	209.4328	0.89	11.446
Component 58	36.0109	0.3539	4.0889
Component 59	29.2366	0.119	4.1455
Component 60	93.231	0.5232	7.7341
Average	61.6917	0.5442	5.25
Time	2074(s)		

Table 5.24: SDS-4 ARIMAX errors per component, components 31-60.

Comparison

Using the average metrics from Table 5.21 and combining them with the average metrics from Table 5.24 to construct Fig. 5.47. Comparing the errors with the previously discussed methods it is plain to see that ARIMAX has comparable errors to the multifidelity model. However it is highly depended on the data it was trained with and there is no guarantee it will as well in other cases. The multifidelity framework can always rely on the high fidelity data for corrections. The framework remains the method with the best performance. Moreover, computation time is what sets the multifidelity framework apart from ARIMAX. Fig. 5.48 illustrates that ARIMAX required 433% more time to fit and predict. While ARIMA, the autoregressive methodology has complexity $O(n)$, things get more complicated as exogenous variables are included. In particular, SDS-4 uses 59

explanatory variables to predict the temperature of 1 component. This is a large number of variables and are expected to slow the process down considerably.

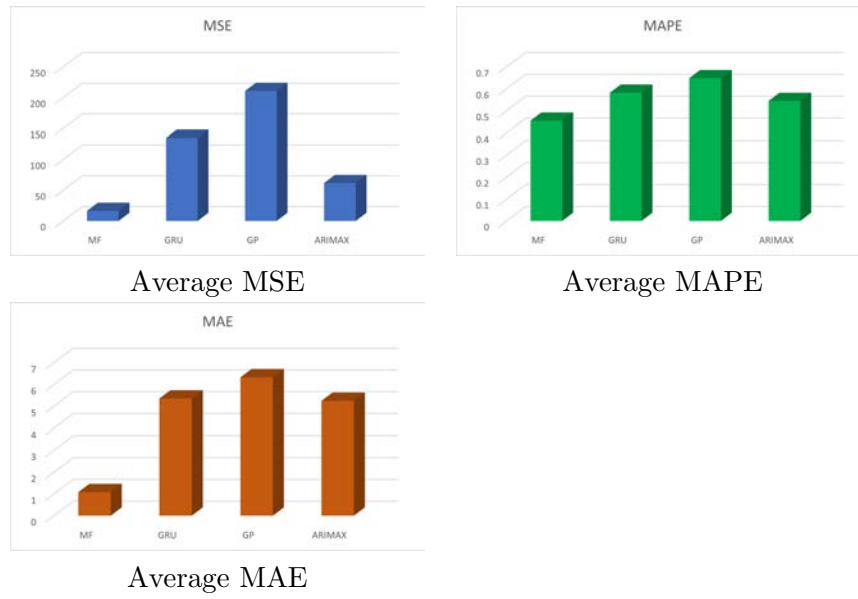


Figure 5.47: SDS-4 average metric comparison, including ARIMAX.

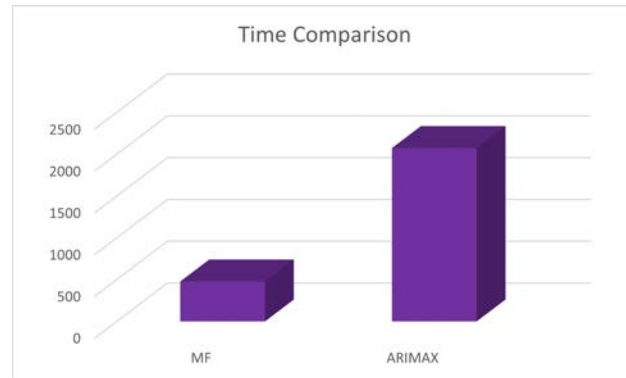


Figure 5.48: MF and ARIMAX time comparison for SDS-4 data.

5.4.8 Robustness Study

To investigate the robustness of the framework to both the number and the location of high fidelity points, it was ran for $N = 100$ scenarios with the number of high fidelity points randomly chosen between 13% and 17% and at random locations.

Robustness Metrics			
Maximax			
<i>Method</i>	\bar{e}_{MSE}	<i>stdv</i>	\bar{e}_{MAE}
MF	0.01345	0.25362	0.45081
GP	0.00168	0.06649	0.08698
Laplace's Principle of Insufficient Reason			
MF	0.02873	0.34498	0.72131
GP	0.00482	0.11664	0.1749
Maximin			
MF	0.07735	0.45478	1.07199
GP	0.02362	0.26192	0.37816
Mean-Variance			
MF	0.00513	0.33427	0.69066
GP	0.00008	0.06878	0.1126

Table 5.25: Robustness to HFM point placement, SDS-4.

In the case of SDS-4, the multifidelity framework again is consistently more robust across all robustness metrics. As a reminder, higher robustness values indicate better performance. The MF robustness values are higher for maximax, describing the best-case scenario, maximin, expressing the worst-case scenario, Laplace's principle of insufficient reason, describing the average behaviour and finally the mean-variance, which describes how the behaviour varies across scenarios. Concerning the mean-variance, the multifidelity framework does not deviate as much as the Gaussian process. The MSE errors per scenario can be seen in Fig. 5.49, 5.50. The multifidelity framework's MSE error is an order of magnitude smaller than the Gaussian process' error.

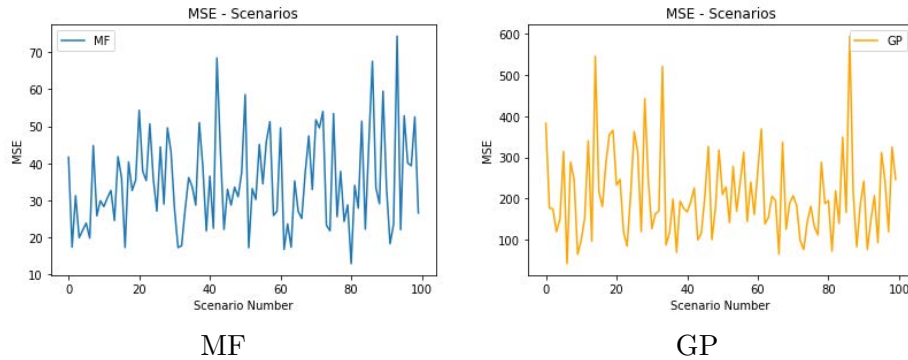


Figure 5.49: MSE per scenario for SDS-4.

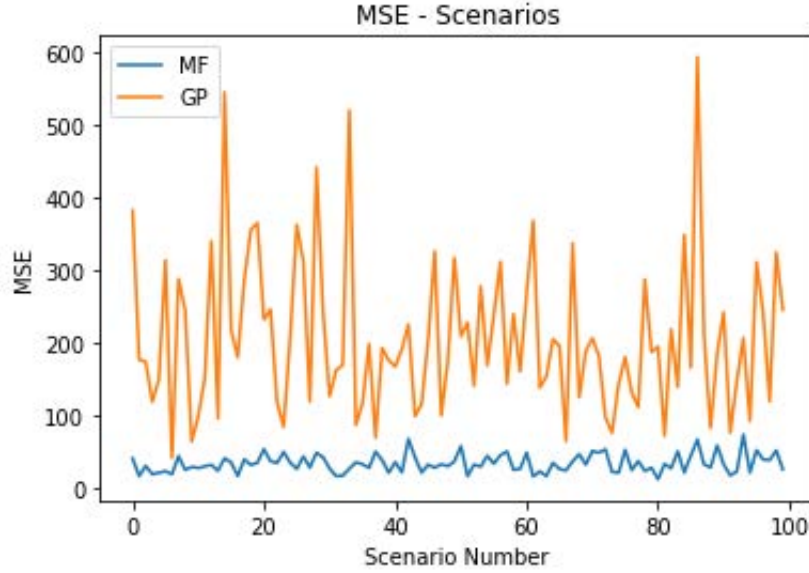


Figure 5.50: MF and GP MSE errors per scenario juxtaposed for SDS-4.

5.4.9 Experiment Summary

SDS-4 data is challenging. Due to a number of operational modes, the thermal response of the satellite varies, creating very diverse graphs. Even in a case like this, the simulation results consistently show that the multifidelity framework works better than the "traditional" approaches. Error metrics, as well as the uncertainty, are reduced in every simulation by the multifidelity network. About real-time simulation, the sliding window keeps computational times manageable and captures the behaviour of all components better than the GRU and the Gaussian process, with a lower uncertainty too. According to the robustness study, the effectiveness of the algorithm remains relatively consistent across the 100 random scenarios. On the other hand, the Gaussian process' response is erratic, with large errors and highly depended on the high fidelity points' placement and number.

5.5 SDS-4 Using the Entire Dataset

5.5.1 Multifidelity Framework Testing - Entire SDS-4 Dataset

The only difference of the setup in Fig. 5.51 in this subsection from the setup in Fig. 5.39 is the number of data used in training. All $N = 8999$ timesteps are used, with 1500 data points, or around 16%, sampled and taken as the high fidelity model.

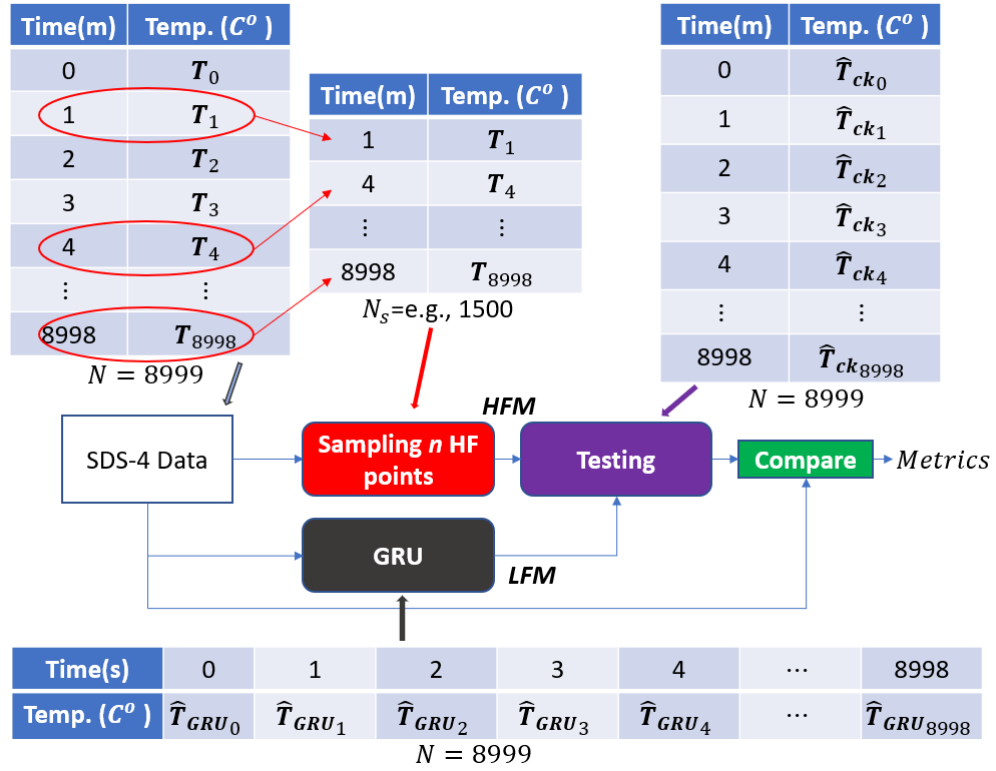


Figure 5.51: Testing approach summary for SDS-4's entire dataset. \mathbf{T} : vector of temperatures, $\hat{\mathbf{T}}_{GRU}$: vector of GRU estimated temperatures, $\hat{\mathbf{T}}_{ck}$: vector of cokriging estimated temperatures, N : number of rows.

5.5.2 Results GRU

Fig 5.52 illustrates the response of the GRU to two example components. The GRU captured well the shape of the graph making it suitable to act as a LFM. The rest of the figures can be found in Appendix A.7.

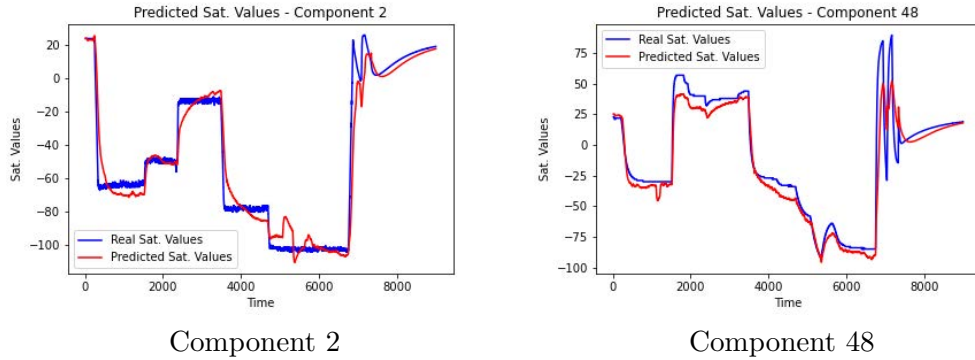


Figure 5.52: Examples of the GRU estimation for SDS-4's entire dataset. y-axis is temperature $T(C^\circ)$ and x-axis is time $t(s)$.

5.5.3 Results Uniform HF Data Points

In this simulation, 16% out of the 9000 data points were added as high fidelity data points. The GRU prediction previously shown was used as the LFM model and noiseless data points were used as the HFM values. In this subsection, the

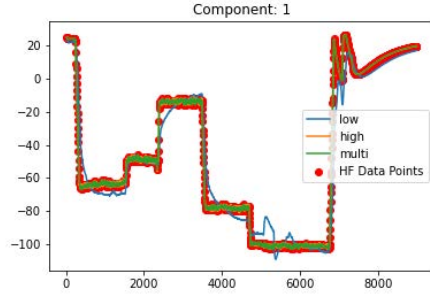
HFV values were taken at even intervals. Tables 5.26, 5.27 summarize the errors and standard deviation of the multifidelity network.

SDS with Uniform High Fidelity Data 1-30				
Components	MSE	MAPE	MAE	sdtv
Component 1	3.48	0.348	3.602	4.603
Component 2	2.756	0.213	3.148	4.323
Component 3	2.78	0.127	0.653	4.267
Component 4	2.41	0.192	0.601	4.608
Component 5	3.11	0.233	0.577	1.516
Component 6	2.05	0.11	0.42	1.346
Component 7	1.88	0.22	0.407	1.626
Component 8	3.95	0.106	0.78	3.138
Component 9	4.94	0.103	0.809	1.63
Component 10	6.49	0.15	1.105	2.974
Component 11	5.1	0.46	0.781	1.608
Component 12	4.26	0.766	1.484	2.312
Component 13	2.56	0.51	0.471	1.288
Component 14	1.95	0.197	0.393	1.338
Component 15	2.23	0.277	0.463	1.417
Component 16	2.54	0.133	0.485	1.434
Component 17	1.29	0.459	0.328	1.393
Component 18	1.3	0.199	0.33	1.231
Component 19	1.56	0.248	0.378	1.288
Component 20	1.72	0.219	0.346	1.39
Component 21	1.42	0.102	0.328	1.194
Component 22	1.59	0.109	0.352	1.29
Component 23	1.44	0.108	0.328	1.412
Component 24	1.22	0.211	0.293	1.216
Component 25	1.79	0.104	0.388	1.508
Component 26	1.53	0.138	0.336	1.398
Component 27	1.69	0.287	0.361	1.7
Component 28	1.1.37	0.284	0.588	3.911
Component 29	1.88	0.181	0.395	1.302
Component 30	1.48	0.081	0.344	1.285

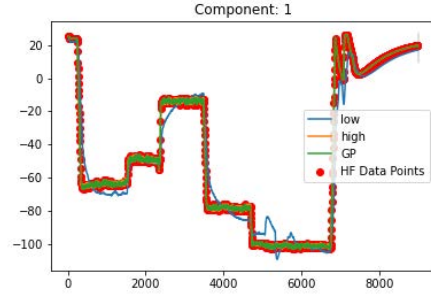
Table 5.26: SDS-4 using the entire dataset. Errors per component 1-30. Uniform data.

SDS with Uniform High Fidelity Data 31-60				
Components	MSE	MAPE	MAE	sdtv
Component 31	1.55	0.186	0.365	1.354
Component 32	1.68	0.553	0.385	1.381
Component 33	1.47	0.604	0.326	1.235
Component 34	1.21	0.021	0.3	1.153
Component 35	1.35	0.022	0.323	1.167
Component 36	1.49	0.023	0.332	1.315
Component 37	2.76	0.107	0.502	1.344
Component 38	1.89	0.025	0.377	1.289
Component 39	1.75	0.032	0.378	1.171
Component 40	1.81	0.266	0.387	1.216
Component 41	7.96	0.106	1.22	5.862
Component 42	7.68	0.495	1.115	6.032
Component 43	9.71	0.299	1.193	6.406
Component 44	8.07	0.099	1.104	6.221
Component 45	2.07	0.104	1.368	4.532
Component 46	2.475	0.106	1.547	4.764
Component 47	2.544	0.122	1.517	4.684
Component 48	6.15	0.666	0.97	5.373
Component 49	7.49	0.499	1.035	5.923
Component 50	7.39	0.422	1.043	5.849
Component 51	5.21	0.127	0.841	5.724
Component 52	5.25	0.292	0.895	5.508
Component 53	6.11	0.101	0.944	6.111
Component 54	1.196	0.303	0.998	9.414
Component 55	1.164	0.108	0.1.081	9.818
Component 56	1.343	0.525	0.1.145	9.715
Component 57	2.27	0.467	0.0.64	2.154
Component 58	1.52	0.303	0.411	1.119
Component 59	2.79	0.339	0.65	2.147
Component 60	6.67	0.165	0.799	4.084
Average	5.1	0.285	0.74	3.074

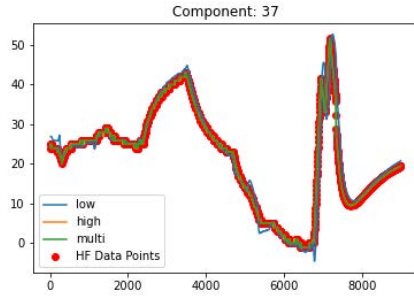
Table 5.27: SDS-4 using the entire dataset. Errors per component 31-60 and averages. Uniform data.



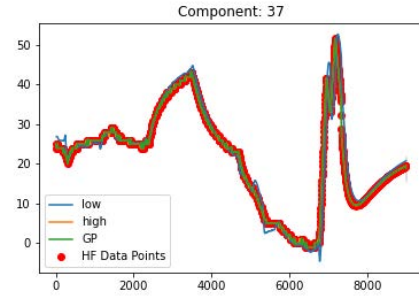
MF - Component 1



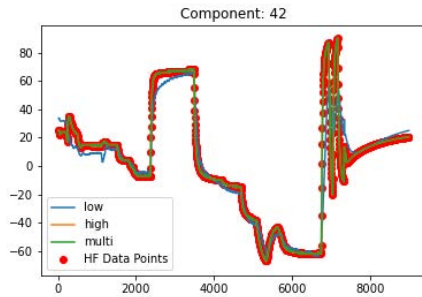
GP - Component 1



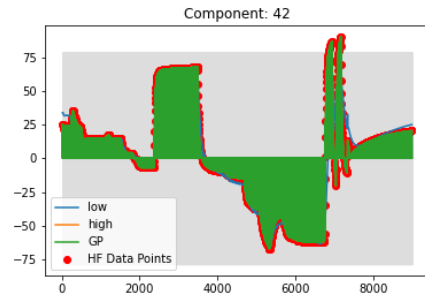
MF - Component 37



GP - Component 37



MF - Component 42



GP - Component 42

Figure 5.53: SDS-4 data for SDS-4's entire dataset. Multifidelity vs Gaussian process. Uniform high fidelity data points. Blue lines (-) indicate the LFM, orange lines (-) the HFM, green lines (-) the multifidelity estimation and red dots (•) the HFM data points used and the gray area (■) the confidence interval for components 1 and 42.

Overall Average Performance			
<i>Metrics</i>	<i>MF</i>	<i>GRU</i>	<i>GP</i>
\bar{e}_{MSE}	5.1	82.695	75.8
\bar{e}_{MAPE}	0.285	0.553	0.588
\bar{e}_{MAE}	0.74	4.31	1.39
$\bar{\sigma}$	0.3074	n/a	1.616
\bar{e}_{MSE} change	n/a	-94%	-93%
\bar{e}_{MAPE} change	n/a	-49%	-49%
\bar{e}_{MAE} change	n/a	-83%	-40%
$\bar{\sigma}$ change	n/a	n/a %	-80.9%

Table 5.28: Error & time metrics overall performance for SDS-4 data using the entire dataset.

Fig. 5.53 illustrates the multifidelity framework’s output against the single fidelity Gaussian process for a three example components. The rest of the figures can be found in Appendix A.8. As expected, since an adequate number of high fidelity data points were used and were uniformly distributed across the timeseries, the Gaussian process and the multifidelity framework have similar responses. However, the Gaussian process can occasionally fail to produce results without tuning. The multifidelity framework does not require online tuning. Table 5.28 summarizes the metrics. While the framework shows stronger average performance with smaller errors and smaller standard deviation, this is due to some Gaussian processes failing. For components where the Gaussian process works, the performance is the same.

5.5.4 Results Random HF Data Points

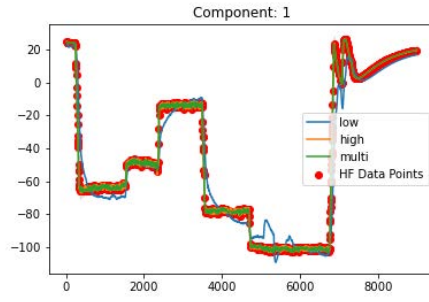
In this simulation, 16% out of the 8999 timesteps were added as high fidelity data points. The location of the high fidelity data points were chosen at random. The GRU prediction previously shown is again the same as in the previous test and noiseless data points were used as the HFM values. The errors and standard deviation for each component are presented in Tables 5.29, 5.30.

SDS with Uniform High Fidelity Data 1-30				
Components	MSE	MAPE	MAE	sdtv
Component 1	15.031	0.264	4.019	0.6882
Component 2	12.123	0.544	3.511	0.6472
Component 3	3.654	0.464	1.068	0.6394
Component 4	3.381	0.217	1.048	0.6533
Component 5	12.23	0.266	0.521	0.2401
Component 6	9.32	0.26	0.373	0.1979
Component 7	9.4	0.147	0.425	0.258
Component 8	24.11	0.027	1.054	0.4776
Component 9	18.41	0.053	1.031	0.2408
Component 10	3.09	0.059	1.198	0.4653
Component 11	18.73	0.039	1.088	0.2672
Component 12	18.38	0.015	1.061	0.2303
Component 13	9.81	0.033	0.337	0.2045
Component 14	9.26	0.289	0.271	0.2028
Component 15	9.46	0.079	0.316	0.2191
Component 16	9.47	0.028	0.396	0.2117
Component 17	6.35	0.193	0.262	0.203
Component 18	6.41	0.353	0.294	0.175
Component 19	6.98	0.295	0.365	0.187
Component 20	9.61	0.14	0.329	0.1994
Component 21	6.02	0.128	0.164	0.1735
Component 22	6.69	0.328	0.29	0.2091
Component 23	6.33	0.103	0.25	0.218
Component 24	6.31	0.375	0.233	0.1866
Component 25	6.98	0.194	0.34	0.2308
Component 26	6.75	0.145	0.285	0.2187
Component 27	9.05	0.189	0.332	0.256
Component 28	6.358	0.327	0.613	0.522
Component 29	6.86	0.389	0.342	0.1943
Component 30	6.67	0.202	0.307	0.1869

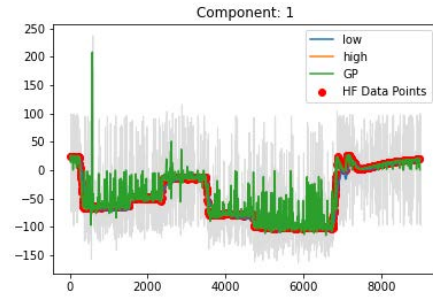
Table 5.29: SDS-4 using the entire dataset. Errors per component 1-30. Random data.

SDS with Uniform High Fidelity Data 31-60				
Components	MSE	MAPE	MAE	sdtv
Component 31	6.82	0.11	0.352	0.1931
Component 32	6.86	0.188	0.349	0.2012
Component 33	6.73	0.351	0.285	0.1765
Component 34	6.22	0.225	0.252	0.1747
Component 35	6.36	0.32	0.27	0.1695
Component 36	6.59	0.249	0.297	0.196
Component 37	9.89	0.136	0.453	0.2111
Component 38	9.44	0.316	0.371	0.1886
Component 39	6.91	0.36	0.317	0.1687
Component 40	6.66	0.101	0.282	0.1781
Component 41	9.184	0.295	1.932	0.8651
Component 42	9.439	0.462	1.922	0.8905
Component 43	9.766	0.481	1.963	0.9354
Component 44	9.637	0.294	1.923	0.9149
Component 45	21.699	0.187	2.339	0.5989
Component 46	27.325	0.111	2.647	0.6084
Component 47	24.619	0.54	2.562	0.6187
Component 48	12.998	0.465	2.034	0.7689
Component 49	15.591	0.115	2.069	0.8557
Component 50	15.693	0.569	2.01	0.8414
Component 51	18.976	0.299	1.83	0.8279
Component 52	24.954	0.139	1.975	0.7734
Component 53	24.783	0.228	2.083	0.8772
Component 54	27.661	0.21	2.231	0.3563
Component 55	27.274	0.67	2.199	0.4171
Component 56	27.604	0.705	2.382	0.4036
Component 57	15.44	0.306	0.91	0.3099
Component 58	6.45	0.199	0.343	0.1519
Component 59	18.08	0.247	0.708	0.296
Component 60	9.929	0.276	1.481	0.607
Average	12.78	0.31	1.02	0.443

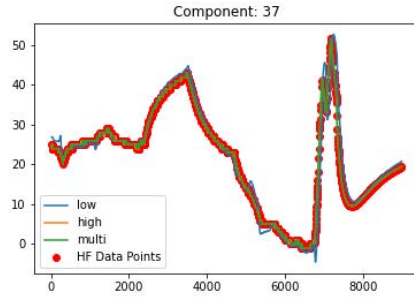
Table 5.30: SDS-4 using the entire dataset. Errors per component 31-60 and averages. Random data.



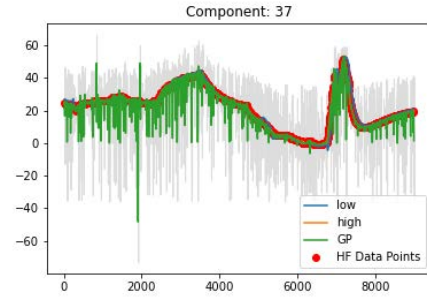
MF - Component 1



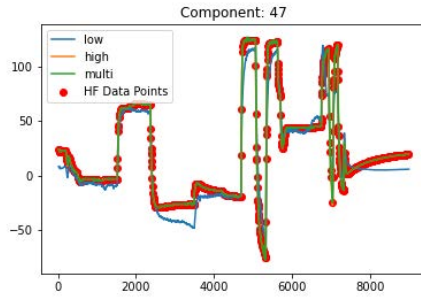
GP - Component 1



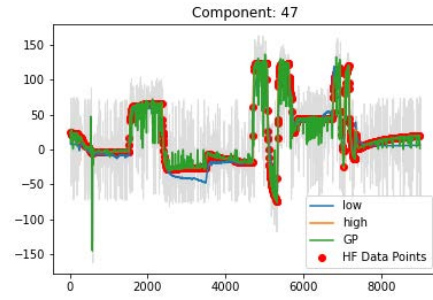
MF - Component 37



GP - Component 37



MF - Component 47



GP - Component 47

Figure 5.54: SDS-4 data, SDS-4 data for SDS-4's entire dataset. Multifidelity vs Gaussian process. Randomly placed high fidelity data points. Blue lines (-) indicate the LFM, orange lines (-) the HFM, green lines (-) the multifidelity estimation and red dots (•) the HFM data points used and the gray area (■) the confidence interval for components 1, 37 and 47.

Overall Average Performance			
<i>Metrics</i>	<i>MF</i>	<i>GRU</i>	<i>GP</i>
\bar{e}_{MSE}	12.78	82.695	113.553
\bar{e}_{MAPE}	0.31	0.553	0.753
\bar{e}_{MAE}	1.02	4.31	3.95
$\bar{\sigma}$	0.443	n/a	8.321
\bar{e}_{MSE} change	n/a	-84%	-89%
\bar{e}_{MAPE} change	n/a	-43%	-59%
\bar{e}_{MAE} change	n/a	-76%	-74%
$\bar{\sigma}$ change	n/a	n/a %	-95%

Table 5.31: Error & time metrics overall performance for SDS-4 data using the entire dataset.

In the case where high fidelity points are placed in random, the multifidelity framework consistently outperforms the single fidelity Gaussian process. The average performance is summarised in Table 5.31. The magnitude of the errors did increase due to the high fidelity points not being uniform but the relative performance of the framework against the Gaussian process became stronger. Fig. 5.54 presents three example components. The rest of the figures can be found in Appendix A.9. If high fidelity points are missing from key points, the Gaussian process' error and standard deviation spike.

5.5.5 Experiment Summary

The purpose of these simulations was to demonstrate that the multifidelity framework works regardless of the amount of data used. The conclusions are very similar to the experiments using 1000 data. Error metrics, as well as the uncertainty, are reduced in every simulation by the multifidelity network. The Gaussian process remains highly dependent on the placement of the high fidelity points. On top of that, the Gaussian process occasionally fails to produce a prediction. On the other hand the multifidelity frameworks always manages to produce an estimation without the need of online tuning or re-estimations.

Chapter 6

Discussion & Conclusion

6.1 Discussion

The purpose of this work was to survey current modelling methodologies and propose a framework for the use within a Digital Twin architecture. The Digital Twin is a dynamic representation of physical entities with their functions, behaviours, and rules. It uses a knowledge based to build a framework, a digital model that describes the physical object and an analytics component to assess performance. The Digital Twin must mimic the behaviour of the physical entity accurate and in real time. This implies the following requirements:

The framework should be able to

1. perform high accuracy simulation,
2. with short computation times to support an online stream of data,
3. be multimodal,
4. be application independent and
5. provide uncertainty quantification for risk assessment.

The competing requirements, accuracy and short computation times, are usually resolved in literature through some type of reduced order modelling. Proper Orthogonal Decomposition (POD) is a method often employed in Digital Twin applications, including applications in the field of aerospace. However, research on the Digital Twin in aerospace remains limited. However, POD has a significant drawback. While POD, in theory, is model independent, it is looking for the direction of maximum variance and in some cases, e.g., turbulence, there is no direction of maximum variance, returning mixed results. Multifidelity modelling was investigated as an alternative approach to reduced order modelling. A low fidelity Gated Recurrent Unit (GRU) is combined by cokriging with high fidelity data obtained either through an expensive FE/FD simulation, or sensors. When working with online data, low fidelity predictions are corrected to estimate the high fidelity prediction. However, whenever a set of high fidelity prediction is generated, the model is updated for a minimum computational cost, since it only requires refitting the model with the new data.

In Section 2, Fig. 3.3 contained a diagram describing the themes constituting the Digital Twin. This work, aims to fill the role of the Realisation and Virtual Entity (VE) in Fig. 6.1. In other words, it adjusts the virtual model based on information from the physical model and the framework's output serves as the digital model or virtual entity. Metrology refers to the way measurements are obtained, e.g., sensors, and are out of the scope of this study. An important topic

of metrology is optimal sensor placement studies. Our work used measurements from performed thermal vacuum tests in which a large number of sensors were used, more than what would be used in an actual space mission.

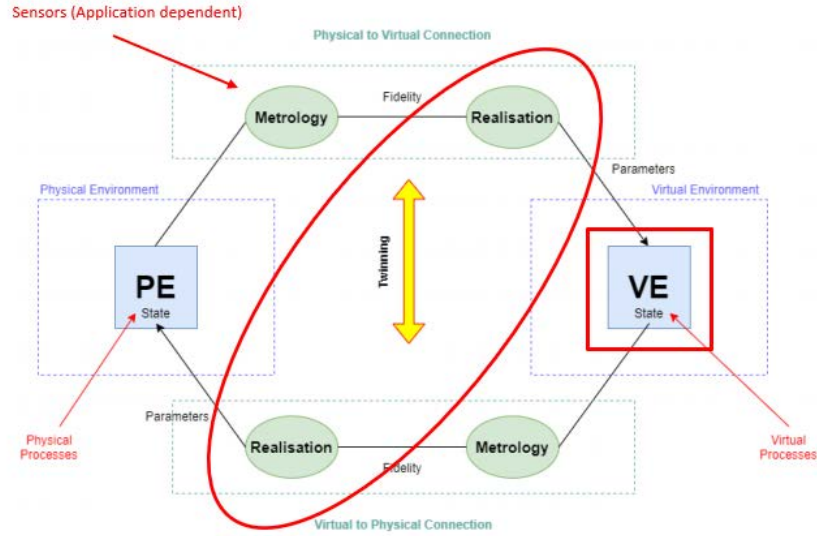


Figure 6.1: A diagram of the Digital Twin. In red are the components using the multifidelity framework.

For the experiments, both simulated and real data, obtained through thermal vacuum were used to test the multifidelity framework against "traditional" methods for timeseries modelling. Extra attention was given when comparing the framework to Gaussian processes fit on high fidelity data. This is because GPs are the single fidelity counterpart to cokriging. This comparison serves to illustrate the benefits of not relying on a single source of information. GPs rely heavily on the placement and number of the high fidelity data points and wherever one of these points is missing the Gaussian process behaves unpredictably with a large uncertainty. Moreover, unless a large number of high fidelity data points are used, the accuracy of GPs is lower than the multifidelity's framework. These extra points will demand even more computational resources that may not be available.

Another method the framework was benchmarked against was artificial neural networks. ANNs are very popular for system modelling because they are universal approximators and are able to detect interdependencies hidden in the inputs. As the state of the physical entity evolves through time, the data describing it is timeseries. Recurrent neural networks show strong performance for timeseries data, especially the LSTM network and its simplified version the Gated Recurrent Unit (GRU). The two main drawbacks of artificial neural networks is that a) they need a lot data and time for training and at the same time their parameters are decided by the designer, often through trial and error, and b) they are black box algorithms and do not provide uncertainty quantification. Across all simulations, the GRU provides predictions of moderate to good accuracy, at short computational times, and was used as the low fidelity model for the proposed multifidelity framework. It was also used as a basis for comparison with other methods. However, a neural network, on its own, cannot be used for Digital Twin modelling for a few reasons.

1. The lack of uncertainty quantification is a major issue. As communication is a major component of the Digital Twin, there needs to be some sort of feedback concerning the level of trust of each measurement. These values will be forwarded to another system, e.g., PHM, which might not be able to distinguish an anomalous state from a simple bad prediction. Uncertainty quantification for neural networks is a difficult task that cannot be done online.
2. The quality of an ANN's prediction is heavily dependent on the data used for its training. Most of the data available usually comes from laboratory testing, e.g., thermal vacuum testing. These tests however might not reflect the space environment. Imperfections in the training data will affect the accuracy of the predictions.
3. If a component fault occurs while the spacecraft is in space and the neural network hasn't been trained with data which includes that fault the predictions could not be trusted. There is no process that can adjust its estimation to the new situation.

The final "traditional" method the framework was compared to is an autoregressive model, in particular the multivariate version, ARIMAX. This method uses past values, also called lags, along with other explanatory variables to predict future values. ARIMAX produced good estimations for all experiments, using only a small number of lags. The two main drawbacks concern the method's inability to predict turning points accurately and its computational complexity. Calculating the coefficients for the explanatory variables is a difficult task. In the case of SDS-4 with 60 components/sensors, each variable to predict has 59 explanatory variables. The computational load for this many variables is very high and as a result the computation time required was much higher than the multifidelity framework's.

Three datasets were used in this study, each of them utilized to draw different conclusions.

- ISSL6U-1 simulated data were obtained with Thermal Desktop. This study aims to provide a simple test case to demonstrate, in principle, that the framework works.
- ISSL6U-1 simulated data, augmented by simulated power data. The extra power data is used to demonstrate the multimodal aspects of this framework.
- SDS-4 data obtained through thermal vacuum testing. This data is used to demonstrate the framework with real data, with behaviours and trends that would appear in a real satellite.

The multifidelity framework performed well across all datasets and was able to capture the real system behaviour with a high degree of accuracy. Moreover, only the multifidelity framework managed to satisfy the requirements outlined early in this subsection. On top of that it is robust to high fidelity points placement and number, scoring high robustness values in both errors and standard deviation against the single fidelity Gaussian process, calculated across a number of testing scenarios.

6.2 Limitations

In this section the limitations of this study are presented and discussed. The main limitations are data related and can be summarised as:

- *Lack of real data for multiple satellites.* This framework was tested using data describing two small satellites, ISSL6U-1 and SDS-4. However, the ISSL6U-1 data were simulated and might not reflect reality as accurately as a live test or data from previous missions. The SDS-4 satellite data were obtained through thermal vacuum testing.
- *Multimodality was demonstrated with simulated data.* The multimodality potential of the framework is demonstrated through simulated data supplementing ISSL6U-1 data. Ideally, real data of multiple subsystems would be available. However due to the complexity of thermal vacuum testing, measurements are restricted to temperature only.
- *Lack of physical implementation.* The framework was not tested with a physical system in the lab and in a real-time manner.
- *Robustness studies are limited because of the computational complexity.* Running multiple tests for multiple scenarios is a very resource intensive process and takes a personal laptop days to run.

6.3 Contributions

The main contributions of this body of work are summarised as,

1. Proposed a framework that can potentially be used as an alternative to reduced order models in a Digital Twin architecture.
2. Tested the proposed framework with simulated satellite thermal data.
3. Tested the multimodality aspect of the proposed framework with simulated satellite thermal and power data.
4. Tested the framework with data obtained through thermal vacuum testing.
5. Evaluated the robustness of the framework to high fidelity data point placement.
6. Evaluated the use of recurrent neural networks to satellite thermal modelling.
7. Discussed what different sources of information offer to multifidelity modelling.
8. Evaluated the proposed method using a practical thermal prediction experiment for real satellites.
9. A review of Digital Twin architecture in the context of space engineering.
10. A review of multifidelity approaches in the context of space engineering.

6.4 Future Research

Multifidelity modelling seems to be a promising methodology for both speeding up high fidelity simulations using low fidelity data and estimating satellite state values in real time as part of a Digital Twin architecture. The next steps that are recommended are the following:

- A physical implementation should be conducted, with a physical system in real-time.

For a full demonstration of the framework's capabilities, ideally, a physical implementation is needed. Even though the data used were obtained through physical testing, the framework could benefit from a real time test with a physical satellite in a thermal vacuum chamber. A proposed test configuration would be to use placed sensor measurements as the high fidelity data and use the GRU to predict future values. Running the sliding window in this case would involve using high fidelity measurements for the part of the window corresponding to past values and only low fidelity measurements for the part of the window corresponding to future values. It is not required to use high fidelity data for future predictions if not available. Of course future values could be supplied by a finite element simulator. If the framework is used in this set-up, the future quantities of interest will be available with sensor-level accuracy.

- A demonstration tool should be developed, e.g., a script that presents the data in a GUI that designers and operators can interact with.

Some preliminary has been done. In a real application, plots are unnecessary. Our approach is illustrated in Fig. 6.2 In practice, a background process, independent of the main program, will be exporting the results to the next stage or subsystem. While the data is being exported at a desired rate, the main program will be calculating the next window. When finished, it will check if the exporting process of the previous window has finished. If not, it will wait. When it is ready, the system will start exporting the new window and start calculating the one after that. A major advantage of this framework is that each measurement can be accompanied by its standard deviation.

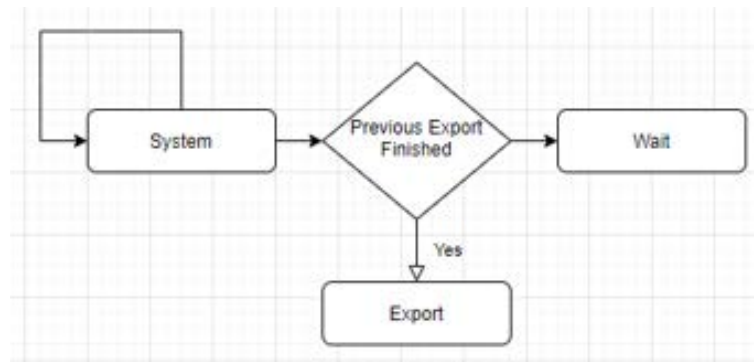


Figure 6.2: Diagram for live computation.

In Fig. 6.3 an example output is shown for the ISS6U-1 data and its 6 component. Each second, a vector of values appears on the screen, along with its standard deviation.


```

time(s) :   c1(degC)   c2(degC)   c3(degC)   c4(degC)   c5(degC)   c6(degC)
30.0s : [16.80829064 17.930192  19.689728  18.370076  17.930192  21.449264 ]
Stdv :  [1.135e-04 2.580e-05 2.720e-05 2.580e-05 2.590e-05 2.580e-05]

31.0s : [16.80021864 17.930192  19.689728  18.370076  17.930192  21.449264 ]
Stdv :  [1.139e-04 2.580e-05 2.770e-05 2.580e-05 2.590e-05 2.580e-05]

32.0s : [16.79214652 17.930192  19.689728  18.370076  17.930192  21.449264 ]
Stdv :  [1.145e-04 2.580e-05 2.830e-05 2.580e-05 2.590e-05 2.580e-05]

33.0s : [16.78407428 17.930192  19.689728  18.370076  17.930192  21.449264 ]
Stdv :  [1.155e-04 2.580e-05 2.890e-05 2.580e-05 2.590e-05 2.580e-05]

34.0s : [16.77600192 17.930192  19.689728  18.370076  17.930192  21.449264 ]
Stdv :  [1.169e-04 2.580e-05 2.960e-05 2.580e-05 2.590e-05 2.580e-05]

35.0s : [16.76792944 17.930192  19.689728  18.370076  17.930192  21.449264 ]
Stdv :  [1.187e-04 2.580e-05 3.030e-05 2.580e-05 2.600e-05 2.580e-05]

36.0s : [16.75985684 17.930192  19.689728  18.370076  17.930192  21.449264 ]
Stdv :  [1.212e-04 2.580e-05 3.110e-05 2.580e-05 2.600e-05 2.580e-05]

```

Figure 6.3: GUI with the state estimation per second.

A more advanced GUI for thermal modelling should include a model of the satellite being simulated, changing in real time the colour of the panels depending on the framework's estimations. This fits better, in human terms, with our expectations of the Digital Twin. If other subsystems are included then different view modes could demonstrate different aspects of the satellites along with their respective estimations.

- The framework can be compared with more methods. There is a large number of methods and approaches in the literature that could be used for comparison and demonstration.

The methods used in this work were chosen because they are a few of the most popular techniques in modelling. However, these techniques are generic and application independent as this was one of the requirements of the framework. It would be interesting to compare the multifidelity framework against more specialised methods and for particular subsystems to assess its performance.

- The multimodality aspect should be demonstrated with a multi subsystem test data of a real satellite instead of simulated data.

The multimodal aspect of this work was demonstrated through simulated power and thermal data for the ISS16U-1 satellite. Data in the field of space engineering are hard to get. However, it would be beneficial to conduct simulations using data collected through tests involving multiple subsystems, e.g., a thermal vacuum test which includes sensors measuring battery and solar power output and/or placing strain gauges on the structure to measure any deformations. These simulations will validate even further the ability of the framework to work with multimodal systems with great accuracy.

- The framework could be developed in C/C++ instead of python to improve speed. Parallel processing can also yield computational gains when using a large amount of data.

The framework in this work was developed using Python 3.6. Python is a user-friendly interpretable programming language. It is very easy to use but it comes at the cost of computational speeds because it amplifies the number of actual CPU instructions required in order to perform a given

statement. On top of that Python, almost always runs in a virtual machine or bytecode interpreter. On the other hand, C/C++ is much faster due to its bare-bones approach on programming. However, due to being a lower level language, it would increase the workload required by the developer. A conversion to C/C++ is recommended for implementation on a real system.

In parallel processing multiple tasks are completed in a parallel manner as in Fig. 6.4. In systems with multiple cores, the tasks can be allocated to different cores greatly reducing computation time.

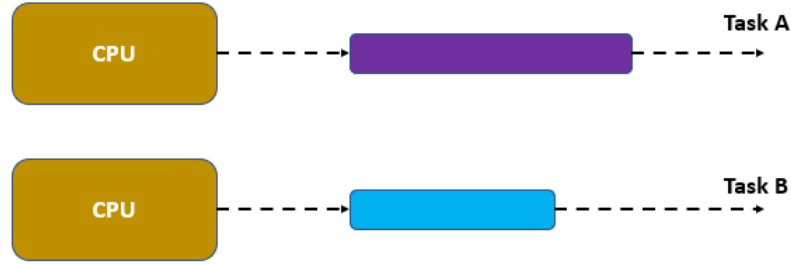


Figure 6.4: Parallel processing diagram.

In the framework, after the GRU calculated a vector of future values, each prediction is combined with its high fidelity counterpart by cokriging. The cokriging for each component can be assigned to a core. For example, the computer that run the simulations for this work has 6 cores. Therefore it can run cokriging for 6 components at a time. Nevertheless, distributing tasks comes with an overhead cost, e.g., transferring copies of the required data to the cores. As a general rule simpler tasks are less efficient when calculated using parallel processing. When the multifidelity framework is used with a relatively small number of values, as with the sliding window, serial processing is recommended. However, when applying cokriging to thousands of data, parallel processing can be helpful. The framework should be optimized to the available hardware.

6.5 Conclusion

This thesis is the continuation of previous work [1, 2]. It presents a review of the Digital Twin and multifidelity modelling in space applications. The work on Digital Twin in aerospace is limited. Especially rare are the papers including implementations on real systems. While a definition of the Digital Twin has not been determined, literature suggests that it is a digital or virtual model of information that mimics the characteristics of a physical object, online. As outlined in Section 1.1, this implies that the requirements of such a framework include two usually competing requirements, high accuracy and manageable computational complexity. Additionally, it must provide uncertainty quantification to run analytics, not be application specific and can be updated online.

Additionally, the differences between data driven and model based methods for the construction of a Digital Twin is examined in Section 3. While physics based methods require a small amount of data and are interpretable, they cannot account for any unmodelled physics. Either due to lack of knowledge or due to complexity, unmodelled physics reduce the maximum accuracy of the system. On the contrary, data driven models are good at capturing the full physics of

the system at the cost of requiring a lot of data and sacrificing interpretability. Multifidelity modelling enables the use of both data driven and physics based models, in a hybrid architecture to mitigate their drawbacks.

This study discusses how multifidelity modelling can enable a Digital Twin architecture. Multifidelity methods can leverage high fidelity data to improve the accuracy of a computationally cheap, low fidelity model. Research with aerospace applications often use Reduced Order Modelling, usually constructed with a Proper Orthogonal Decomposition basis, because it is an application independent approach, can be updated online with new data and mitigates the computational burden. However, as discussed in Chapter 2, it is unsuitable in some situations, particularly broadband phenomena, where there is no direction of maximum variance. As an alternative, an online multifidelity framework for small satellites is proposed. After reviewing each individual component i.e., the low fidelity surrogate model, the high fidelity model and the multifidelity methods; it is observed that individually they can function effectively. The framework uses a GRU as a low fidelity model as a surrogate for the physical system and either a finite element or live data for the high fidelity model. Cokriging is used to fuse the two models and provide a multifidelity estimate, as well as estimate uncertainty. If small windows in time are taken, the computational load remains manageable and can be adjusted online. It can potentially provide a complete analysis of a physical system, including unmodelled physics and subsystem interdependencies, and satisfy the described requirements.

Finally, the framework additionally outputs an uncertainty estimation, which is highly sought after in the space industry. Moreover I aim to promote the use of machine learning in space applications as the technology has matured and the way is open for its use in more high risk environments.

The target application was thermal analysis of ISSL6U-1 and SDS-4, two 6U and 5U satellites respectively. The framework was benchmarked against three "traditional" methods, artificial neural networks, the Gated Recurrent Unit in particular, Gaussian processes and an autoregressive method, ARIMAX. The ISSL6U-1 was supplemented with simulated data for power components to test the multimodality aspect of the framework. The SDS-4 data were obtained through thermal vacuum testing conducted by the Japan Aerospace Exploration Agency (JAXA).

The framework shows consistently better accuracy across all simulations against the GRU and better or similar against the Gaussian process and ARIMAX. However, drawing information from multiple sources, the multifidelity framework is more robust than Gaussian processes to noisy data and missing values. The multifidelity framework is more costly than both the GRU and the Gaussian process, however it is less computationally intensive than ARIMAX. Finally, in order to test real-time computation, a sliding window technique was applied. Moving the window forward in time, the prediction of future values can be predicted, one window at a time. The multifidelity framework showed consistently better performance than the other methods.

Publications

- Kontaxoglou, Anastasios, et al. "Towards a Digital Twin Enabled Multifidelity Framework for Small Satellites." PHM Society European Conference. Vol. 6. No. 1. 2021.
- Kontaxoglou, Anastasios, et al. "Use of Cokriging for Thermal Analysis in Small Satellites", IEEE Aerospace Conference Proceedings, 2022

List of Tables

Table 2.1	Representative examples of multifidelity methods.	9
Table 2.2	List of Digital Twin review papers per application.	34
Table 3.1	List of Digital Twin review papers by topic.	36
Table 3.2	Model based vs data driven approaches comparison.	40
Table 3.3	Data combinations, advantages and disadvantages.	41
Table 3.4	What LFM and HFM information sources have to offer. . . .	42
Table 4.1	Metrics and their T_1 , T_2 and T_3 transformations.	58
Table 5.1	Generated scenarios for framework training and testing. . . .	66
Table 5.2	ISSL6U-1 data timeseries. Each cell contains the temperature for an ISSL6U-1 component.	66
Table 5.3	Error & time metrics. 6 HF points - noiseless and uniform. .	70
Table 5.4	Error & time metrics. 16 HF points - noiseless and uniform. .	71
Table 5.5	Error & time metrics. 16 HF points - noisy and random. . .	74
Table 5.6	ISSL6U-1 and power data timeseries. Cells contain a temperature value for each component and three power measurements. .	76
Table 5.7	Error & time metrics. 15 HF points - randomly placed for components 1-3.	83
Table 5.8	Error & time metrics. 15 HF points - randomly placed for components 4-6.	85
Table 5.9	Error & time metrics. 15 HF points - randomly placed for components 7-9.	87
Table 5.10	Metrics' averages for ISSL6U-1 data with power. 15 HF points - randomly placed.	87
Table 5.11	Error & time metrics for the overall performance of the sliding window with ISSL6U-1 data with power.	92
Table 5.12	ARIMAX Errors per component, ISSL6U-1 data with power. .	94

Table 5.13 Robustness to HFM point placement, ISSL6U-1 data with power.	96
Table 5.14 Summary of SDS-4 characteristics.	100
Table 5.15 SDS-4 data timeseries. Each temperature T corresponds to a sensor placed on the satellite's body.	100
Table 5.16 SDS-4 Errors per component 1-30. Uniform data.	105
Table 5.17 SDS-4 Errors per component 31-60 and averages. Uniform data.	106
Table 5.18 Error & Time metrics overall average performance SDS-4 with uniform data.	108
Table 5.19 SDS-4 Errors per component, components 1-30.	109
Table 5.20 SDS-4 Errors per component, components 31-60, and averages.	110
Table 5.21 Error & time metrics overall performance for random HF data placement, SDS-4 data.	112
Table 5.22 Error & time metrics overall performance for SDS-4 data with the sliding window.	113
Table 5.23 SDS-4 ARIMAX errors per component, components 1-30.	115
Table 5.24 SDS-4 ARIMAX errors per component, components 31-60.	116
Table 5.25 Robustness to HFM point placement, SDS-4.	118
Table 5.26 SDS-4 using the entire dataset. Errors per component 1-30. Uniform data.	121
Table 5.27 SDS-4 using the entire dataset. Errors per component 31-60 and averages. Uniform data.	122
Table 5.28 Error & time metrics overall performance for SDS-4 data using the entire dataset.	124
Table 5.29 SDS-4 using the entire dataset. Errors per component 1-30. Random data.	125
Table 5.30 SDS-4 using the entire dataset. Errors per component 31-60 and averages. Random data.	126
Table 5.31 Error & time metrics overall performance for SDS-4 data using the entire dataset.	128

List of Figures

Figure 1.1	Small satellite launches, based on numbers found in NSR [4].	3
Figure 2.1	Examples of fidelity reduction methods.	7
Figure 2.2	Gaussian process regression. \bullet : observed data \mathbf{X} , $-$: mean function estimated by the observed data points, \blacksquare : the confidence interval and green points \mathbf{X}_* are new points to be predicted.	11
Figure 2.3	An example of LFM and HFM. $-$: Low fidelity model \mathbf{X} , $-$: High fidelity model.	16
Figure 2.4	Cokriging with constant ρ . $-$: Low fidelity model \mathbf{X} , $-$: High fidelity model, $-$: Multifidelity model.	17
Figure 2.5	Relationship between the LFM and HFM.	17
Figure 2.6	Cokriging with nonlinear ρ . $-$: Low fidelity model \mathbf{X} , $-$: High fidelity model, $-$: Multifidelity model.	17
Figure 2.7	Artificial neuron, where x_i : inputs, w_i : weights, f : activation function and y : output [49].	24
Figure 2.8	An abstract schematic of a recurrent neural network[49].	25
Figure 2.9	A recurrent neural network in chain form [52].	25
Figure 2.10	GRU layout. x_t, h_t are input and output vectors, \hat{h}_t is the candidate activation vector and z_t, r_t are the update and reset gate vectors.	27
Figure 2.11	Stacked CNN and LSTM layers.	27
Figure 2.12	CNN-LSTM vs GRU results, SDS-4 data. Blue lines ($-$) indicate the real satellite values and red lines ($-$) the predicted data points for component 33.	28
Figure 2.13	An abstract schematic of a feedforward neural network.	29
Figure 2.14	Example of FE nodes.	32
Figure 3.1	The Digital Twin on Google trends.	35
Figure 3.2	The Digital Twin's main components.	37
Figure 3.3	The Digital Twin diagram with the connections between the virtual and the physical entities.	37
Figure 3.4	Hierarchy of physics models in simulations.	39
Figure 3.5	An example of the direction of maximum variance. $-$ is the direction of maximum variance.	44
Figure 3.6	An example where the direction of maximum variance is not clear. $-$ is the direction of maximum variance.	44
Figure 4.1	The general approach for a multifidelity framework.	47
Figure 4.2	LFM and HFM and their inputs. QoI is the quantity of interest.	47
Figure 4.3	Schematic of the proposed multifidelity framework.	48

Figure 4.4	Timekeeping in the multifidelity framework. H , L are high and low fidelity data points, T_1 , T_2 are intervals between high fidelity data, T_{s_L} is the sampling interval of the low fidelity model and t_{window} is the pre defined length of the update window.	49
Figure 4.5	How the proposed framework fits the conception of the Digital Twin.	49
Figure 4.6	Interdependencies in the hidden layers.	50
Figure 4.7	The multimodal framework configuration.	51
Figure 4.8	Example and counter example of high fidelity point distributions.	52
Figure 4.9	Process for extracting a robustness value.	54
Figure 4.10	Conditions affecting ranking stability. A high stability in ranking indicates that two metrics will rank the decision alternatives the same, whereas a low stability indicates that two metrics will rank the decision alternatives differently.	55
Figure 5.1	The sliding window. w is the width of the window and T is one period.	62
Figure 5.2	Simplified ISSL6U-1 layout.	62
Figure 5.3	Sources of radiation for a satellite.	63
Figure 5.4	Thermal effects on satellites.	64
Figure 5.5	The beta angle β	65
Figure 5.6	An example orbit with data downlink.	65
Figure 5.7	Training approach summary for ISSL6U-1 data. \mathbf{T} : vector of temperatures, N : number of rows.	67
Figure 5.8	Testing approach summary for ISSL6U-1 data. \mathbf{T} : vector of temperatures, $\hat{\mathbf{T}}_{\text{GRU}}$: vector of GRU estimated temperatures, $\hat{\mathbf{T}}_{\text{ck}}$: vector of cokriging estimated temperatures, N : number of rows.	68
Figure 5.9	Testing approach summary for ISSL6U-1 data for GP and ARIMAX. \mathbf{T} : vector of temperatures, $\hat{\mathbf{T}}_{\text{GRU}}$: vector of GRU estimated temperatures, $\hat{\mathbf{T}}_{\text{ck}}$: vector of cokriging estimated temperatures, N : number of rows.	69
Figure 5.10	GRU results for ISSL6U-1 data. Red lines (-) indicate the predicted temperatures in the y-axis in $^{\circ}\text{C}$ and blue lines (-) indicate the real temperatures for the 6 components. Time indicates percentage of the orbit with 1 being a complete orbit.	69
Figure 5.11	Using 6 noiseless and uniformly placed HFM points for ISSL6U-1 data. Blue lines (-) indicate the LFM, orange lines (-) the HFM, green lines (-) the multifidelity estimation and red dots (•) the HFM data points used and the gray area (■) the confidence interval for the 6 components.	70
Figure 5.12	Using 16 noiseless and uniformly placed HFM points for ISSL6U-1 data. Blue lines (-) indicate the LFM, orange lines (-) the HFM, green lines (-) the multifidelity estimation and red dots (•) the HFM data points used and the gray area (■) the confidence interval for the 6 components.	71
Figure 5.13	Using 16 noisy and randomly placed HFM points for ISSL6U-1 data. Blue lines (-) indicate the LFM, orange lines (-) the HFM, green lines (-) the multifidelity estimation and red dots (•) the HFM data points used and the gray area (■) the confidence interval for components 1-3.	72

Figure 5.14	Using 16 noisy and randomly placed HFM points for ISSL6U-1 data. Blue lines (-) indicate the LFM, orange lines (-) the HFM, green lines (-) the multifidelity estimation and red dots (•) the HFM data points used and the gray area (■) the confidence interval for components 4-6.	73
Figure 5.15	Voltage per cell - temperature for acid-lead batteries [144].	76
Figure 5.16	An example orbit including the battery states.	78
Figure 5.17	Training approach summary for ISSL6U-1 data with power data. \mathbf{T} : vector of temperatures, \mathbf{P} : vector of power measurements, N : number of rows.	79
Figure 5.18	Testing approach summary for ISSL6U-1 data with power data. \mathbf{T} : vector of temperatures, $\hat{\mathbf{T}}_{\text{GRU}}$: vector of GRU estimated temperatures, $\hat{\mathbf{T}}_{\text{ck}}$: vector of cokriging,, $\hat{\mathbf{P}}_{\text{GRU}}$: vector of GRU estimated power measurements, $\hat{\mathbf{P}}_{\text{ck}}$: vector of cokriging estimated power measurements, N : number of rows.	79
Figure 5.19	Multifidelity framework's output format.	80
Figure 5.20	Testing approach summary for ISSL6U-1 data with power for GP and ARIMAX. \mathbf{T} : vector of temperatures, $\hat{\mathbf{T}}_{\text{GRU}}$: vector of GRU estimated temperatures, $\hat{\mathbf{T}}_{\text{ck}}$: vector of cokriging estimated temperatures, N : number of rows.	80
Figure 5.21	GRU results for ISSL6U-1 data with power. Red lines (-) indicate the predicted temperatures and blue lines (-) indicate the real temperatures for the 9 components.	81
Figure 5.22	Multimodal simulation, including power data, with 15 HF points randomly placed. Blue lines (-) indicate the LFM, orange lines (-) the HFM, green lines (-) the multifidelity estimation and red dots (•) the HFM data points used and the gray area (■) the confidence interval for components 1-3.	82
Figure 5.23	Multimodal simulation, including power data, with 15 HF points randomly placed. Blue lines (-) indicate the LFM, orange lines (-) the HFM, green lines (-) the multifidelity estimation and red dots (•) the HFM data points used and the gray area (■) the confidence interval for components 4-6.	84
Figure 5.24	Multimodal simulation, including power data, with 15 HF points randomly placed. Blue lines (-) indicate the LFM, orange lines (-) the HFM, green lines (-) the multifidelity estimation and red dots (•) the HFM data points used and the gray area (■) the confidence interval for components 7-9.	86
Figure 5.25	An example of a window for ISSL6U-1 data with power. The vertical blue lines indicate the point from which future values are calculated. Horizontal blue lines (-) indicate the LFM, orange lines (-) the HFM, green lines (-) the multifidelity estimation and the gray area (■) the confidence interval for components 1-3. . . .	88
Figure 5.26	The overall result after sliding windows for ISSL6U-1 with power data. Blue lines (-) indicate the LFM, orange lines (-) the HFM, green lines (-) the multifidelity estimation and the gray area (■) the confidence interval for components 1-3.	89
Figure 5.27	The overall result after sliding windows for ISSL6U-1 with power data. Blue lines (-) indicate the LFM, orange lines (-) the HFM, green lines (-) the multifidelity estimation and the gray area (■) the confidence interval for components 4-6.	90

Figure 5.28 The overall result after sliding windows for ISSL6U-1 with power data. Blue lines (-) indicate the LFM, orange lines (-) the HFM, green lines (-) the multifidelity estimation and the gray area (■) the confidence interval for components 7-9.	91
Figure 5.29 The overall result for all components after sliding windows for ISSL6U-1 with power data.. Blue lines (-) indicate the LFM, orange lines (-) the HFM, green lines (-) the multifidelity estimation and the gray area (■) the confidence interval for components 7-9.	92
Figure 5.30 ARIMAX for components 1-8, ISSL6U-1 data with power. Blue lines (-) indicate the LFM, orange lines (-) the HFM, green lines (-) the multifidelity estimation for components 1-8. The x-axis represents the percentage of the orbit between 0 – 100%. . . .	93
Figure 5.31 ARIMAX for component 9, ISSL6U-1 data with power. Blue lines (-) indicate the LFM, orange lines (-) the HFM, green lines (-) the multifidelity estimation for components 1-8. The x-axis represents the percentage of the orbit between 0 – 100%. . . .	94
Figure 5.32 Average metric comparison, including ARIMAX. ISSL6U-1 data with power.	95
Figure 5.33 MSE per scenario, ISSL6U-1 data with power.	96
Figure 5.34 MF and GP MSE errors per scenario juxtaposed, ISSL6U-1 data with power.	97
Figure 5.35 Point distribution in failed scenario 62.	97
Figure 5.36 Time allocation for ISSL6U-1 data with power components.	98
Figure 5.37 SDS-4 satellite model [146].	99
Figure 5.38 Training approach summary for SDS-4 data. \mathbf{T} : vector of temperatures, N : number of rows.	102
Figure 5.39 Testing approach summary for SDS-4 data. \mathbf{T} : vector of temperatures, $\hat{\mathbf{T}}_{\text{GRU}}$: vector of GRU estimated temperatures, $\hat{\mathbf{T}}_{\text{ck}}$: vector of cokriging estimated temperatures, N : number of rows.	103
Figure 5.40 Testing approach summary for SDS-4 data for GP and ARIMAX. \mathbf{T} : vector of temperatures, $\hat{\mathbf{T}}_{\text{GRU}}$: vector of GRU estimated temperatures, $\hat{\mathbf{T}}_{\text{ck}}$: vector of cokriging estimated temperatures, N : number of rows.	104
Figure 5.41 Examples of the GRU estimation for SDS-4 data. y-axis is temperature $T(^{\circ}\text{C})$ and x-axis is time $t(s)$	104
Figure 5.42 SDS-4 data, multifidelity vs Gaussian process. Uniform high fidelity data points. Blue lines (-) indicate the LFM, orange lines (-) the HFM, green lines (-) the multifidelity estimation and red dots (●) the HFM data points used and the gray area (■) the confidence interval for components 1 and 42.	107
Figure 5.43 SDS-4 data, multifidelity vs Gaussian process. Randomly placed high fidelity data points. Blue lines (-) indicate the LFM, orange lines (-) the HFM, green lines (-) the multifidelity estimation and red dots (●) the HFM data points used and the gray area (■) the confidence interval for components 1, 37 and 47.	111
Figure 5.44 An example of a window for component 43 of SDS-4. The vertical blue lines indicate the point from which future values are calculated. Horizontal blue lines (-) indicate the LFM, orange lines (-) the HFM, green lines (-) the multifidelity estimation and the gray area (■) the confidence interval.	112

Figure 5.45	An example of a window for component 57 of SDS-4 data. The vertical blue lines indicate the point from which future values are calculated. Horizontal blue lines (-) indicate the LFM, orange lines (-) the HFM, green lines (-) the multifidelity estimation and the gray area (■) the confidence interval.	113
Figure 5.46	ARIMAX SDS-4 data. Blue lines (-) indicate the LFM, orange lines (-) the ARIMAX prediction for components 1-4. . . .	114
Figure 5.47	SDS-4 average metric comparison, including ARIMAX. . .	117
Figure 5.48	MF and ARIMAX time comparison for SDS-4 data. . . .	117
Figure 5.49	MSE per scenario for SDS-4.	118
Figure 5.50	MF and GP MSE errors per scenario juxtaposed for SDS-4. . . .	119
Figure 5.51	Testing approach summary for SDS-4's entire dataset. T : vector of temperatures, $\hat{\mathbf{T}}_{\text{GRU}}$: vector of GRU estimated temperatures, $\hat{\mathbf{T}}_{\text{ck}}$: vector of cokriging estimated temperatures, N : number of rows.	120
Figure 5.52	Examples of the GRU estimation for SDS-4's entire dataset. y-axis is temperature $T(C^\circ)$ and x-axis is time $t(s)$	120
Figure 5.53	SDS-4 data for SDS-4's entire dataset. Multifidelity vs Gaussian process. Uniform high fidelity data points. Blue lines (-) indicate the LFM, orange lines (-) the HFM, green lines (-) the multifidelity estimation and red dots (•) the HFM data points used and the gray area (■) the confidence interval for components 1 and 42.	123
Figure 5.54	SDS-4 data, SDS-4 data for SDS-4's entire dataset. Multifidelity vs Gaussian process. Randomly placed high fidelity data points. Blue lines (-) indicate the LFM, orange lines (-) the HFM, green lines (-) the multifidelity estimation and red dots (•) the HFM data points used and the gray area (■) the confidence interval for components 1, 37 and 47.	127
Figure 6.1	A diagram of the Digital Twin. In red are the components using the multifidelity framework.	130
Figure 6.2	Diagram for live computation.	133
Figure 6.3	GUI with the state estimation per second.	134
Figure 6.4	Parallel processing diagram.	135
Figure A.1	GRU results, SDS-4 data. Blue lines (-) indicate the real satellite values and red lines (-) the predicted data points for components 0-8.	155
Figure A.2	GRU results, SDS-4 data. Blue lines (-) indicate the real satellite values and red lines (-) the predicted data points for components 8-16.	156
Figure A.3	GRU results, SDS-4 data. Blue lines (-) indicate the real satellite values and red lines (-) the predicted data points for components 16-24.	157
Figure A.4	GRU results, SDS-4 data. Blue lines (-) indicate the real satellite values and red lines (-) the predicted data points for components 24-32.	158
Figure A.5	GRU results, SDS-4 data. Blue lines (-) indicate the real satellite values and red lines (-) the predicted data points for components 32-40.	159

Figure A.6 GRU results, SDS-4 data. Blue lines (-) indicate the real satellite values and red lines (-) the predicted data points for components 40-48.	160
Figure A.7 GRU results, SDS-4 data. Blue lines (-) indicate the real satellite values and red lines (-) the predicted data points for components 48-56.	161
Figure A.8 GRU results, SDS-4 data. Blue lines (-) indicate the real satellite values and red lines (-) the predicted data points for components 57-60.	162
Figure A.9 GRU results, SDS-4 data. Blue lines (-) indicate the real satellite values and red lines (-) the predicted data points for all components.	162
Figure A.10 SDS-4 data. Multifidelity vs Gaussian process. Non-random high fidelity data points. Blue lines (-) indicate the LFM, orange lines (-) the HFM, green lines (-) the multifidelity estimation and red dots (•) the HFM data points used and the gray area (■) the confidence interval for components 1-4.	163
Figure A.11 SDS-4 data. Multifidelity vs Gaussian process. Non-random high fidelity data points. Blue lines (-) indicate the LFM, orange lines (-) the HFM, green lines (-) the multifidelity estimation and red dots (•) the HFM data points used and the gray area (■) the confidence interval for components 5-8.	164
Figure A.12 SDS-4 data. Multifidelity vs Gaussian process. Non-random high fidelity data points. Blue lines (-) indicate the LFM, orange lines (-) the HFM, green lines (-) the multifidelity estimation and red dots (•) the HFM data points used and the gray area (■) the confidence interval for components 9-12.	165
Figure A.13 SDS-4 data. Multifidelity vs Gaussian process. Non-random high fidelity data points. Blue lines (-) indicate the LFM, orange lines (-) the HFM, green lines (-) the multifidelity estimation and red dots (•) the HFM data points used and the gray area (■) the confidence interval for components 13-16.	166
Figure A.14 SDS-4 data. Multifidelity vs Gaussian process. Non-random high fidelity data points. Blue lines (-) indicate the LFM, orange lines (-) the HFM, green lines (-) the multifidelity estimation and red dots (•) the HFM data points used and the gray area (■) the confidence interval for components 17-20.	167
Figure A.15 SDS-4 data. Multifidelity vs Gaussian process. Non-random high fidelity data points. Blue lines (-) indicate the LFM, orange lines (-) the HFM, green lines (-) the multifidelity estimation and red dots (•) the HFM data points used and the gray area (■) the confidence interval for components 21-24.	168
Figure A.16 SDS-4 data. Multifidelity vs Gaussian process. Non-random high fidelity data points. Blue lines (-) indicate the LFM, orange lines (-) the HFM, green lines (-) the multifidelity estimation and red dots (•) the HFM data points used and the gray area (■) the confidence interval for components 25-28.	169
Figure A.17 SDS-4 data. Multifidelity vs Gaussian process. Non-random high fidelity data points. Blue lines (-) indicate the LFM, orange lines (-) the HFM, green lines (-) the multifidelity estimation and red dots (•) the HFM data points used and the gray area (■) the confidence interval for components 29-32.	170

Figure A.18 SDS-4 data. Multifidelity vs Gaussian process. Non-random high fidelity data points. Blue lines (-) indicate the LFM, orange lines (-) the HFM, green lines (-) the multifidelity estimation and red dots (•) the HFM data points used and the gray area (■) the confidence interval for components 33-36.	171
Figure A.19 SDS-4 data. Multifidelity vs Gaussian process. Non-random high fidelity data points. Blue lines (-) indicate the LFM, orange lines (-) the HFM, green lines (-) the multifidelity estimation and red dots (•) the HFM data points used and the gray area (■) the confidence interval for components 37-40.	172
Figure A.20 SDS-4 data. Multifidelity vs Gaussian process. Non-random high fidelity data points. Blue lines (-) indicate the LFM, orange lines (-) the HFM, green lines (-) the multifidelity estimation and red dots (•) the HFM data points used and the gray area (■) the confidence interval for components 41-44.	173
Figure A.21 SDS-4 data. Multifidelity vs Gaussian process. Non-random high fidelity data points. Blue lines (-) indicate the LFM, orange lines (-) the HFM, green lines (-) the multifidelity estimation and red dots (•) the HFM data points used and the gray area (■) the confidence interval for components 45-48.	174
Figure A.22 SDS-4 data. Multifidelity vs Gaussian process. Non-random high fidelity data points. Blue lines (-) indicate the LFM, orange lines (-) the HFM, green lines (-) the multifidelity estimation and red dots (•) the HFM data points used and the gray area (■) the confidence interval for components 49-52.	175
Figure A.23 SDS-4 data. Multifidelity vs Gaussian process. Non-random high fidelity data points. Blue lines (-) indicate the LFM, orange lines (-) the HFM, green lines (-) the multifidelity estimation and red dots (•) the HFM data points used and the gray area (■) the confidence interval for components 53-56.	176
Figure A.24 SDS-4 data. Multifidelity vs Gaussian process. Non-random high fidelity data points. Blue lines (-) indicate the LFM, orange lines (-) the HFM, green lines (-) the multifidelity estimation and red dots (•) the HFM data points used and the gray area (■) the confidence interval for components 57-60.	177
Figure A.25 SDS-4 data. Multifidelity vs Gaussian process. Random high fidelity data points. Blue lines (-) indicate the LFM, orange lines (-) the HFM, green lines (-) the multifidelity estimation and red dots (•) the HFM data points used and the gray area (■) the confidence interval for components 1-4.	178
Figure A.26 SDS-4 data. Multifidelity vs Gaussian process. Random high fidelity data points. Blue lines (-) indicate the LFM, orange lines (-) the HFM, green lines (-) the multifidelity estimation and red dots (•) the HFM data points used and the gray area (■) the confidence interval for components 5-8.	179
Figure A.27 SDS-4 data. Multifidelity vs Gaussian process. Random high fidelity data points. Blue lines (-) indicate the LFM, orange lines (-) the HFM, green lines (-) the multifidelity estimation and red dots (•) the HFM data points used and the gray area (■) the confidence interval for components 9-12.	180

Figure A.28 SDS-4 data. Multifidelity vs Gaussian process. Random high fidelity data points. Blue lines (-) indicate the LFM, orange lines (-) the HFM, green lines (-) the multifidelity estimation and red dots (•) the HFM data points used and the gray area (■) the confidence interval for components 13-16.	181
Figure A.29 SDS-4 data. Multifidelity vs Gaussian process. Random high fidelity data points. Blue lines (-) indicate the LFM, orange lines (-) the HFM, green lines (-) the multifidelity estimation and red dots (•) the HFM data points used and the gray area (■) the confidence interval for components 17-20.	182
Figure A.30 SDS-4 data. Multifidelity vs Gaussian process. Random high fidelity data points. Blue lines (-) indicate the LFM, orange lines (-) the HFM, green lines (-) the multifidelity estimation and red dots (•) the HFM data points used and the gray area (■) the confidence interval for components 21-24.	183
Figure A.31 SDS-4 data. Multifidelity vs Gaussian process. Random high fidelity data points. Blue lines (-) indicate the LFM, orange lines (-) the HFM, green lines (-) the multifidelity estimation and red dots (•) the HFM data points used and the gray area (■) the confidence interval for components 25-28.	184
Figure A.32 SDS-4 data. Multifidelity vs Gaussian process. Random high fidelity data points. Blue lines (-) indicate the LFM, orange lines (-) the HFM, green lines (-) the multifidelity estimation and red dots (•) the HFM data points used and the gray area (■) the confidence interval for components 29-32.	185
Figure A.33 SDS-4 data. Multifidelity vs Gaussian process. Random high fidelity data points. Blue lines (-) indicate the LFM, orange lines (-) the HFM, green lines (-) the multifidelity estimation and red dots (•) the HFM data points used and the gray area (■) the confidence interval for components 33-36.	186
Figure A.34 SDS-4 data. Multifidelity vs Gaussian process. Random high fidelity data points. Blue lines (-) indicate the LFM, orange lines (-) the HFM, green lines (-) the multifidelity estimation and red dots (•) the HFM data points used and the gray area (■) the confidence interval for components 37-40.	187
Figure A.35 SDS-4 data. Multifidelity vs Gaussian process. Random high fidelity data points. Blue lines (-) indicate the LFM, orange lines (-) the HFM, green lines (-) the multifidelity estimation and red dots (•) the HFM data points used and the gray area (■) the confidence interval for components 41-44.	188
Figure A.36 SDS-4 data. Multifidelity vs Gaussian process. Random high fidelity data points. Blue lines (-) indicate the LFM, orange lines (-) the HFM, green lines (-) the multifidelity estimation and red dots (•) the HFM data points used and the gray area (■) the confidence interval for components 45-48.	189
Figure A.37 SDS-4 data. Multifidelity vs Gaussian process. Random high fidelity data points. Blue lines (-) indicate the LFM, orange lines (-) the HFM, green lines (-) the multifidelity estimation and red dots (•) the HFM data points used and the gray area (■) the confidence interval for components 49-52.	190

Figure A.38 SDS-4 data. Multifidelity vs Gaussian process. Random high fidelity data points. Blue lines (-) indicate the LFM, orange lines (-) the HFM, green lines (-) the multifidelity estimation and red dots (•) the HFM data points used and the gray area (■) the confidence interval for components 53-56.	191
Figure A.39 SDS-4 data. Multifidelity vs Gaussian process. Random high fidelity data points. Blue lines (-) indicate the LFM, orange lines (-) the HFM, green lines (-) the multifidelity estimation and red dots (•) the HFM data points used and the gray area (■) the confidence interval for components 57-60.	192
Figure A.40 SDS-4 data. Multifidelity vs Gaussian process. Random high fidelity data points using the sliding window. Blue lines (-) indicate the LFM, orange lines (-) the HFM, green lines (-) the multifidelity estimation and red dots (•) the HFM data points used and the gray area (■) the confidence interval for components 1-4. .	193
Figure A.41 SDS-4 data. Multifidelity vs Gaussian process. Random high fidelity data points using the sliding window. Blue lines (-) indicate the LFM, orange lines (-) the HFM, green lines (-) the multifidelity estimation and red dots (•) the HFM data points used and the gray area (■) the confidence interval for components 5-8. .	194
Figure A.42 SDS-4 data. Multifidelity vs Gaussian process. Random high fidelity data points using the sliding window. Blue lines (-) indicate the LFM, orange lines (-) the HFM, green lines (-) the multifidelity estimation and red dots (•) the HFM data points used and the gray area (■) the confidence interval for components 9-12.	195
Figure A.43 SDS-4 data. Multifidelity vs Gaussian process. Random high fidelity data points using the sliding window. Blue lines (-) indicate the LFM, orange lines (-) the HFM, green lines (-) the multifidelity estimation and red dots (•) the HFM data points used and the gray area (■) the confidence interval for components 13-16.	196
Figure A.44 SDS-4 data. Multifidelity vs Gaussian process. Random high fidelity data points using the sliding window. Blue lines (-) indicate the LFM, orange lines (-) the HFM, green lines (-) the multifidelity estimation and red dots (•) the HFM data points used and the gray area (■) the confidence interval for components 17-20.	197
Figure A.45 SDS-4 data. Multifidelity vs Gaussian process. Random high fidelity data points using the sliding window. Blue lines (-) indicate the LFM, orange lines (-) the HFM, green lines (-) the multifidelity estimation and red dots (•) the HFM data points used and the gray area (■) the confidence interval for components 21-24.	198
Figure A.46 SDS-4 data. Multifidelity vs Gaussian process. Random high fidelity data points using the sliding window. Blue lines (-) indicate the LFM, orange lines (-) the HFM, green lines (-) the multifidelity estimation and red dots (•) the HFM data points used and the gray area (■) the confidence interval for components 25-28.	199
Figure A.47 SDS-4 data. Multifidelity vs Gaussian process. Random high fidelity data points using the sliding window. Blue lines (-) indicate the LFM, orange lines (-) the HFM, green lines (-) the multifidelity estimation and red dots (•) the HFM data points used and the gray area (■) the confidence interval for components 29-32.	200

Figure A.48 SDS-4 data. Multifidelity vs Gaussian process. Random high fidelity data points using the sliding window. Blue lines (-) indicate the LFM, orange lines (-) the HFM, green lines (-) the multifidelity estimation and red dots (•) the HFM data points used and the gray area (■) the confidence interval for components 33-36.201

Figure A.49 SDS-4 data. Multifidelity vs Gaussian process. Random high fidelity data points using the sliding window. Blue lines (-) indicate the LFM, orange lines (-) the HFM, green lines (-) the multifidelity estimation and red dots (•) the HFM data points used and the gray area (■) the confidence interval for components 37-40.202

Figure A.50 SDS-4 data. Multifidelity vs Gaussian process. Random high fidelity data points using the sliding window. Blue lines (-) indicate the LFM, orange lines (-) the HFM, green lines (-) the multifidelity estimation and red dots (•) the HFM data points used and the gray area (■) the confidence interval for components 41-44.203

Figure A.51 SDS-4 data. Multifidelity vs Gaussian process. Random high fidelity data points using the sliding window. Blue lines (-) indicate the LFM, orange lines (-) the HFM, green lines (-) the multifidelity estimation and red dots (•) the HFM data points used and the gray area (■) the confidence interval for components 45-48.204

Figure A.52 SDS-4 data. Multifidelity vs Gaussian process. Random high fidelity data points using the sliding window. Blue lines (-) indicate the LFM, orange lines (-) the HFM, green lines (-) the multifidelity estimation and red dots (•) the HFM data points used and the gray area (■) the confidence interval for components 49-52.205

Figure A.53 SDS-4 data. Multifidelity vs Gaussian process. Random high fidelity data points using the sliding window. Blue lines (-) indicate the LFM, orange lines (-) the HFM, green lines (-) the multifidelity estimation and red dots (•) the HFM data points used and the gray area (■) the confidence interval for components 53-56.206

Figure A.54 SDS-4 data. Multifidelity vs Gaussian process. Random high fidelity data points using the sliding window. Blue lines (-) indicate the LFM, orange lines (-) the HFM, green lines (-) the multifidelity estimation and red dots (•) the HFM data points used and the gray area (■) the confidence interval for components 57-60.207

Figure A.55 ARIMAX SDS-4data. Blue lines (-) indicate the LFM, orange lines (-) the ARIMAX prediction for components 1-8. . . . 208

Figure A.56 ARIMAX SDS-4data. Blue lines (-) indicate the LFM, orange lines (-) the ARIMAX prediction for components 9-16. . . . 209

Figure A.57 ARIMAX SDS-4data. Blue lines (-) indicate the LFM, orange lines (-) the ARIMAX prediction for components 17-24. . . . 210

Figure A.58 ARIMAX SDS-4data. Blue lines (-) indicate the LFM, orange lines (-) the ARIMAX prediction for components 25-32. . . . 211

Figure A.59 ARIMAX SDS-4data. Blue lines (-) indicate the LFM, orange lines (-) the ARIMAX prediction for components 33-40. . . . 212

Figure A.60 ARIMAX SDS-4data. Blue lines (-) indicate the LFM, orange lines (-) the ARIMAX prediction for components 41-48. . . . 213

Figure A.61 ARIMAX SDS-4data. Blue lines (-) indicate the LFM, orange lines (-) the ARIMAX prediction for components 49-56. . . . 214

Figure A.62 ARIMAX SDS-4 data. Blue lines (-) indicate the LFM, orange lines (-) the ARIMAX prediction for components 57-60. . . . 215

Figure A.63 GRU results using the entire SDS-4 dataset. Blue lines (-) indicate the real satellite values and red lines (-) the predicted data points for components 0-8.	216
Figure A.64 GRU results using the entire SDS-4 dataset. Blue lines (-) indicate the real satellite values and red lines (-) the predicted data points for components 8-16.	217
Figure A.65 GRU results using the entire SDS-4 dataset. Blue lines (-) indicate the real satellite values and red lines (-) the predicted data points for components 16-24.	218
Figure A.66 GRU results using the entire SDS-4 dataset. Blue lines (-) indicate the real satellite values and red lines (-) the predicted data points for components 24-32.	219
Figure A.67 GRU results using the entire SDS-4 dataset. Blue lines (-) indicate the real satellite values and red lines (-) the predicted data points for components 32-40.	220
Figure A.68 GRU results using the entire SDS-4 dataset. Blue lines (-) indicate the real satellite values and red lines (-) the predicted data points for components 40-48.	221
Figure A.69 GRU results using the entire SDS-4 dataset. Blue lines (-) indicate the real satellite values and red lines (-) the predicted data points for components 48-56.	222
Figure A.70 GRU results using the entire SDS-4 dataset. Blue lines (-) indicate the real satellite values and red lines (-) the predicted data points for components 57-60.	223
Figure A.71 GRU results using the entire SDS-4 dataset. Blue lines (-) indicate the real satellite values and red lines (-) the predicted data points for all components.	223
Figure A.72 SDS-4 data using the entire dataset. Multifidelity vs Gaussian process. Uniform high fidelity data points. Blue lines (-) indicate the LFM, orange lines (-) the HFM, green lines (-) the multifidelity estimation and red dots (•) the HFM data points used and the gray area (■) the confidence interval for components 1-4.	224
Figure A.73 SDS-4 data using the entire dataset. Multifidelity vs Gaussian process. Uniform high fidelity data points. Blue lines (-) indicate the LFM, orange lines (-) the HFM, green lines (-) the multifidelity estimation and red dots (•) the HFM data points used and the gray area (■) the confidence interval for components 5-8.	225
Figure A.74 SDS-4 data using the entire dataset. Multifidelity vs Gaussian process. Uniform high fidelity data points. Blue lines (-) indicate the LFM, orange lines (-) the HFM, green lines (-) the multifidelity estimation and red dots (•) the HFM data points used and the gray area (■) the confidence interval for components 9-12.	226
Figure A.75 SDS-4 data using the entire dataset. Multifidelity vs Gaussian process. Uniform high fidelity data points. Blue lines (-) indicate the LFM, orange lines (-) the HFM, green lines (-) the multifidelity estimation and red dots (•) the HFM data points used and the gray area (■) the confidence interval for components 13-16.	227

Figure A.76 SDS-4 data using the entire dataset. Multifidelity vs Gaussian process. Uniform high fidelity data points. Blue lines (-) indicate the LFM, orange lines (-) the HFM, green lines (-) the multifidelity estimation and red dots (•) the HFM data points used and the gray area (■) the confidence interval for components 17-20.	228
Figure A.77 SDS-4 data using the entire dataset. Multifidelity vs Gaussian process. Uniform high fidelity data points. Blue lines (-) indicate the LFM, orange lines (-) the HFM, green lines (-) the multifidelity estimation and red dots (•) the HFM data points used and the gray area (■) the confidence interval for components 21-24.	229
Figure A.78 SDS-4 data using the entire dataset. Multifidelity vs Gaussian process. Uniform high fidelity data points. Blue lines (-) indicate the LFM, orange lines (-) the HFM, green lines (-) the multifidelity estimation and red dots (•) the HFM data points used and the gray area (■) the confidence interval for components 25-28.	230
Figure A.79 SDS-4 data using the entire dataset. Multifidelity vs Gaussian process. Uniform high fidelity data points. Blue lines (-) indicate the LFM, orange lines (-) the HFM, green lines (-) the multifidelity estimation and red dots (•) the HFM data points used and the gray area (■) the confidence interval for components 29-32.	231
Figure A.80 SDS-4 data using the entire dataset. Multifidelity vs Gaussian process. Uniform high fidelity data points. Blue lines (-) indicate the LFM, orange lines (-) the HFM, green lines (-) the multifidelity estimation and red dots (•) the HFM data points used and the gray area (■) the confidence interval for components 33-36.	232
Figure A.81 SDS-4 data using the entire dataset. Multifidelity vs Gaussian process. Uniform high fidelity data points. Blue lines (-) indicate the LFM, orange lines (-) the HFM, green lines (-) the multifidelity estimation and red dots (•) the HFM data points used and the gray area (■) the confidence interval for components 37-40.	233
Figure A.82 SDS-4 data using the entire dataset. Multifidelity vs Gaussian process. Uniform high fidelity data points. Blue lines (-) indicate the LFM, orange lines (-) the HFM, green lines (-) the multifidelity estimation and red dots (•) the HFM data points used and the gray area (■) the confidence interval for components 41-44.	234
Figure A.83 SDS-4 data using the entire dataset. Multifidelity vs Gaussian process. Uniform high fidelity data points. Blue lines (-) indicate the LFM, orange lines (-) the HFM, green lines (-) the multifidelity estimation and red dots (•) the HFM data points used and the gray area (■) the confidence interval for components 45-48.	235

Figure A.84 SDS-4 data using the entire dataset. Multifidelity vs Gaussian process. Uniform high fidelity data points. Blue lines (-) indicate the LFM, orange lines (-) the HFM, green lines (-) the multifidelity estimation and red dots (•) the HFM data points used and the gray area (■) the confidence interval for components 49-52.	236
Figure A.85 SDS-4 data using the entire dataset. Multifidelity vs Gaussian process. Uniform high fidelity data points. Blue lines (-) indicate the LFM, orange lines (-) the HFM, green lines (-) the multifidelity estimation and red dots (•) the HFM data points used and the gray area (■) the confidence interval for components 53-56.	237
Figure A.86 SDS-4 data using the entire dataset. Multifidelity vs Gaussian process. Uniform high fidelity data points. Blue lines (-) indicate the LFM, orange lines (-) the HFM, green lines (-) the multifidelity estimation and red dots (•) the HFM data points used and the gray area (■) the confidence interval for components 57-60.	238
Figure A.87 SDS-4 data using the entire dataset. Multifidelity vs Gaussian process. Random high fidelity data points. Blue lines (-) indicate the LFM, orange lines (-) the HFM, green lines (-) the multifidelity estimation and red dots (•) the HFM data points used and the gray area (■) the confidence interval for components 1-4.	239
Figure A.88 SDS-4 data using the entire dataset. Multifidelity vs Gaussian process. Random high fidelity data points. Blue lines (-) indicate the LFM, orange lines (-) the HFM, green lines (-) the multifidelity estimation and red dots (•) the HFM data points used and the gray area (■) the confidence interval for components 5-8.	240
Figure A.89 SDS-4 data using the entire dataset. Multifidelity vs Gaussian process. Random high fidelity data points. Blue lines (-) indicate the LFM, orange lines (-) the HFM, green lines (-) the multifidelity estimation and red dots (•) the HFM data points used and the gray area (■) the confidence interval for components 9-12.	241
Figure A.90 SDS-4 data using the entire dataset. Multifidelity vs Gaussian process. Random high fidelity data points. Blue lines (-) indicate the LFM, orange lines (-) the HFM, green lines (-) the multifidelity estimation and red dots (•) the HFM data points used and the gray area (■) the confidence interval for components 13-16.	242
Figure A.91 SDS-4 data using the entire dataset. Multifidelity vs Gaussian process. Random high fidelity data points. Blue lines (-) indicate the LFM, orange lines (-) the HFM, green lines (-) the multifidelity estimation and red dots (•) the HFM data points used and the gray area (■) the confidence interval for components 17-20.	243

Figure A.92 SDS-4 data using the entire dataset. Multifidelity vs Gaussian process. Random high fidelity data points. Blue lines (-) indicate the LFM, orange lines (-) the HFM, green lines (-) the multifidelity estimation and red dots (•) the HFM data points used and the gray area (■) the confidence interval for components 21-24.	244
Figure A.93 SDS-4 data using the entire dataset. Multifidelity vs Gaussian process. Random high fidelity data points. Blue lines (-) indicate the LFM, orange lines (-) the HFM, green lines (-) the multifidelity estimation and red dots (•) the HFM data points used and the gray area (■) the confidence interval for components 25-28.	245
Figure A.94 SDS-4 data using the entire dataset. Multifidelity vs Gaussian process. Random high fidelity data points. Blue lines (-) indicate the LFM, orange lines (-) the HFM, green lines (-) the multifidelity estimation and red dots (•) the HFM data points used and the gray area (■) the confidence interval for components 29-32.	246
Figure A.95 SDS-4 data using the entire dataset. Multifidelity vs Gaussian process. Random high fidelity data points. Blue lines (-) indicate the LFM, orange lines (-) the HFM, green lines (-) the multifidelity estimation and red dots (•) the HFM data points used and the gray area (■) the confidence interval for components 33-36.	247
Figure A.96 SDS-4 data using the entire dataset. Multifidelity vs Gaussian process. Random high fidelity data points. Blue lines (-) indicate the LFM, orange lines (-) the HFM, green lines (-) the multifidelity estimation and red dots (•) the HFM data points used and the gray area (■) the confidence interval for components 37-40.	248
Figure A.97 SDS-4 data using the entire dataset. Multifidelity vs Gaussian process. Random high fidelity data points. Blue lines (-) indicate the LFM, orange lines (-) the HFM, green lines (-) the multifidelity estimation and red dots (•) the HFM data points used and the gray area (■) the confidence interval for components 41-44.	249
Figure A.98 SDS-4 data using the entire dataset. Multifidelity vs Gaussian process. Random high fidelity data points. Blue lines (-) indicate the LFM, orange lines (-) the HFM, green lines (-) the multifidelity estimation and red dots (•) the HFM data points used and the gray area (■) the confidence interval for components 45-48.	250
Figure A.99 SDS-4 data using the entire dataset. Multifidelity vs Gaussian process. Random high fidelity data points. Blue lines (-) indicate the LFM, orange lines (-) the HFM, green lines (-) the multifidelity estimation and red dots (•) the HFM data points used and the gray area (■) the confidence interval for components 49-52.	251

Figure A.100	SDS-4 data using the entire dataset. Multifidelity vs Gaussian process. Random high fidelity data points. Blue lines (-) indicate the LFM, orange lines (-) the HFM, green lines (-) the multifidelity estimation and red dots (•) the HFM data points used and the gray area (■) the confidence interval for components 53-56.	252
Figure A.101	SDS-4 data using the entire dataset. Multifidelity vs Gaussian process. Random high fidelity data points. Blue lines (-) indicate the LFM, orange lines (-) the HFM, green lines (-) the multifidelity estimation and red dots (•) the HFM data points used and the gray area (■) the confidence interval for components 57-60.	253

Appendix A

A.1 Multivariate Normal Distribution Theorem

The Marginal and Conditional Distributions of MVN theorem: suppose $X = (x_1, x_2)$ is a joint Gaussian with parameters

$$\mu = \begin{bmatrix} \mu_1 \\ \mu_2 \end{bmatrix}, \Sigma = \begin{bmatrix} \Sigma_{11} & \Sigma_{12} \\ \Sigma_{21} & \Sigma_{22} \end{bmatrix}, \Lambda = \Sigma^{-1} = \begin{bmatrix} \Lambda_{11} & \Lambda_{12} \\ \Lambda_{21} & \Lambda_{22} \end{bmatrix} \quad (\text{A.1})$$

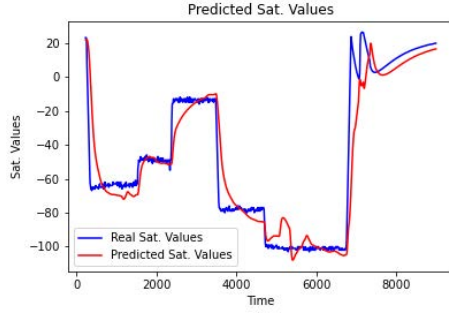
then the marginals are given by

$$\begin{aligned} p(x_1) &= \mathcal{N}(x_1 \mid \mu_1, \Sigma_{11}) \\ p(x_2) &= \mathcal{N}(x_2 \mid \mu_2, \Sigma_{22}) \end{aligned} \quad (\text{A.2})$$

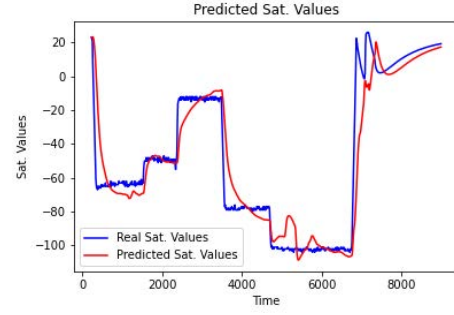
and the posterior conditional is given by

$$\begin{aligned} p(x_1 \mid x_2) &= \mathcal{N}(x_1 \mid \mu_{1|2}, \Sigma_{1|2}) \\ \mu_{1|2} &= \mu_1 + \Sigma_{12}\Sigma_{22}^{-1}(x_2 - \mu_2) \\ &= \mu_1 - \Lambda_{11}^{-1}\Lambda_{12}(x_2 - \mu_2) \\ &= \Sigma_{1|2}(\Lambda_{11}\mu_1 - \Lambda_{12}(x_2 - \mu_2)) \\ \Sigma_{1|2} &= \Sigma_{11} - \Sigma_{12}\Sigma_{22}^{-1}\Sigma_{21} = \Lambda_{11}^{-1} \end{aligned}$$

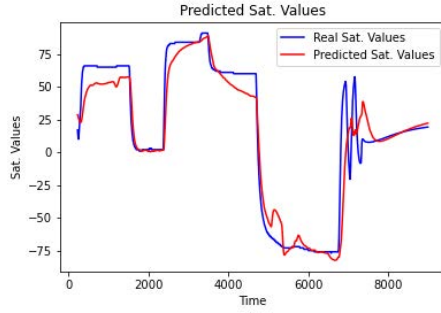
A.2 GRU SDS-4



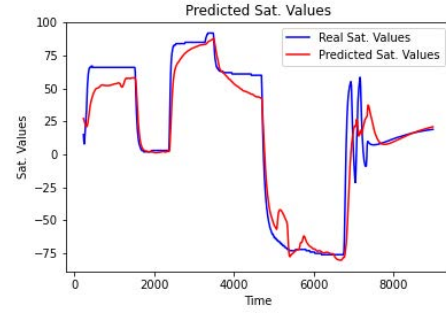
MF - Component 1



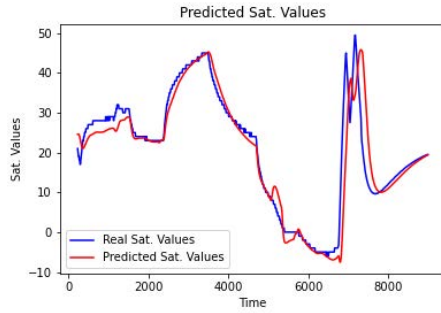
GP - Component 2



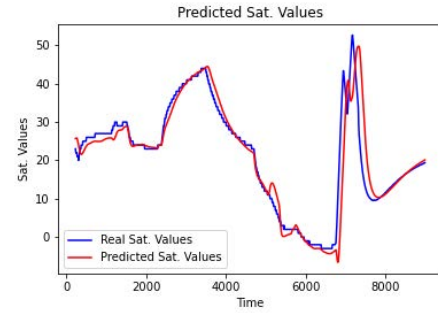
MF - Component 3



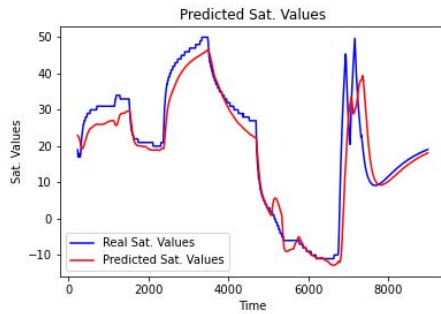
GP - Component 4



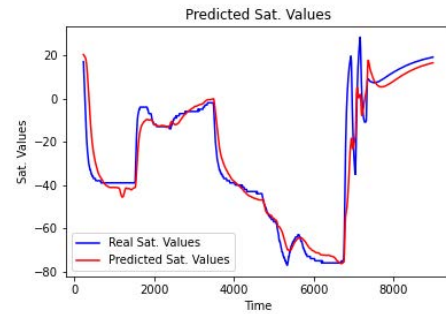
MF - Component 5



GP - Component 6

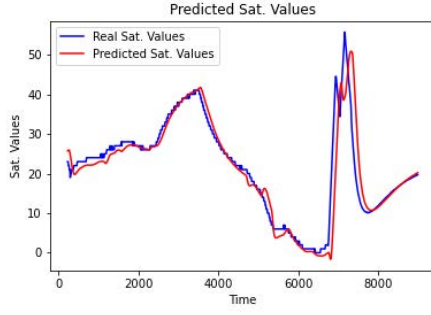


MF - Component 7

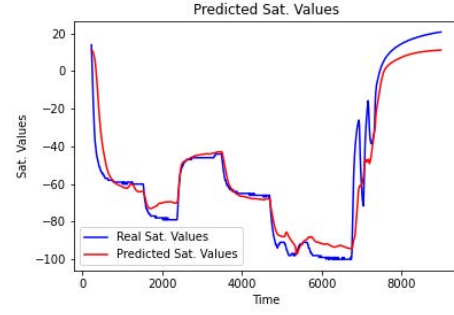


GP - Component 8

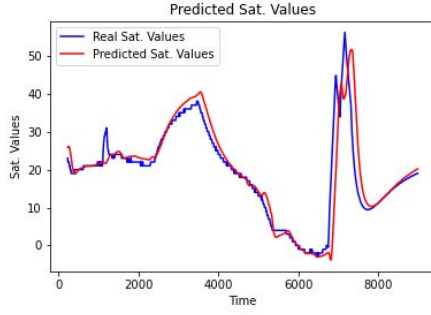
Figure A.1: GRU results, SDS-4 data. Blue lines (—) indicate the real satellite values and red lines (—) the predicted data points for components 0-8.



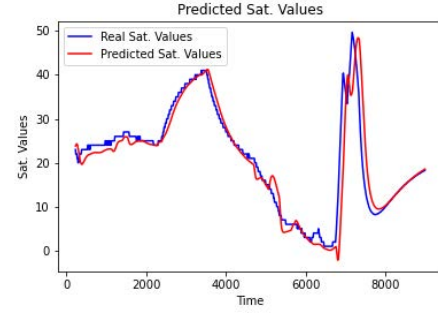
MF - Component 9



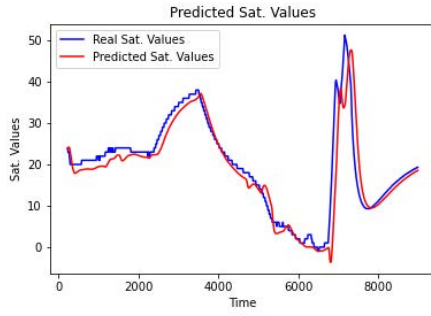
GP - Component 10



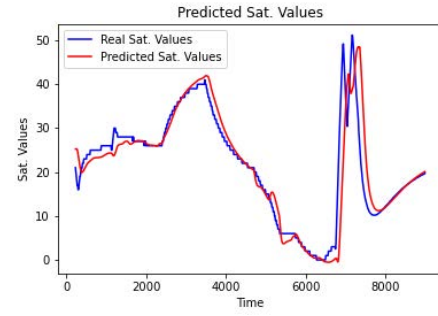
MF - Component 11



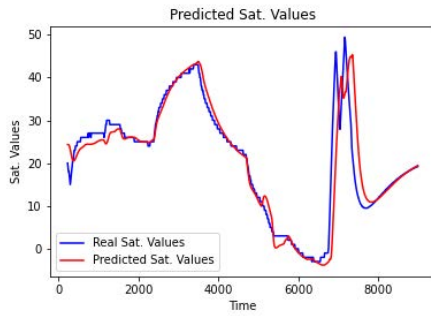
GP - Component 12



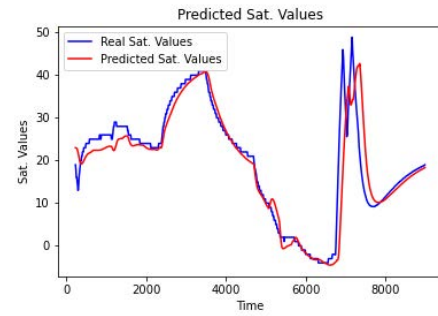
MF - Component 13



GP - Component 14

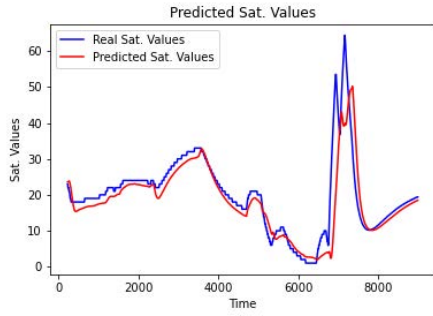


MF - Component 15

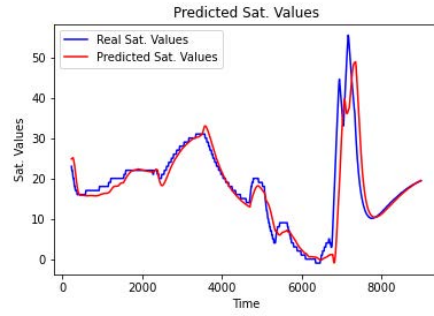


GP - Component 16

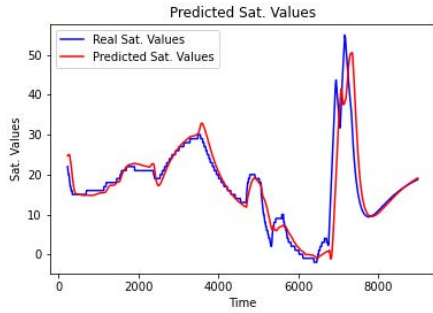
Figure A.2: GRU results, SDS-4 data. Blue lines (—) indicate the real satellite values and red lines (—) the predicted data points for components 8-16.



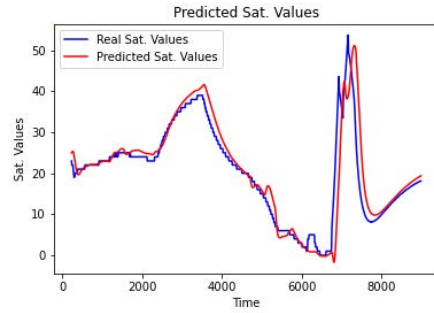
MF - Component 17



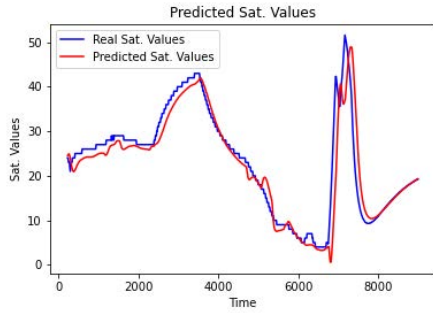
GP - Component 18



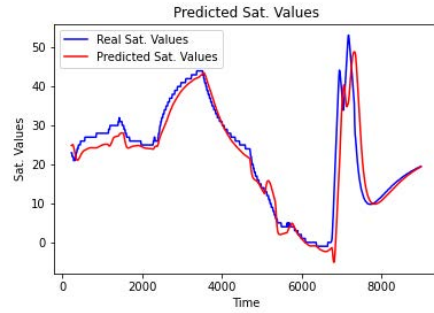
MF - Component 19



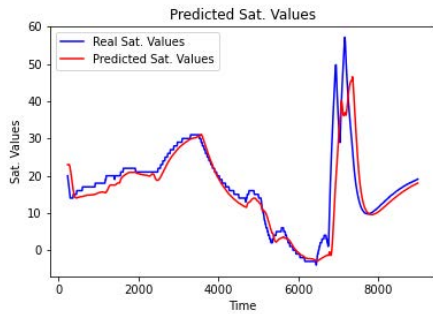
GP - Component 20



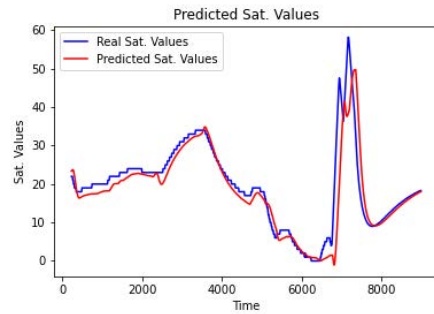
MF - Component 21



GP - Component 22

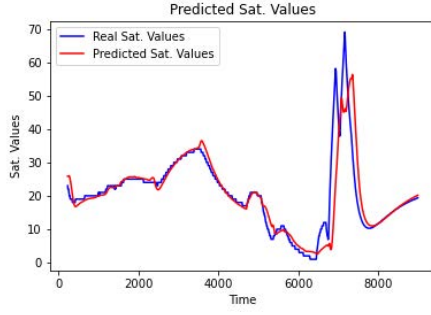


MF - Component 23

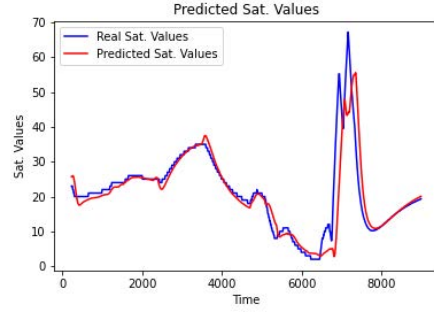


GP - Component 24

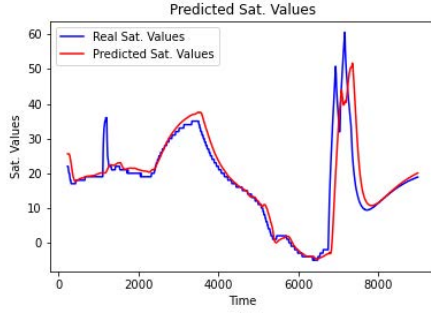
Figure A.3: GRU results, SDS-4 data. Blue lines (—) indicate the real satellite values and red lines (—) the predicted data points for components 16-24.



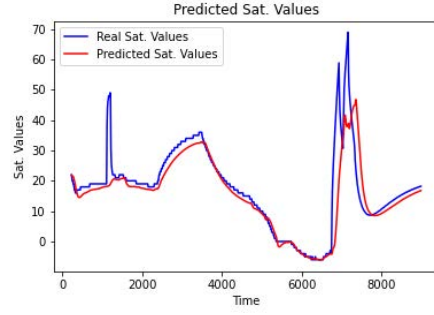
MF - Component 25



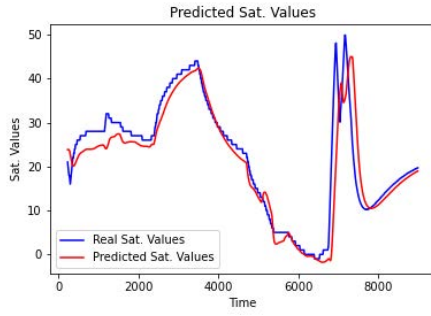
GP - Component 26



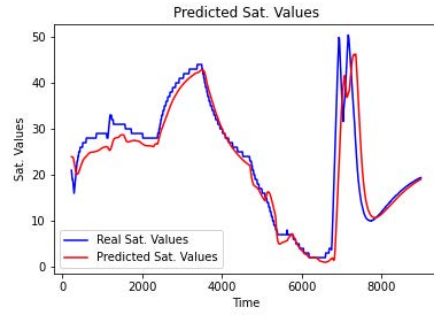
MF - Component 27



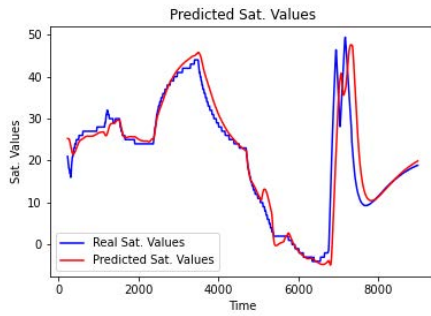
GP - Component 28



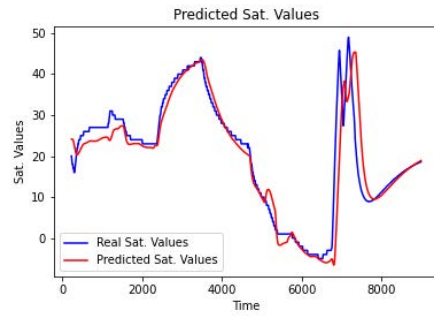
MF - Component 29



GP - Component 30

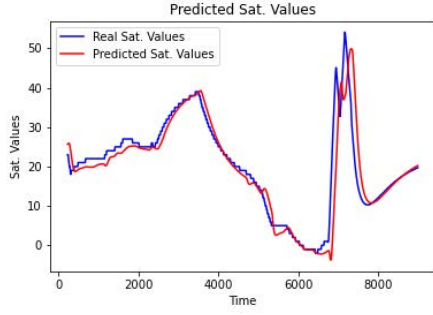


MF - Component 31

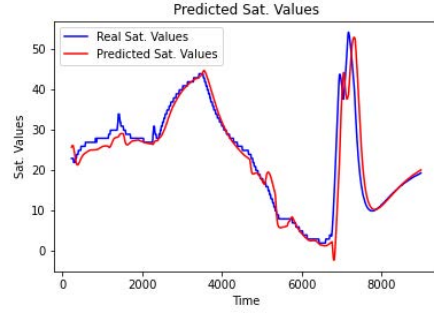


GP - Component 32

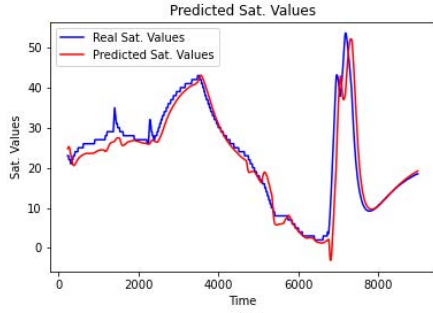
Figure A.4: GRU results, SDS-4 data. Blue lines (—) indicate the real satellite values and red lines (—) the predicted data points for components 24-32.



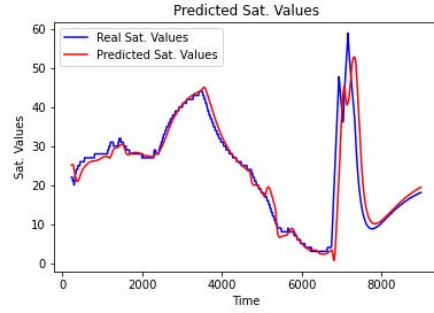
MF - Component 33



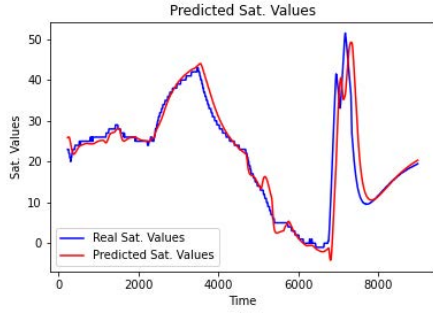
GP - Component 34



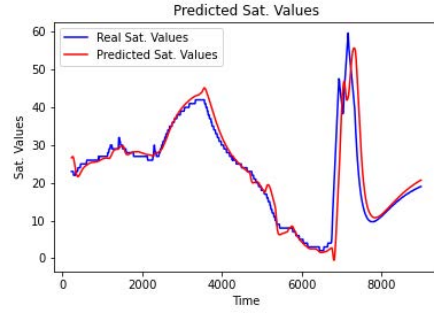
MF - Component 35



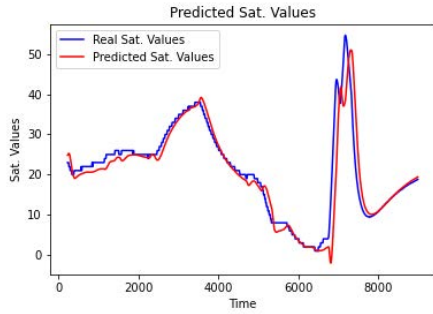
GP - Component 36



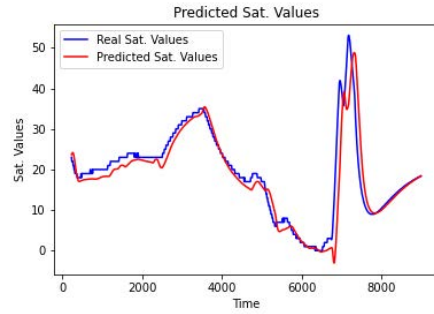
MF - Component 37



GP - Component 38

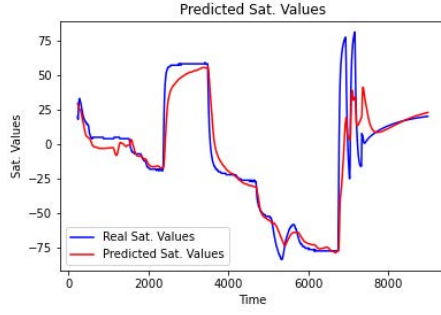


MF - Component 39

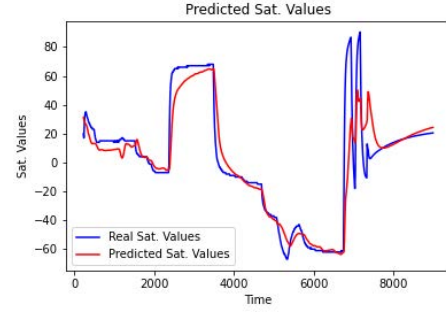


GP - Component 40

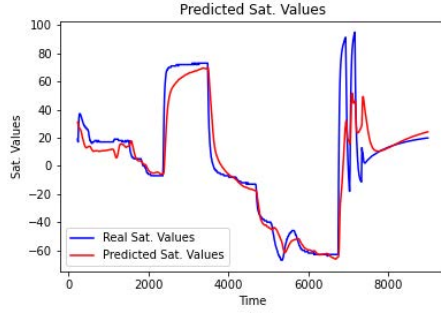
Figure A.5: GRU results, SDS-4 data. Blue lines (—) indicate the real satellite values and red lines (—) the predicted data points for components 32-40.



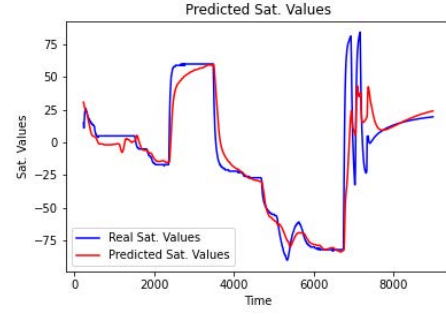
MF - Component 41



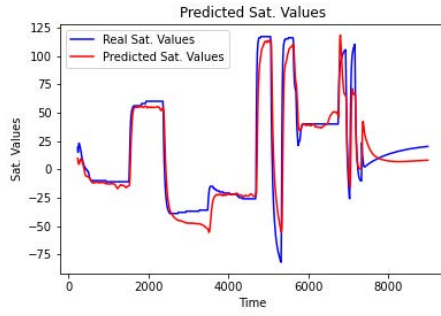
GP - Component 42



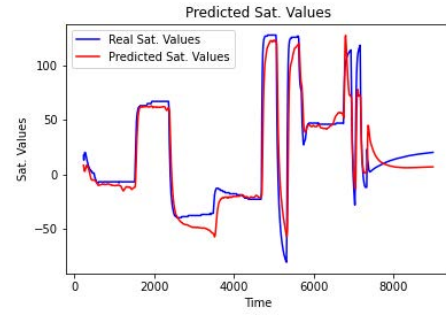
MF - Component 43



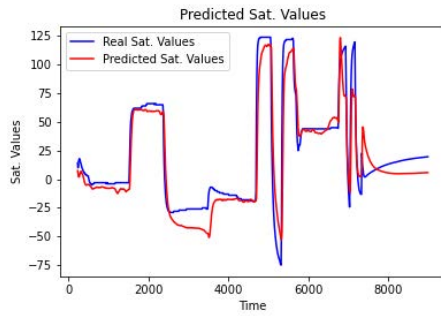
GP - Component 44



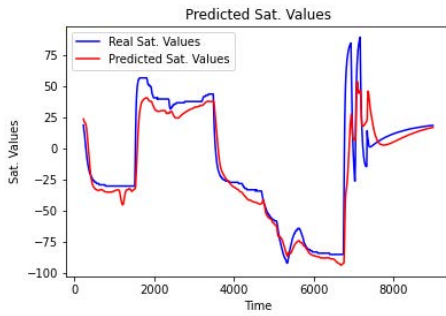
MF - Component 45



GP - Component 46

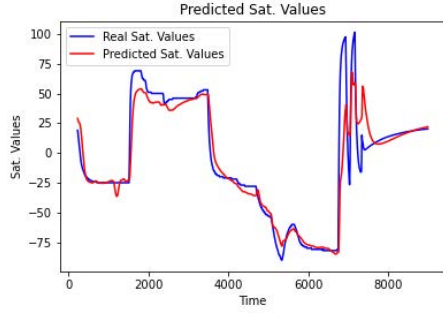


MF - Component 47

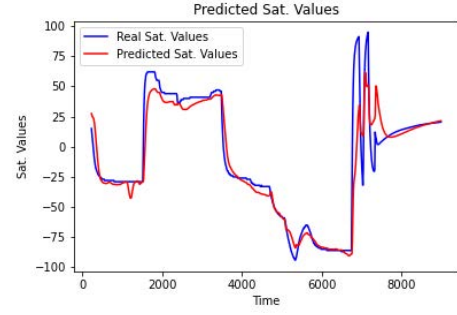


GP - Component 48

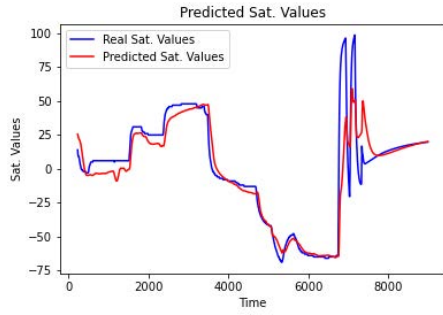
Figure A.6: GRU results, SDS-4 data. Blue lines (—) indicate the real satellite values and red lines (—) the predicted data points for components 40-48.



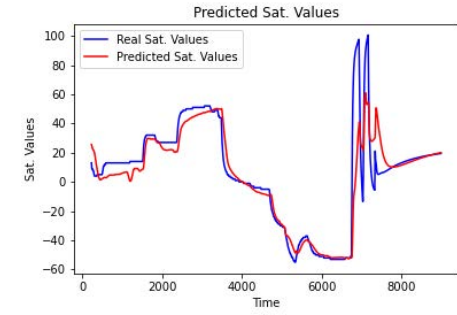
MF - Component 49



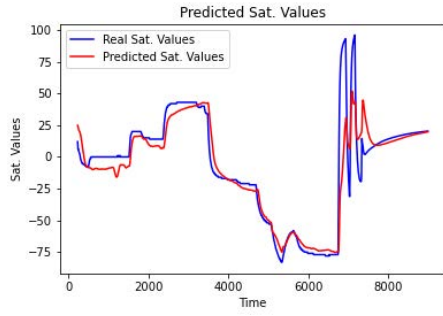
GP - Component 50



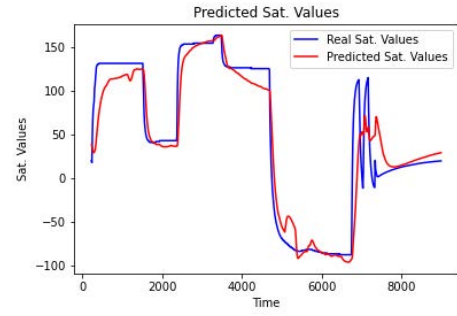
MF - Component 51



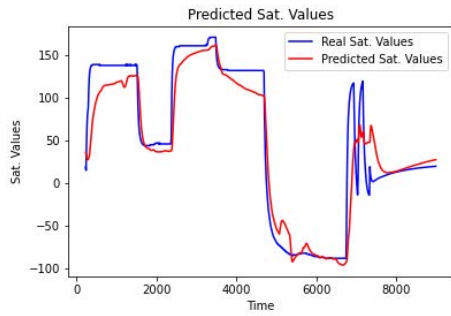
GP - Component 52



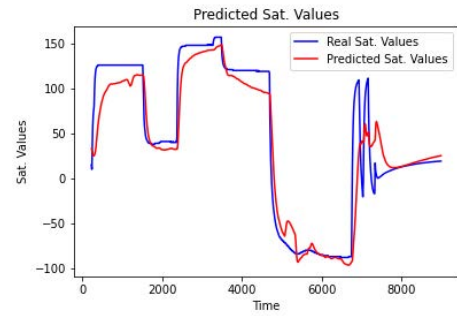
MF - Component 53



GP - Component 54

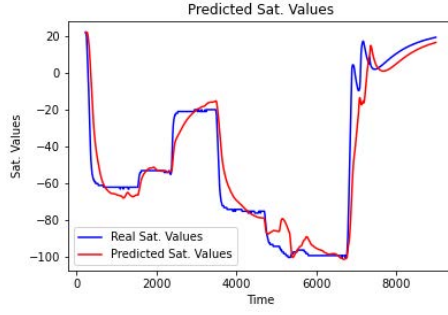


MF - Component 55

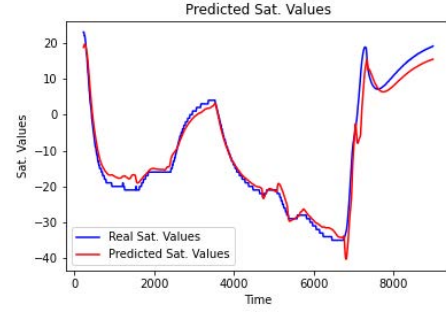


GP - Component 56

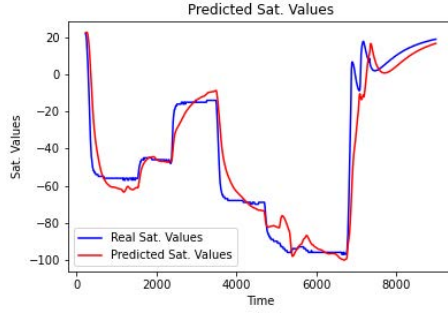
Figure A.7: GRU results, SDS-4 data. Blue lines (—) indicate the real satellite values and red lines (—) the predicted data points for components 48-56.



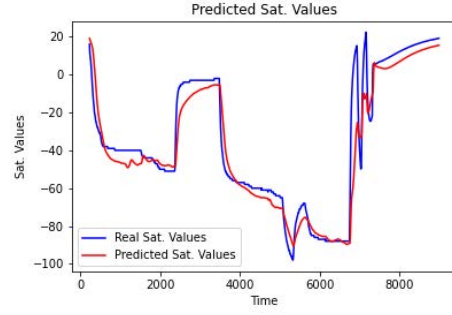
MF - Component 57



GP - Component 58



MF - Component 59



GP - Component 60

Figure A.8: GRU results, SDS-4 data. Blue lines (-) indicate the real satellite values and red lines (-) the predicted data points for components 57-60.

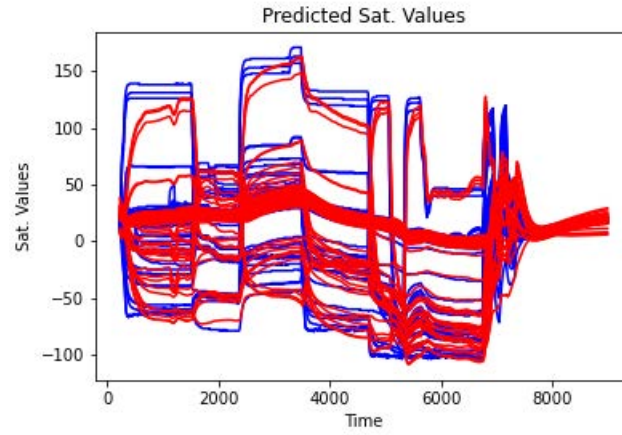
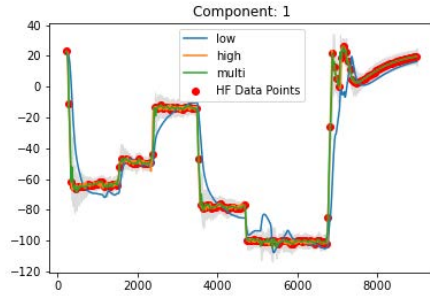
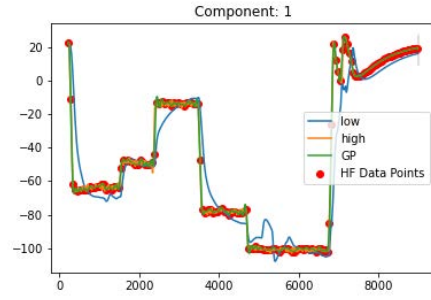


Figure A.9: GRU results, SDS-4 data. Blue lines (-) indicate the real satellite values and red lines (-) the predicted data points for all components.

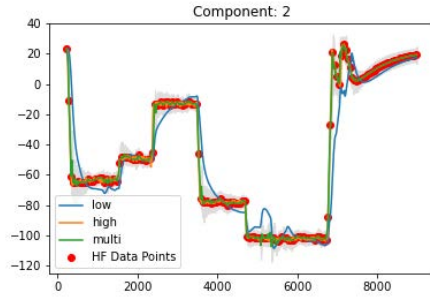
A.3 Uniform SDS-4 HF Point Placement



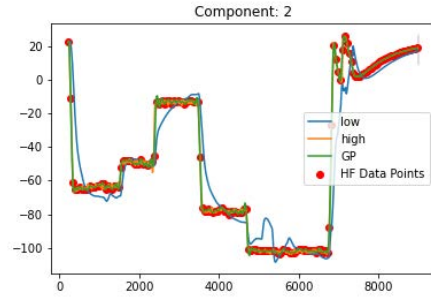
MF - Component 1



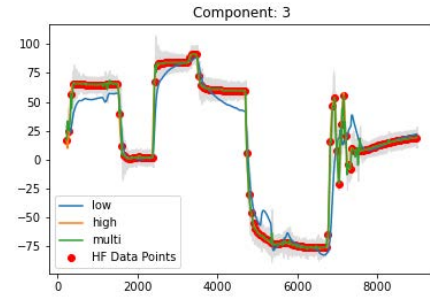
GP - Component 1



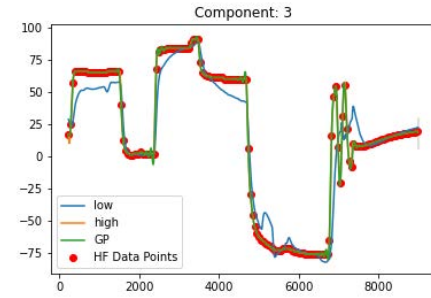
MF - Component 2



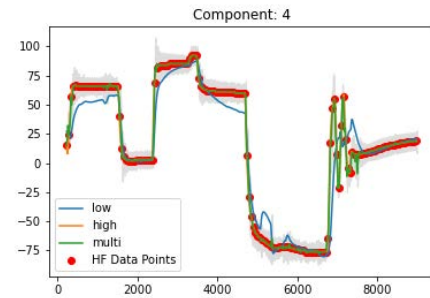
GP - Component 2



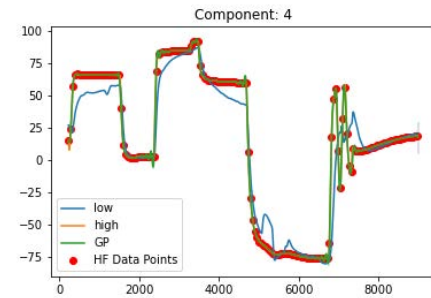
MF - Component 3



GP - Component 3

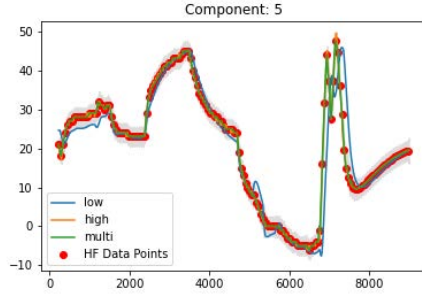


MF - Component 4

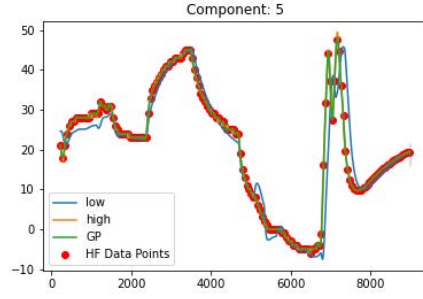


GP - Component 4

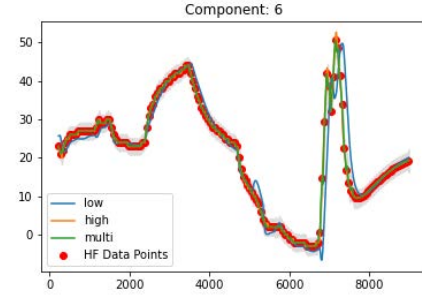
Figure A.10: SDS-4 data. Multifidelity vs Gaussian process. Non-random high fidelity data points. Blue lines (-) indicate the LFM, orange lines (-) the HFM, green lines (-) the multifidelity estimation and red dots (•) the HFM data points used and the gray area (■) the confidence interval for components 1-4.



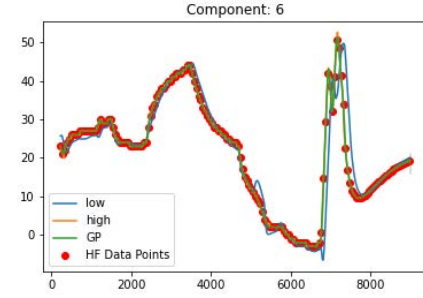
MF - Component 5



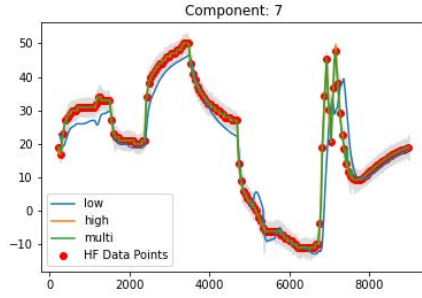
GP - Component 5



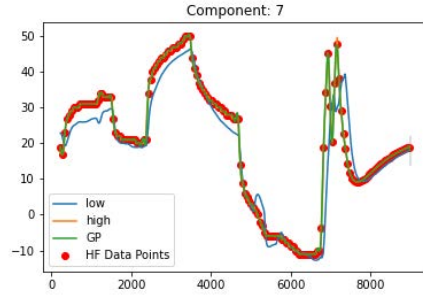
MF - Component 6



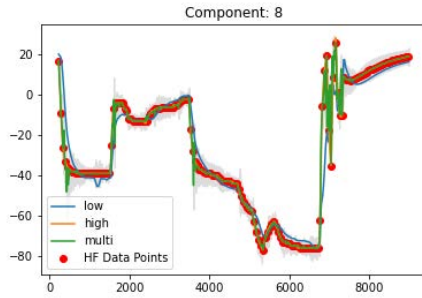
GP - Component 6



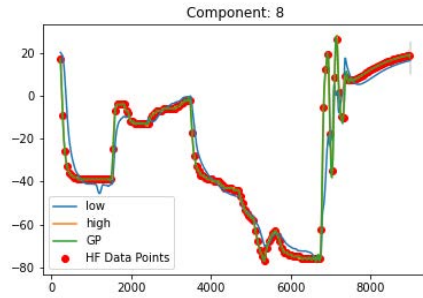
MF - Component 7



GP - Component 7

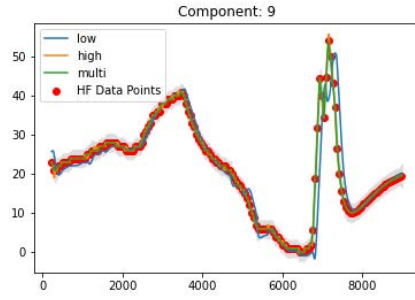


MF - Component 8

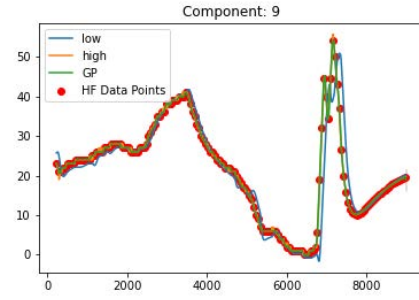


GP - Component 8

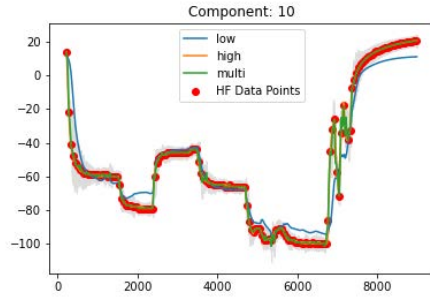
Figure A.11: SDS-4 data. Multifidelity vs Gaussian process. Non-random high fidelity data points. Blue lines (-) indicate the LFM, orange lines (-) the HFM, green lines (-) the multifidelity estimation and red dots (•) the HFM data points used and the gray area (■) the confidence interval for components 5-8.



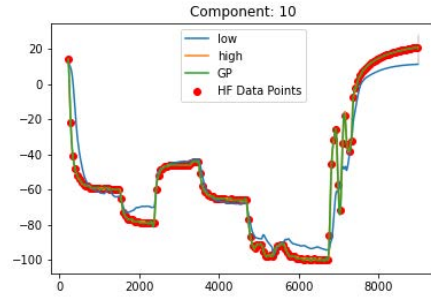
MF - Component 9



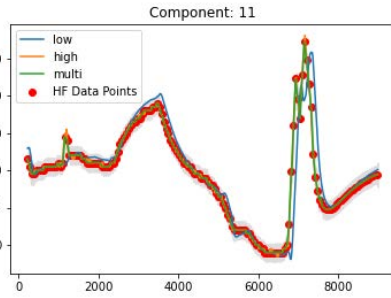
GP - Component 9



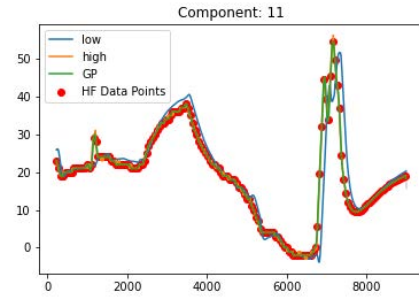
MF - Component 10



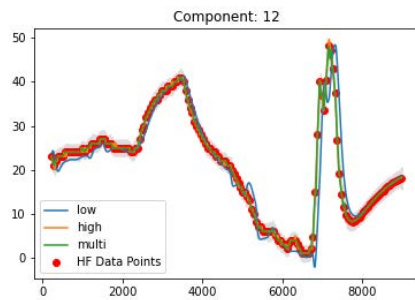
GP - Component 10



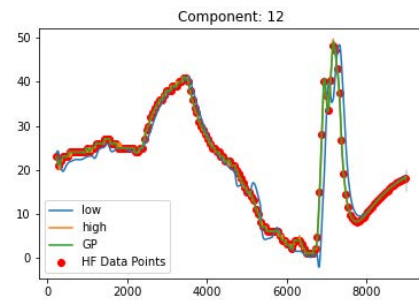
MF - Component 11



GP - Component 11

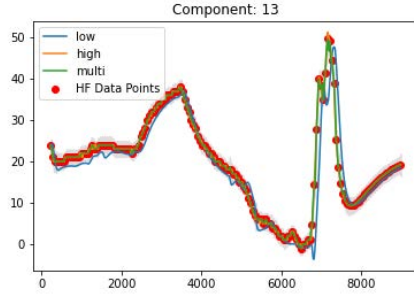


MF - Component 12

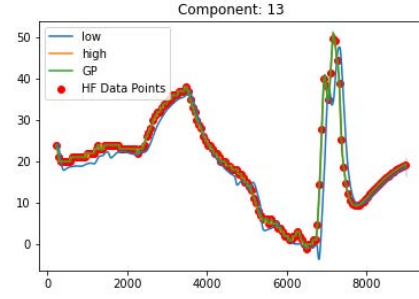


GP - Component 12

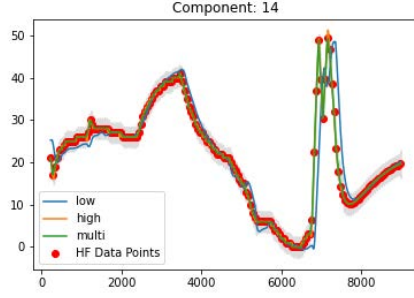
Figure A.12: SDS-4 data. Multifidelity vs Gaussian process. Non-random high fidelity data points. Blue lines (-) indicate the LFM, orange lines (-) the HFM, green lines (-) the multifidelity estimation and red dots (•) the HFM data points used and the gray area (■) the confidence interval for components 9-12.



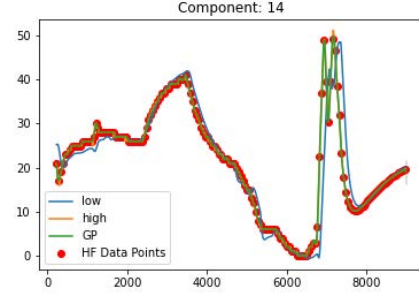
MF - Component 13



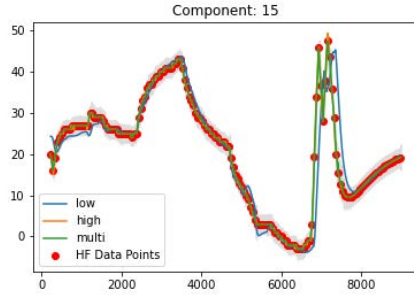
GP - Component 13



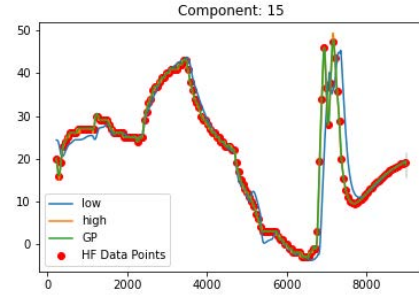
MF - Component 14



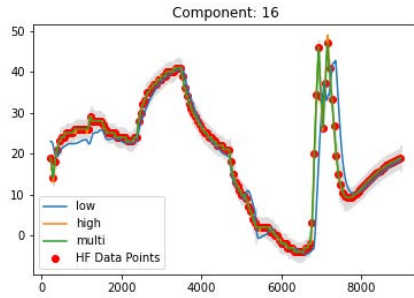
GP - Component 14



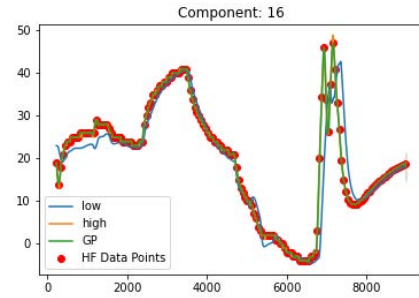
MF - Component 15



GP - Component 15

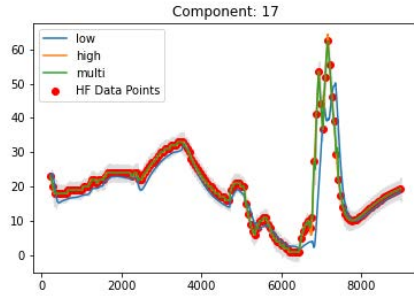


MF - Component 16

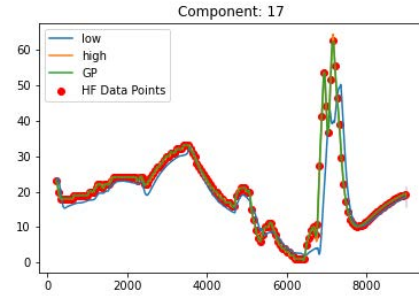


GP - Component 16

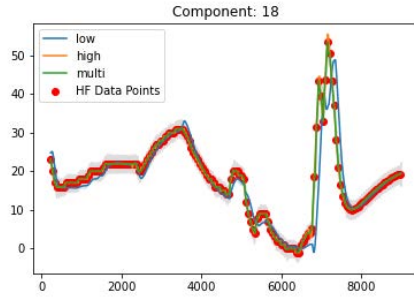
Figure A.13: SDS-4 data. Multifidelity vs Gaussian process. Non-random high fidelity data points. Blue lines (-) indicate the LFM, orange lines (-) the HFM, green lines (-) the multifidelity estimation and red dots (•) the HFM data points used and the gray area (■) the confidence interval for components 13-16.



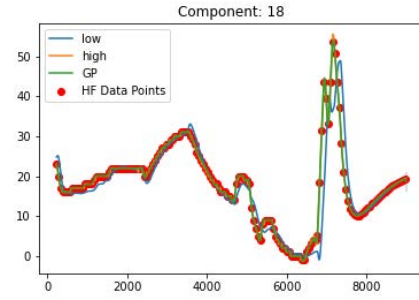
MF - Component 17



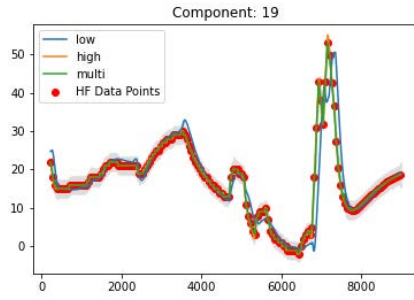
GP - Component 17



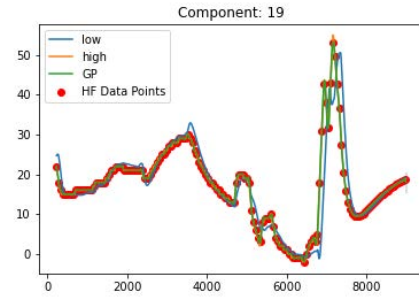
MF - Component 18



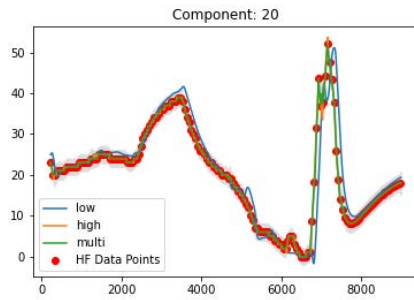
GP - Component 18



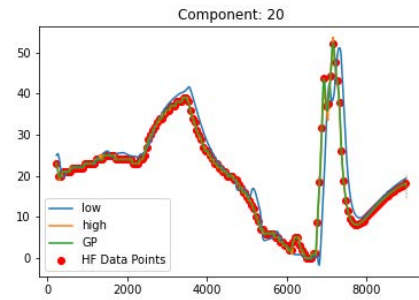
MF - Component 19



GP - Component 19

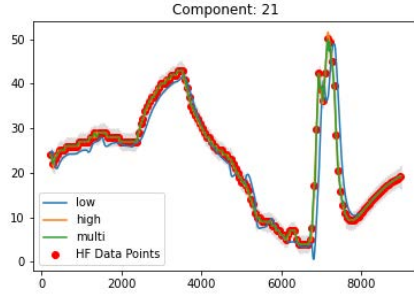


MF - Component 20

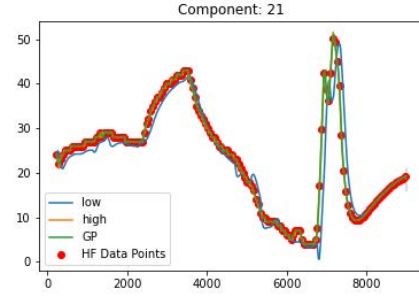


GP - Component 20

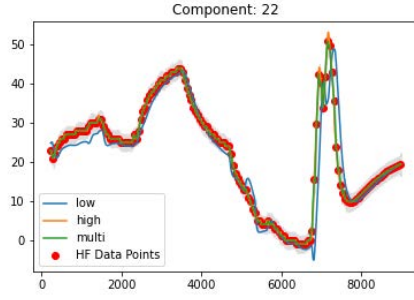
Figure A.14: SDS-4 data. Multifidelity vs Gaussian process. Non-random high fidelity data points. Blue lines (-) indicate the LFM, orange lines (-) the HFM, green lines (-) the multifidelity estimation and red dots (•) the HFM data points used and the gray area (■) the confidence interval for components 17-20.



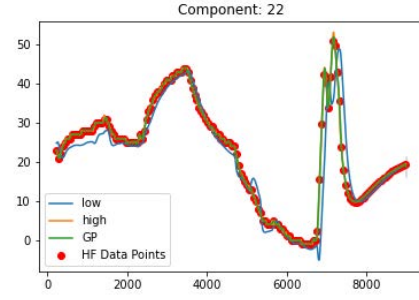
MF - Component 21



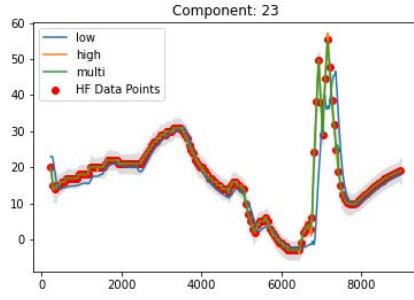
GP - Component 21



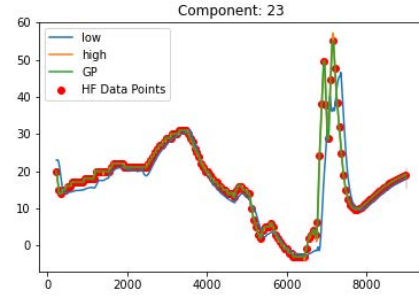
MF - Component 22



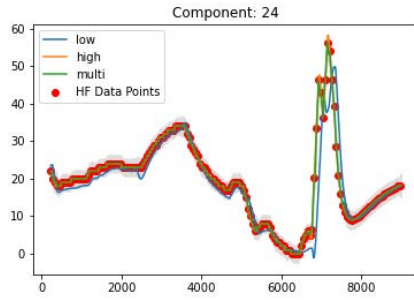
GP - Component 22



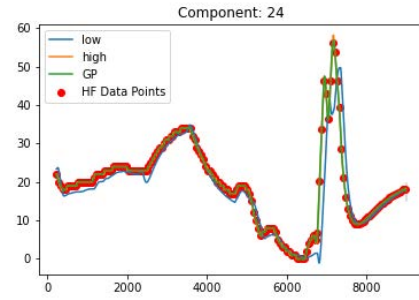
MF - Component 23



GP - Component 23

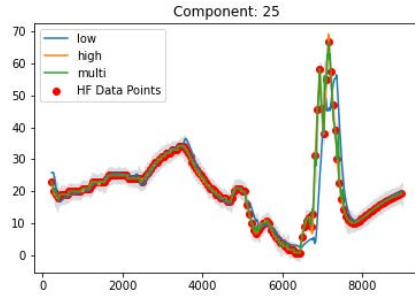


MF - Component 24

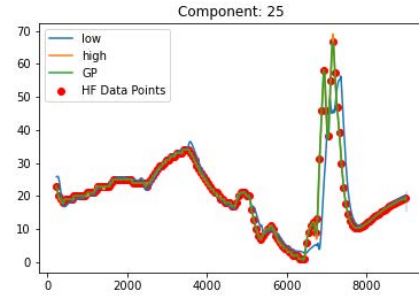


GP - Component 24

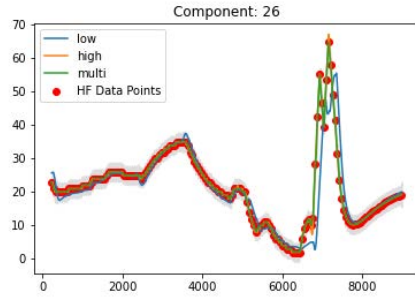
Figure A.15: SDS-4 data. Multifidelity vs Gaussian process. Non-random high fidelity data points. Blue lines (-) indicate the LFM, orange lines (-) the HFM, green lines (-) the multifidelity estimation and red dots (•) the HFM data points used and the gray area (■) the confidence interval for components 21-24.



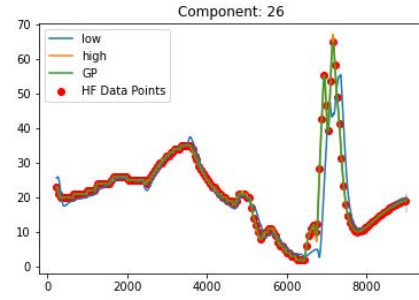
MF - Component 25



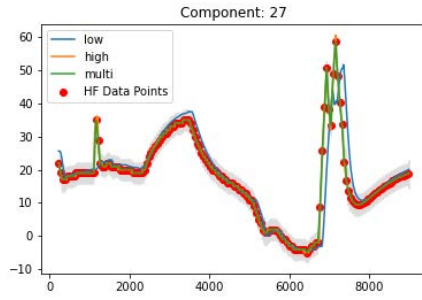
GP - Component 25



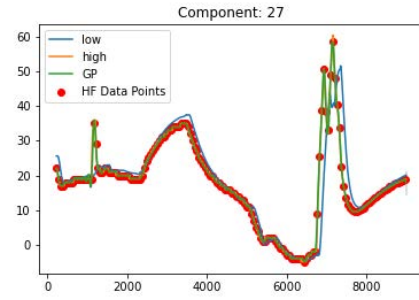
MF - Component 26



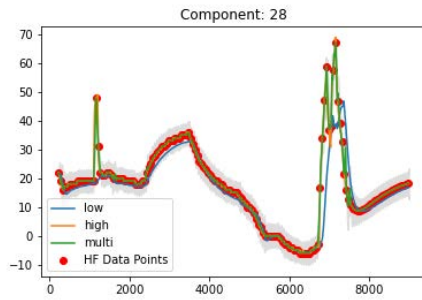
GP - Component 26



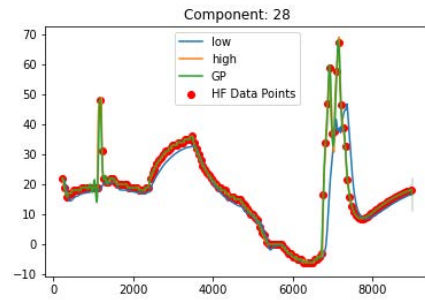
MF - Component 27



GP - Component 27

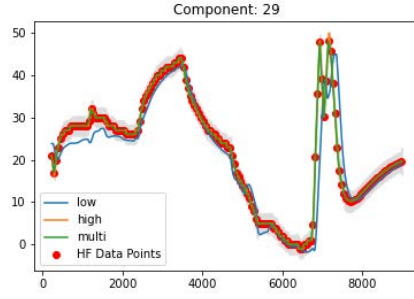


MF - Component 28

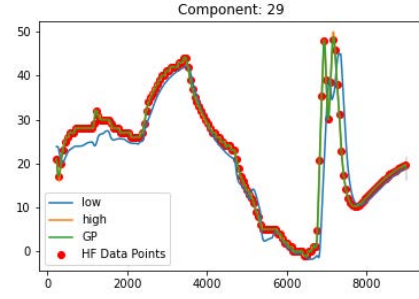


GP - Component 28

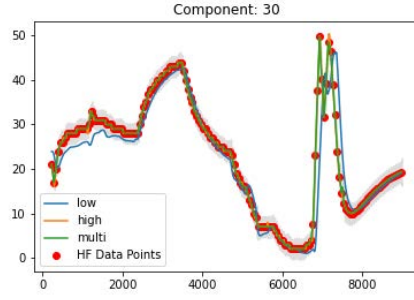
Figure A.16: SDS-4 data. Multifidelity vs Gaussian process. Non-random high fidelity data points. Blue lines (-) indicate the LFM, orange lines (-) the HFM, green lines (-) the multifidelity estimation and red dots (•) the HFM data points used and the gray area (■) the confidence interval for components 25-28.



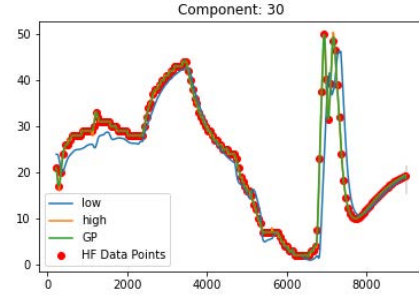
MF - Component 29



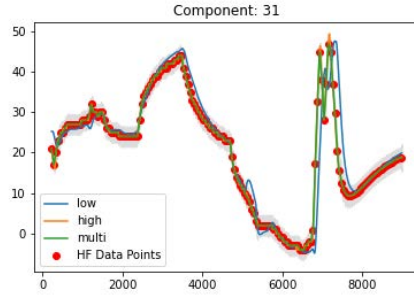
GP - Component 29



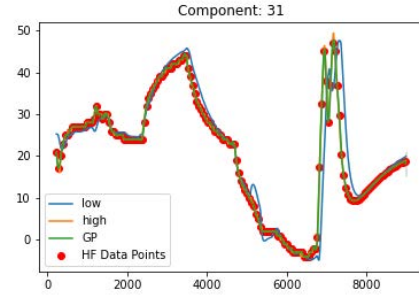
MF - Component 30



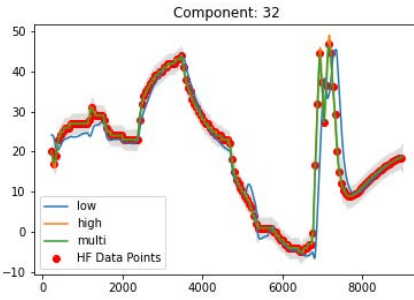
GP - Component 30



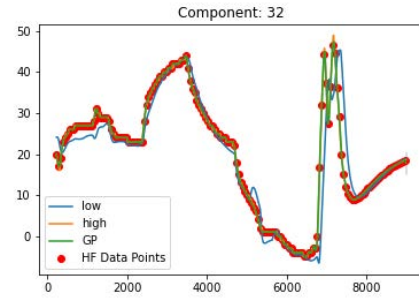
MF - Component 31



GP - Component 31

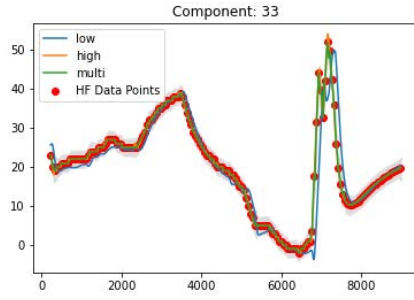


MF - Component 32

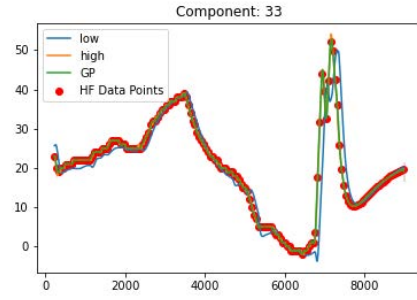


GP - Component 32

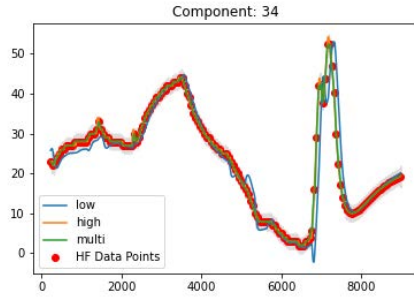
Figure A.17: SDS-4 data. Multifidelity vs Gaussian process. Non-random high fidelity data points. Blue lines (-) indicate the LFM, orange lines (-) the HFM, green lines (-) the multifidelity estimation and red dots (•) the HFM data points used and the gray area (■) the confidence interval for components 29-32.



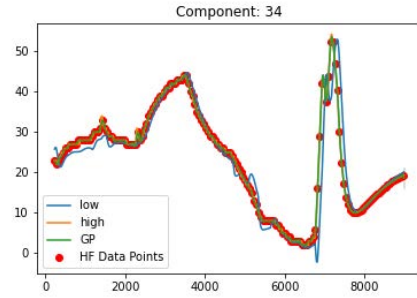
MF - Component 33



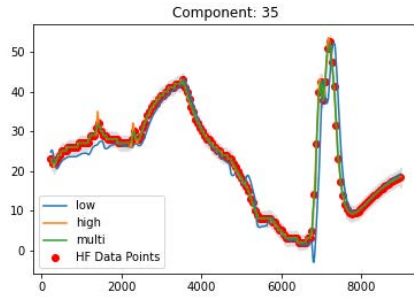
GP - Component 33



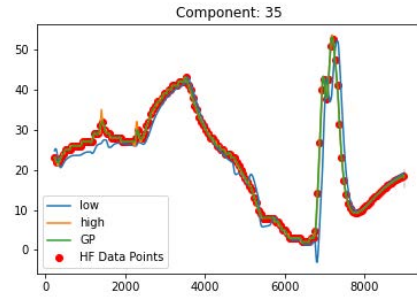
MF - Component 34



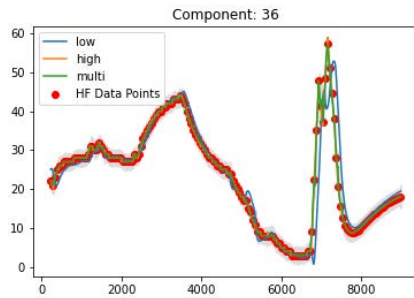
GP - Component 34



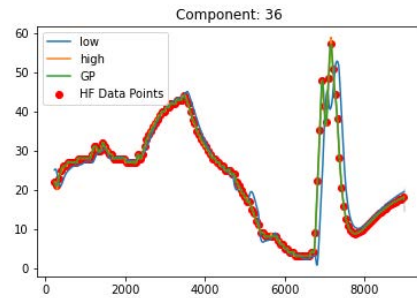
MF - Component 35



GP - Component 35

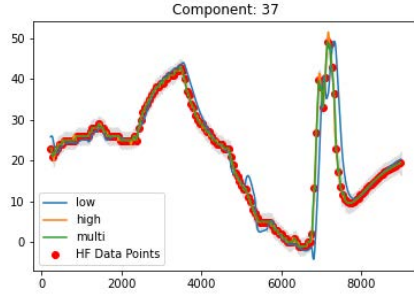


MF - Component 36

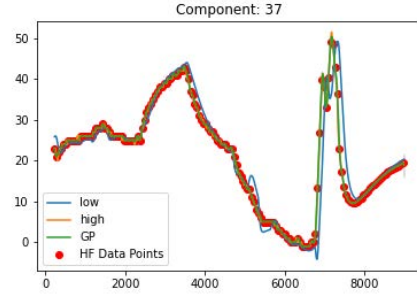


GP - Component 36

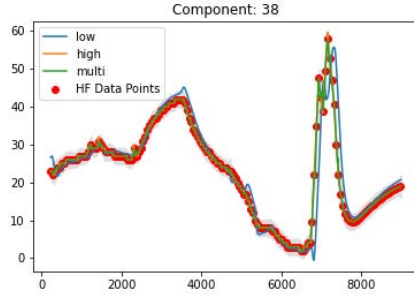
Figure A.18: SDS-4 data. Multifidelity vs Gaussian process. Non-random high fidelity data points. Blue lines (-) indicate the LFM, orange lines (-) the HFM, green lines (-) the multifidelity estimation and red dots (•) the HFM data points used and the gray area (■) the confidence interval for components 33-36.



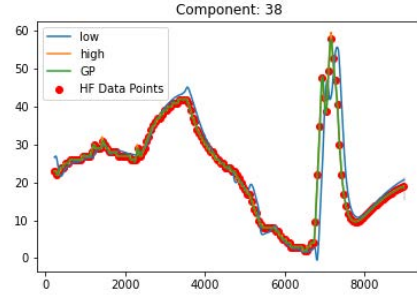
MF - Component 37



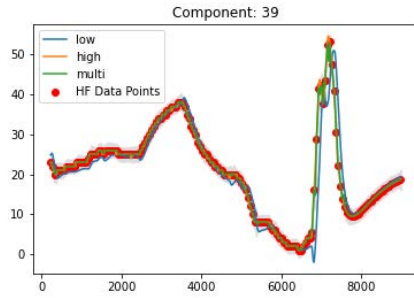
GP - Component 37



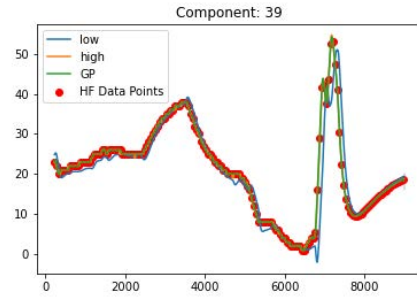
MF - Component 38



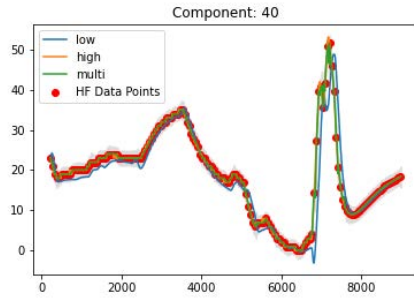
GP - Component 38



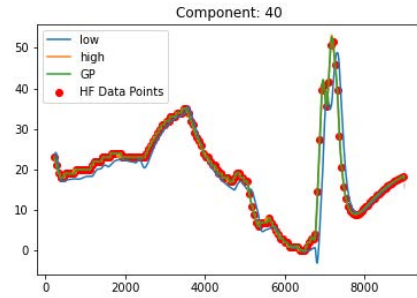
MF - Component 39



GP - Component 39

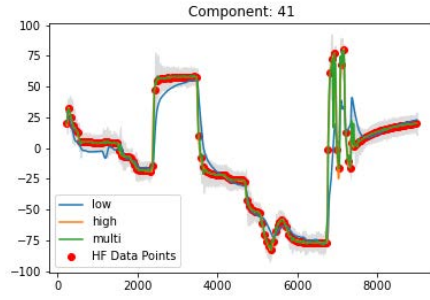


MF - Component 40

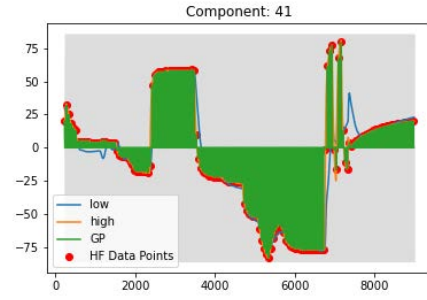


GP - Component 40

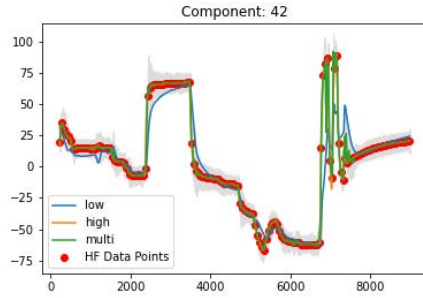
Figure A.19: SDS-4 data. Multifidelity vs Gaussian process. Non-random high fidelity data points. Blue lines (-) indicate the LFM, orange lines (-) the HFM, green lines (-) the multifidelity estimation and red dots (•) the HFM data points used and the gray area (■) the confidence interval for components 37-40.



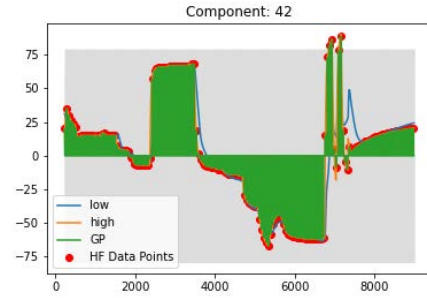
MF - Component 41



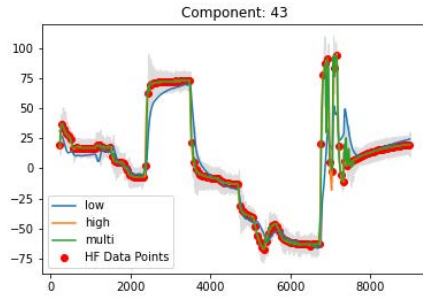
GP - Component 41



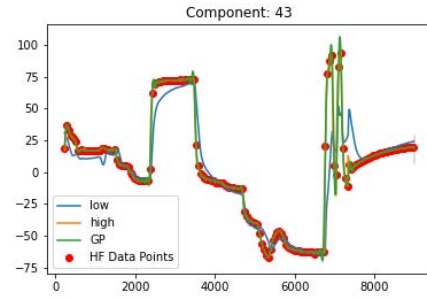
MF - Component 42



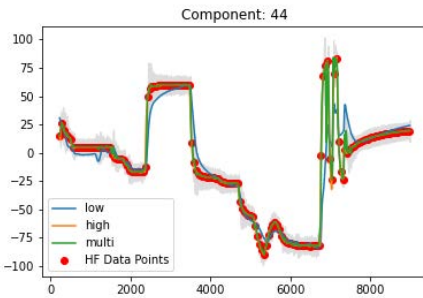
GP - Component 42



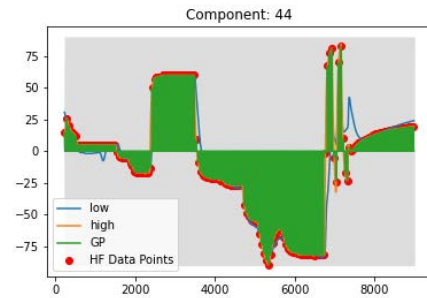
MF - Component 43



GP - Component 43

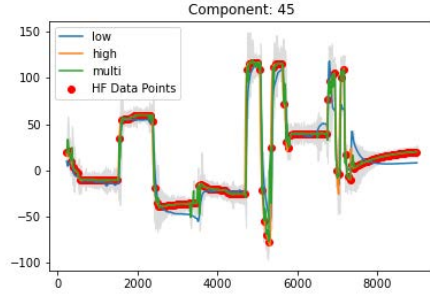


MF - Component 44

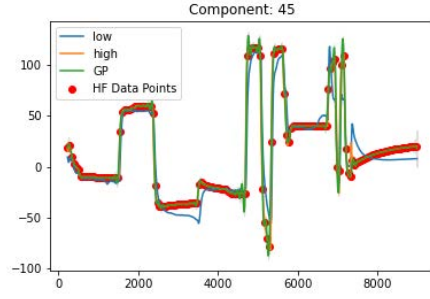


GP - Component 44

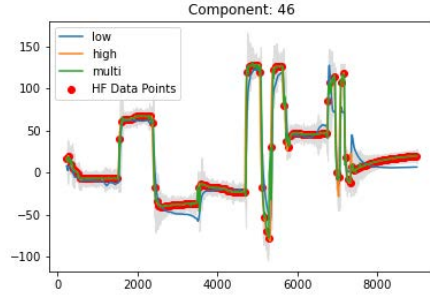
Figure A.20: SDS-4 data. Multifidelity vs Gaussian process. Non-random high fidelity data points. Blue lines (-) indicate the LFM, orange lines (-) the HFM, green lines (-) the multifidelity estimation and red dots (•) the HFM data points used and the gray area (■) the confidence interval for components 41-44.



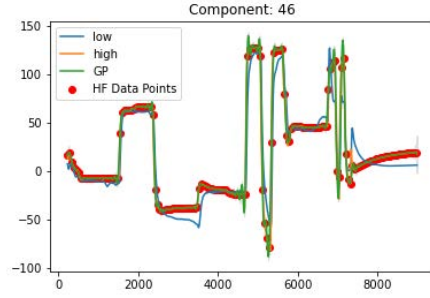
MF - Component 45



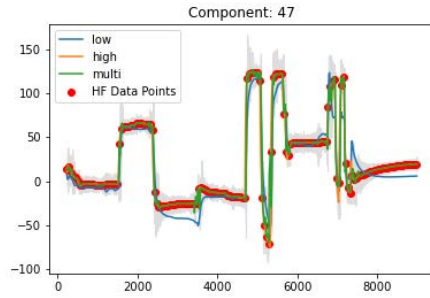
GP - Component 45



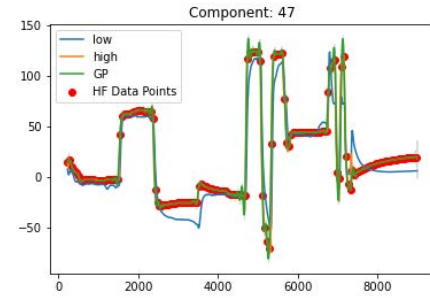
MF - Component 46



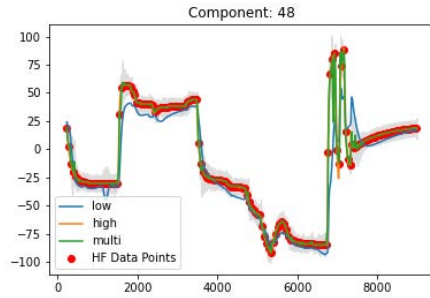
GP - Component 46



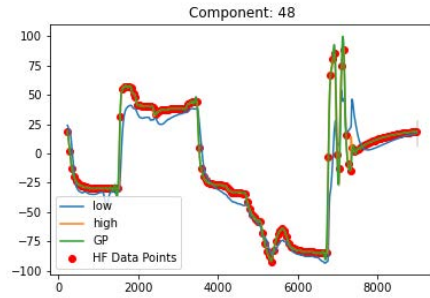
MF - Component 47



GP - Component 47

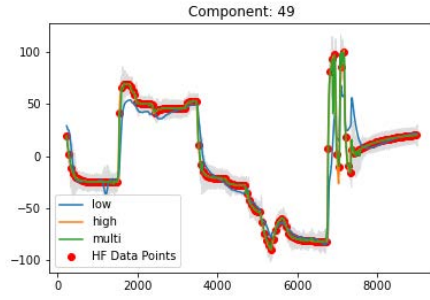


MF - Component 48

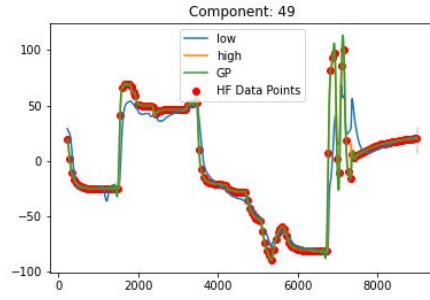


GP - Component 48

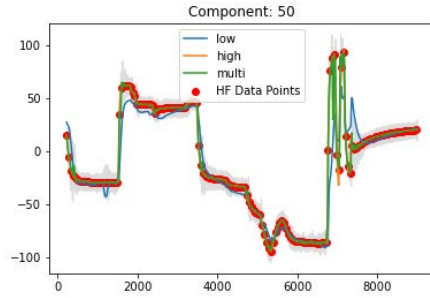
Figure A.21: SDS-4 data. Multifidelity vs Gaussian process. Non-random high fidelity data points. Blue lines (-) indicate the LFM, orange lines (-) the HFM, green lines (-) the multifidelity estimation and red dots (•) the HFM data points used and the gray area (■) the confidence interval for components 45-48.



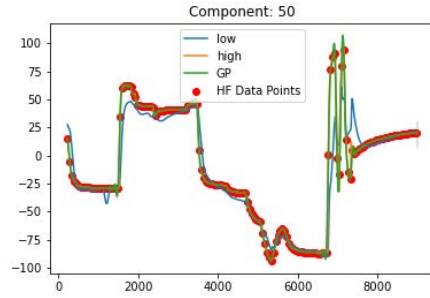
MF - Component 49



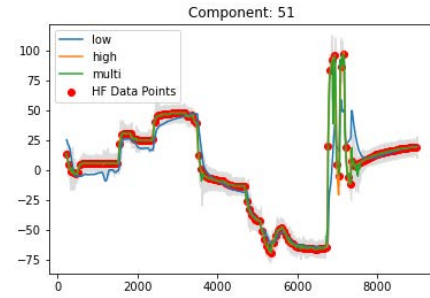
GP - Component 49



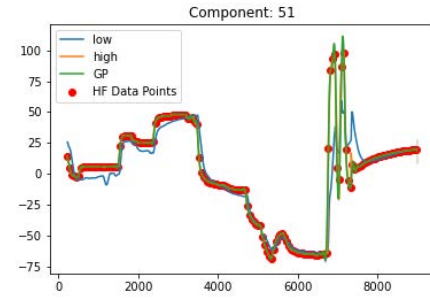
MF - Component 50



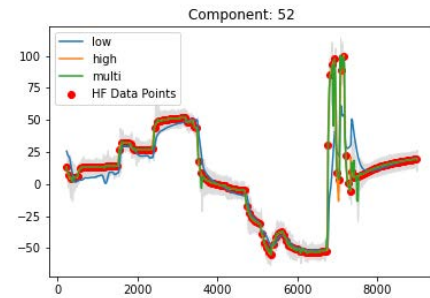
GP - Component 50



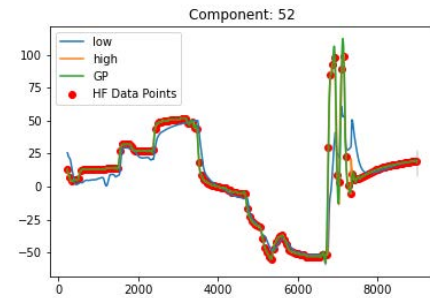
MF - Component 51



GP - Component 51

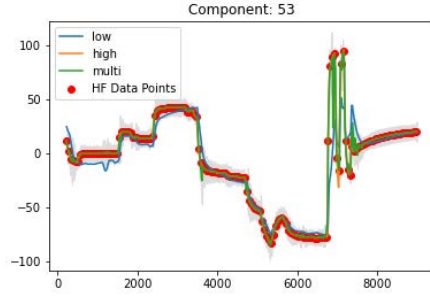


MF - Component 52

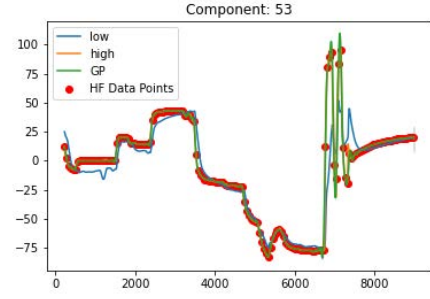


GP - Component 52

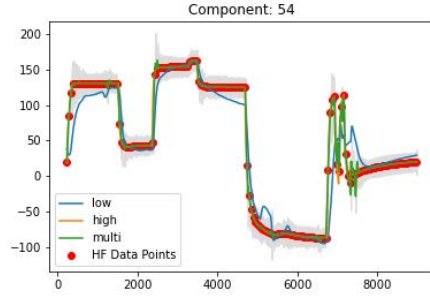
Figure A.22: SDS-4 data. Multifidelity vs Gaussian process. Non-random high fidelity data points. Blue lines (-) indicate the LFM, orange lines (-) the HFM, green lines (-) the multifidelity estimation and red dots (•) the HFM data points used and the gray area (■) the confidence interval for components 49-52.



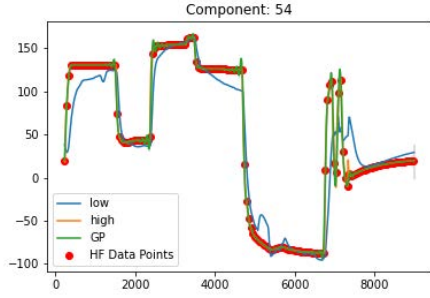
MF - Component 53



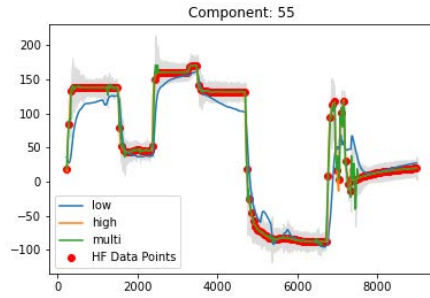
GP - Component 53



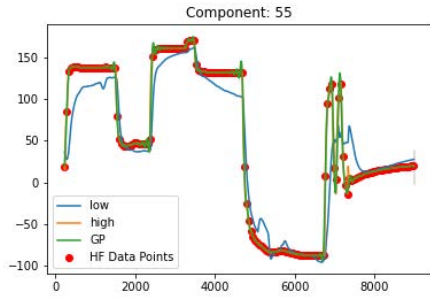
MF - Component 54



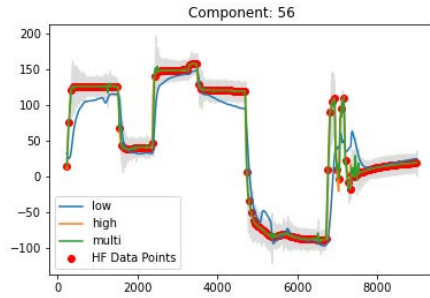
GP - Component 54



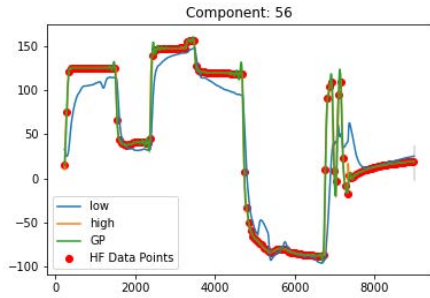
MF - Component 55



GP - Component 55

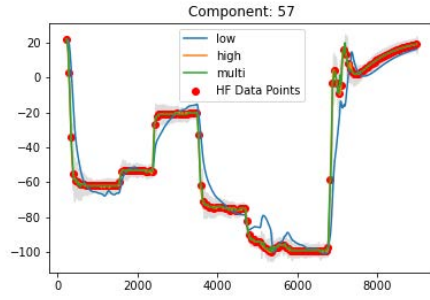


MF - Component 56

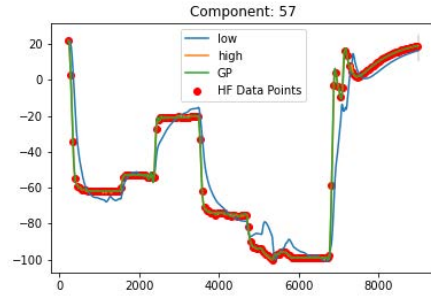


GP - Component 56

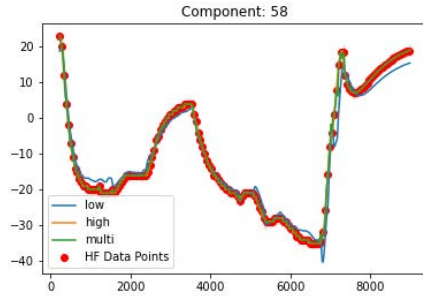
Figure A.23: SDS-4 data. Multifidelity vs Gaussian process. Non-random high fidelity data points. Blue lines (-) indicate the LFM, orange lines (-) the HFM, green lines (-) the multifidelity estimation and red dots (•) the HFM data points used and the gray area (■) the confidence interval for components 53-56.



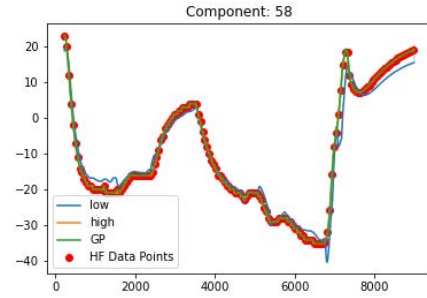
MF - Component 57



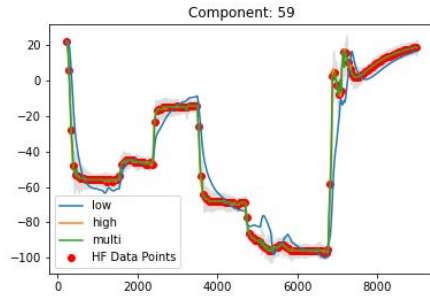
GP - Component 57



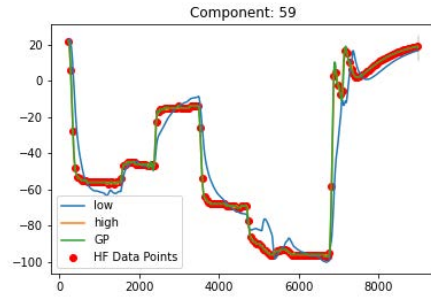
MF - Component 58



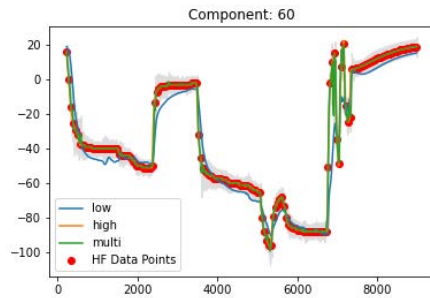
GP - Component 58



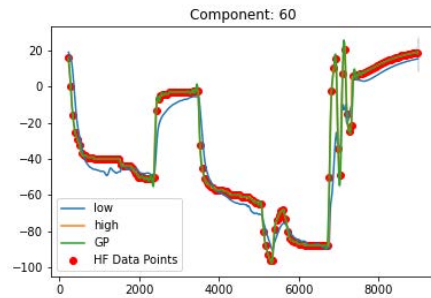
MF - Component 59



GP - Component 59



MF - Component 60



GP - Component 60

Figure A.24: SDS-4 data. Multifidelity vs Gaussian process. Non-random high fidelity data points. Blue lines (-) indicate the LFM, orange lines (-) the HFM, green lines (-) the multifidelity estimation and red dots (•) the HFM data points used and the gray area (■) the confidence interval for components 57-60.

A.4 Random SDS-4 HF Points Placement

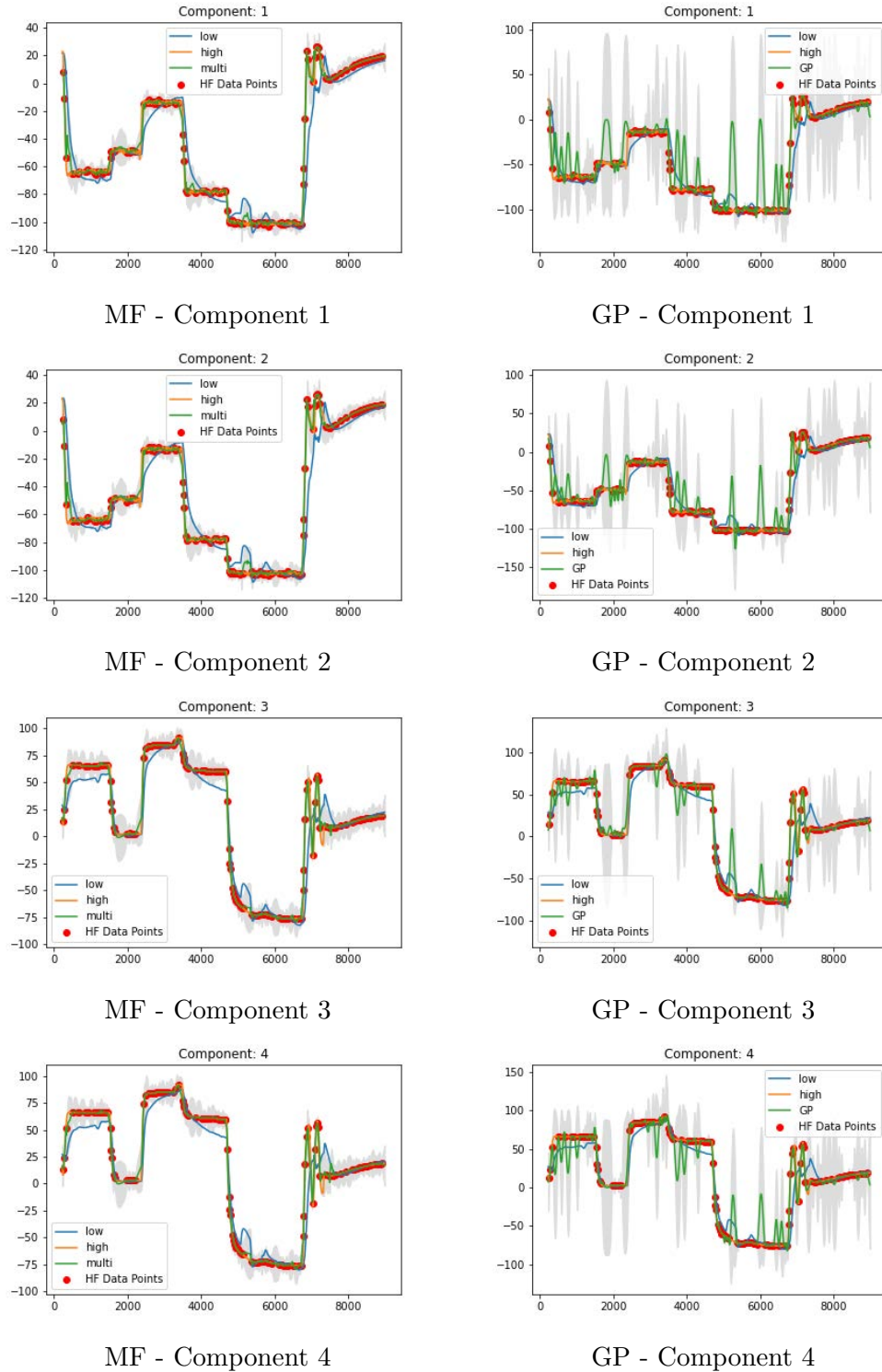
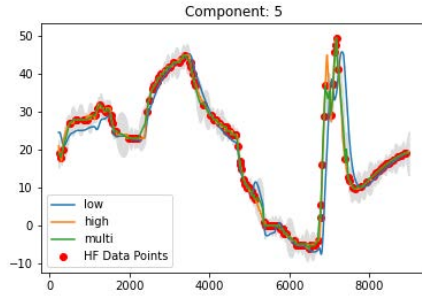
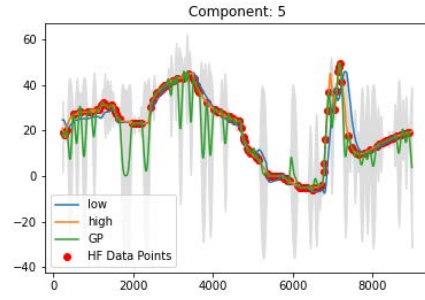


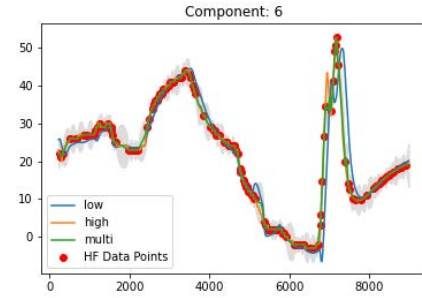
Figure A.25: SDS-4 data. Multifidelity vs Gaussian process. Random high fidelity data points. Blue lines (-) indicate the LFM, orange lines (-) the HFM, green lines (-) the multifidelity estimation and red dots (•) the HFM data points used and the gray area (■) the confidence interval for components 1-4.



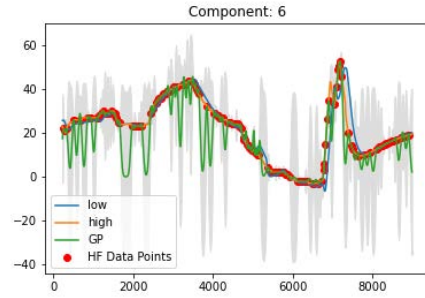
MF - Component 5



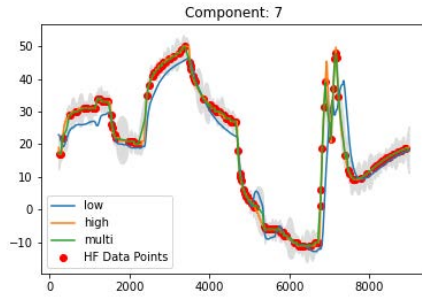
GP - Component 5



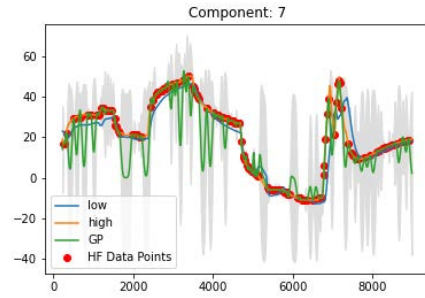
MF - Component 6



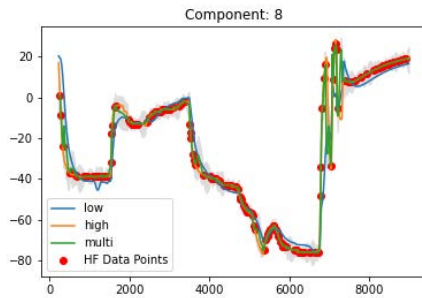
GP - Component 6



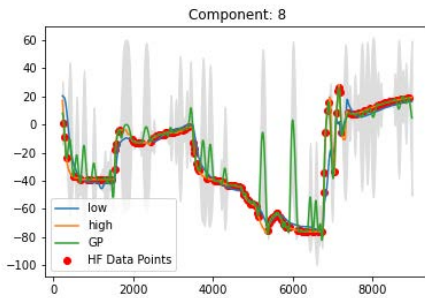
MF - Component 7



GP - Component 7

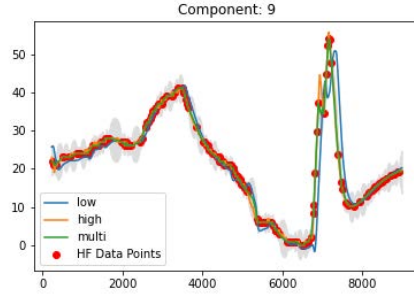


MF - Component 8

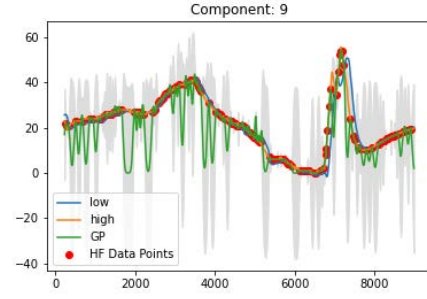


GP - Component 8

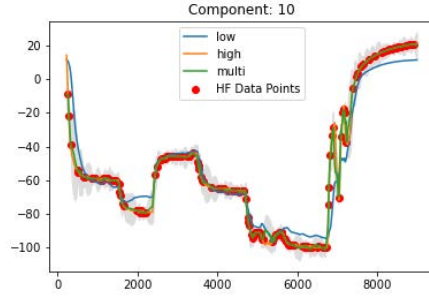
Figure A.26: SDS-4 data. Multifidelity vs Gaussian process. Random high fidelity data points. Blue lines (-) indicate the LFM, orange lines (-) the HFM, green lines (-) the multifidelity estimation and red dots (•) the HFM data points used and the gray area (■) the confidence interval for components 5-8.



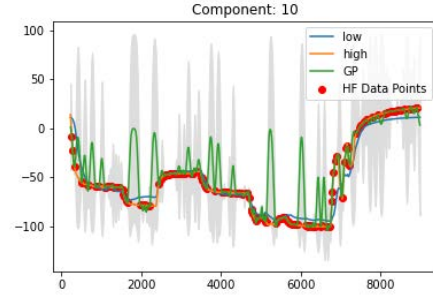
MF - Component 9



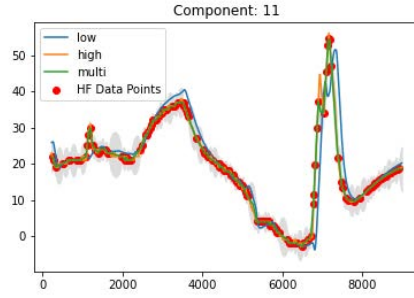
GP - Component 9



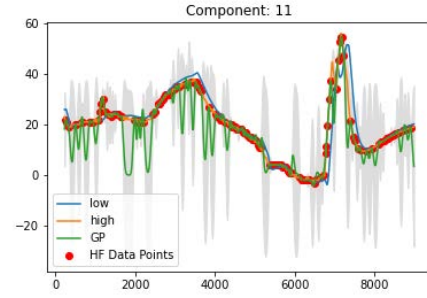
MF - Component 10



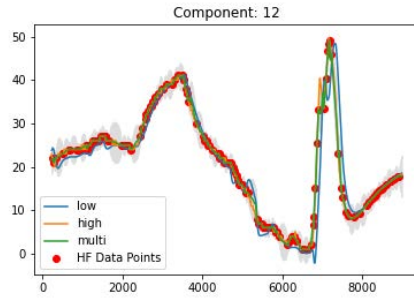
GP - Component 10



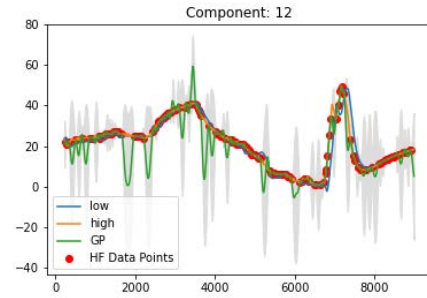
MF - Component 11



GP - Component 11

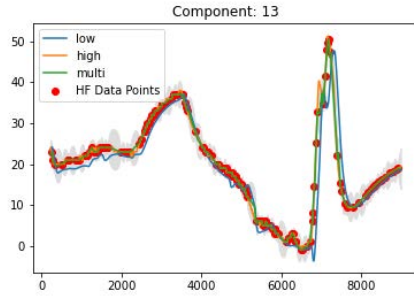


MF - Component 12

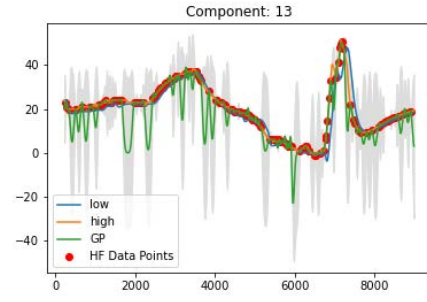


GP - Component 12

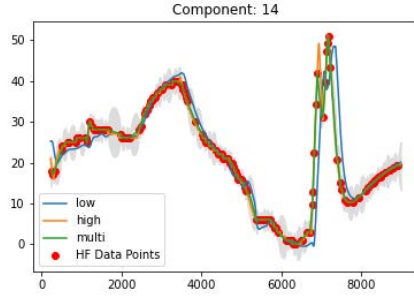
Figure A.27: SDS-4 data. Multifidelity vs Gaussian process. Random high fidelity data points. Blue lines (-) indicate the LFM, orange lines (-) the HFM, green lines (-) the multifidelity estimation and red dots (•) the HFM data points used and the gray area (■) the confidence interval for components 9-12.



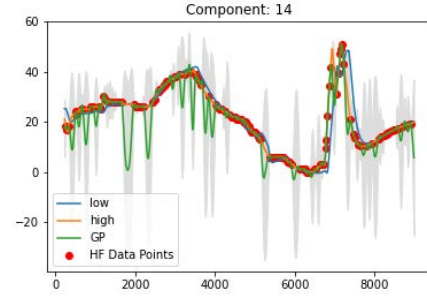
MF - Component 13



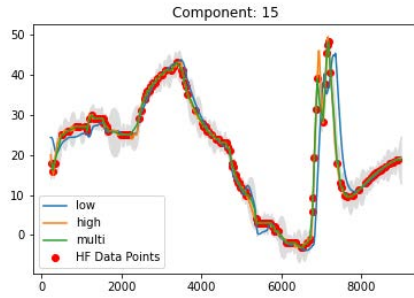
GP - Component 13



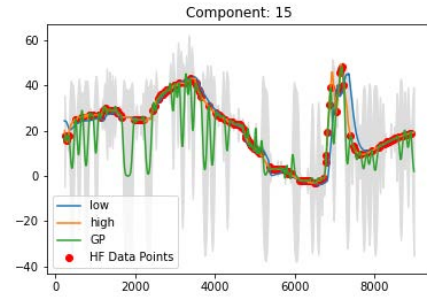
MF - Component 14



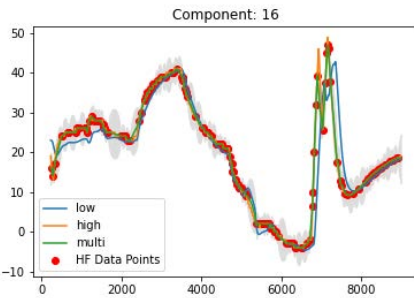
GP - Component 14



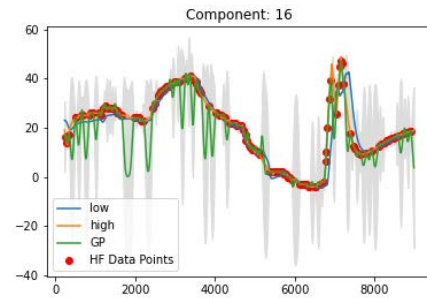
MF - Component 15



GP - Component 15

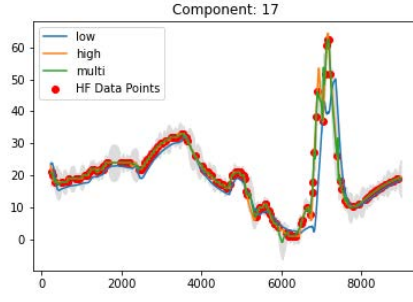


MF - Component 16

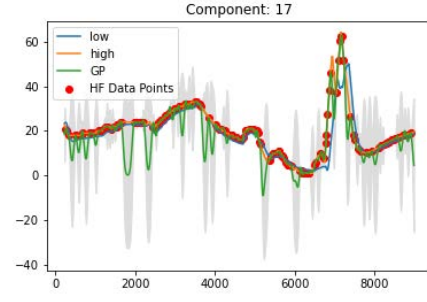


GP - Component 16

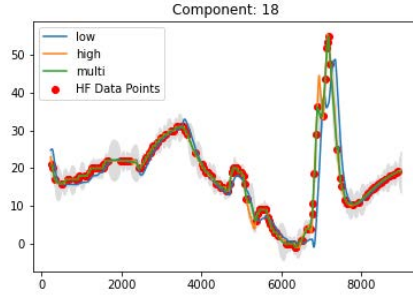
Figure A.28: SDS-4 data. Multifidelity vs Gaussian process. Random high fidelity data points. Blue lines (-) indicate the LFM, orange lines (-) the HFM, green lines (-) the multifidelity estimation and red dots (•) the HFM data points used and the gray area (■) the confidence interval for components 13-16.



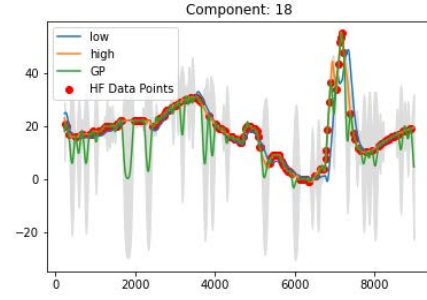
MF - Component 17



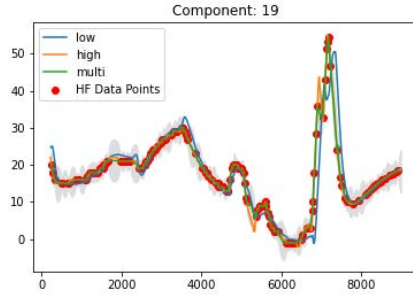
GP - Component 17



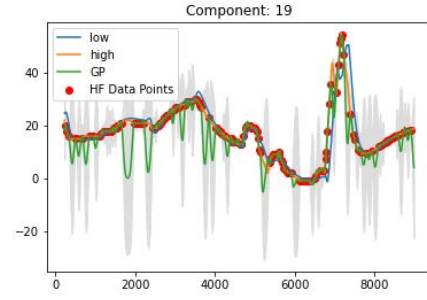
MF - Component 18



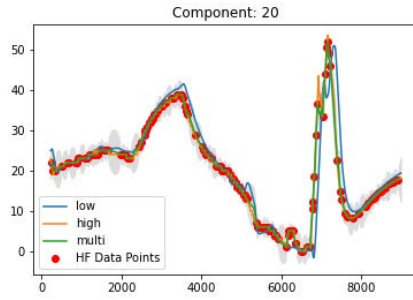
GP - Component 18



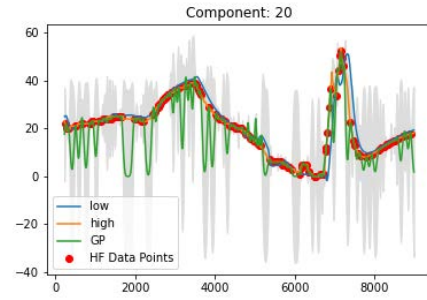
MF - Component 19



GP - Component 19

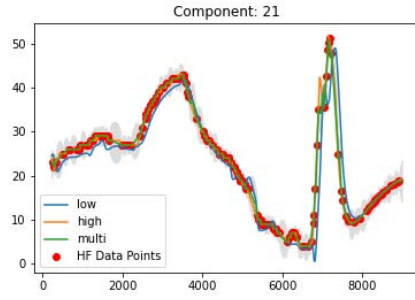


MF - Component 20

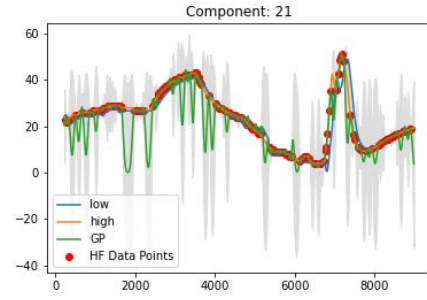


GP - Component 20

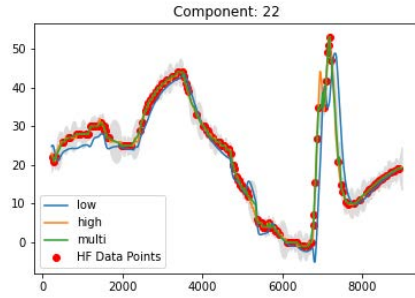
Figure A.29: SDS-4 data. Multifidelity vs Gaussian process. Random high fidelity data points. Blue lines (-) indicate the LFM, orange lines (-) the HFM, green lines (-) the multifidelity estimation and red dots (•) the HFM data points used and the gray area (■) the confidence interval for components 17-20.



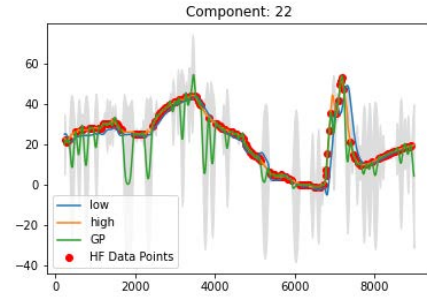
MF - Component 21



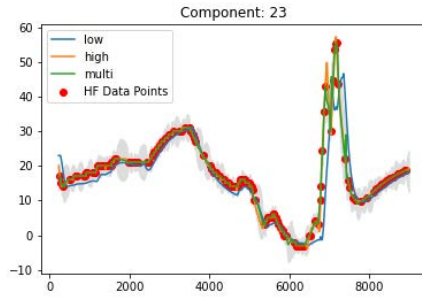
GP - Component 21



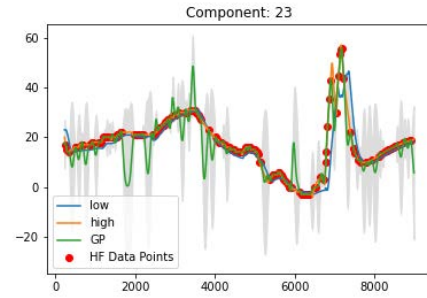
MF - Component 22



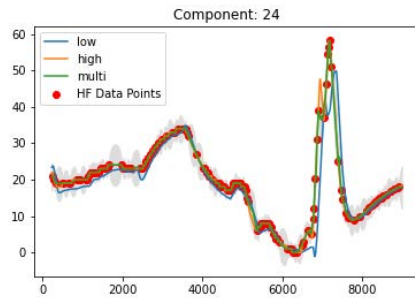
GP - Component 22



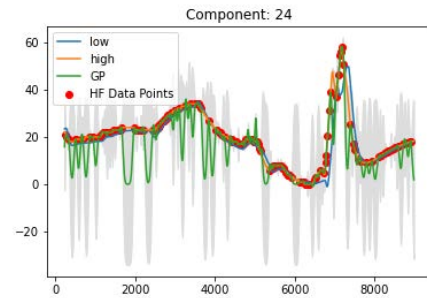
MF - Component 23



GP - Component 23

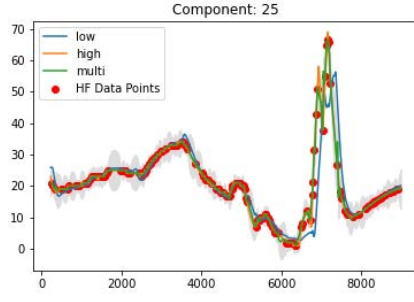


MF - Component 24

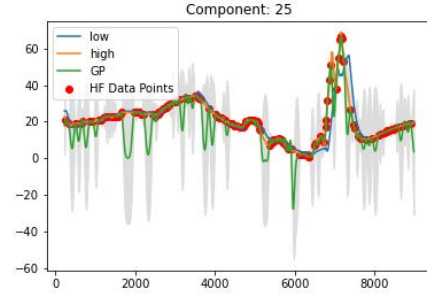


GP - Component 24

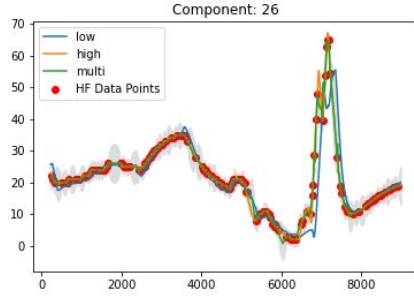
Figure A.30: SDS-4 data. Multifidelity vs Gaussian process. Random high fidelity data points. Blue lines (-) indicate the LFM, orange lines (-) the HFM, green lines (-) the multifidelity estimation and red dots (•) the HFM data points used and the gray area (■) the confidence interval for components 21-24.



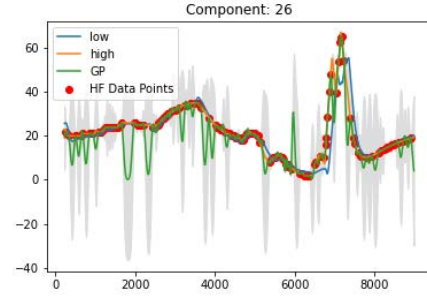
MF - Component 25



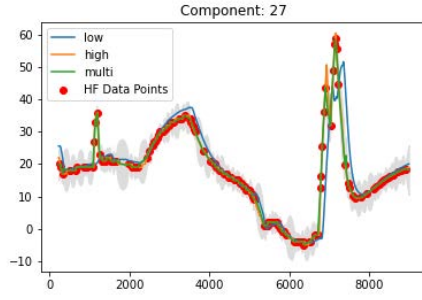
GP - Component 25



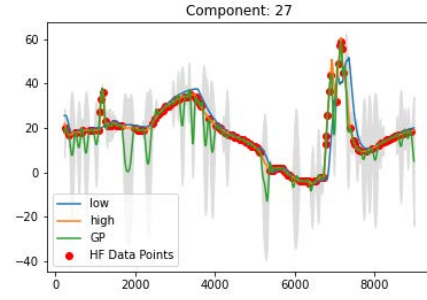
MF - Component 26



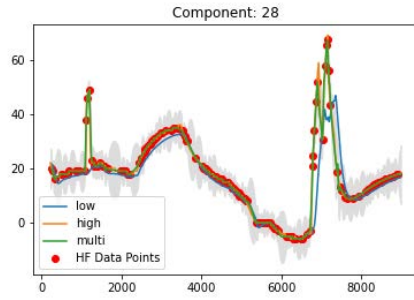
GP - Component 26



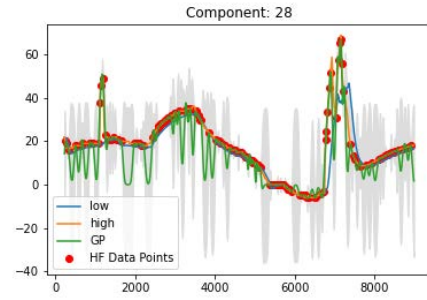
MF - Component 27



GP - Component 27

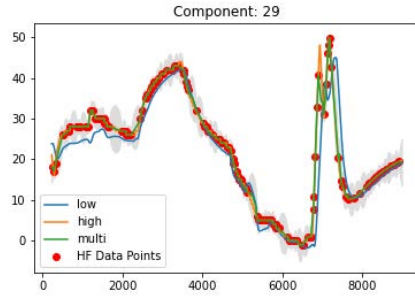


MF - Component 28

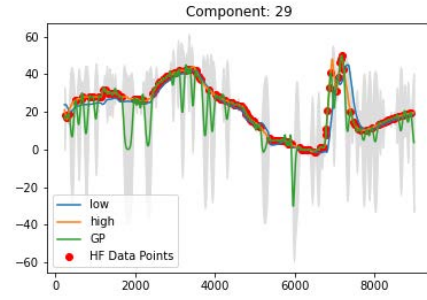


GP - Component 28

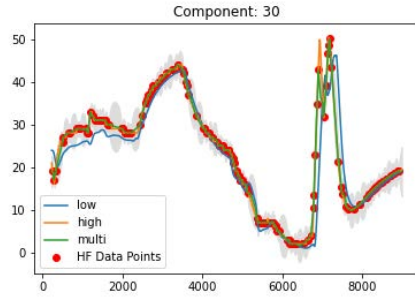
Figure A.31: SDS-4 data. Multifidelity vs Gaussian process. Random high fidelity data points. Blue lines (-) indicate the LFM, orange lines (-) the HFM, green lines (-) the multifidelity estimation and red dots (•) the HFM data points used and the gray area (■) the confidence interval for components 25-28.



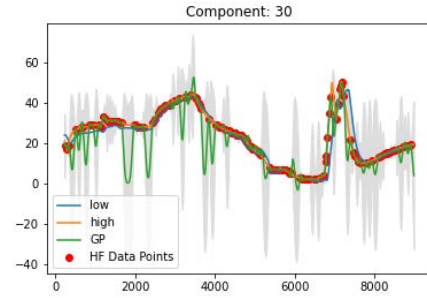
MF - Component 29



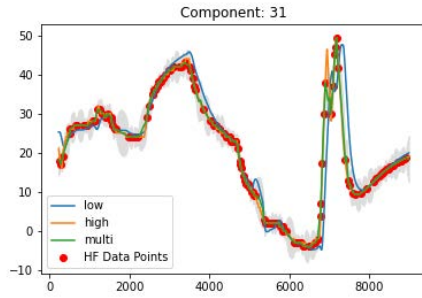
GP - Component 29



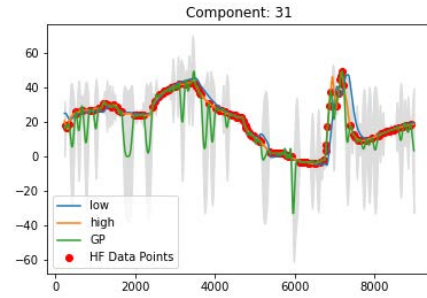
MF - Component 30



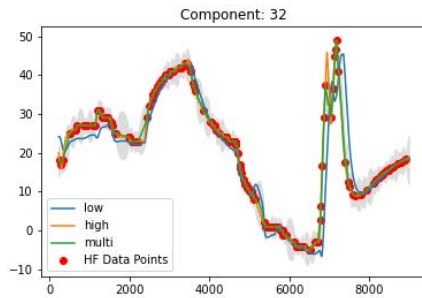
GP - Component 30



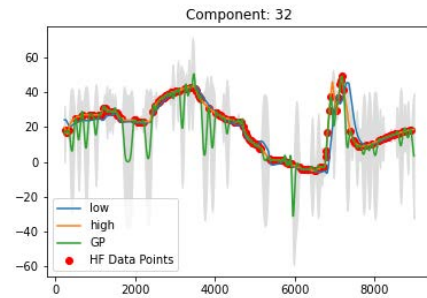
MF - Component 31



GP - Component 31

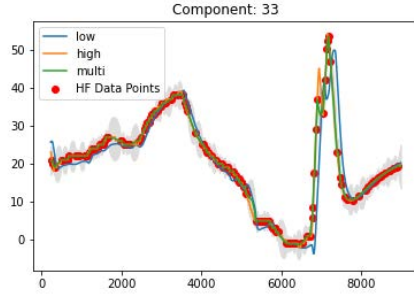


MF - Component 32

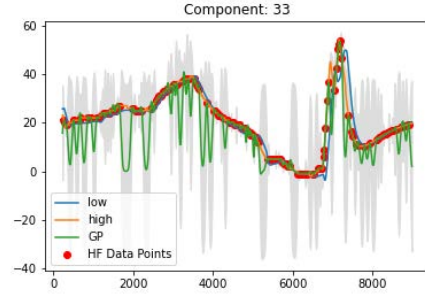


GP - Component 32

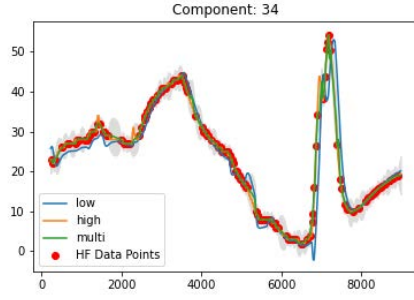
Figure A.32: SDS-4 data. Multifidelity vs Gaussian process. Random high fidelity data points. Blue lines (-) indicate the LFM, orange lines (-) the HFM, green lines (-) the multifidelity estimation and red dots (•) the HFM data points used and the gray area (■) the confidence interval for components 29-32.



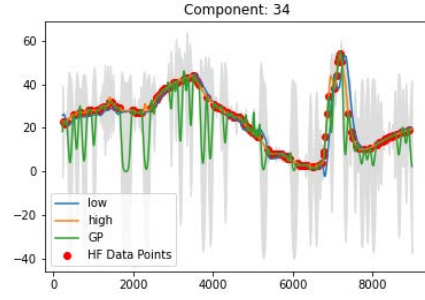
MF - Component 33



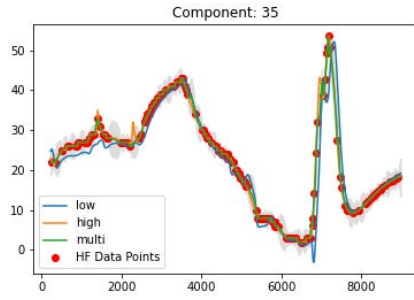
GP - Component 33



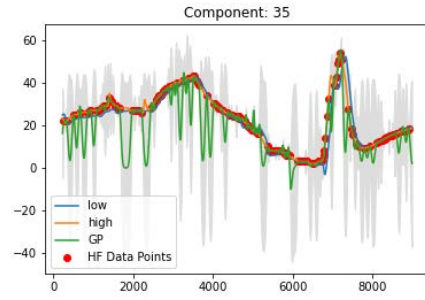
MF - Component 34



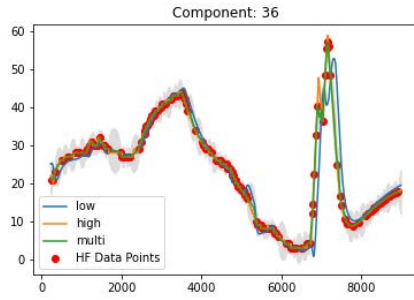
GP - Component 34



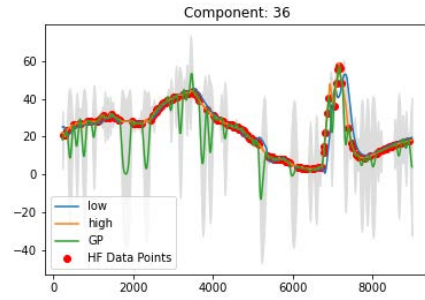
MF - Component 35



GP - Component 35

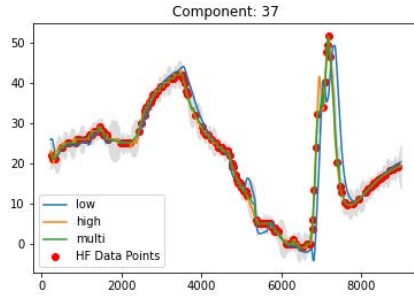


MF - Component 36

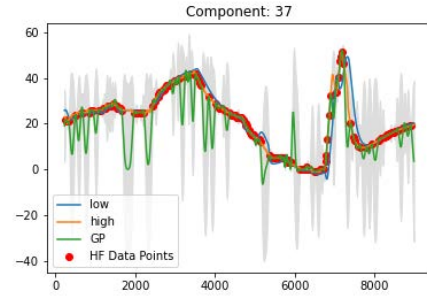


GP - Component 36

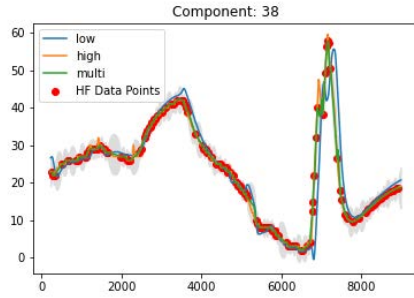
Figure A.33: SDS-4 data. Multifidelity vs Gaussian process. Random high fidelity data points. Blue lines (-) indicate the LFM, orange lines (-) the HFM, green lines (-) the multifidelity estimation and red dots (•) the HFM data points used and the gray area (■) the confidence interval for components 33-36.



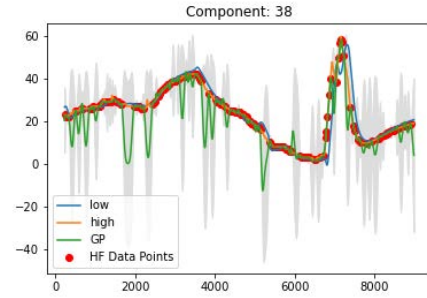
MF - Component 37



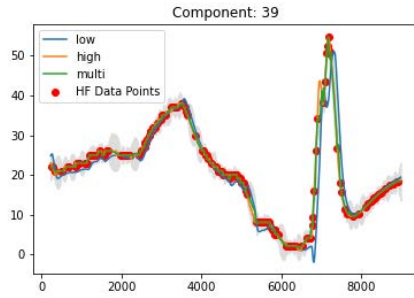
GP - Component 37



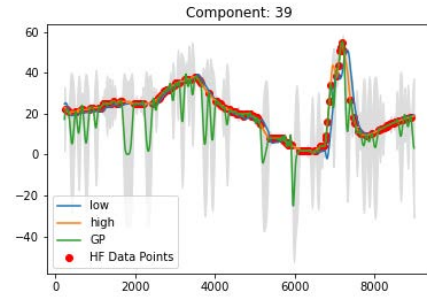
MF - Component 38



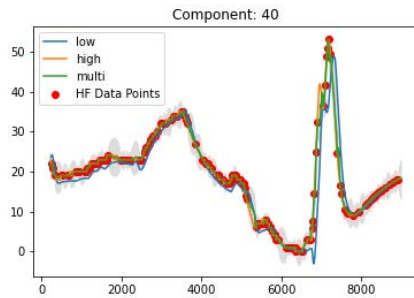
GP - Component 38



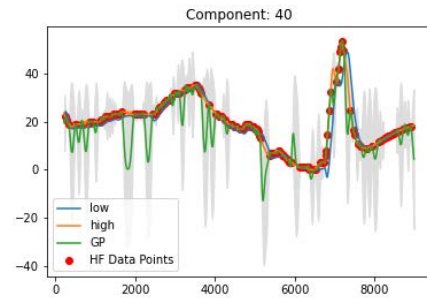
MF - Component 39



GP - Component 39

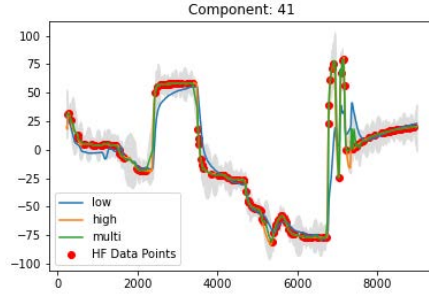


MF - Component 40

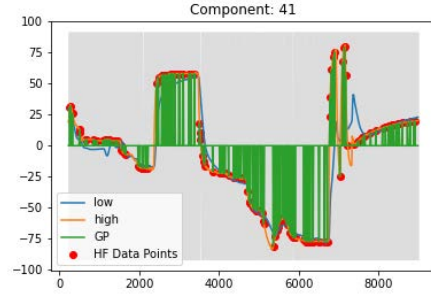


GP - Component 40

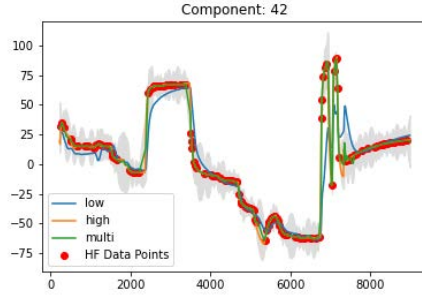
Figure A.34: SDS-4 data. Multifidelity vs Gaussian process. Random high fidelity data points. Blue lines (-) indicate the LFM, orange lines (-) the HFM, green lines (-) the multifidelity estimation and red dots (•) the HFM data points used and the gray area (■) the confidence interval for components 37-40.



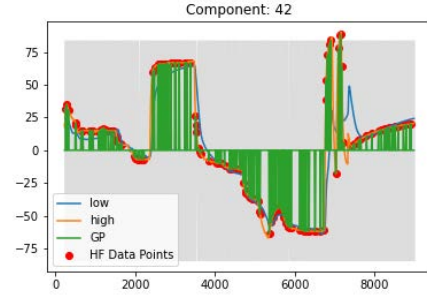
MF - Component 41



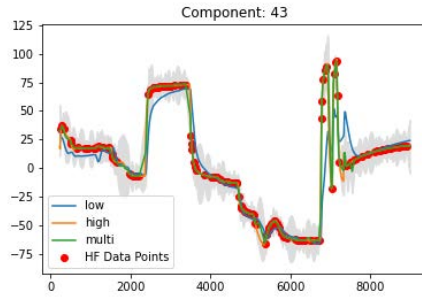
GP - Component 41



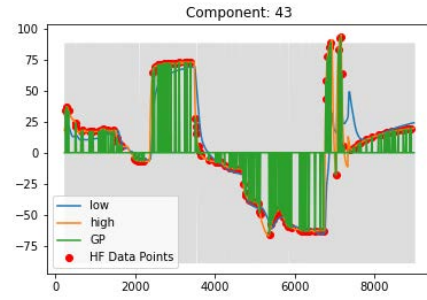
MF - Component 42



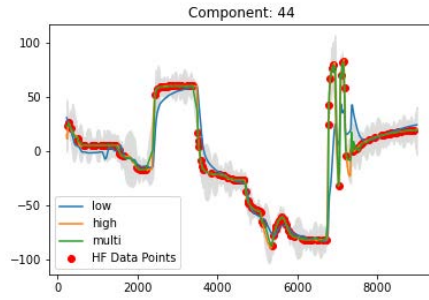
GP - Component 42



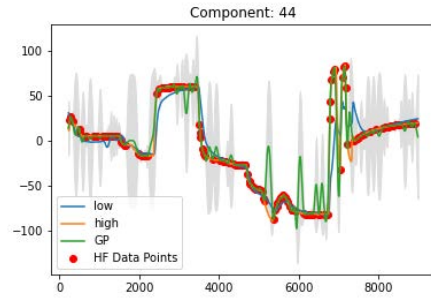
MF - Component 43



GP - Component 43

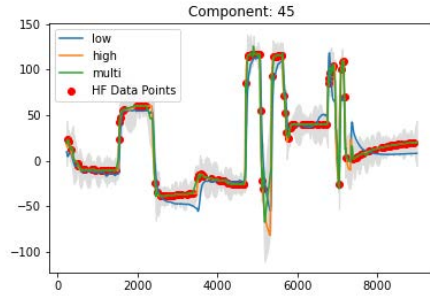


MF - Component 44

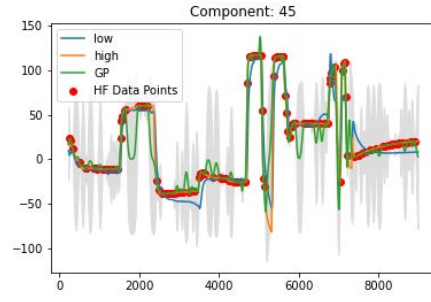


GP - Component 44

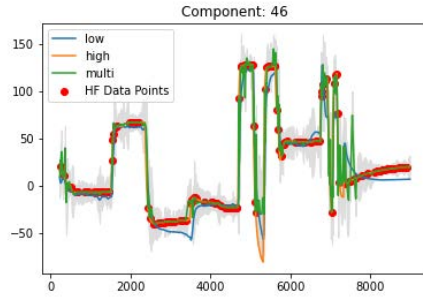
Figure A.35: SDS-4 data. Multifidelity vs Gaussian process. Random high fidelity data points. Blue lines (-) indicate the LFM, orange lines (-) the HFM, green lines (-) the multifidelity estimation and red dots (•) the HFM data points used and the gray area (■) the confidence interval for components 41-44.



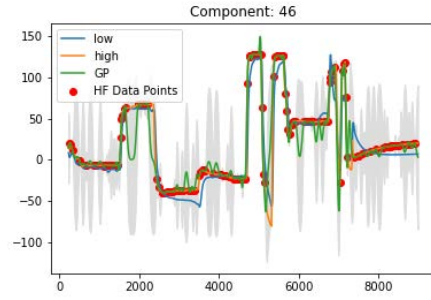
MF - Component 45



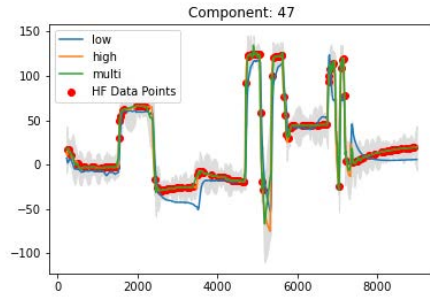
GP - Component 45



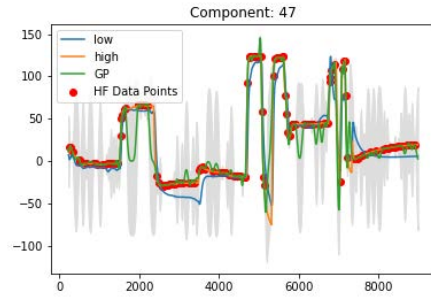
MF - Component 46



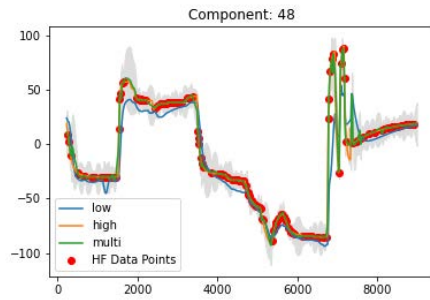
GP - Component 46



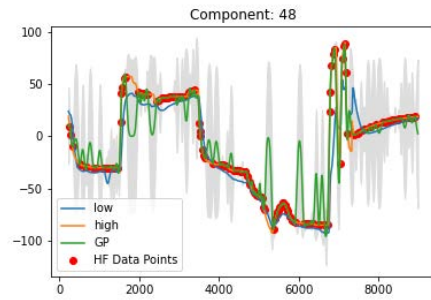
MF - Component 47



GP - Component 47

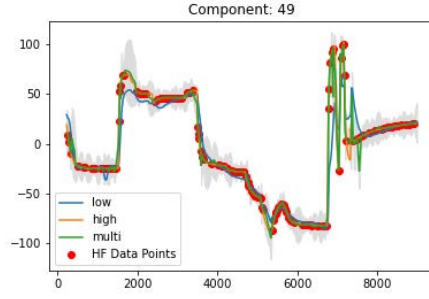


MF - Component 48

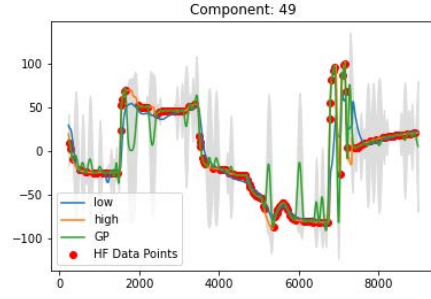


GP - Component 48

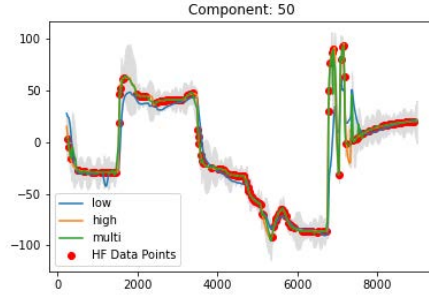
Figure A.36: SDS-4 data. Multifidelity vs Gaussian process. Random high fidelity data points. Blue lines (-) indicate the LFM, orange lines (-) the HFM, green lines (-) the multifidelity estimation and red dots (•) the HFM data points used and the gray area (■) the confidence interval for components 45-48.



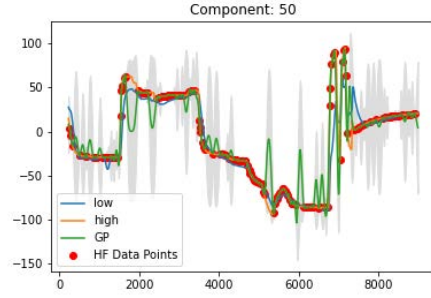
MF - Component 49



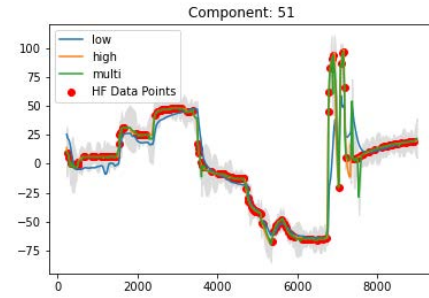
GP - Component 49



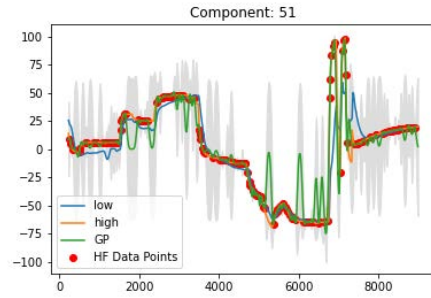
MF - Component 50



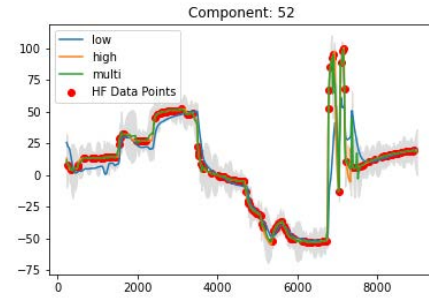
GP - Component 50



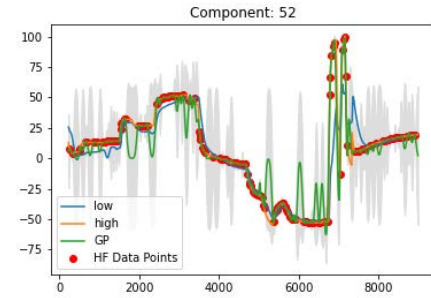
MF - Component 51



GP - Component 51

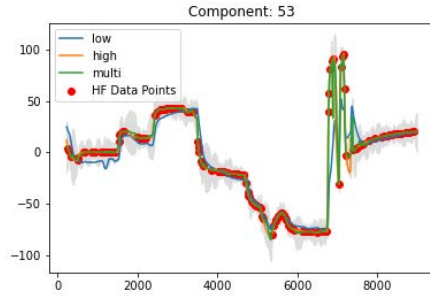


MF - Component 52

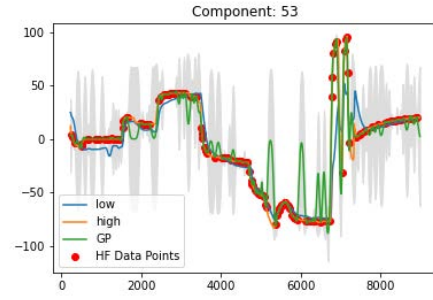


GP - Component 52

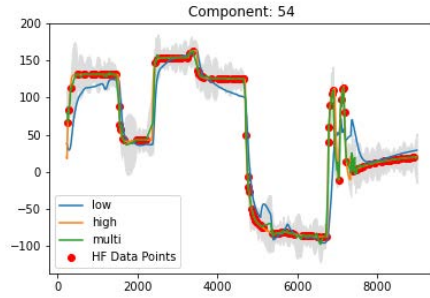
Figure A.37: SDS-4 data. Multifidelity vs Gaussian process. Random high fidelity data points. Blue lines (-) indicate the LFM, orange lines (-) the HFM, green lines (-) the multifidelity estimation and red dots (•) the HFM data points used and the gray area (■) the confidence interval for components 49-52.



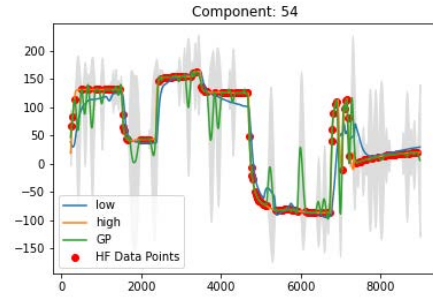
MF - Component 53



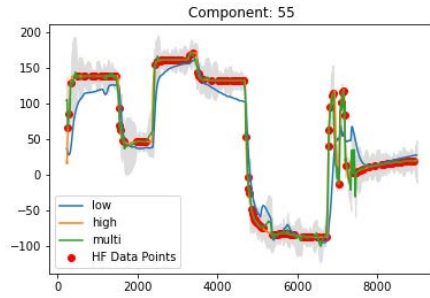
GP - Component 53



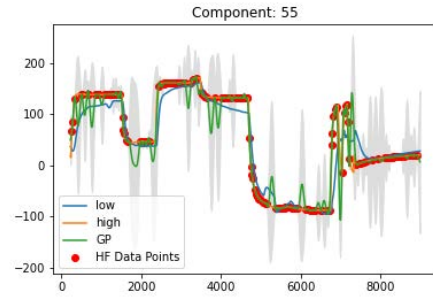
MF - Component 54



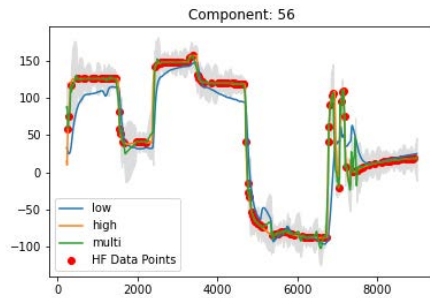
GP - Component 54



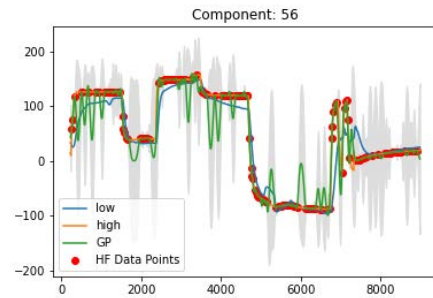
MF - Component 55



GP - Component 55

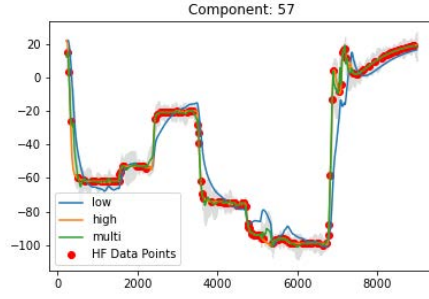


MF - Component 56

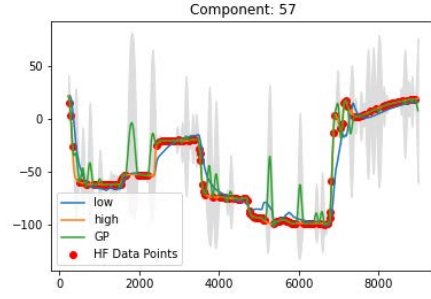


GP - Component 56

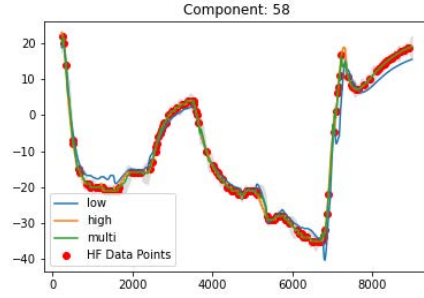
Figure A.38: SDS-4 data. Multifidelity vs Gaussian process. Random high fidelity data points. Blue lines (-) indicate the LFM, orange lines (-) the HFM, green lines (-) the multifidelity estimation and red dots (•) the HFM data points used and the gray area (■) the confidence interval for components 53-56.



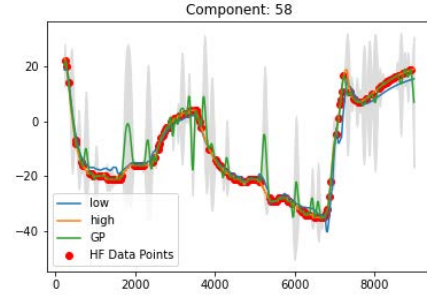
MF - Component 57



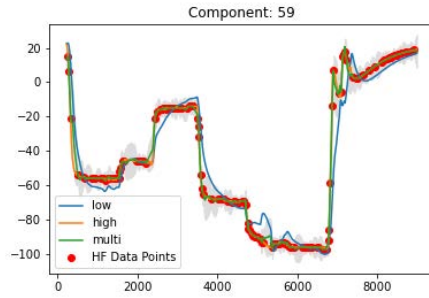
GP - Component 57



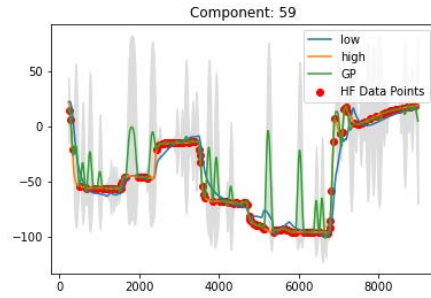
MF - Component 58



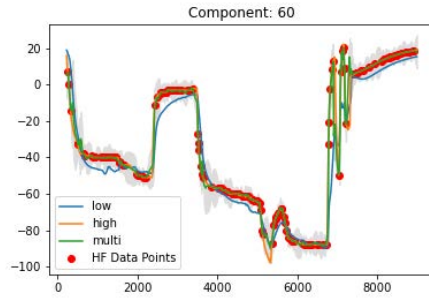
GP - Component 58



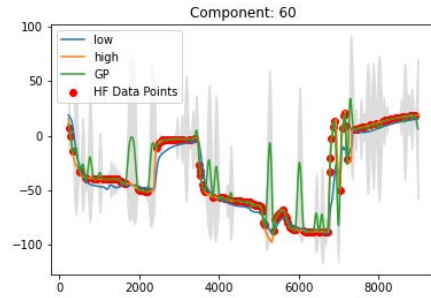
MF - Component 59



GP - Component 59



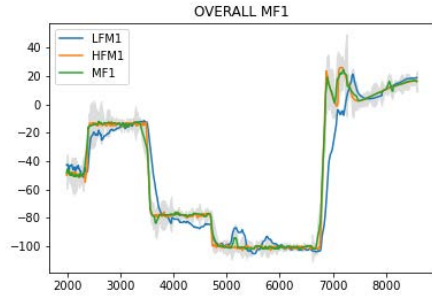
MF - Component 60



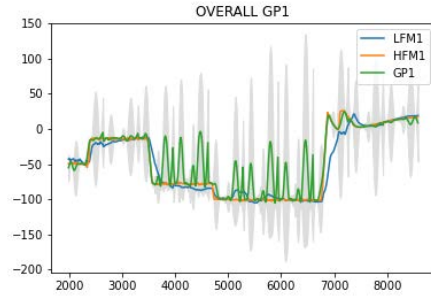
GP - Component 60

Figure A.39: SDS-4 data. Multifidelity vs Gaussian process. Random high fidelity data points. Blue lines (-) indicate the LFM, orange lines (-) the HFM, green lines (-) the multifidelity estimation and red dots (•) the HFM data points used and the gray area (■) the confidence interval for components 57-60.

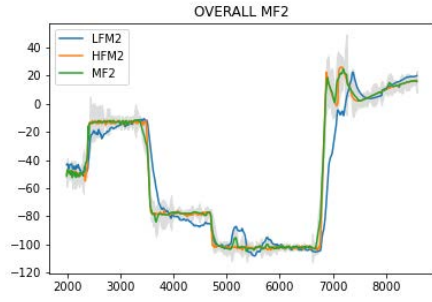
A.5 SDS-4 Overall With Windows



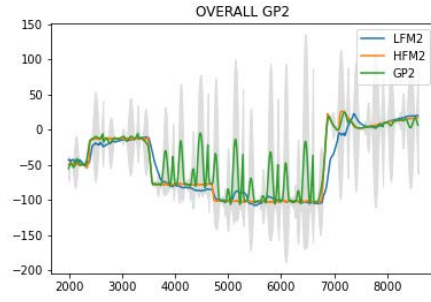
MF - Component 1



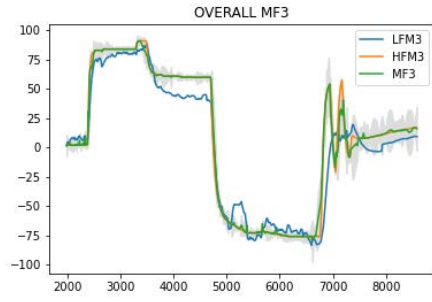
GP - Component 1



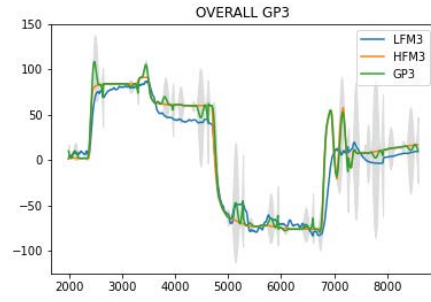
MF - Component 2



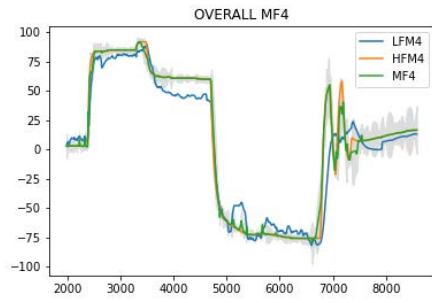
GP - Component 2



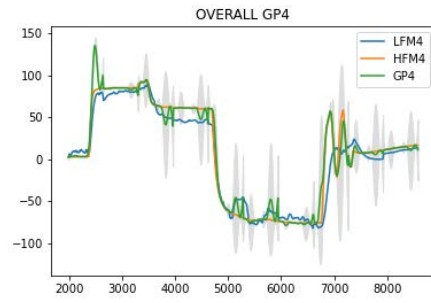
MF - Component 3



GP - Component 3

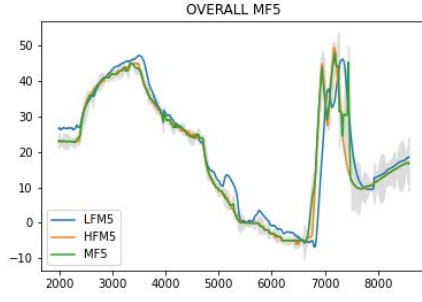


MF - Component 4

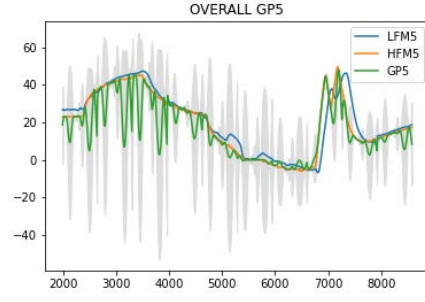


GP - Component 4

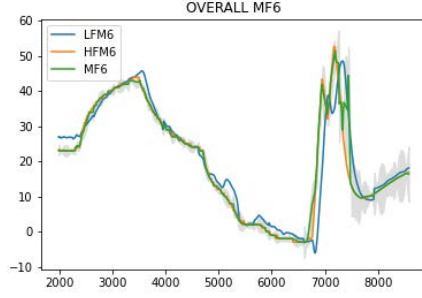
Figure A.40: SDS-4 data. Multifidelity vs Gaussian process. Random high fidelity data points using the sliding window. Blue lines (-) indicate the LFM, orange lines (-) the HFM, green lines (-) the multifidelity estimation and red dots (•) the HFM data points used and the gray area (■) the confidence interval for components 1-4.



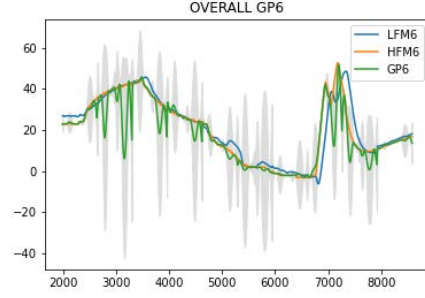
MF - Component 5



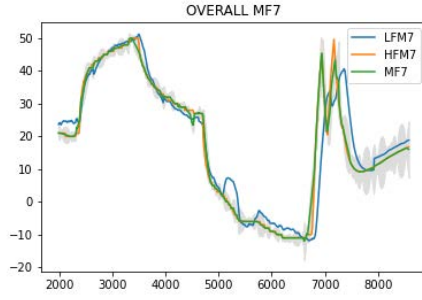
GP - Component 5



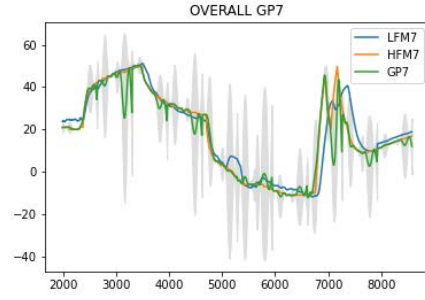
MF - Component 6



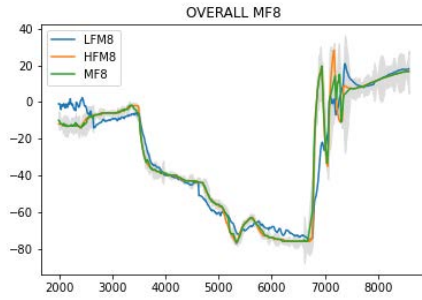
GP - Component 6



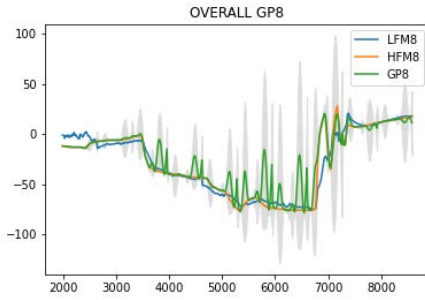
MF - Component 7



GP - Component 7

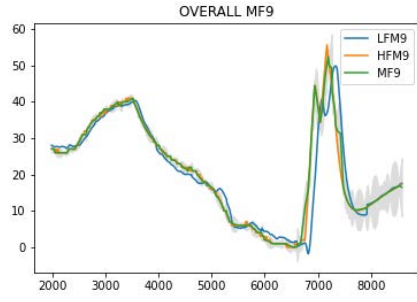


MF - Component 8

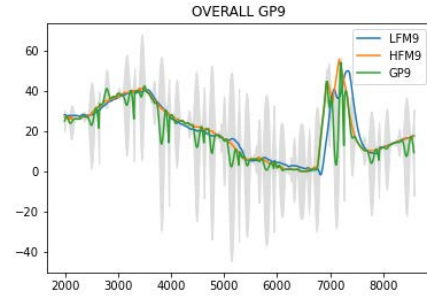


GP - Component 8

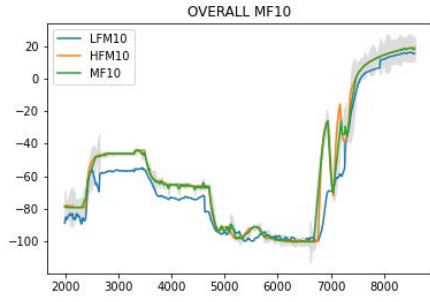
Figure A.41: SDS-4 data. Multifidelity vs Gaussian process. Random high fidelity data points using the sliding window. Blue lines (-) indicate the LFM, orange lines (-) the HFM, green lines (-) the multifidelity estimation and red dots (•) the HFM data points used and the gray area (■) the confidence interval for components 5-8.



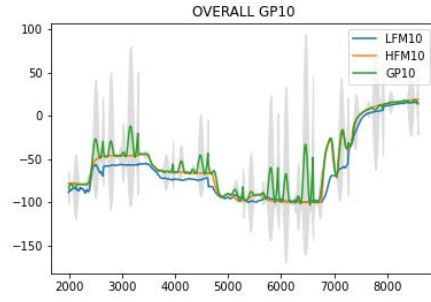
MF - Component 9



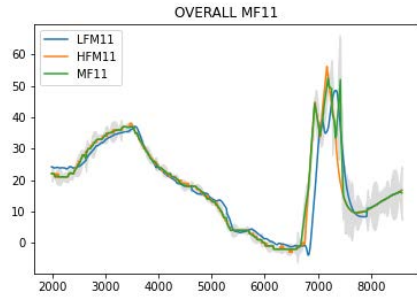
GP - Component 9



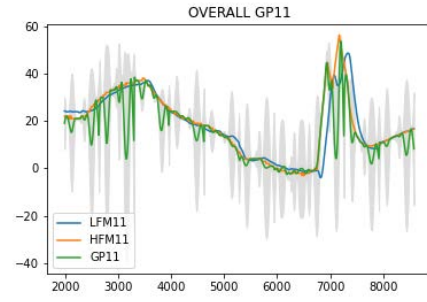
MF - Component 10



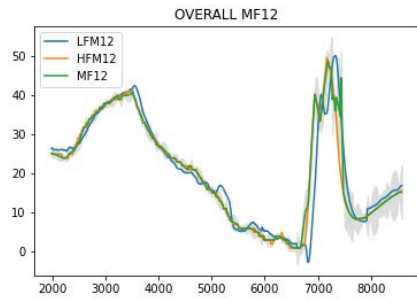
GP - Component 10



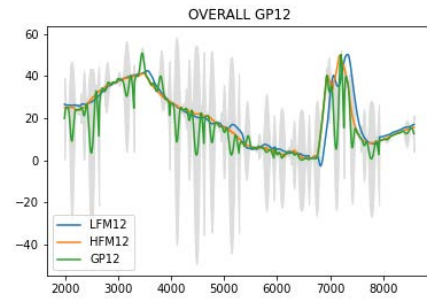
MF - Component 11



GP - Component 11

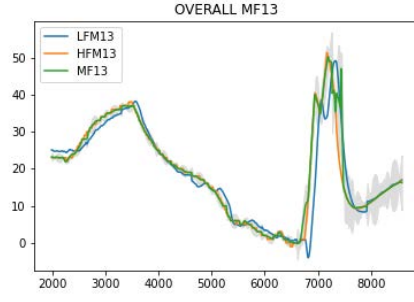


MF - Component 12

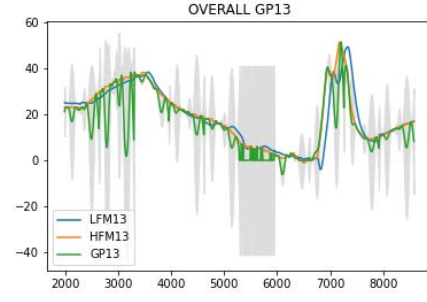


GP - Component 12

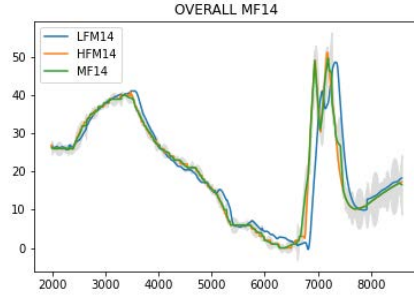
Figure A.42: SDS-4 data. Multifidelity vs Gaussian process. Random high fidelity data points using the sliding window. Blue lines (-) indicate the LFM, orange lines (-) the HFM, green lines (-) the multifidelity estimation and red dots (•) the HFM data points used and the gray area (■) the confidence interval for components 9-12.



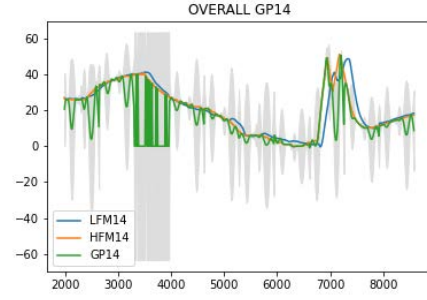
MF - Component 13



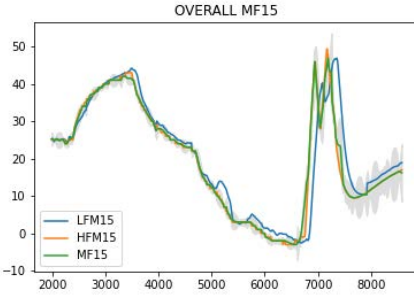
GP - Component 13



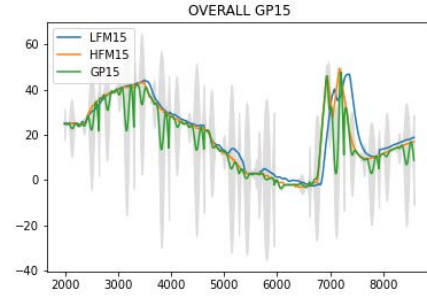
MF - Component 14



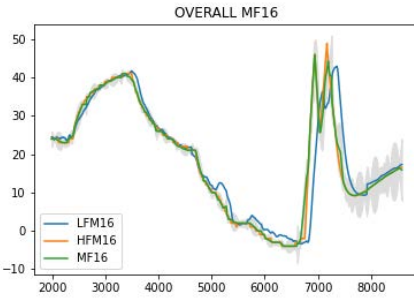
GP - Component 14



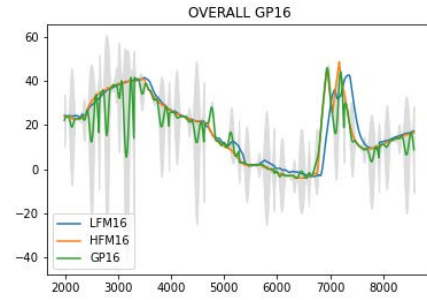
MF - Component 15



GP - Component 15

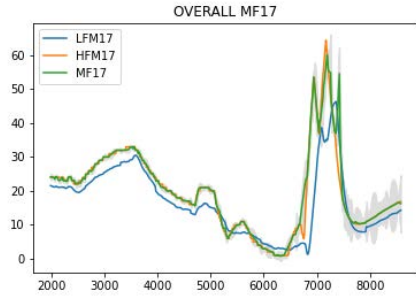


MF - Component 16

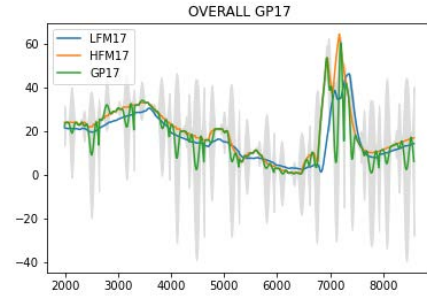


GP - Component 16

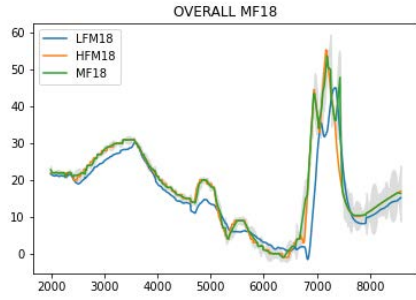
Figure A.43: SDS-4 data. Multifidelity vs Gaussian process. Random high fidelity data points using the sliding window. Blue lines (-) indicate the LFM, orange lines (-) the HFM, green lines (-) the multifidelity estimation and red dots (•) the HFM data points used and the gray area (■) the confidence interval for components 13-16.



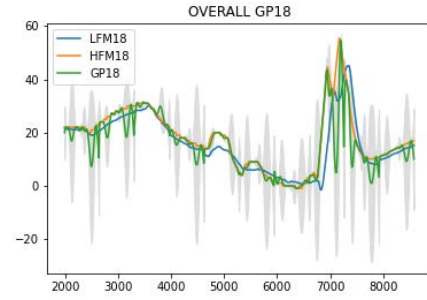
MF - Component 17



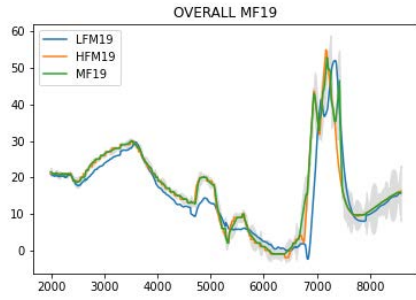
GP - Component 17



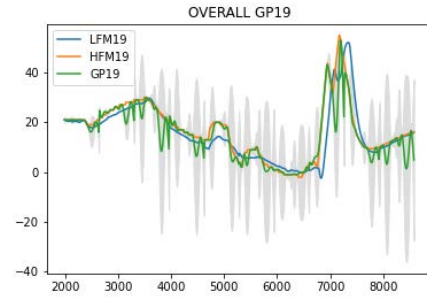
MF - Component 18



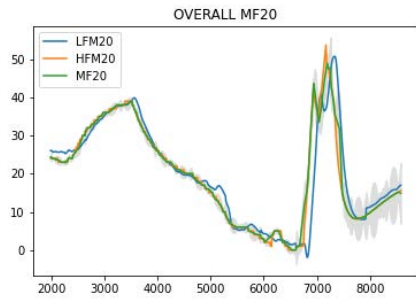
GP - Component 18



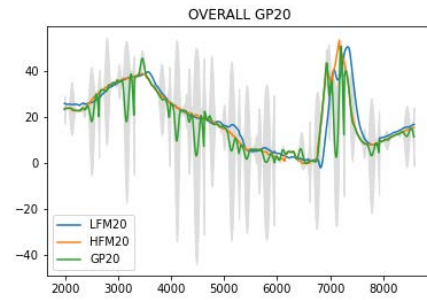
MF - Component 19



GP - Component 19

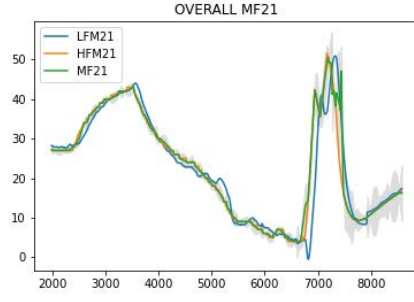


MF - Component 20

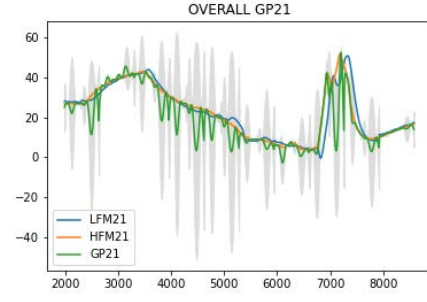


GP - Component 20

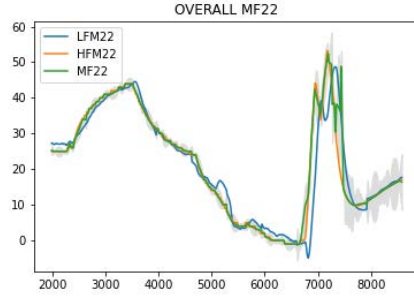
Figure A.44: SDS-4 data. Multifidelity vs Gaussian process. Random high fidelity data points using the sliding window. Blue lines (-) indicate the LFM, orange lines (-) the HFM, green lines (-) the multifidelity estimation and red dots (•) the HFM data points used and the gray area (■) the confidence interval for components 17-20.



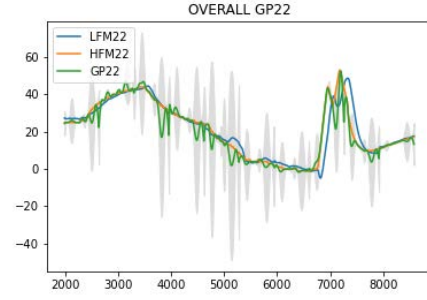
MF - Component 21



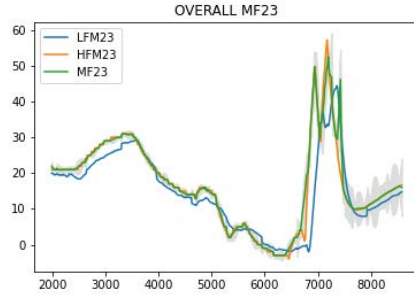
GP - Component 21



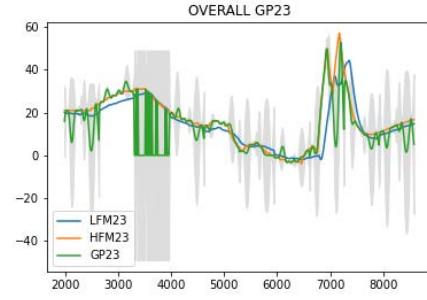
MF - Component 22



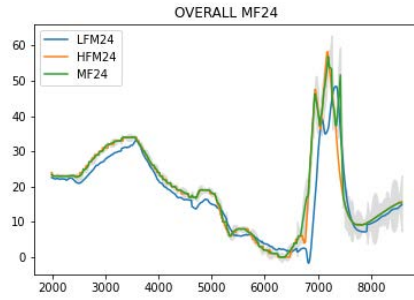
GP - Component 22



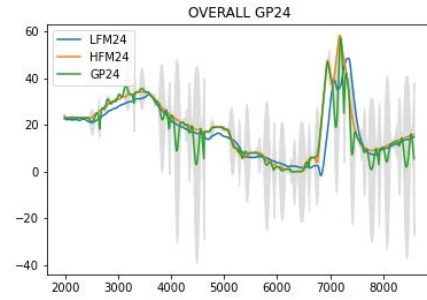
MF - Component 23



GP - Component 23

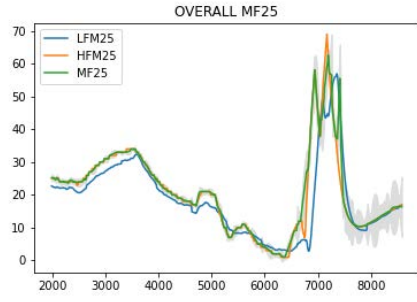


MF - Component 24

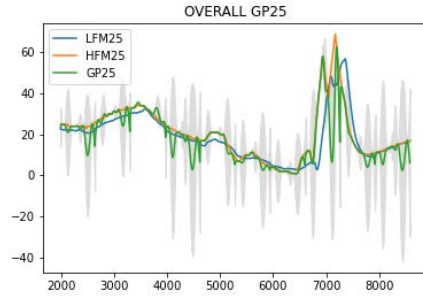


GP - Component 24

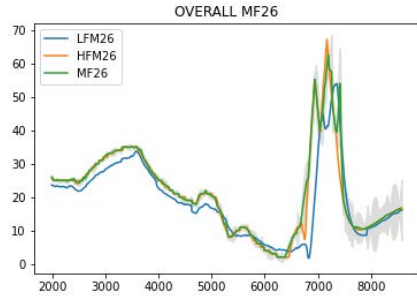
Figure A.45: SDS-4 data. Multifidelity vs Gaussian process. Random high fidelity data points using the sliding window. Blue lines (-) indicate the LFM, orange lines (-) the HFM, green lines (-) the multifidelity estimation and red dots (•) the HFM data points used and the gray area (■) the confidence interval for components 21-24.



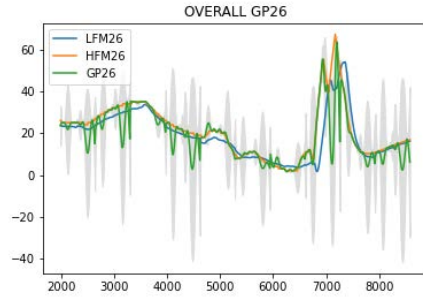
MF - Component 25



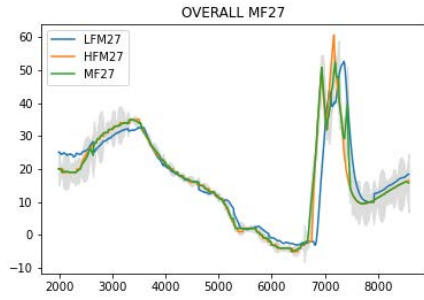
GP - Component 25



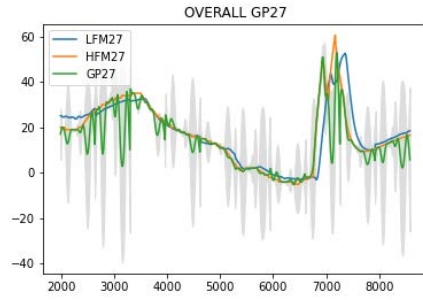
MF - Component 26



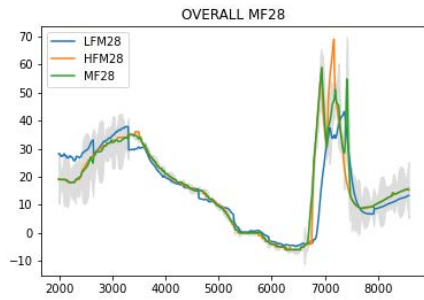
GP - Component 26



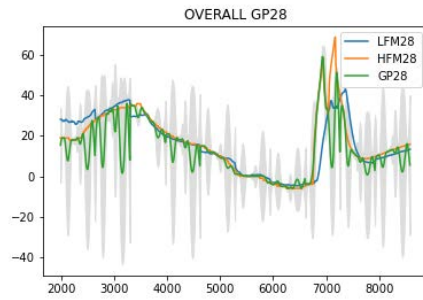
MF - Component 27



GP - Component 27

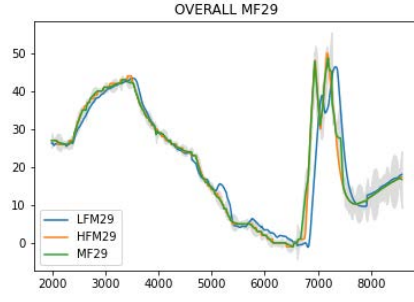


MF - Component 28

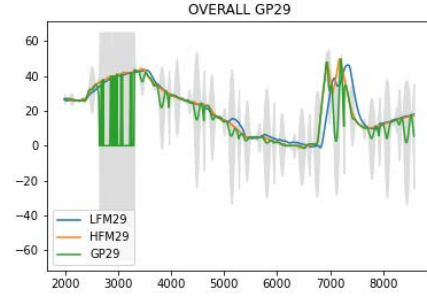


GP - Component 28

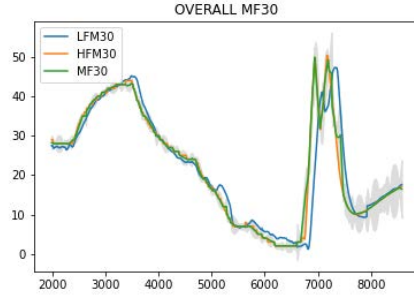
Figure A.46: SDS-4 data. Multifidelity vs Gaussian process. Random high fidelity data points using the sliding window. Blue lines (-) indicate the LFM, orange lines (-) the HFM, green lines (-) the multifidelity estimation and red dots (•) the HFM data points used and the gray area (■) the confidence interval for components 25-28.



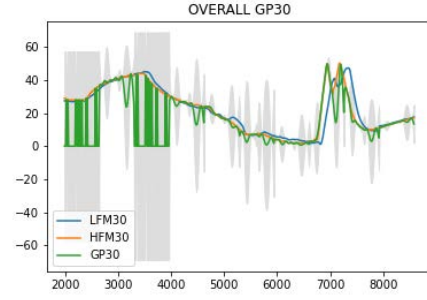
MF - Component 29



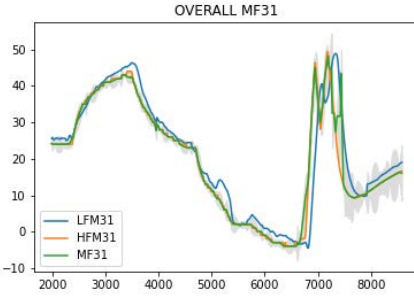
GP - Component 29



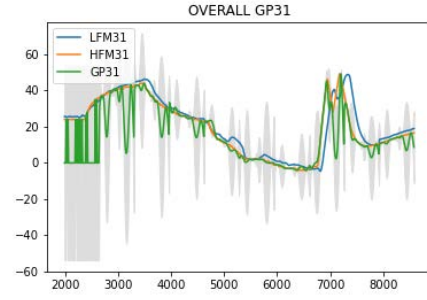
MF - Component 30



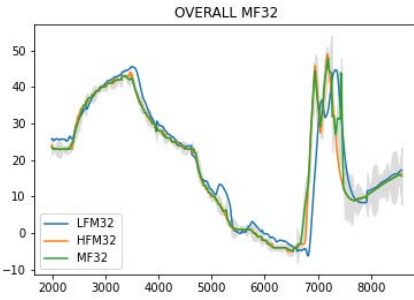
GP - Component 30



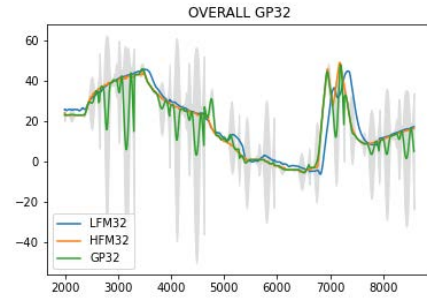
MF - Component 31



GP - Component 31

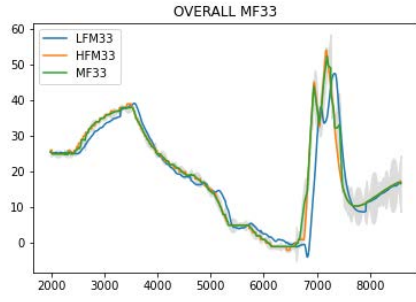


MF - Component 32

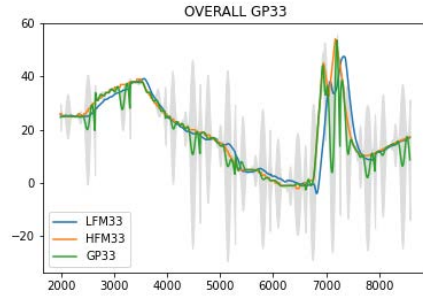


GP - Component 32

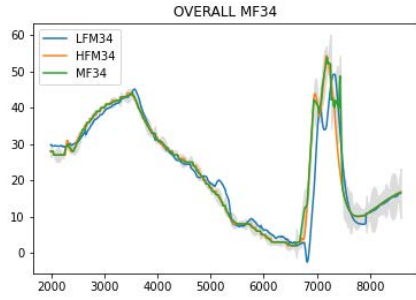
Figure A.47: SDS-4 data. Multifidelity vs Gaussian process. Random high fidelity data points using the sliding window. Blue lines (-) indicate the LFM, orange lines (-) the HFM, green lines (-) the multifidelity estimation and red dots (•) the HFM data points used and the gray area (■) the confidence interval for components 29-32.



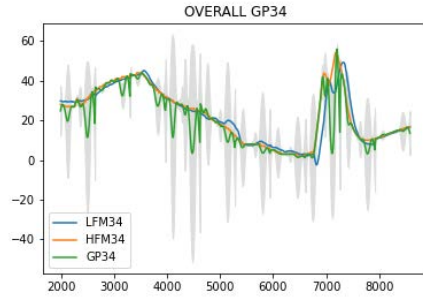
MF - Component 33



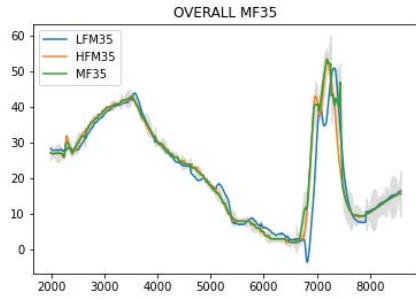
GP - Component 33



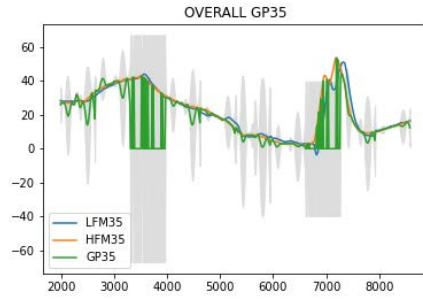
MF - Component 34



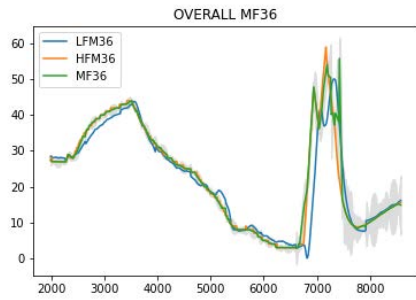
GP - Component 34



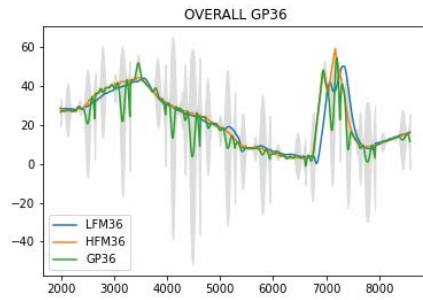
MF - Component 35



GP - Component 35

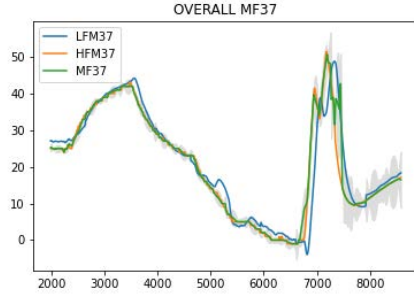


MF - Component 36

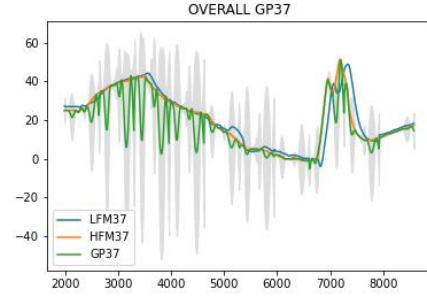


GP - Component 36

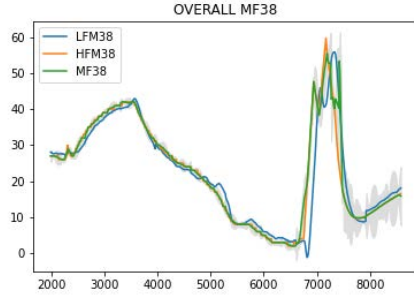
Figure A.48: SDS-4 data. Multifidelity vs Gaussian process. Random high fidelity data points using the sliding window. Blue lines (-) indicate the LFM, orange lines (-) the HFM, green lines (-) the multifidelity estimation and red dots (•) the HFM data points used and the gray area (■) the confidence interval for components 33-36.



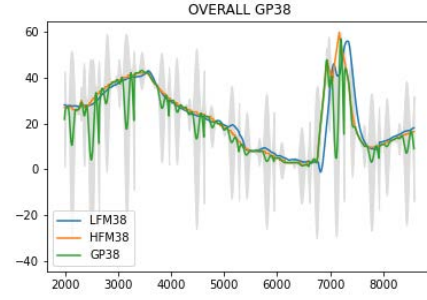
MF - Component 37



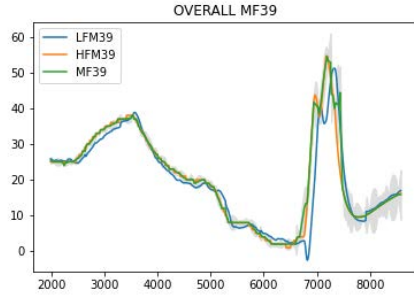
GP - Component 37



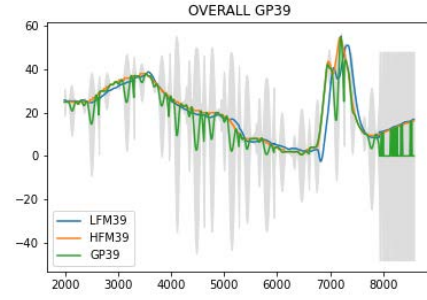
MF - Component 38



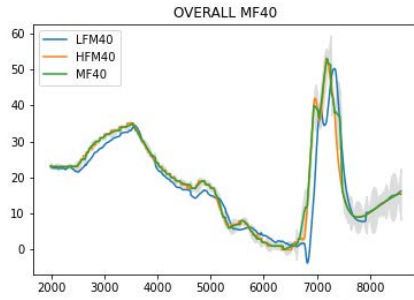
GP - Component 38



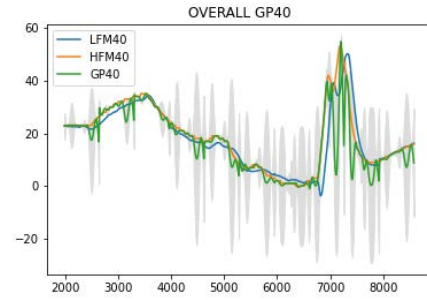
MF - Component 39



GP - Component 39

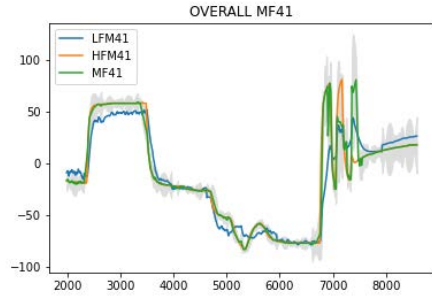


MF - Component 40

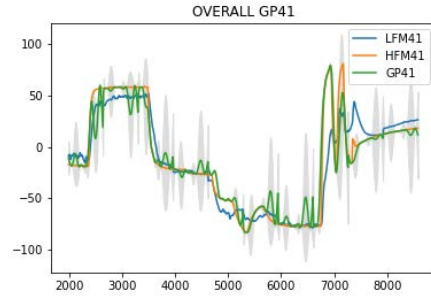


GP - Component 40

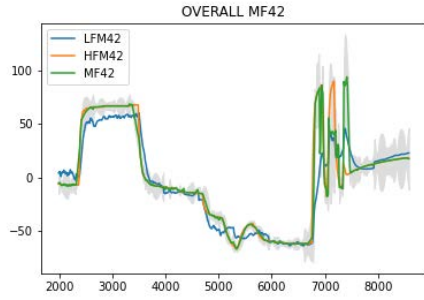
Figure A.49: SDS-4 data. Multifidelity vs Gaussian process. Random high fidelity data points using the sliding window. Blue lines (-) indicate the LFM, orange lines (-) the HFM, green lines (-) the multifidelity estimation and red dots (•) the HFM data points used and the gray area (■) the confidence interval for components 37-40.



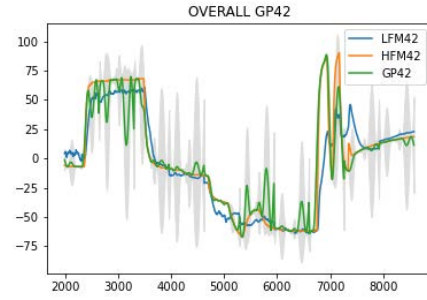
MF - Component 41



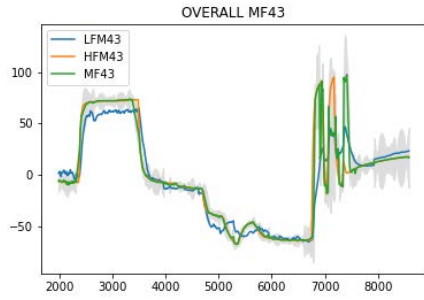
GP - Component 41



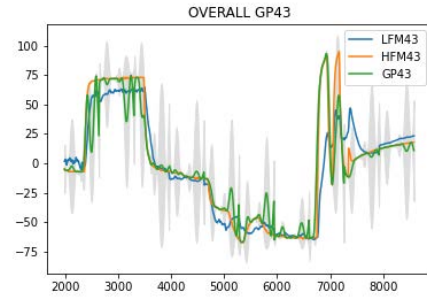
MF - Component 42



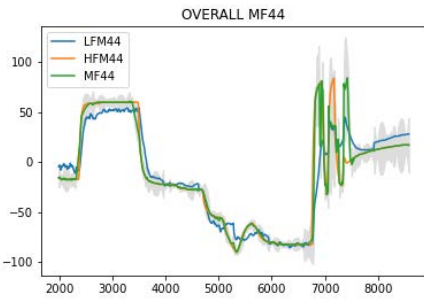
GP - Component 42



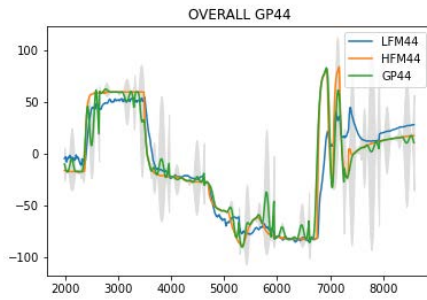
MF - Component 43



GP - Component 43

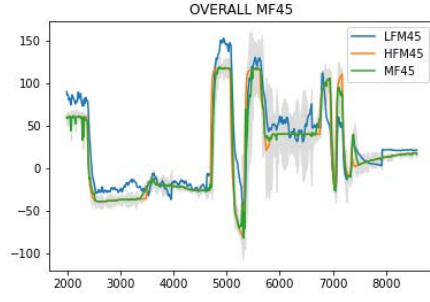


MF - Component 44

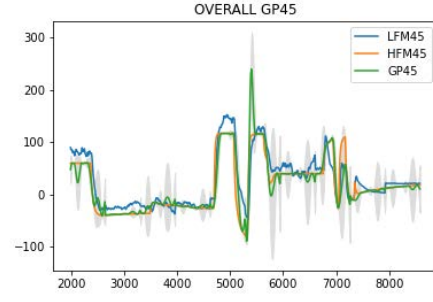


GP - Component 44

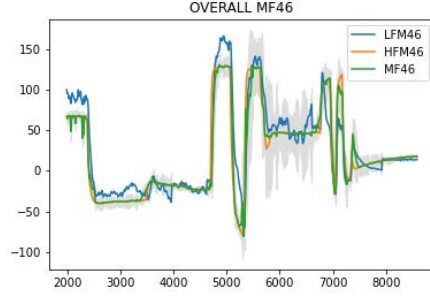
Figure A.50: SDS-4 data. Multifidelity vs Gaussian process. Random high fidelity data points using the sliding window. Blue lines (-) indicate the LFM, orange lines (-) the HFM, green lines (-) the multifidelity estimation and red dots (•) the HFM data points used and the gray area (■) the confidence interval for components 41-44.



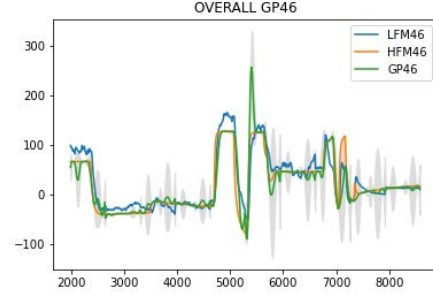
MF - Component 45



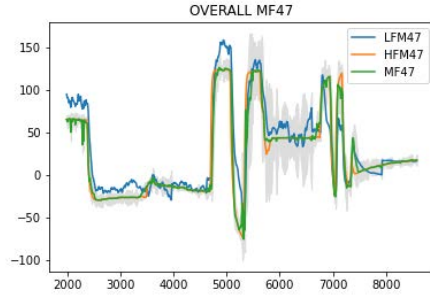
GP - Component 45



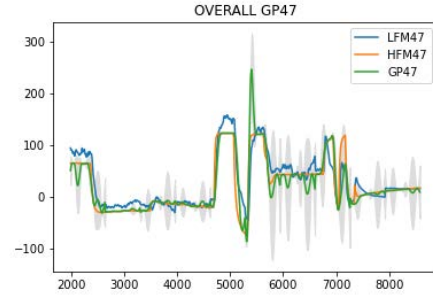
MF - Component 46



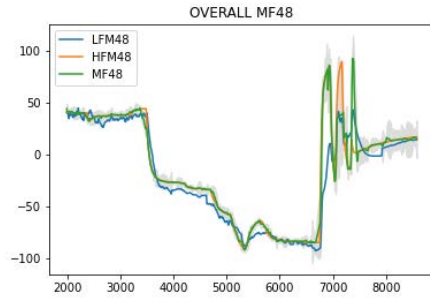
GP - Component 46



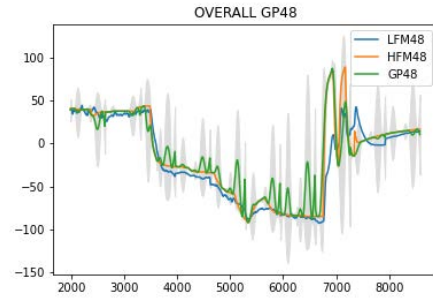
MF - Component 47



GP - Component 47

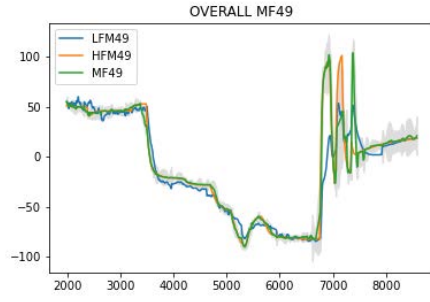


MF - Component 48

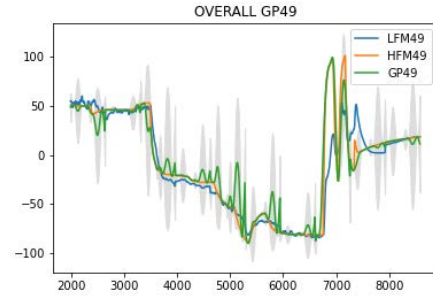


GP - Component 48

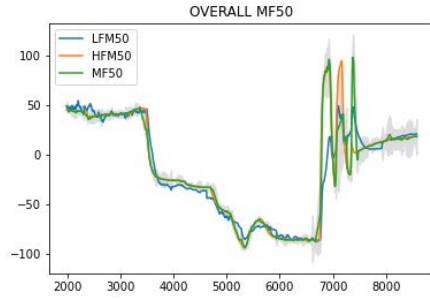
Figure A.51: SDS-4 data. Multifidelity vs Gaussian process. Random high fidelity data points using the sliding window. Blue lines (-) indicate the LFM, orange lines (-) the HFM, green lines (-) the multifidelity estimation and red dots (•) the HFM data points used and the gray area (■) the confidence interval for components 45-48.



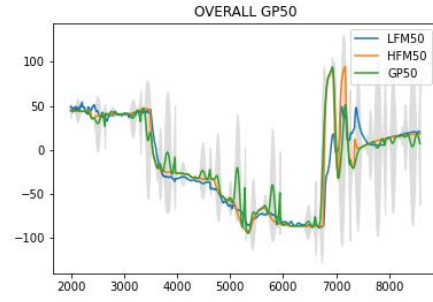
MF - Component 49



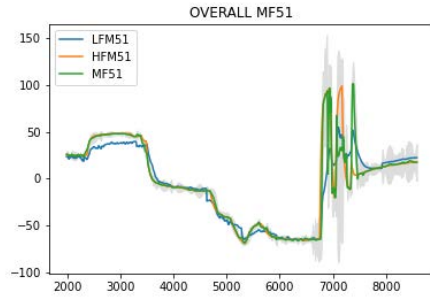
GP - Component 49



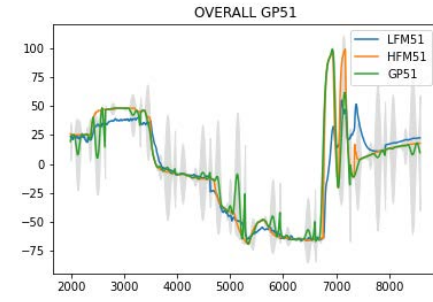
MF - Component 50



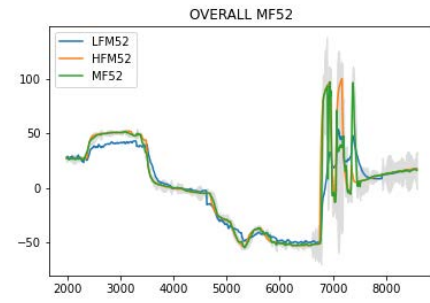
GP - Component 50



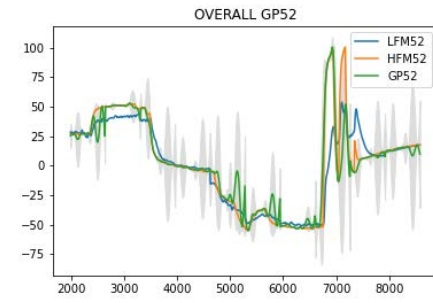
MF - Component 51



GP - Component 51

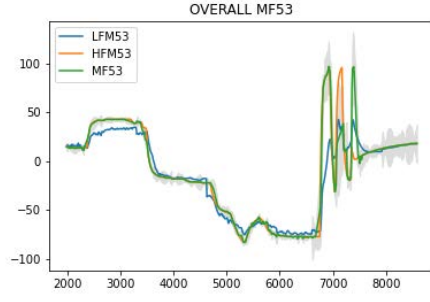


MF - Component 52

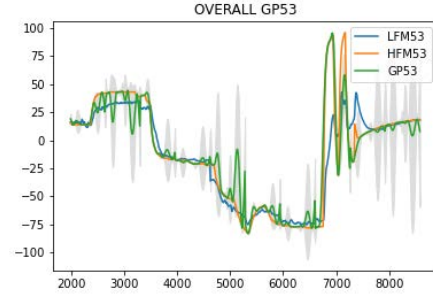


GP - Component 52

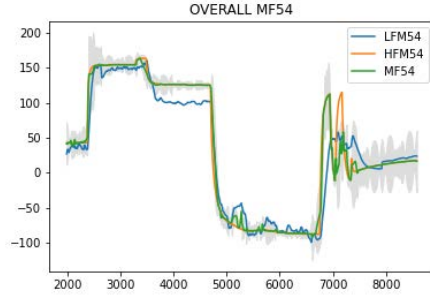
Figure A.52: SDS-4 data. Multifidelity vs Gaussian process. Random high fidelity data points using the sliding window. Blue lines (-) indicate the LFM, orange lines (-) the HFM, green lines (-) the multifidelity estimation and red dots (•) the HFM data points used and the gray area (■) the confidence interval for components 49-52.



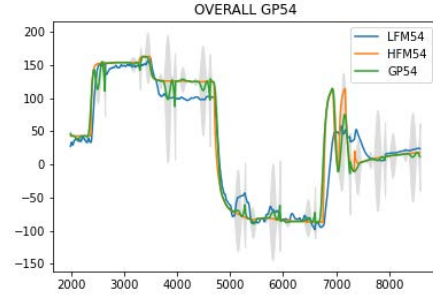
MF - Component 53



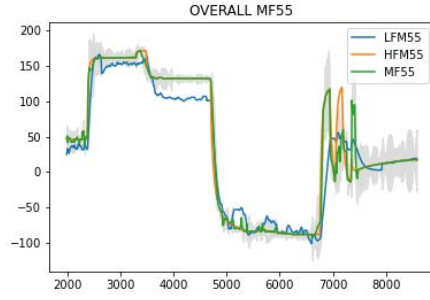
GP - Component 53



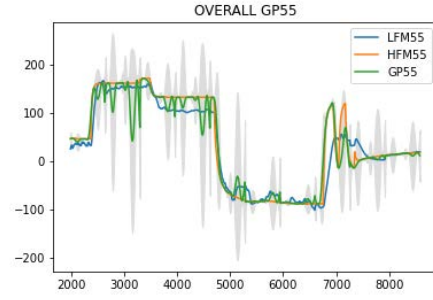
MF - Component 54



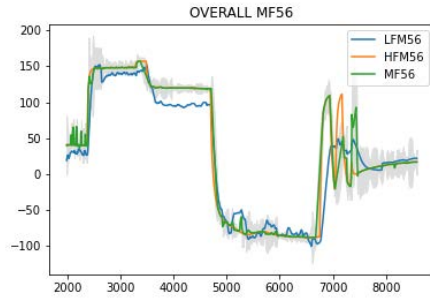
GP - Component 54



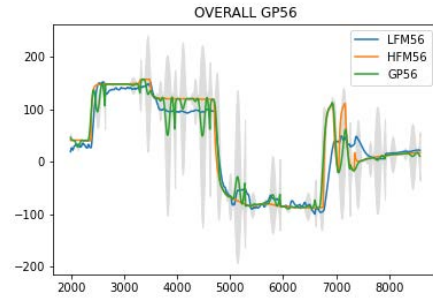
MF - Component 55



GP - Component 55

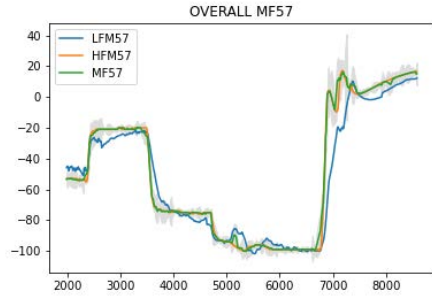


MF - Component 56

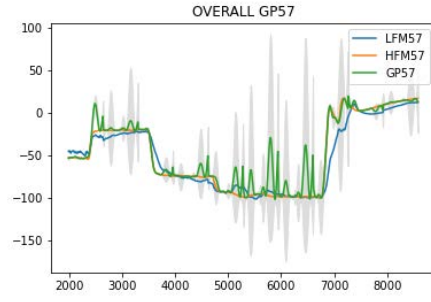


GP - Component 56

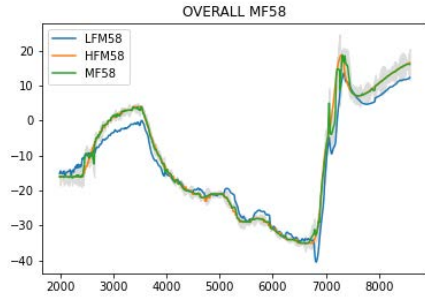
Figure A.53: SDS-4 data. Multifidelity vs Gaussian process. Random high fidelity data points using the sliding window. Blue lines (-) indicate the LFM, orange lines (-) the HFM, green lines (-) the multifidelity estimation and red dots (•) the HFM data points used and the gray area (■) the confidence interval for components 53-56.



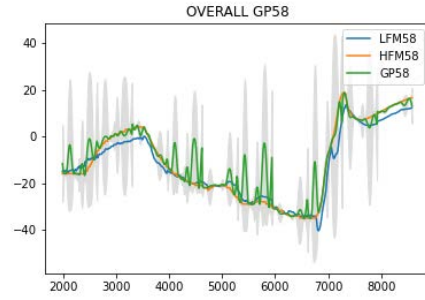
MF - Component 57



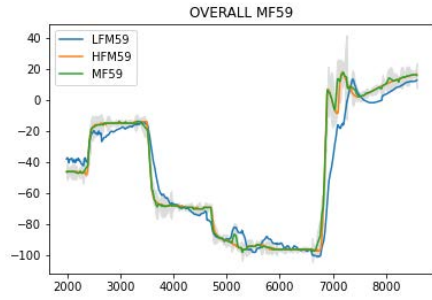
GP - Component 57



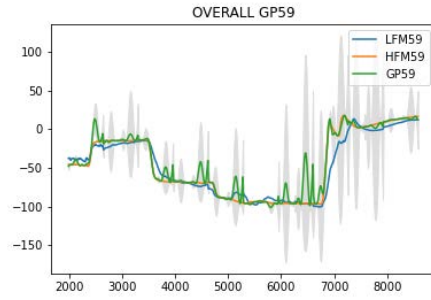
MF - Component 58



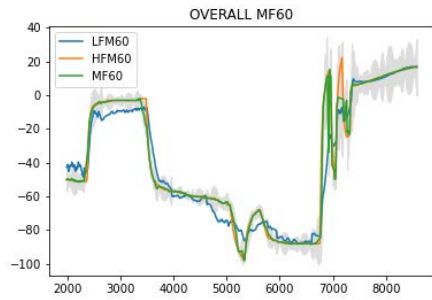
GP - Component 58



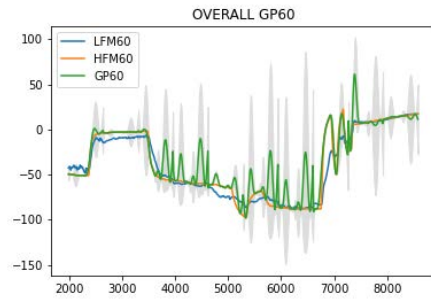
MF - Component 59



GP - Component 59



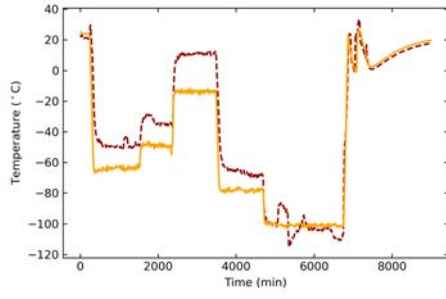
MF - Component 60



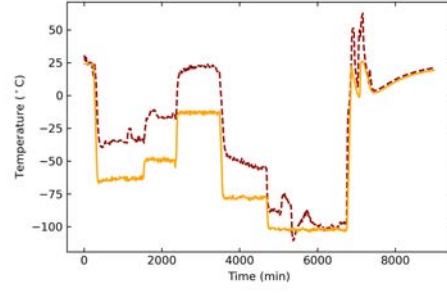
GP - Component 60

Figure A.54: SDS-4 data. Multifidelity vs Gaussian process. Random high fidelity data points using the sliding window. Blue lines (-) indicate the LFM, orange lines (-) the HFM, green lines (-) the multifidelity estimation and red dots (•) the HFM data points used and the gray area (■) the confidence interval for components 57-60.

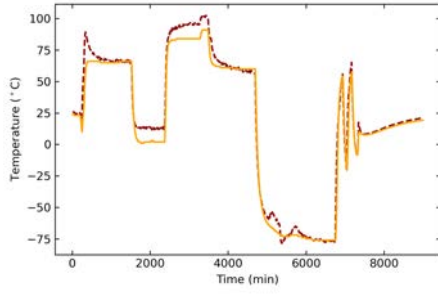
A.6 ARIMAX SDS-4



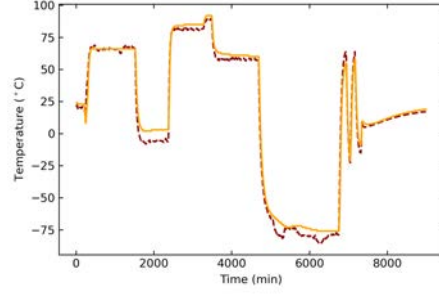
Component 1



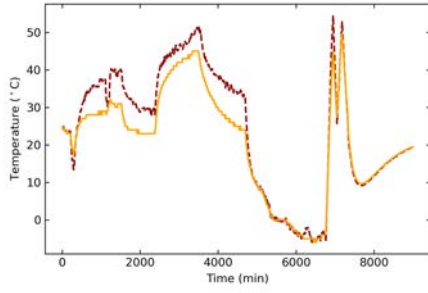
Component 2



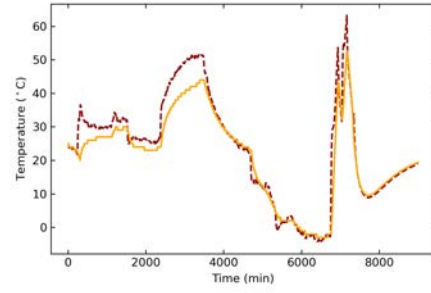
Component 3



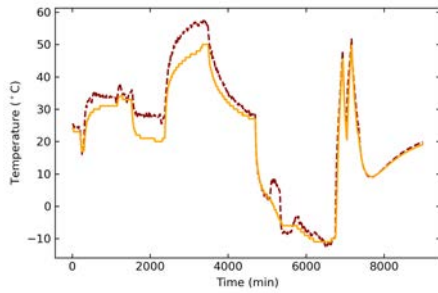
Component 4



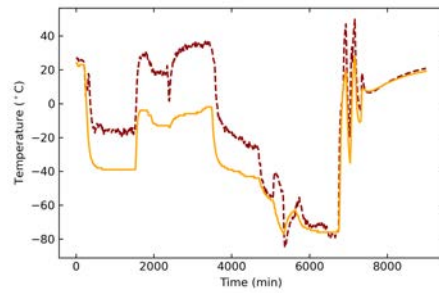
Component 5



Component 6

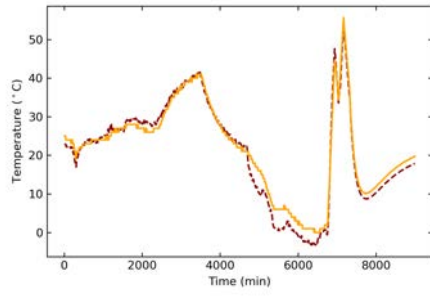


Component 7

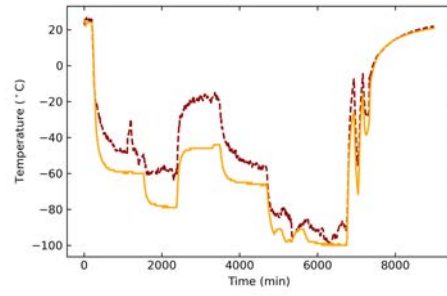


Component 8

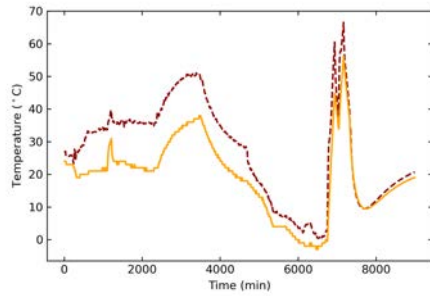
Figure A.55: ARIMAX SDS-4data. Blue lines (—) indicate the LFM, orange lines (---) the ARIMAX prediction for components 1-8.



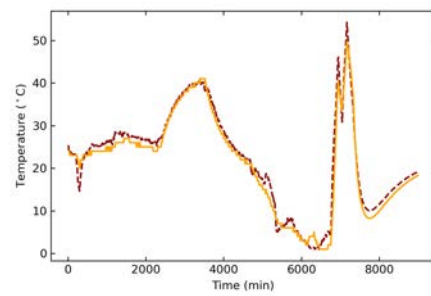
Component 9



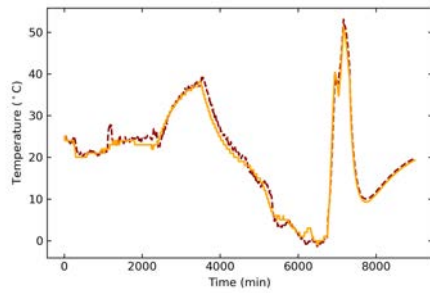
Component 10



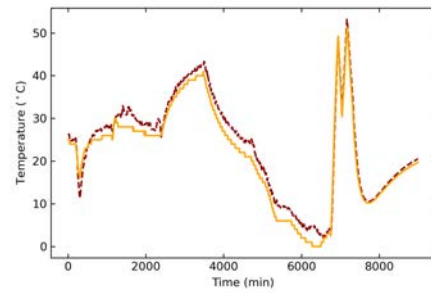
Component 11



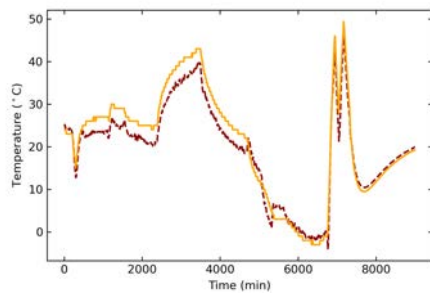
Component 12



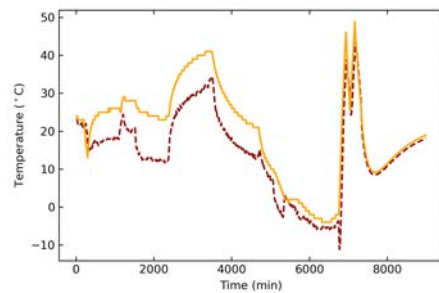
Component 13



Component 14

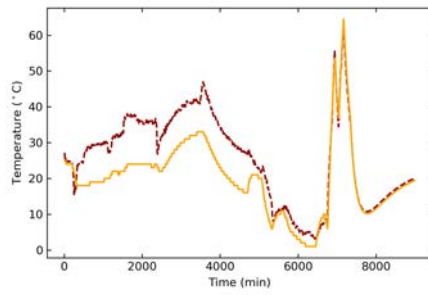


Component 15

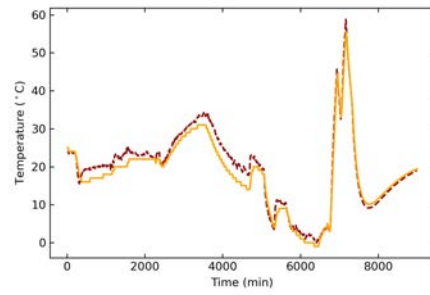


Component 16

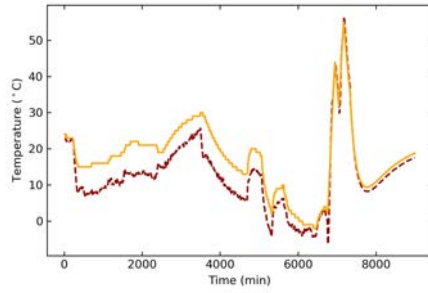
Figure A.56: ARIMAX SDS-4data. Blue lines (-) indicate the LFM, orange lines (-) the ARIMAX prediction for components 9-16.



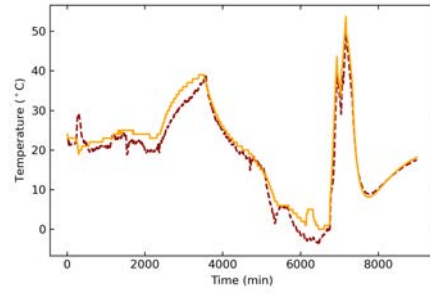
Component 17



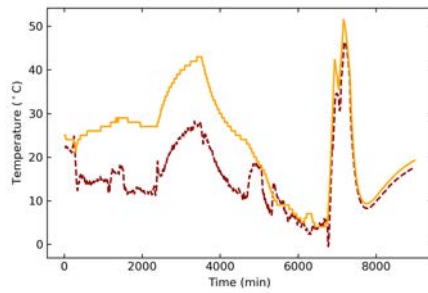
Component 18



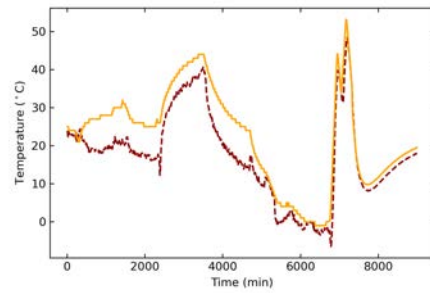
Component 19



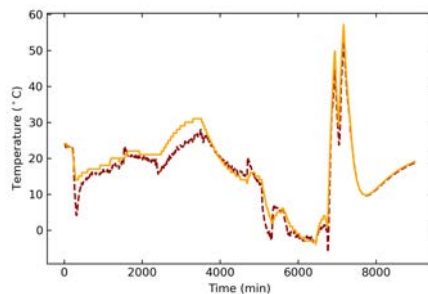
Component 20



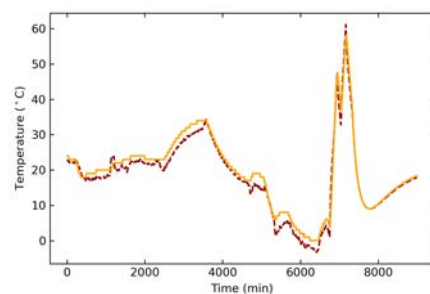
Component 21



Component 22

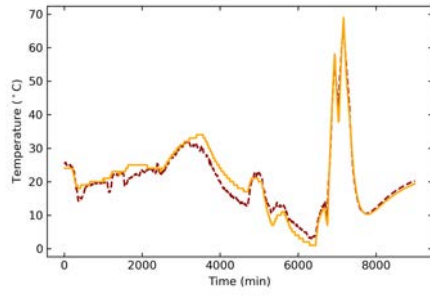


Component 23

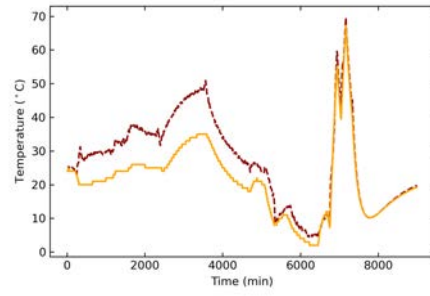


Component 24

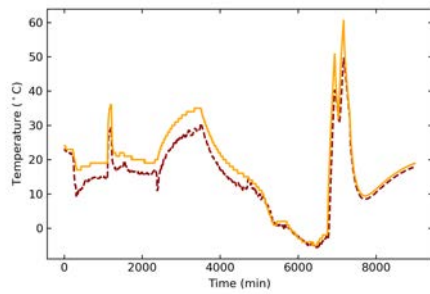
Figure A.57: ARIMAX SDS-4data. Blue lines (—) indicate the LFM, orange lines (---) the ARIMAX prediction for components 17-24.



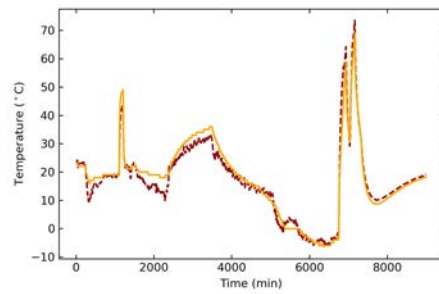
Component 25



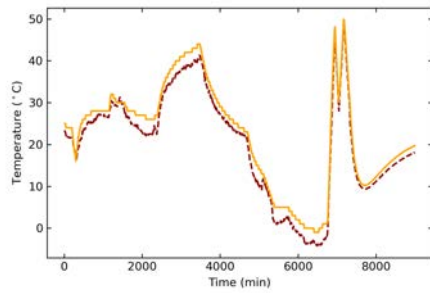
Component 26



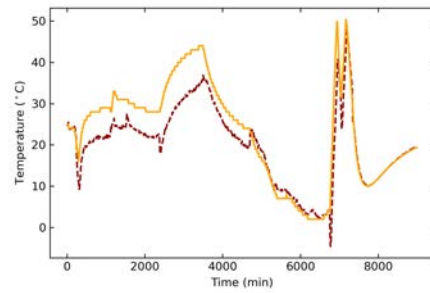
Component 27



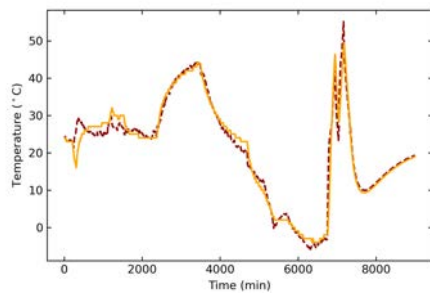
Component 28



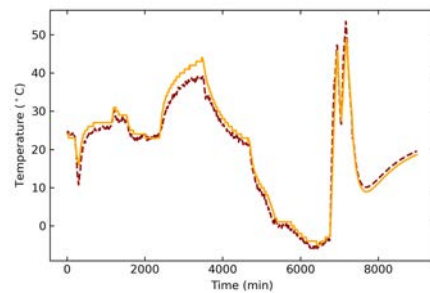
Component 29



Component 30

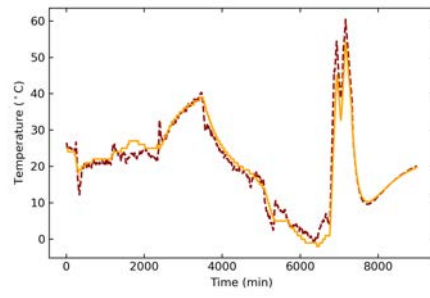


Component 31

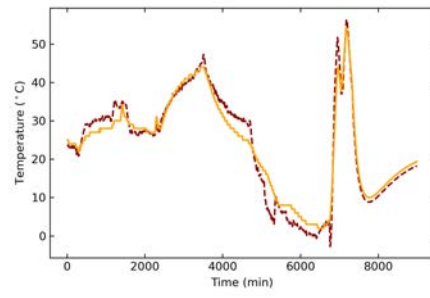


Component 32

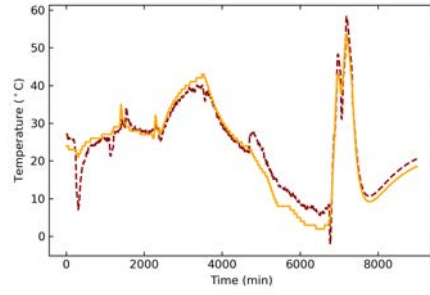
Figure A.58: ARIMAX SDS-4data. Blue lines (—) indicate the LFM, orange lines (---) the ARIMAX prediction for components 25-32.



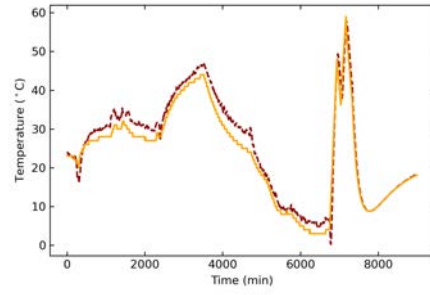
Component 33



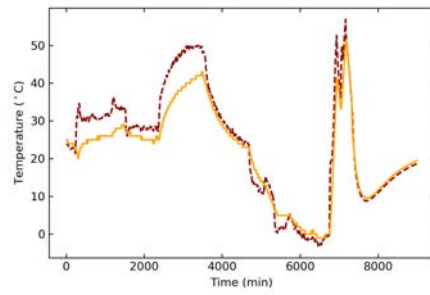
Component 34



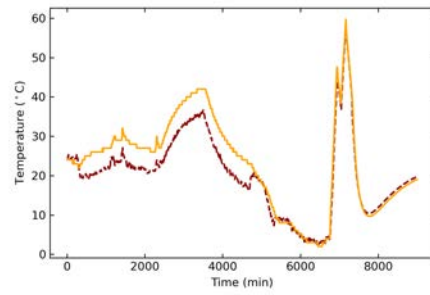
Component 35



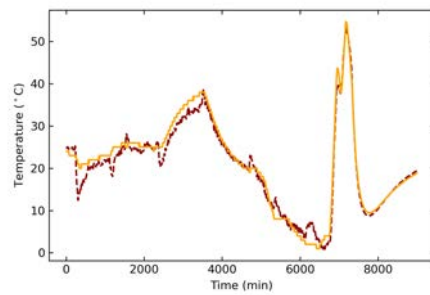
Component 36



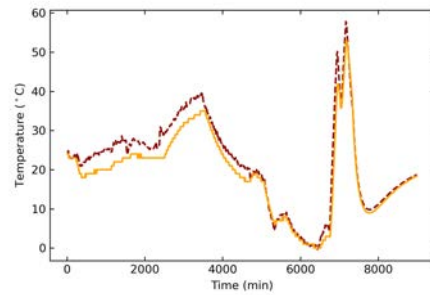
Component 37



Component 38

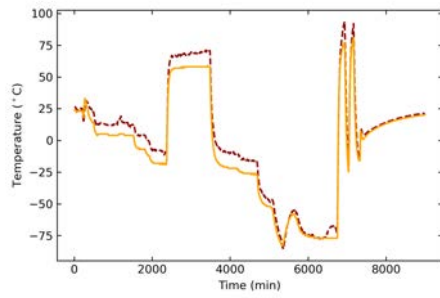


Component 39

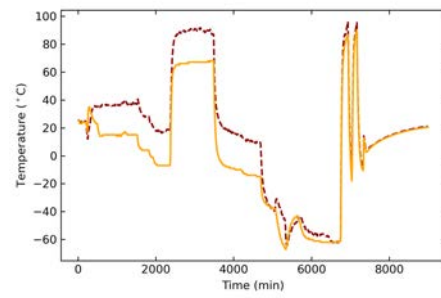


Component 40

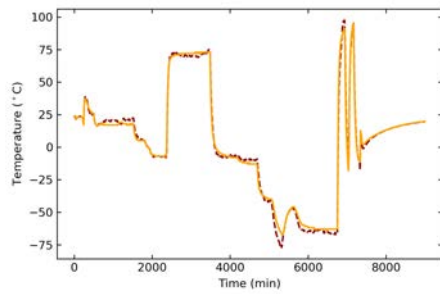
Figure A.59: ARIMAX SDS-4data. Blue lines (—) indicate the LFM, orange lines (---) the ARIMAX prediction for components 33-40.



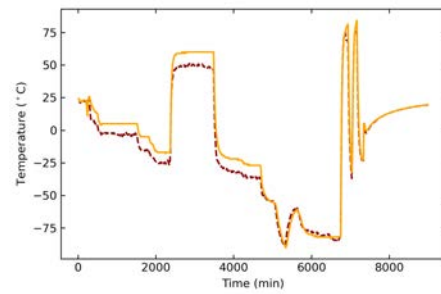
Component 41



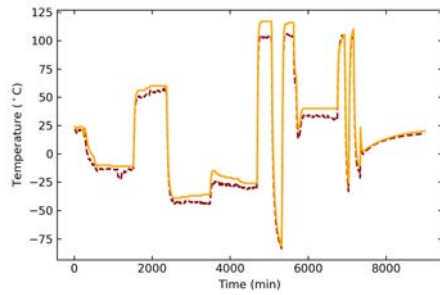
Component 42



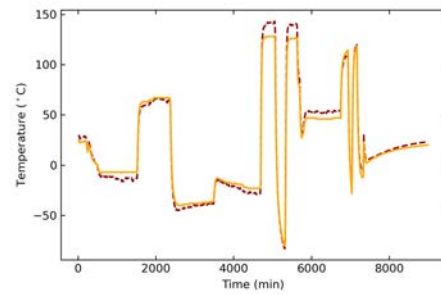
Component 43



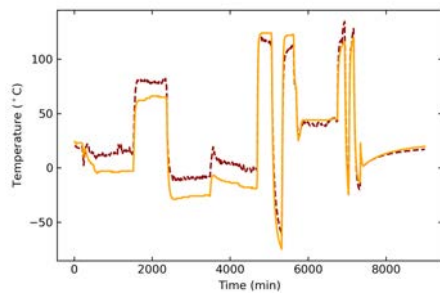
Component 44



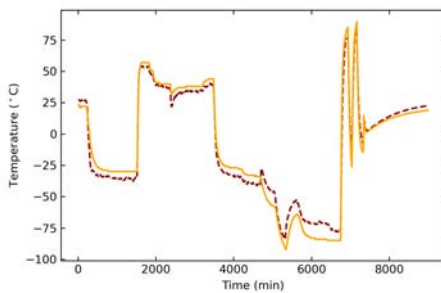
Component 45



Component 46

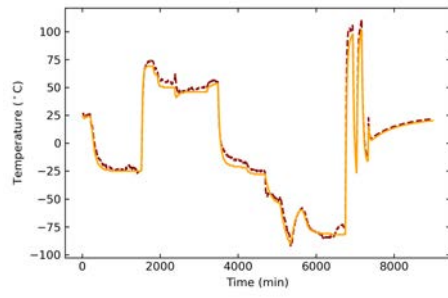


Component 47

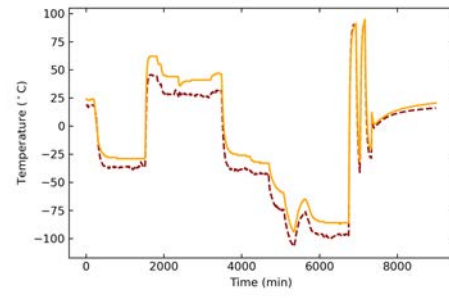


Component 48

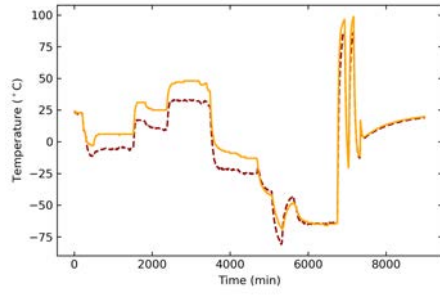
Figure A.60: ARIMAX SDS-4data. Blue lines (-) indicate the LFM, orange lines (-) the ARIMAX prediction for components 41-48.



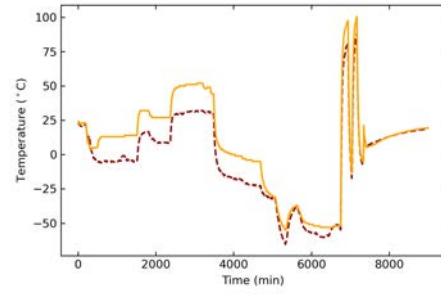
Component 49



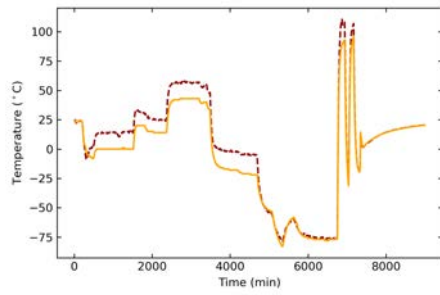
Component 50



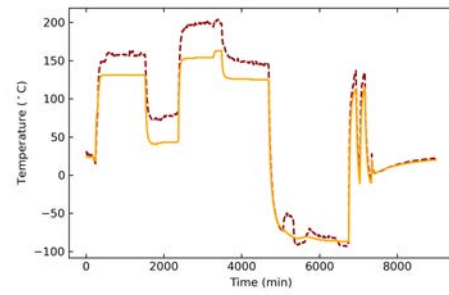
Component 51



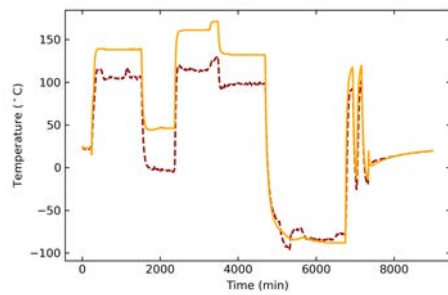
Component 52



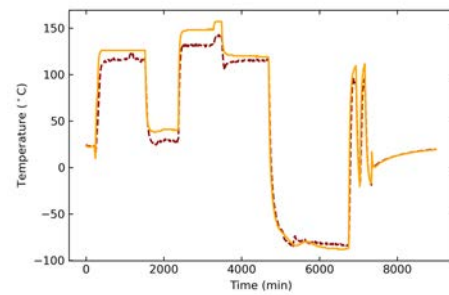
Component 53



Component 54

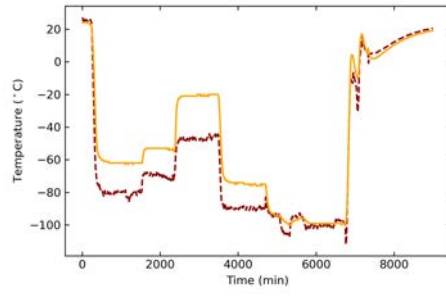


Component 55

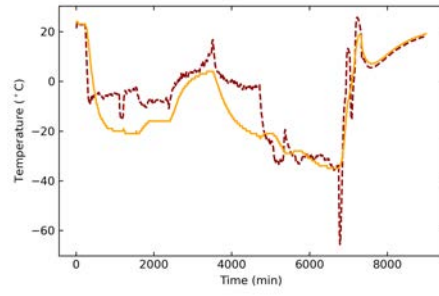


Component 56

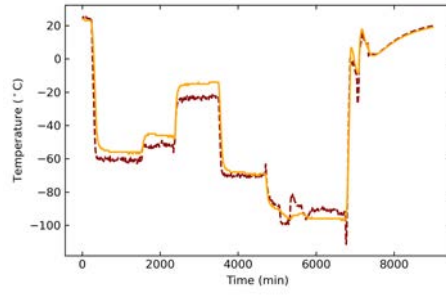
Figure A.61: ARIMAX SDS-4data. Blue lines (—) indicate the LFM, orange lines (---) the ARIMAX prediction for components 49-56.



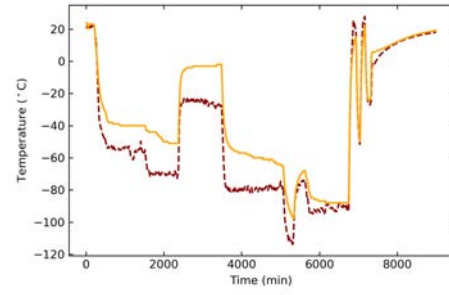
Component 57



Component 58



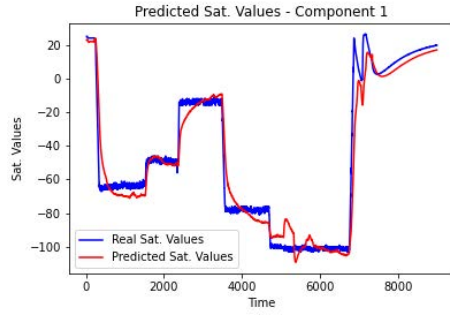
Component 59



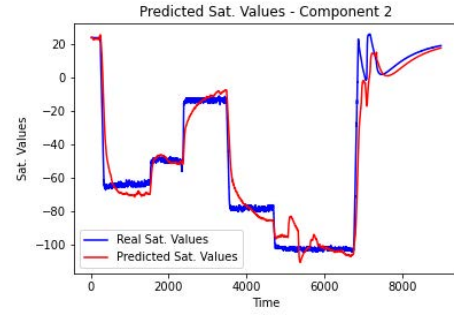
Component 60

Figure A.62: ARIMAX SDS-4 data. Blue lines (—) indicate the LFM, orange lines (---) the ARIMAX prediction for components 57-60.

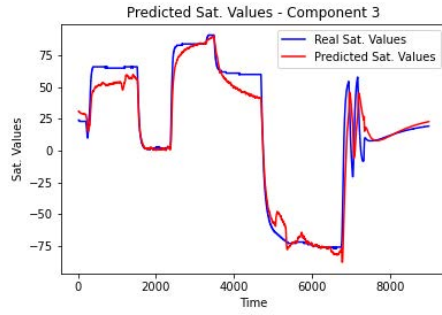
A.7 GRU SDS-4 HF Points Placement - Entire Dataset



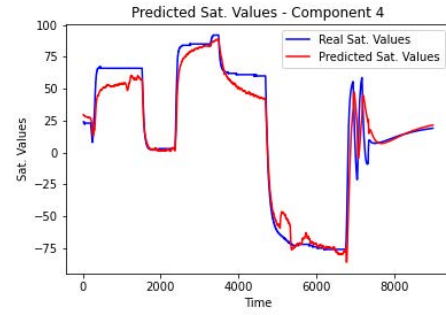
MF - Component 1



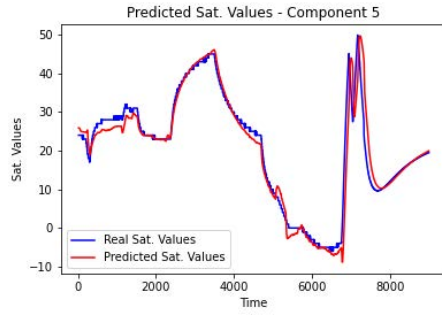
GP - Component 2



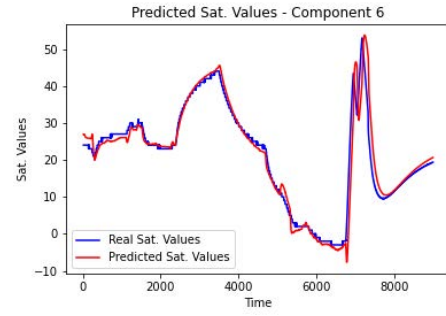
MF - Component 3



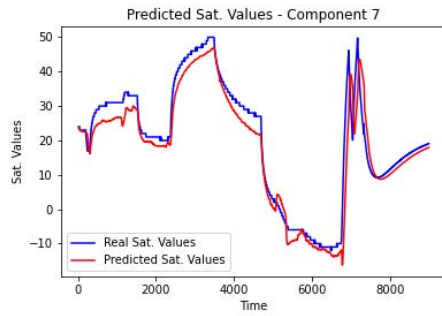
GP - Component 4



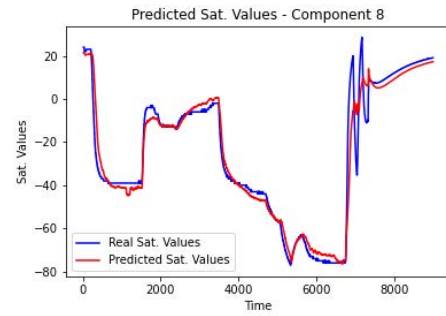
MF - Component 5



GP - Component 6

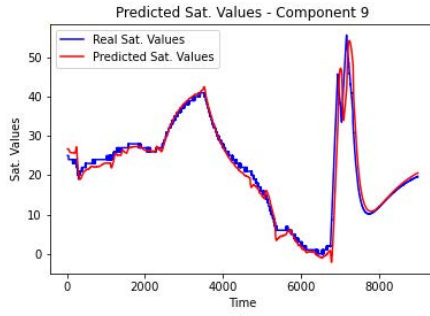


MF - Component 7

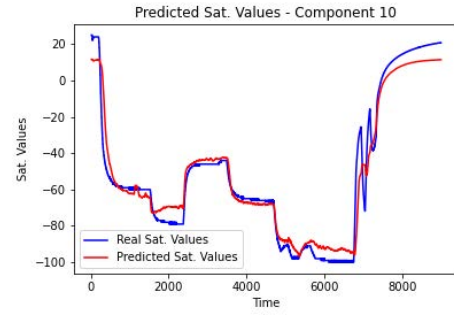


GP - Component 8

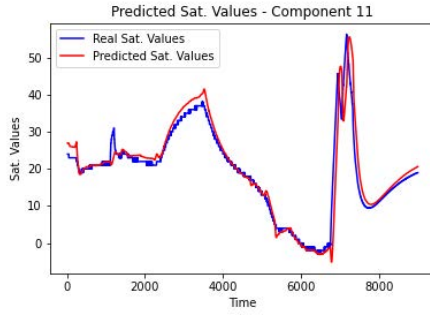
Figure A.63: GRU results using the entire SDS-4 dataset. Blue lines (—) indicate the real satellite values and red lines (—) the predicted data points for components 0-8.



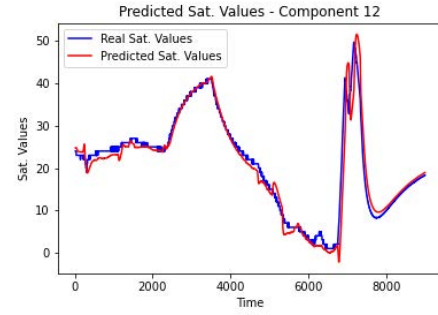
MF - Component 9



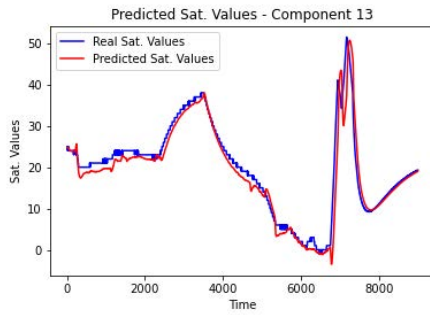
GP - Component 10



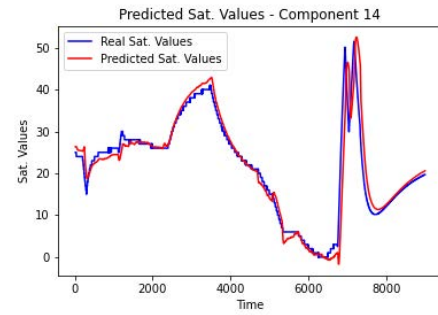
MF - Component 11



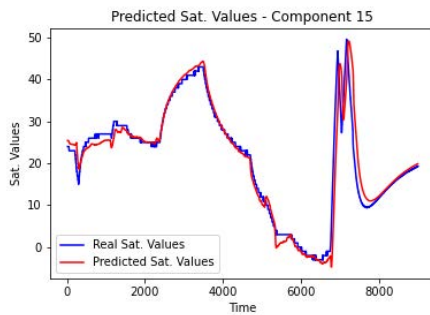
GP - Component 12



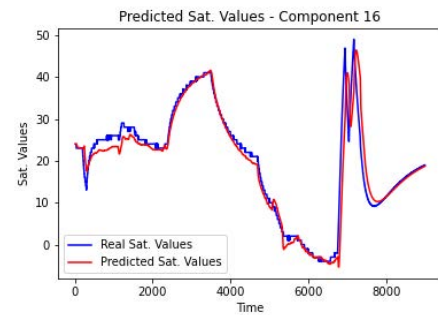
MF - Component 13



GP - Component 14

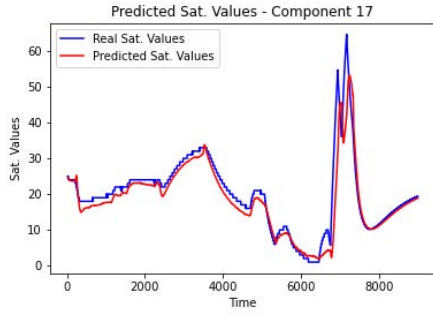


MF - Component 15

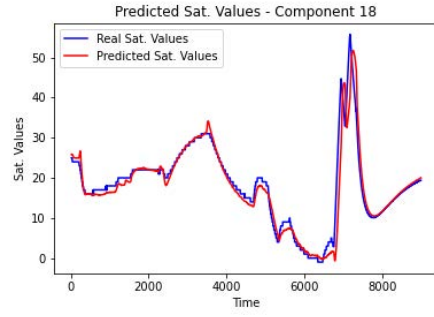


GP - Component 16

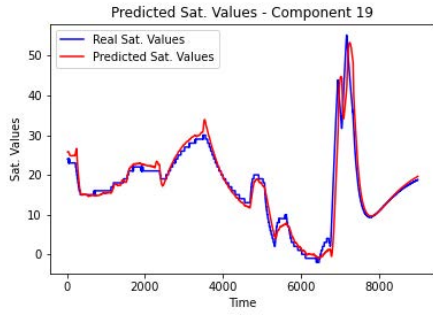
Figure A.64: GRU results using the entire SDS-4 dataset. Blue lines (—) indicate the real satellite values and red lines (—) the predicted data points for components 8-16.



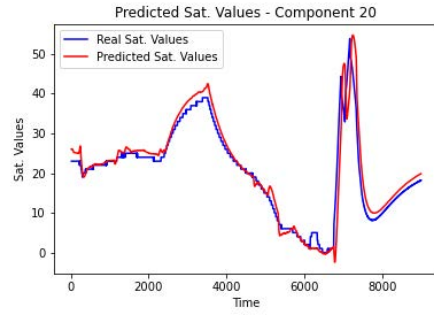
MF - Component 17



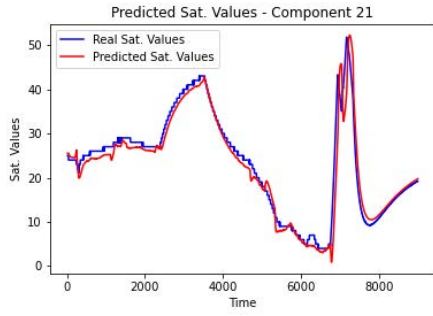
GP - Component 18



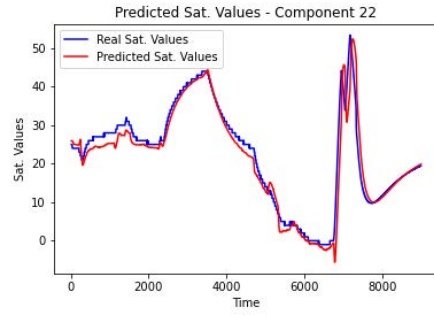
MF - Component 19



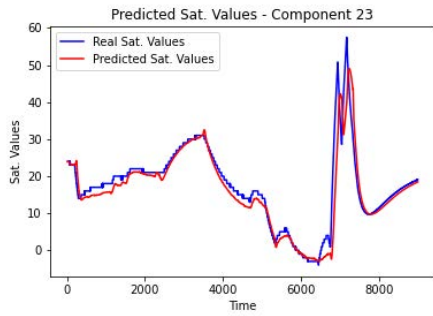
GP - Component 20



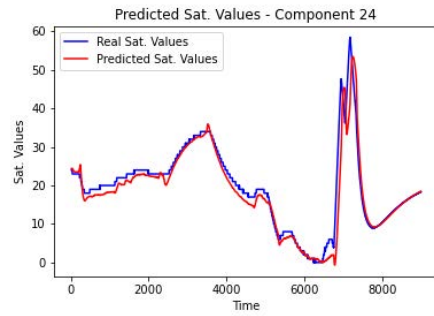
MF - Component 21



GP - Component 22

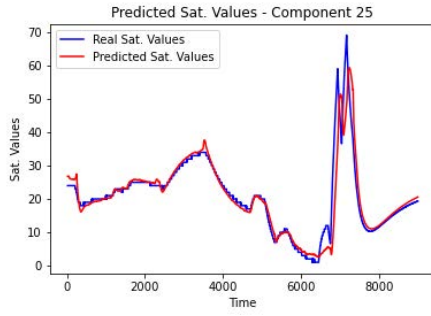


MF - Component 23

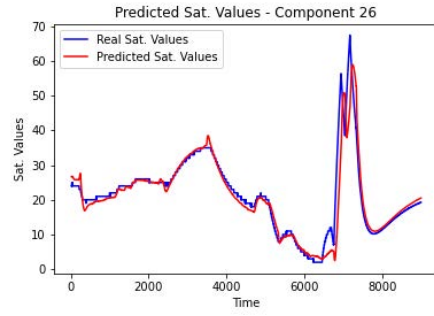


GP - Component 24

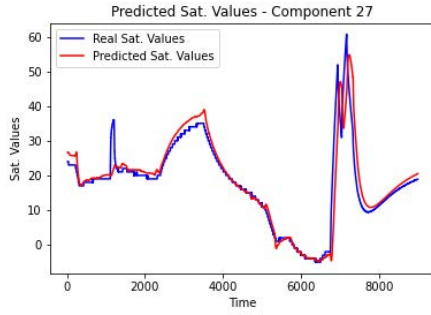
Figure A.65: GRU results using the entire SDS-4 dataset. Blue lines (—) indicate the real satellite values and red lines (—) the predicted data points for components 16-24.



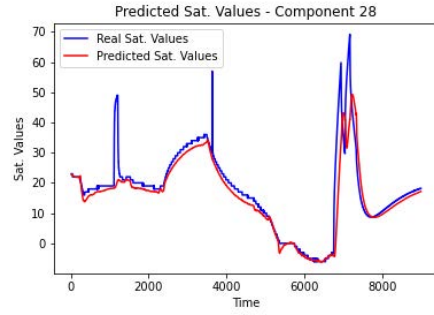
MF - Component 25



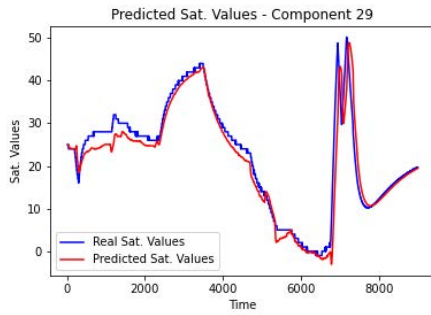
GP - Component 26



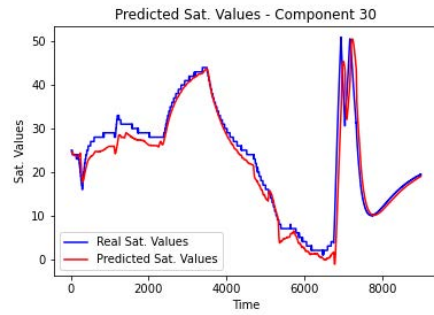
MF - Component 27



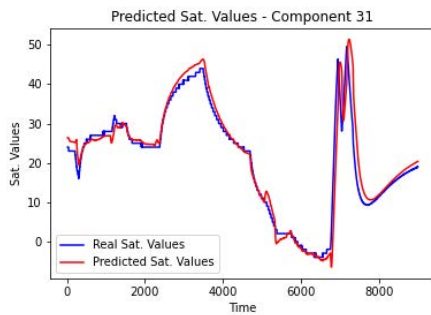
GP - Component 28



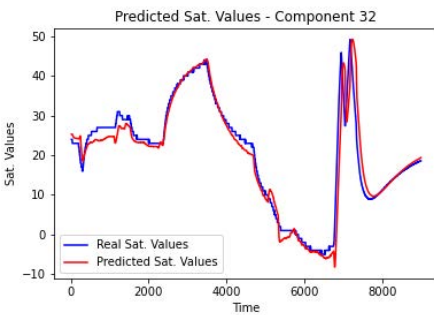
MF - Component 29



GP - Component 30

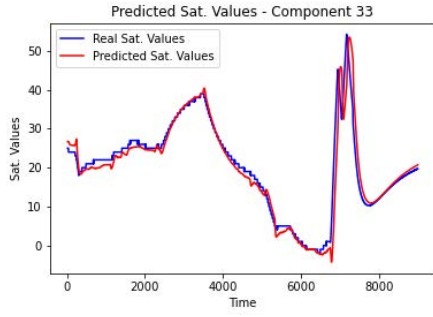


MF - Component 31

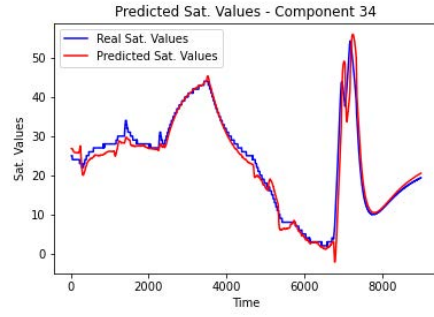


GP - Component 32

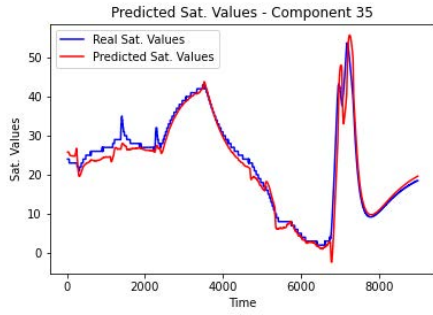
Figure A.66: GRU results using the entire SDS-4 dataset. Blue lines (—) indicate the real satellite values and red lines (—) the predicted data points for components 24-32.



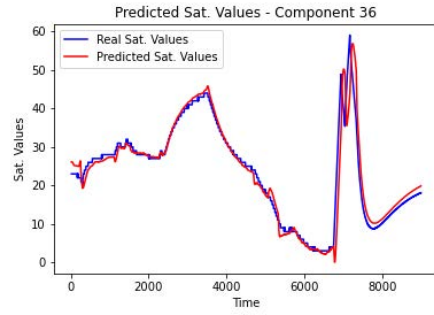
MF - Component 33



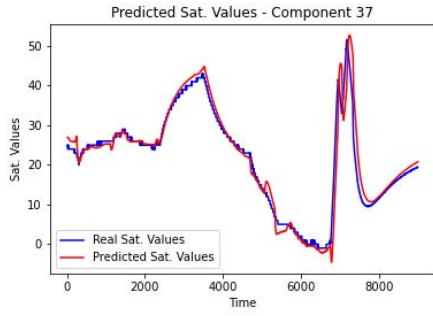
GP - Component 34



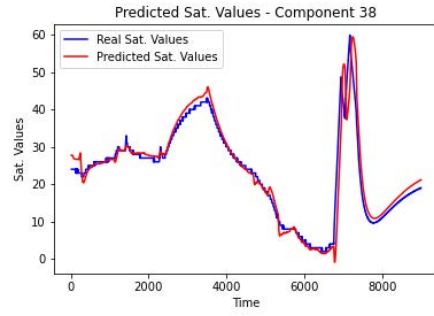
MF - Component 35



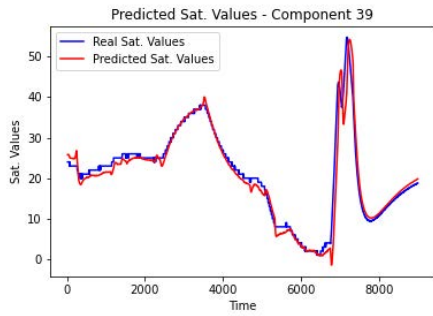
GP - Component 36



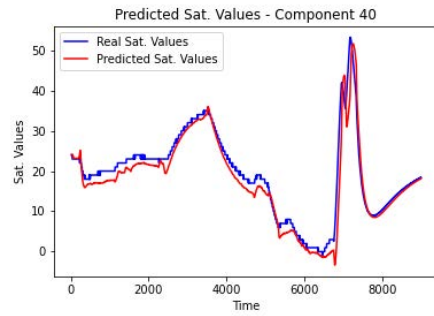
MF - Component 37



GP - Component 38

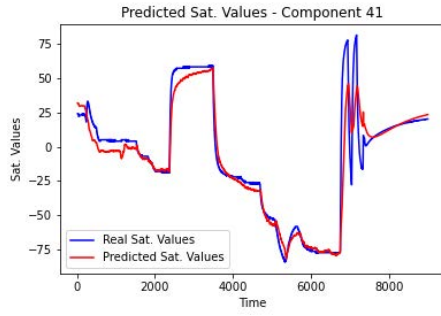


MF - Component 39

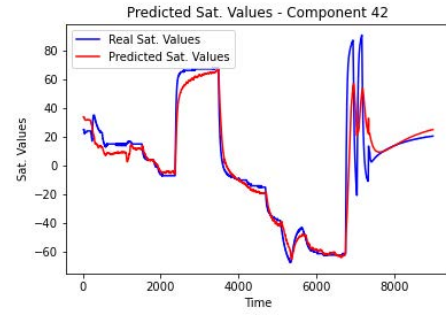


GP - Component 40

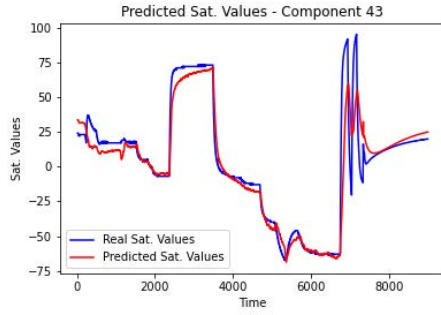
Figure A.67: GRU results using the entire SDS-4 dataset. Blue lines (—) indicate the real satellite values and red lines (—) the predicted data points for components 32-40.



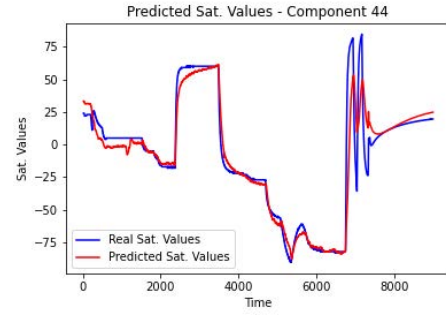
MF - Component 41



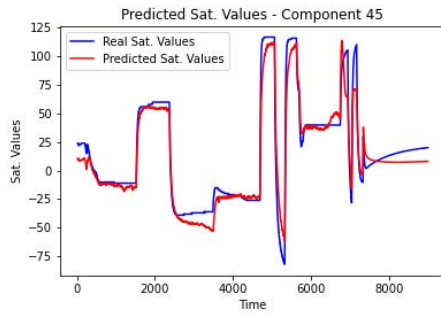
GP - Component 42



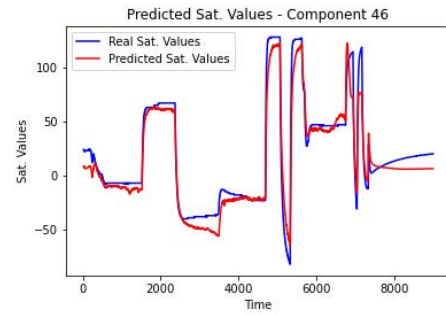
MF - Component 43



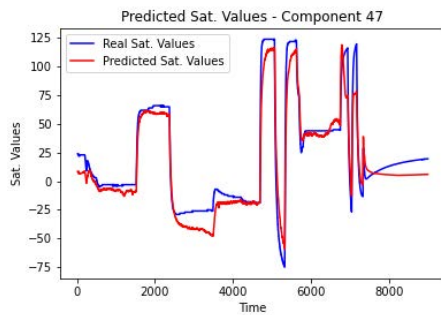
GP - Component 44



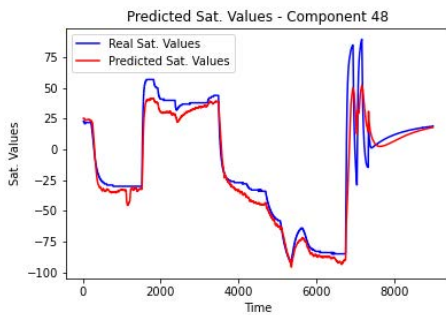
MF - Component 45



GP - Component 46

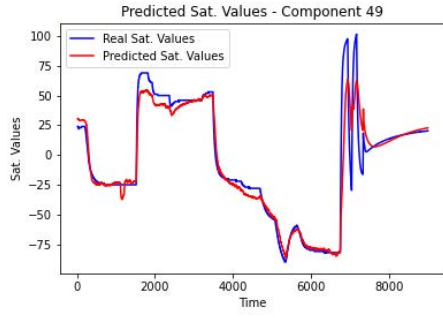


MF - Component 47

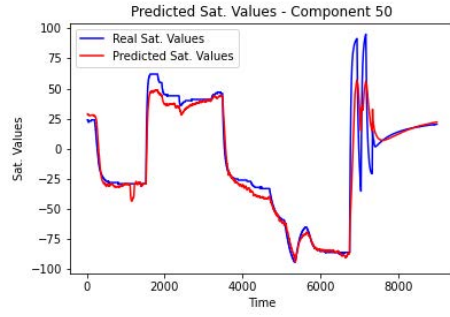


GP - Component 48

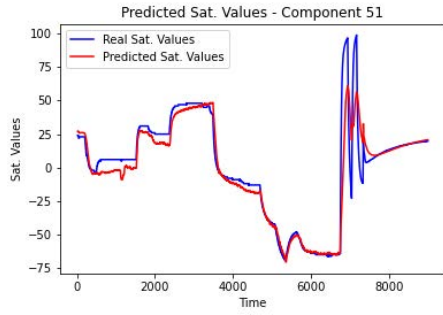
Figure A.68: GRU results using the entire SDS-4 dataset. Blue lines (—) indicate the real satellite values and red lines (—) the predicted data points for components 40-48.



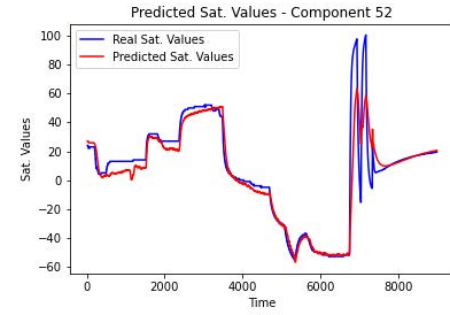
MF - Component 49



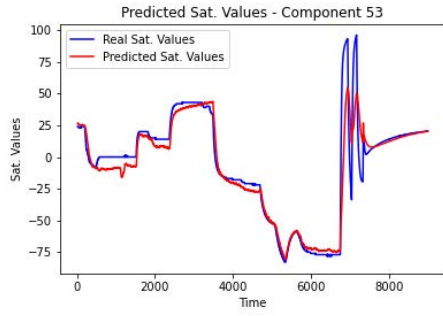
GP - Component 50



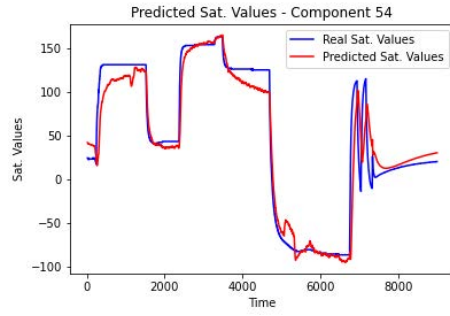
MF - Component 51



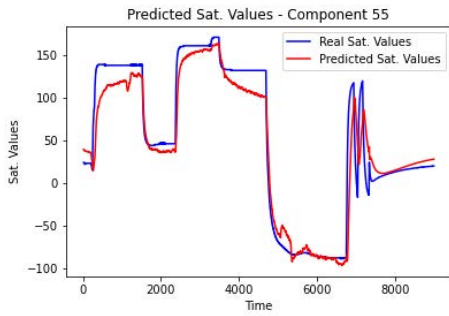
GP - Component 52



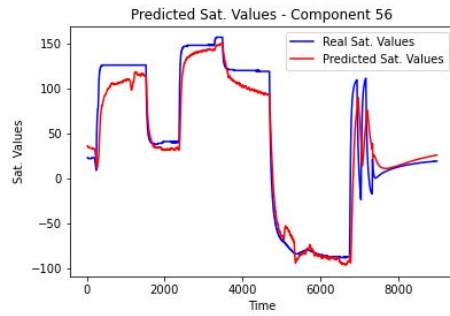
MF - Component 53



GP - Component 54



MF - Component 55



GP - Component 56

Figure A.69: GRU results using the entire SDS-4 dataset. Blue lines (—) indicate the real satellite values and red lines (—) the predicted data points for components 48-56.

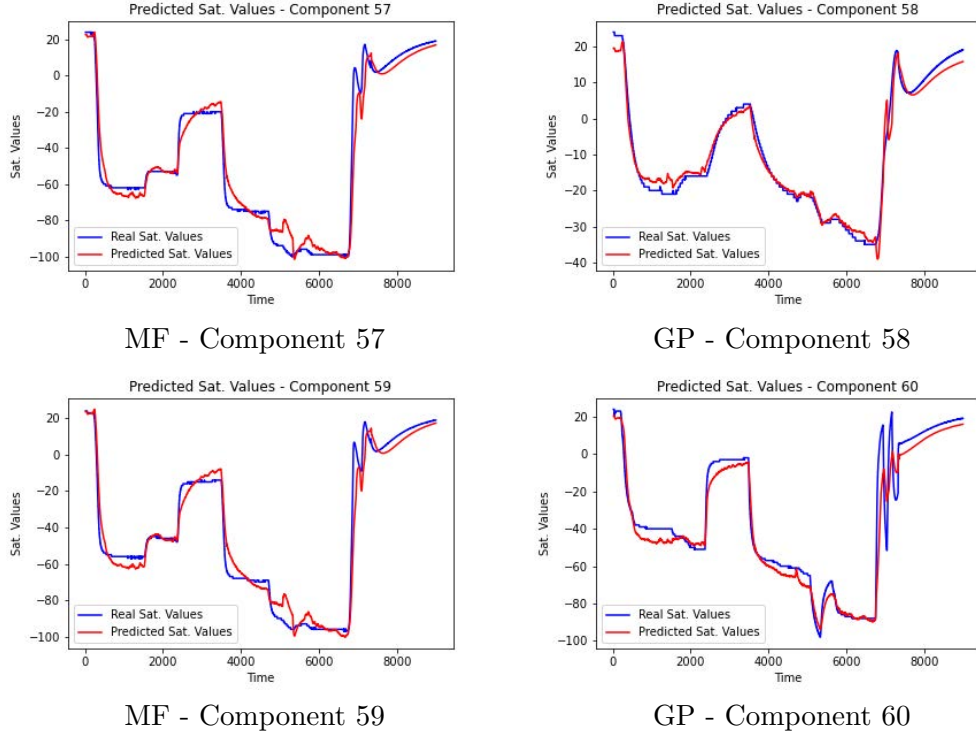


Figure A.70: GRU results using the entire SDS-4 dataset. Blue lines (—) indicate the real satellite values and red lines (—) the predicted data points for components 57-60.

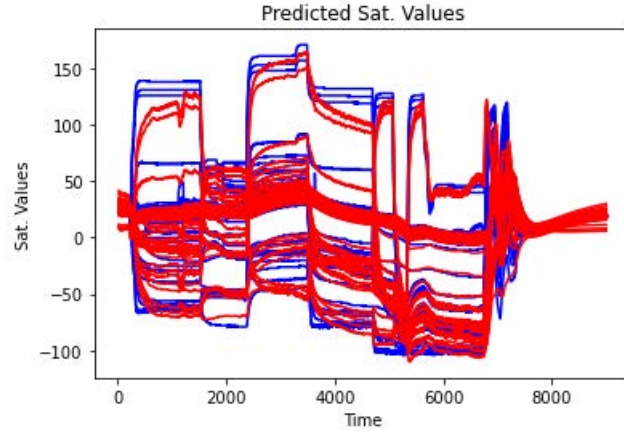
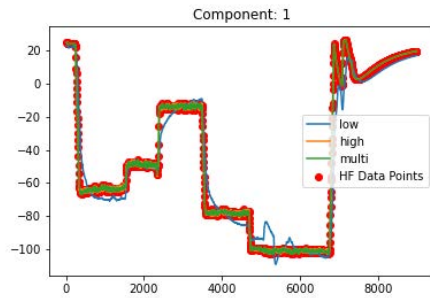
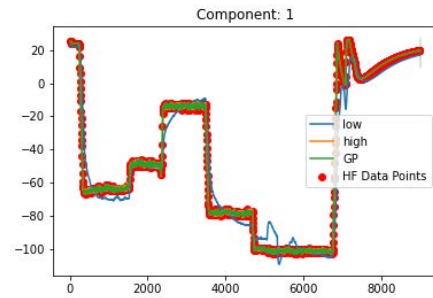


Figure A.71: GRU results using the entire SDS-4 dataset. Blue lines (—) indicate the real satellite values and red lines (—) the predicted data points for all components.

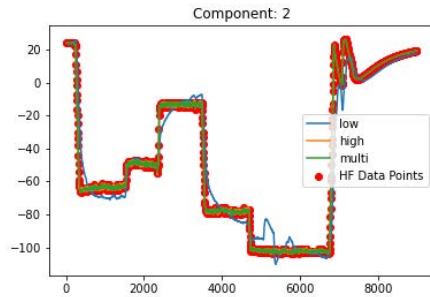
A.8 Uniform SDS-4 HF Points Placement - Entire Dataset



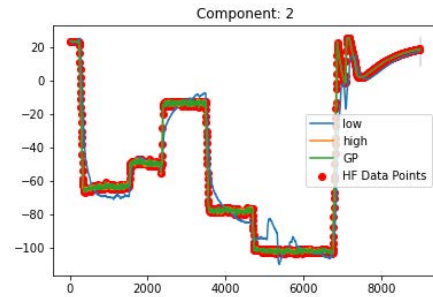
MF - Component 1



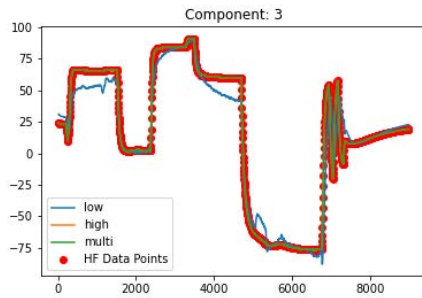
GP - Component 1



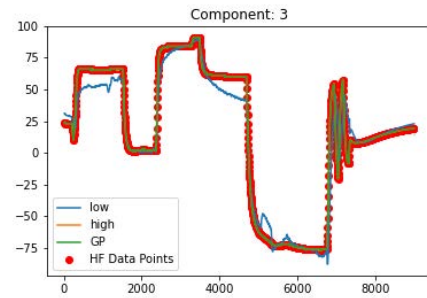
MF - Component 2



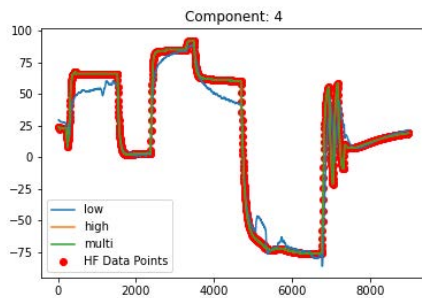
GP - Component 2



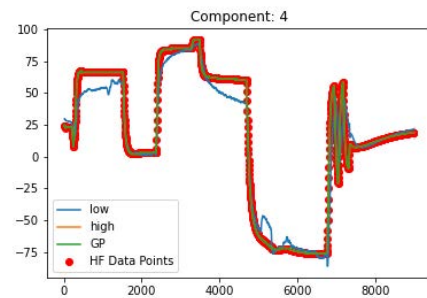
MF - Component 3



GP - Component 3

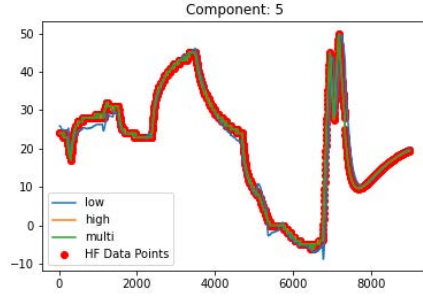


MF - Component 4

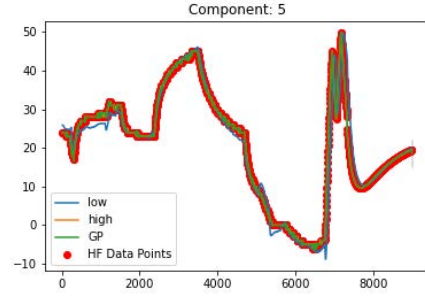


GP - Component 4

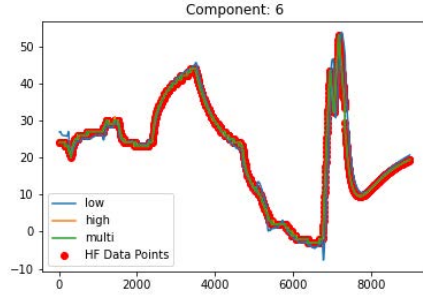
Figure A.72: SDS-4 data using the entire dataset. Multifidelity vs Gaussian process. Uniform high fidelity data points. Blue lines (-) indicate the LFM, orange lines (-) the HFM, green lines (-) the multifidelity estimation and red dots (•) the HFM data points used and the gray area (■) the confidence interval for components 1-4.



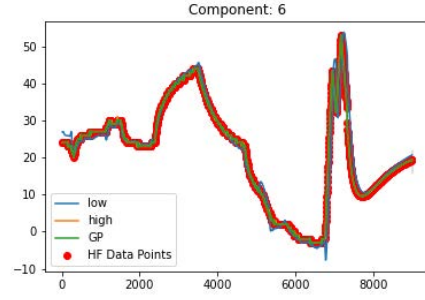
MF - Component 5



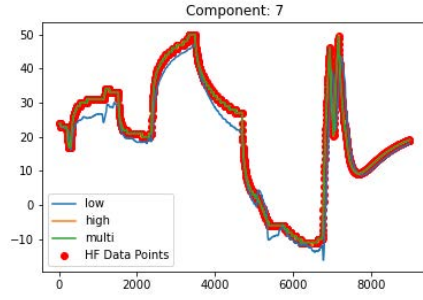
GP - Component 5



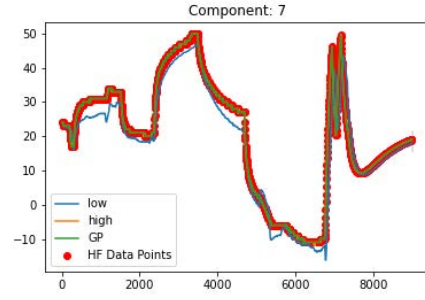
MF - Component 6



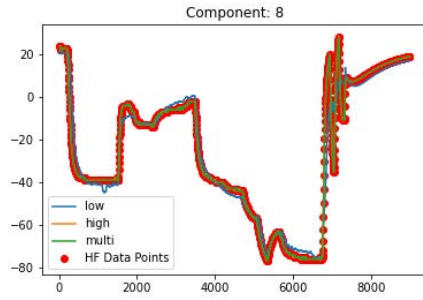
GP - Component 6



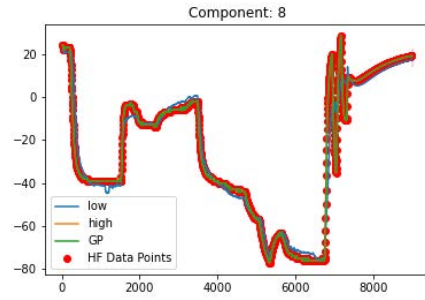
MF - Component 7



GP - Component 7

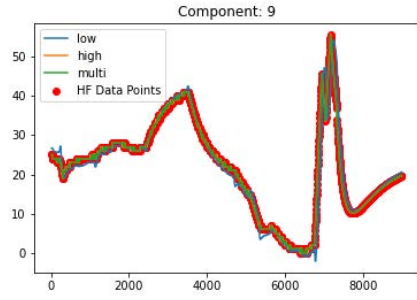


MF - Component 8

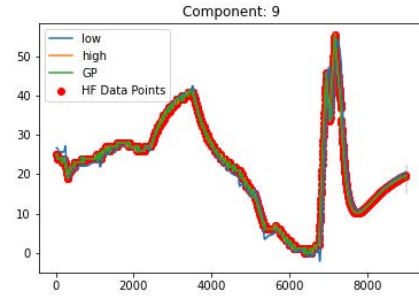


GP - Component 8

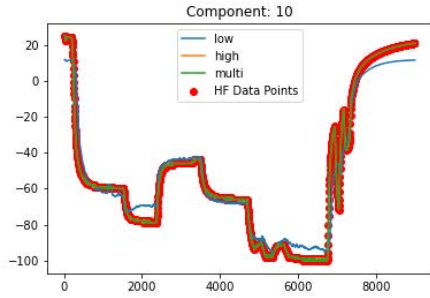
Figure A.73: SDS-4 data using the entire dataset. Multifidelity vs Gaussian process. Uniform high fidelity data points. Blue lines (-) indicate the LFM, orange lines (-) the HFM, green lines (-) the multifidelity estimation and red dots (•) the HFM data points used and the gray area (■) the confidence interval for components 5-8.



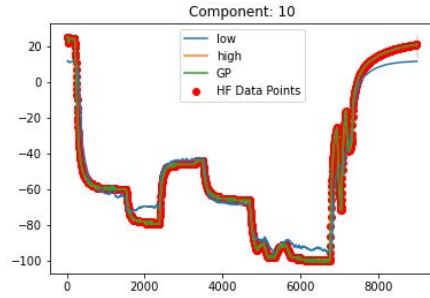
MF - Component 9



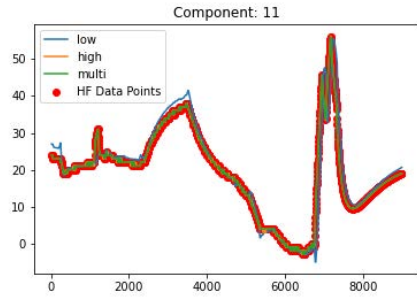
GP - Component 9



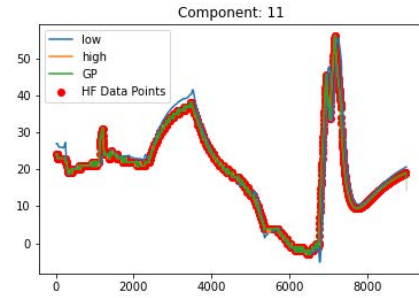
MF - Component 10



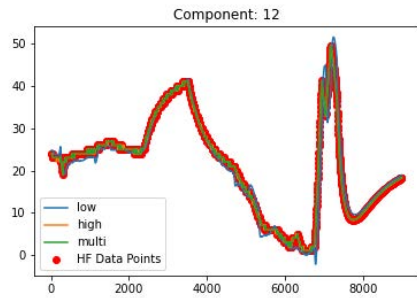
GP - Component 10



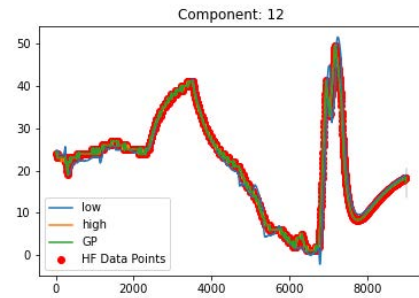
MF - Component 11



GP - Component 11

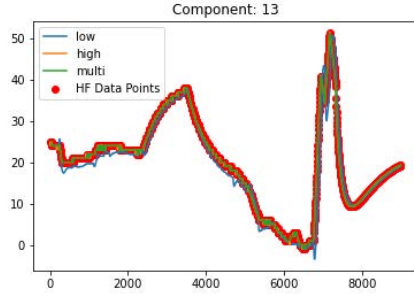


MF - Component 12

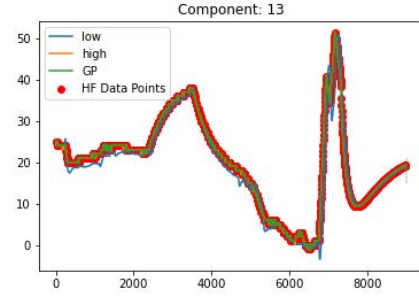


GP - Component 12

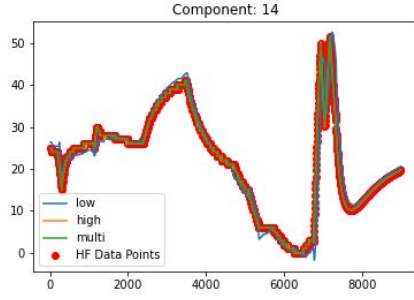
Figure A.74: SDS-4 data using the entire dataset. Multifidelity vs Gaussian process. Uniform high fidelity data points. Blue lines (-) indicate the LFM, orange lines (-) the HFM, green lines (-) the multifidelity estimation and red dots (•) the HFM data points used and the gray area (■) the confidence interval for components 9-12.



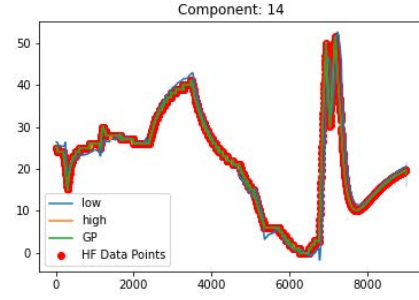
MF - Component 13



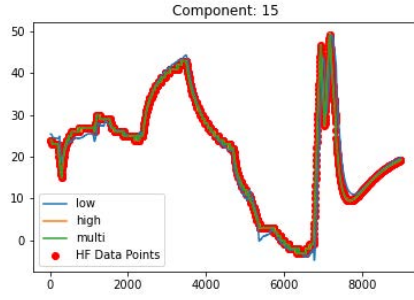
GP - Component 13



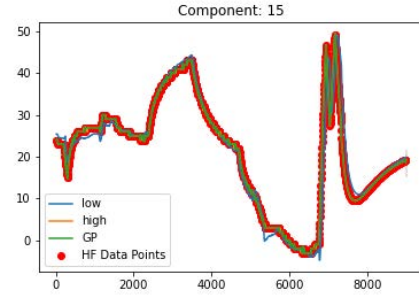
MF - Component 14



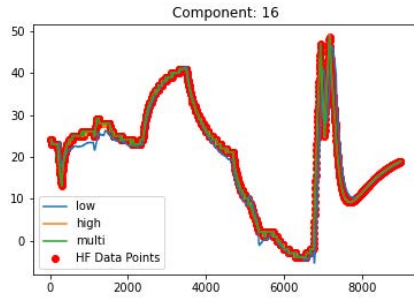
GP - Component 14



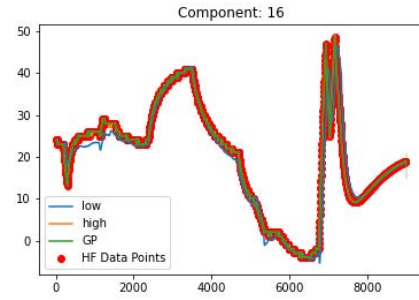
MF - Component 15



GP - Component 15

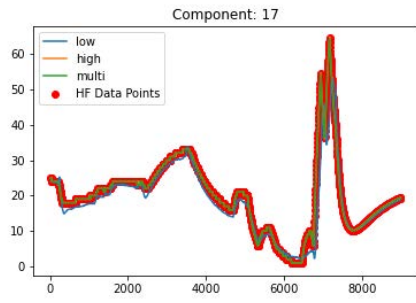


MF - Component 16

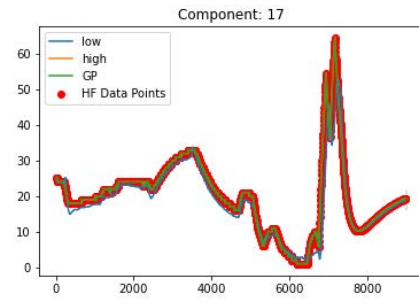


GP - Component 16

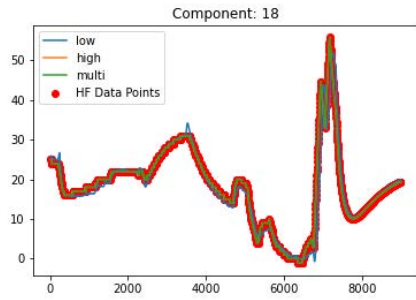
Figure A.75: SDS-4 data using the entire dataset. Multifidelity vs Gaussian process. Uniform high fidelity data points. Blue lines (-) indicate the LFM, orange lines (-) the HFM, green lines (-) the multifidelity estimation and red dots (•) the HFM data points used and the gray area (■) the confidence interval for components 13-16.



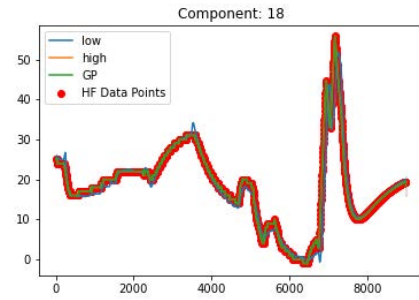
MF - Component 17



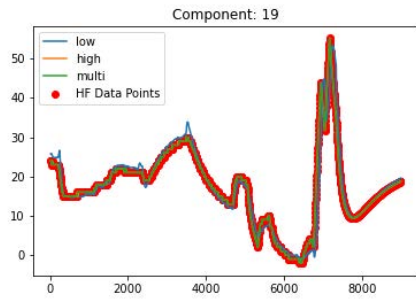
GP - Component 17



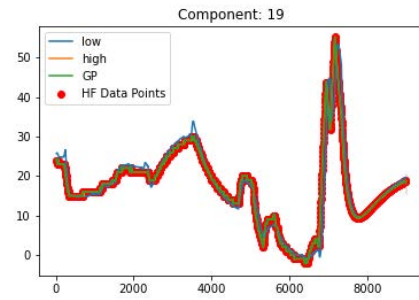
MF - Component 18



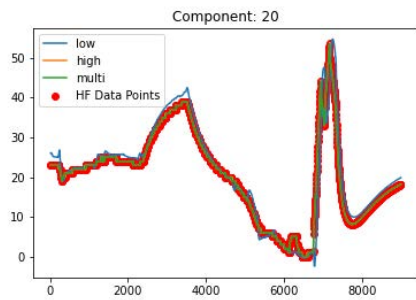
GP - Component 18



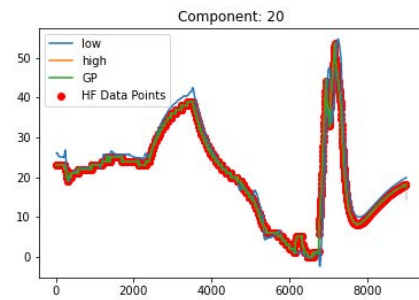
MF - Component 19



GP - Component 19

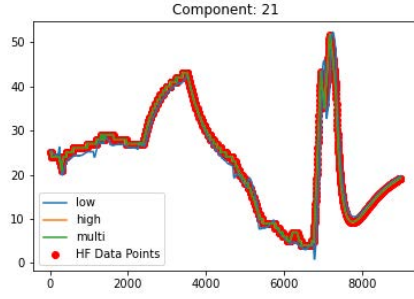


MF - Component 20

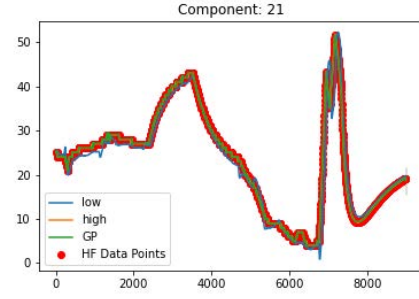


GP - Component 20

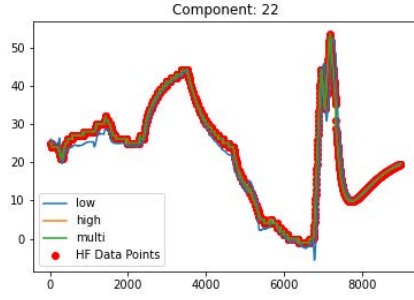
Figure A.76: SDS-4 data using the entire dataset. Multifidelity vs Gaussian process. Uniform high fidelity data points. Blue lines (-) indicate the LFM, orange lines (-) the HFM, green lines (-) the multifidelity estimation and red dots (•) the HFM data points used and the gray area (■) the confidence interval for components 17-20.



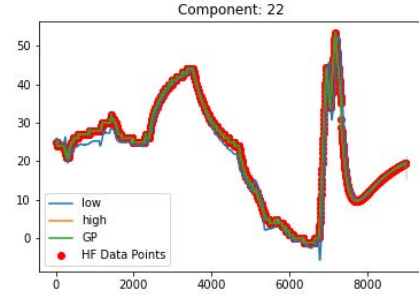
MF - Component 21



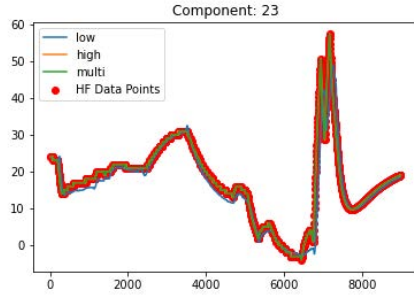
GP - Component 21



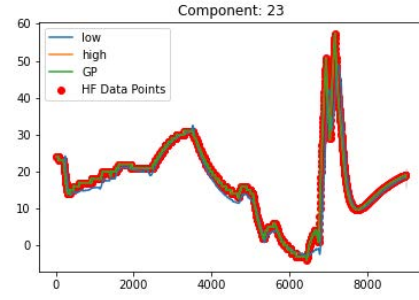
MF - Component 22



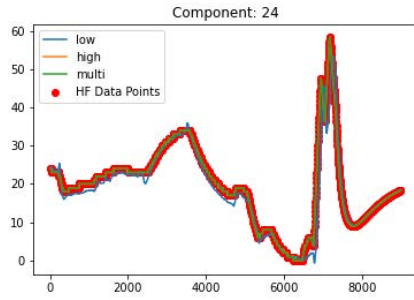
GP - Component 22



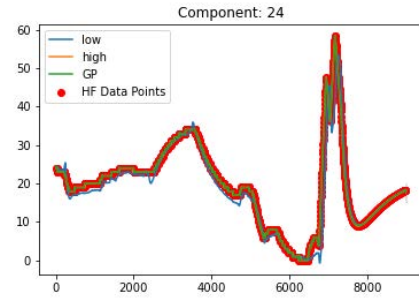
MF - Component 23



GP - Component 23

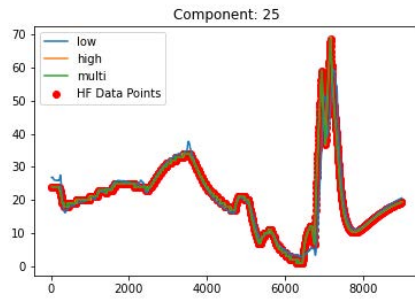


MF - Component 24

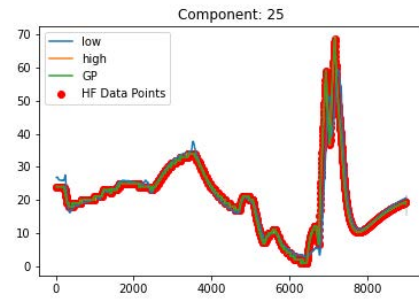


GP - Component 24

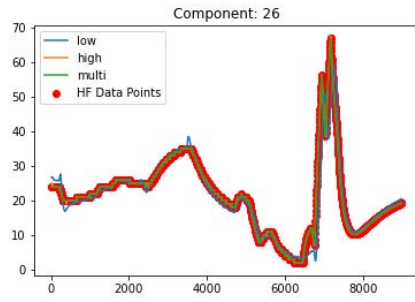
Figure A.77: SDS-4 data using the entire dataset. Multifidelity vs Gaussian process. Uniform high fidelity data points. Blue lines (-) indicate the LFM, orange lines (-) the HFM, green lines (-) the multifidelity estimation and red dots (•) the HFM data points used and the gray area (■) the confidence interval for components 21-24.



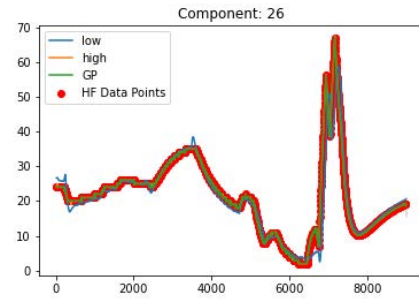
MF - Component 25



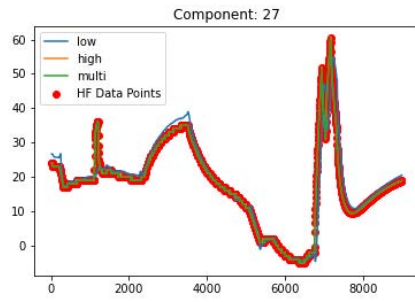
GP - Component 25



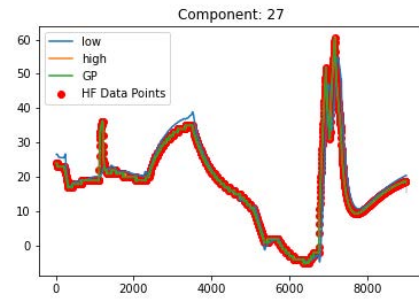
MF - Component 26



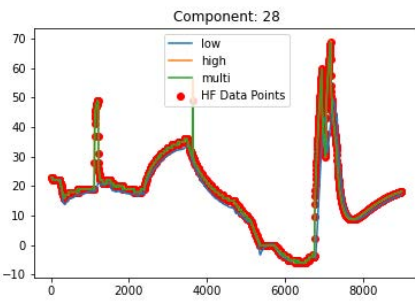
GP - Component 26



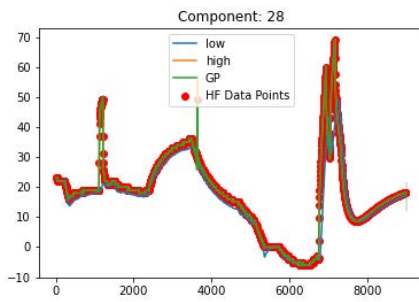
MF - Component 27



GP - Component 27

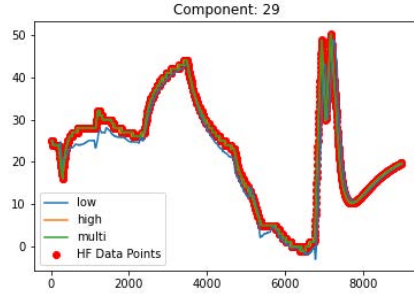


MF - Component 28

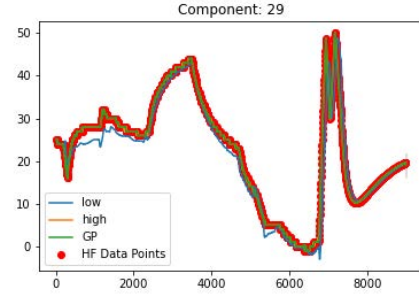


GP - Component 28

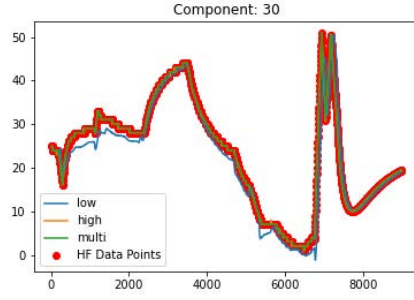
Figure A.78: SDS-4 data using the entire dataset. Multifidelity vs Gaussian process. Uniform high fidelity data points. Blue lines (-) indicate the LFM, orange lines (-) the HFM, green lines (-) the multifidelity estimation and red dots (•) the HFM data points used and the gray area (■) the confidence interval for components 25-28.



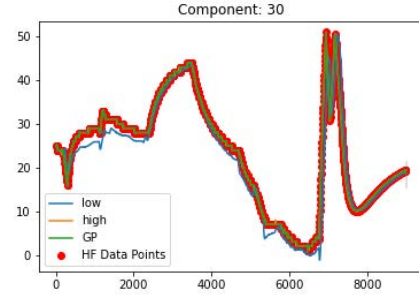
MF - Component 29



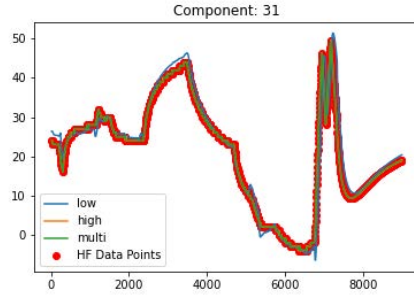
GP - Component 29



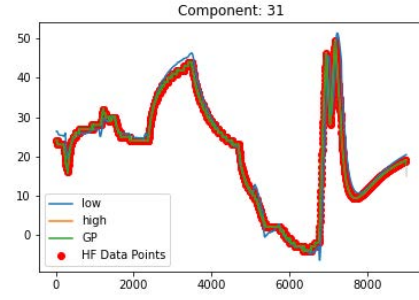
MF - Component 30



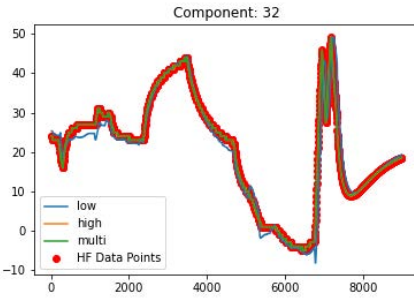
GP - Component 30



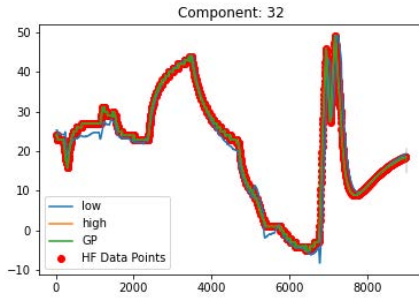
MF - Component 31



GP - Component 31

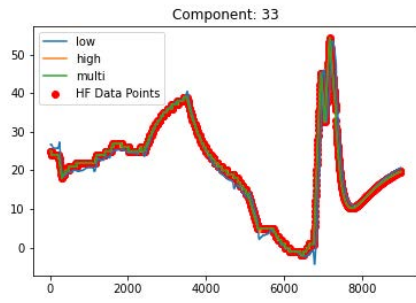


MF - Component 32

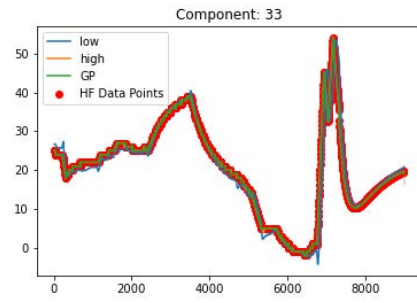


GP - Component 32

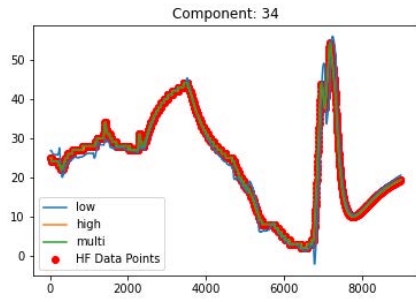
Figure A.79: SDS-4 data using the entire dataset. Multifidelity vs Gaussian process. Uniform high fidelity data points. Blue lines (-) indicate the LFM, orange lines (-) the HFM, green lines (-) the multifidelity estimation and red dots (•) the HFM data points used and the gray area (■) the confidence interval for components 29-32.



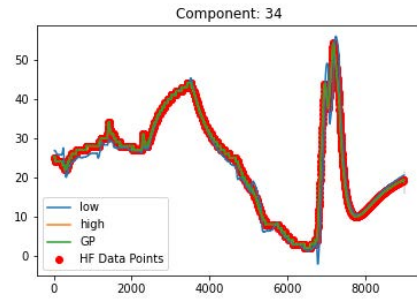
MF - Component 33



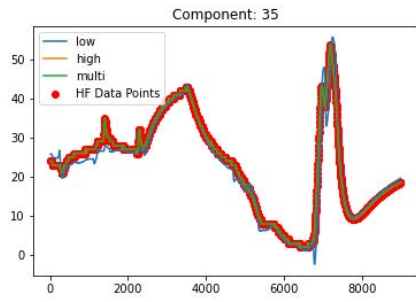
GP - Component 33



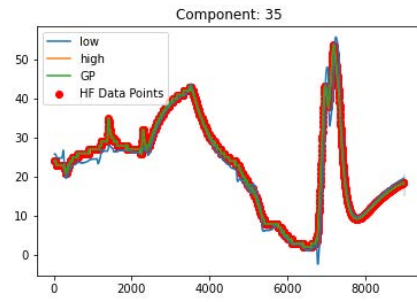
MF - Component 34



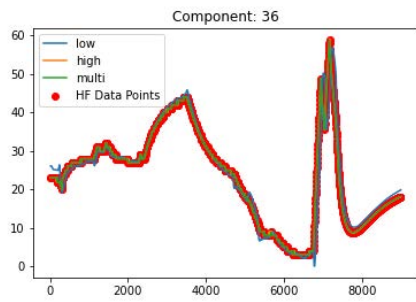
GP - Component 34



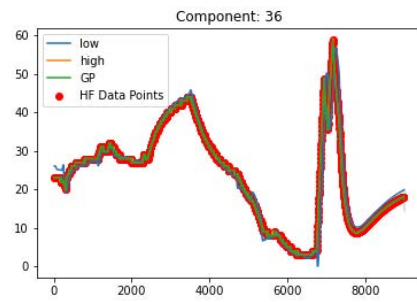
MF - Component 35



GP - Component 35

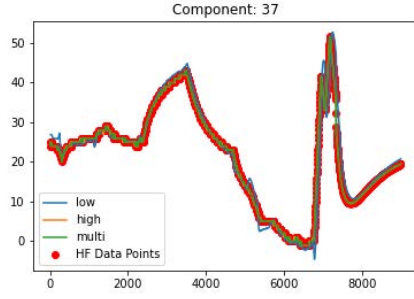


MF - Component 36

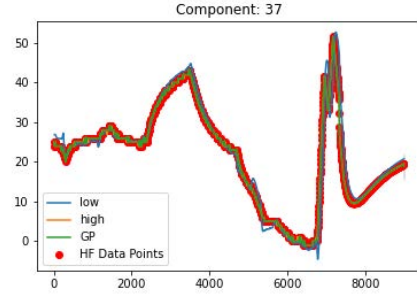


GP - Component 36

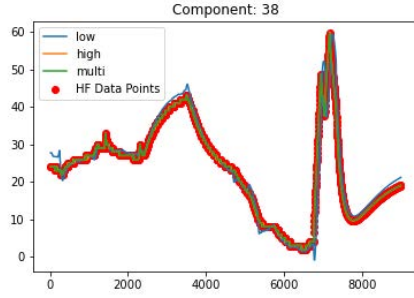
Figure A.80: SDS-4 data using the entire dataset. Multifidelity vs Gaussian process. Uniform high fidelity data points. Blue lines (-) indicate the LFM, orange lines (-) the HFM, green lines (-) the multifidelity estimation and red dots (•) the HFM data points used and the gray area (■) the confidence interval for components 33-36.



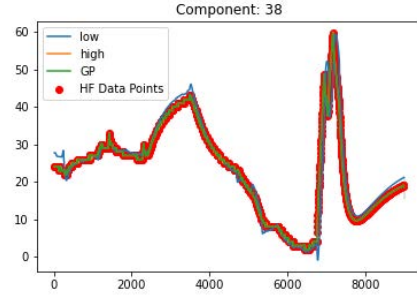
MF - Component 37



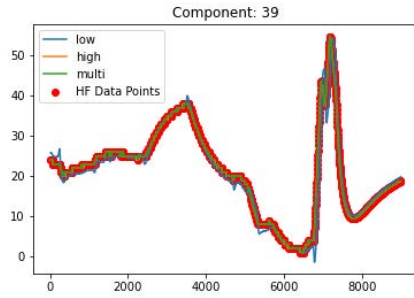
GP - Component 37



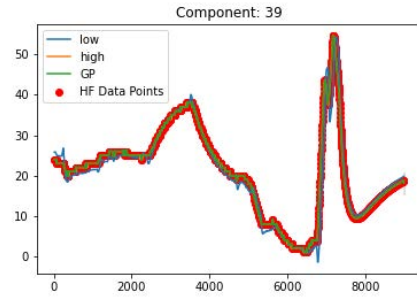
MF - Component 38



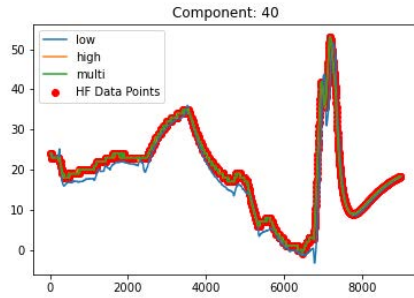
GP - Component 38



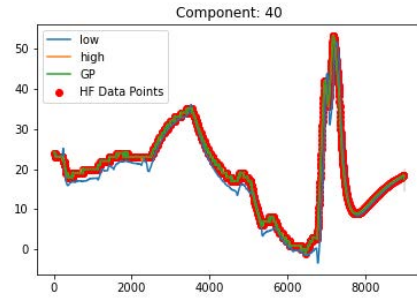
MF - Component 39



GP - Component 39

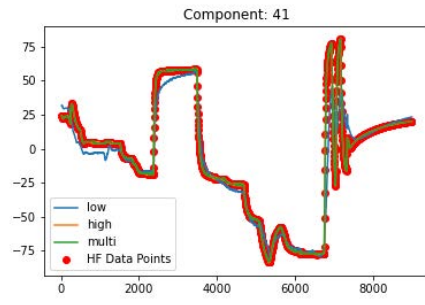


MF - Component 40

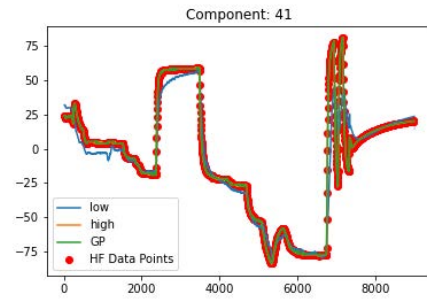


GP - Component 40

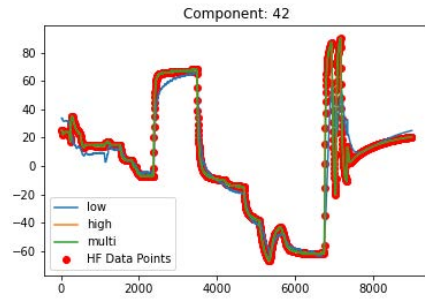
Figure A.81: SDS-4 data using the entire dataset. Multifidelity vs Gaussian process. Uniform high fidelity data points. Blue lines (-) indicate the LFM, orange lines (-) the HFM, green lines (-) the multifidelity estimation and red dots (•) the HFM data points used and the gray area (■) the confidence interval for components 37-40.



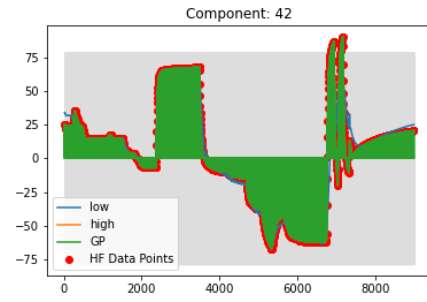
MF - Component 41



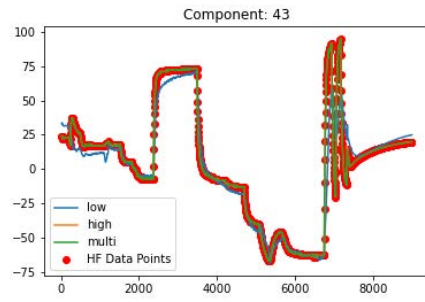
GP - Component 41



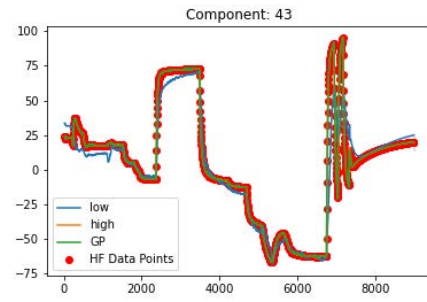
MF - Component 42



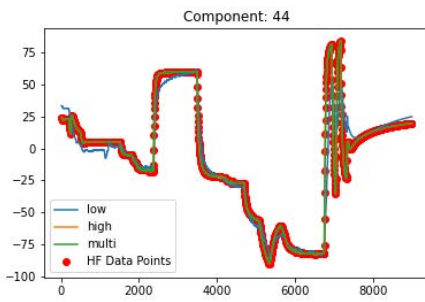
GP - Component 42



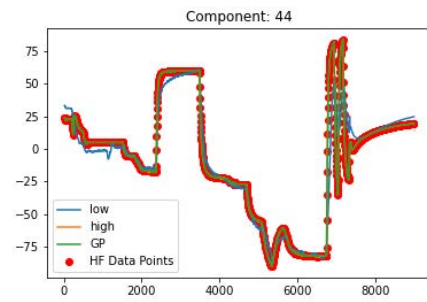
MF - Component 43



GP - Component 43

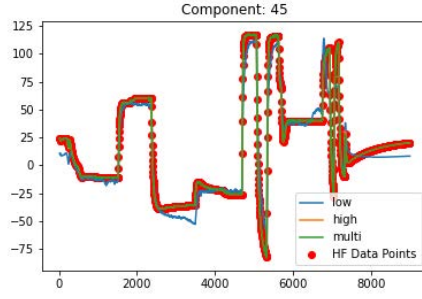


MF - Component 44

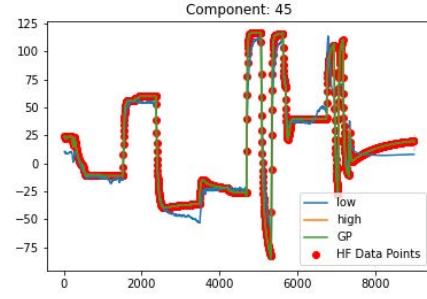


GP - Component 44

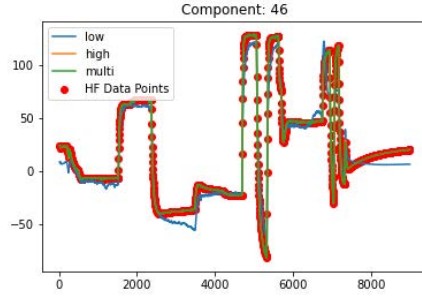
Figure A.82: SDS-4 data using the entire dataset. Multifidelity vs Gaussian process. Uniform high fidelity data points. Blue lines (-) indicate the LFM, orange lines (-) the HFM, green lines (-) the multifidelity estimation and red dots (•) the HFM data points used and the gray area (■) the confidence interval for components 41-44.



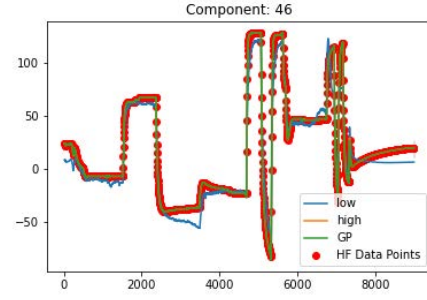
MF - Component 45



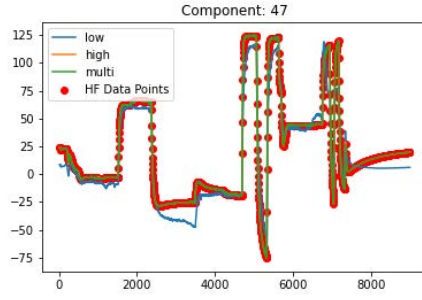
GP - Component 45



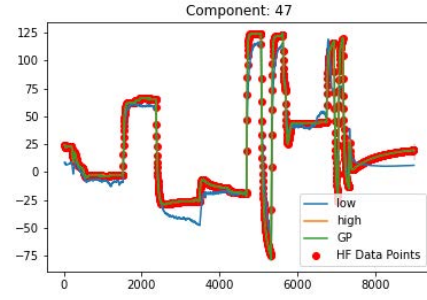
MF - Component 46



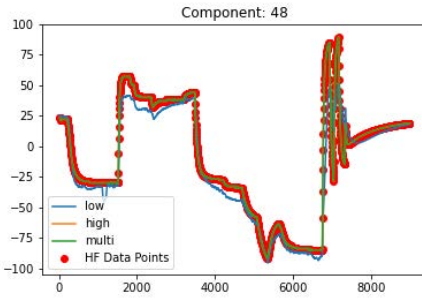
GP - Component 46



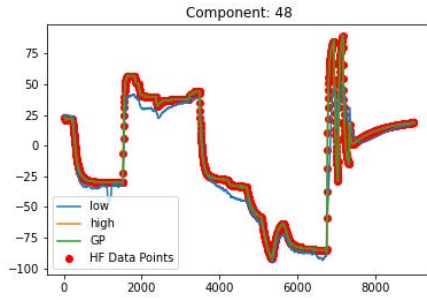
MF - Component 47



GP - Component 47

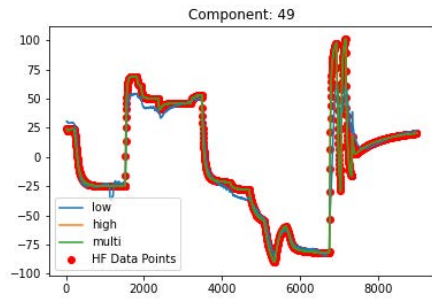


MF - Component 48

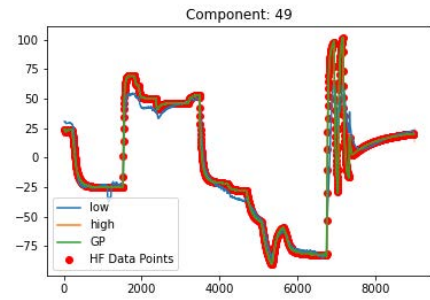


GP - Component 48

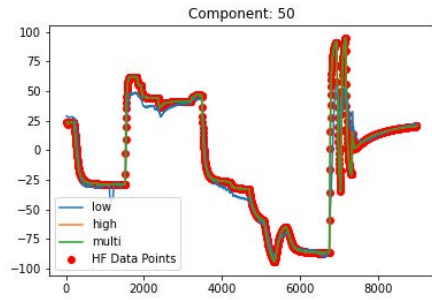
Figure A.83: SDS-4 data using the entire dataset. Multifidelity vs Gaussian process. Uniform high fidelity data points. Blue lines (-) indicate the LFM, orange lines (-) the HFM, green lines (-) the multifidelity estimation and red dots (•) the HFM data points used and the gray area (■) the confidence interval for components 45-48.



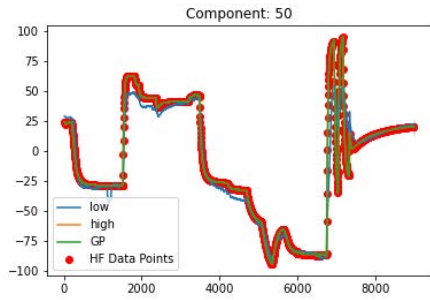
MF - Component 49



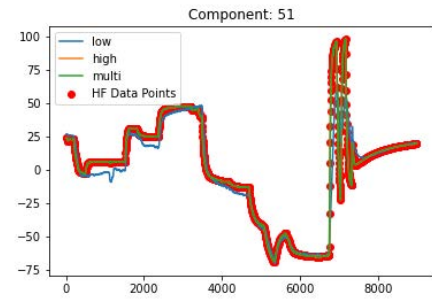
GP - Component 49



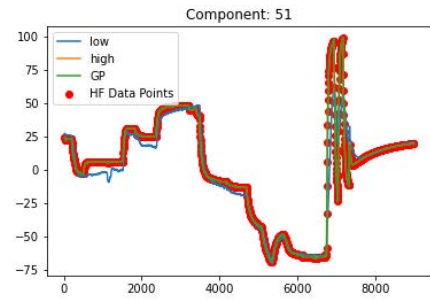
MF - Component 50



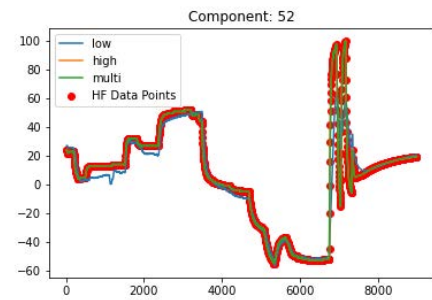
GP - Component 50



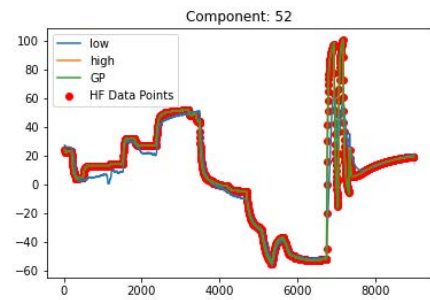
MF - Component 51



GP - Component 51

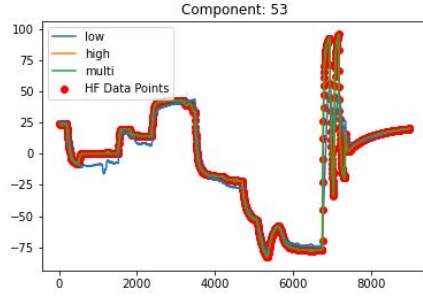


MF - Component 52

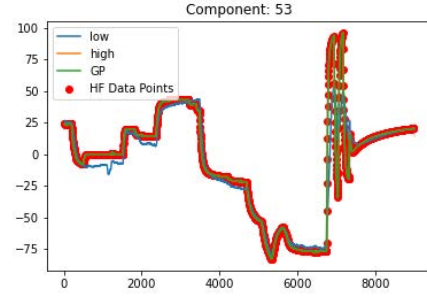


GP - Component 52

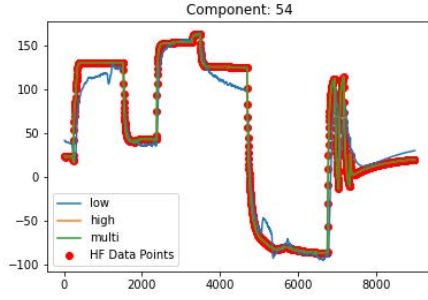
Figure A.84: SDS-4 data using the entire dataset. Multifidelity vs Gaussian process. Uniform high fidelity data points. Blue lines (-) indicate the LFM, orange lines (-) the HFM, green lines (-) the multifidelity estimation and red dots (•) the HFM data points used and the gray area (■) the confidence interval for components 49-52.



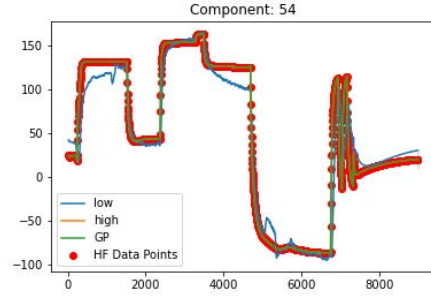
MF - Component 53



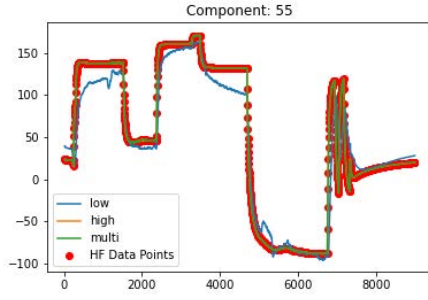
GP - Component 53



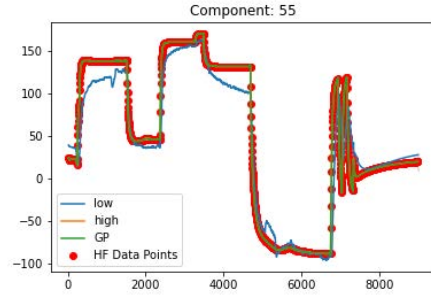
MF - Component 54



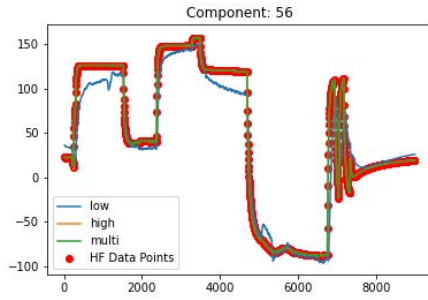
GP - Component 54



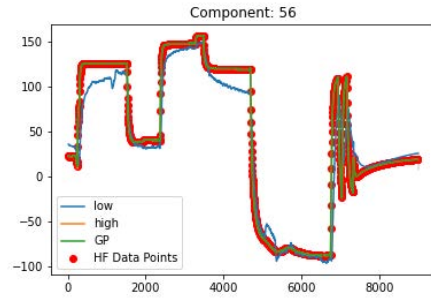
MF - Component 55



GP - Component 55

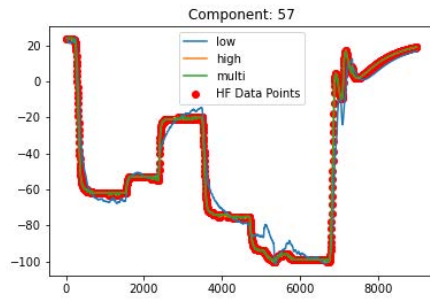


MF - Component 56

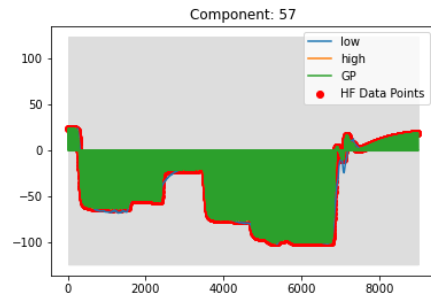


GP - Component 56

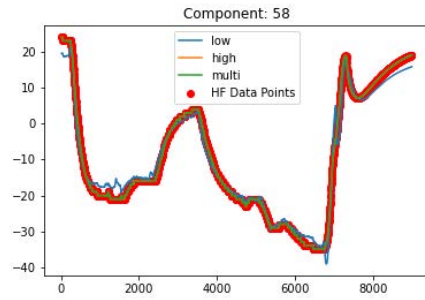
Figure A.85: SDS-4 data using the entire dataset. Multifidelity vs Gaussian process. Uniform high fidelity data points. Blue lines (-) indicate the LFM, orange lines (-) the HFM, green lines (-) the multifidelity estimation and red dots (•) the HFM data points used and the gray area (■) the confidence interval for components 53-56.



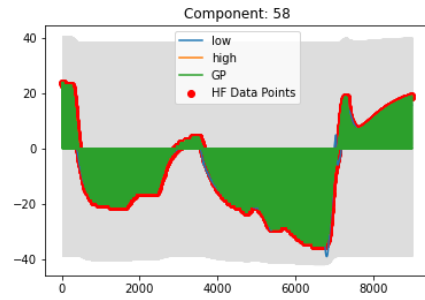
MF - Component 57



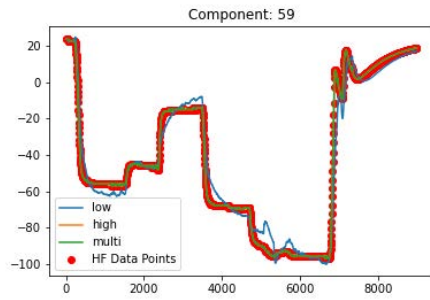
GP - Component 57



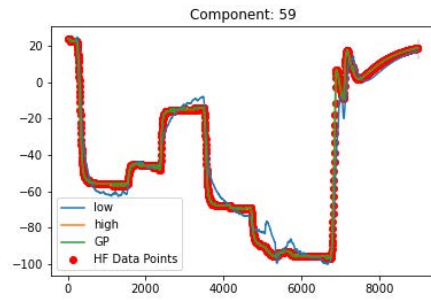
MF - Component 58



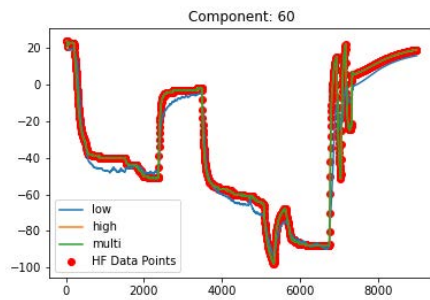
GP - Component 58



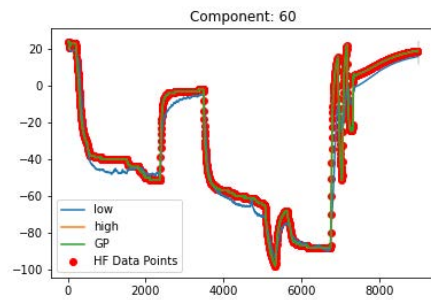
MF - Component 59



GP - Component 59



MF - Component 60



GP - Component 60

Figure A.86: SDS-4 data using the entire dataset. Multifidelity vs Gaussian process. Uniform high fidelity data points. Blue lines (-) indicate the LFM, orange lines (-) the HFM, green lines (-) the multifidelity estimation and red dots (•) the HFM data points used and the gray area (■) the confidence interval for components 57-60.

A.9 Random SDS-4 HF Points Placement - Entire Dataset

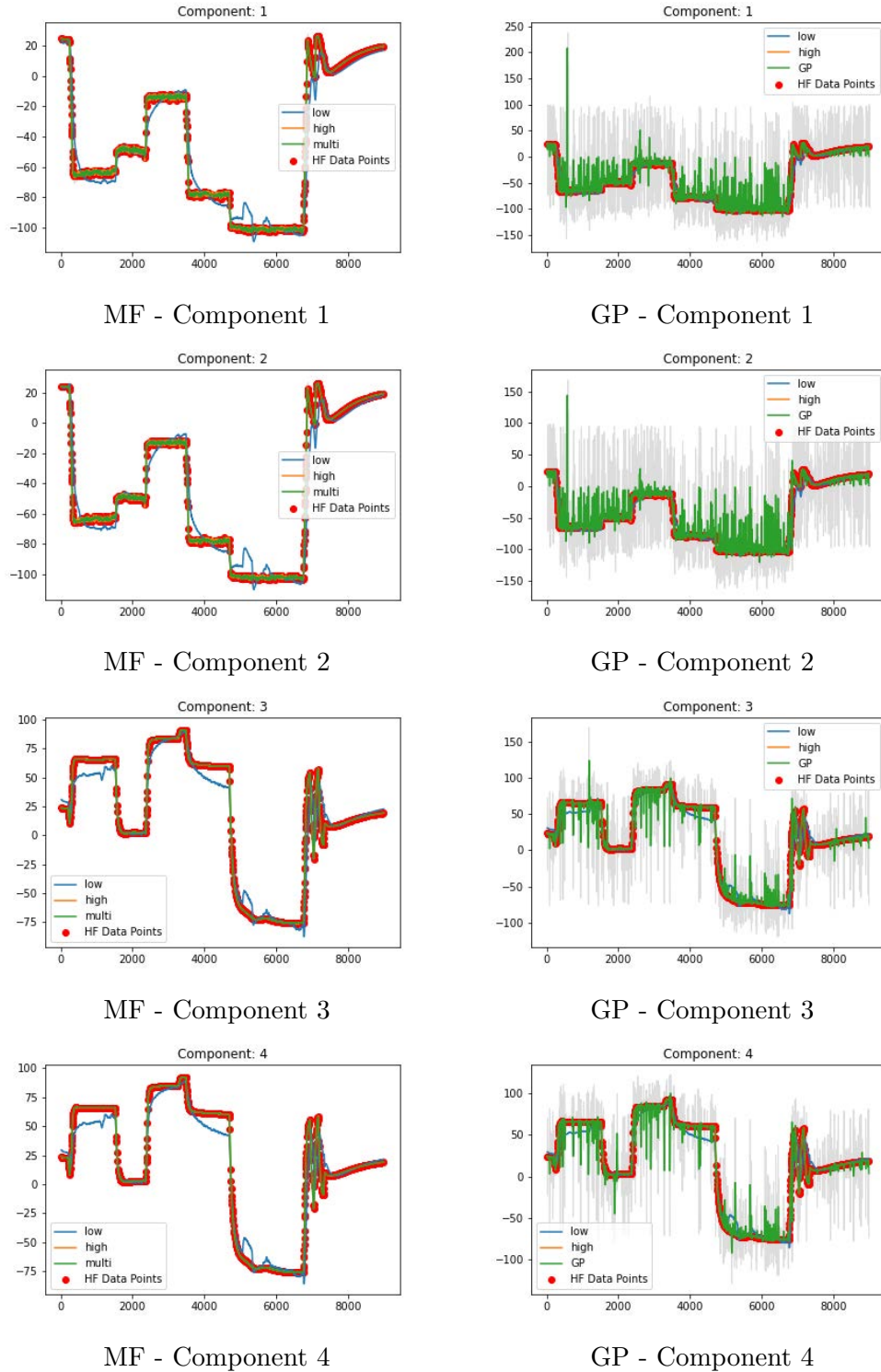
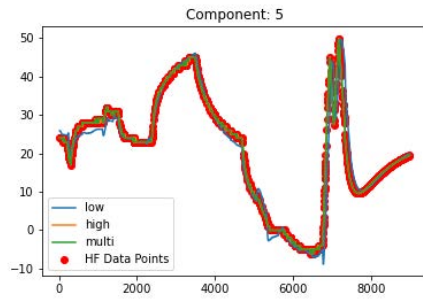
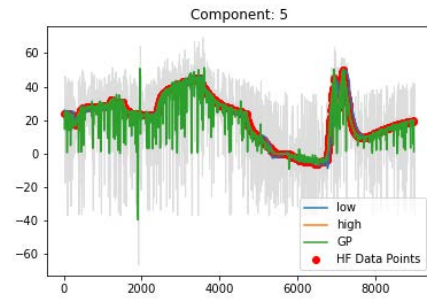


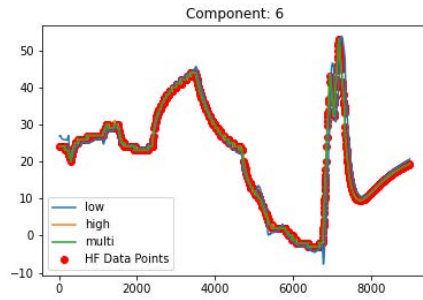
Figure A.87: SDS-4 data using the entire dataset. Multifidelity vs Gaussian process. Random high fidelity data points. Blue lines (-) indicate the LFM, orange lines (-) the HFM, green lines (-) the multifidelity estimation and red dots (•) the HFM data points used and the gray area (■) the confidence interval for components 1-4.



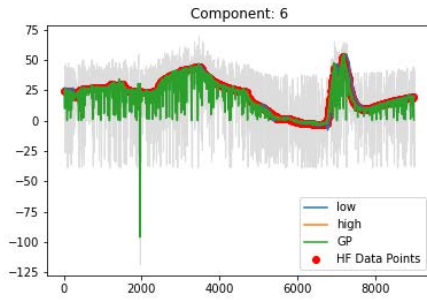
MF - Component 5



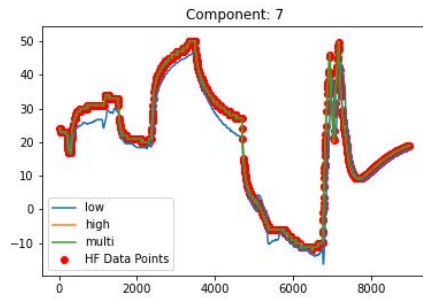
GP - Component 5



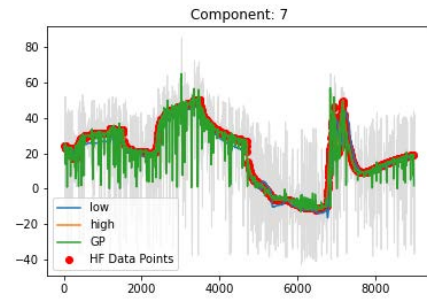
MF - Component 6



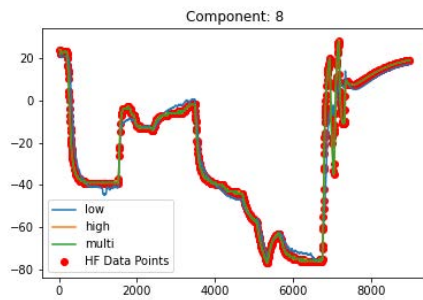
GP - Component 6



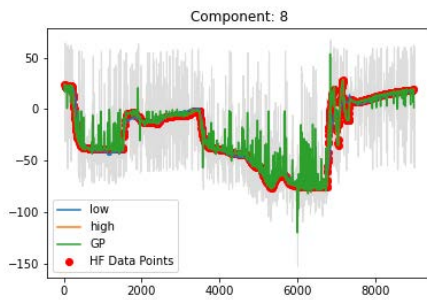
MF - Component 7



GP - Component 7

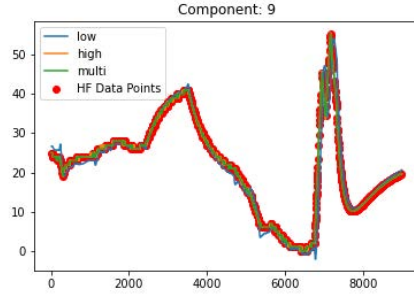


MF - Component 8

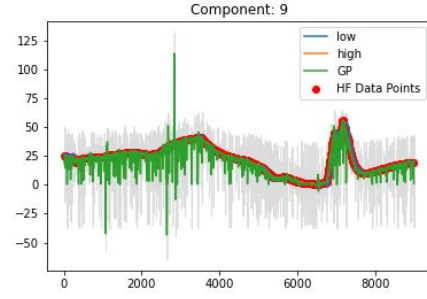


GP - Component 8

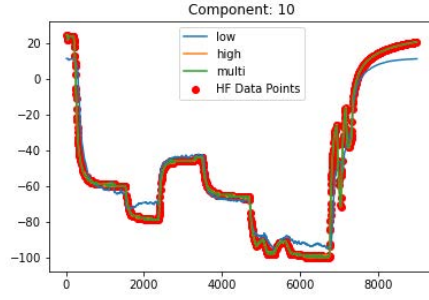
Figure A.88: SDS-4 data using the entire dataset. Multifidelity vs Gaussian process. Random high fidelity data points. Blue lines (-) indicate the LFM, orange lines (-) the HFM, green lines (-) the multifidelity estimation and red dots (•) the HFM data points used and the gray area (■) the confidence interval for components 5-8.



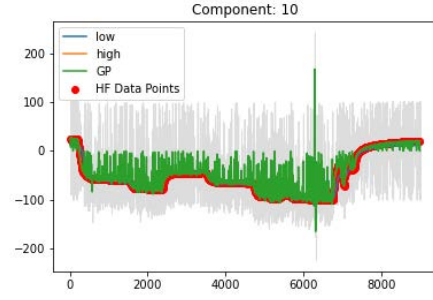
MF - Component 9



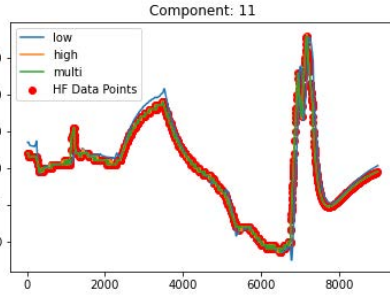
GP - Component 9



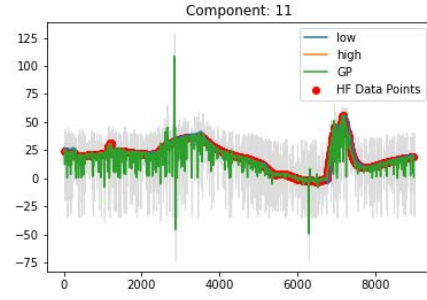
MF - Component 10



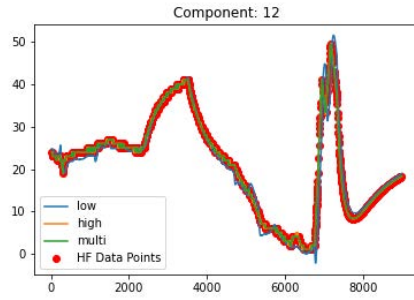
GP - Component 10



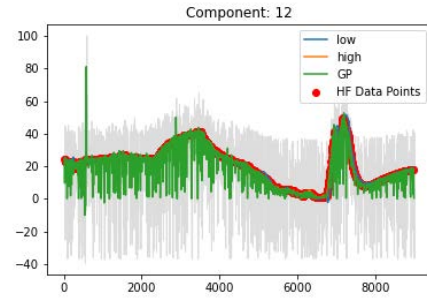
MF - Component 11



GP - Component 11

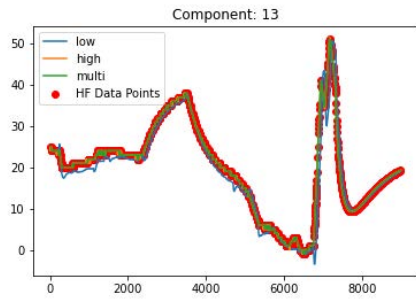


MF - Component 12

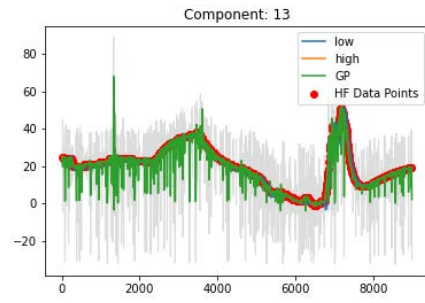


GP - Component 12

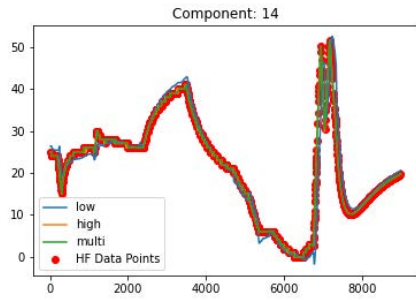
Figure A.89: SDS-4 data using the entire dataset. Multifidelity vs Gaussian process. Random high fidelity data points. Blue lines (-) indicate the LFM, orange lines (-) the HFM, green lines (-) the multifidelity estimation and red dots (•) the HFM data points used and the gray area (■) the confidence interval for components 9-12.



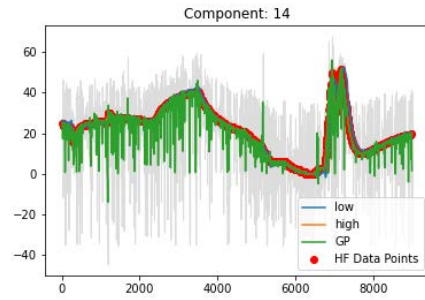
MF - Component 13



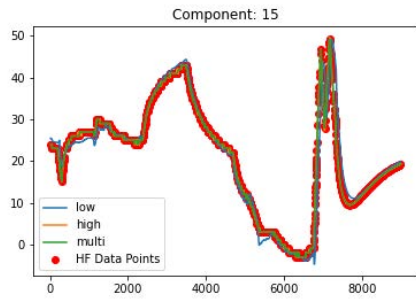
GP - Component 13



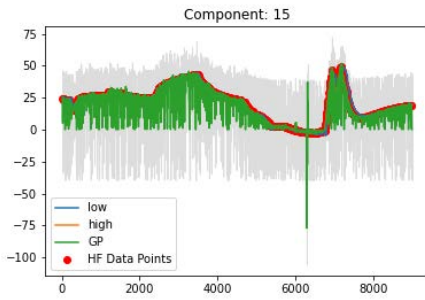
MF - Component 14



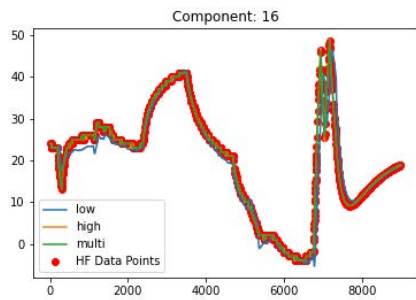
GP - Component 14



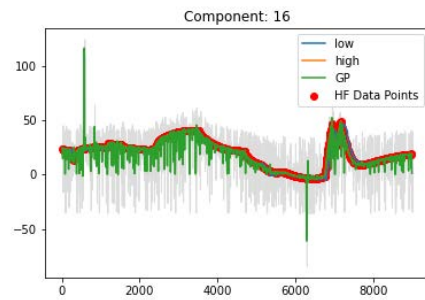
MF - Component 15



GP - Component 15

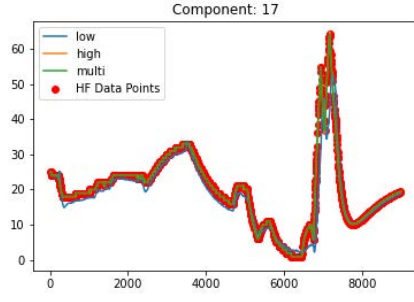


MF - Component 16

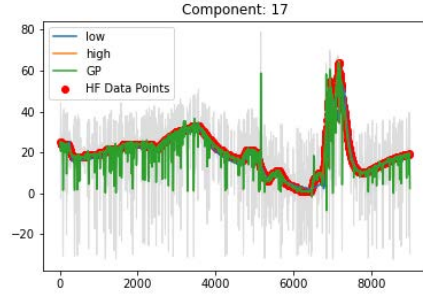


GP - Component 16

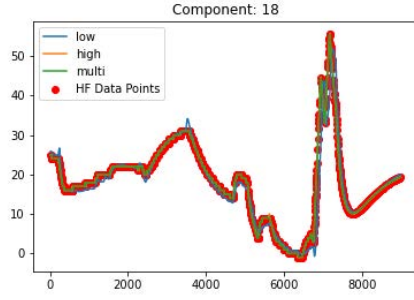
Figure A.90: SDS-4 data using the entire dataset. Multifidelity vs Gaussian process. Random high fidelity data points. Blue lines (-) indicate the LFM, orange lines (-) the HFM, green lines (-) the multifidelity estimation and red dots (•) the HFM data points used and the gray area (■) the confidence interval for components 13-16.



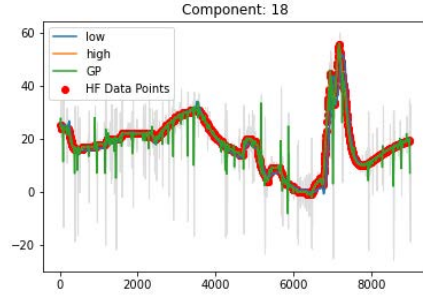
MF - Component 17



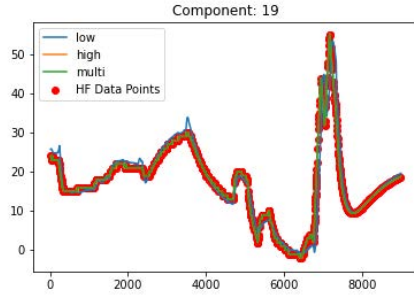
GP - Component 17



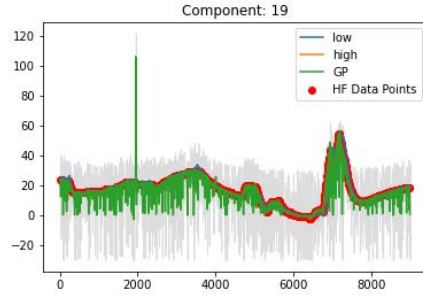
MF - Component 18



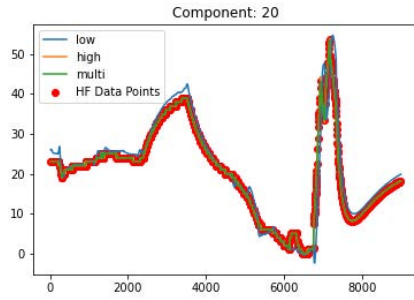
GP - Component 18



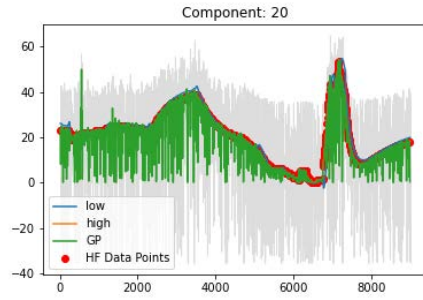
MF - Component 19



GP - Component 19

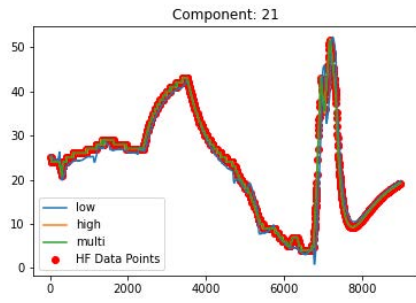


MF - Component 20

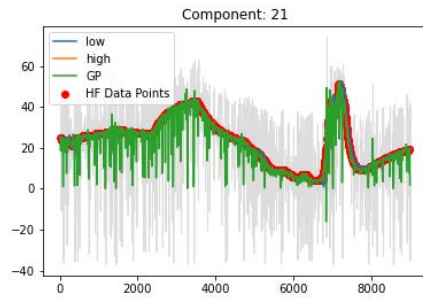


GP - Component 20

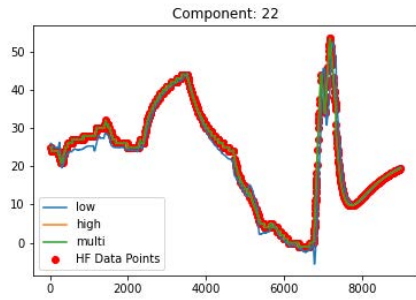
Figure A.91: SDS-4 data using the entire dataset. Multifidelity vs Gaussian process. Random high fidelity data points. Blue lines (-) indicate the LFM, orange lines (-) the HFM, green lines (-) the multifidelity estimation and red dots (•) the HFM data points used and the gray area (■) the confidence interval for components 17-20.



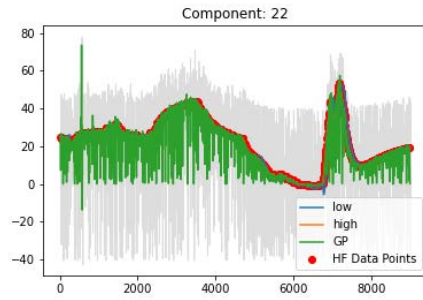
MF - Component 21



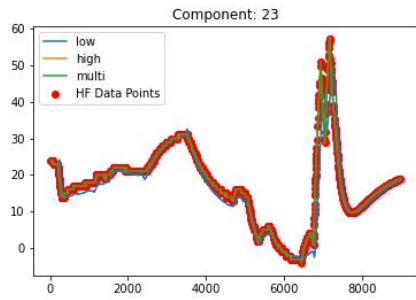
GP - Component 21



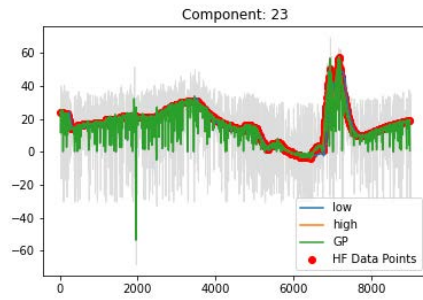
MF - Component 22



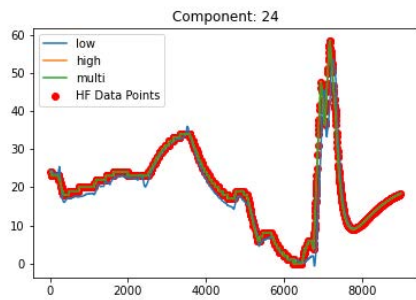
GP - Component 22



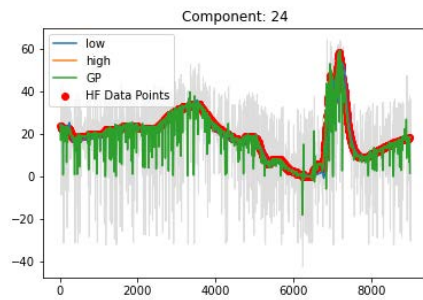
MF - Component 23



GP - Component 23

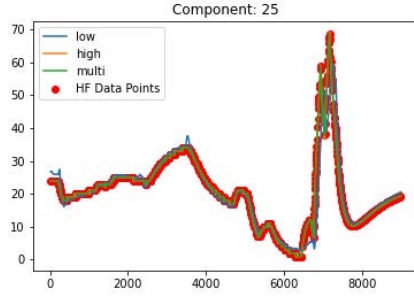


MF - Component 24

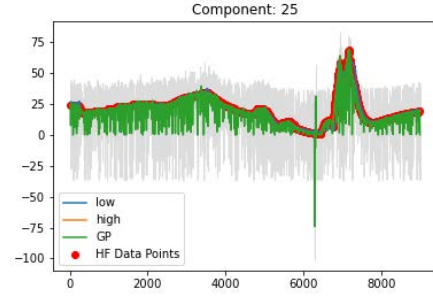


GP - Component 24

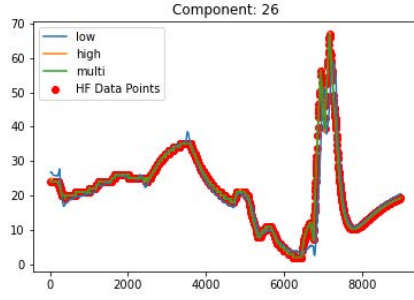
Figure A.92: SDS-4 data using the entire dataset. Multifidelity vs Gaussian process. Random high fidelity data points. Blue lines (-) indicate the LFM, orange lines (-) the HFM, green lines (-) the multifidelity estimation and red dots (•) the HFM data points used and the gray area (■) the confidence interval for components 21-24.



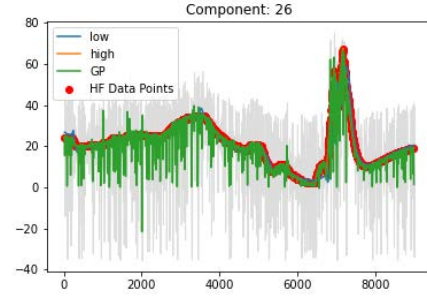
MF - Component 25



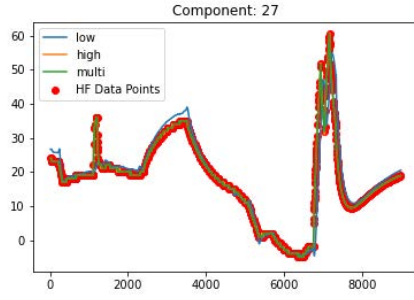
GP - Component 25



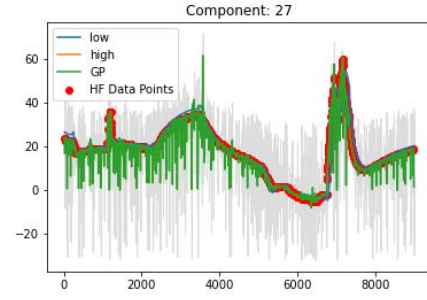
MF - Component 26



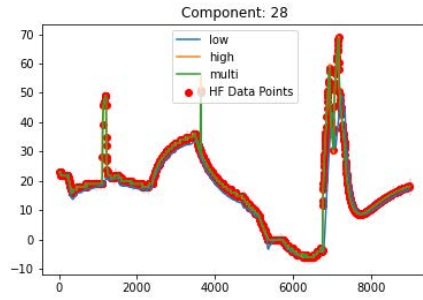
GP - Component 26



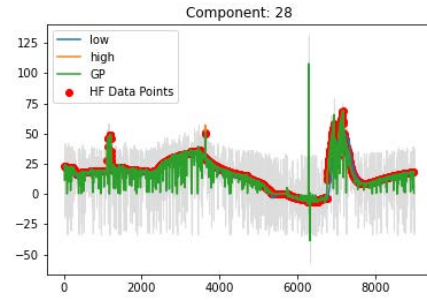
MF - Component 27



GP - Component 27

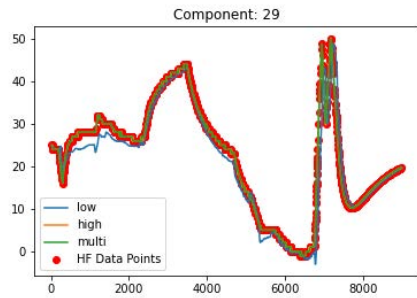


MF - Component 28

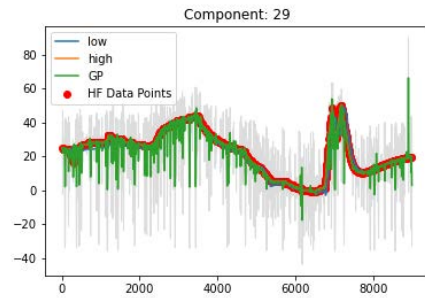


GP - Component 28

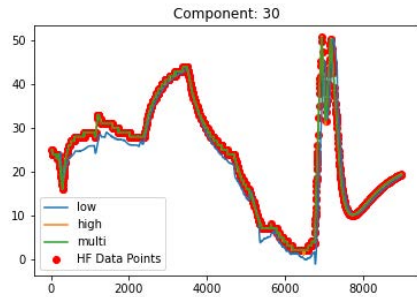
Figure A.93: SDS-4 data using the entire dataset. Multifidelity vs Gaussian process. Random high fidelity data points. Blue lines (-) indicate the LFM, orange lines (-) the HFM, green lines (-) the multifidelity estimation and red dots (•) the HFM data points used and the gray area (■) the confidence interval for components 25-28.



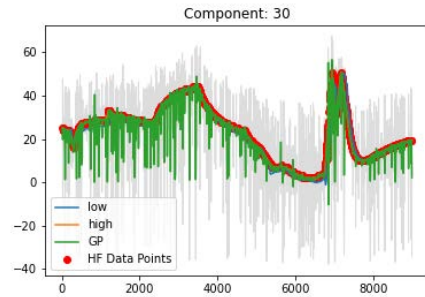
MF - Component 29



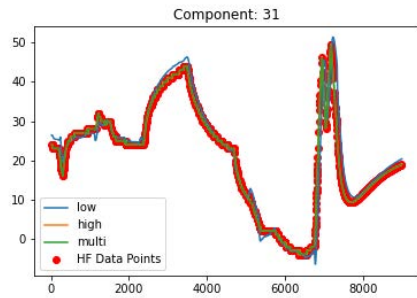
GP - Component 29



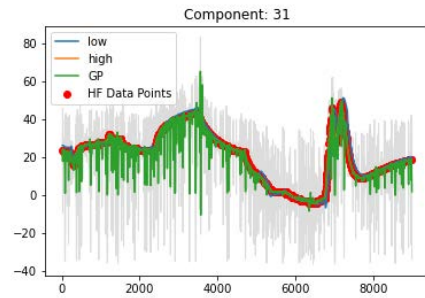
MF - Component 30



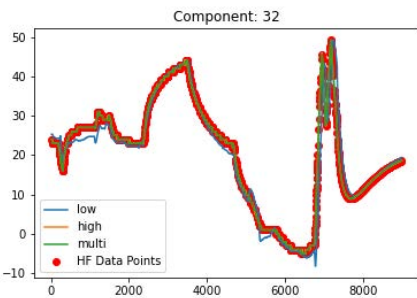
GP - Component 30



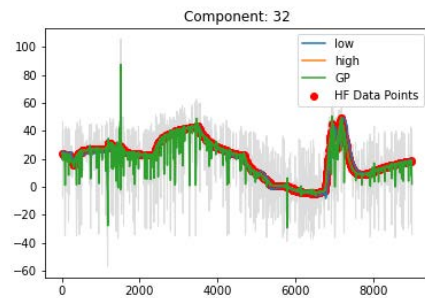
MF - Component 31



GP - Component 31

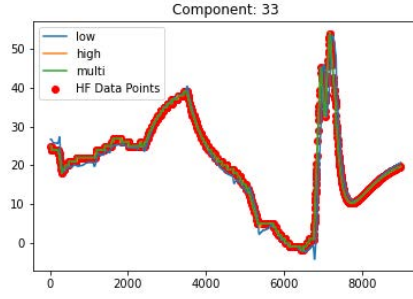


MF - Component 32

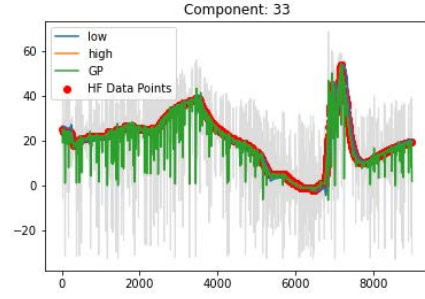


GP - Component 32

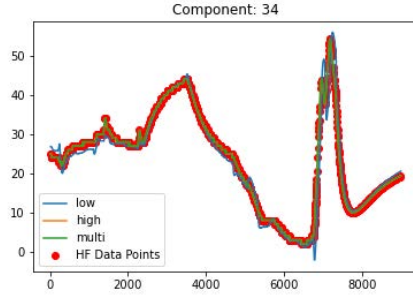
Figure A.94: SDS-4 data using the entire dataset. Multifidelity vs Gaussian process. Random high fidelity data points. Blue lines (-) indicate the LFM, orange lines (-) the HFM, green lines (-) the multifidelity estimation and red dots (•) the HFM data points used and the gray area (■) the confidence interval for components 29-32.



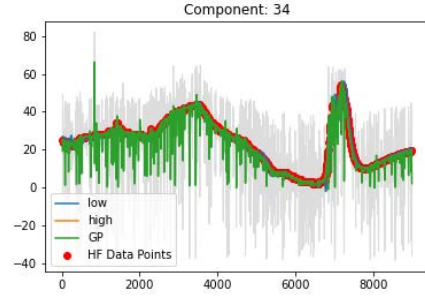
MF - Component 33



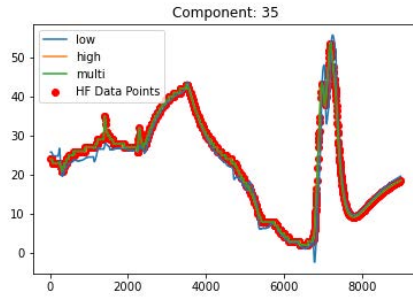
GP - Component 33



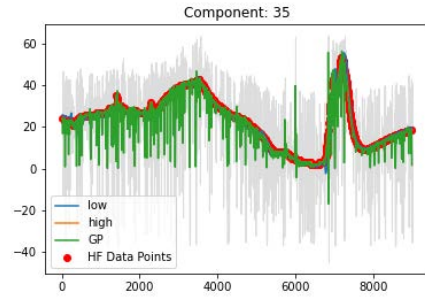
MF - Component 34



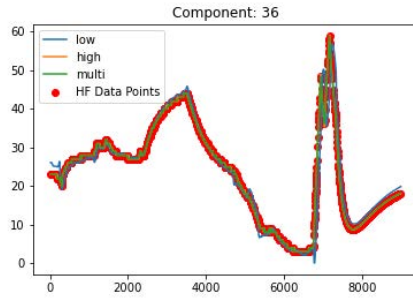
GP - Component 34



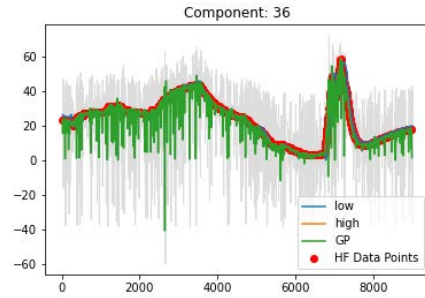
MF - Component 35



GP - Component 35

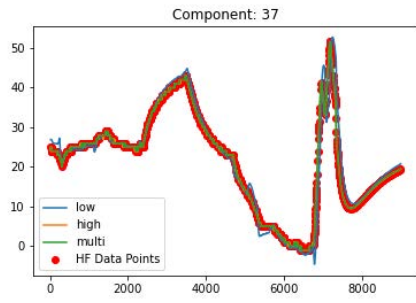


MF - Component 36

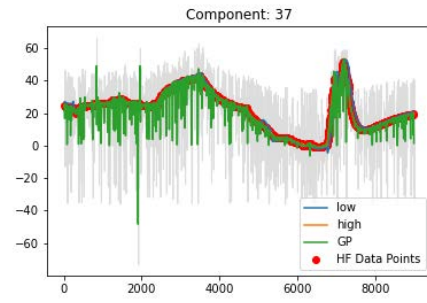


GP - Component 36

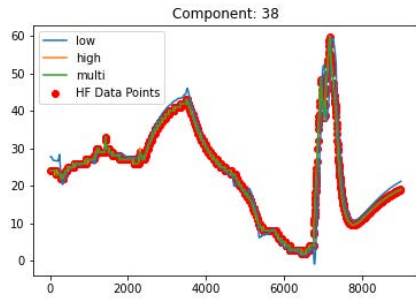
Figure A.95: SDS-4 data using the entire dataset. Multifidelity vs Gaussian process. Random high fidelity data points. Blue lines (-) indicate the LFM, orange lines (-) the HFM, green lines (-) the multifidelity estimation and red dots (•) the HFM data points used and the gray area (■) the confidence interval for components 33-36.



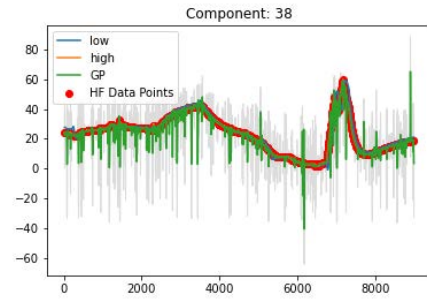
MF - Component 37



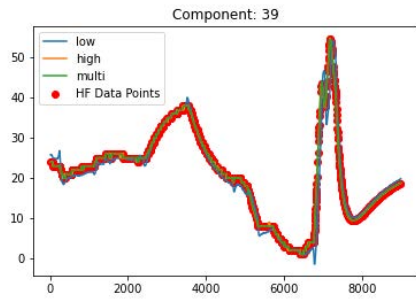
GP - Component 37



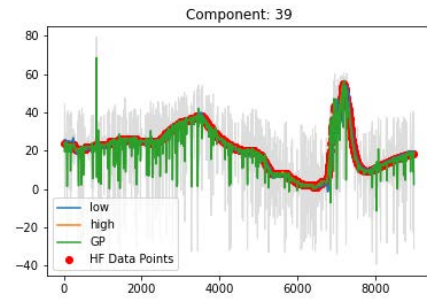
MF - Component 38



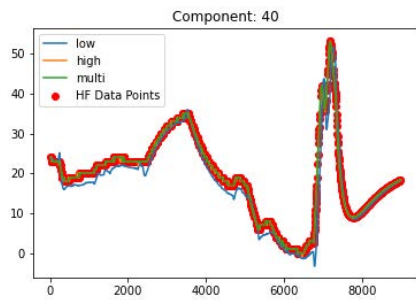
GP - Component 38



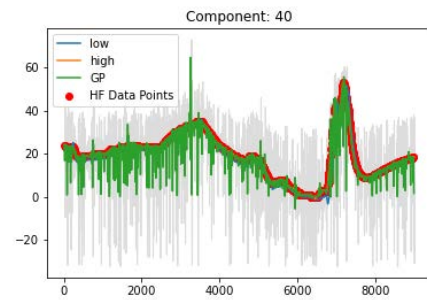
MF - Component 39



GP - Component 39

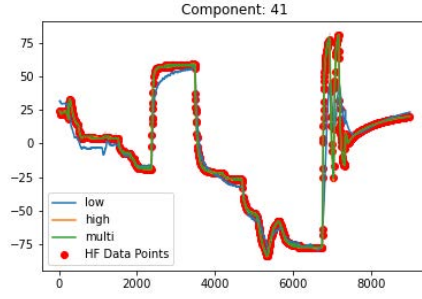


MF - Component 40

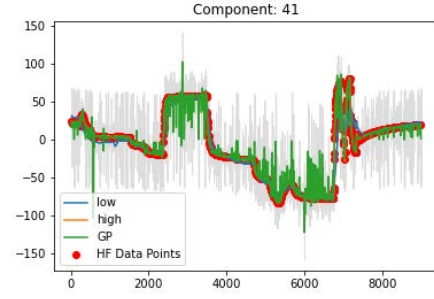


GP - Component 40

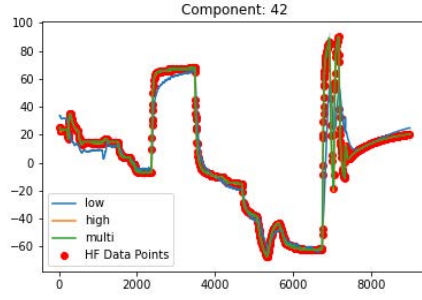
Figure A.96: SDS-4 data using the entire dataset. Multifidelity vs Gaussian process. Random high fidelity data points. Blue lines (-) indicate the LFM, orange lines (-) the HFM, green lines (-) the multifidelity estimation and red dots (•) the HFM data points used and the gray area (■) the confidence interval for components 37-40.



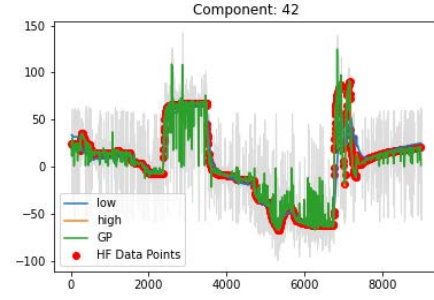
MF - Component 41



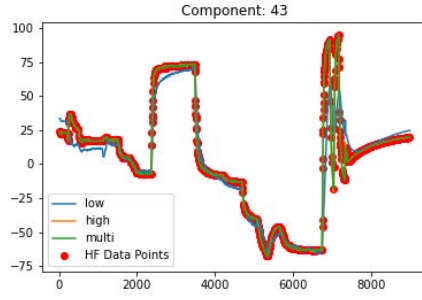
GP - Component 41



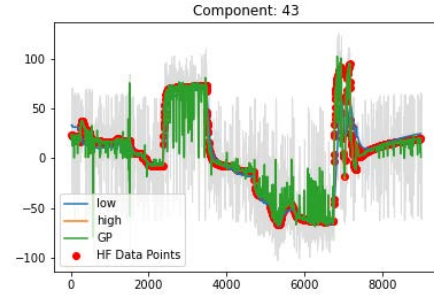
MF - Component 42



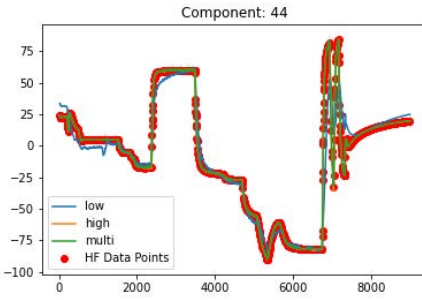
GP - Component 42



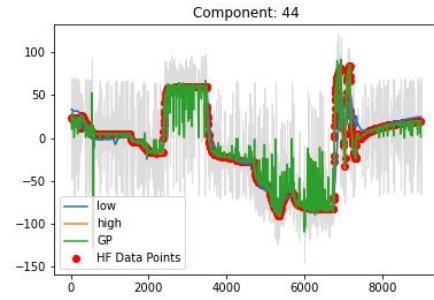
MF - Component 43



GP - Component 43

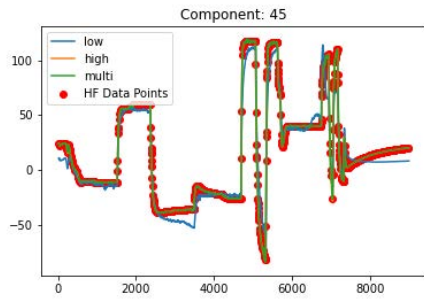


MF - Component 44

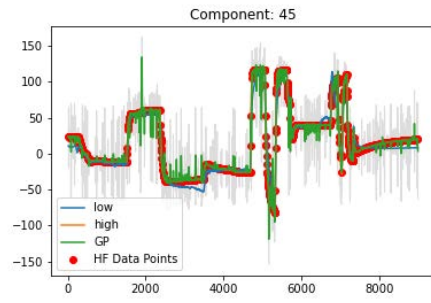


GP - Component 44

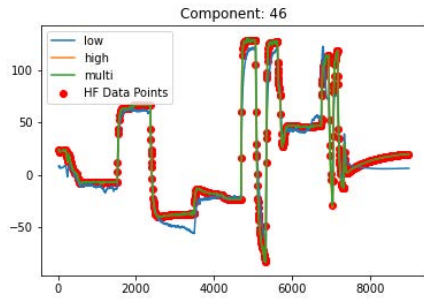
Figure A.97: SDS-4 data using the entire dataset. Multifidelity vs Gaussian process. Random high fidelity data points. Blue lines (-) indicate the LFM, orange lines (-) the HFM, green lines (-) the multifidelity estimation and red dots (•) the HFM data points used and the gray area (■) the confidence interval for components 41-44.



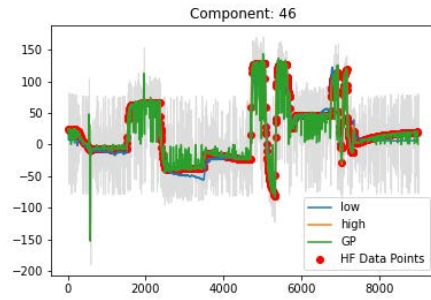
MF - Component 45



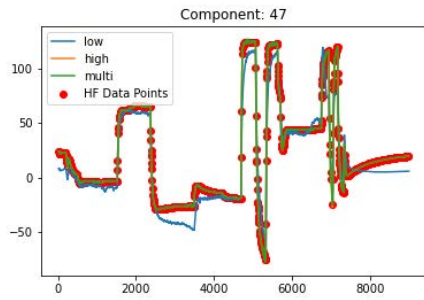
GP - Component 45



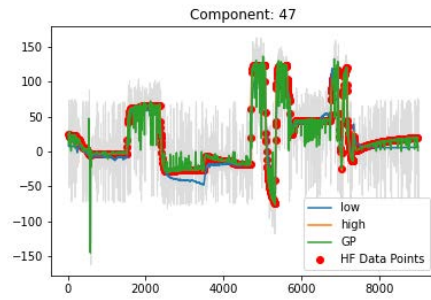
MF - Component 46



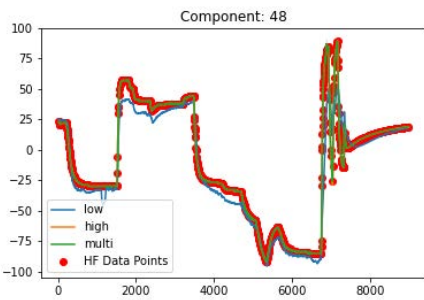
GP - Component 46



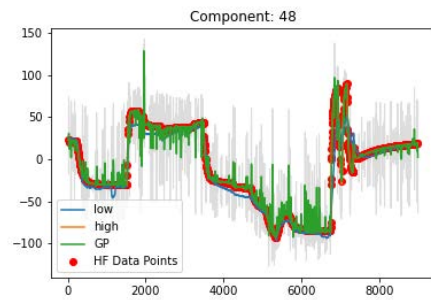
MF - Component 47



GP - Component 47

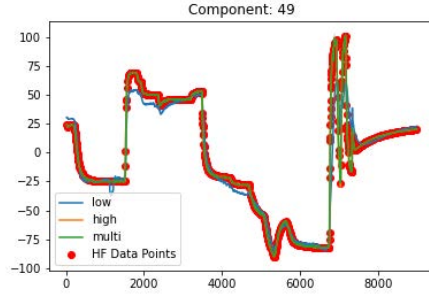


MF - Component 48

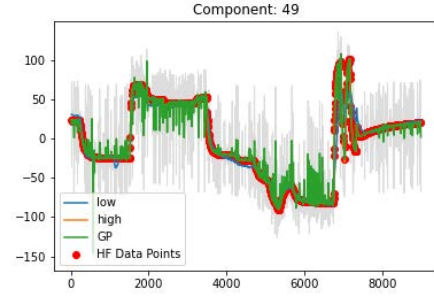


GP - Component 48

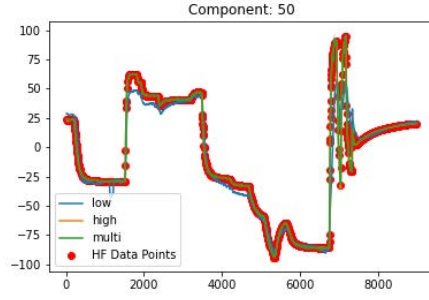
Figure A.98: SDS-4 data using the entire dataset. Multifidelity vs Gaussian process. Random high fidelity data points. Blue lines (-) indicate the LFM, orange lines (-) the HFM, green lines (-) the multifidelity estimation and red dots (•) the HFM data points used and the gray area (■) the confidence interval for components 45-48.



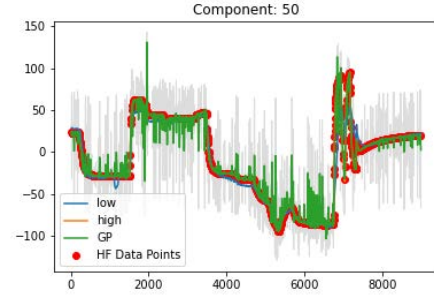
MF - Component 49



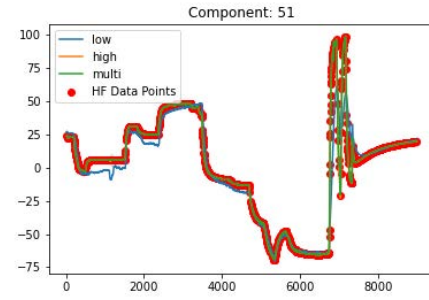
GP - Component 49



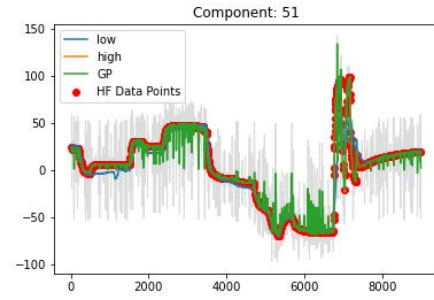
MF - Component 50



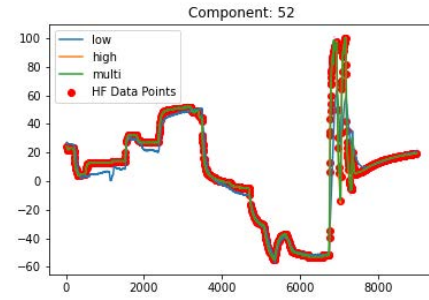
GP - Component 50



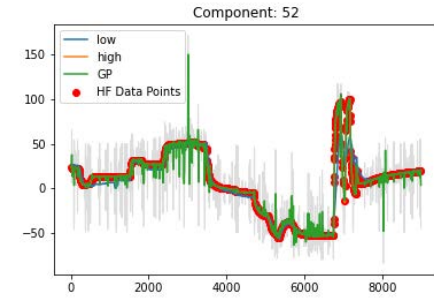
MF - Component 51



GP - Component 51

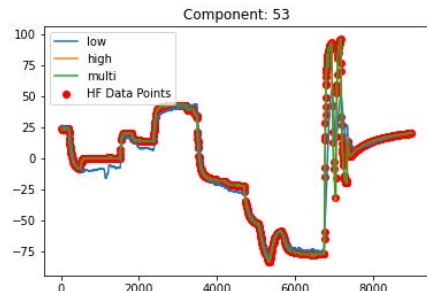


MF - Component 52

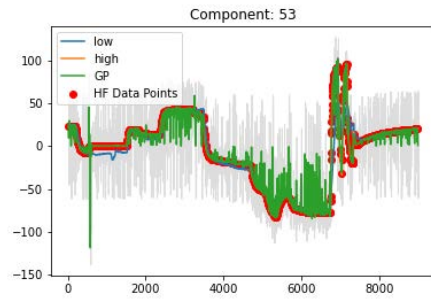


GP - Component 52

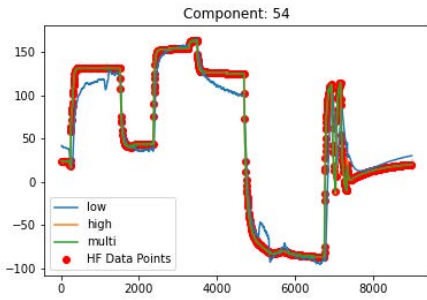
Figure A.99: SDS-4 data using the entire dataset. Multifidelity vs Gaussian process. Random high fidelity data points. Blue lines (-) indicate the LFM, orange lines (-) the HFM, green lines (-) the multifidelity estimation and red dots (•) the HFM data points used and the gray area (■) the confidence interval for components 49-52.



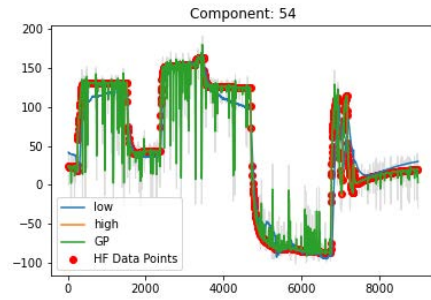
MF - Component 53



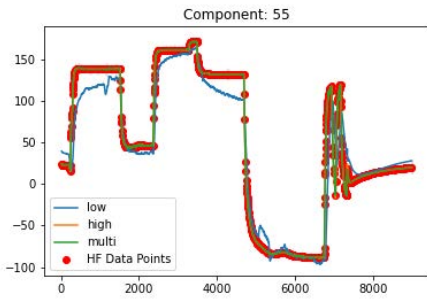
GP - Component 53



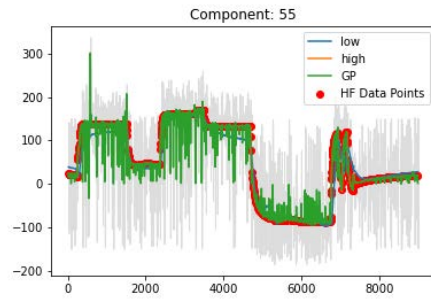
MF - Component 54



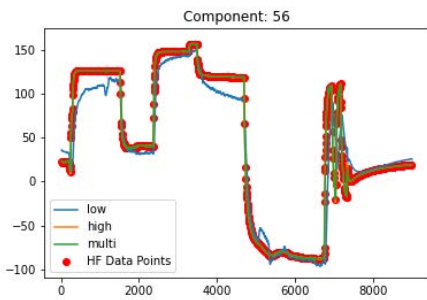
GP - Component 54



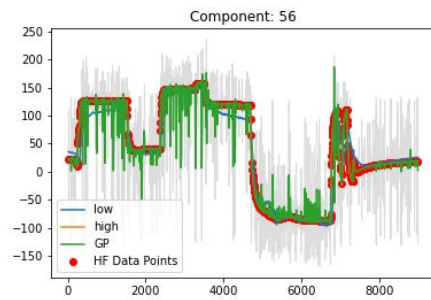
MF - Component 55



GP - Component 55

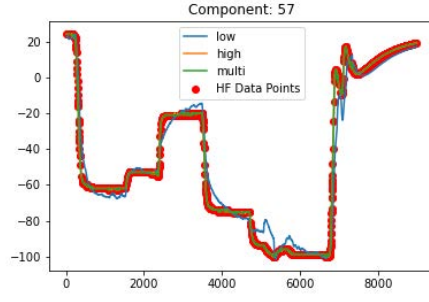


MF - Component 56

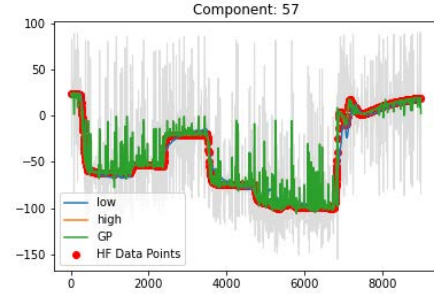


GP - Component 56

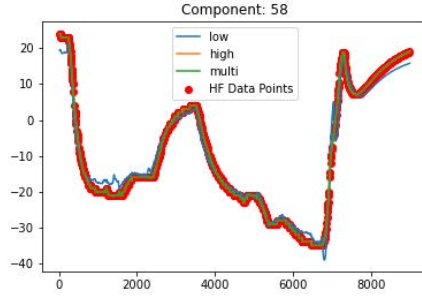
Figure A.100: SDS-4 data using the entire dataset. Multifidelity vs Gaussian process. Random high fidelity data points. Blue lines (-) indicate the LFM, orange lines (-) the HFM, green lines (-) the multifidelity estimation and red dots (•) the HFM data points used and the gray area (■) the confidence interval for components 53-56.



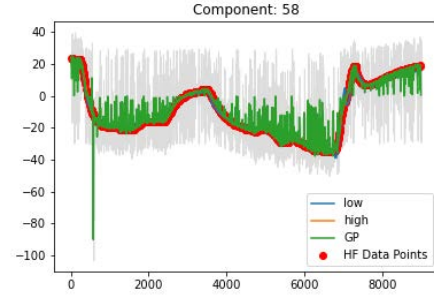
MF - Component 57



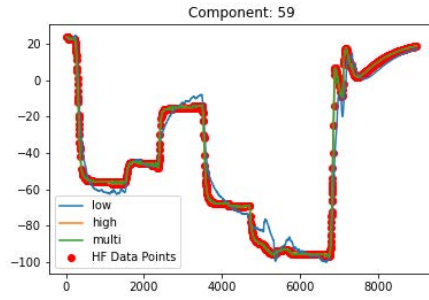
GP - Component 57



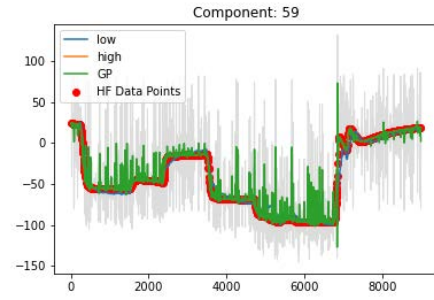
MF - Component 58



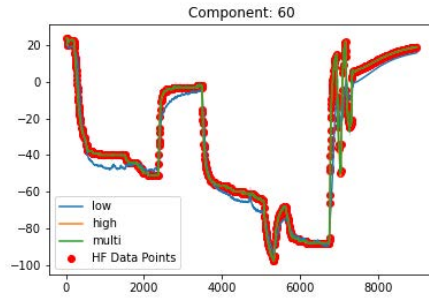
GP - Component 58



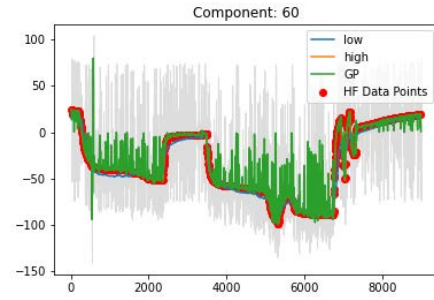
MF - Component 59



GP - Component 59



MF - Component 60



GP - Component 60

Figure A.101: SDS-4 data using the entire dataset. Multifidelity vs Gaussian process. Random high fidelity data points. Blue lines (-) indicate the LFM, orange lines (-) the HFM, green lines (-) the multifidelity estimation and red dots (•) the HFM data points used and the gray area (■) the confidence interval for components 57-60.

References

- [1] Anastasios Kontaxoglou, Seiji Tsutsumi, Samir Khan, and Shinichi Nakasuka. Towards a digital twin enabled multifidelity framework for small satellites. In *PHM Society European Conference*, volume 6, pages 10–10, 2021.
- [2] Anastasios Kontaxoglou, Seiji Tsutsumi, Samir Khan, Toshihiro Shibukawa, and Shinichi Nakasuka. Use of cokriging for thermal analysis in small satellites. *2022 IEEE Aerospace Conference*, 2022.
- [3] Martin Sweeting. Modern small satellites-changing the economics of space. *Proceedings of the IEEE*, 106(3):343–361, 2018.
- [4] NSR. Global satellite manufacturing and launch markets, 6th edition report (gsmlm11). 2019.
- [5] Benjamin Peherstorfer, Karen Willcox, and Max Gunzburger. Survey of multifidelity methods in uncertainty propagation, inference, and optimization. *SIAM Review*, 60(3):550–591, 2018. ISSN 00361445. doi: 10.1137/16M1082469.
- [6] Andy J. Keane. Cokriging for robust design optimization. *AIAA Journal*, 50(11):2351–2364, 2012. ISSN 00011452. doi: 10.2514/1.J051391.
- [7] Chanyoung Park, Raphael T. Haftka, and Nam H. Kim. Remarks on multifidelity surrogates. *Structural and Multidisciplinary Optimization*, 55(3): 1029–1050, 2017. ISSN 16151488. doi: 10.1007/s00158-016-1550-y.
- [8] Andrew March, Karen Willcox, and Qiqi Wang. Gradient-based multifidelity optimisation for aircraft design using bayesian model calibration. *The Aeronautical Journal*, 115(1174):729–738, 2011.
- [9] Raghu Pasupathy, Bruce W Schmeiser, Michael R Taaffe, and Jin Wang. Control-variate estimation using estimated control means. *Iie Transactions*, 44(5):381–385, 2012.
- [10] Benjamin Peherstorfer, Tiangang Cui, Youssef Marzouk, and Karen Willcox. Multifidelity importance sampling. *Computer Methods in Applied Mechanics and Engineering*, 300:490–509, 2016.
- [11] Benjamin Peherstorfer and Karen Willcox. Online adaptive model reduction for nonlinear systems via low-rank updates. *SIAM Journal on Scientific Computing*, 37(4):A2123–A2150, 2015.
- [12] Rasmussen CE& Williams CKI. Gaussian processes for machine learning. *International Journal of Neural Systems*, 14, 2006.

- [13] Carl Edward Rasmussen. Gaussian processes in machine learning. In *Summer school on machine learning*, pages 63–71. Springer, 2003.
- [14] D. Duvenaud. The kernel cookbook. URL <https://www.cs.toronto.edu/~duvenaud/cookbook/>.
- [15] Eric Schulz, Maarten Speekenbrink, and Andreas Krause. A tutorial on gaussian process regression: Modelling, exploring, and exploiting functions. *Journal of Mathematical Psychology*, 85:1–16, 2018. ISSN 0022-2496. doi: <https://doi.org/10.1016/j.jmp.2018.03.001>.
- [16] Jie Wang. An intuitive tutorial to gaussian processes regression. *arXiv preprint arXiv:2009.10862*, 2020.
- [17] Mark Gibbs and David JC MacKay. Efficient implementation of gaussian processes. 1997.
- [18] Carl T Kelley. *Iterative methods for linear and nonlinear equations*. SIAM, 1995.
- [19] Vikas C Raykar and Ramani Duraiswami. Fast large scale gaussian process regression using approximate matrix-vector products. In *Learning workshop*, 2007.
- [20] Donald R. Jones. A Taxonomy of Global Optimization Methods Based on Response Surfaces. *Journal of Global Optimization*, 21(4):345–383, 2001. ISSN 09255001. doi: 10.1023/A:1012771025575.
- [21] D. J.J. Toal, N. W. Bressloff, A. J. Keane, and C. M.E. Holden. The development of a hybridized particle swarm for kriging hyperparameter tuning. *Engineering Optimization*, 43(6):675–699, 2011. ISSN 0305215X. doi: 10.1080/0305215X.2010.508524.
- [22] Mike B. Giles. Collected matrix derivative results for forward and reverse mode algorithmic differentiation. *Lecture Notes in Computational Science and Engineering*, 64 LNCSE:35–44, 2008. ISSN 14397358. doi: 10.1007/978-3-540-68942-3{_}4.
- [23] Mingkai Qu, Jian Chen, Biao Huang, and Yongcun Zhao. Resampling with in situ field portable x-ray fluorescence spectrometry (fpxrf) to reduce the uncertainty in delineating the remediation area of soil heavy metals. *Environmental Pollution*, 271:116310, 2021.
- [24] A Rodríguez-Lizana, MJ Pereira, M Castro Ribeiro, A Soares, L Azevedo, A Miranda-Fuentes, and J Llorens. Spatially variable pesticide application in olive groves: Evaluation of potential pesticide-savings through stochastic spatial simulation algorithms. *Science of The Total Environment*, 778: 146111, 2021.
- [25] M. C. Kennedy and A. O’Hagan. Predicting the output from a complex computer code when fast approximations are available. *Biometrika*, 87(1): 1–13, 2000. ISSN 00063444. doi: 10.1093/biomet/87.1.1.
- [26] Marc C Kennedy and Anthony O’Hagan. Bayesian calibration of computer models. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 63(3):425–464, 2001.

- [27] Neil D. Lawrence. Multifidelity modelling, 2021. URL https://colab.research.google.com/github/mlatcl/mlphysical/blob/gh-pages/_notebooks/05-02-multifidelity.ipynb#scrollTo=aNMam_u0qhCQ.
- [28] Loic Le Gratiet and Claire Cannamela. Cokriging-based sequential design strategies using fast cross-validation techniques for multi-fidelity computer codes. *Technometrics*, 57(3):418–427, 2015.
- [29] Dave Higdon, Marc Kennedy, James C Cavendish, John A Cafeo, and Robert D Ryne. Combining field data and computer simulations for calibration and prediction. *SIAM Journal on Scientific Computing*, 26(2):448–466, 2004.
- [30] P. Perdikaris, D. Venturi, J. O. Royset, and G. E. Karniadakis. Multifidelity modelling via recursive co-kriging and Gaussian-Markov random fields. *Proceedings of the Royal Society A: Mathematical, Physical and Engineering Sciences*, 471(2179), 2015. ISSN 14712946. doi: 10.1098/rspa.2015.0018.
- [31] Peter Z.G. Qian and C. F. Jeff Wu. Bayesian hierarchical modeling for integrating low-accuracy and high-accuracy experiments. *Technometrics*, 50(2):192–204, 2008. ISSN 00401706. doi: 10.1198/004017008000000082.
- [32] Alexander IJ Forrester, András Sóbester, and Andy J Keane. Multi-fidelity optimization via surrogate modelling. *Proceedings of the royal society a: mathematical, physical and engineering sciences*, 463(2088):3251–3269, 2007.
- [33] Loic Le Gratiet and Josselin Garnier. Recursive co-kriging model for design of computer experiments with multiple levels of fidelity. *International Journal for Uncertainty Quantification*, 4(5):365–386, 2014. ISSN 21525099.
- [34] David J.J. Toal. Some considerations regarding the use of multi-fidelity Kriging in the construction of surrogate models. *Structural and Multidisciplinary Optimization*, 51(6):1223–1245, 2015. ISSN 16151488. doi: 10.1007/s00158-014-1209-5.
- [35] Yiming Zhang, Nam H Kim, Chanyoung Park, and Raphael T Haftka. Multifidelity surrogate based on single linear regression. *AIAA Journal*, 56(12):4944–4952, 2018.
- [36] Jerome Sacks, William J Welch, Toby J Mitchell, and Henry P Wynn. Design and analysis of computer experiments. *Statistical science*, 4(4):409–423, 1989.
- [37] Xiu Yang, David Barajas-Solano, Guzel Tartakovsky, and Alexandre M Tartakovsky. Physics-informed cokriging: A gaussian-process-regression-based multifidelity method for data-model convergence. *Journal of Computational Physics*, 395:410–431, 2019.
- [38] Qi Zhou, Yuda Wu, Zhendong Guo, Jiexiang Hu, and Peng Jin. A generalized hierarchical co-kriging model for multi-fidelity data fusion. *Structural and Multidisciplinary Optimization*, 62:1885–1904, 2020.
- [39] Andrew S Thelen, Leifur T Leifsson, and Philip S Beran. Multifidelity flutter prediction using regression cokriging with adaptive sampling. *Journal of Fluids and Structures*, 97:103081, 2020.

- [40] George E P Box, Gwilym M Jenkins, Gregory C Reinsel, and Greta M Ljung. *Time series analysis: forecasting and control*. John Wiley & Sons, 2015.
- [41] S. L. Ho, M. Xie, and T. N. Goh. A comparative study of neural network and Box-Jenkins ARIMA modeling in time series prediction. *Computers and Industrial Engineering*, 42(2-4):371–375, 2002. ISSN 03608352. doi: 10.1016/S0360-8352(02)00036-0.
- [42] Priyamvada and Rajesh Wadhvani. Review on various models for time series forecasting. *Proceedings of the International Conference on Inventive Computing and Informatics, ICICI 2017*, (Icici):405–410, 2018. doi: 10.1109/ICICI.2017.8365383.
- [43] Alan Pankratz. *Forecasting with dynamic regression models*, volume 935. John Wiley & Sons, 2012.
- [44] Q. L. Jing, Q. Cheng, J. M. Marshall, W. B. Hu, Z. C. Yang, and J. H. Lu. Imported cases and minimum temperature drive dengue transmission in Guangzhou, China: evidence from ARIMAX model. *Epidemiology and Infection*, 146(10):1226–1235, 2018. ISSN 14694409. doi: 10.1017/S0950268818001176.
- [45] Wiwik Anggraeni, Kuntoro Boga Andri, Sumaryanto, and Faizal Mahananto. The Performance of ARIMAX Model and Vector Autoregressive (VAR) Model in Forecasting Strategic Commodity Price in Indonesia. *Procedia Computer Science*, 124:189–196, 2017. ISSN 18770509. doi: 10.1016/j.procs.2017.12.146.
- [46] Billy M. Williams. Multivariate vehicular traffic flow prediction: Evaluation of ARIMAX modeling. *Transportation Research Record*, (1776):194–200, 2001. ISSN 03611981. doi: 10.3141/1776-25.
- [47] Amir Jalalkamali, Mehdi Moradi, and Nasrin Moradi. Application of several artificial intelligence models and arimax model for forecasting drought using the standardized precipitation index. *International journal of environmental science and technology*, 12(4):1201–1210, 2015.
- [48] Ming-Tsung Lee. *Short-term Freeway Traffic Flow Forecasting with ARIMAX Modeling*. PhD thesis, Dissertation, Braunschweig, Technische Universität Braunschweig, 2010.
- [49] Anastasios Kontaxoglou. Uncalibrated robot visual servoing(in greek, 2017. URL <http://artemis.cslab.ece.ntua.gr:8080/jspui/handle/123456789/13498>.
- [50] Mohamad H Hassoun et al. *Fundamentals of artificial neural networks*. MIT press, 1995.
- [51] Steven W Smith, Static Linearity, Sinusoidal Fidelity, Common Decompositions, Polar Notation, Polar Nuisances, Band-Pass High-Pass, FFT Convolution, Step Response Overshoot, Human Hearing, et al. The scientist and engineer’s guide to digital signal processing by steven w. smith, ph. d.
- [52] Christopher Olah. Understanding lstm networks. 2015.

- [53] Yoshua Bengio, Patrice Simard, and Paolo Frasconi. Learning long-term dependencies with gradient descent is difficult. *IEEE transactions on neural networks*, 5(2):157–166, 1994.
- [54] Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural computation*, 9(8):1735–1780, 1997.
- [55] Zheng Zhao, Weihai Chen, Xingming Wu, Peter C.V. Chen, and Jingmeng Liu. LSTM network: A deep learning approach for short-term traffic forecast. *IET Image Processing*, 11(1):68–75, 2017. ISSN 17519659. doi: 10.1049/iet-its.2016.0208.
- [56] Naveena Reddy Mettu and T. Sasikala. *Prediction Analysis on Web Traffic Data Using Time Series Modeling, RNN and Ensembling Techniques*, volume 26. Springer International Publishing, 2019. ISBN 9783030031466. doi: 10.1007/978-3-030-03146-6_{-}67.
- [57] Elisa Mussumeci and Flávio Codeço Coelho. Large-scale multivariate forecasting models for Dengue - LSTM versus random forest regression. *Spatial and Spatio-temporal Epidemiology*, 35:100372, 2020. ISSN 18775853. doi: 10.1016/j.sste.2020.100372.
- [58] Sima Siami-Namini, Neda Tavakoli, and Akbar Siami Namin. A Comparison of ARIMA and LSTM in Forecasting Time Series. pages 1394–1401. IEEE, 2019. ISBN 9781538668047. doi: 10.1109/ICMLA.2018.00227.
- [59] Zhong Qi Li, Hong Qiu Pan, Qiao Liu, Huan Song, and Jian Ming Wang. Comparing the performance of time series models with or without meteorological factors in predicting incident pulmonary tuberculosis in eastern China. *Infectious Diseases of Poverty*, 9(1):1–11, 2020. ISSN 20499957. doi: 10.1186/s40249-020-00771-7.
- [60] Giulia Serafini, Ping Yi, Qingquan Zhang, Marco Brambilla, Jiayue Wang, Yiwei Hu, and Beibei Li. Sentiment-Driven Price Prediction of the Bitcoin based on Statistical and Deep Learning Approaches. *Proceedings of the International Joint Conference on Neural Networks*, 2020. doi: 10.1109/IJCNN48605.2020.9206704.
- [61] Kyunghyun Cho, Bart Van Merriënboer, Caglar Gulcehre, Dzmitry Bahdanau, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. Learning phrase representations using rnn encoder-decoder for statistical machine translation. *arXiv preprint arXiv:1406.1078*, 2014.
- [62] Peter T Yamak, Li Yujian, and Pius K Gadosey. A comparison between arima, lstm, and gru for time series forecasting. In *Proceedings of the 2019 2nd International Conference on Algorithms, Computing and Artificial Intelligence*, pages 49–55, 2019.
- [63] Xin Wei, Lulu Zhang, Hao-Qing Yang, Limin Zhang, and Yang-Ping Yao. Machine learning for pore-water pressure time-series prediction: application of recurrent neural networks. *Geoscience Frontiers*, 12(1):453–467, 2021.
- [64] Li-Pang Chen. Mehryar mohri, afshin rostamizadeh, and ameer talwalkar: Foundations of machine learning, 2019.
- [65] Sudeep Raja. Fnnns, rnnns, lstm and blstm, 2021.

- [66] T. Akita, R. Takaki, and E. Shima. A new adaptive estimation method of spacecraft thermal mathematical model with an ensemble Kalman filter. *Acta Astronautica*, 73:144–155, 2012. ISSN 00945765. doi: 10.1016/j.actaastro.2011.11.003.
- [67] Haoran Ji and Dongxu Li. A novel nonlinear finite element method for structural dynamic modeling of spacecraft under large deformation. *Thin-Walled Structures*, 165:107926, 2021. ISSN 0263-8231. doi: <https://doi.org/10.1016/j.tws.2021.107926>.
- [68] Nam H Kim, Bhavani V Sankar, and Ashok V Kumar. *Introduction to finite element analysis and design*. John Wiley & Sons, 2018.
- [69] Aertia Software. Thermal desktop de c and r technologies. URL <http://www.aertia.com/productos.asp?pid=190>.
- [70] M. Mohanraj, S. Jayaraj, and C. Muraleedharan. Applications of artificial neural networks for thermal analysis of heat exchangers – a review. *International Journal of Thermal Sciences*, 90:150–172, 2015. ISSN 1290-0729. doi: <https://doi.org/10.1016/j.ijthermalsci.2014.11.030>.
- [71] Gulrukh Turabee, Muhammad Raza Khowja, Paolo Giangrande, Vincenzo Madonna, Georgina Cosma, Gaurang Vakil, Chris Gerada, and Michael Galea. The role of neural networks in predicting the thermal life of electrical machines. *IEEE Access*, 8:40283–40297, 2020. doi: 10.1109/ACCESS.2020.2975985.
- [72] Maximilian Winter and Christian Breitsamter. Nonlinear identification via connected neural networks for unsteady aerodynamic analysis. *Aerospace Science and Technology*, 77:802–818, 2018.
- [73] JD Reis Junior, AM Ambrosio, FL de Sousa, and DF Silva. Spacecraft real-time thermal simulation using artificial neural networks. *Journal of the Brazilian Society of Mechanical Sciences and Engineering*, 43(4):1–17, 2021.
- [74] Massimo Tipaldi, Lorenzo Feruglio, Pierre Denis, and Gianni D’Angelo. On applying ai-driven flight data analysis for operational spacecraft model-based diagnostics. *Annual Reviews in Control*, 49:197–211, 2020.
- [75] Renhe Shi, Teng Long, and Hexi Baoyin. Multi-fidelity and multi-objective optimization of low-thrust transfers with control strategy for all-electric geostationary satellites. *Acta Astronautica*, 177:577–587, 2020.
- [76] Renhe Shi, Li Liu, Teng Long, Yufei Wu, and G. Gary Wang. Multi-Fidelity Modeling and Adaptive Co-Kriging-Based Optimization for All-Electric Geostationary Orbit Satellite Systems. *Journal of Mechanical Design*, 142(2):1–13, 2020. ISSN 1050-0472. doi: 10.1115/1.4044321.
- [77] Satoshi Ueda and Hideaki Ogawa. Multi-fidelity approach for global trajectory optimization using gpu-based highly parallel architecture. *Aerospace Science and Technology*, page 106829, 2021.
- [78] E Minisci, M Vasile, and H Liqiang. Robust multi-fidelity design of a micro re-entry unmanned space vehicle. *Proceedings of the Institution of Mechanical Engineers, Part G: Journal of Aerospace Engineering*, 225(11):1195–1209, 2011.

- [79] Edmondo Minisci and Massimiliano Vasile. Robust design of a reentry unmanned space vehicle by multifidelity evolution control. *AIAA journal*, 51(6):1284–1295, 2013.
- [80] Anirban Chaudhuri, Remi Lam, and Karen Willcox. Multifidelity uncertainty propagation via adaptive surrogates in coupled multidisciplinary systems. *AIAA Journal*, 56(1):235–249, 2018.
- [81] Mario J Santos, Serhat Hosder, and Thomas K West. Multifidelity turbulent heating prediction of hypersonic inflatable aerodynamic decelerators with surface scalloping. *Journal of Spacecraft and Rockets*, pages 1–14, 2021.
- [82] CK Lo, CH Chen, and Ray Y Zhong. A review of digital twin in product design and development. *Advanced Engineering Informatics*, 48:101297, 2021.
- [83] Rakesh Kumar Phanden, Priavrat Sharma, and Anubhav Dubey. A review on simulation in digital twin for aerospace, manufacturing and robotics. *Materials today: proceedings*, 38:174–178, 2021.
- [84] Tianhu Deng, Keren Zhang, and Zuo-Jun Max Shen. A systematic review of a digital twin city: A new pattern of urban governance toward smart cities. *Journal of Management Science and Engineering*, 2021.
- [85] De-Graft Joe Opoku, Srinath Perera, Robert Osei-Kyei, and Maria Rashidi. Digital twin application in the construction industry: A literature review. *Journal of Building Engineering*, page 102726, 2021.
- [86] Concetta Semeraro, Mario Lezoche, Hervé Panetto, and Michele Dassisti. Digital twin paradigm: A systematic literature review. *Computers in Industry*, 130:103469, 2021.
- [87] Dionisis Andronas, George Kokotinis, and Sotiris Makris. On modelling and handling of flexible materials: A review on digital twins and planning systems. *Procedia CIRP*, 97:447–452, 2021.
- [88] Mengnan Liu, Shuiliang Fang, Huiyue Dong, and Cunzhi Xu. Review of digital twin about concepts, technologies, and industrial applications. *Journal of Manufacturing Systems*, 58:346–361, 2021.
- [89] M Mazhar Rathore, Syed Attique Shah, Dharendra Shukla, Elmahdi Bentafat, and Spiridon Bakiras. The role of ai, machine learning, and big data in digital twinning: A systematic literature review, challenges, and opportunities. *IEEE Access*, 9:32030–32052, 2021.
- [90] Ghanishtha Bhatti, Harshit Mohan, and R Raja Singh. Towards the future of smart electric vehicles: Digital twin technology. *Renewable and Sustainable Energy Reviews*, 141:110801, 2021.
- [91] Feng Jiang, Ling Ma, Tim Broyd, and Ke Chen. Digital twin and its implementations in the civil engineering sector. *Automation in Construction*, 130:103838, 2021. ISSN 0926-5805. doi: <https://doi.org/10.1016/j.autcon.2021.103838>.

- [92] Jiewu Leng, Dewen Wang, Weiming Shen, Xinyu Li, Qiang Liu, and Xin Chen. Digital twins-based smart manufacturing system design in industry 4.0: A review. *Journal of Manufacturing Systems*, 60:119–137, 2021. ISSN 0278-6125. doi: <https://doi.org/10.1016/j.jmsy.2021.05.011>.
- [93] David Jones, Chris Snider, Aydin Nassehi, Jason Yon, and Ben Hicks. Characterising the Digital Twin: A systematic literature review. *CIRP Journal of Manufacturing Science and Technology*, 29:36–52, 2020. ISSN 17555817. doi: [10.1016/j.cirpj.2020.02.002](https://doi.org/10.1016/j.cirpj.2020.02.002).
- [94] Adil Rasheed, Omer San, and Trond Kvamsdal. Digital twin: Values, challenges and enablers from a modeling perspective. *Ieee Access*, 8:21980–22012, 2020.
- [95] Itxaro Errandonea, Sergio Beltrán, and Saioa Arrizabalaga. Digital twin for maintenance: A literature review. *Computers in Industry*, 123:103316, 2020.
- [96] Ziyue Chen and Lizhen Huang. Digital twins for information-sharing in remanufacturing supply chain: A review. *Energy*, page 119712, 2020.
- [97] Yuqian Lu, Chao Liu, Kevin I-Kai Wang, Huiyue Huang, and Xun Xu. Digital twin-driven smart manufacturing: Connotation, reference model, applications and research issues. *Robotics and Computer-Integrated Manufacturing*, 61:101837, 2020. ISSN 0736-5845. doi: <https://doi.org/10.1016/j.rcim.2019.101837>.
- [98] Maggie Mashaly. Connecting the twins: A review on digital twin technology & its networking requirements. *Procedia Computer Science*, 184:299–305, 2021.
- [99] H. Yin Z and L. Wang. Application and development prospect of digital twin technology in aerospace. *IFAC-PapersOnLine*, 53(5):732–737, 2020. ISSN 2405-8963. doi: <https://doi.org/10.1016/j.ifacol.2021.04.165>. 3rd IFAC Workshop on Cyber-Physical and Human Systems CPHS 2020.
- [100] Tsega Y Melesse, Valentina Di Pasquale, and Stefano Riemma. Digital twin models in industrial operations: a systematic literature review. *Procedia Manufacturing*, 42:267–272, 2020.
- [101] Dmytro Adamenko, Steffen Kunnen, Robin Pluhnau, André Loibl, and Arun Nagarajah. Review and comparison of the methods of designing the digital twin. *Procedia CIRP*, 91:27–32, 2020.
- [102] Hakan Aydemir, Ugur Zengin, and Umut Durak. The digital twin paradigm for aircraft review and outlook. In *AIAA Scitech 2020 Forum*, page 0553, 2020.
- [103] Mengnan Liu, Shuiliang Fang, Huiyue Dong, and Cunzhi Xu. Review of digital twin about concepts , technologies , and industrial applications. *Journal of Manufacturing Systems*, (June):1–16, 2020. ISSN 0278-6125. doi: [10.1016/j.jmsy.2020.06.017](https://doi.org/10.1016/j.jmsy.2020.06.017).
- [104] Yan Xu, Yanming Sun, Xiaolong Liu, and Yonghua Zheng. A Digital-Twin-Assisted Fault Diagnosis Using Deep Transfer Learning. *IEEE Access*, 7(c):19990–19999, 2019. ISSN 21693536. doi: [10.1109/ACCESS.2018.2890566](https://doi.org/10.1109/ACCESS.2018.2890566).

- [105] Oney Erge and Eric van Oort. Combining physics-based and data-driven modeling in well construction: Hybrid fluid dynamics modeling. *Journal of Natural Gas Science and Engineering*, 97:104348, 2022. ISSN 1875-5100. doi: <https://doi.org/10.1016/j.jngse.2021.104348>. URL <https://www.sciencedirect.com/science/article/pii/S1875510021005436>.
- [106] Sindre Stenen Blakseth, Adil Rasheed, Trond Kvamsdal, and Omer San. Combining physics-based and data-driven techniques for reliable hybrid analysis and modeling using the corrective source term approach. *arXiv preprint arXiv:2206.03451*, 2022.
- [107] Joel Sansana, Mark N. Joswiak, Ivan Castillo, Zhenyu Wang, Ricardo Rendall, Leo H. Chiang, and Marco S. Reis. Recent trends on hybrid modeling for industry 4.0. *Computers and Chemical Engineering*, 151:107365, 2021. ISSN 0098-1354. doi: <https://doi.org/10.1016/j.compchemeng.2021.107365>.
- [108] Rahul Rai and Chandan K Sahu. Driven by data or derived through physics? a review of hybrid physics guided machine learning techniques with cyber-physical system (cps) focus. *IEEE Access*, 8:71050–71073, 2020.
- [109] Anuj Karpatne, Gowtham Atluri, James H Faghmous, Michael Steinbach, Arindam Banerjee, Auroop Ganguly, Shashi Shekhar, Nagiza Samatova, and Vipin Kumar. Theory-guided data science: A new paradigm for scientific discovery from data. *IEEE Transactions on knowledge and data engineering*, 29(10):2318–2331, 2017.
- [110] Mark Alber, Adrian Buganza Tepole, William R Cannon, Suvranu De, Salvador Dura-Bernal, Krishna Garikipati, George Karniadakis, William W Lytton, Paris Perdikaris, Linda Petzold, et al. Integrating machine learning and multiscale modeling—perspectives, challenges, and opportunities in the biological, biomedical, and behavioral sciences. *NPJ digital medicine*, 2(1): 1–11, 2019.
- [111] S. A. Vaghefi, M. A. Jafari, J. Zhu, J. Brouwer, and Y. Lu. A hybrid physics-based and data driven approach to optimal control of building cooling/heating systems. *IEEE Transactions on Automation Science and Engineering*, 13(2):600–610, 2016. doi: 10.1109/TASE.2014.2356337.
- [112] Vivek Dua. An artificial neural network approximation based decomposition approach for parameter estimation of system of ordinary differential equations. *Computers & chemical engineering*, 35(3):545–553, 2011.
- [113] Mohammadkazem Sadoughi and Chao Hu. Physics-based convolutional neural network for fault diagnosis of rolling element bearings. *IEEE Sensors Journal*, 19(11):4181–4192, 2019.
- [114] Adar Kahana, Eli Turkel, Shai Dekel, and Dan Givoli. Obstacle segmentation based on the wave equation and deep learning. *Journal of Computational Physics*, 413:109458, 2020.
- [115] Maziar Raissi, Paris Perdikaris, and George E Karniadakis. Physics-informed neural networks: A deep learning framework for solving forward and inverse problems involving nonlinear partial differential equations. *Journal of Computational Physics*, 378:686–707, 2019.

- [116] Yin hao Zhu, Nicholas Zabaras, Phaedon-Stelios Koutsourelakis, and Paris Perdikaris. Physics-constrained deep learning for high-dimensional surrogate modeling and uncertainty quantification without labeled data. *Journal of Computational Physics*, 394:56–81, 2019.
- [117] AIAA and AIA. Digital twin: Definition & value an aiaa and aia position paper, 2020. URL [https://www.aiaa.org/docs/default-source/uploadedfiles/issues-and-advocacy/policy-papers/digital-twin-institute-position-paper-\(december-2020\).pdf](https://www.aiaa.org/docs/default-source/uploadedfiles/issues-and-advocacy/policy-papers/digital-twin-institute-position-paper-(december-2020).pdf).
- [118] Michael G Kapteyn, David J Knezevic, DBP Huynh, Minh Tran, and Karen E Willcox. Data-driven physics-based digital twins via a library of component-based reduced-order models. *International Journal for Numerical Methods in Engineering*, 2020.
- [119] Yumei Ye, Qiang Yang, Fan Yang, Yanyan Huo, and Songhe Meng. Digital twin for the structural health management of reusable spacecraft: A case study. *Engineering Fracture Mechanics*, 234:107076, 2020.
- [120] Duansen Shangguan, Liping Chen, and Jianwan Ding. A digital twin-based approach for the fault diagnosis and health monitoring of a complex satellite system. *Symmetry*, 12(8):1307, 2020.
- [121] Ting-Yu Chen. Eigenvector derivatives for doubly repeated eigenvalues. *AIAA journal*, 34(7):1531–1533, 1996.
- [122] Shaban Alyari Shourehdeli and Mohammad Hossein Mesbahi. Application of roe’s flux-difference splitting for turbulent reacting flow with k- ϵ turbulence model. In *Applied Mechanics and Materials*, volume 110, pages 3405–3414. Trans Tech Publ, 2012.
- [123] A Nawab and L Di Mare. Re τ scaling of pod modes in plane channel flow. *Physics of Fluids*, 30(5):055109, 2018.
- [124] Yijin Guan, Zhihang Yuan, Guangyu Sun, and Jason Cong. Fpga-based accelerator for long short-term memory recurrent neural networks. In *2017 22nd Asia and South Pacific Design Automation Conference (ASP-DAC)*, pages 629–634, 2017. doi: 10.1109/ASPDAC.2017.7858394.
- [125] Zhanrui Sun, Yongxin Zhu, Yu Zheng, Hao Wu, Zihao Cao, Peng Xiong, Junjie Hou, Tian Huang, and Zhiqiang Que. Fpga acceleration of lstm based on data for test flight. In *2018 IEEE International Conference on Smart Cloud (SmartCloud)*, pages 1–6. IEEE, 2018.
- [126] NianYi Wang, Jing Nie, JingBin Li, Kang Wang, and ShunKang Ling. A compression strategy to accelerate lstm meta-learning on fpga. *ICT Express*, 2022.
- [127] Hirokazu Suzuki, Seiji Tsutsumi, and Yukinori Sato. An fpga implementation of a gaussian process based predictor for sequential time series data. In *2020 Eighth International Symposium on Computing and Networking Workshops (CANDARW)*, pages 445–449. IEEE, 2020.
- [128] Frank R. Hampel. A General Qualitative Definition of Robustness. *The Annals of Mathematical Statistics*, 42(6):1887 – 1896, 1971. doi: 10.1214/aoms/1177693054.

- [129] Cameron McPhail, HR Maier, JH Kwakkel, M Giuliani, A Castelletti, and S Westra. Robustness metrics: How are they calculated, when should they be used and why do they give different results? *Earth's Future*, 6(2): 169–191, 2018.
- [130] Abraham Wald. Statistical decision functions. 1950.
- [131] JH Kwakkel, S Eker, E Pruyt, M Doumpos, C Zopounidis, and E Grigoroudis. Robustness analysis in decision aiding, optimization, and analytics, 2016.
- [132] Leonard J Savage. The theory of statistical decision. *Journal of the American Statistical association*, 46(253):55–67, 1951.
- [133] Herbert A Simon. Rational choice and the structure of the environment. *Psychological review*, 63(2):129, 1956.
- [134] Leonid Hurwicz. Optimality criteria for decision making under ignorance. Technical report, Cowles Commission Discussion Paper, Statistics, 1951.
- [135] Pierre Simon LaPlace. A physiological essay on probabilities. translation by fw truscott and fl emory, 1951.
- [136] Matteo Giuliani and Andrea Castelletti. Is robustness really robust? how different definitions of robustness impact decision-making under climate change. *Climatic Change*, 135(3):409–424, 2016.
- [137] Caner Hamarat, Jan H Kwakkel, Erik Pruyt, and Erwin T Loonen. An exploratory approach for adaptive policymaking by using multi-objective robust optimization. *Simulation Modelling Practice and Theory*, 46:25–39, 2014.
- [138] Vlasios Voudouris, Ken'ichi Matsumoto, John Sedgwick, Robert Rigby, Dimitrios Stasinopoulos, and Michael Jefferson. Exploring the production of natural gas through the lenses of the aceges model. *Energy Policy*, 64: 124–133, 2014.
- [139] Martin Kenneth Starr. *Product design and decision theory*. Prentice-Hall, 1963.
- [140] GO Schneller and GP Sphicas. Decision making under uncertainty: Starr's domain criterion. *Theory and Decision*, 15(4):321–336, 1983.
- [141] J. Duha and G. B. Afonso. Thermal force effects on satellites. *Revista Brasileira de Geofisica*, 17(2-3):162–173, 1999. ISSN 0102261X. doi: 10.1590/S0102-261X1999000200005.
- [142] Vasily Gorev, Anatoly Pelemeshko, Alexander Zadorozhny, and Aleksey Sidorchuk. Thermal deformation of 3U CubeSat in low Earth orbit. *MATEC Web of Conferences*, 158:1–5, 2018. ISSN 2261236X. doi: 10.1051/matecconf/201815801013.
- [143] H. Pälike. Earth — orbital variation (including milankovitch cycles). In Richard C. Selley, L. Robin M. Cocks, and Ian R. Plimer, editors, *Encyclopedia of Geology*, pages 410–421. Elsevier, Oxford, 2005. ISBN 978-0-12-369396-9. doi: <https://doi.org/10.1016/B0-12-369396-9/00123-4>.

- [144] Betta batteries.
- [145] Michel Mattei, Gilles Notton, Christian Cristofari, Marc Muselli, and Philippe Poggi. Calculation of the polycrystalline pv module temperature using a simple method of energy balance. *Renewable energy*, 31(4):553–567, 2006.
- [146] Japanese aerospace exploration agency (in japanese). URL <https://www.kenkai.jaxa.jp/research/pastpro/sds4/sds4-outline.html>.
- [147] Qingsong Wen, Liang Sun, Fan Yang, Xiaomin Song, Jingkun Gao, Xue Wang, and Huan Xu. Time series data augmentation for deep learning: A survey. *arXiv preprint arXiv:2002.12478*, 2020.
- [148] Brian Kenji Iwana and Seiichi Uchida. An empirical survey of data augmentation for time series classification with neural networks. *Plos one*, 16(7):e0254841, 2021.
- [149] Khandakar M Rashid and Joseph Louis. Window-warping: a time series data augmentation of imu data for construction equipment activity identification. In *ISARC. Proceedings of the International Symposium on Automation and Robotics in Construction*, volume 36, pages 651–657. IAARC Publications, 2019.
- [150] Chris M Bishop. Training with noise is equivalent to tikhonov regularization. *Neural computation*, 7(1):108–116, 1995.
- [151] Guozhong An. The effects of adding noise during backpropagation training on a generalization performance. *Neural computation*, 8(3):643–674, 1996.
- [152] Terry T Um, Franz MJ Pfister, Daniel Pichler, Satoshi Endo, Muriel Lang, Sandra Hirche, Urban Fietzek, and Dana Kulić. Data augmentation of wearable sensor data for parkinson’s disease monitoring using convolutional neural networks. In *Proceedings of the 19th ACM international conference on multimodal interaction*, pages 216–220, 2017.