Temperature Prediction Using Thermal Fluctuations from Wireless Sensor Networks in Adaptive Filter Model

by

Mitchell D. Hwang

Submitted to the Department of Electrical Engineering and Computer Science
in partial fulfillment of the requirements for the degree of

Master of Engineering in Computer Science and Engineering

at the

MASSACHUSETTS INSTITUTE OF TECHNOLOGY

February 2020

Author . . . .
Department of Electrical Engineering and Computer Science
Dec 13, 2019

Certified by . . . .
Joseph D. Steinmeyer
Principal Lecturer
Thesis Supervisor

Accepted by . . . .
Katrina LaCurts
Chair, Master of Engineering Thesis Committee

Temperature Prediction Using Thermal Fluctuations from Wireless Sensor Networks in Adaptive Filter Model

by

Mitchell D. Hwang

Submitted to the Department of Electrical Engineering and Computer Science
on Dec 13, 2019, in partial fulfillment of the
requirements for the degree of
Master of Engineering in Computer Science and Engineering

Abstract

In many scientific experiments, it is imperative to minimize the unintended effects of variables other than the independent variables. Temperature, pressure, and gas levels are factors controlled to a certain extent using expensive climate-controlling technology, yet the resolution for monitoring their levels is generally low. The downward scaling of communication-enabled electronics in size, cost, and energy provides a potential toolset for tracking such data with high spatial and temporal resolutions. We establish a data collection methodology through a low-cost, small footprint distributed network system of modules that records data in a remote server. The system architecture allows for increased spatial resolutions, demonstrates high precision of measurements, and investigates room dynamics. Modules are fabricated using commercial sensors such as the ESP8266, BME680, and TCS34725. In this paper, we propose a temperature prediction model using adaptive filter methodologies to learn the relationship between thermal fluctuations at distinct locations within a lab environment.

Thesis Supervisor: Joseph D. Steinmeyer
Title: Principal Lecturer

Acknowledgments

I would like to thank my advisor Joseph Steinmeyer for providing me with this learning opportunity. Without his guidance and patience throughout the program, this thesis could not have been completed, and I cannot imagine having a kinder or more supportive mentor. His wealth of knowledge and innovative thinking are inspiring, and each time I work with him, I feel a wave of assurance and am invigorated to do more. Above that, he is extremely thoughtful and caring, and he is a source of inspiration for me and many others.

I am grateful to Tony Eng for giving me the chance to teach numerous students in parallel to the program. I admire his expertise in communicating technical concepts and charisma as a speaker. Over the years, I have learned a great deal from him and from my students, and I could not have asked for a better course to assist in teaching.

My time at MIT with all the classes I took, all the clubs I joined, and all the people I met has been nothing short of amazing.

I thank all my friends that have stuck with me through thick and thin. I am proud to have called Next Two East my home during undergrad, and I am grateful to come home to my roommates Alicia, Ben, Brenda, and Famien during grad. Special thanks to Olivia for being an amazing friend from freshman year.

I would like to thank my mother for all that she has done for me. Without her continuous support I would not have grown into the person I am now.

Contents

1 Introduction **17**

1.1 Background 17

1.2 Contributions 19

1.3 Organization 20

2 Related Work **21**

2.1 Tracking 21

2.2 Localization 22

2.3 Data Reduction 22

2.4 LMS-based Prediction Scheme 23

3 System Overview **25**

3.1 System Environment 26

3.2 Wireless Sensor Nodes 26

3.2.1 Components 26

3.2.2 Firmware 27

3.3 Data Server 28

4 Problem Setup, LMS-based Model, and Experimental Procedure **31**

4.1 Wiener Filters 31

4.2 LMS Algorithm 32

4.3 Problem Definition 32

4.4 Transversal Filter and Linear Combiner Model 33

4.5 Procedure 34
4.5.1 Data Preprocessing 35
4.5.2 Locating Training Spots 35
4.5.3 LMS Filter Weights 39
4.5.4 Weight Heat Maps 39
4.5.5 Temperature Predicting 39

5 Performance 41
5.1 Clustered Setup 41
5.1.1 Temperature Data Results 42
5.1.2 Training Spots 43
5.1.3 Weight Heatmaps 43
5.1.4 Temperature Predictor Performance 44
5.2 Semi-Clustered Setup 44
5.2.1 Temperature Data Collected 45
5.2.2 Training Spots 46
5.2.3 Weight Heatmaps 46
5.2.4 Temperature Predictor Performance 46
5.3 Spread Out Setup 47
5.3.1 Temperature Data Collected 47
5.3.2 Training Spots 48
5.3.3 Weight Heatmaps 48
5.3.4 Temperature Predictor Performance 48
5.4 Implementation Details and Results 49

6 Conclusion 51

7 Supplemental Work 53
7.1 Design 53
7.2 ESP32 54

A Tables 55

B Figures **59**

List of Figures

3-1 System Architecture - environment, sensor nodes, and data server . . 25

3-2 Outline of room 38-500 on MIT campus 26

3-3 Sensor Node comprised of the ESP8266, BME680, and TCS34725 . . 27

3-4 Example rows in SQLite database 28

3-5 Example plots of measurements from one node 29

4-1 Transversal Filter Structure 32

4-2 Linear Combiner Structure 32

4-3 Adaptive Transversal Filter and Linear Combiner 34

4-4 Plot of Cumulative Standard Deviations Over Time 38

4-5 Example Heat Map for 8 sensor nodes with history length 5 40

4-6 Example Error plot with RMSE calculated 40

5-1 Clustered Formation of Wireless Sensor Nodes indicated by circles . . 42

5-2 Semi-Clustered Formation of Wireless Sensor Nodes, indicated by circles 45

5-3 Spread Out Formation of Wireless Sensor Nodes, indicated by circles 47

B-1 Temperatures of Sensor nodes in Clustered Set up 60

B-2 Temperatures of Unclustered Sensor nodes in Clustered Set up 60

B-3 Temperatures of Clustered Sensor nodes in Clustered Set up 61

B-4 Clustered Setup Cumulative Standard Deviations 61

B-5 Heatmaps from Most Significant Time Period 62

B-6 Heatmaps from 2nd Most Significant Time Period 62

B-7 Heatmaps from 3rd Most Significant Time Period 63

B-8 Heatmaps from Average Significance Time Period 63
B-9 Heatmaps from Least Significant Time Period 64
B-10 Clustered Short Term Performances For Most Significant Time Period 65
B-11 Clustered Short Term Performances For Average Significance Time Period 66
B-12 Clustered Short Term Performances For Least Significant Time Period 67
B-13 Clustered Long Term Performances For Most Significant Time Period 68
B-14 Clustered Long Term Performances For Average Significance Time Period 69
B-15 Clustered Long Term Performances For Least Significant Time Period 70
B-16 Temperatures of Sensor nodes in Semi-Clustered Set up 71
B-17 Temperatures of Sensor nodes in Semi-Clustered Set up 71
B-18 Semi-Clustered Setup Cumulative Standard Deviations 72
B-19 Semi-Clustered Setup Cumulative Standard Deviations 72
B-20 Heatmaps from Most Significant Time Period Before Swap 73
B-21 Heatmaps from Most Significant Time Period After Swap 73
B-22 Heatmaps from Average Significance Time Period Before Swap 74
B-23 Heatmaps from Average Significance Time Period After Swap 74
B-24 Heatmaps from Least Significant Time Period Before Swap 75
B-25 Heatmaps from Least Significant Time Period After Swap 75
B-26 Semi-Clustered Short Term Performances For Most Significant Time Period Before Swap 76
B-27 Semi-Clustered Short Term Performances For Most Significant Time Period After Swap 77
B-28 Semi-Clustered Short Term Performances For Average Significance Time Period Before Swap 78
B-29 Semi-Clustered Short Term Performances For Average Significance Time Period After Swap 79
B-30 Semi-Clustered Short Term Performances For Least Significant Time Period Before Swap 80

B-31 Semi-Clustered Short Term Performances For Least Significant Time Period After Swap 81
B-32 Semi-Clustered Long Term Performances For Most Significant Time Period Before Swap 82
B-33 Semi-Clustered Long Term Performances For Most Significant Time Period After Swap 83
B-34 Semi-Clustered Long Term Performances For Average Significance Time Period Before Swap 84
B-35 Semi-Clustered Long Term Performances For Average Significance Time Period After Swap 85
B-36 Semi-Clustered Long Term Performances For Least Significant Time Period Before Swap 86
B-37 Semi-Clustered Long Term Performances For Least Significant Time Period After Swap 87
B-38 Temperatures of Sensor nodes in Spread Out Set up 88
B-39 Spread Out Setup Cumulative Standard Deviations 88
B-40 Heatmaps from Most Significant Time Period 89
B-41 Heatmaps from Average Significance Time Period 89
B-42 Heatmaps from Least Significant Time Period 90
B-43 Spread Out Short Term Performances For Most Significant Time Period 91
B-44 Spread Out Short Term Performances For Average Significance Time Period 92
B-45 Spread Out Short Term Performances For Least Significant Time Period 93
B-46 Spread Out Long Term Performances For Most Significant Time Period 94
B-47 Spread Out Long Term Performances For Average Significance Time Period 95
B-48 Spread Out Long Term Performances For Least Significant Time Period 96

List of Tables

A.1 Clustered Setup Short Term Performance RMS Error 55
A.2 Clustered Setup Long Term Performance RMS Error 55
A.3 Semi-Clustered Setup Short Term Performance RMS Error Before Swap 56
A.4 Semi-Clustered Setup Short Term Performance RMS Error After Swap 56
A.5 Semi-Clustered Setup Long Term Performance RMS Error Before Swap 56
A.6 Semi-Clustered Setup Long Term Performance RMS Error After Swap 57
A.7 Spread Out Setup Short Term Performance RMS Error 57
A.8 Spread Out Setup Long Term Performance RMS Error 57

Chapter 1

Introduction

1.1 Background

In the life sciences, logging the environmental conditions of experiments is often critical to ensuring reproducibility. Temperature, pressure, humidity, gas concentration, and lux typically play important roles in biological experiments; slightly different values of such metrics for the same experiment can significantly change the results. As such, an incomplete picture of these factors may confound the underlying mechanisms being monitored and mislead researchers, causing the researchers to rerun experiments to ensure that marginal fluctuations did not lead to varying results [1]. For that reason, it is vital to accurately monitor the experimental environment.

Controlling and monitoring environmental factors have been made possible through intricate climate-controlled buildings using a heating, ventilating, and air conditioning system [2] and cell culture incubators that require precise handling [3, 4, 5]. However, their resolution of control is limited to setting environmental factors to one-dimensional values, and knowing whether environmental conditions are completely uniform or how they vary across the environment is not trivial. When measured at locations meters apart, environmental factors can significantly differ and experience varying levels of fluctuations throughout the day [6]. Placing sensors at every relevant location does not practically scale, for sensors at numerous relevant locations can quickly strain financial costs and become obtrusive in an experimental setting.

However, understanding how fluctuations from one location pervades and affects ensuing fluctuations at another location can help with understanding the dynamics of a closed environment. To do so, it is crucial to track these fluctuations at a high spatial and high temporal resolution.

The recent rise in the Internet of Things (IoT) has promoted wireless sensor networks (WSN) as an alternate means for environmental monitoring. WSNs are spatially flexible and effectively capture metrics from various parts of a local environment [7]. This property has allowed WSNs to be used for various tasks related to environmental monitoring such as tracking [8, 9], localization [10], data reduction [11], and energy conservation [12]; it makes a WSN a suitable candidate to log the environmental fluctuations at different locations within a confined space. However, knowing the number and configuration of sensor nodes necessary to properly monitor an environment is difficult. A WSN without enough nodes or a proper configuration can fail to capture important signals in the environment. Recognizing how a node's measurements can be interpolated from neighbor nodes' measurements is essential in order to save energy, transmit less redundant information, and simplify the complexity of the WSN.

In order to understand how fluctuations from different locations affect each other, we created a WSN with a low cost and a small footprint design, developed a system to collect and store measurements, and set up several experiments targeting temperature fluctuations. We present a novel approach towards modeling how temperature signals at arbitrarily sparse locations within an environment can be predicted. A correct model not only provides more transparency into how environmental factors pervade their environment but also simplifies the monitoring system by identifying and removing inessential nodes whose measurements can be interpolated. Our belief is that a temperature signal's high temporal fluctuations are correlated to both its and nearby temperature signals' previous values.

1.2 Contributions

In this paper, we show how a WSN created from commercially available sensors and microcontrollers can be used to model room dynamics. More specifically, our work contributes towards the methodology of using a LMS-based adaptive filter to predict temperatures monitored by a WSN. The main goal of this paper is to test the hypothesis that, within an environment, temperatures at different locations are inherently related, and their future values can be determined by learning high temporal trends in their past values.

In doing so, we constructed an end-to-end system where multiple wireless sensor nodes asynchronously log measurements in a central server. The primary function of this system was to capture various environmental factor signals at distinct locations at a high frequency to avoid aliasing, the misidentification of a signal due to improper sampling. We trained our adaptive filter model with these measurements to identify the attributes of temperature signals that reveal thermal trends.

With an accurate characterization of how environmental factors propagate throughout a room, we can simplify the data models describing those dynamics. Furthermore, simplification of information networks by removing nonessential sensor nodes allows WSNs to transmit less redundant data and reduce energy consumption.

This thesis explores temperature prediction through high-temporal trends with four principal contributions:

1. demonstrating that significant high temporal trends do not provide value in our configuration for predicting temperatures with room-temperature fluctuations

2. revealing that distinguishability between two temperature signals in a general room positively correlates to the physical distance in between them

3. providing evidence that a static linear model does not adequately characterize the dynamics between temperature fluctuations

4. introducing a scalable low-cost and small footprint WSN to monitor temperature, humidity, pressure, gas, and lux

1.3 Organization

Chapter 2 introduces existing applications of WSNs and LMS-based filters. Chapter 3 presents an overview of our system's architecture. Chapter 4 details our procedure for training and evaluating a LMS-based model for predicting temperatures. Chapter 5 dives into different WSN configuration setups and the performance analysis of the trained models. Chapter 6 concludes the paper with a summary of our work, our results, and their implications. Chapter 7 introduces supplemental work done for the project.

Chapter 2

Related Work

Using a WSN for a prediction focused, LMS-based scheme is novel. However, there exist several other studies of using WSNs for other applications, and LMS filters have been used to predict temperature on a smaller scale. The following are some applications of our WSN and their comparison to how other related works have been used in the field.

2.1 Tracking

Our WSN tracks fluctuations throughout a room by frequently recording environmental factors. In doing so, the WSN passively tracks events, which we define as the underlying mechanisms that induce fluctuations throughout the room, such as convection currents for heat transfer. This differs from other WSNs that actively identify and track events to conserve energy by turning off sensor nodes that are far from the event [8]. Another work, unlike both of these cases where the sensors of the WSN are statically situated, directly tracks moving objects by physically attaching sensors to the objects [9].

2.2 Localization

Through our methodology, we aim to have our sensor nodes learn to predict their future values based on other nodes' previous values. We believe that relations between these nodes can be constructed even without knowing distances between nodes or their absolute locations within their environment, so we opt to exclude location information from our model. We expect since a room is generally well behaved in terms of thermal and environmental gradients, that nodes closer together will have a stronger impact in predictions and that these patters should be learned by our model during training. Some other works place more significance in determining relative spatial coordinates of their nodes, for their nodes perform actions dependent on location [10].

2.3 Data Reduction

Due to the importance of fluctuations in our model, we aim for sensor nodes to capture as much information about the environment as possible for purposes of avoiding aliasing. Our system logs data at a rate of 45 measurements per node per minute, far faster than any reasonable environmental gradient that we expect to encounter. After successfully identifying patterns in the data, our models can be optimized to recognize which nodes are unnecessary for data estimation and identify the optimal rate of measurements to estimate accurately, thereby reducing data transmissions and hardware needs. As a result of the high throughput of data being logged, our sensor nodes post their data in batches to avoid congestion while the server writes the data to a database. Other works place more importance in optimizing bandwidth, energy, and data throughput usage, and employ a more complex adaptive data aggregation scheme to wirelessly transfer information [11]. In some schemes data aggregation eliminates redundancy in information collected by nodes, thus reducing transmissions and energy usage [12].

2.4 LMS-based Prediction Scheme

Other works have used a LMS-based temperature prediction scheme to maximize the performance of their systems ranging from the chip level to the level of a power plant [13, 14]. Similar to how our system applies a LMS filter to learn how fluctuations influence distinct locations within a space to predict temperature readings, another work focuses on energy efficiency by applying a LMS filter to predict data and reduce transmissions [15].

Chapter 3

System Overview

There are three main components of our system as shown below in Figure 3-1. The first is the physical environment in which temperature, humidity, pressure, gas, and lux fluctuations are being monitored. Next is the WSN that comprises of independent sensor nodes that each asynchronously post aforementioned measurements to our server's web application endpoint. The last is the centralized data server that hosts the web application and records the measurements in a SQLite database.

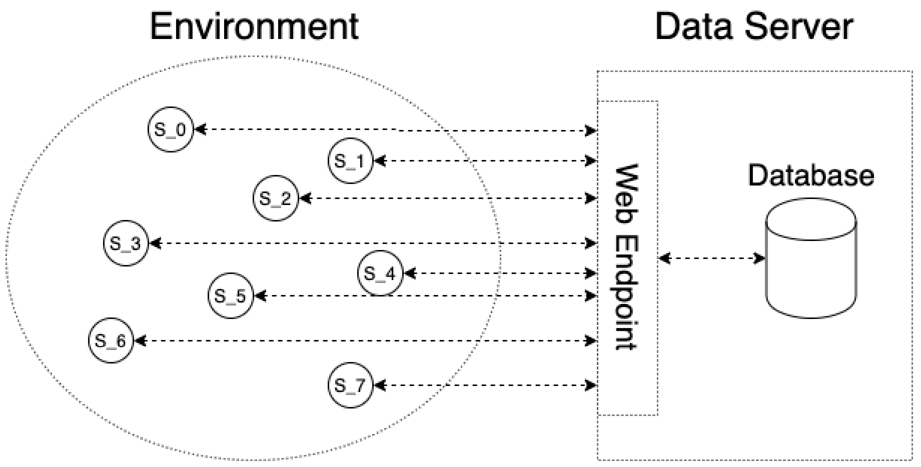


Figure 3-1: System Architecture - environment, sensor nodes, and data server

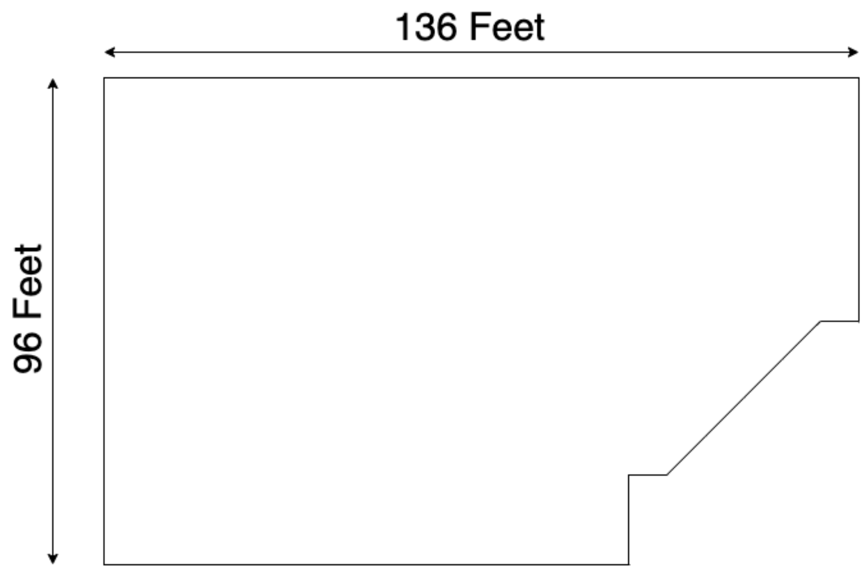


Figure 3-2: Outline of room 38-500 on MIT campus

3.1 System Environment

For our environment, we chose to use lab room 38-500 on MIT campus. It features a spacious and air conditioned environment that is easily accessible. Though it is a public space for those affiliated with MIT, there are many discreet places to station the sensor nodes, such as on window sills and on the top shelves of lab benches. Figure 3-2 depicts an outline of the lab space.

3.2 Wireless Sensor Nodes

A secondary goal for our work was to produce an inexpensive and small solution for environmental monitoring, so we sought to build our own wireless sensor nodes. In total, we built eight nodes to collect data across the lab.

3.2.1 Components

The physical makeup of our wireless sensor nodes consists of Espressif's ESP8266 system-on-a-chip microcontroller on the D1 mini breakout board, Bosch Sensortec's BME680 environmental sensor on Adafruit's breakout board, and the TCS34725 color sensor on Adafruit's breakout board. The ESP8266, popular for its low price, has Wi-Fi capabilities that allow for our nodes to operate wirelessly from our server. Each node monitors a variety of environmental factors within the lab by using the BME680 to capture temperature, humidity, barometric pressure, and volatile organic compounds (VOC) and the TCS34725 to measure illuminance. Below, Figure 3-3 demonstrates how the three components can be hooked up on a breadboard.

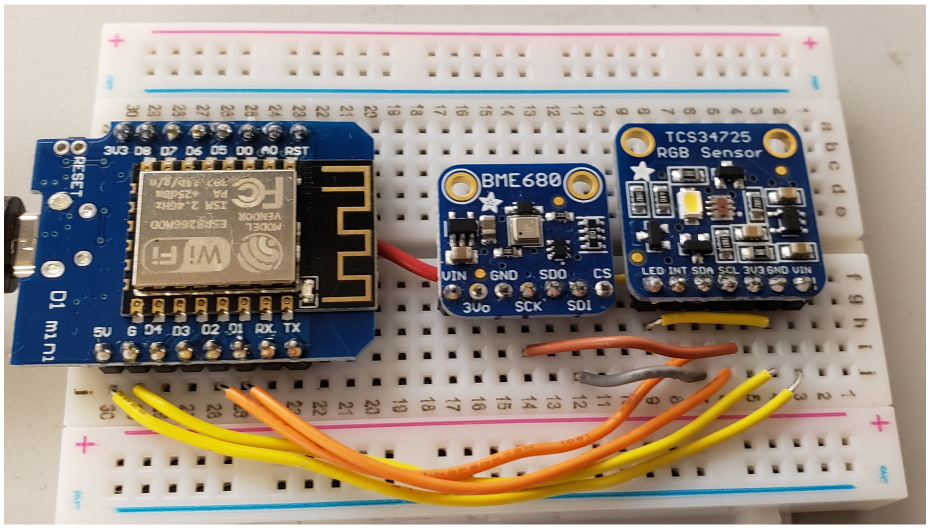
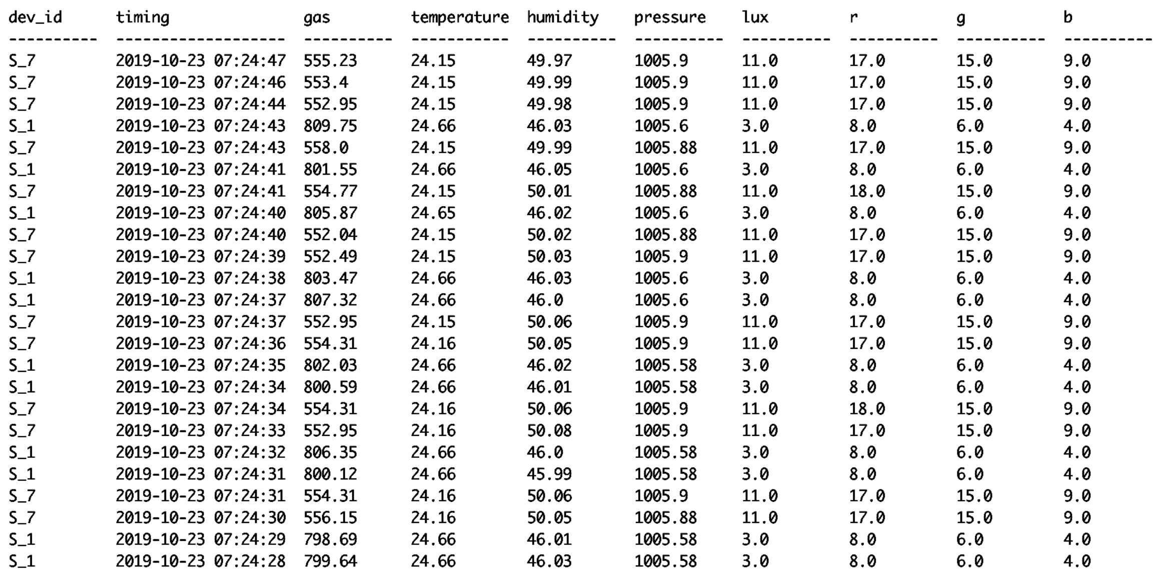


Figure 3-3: Sensor Node comprised of the ESP8266, BME680, and TCS34725

3.2.2 Firmware

The firmware programmed onto the ESP8266 microcontroller directs each node to collect and send data to the server. Capturing data at a high temporal resolution is essential to ensure that we did not overlook subtle yet significant changes in data. Our nodes performed readings at a rate of 45 measurements every minute. However, synchronizing measurement timings among nodes proved challenging, so we opted to have each node retrieve the local time from the server when posting their data and independently associate timestamps with each measurement. To reduce data transmission frequency to the server, nodes store measurements in memory and sent them in a batch once they reached a fixed number of 60 measurements.

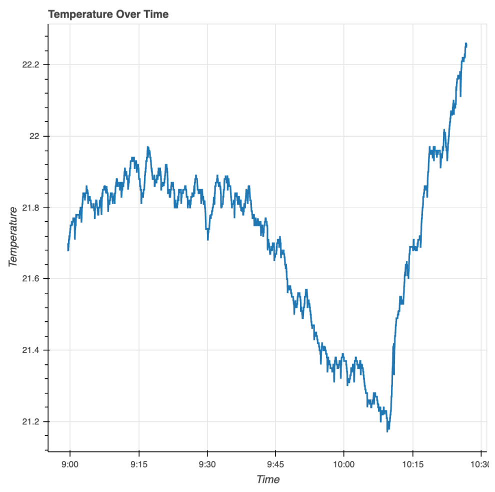


```
dev_id      timing               gas         temperature  humidity    pressure    lux         r           g           b
----------  -------------------  ----------  -----------  ----------  ----------  ----------  ----------  ----------  ----------
S_7         2019-10-23 07:24:47  555.23      24.15        49.97       1005.9      11.0        17.0        15.0        9.0
S_7         2019-10-23 07:24:46  553.4       24.15        49.99       1005.9      11.0        17.0        15.0        9.0
S_7         2019-10-23 07:24:44  552.95      24.15        49.98       1005.9      11.0        17.0        15.0        9.0
S_1         2019-10-23 07:24:43  809.75      24.66        46.03       1005.6      3.0         8.0         6.0         4.0
S_7         2019-10-23 07:24:43  558.0       24.15        49.99       1005.88     11.0        17.0        15.0        9.0
S_1         2019-10-23 07:24:41  801.55      24.66        46.05       1005.6      3.0         8.0         6.0         4.0
S_7         2019-10-23 07:24:41  554.77      24.15        50.01       1005.88     11.0        18.0        15.0        9.0
S_1         2019-10-23 07:24:40  805.87      24.65        46.02       1005.6      3.0         8.0         6.0         4.0
S_7         2019-10-23 07:24:40  552.04      24.15        50.02       1005.88     11.0        17.0        15.0        9.0
S_7         2019-10-23 07:24:39  552.49      24.15        50.03       1005.9      11.0        17.0        15.0        9.0
S_1         2019-10-23 07:24:38  803.47      24.66        46.03       1005.6      3.0         8.0         6.0         4.0
S_1         2019-10-23 07:24:37  807.32      24.66        46.0        1005.6      3.0         8.0         6.0         4.0
S_7         2019-10-23 07:24:37  552.95      24.15        50.06       1005.9      11.0        17.0        15.0        9.0
S_7         2019-10-23 07:24:36  554.31      24.16        50.05       1005.9      11.0        17.0        15.0        9.0
S_1         2019-10-23 07:24:35  802.03      24.66        46.02       1005.58     3.0         8.0         6.0         4.0
S_1         2019-10-23 07:24:34  800.59      24.66        46.01       1005.58     3.0         8.0         6.0         4.0
S_7         2019-10-23 07:24:34  554.31      24.16        50.06       1005.9      11.0        18.0        15.0        9.0
S_7         2019-10-23 07:24:33  552.95      24.16        50.08       1005.9      11.0        17.0        15.0        9.0
S_1         2019-10-23 07:24:32  806.35      24.66        46.0        1005.58     3.0         8.0         6.0         4.0
S_1         2019-10-23 07:24:31  800.12      24.66        45.99       1005.58     3.0         8.0         6.0         4.0
S_7         2019-10-23 07:24:31  554.31      24.16        50.06       1005.9      11.0        17.0        15.0        9.0
S_7         2019-10-23 07:24:30  556.15      24.16        50.05       1005.88     11.0        17.0        15.0        9.0
S_1         2019-10-23 07:24:29  798.69      24.66        46.01       1005.58     3.0         8.0         6.0         4.0
S_1         2019-10-23 07:24:28  799.64      24.66        46.03       1005.58     3.0         8.0         6.0         4.0
```

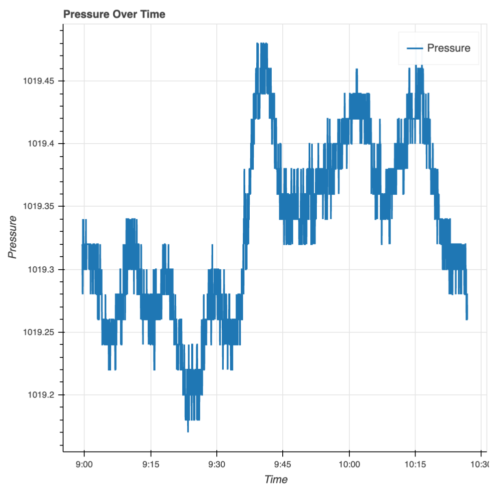
Figure 3-4: Example rows in SQLite database

3.3 Data Server

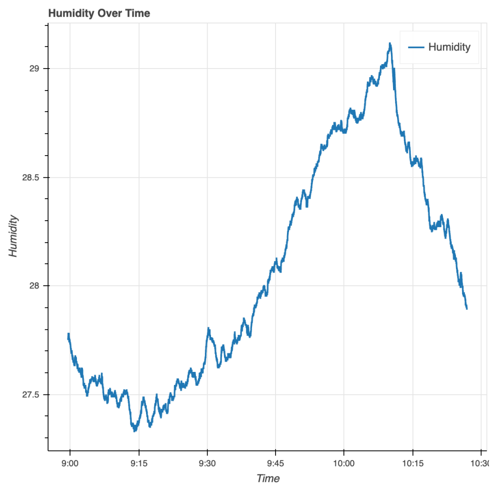
The server on the receiving end of data transmissions from sensor nodes hosts a flask-based web application with an endpoint specific for nodes to post data and for generating plots of posted data for viewing. Upon receiving and parsing the transmitted data, the server writes to a table in a SQLite database, recording the node identifier, the measurement time stamp, and the various environmental metrics as shown in Figure 3-4. Afterwards, the server returns the local time to the transmitting node to update the node's measurement timing. Moreover, the server plots each node's measurements over time, as shown in Figure 3-5. That way, trends can be visually identified, and any node outages can be easily discovered.



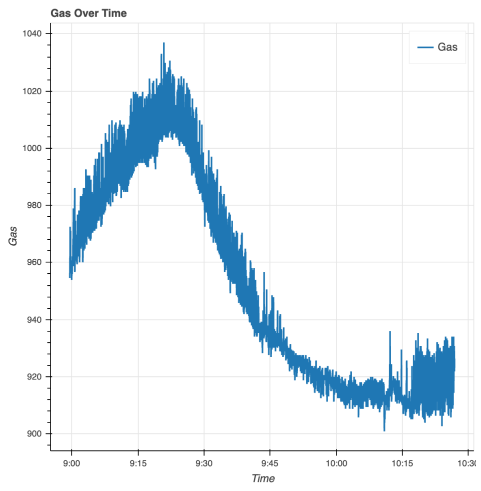
(a) Temperature



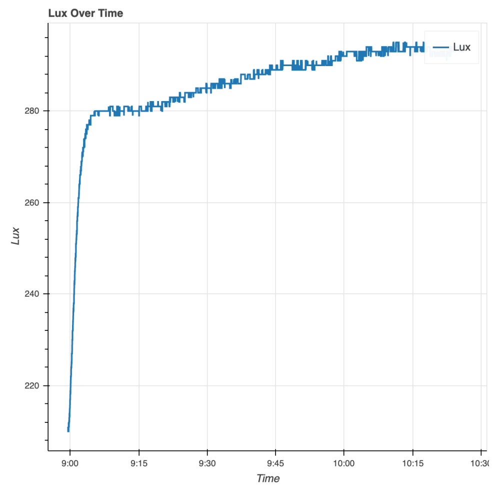
(b) Pressure



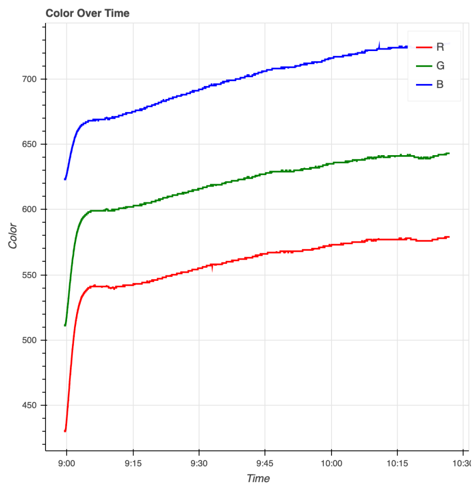
(c) Humidity



(d) Gas



(e) Lux



(f) Red Green Blue Intensity

Figure 3-5: Example plots of measurements from one node

Chapter 4

Problem Setup, LMS-based Model, and Experimental Procedure

In this chapter, we start with introducing the ideas of adaptive Wiener filters and the highly regarded LMS algorithm. We proceed with formalizing the problem and examine how our model looks to solve it. We then detail the steps in our experimental procedure for processing data, training the model, and evaluating the model.

4.1 Wiener Filters

Wiener filters, a class of adaptive linear filters, minimize the mean squared error (MSE) between a desired random process's signal and an estimated signal generated via a linear combination of past and present values of signals related to that random process. Typically, many realizations of the signals' statistics are required for these filters to estimate reliably. However, these filters can be designed with the assumption that the signals are ergodic, meaning that they are stationary and their statistical averages are equivalent to their time averages. Adaptive linear filters commonly come in the form of a transversal structure, as shown in Figure 4-1, where the output signal is estimated by linearly combining delayed samples of one signal, and a linear combiner structure, as shown in Figure 4-2, where the output signal is estimated by linearly combining several different signals.

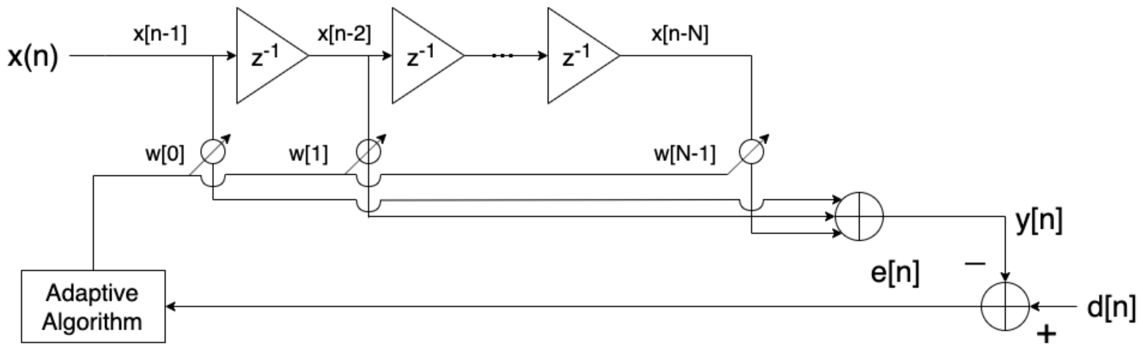


Figure 4-1: Transversal Filter Structure

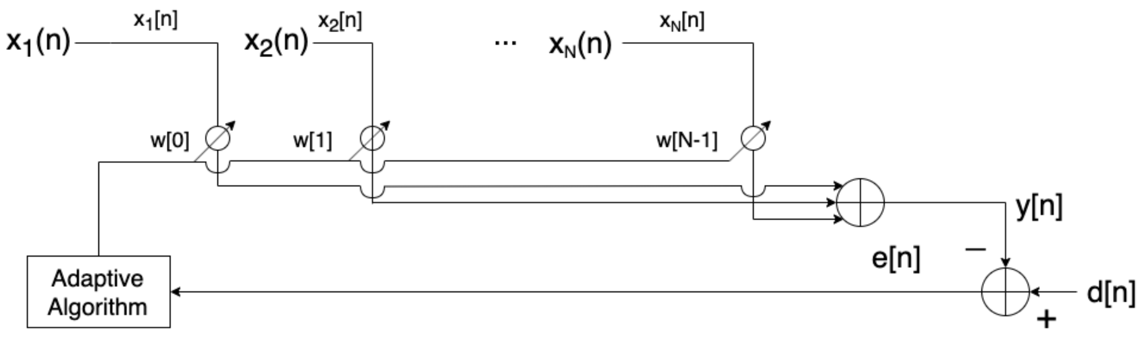


Figure 4-2: Linear Combiner Structure

4.2 LMS Algorithm

The LMS algorithm is the most popular solution to determining Wiener filter coefficients and is a stochastic implementation of the steepest-descent algorithm. It aims to minimize the MSE between the output and desired signals, and, for our purposes, our only requirement is that weights converge for our filter.

4.3 Problem Definition

A WSN is located in a spacious lab environment with several independent nodes arbitrarily spaced out from one another, as shown in Figure 3-2. Each node frequently measures its local temperature. We design a model that can predict future temperature values at the node locations without relying on spatial relations between the nodes or how the lab space is controlled.

4.4 Transversal Filter and Linear Combiner Model

The fluctuations in temperatures at precise locations are hard to determine. Treating temperature signals as a random process, we apply the idea of a Wiener filter with a transversal structure to use past and present values of the temperature signal at one node to predict the signal's future values. Moreover, we believe there are correlations between temperature fluctuations at distinct parts of the room. Therefore, we expand our filter structure with a flavor of the linear combiner structure by including other nodes' past and present values of their temperature signals as inputs.

More formally, with N distinct nodes, we define temperature signals at the node locations as $X_1, X_2, ..., X_N$. For each temperature signal, X_i, we build a Wiener filter with desired output as that temperature signal, X_i, and filter input based on histories of all the temperature signals. By nature, the measurements are discrete-time signals, so with history length H, our input, $x(n)$, of size $N * H$ is of the form:

$$\begin{aligned} x(n) = [x_1(n), x_2(n), ..., x_N(n)] = [&X_1[-1], X_1[-2], ..., X_1[-H], \\ &X_2[-1], X_2[-2], ..., X_2[-H], \\ &..., \\ &X_N[-1], X_N[-2], ..., X_N[-H]] \end{aligned} \tag{4.1}$$

With filter tap weights for each input, $w_i(n)$, our filter is depicted in Figure 4-3. This leads to a filter output, $y(n)$, as shown below in Equation 4.2.

$$y(n) = \sum_{i=0}^{N-1} \sum_{j=0}^{H-1} w_i[j] x_i[j] \tag{4.2}$$

We optimize tap weights $w_i(n)$ such that the estimation error $e(n) = x_i(n) - y(n)$ is minimized by using the LMS algorithm.

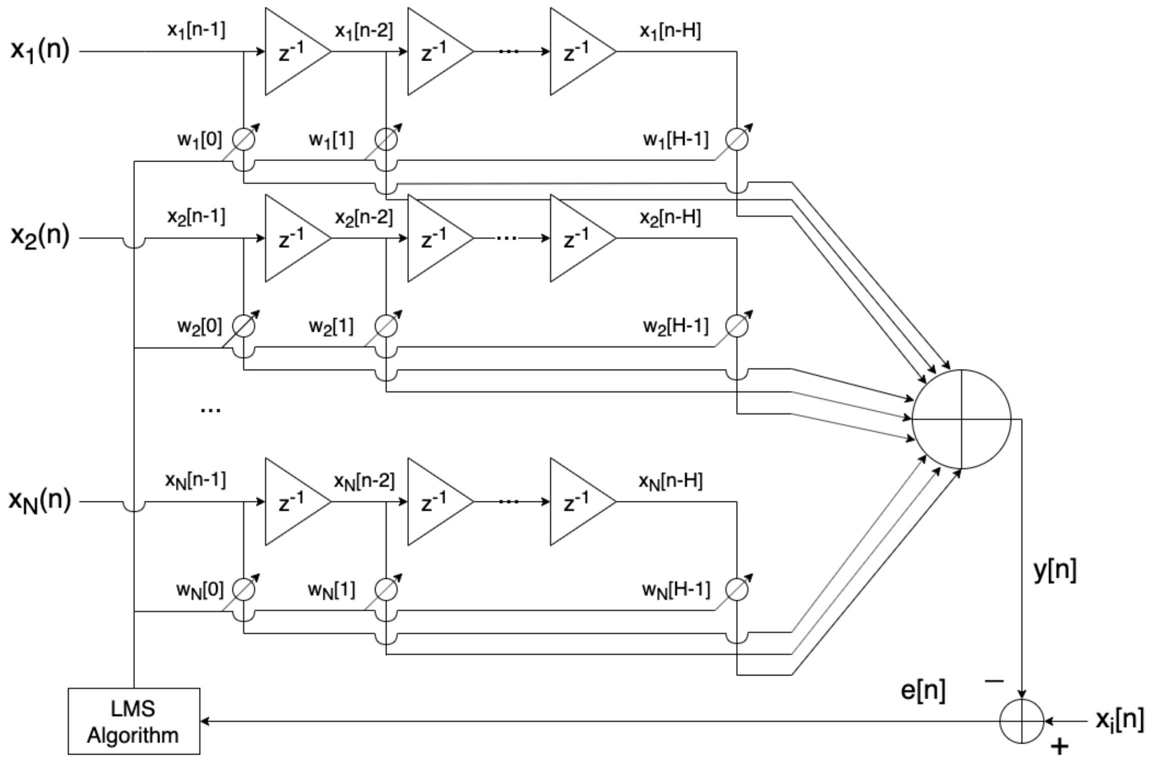


Figure 4-3: Adaptive Transversal Filter and Linear Combiner

4.5 Procedure

Using the Wiener filter model discussed prior, we lay out the steps taken in training and evaluating the model. At a high level, our procedure begins with preparing the data to train the Wiener filter. We then take measures to find the best datasets for training weight coefficients by looking for significant changes. Determining significance is not straightforward, so we elaborate on an extensive process to elevate periods with stronger fluctuations. Afterwards, we feed those datasets into the LMS algorithm to obtain optimal weight coefficients. Because we are looking to predict temperature values for every sensor node, the number of weight coefficient sets is equivalent to the number of nodes. We end with comparing weight coefficient distributions among those sets and evaluating our model by using each weight set to predict future temperature signal values.

4.5.1 Data Preprocessing

Wiener filters generally require priori knowledge of the desired signal. However, with the assumption that the desired signal is ergodic, time averages of the signals can be used in lieu of having the priori. Therefore, the data is preprocessed into discretized time averages before being fed into the LMS algorithm and filter model. Moreover, time averages overcome the lack of time synchronous measurements between different sensor nodes as mentioned in subsection 3.2.2, providing measurement normalization.

4.5.2 Locating Training Spots

Through a rigorous process, we deliberately identify periods of data to best train our model. Naturally, temperatures fluctuate throughout the day, noticeably rising while the sun is out and falling throughout the night. As such, we suspect that there are certain time periods where an event has occurred to better train our model than others time periods where no event has occurred. Therefore, before training weight coefficients, we search through the data for such events that act as a unit input to our system and our filter. However, the duration of an event and the extent that measurements fluctuate during an event are not obvious. For that reason, we investigate multiple temporal resolutions of fluctuations by parsing datasets and measuring the temperature fluctuations between data points separated by various time deltas.

Time Deltas

We expect significant changes within the time frame of an hour or less to indicate that an event has occurred. The exact length of time at which an event occurs is unknown, so we look for quick swings in temperature at varying temporal resolutions from one minute to one hour. Doing this helps avoid misclassification of random noise at high temporal resolutions and overall daily temperature trends at low temporal resolutions as significant events. With the data in the form of time averages from the process detailed in subsection 4.5.1, we iterate through each node's temperature data and compute the temperature changes for varying time deltas ranging from temperatures measured one minute apart to those measured one hour apart. In doing so, we obtain multiple sets of temperature fluctuations for each node's temperature data.

Significant Periods

Determining whether one time period is significant enough to indicate an event is not trivial. However, we can use the following observations to quantify significance.

1. If an event has occurred, we expect it to function as a unit impulse to our system and reflect significant fluctuations for its time period regardless of the temporal resolution of the fluctuations; significant fluctuations between data points a few minutes apart will aggregate and manifest between data points further apart. For that reason, we can determine significance in a manner self-contained to time deltas. We aggregate time period significance over all time deltas to cause events to stand out.

2. Distinct sensor nodes may experience varying magnitudes of temperature change for the same time period. One node's most significant temperature difference may be considered small in comparison to a node in a more volatile location. As such, we compare a node's temperature fluctuation only to other fluctuations of that same node to determine significance.

3. Temperature fluctuations in distinct locations in the lab area may be negatively correlated. For example, temperature changes in one part of the lab may trigger heating and cooling vents to counteract them. Thus, a sensor node at that part of the room may experience a temperature change opposite to one near a vent. Therefore, we are concerned more with the magnitude of temperature differences rather than their values.

With the aforementioned premises, we have scoped the problem of locating significant events to ascertaining what fluctuations qualify as significant for each sensor node. For this problem, we employ standard deviation as our main statistic. Within each set of temperature fluctuation magnitudes for a particular sensor node, we compute the standard deviation. Thereafter, we quantify a time period's significance by its fluctuations' deviations from the mean fluctuation.

Procedure To Locate Training Spots

The procedure to locate training spots is laid out in pseudocode in algorithm 1, and an example plot of accumulating standard deviations by following the procedure can be found in Figure 4-4. Peaks in the plot indicate the points in time where an event has most likely occurred to cause significant fluctuations in the data, so we train our model with data from the those time periods. However, troughs indicate time periods where temperature fluctuations were lower than average. With multiple sets of temperature fluctuations as discussed in section 4.5.2 and a way to quantify significance, we detail our procedure for locating training spots in the dataset at a high level in three steps below.

1. Separate all fluctuation data sets according to time deltas. Accumulate each time delta's averages of significance values.

2. To get each time delta's averages of significance values, separate the time delta's fluctuations set by sensor nodes. Compute the time period deviations for each node, and average over all nodes.

3. To compute the time period deviations for each node, compute the standard deviation of fluctuations for that node for the corresponding time delta, and divide every fluctuation value by that standard deviation.

```
**Data:** Various temperature fluctuation magnitudes for different time deltas for
          each sensor node
**Result:** Finding time periods to train the adaptive Wiener filter.
cumulativeDeviation = []
for time delta $d$ do
    totalDeviations = []
    for sensor node $n$ do
        fluctuations = get_fluctuations($d$, $n$)
        stDev = get_stDev(fluctuations)
        deviations = fluctuations/stDev
        totalDeviations.append(deviations)
    end
    averageDeviation = average(totalDeviations)
    cumulativeDeviation.append(averageDeviation)
end
```

Algorithm 1: Algorithm to locate training periods

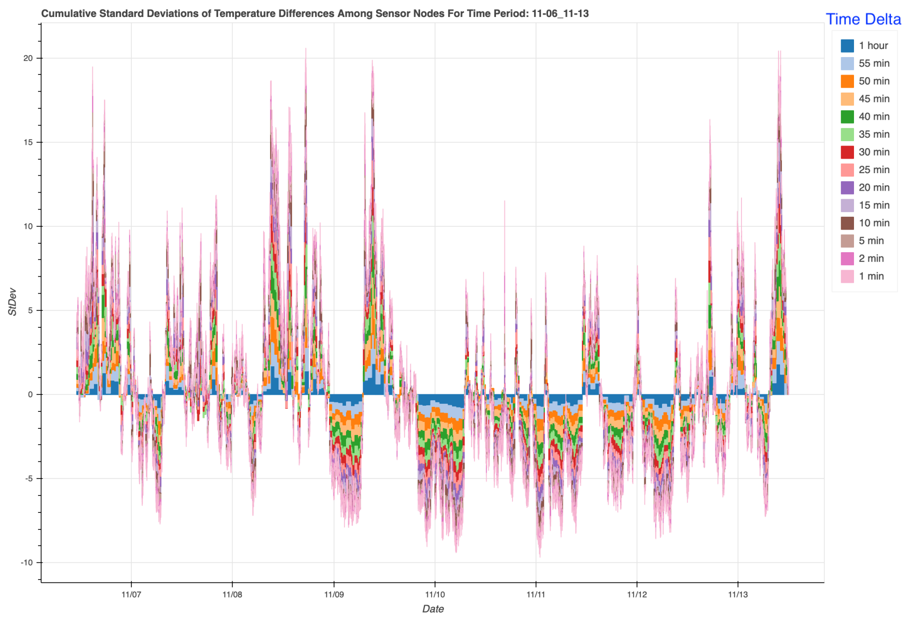


Figure 4-4: Plot of Cumulative Standard Deviations Over Time

4.5.3 LMS Filter Weights

After locating suitable time periods to train the model on, we proceeded to use the well-regarded LMS algorithm discussed in section 4.2. Because we are looking to predict temperature values for each sensor node, the LMS algorithm is run and optimized per sensor node temperatures, generating a set of weights for every specific temperature predictor.

4.5.4 Weight Heat Maps

The weights converged on from running the LMS algorithm may provide insight into how fluctuations from one location in the lab space affects future fluctuations in other locations. Lower weight coefficients signify that the corresponding input does not contribute towards predicting, whereas high coefficients indicate that the input highly correlates with the desired signal. To better visualize weight values, we generate their heat maps, such as the one shown in Figure 4-5. Each cell in the heat map corresponds to a particular input to the model. Recalling the model structure from section 4.5, inputs are previous temperature values, as represented by columns, recorded for each sensor node, as represented by rows.

4.5.5 Temperature Predicting

The final step in our procedure is to evaluate our temperature predicting models. With weight coefficients trained on a selected time period, we proceed to estimate the measured data from the sensors immediately following that time period. The structure for predicting temperature closely follows that of Figure 4-3, except the errors are not fed back into modifying weight coefficients. The data used for prediction range from an hour following the training period to an entire day following the training period, respectively resulting in the short term and long term performances. In addition to plotting the errors from prediction, we also measure the root mean square error (RMSE) of the errors to quantify the performance of our predictor. Figure 4-6 below shows an example plot of errors for one sensor node's temperature predictor.

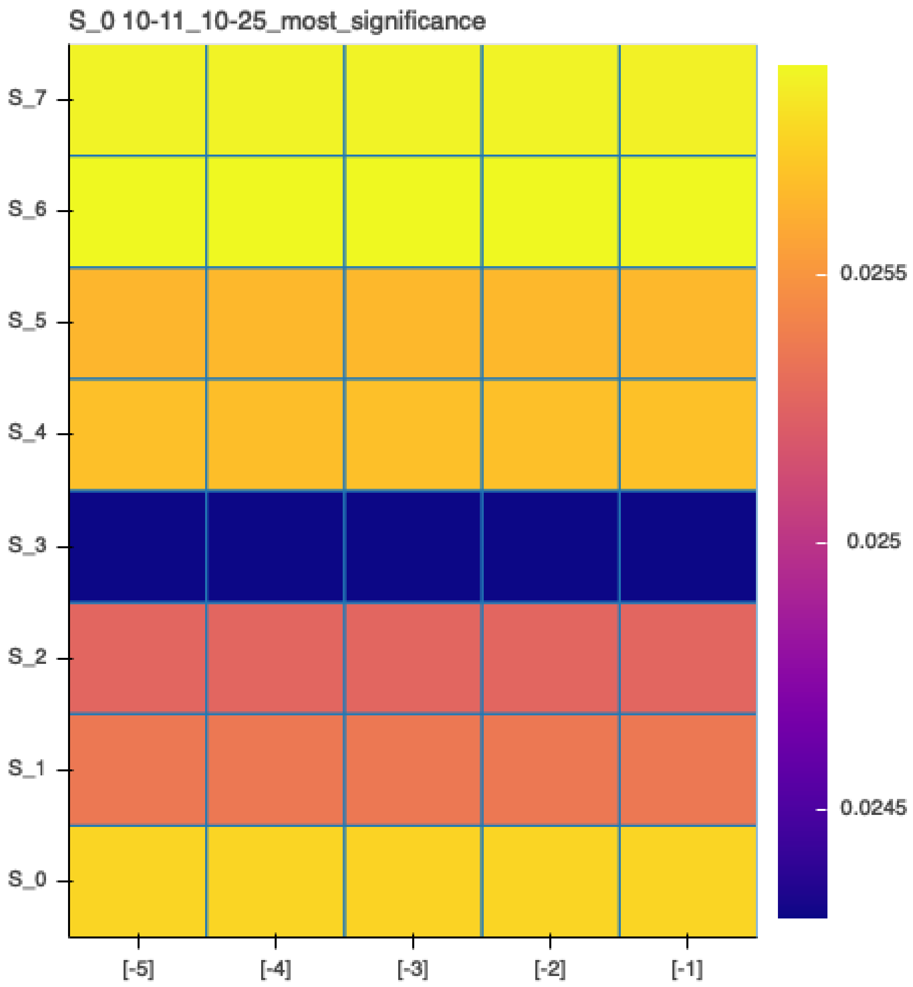


Figure 4-5: Example Heat Map for 8 sensor nodes with history length 5

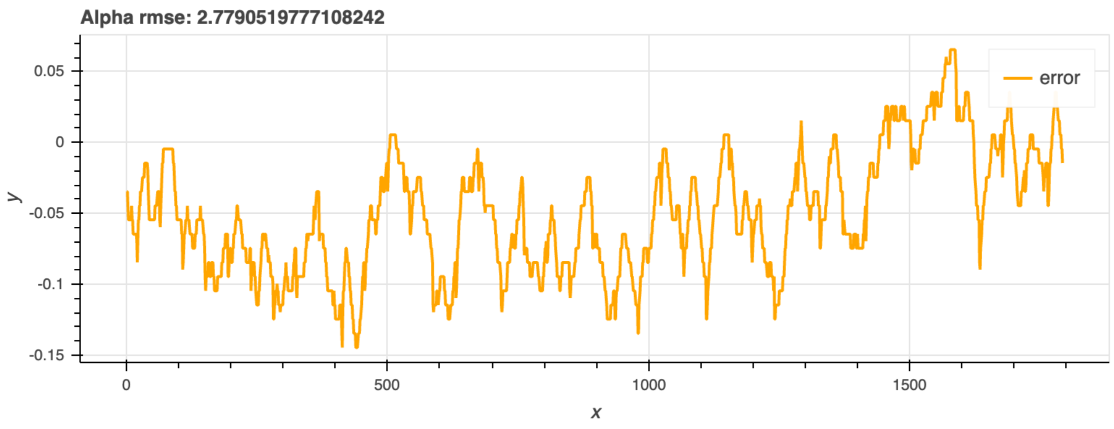


Figure 4-6: Example Error plot with RMSE calculated

Chapter 5

Performance

With our data models and system infrastructure in place, we began to monitor the lab environment and measure prediction performance. The data was collected by the eight sensor nodes in three separate stages. The first stage had half of the sensors clustered together in one location and the other half unclustered in a semi-circle formation from the cluster. In the next, we split the cluster in two and left the other sensors in place, forming a well spread out semi-clustered formation. In the final stage, we dispersed the sensors even further so that they nearly spanned the entire lab space. In each setup, we map the node layout, discuss the temperatures collected, different training spots, and heatmaps of weight coefficients, and ultimately evaluate the predictors' short-term performance over one hour and long-term performance over one day. Afterwards, we include further implementation details and discuss our findings from our analysis.

5.1 Clustered Setup

In the clustered setup, we placed our WSN nodes at locations depicted below in Figure 5-1. By having nodes clustered together, we anticipated that fluctuations from these nodes would have a greater impact on the future fluctuations of these nodes when compared to the fluctuations originating from the other locations. In addition, the node on the top left of the diagram is right next to a window. As such, we anticipated its temperature values to significantly differ from the rest.

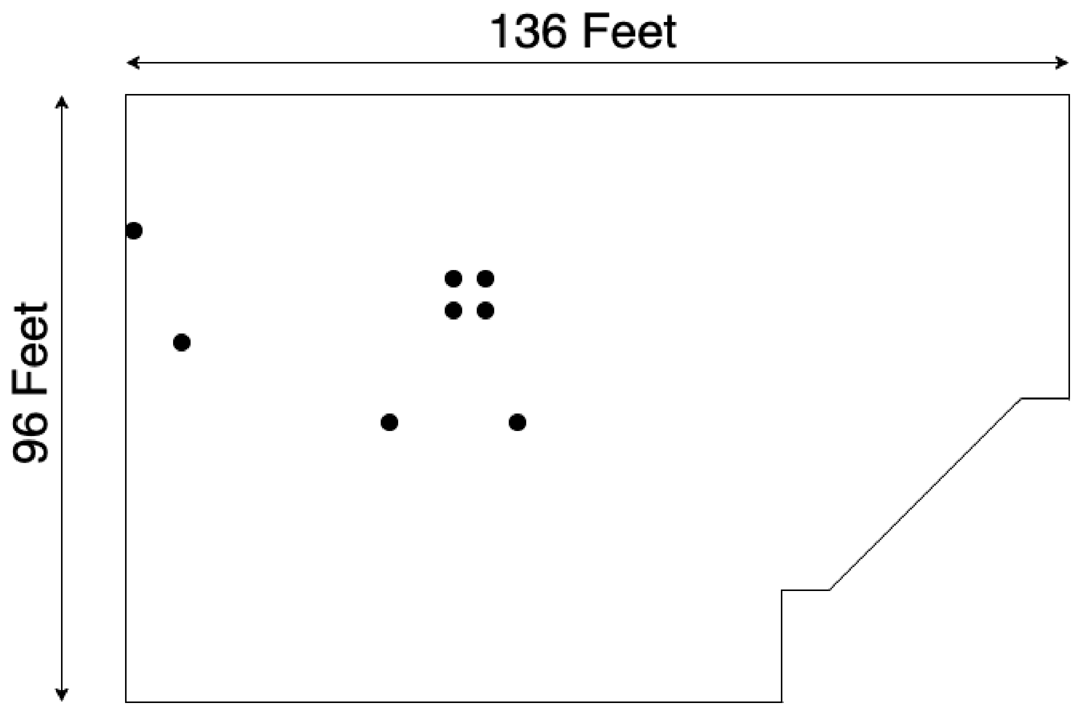


Figure 5-1: Clustered Formation of Wireless Sensor Nodes indicated by circles

5.1.1 Temperature Data Results

Figure B-1 shows plots of the data collected from the all the nodes in the clustered setup. Figure B-2 and Figure B-3 depicts the unclustered node temperature measurements and the clustered node temperature measurements, respectively. From these plots, there are a number of key aspects to note.

1. During the data collection process, two sensor nodes went down, hence the two visibly linear lines in the first half. As a result, we will avoid the time periods in between from training and evaluating.

2. As expected, the node by the window consistently has temperature measurements lower than the rest. To make sure the sensor isn't faulty, we decided to swap it with another node in the next setup.

3. Temperature signals from the nodes have highly similar trends in how their temperature measurements fluctuate throughout the measuring period. It appears that the temperature signals are close to being the same signal at different offsets, but they are still distinct. As such, we expect our input signals to have varying weights for estimating the desired signal. More specifically, we expect the past values of the desired signal to contribute the most to the estimation.

4. Temperatures are easily distinguishable between nodes not part of the cluster, as shown in Figure B-2. Yet in Figure B-3, one can hardly distinguish temperatures measured by clustered nodes. From this, we see that different locations within the lab space experience similar temperature fluctuation shapes. As such, we spaced out nodes further during the next temperature collection period.

5.1.2 Training Spots

Figure B-4 shows the cumulative standard deviations of all time deltas generated by the process discussed in subsection 4.5.2. For this setup, we look at the top three most significant time periods indicated by peaks, an arbitrary average significance period indicated by the rest level, and the least significant period indicated by troughs. We training on multiple data sets to determine fluctuation significance utility in signal estimation.

5.1.3 Weight Heatmaps

Feeding the top three most significant, an average significance, and the least significant time periods in Figure B-4 into the LMS algorithm, we obtain heatmaps in Figure B-5 through Figure B-9 in Appendix B.

A couple of realizations can be gleaned from these heatmaps. The heatmaps of weights among all the temperature predictors for each time period look identical, but the actual weight values are not the same. This counters our speculations of what the filter weights would look like in subsection 5.1.1. For the temperature signal being estimated, we expected its own past values to contribute the most towards its estimation, hence having the highest weight coefficients. Moreover the weight coefficients for the historical inputs are nearly identical. This is because temperature measurements a few seconds apart don't fluctuate greatly. Thus, the inputs are close to being the same value, causing weights to converge to similar values for a linear model.

5.1.4 Temperature Predictor Performance

Using the weight coefficients trained from the most significant, the average significance, and the least significant time periods, we evaluate the respective temperature predictors for each node.

We evaluate over the short term of one hour and the long term of one day, and generate the short term error plots in Figure B-10, Figure B-11, and Figure B-12 and long term error plots in Figure B-13, Figure B-14, and Figure B-15. The respective RMS errors can be found in Table A.1 and Table A.2.

The results indicate that most temperature predictors trained over data from the least significant time period performed the best in predicting temperatures one hour into the future. Over the period of a day, a majority of the temperature predictors performed better when trained over the most significant time period. Contrary to expectations, periods of high fluctuations do not act as a unit impulse to our system in the short-run.

5.2 Semi-Clustered Setup

In the semi-clustered setup, we placed our WSN nodes at locations depicted below in Figure 5-2. As seen in the clustered setup, temperature values between nodes were fairly equal due to half of them clustering. By splitting up the cluster, we expect more variation in temperatures recorded. Also to ensure that the sensor node by the window wasn't faulty, we swapped it with another halfway through, hence there are two sets of temperature measurements for this set up.

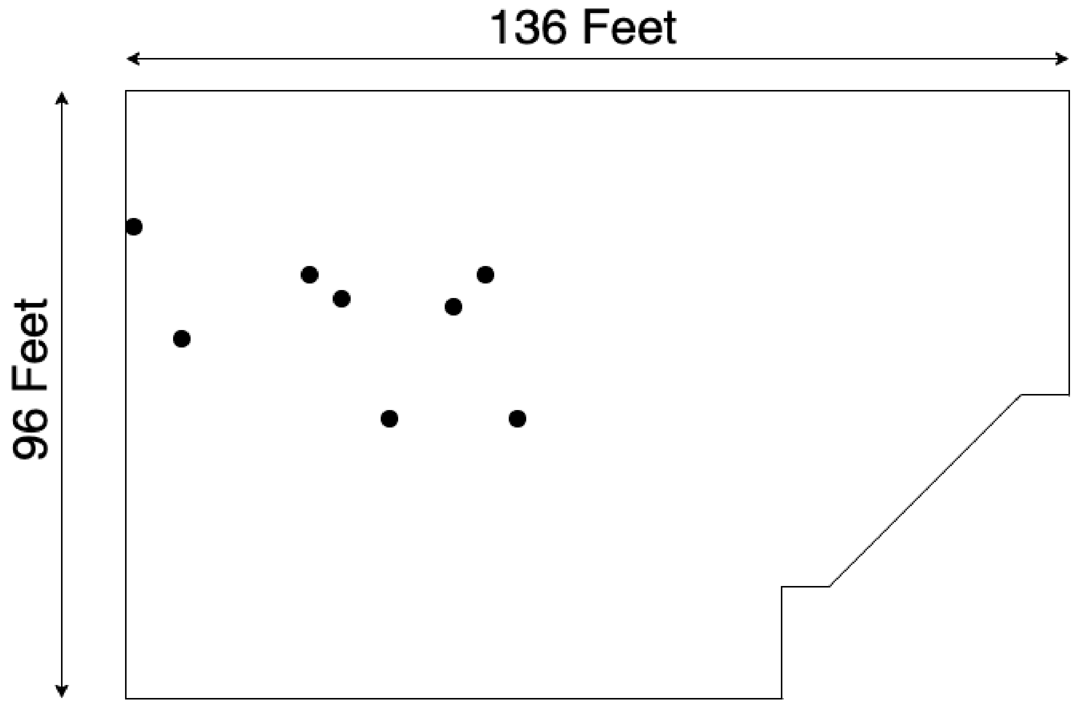


Figure 5-2: Semi-Clustered Formation of Wireless Sensor Nodes, indicated by circles

5.2.1 Temperature Data Collected

Figure B-16 and Figure B-17 show plots of the data collected during the semi-clustered setup. From the data collected, we notice the following.

1. Between the two data collection periods in the semi-clustered setup, two nodes, one of which was next to the window, had been swapped. Even after the swap, the node by the window consistently experienced colder temperatures. Therefore, we have confidence that sensors are reporting correct temperatures.

2. At times, temperature measurement trends between the nodes agree with each other, yet unlike the clustered setup, there are periods where trends clearly diverge. As such, we hope to see more variety in weight ratios resulting from the LMS algorithm.

3. There are more instances where temperature measurements cross, and signals appear more distinct in this setup compared to the previous setup, so we hope

to glean more knowledge about how fluctuations at distinct locations affect each other.

5.2.2 Training Spots

Figure B-18 and Figure B-19 show the cumulative standard deviations for the semi-clustered setup. For each, we look at the most significant time period, an arbitrary average significance period, and the least significant period. By training and evaluating on each, we again test the importance of fluctuation significance in training our model.

5.2.3 Weight Heatmaps

Feeding the time periods corresponding to the most significant, average significance, and least significant time periods in Figure B-18 and Figure B-19 into the LMS algorithm, we obtain the following weight distributions displayed as heatmaps in Figure B-20 through Figure B-25 in Appendix B.

Despite higher distinctiveness between temperature signals, the heatmaps of weights among all the temperature predictors for each time period still look identical. The actual weight values are not the same, but this refutes our expectation that the LMS algorithm would produce larger weight coefficients for the inputs belonging to the desired signal. We proceed to make temperature signals more distinct in the next setup by isolating sensor nodes even further and spreading them across the entire lab space.

5.2.4 Temperature Predictor Performance

The performance error plots can be found from Figure B-26 through Figure B-37. The respective RMS errors can be found in Table A.3, Table A.4, Table A.5 and Table A.6.

Similar to the results from the clustered setup, short term performance favors training over periods of least significance. However, for long term performance, training over periods of average fluctuation significance leads to better performance.

5.3 Spread Out Setup

When splitting up the clusters from the clustered to semi-clustered setups, temperature signals were more easily distinguishable. Following this idea, we proceeded to spread out the node sensors even further as depicted below in Figure 5-3. We predict that temperature signals will be even more distinct.

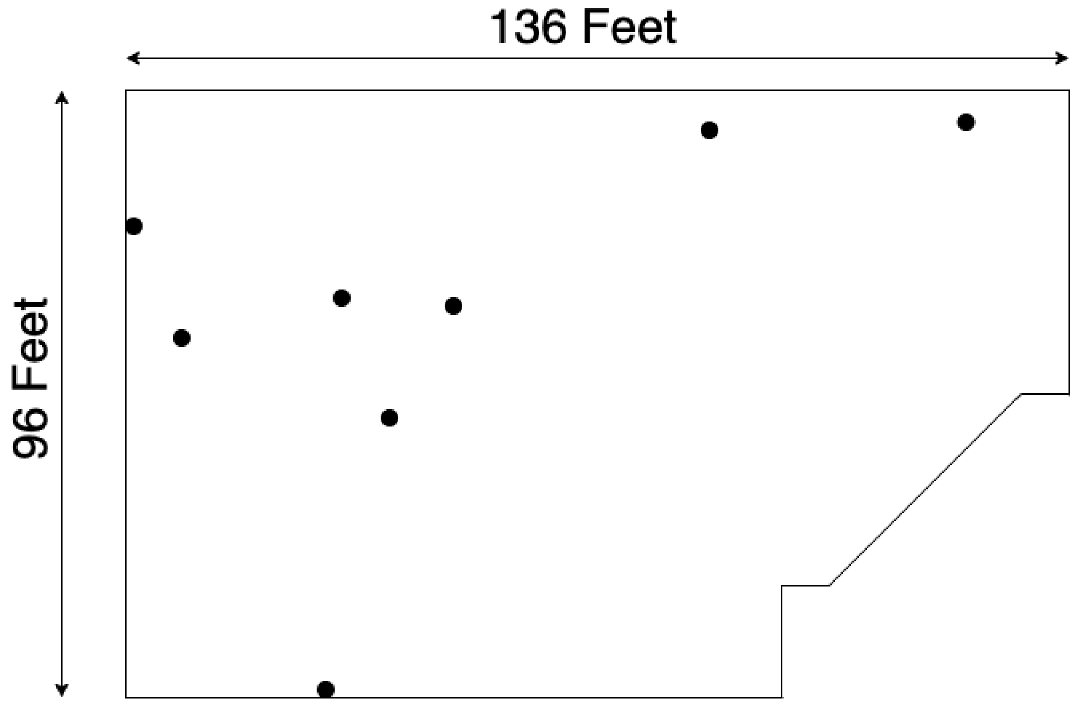


Figure 5-3: Spread Out Formation of Wireless Sensor Nodes, indicated by circles

5.3.1 Temperature Data Collected

Figure B-38 shows a plot of the data collected during the spread out setup, and the following details can be observed:

1. One temperature signal is extremely volatile, and experienced stronger fluctuations than other nodes for a significant portion of the data collection period.

2. Signals are more easily distinguishable than in either the clustered or semi-clustered setups.

3. For the most part, signals undergo similar trends.

5.3.2 Training Spots

Figure B-39 shows the cumulative standard deviations for the spread out setup. Continuing to test the importance of fluctuation significance, we look at the most significant time period, an arbitrary average significance period, and the least significant period.

5.3.3 Weight Heatmaps

Feeding the time periods corresponding to the most significant, average significance, and least significant time periods in Figure B-39 into the LMS algorithm, we obtain the following weight distributions displayed as heatmaps in Figure B-40 through Figure B-42 in Appendix B.

By spreading out the nodes even further, we see that the weights are less uniform among the different node inputs. However, the weight ratios among the temperature predictors for each time period are still indistinguishable.

5.3.4 Temperature Predictor Performance

The performance error plots can be found Figure B-43 through Figure B-48. The respective RMS errors can be found in Table A.7 and Table A.8.

Even with more distinct signals, training over the least significant time period results in lower RMS error values in the short run. With parallels to the semi-clustered out setup, the performance over the long run is better when training over a period of average fluctuations.

5.4 Implementation Details and Results

For our procedure, there were a number of parameters that we needed to set to process the data, feed it into the LMS algorithm, and evaluate performance.

We preprocessed our data with time averages every two seconds. By doing so, we capture most of the measurements while normalizing the timing between them. As mentioned in subsection 3.2.2, nodes measured at a rate of 45 measurements per minute. As such, two second averages were the highest temporal resolution possible. Taking longer averages would oppose our investigation of high temporal trends in temperature signals.

For our use case, we did not fine tune the learning rate and weight initialization parameters fed into the LMS algorithm. Our use of the LMS algorithm was primarily to converge on weights that characterized the system, so we initialized the weights to zero. As it turns out in our configuration, the weight ratios of filter inputs among the different temperature predictors were similar, hence the heat maps looked similar across temperature predictors for a particular data set. By nature of the linear model and dealing with room temperature fluctuations, the weight ratios corresponded to ratios between the averages of the temperature signals.

When feeding our data sets into the LMS algorithm, our time steps between input vectors was also two seconds. In other words, we fed our time averages one after another for the model to learn high temporal trends. The history length that we targeted for our filter model was primarily five measurements in the past. Together, this meant that our input vector contained temperatures from all nodes two four, six, eight, and ten seconds in the past to estimate the next temperature value.

As shown in the results of our temperature predictors' performances, ultimately we could not use high temporal trends in our temperature signals for prediction. In fact, using periods of low fluctuations to predict in the near future worked better because room temperature fluctuations will not change significantly step-by-step. However, over a longer period of time, training over periods of average fluctuation significance and high fluctuation significance predicted better. Despite this, our estimations significantly deviated from the actual measurements, suggesting that a linear model does not characterize temperature fluctuations well. On the other hand, as supported by our temperature collection over three distinct setups, temperature signals are more distinguishable the further apart they are in distance.

Chapter 6

Conclusion

In this paper, we proposed a temperature prediction scheme that focused on learning how temperature fluctuations at distinct locations in a room interrelate and pervasively affect future temperatures. Through analyzing the data collected by our custom wireless sensor nodes in three distinct configurations, we computed fluctuation significance for periods of measurements to train the temperature predictors. Using a transversal filter and linear combiner model, we trained weight coefficients via the LMS algorithm and evaluated performance over data succeeding the training data.

Our findings indicate that temperature signals at two different locations are visibly more distinguishable the further apart they are. Training our transversal filter and linear combiner model with data sets containing high temporal trends of high fluctuation significance does not predict temperatures better than trends of no or little fluctuation significance. Predicting these temperature signals with constant weight coefficients results in significant deviation between the estimated and actual signals in the long run, and thus our model does not characterize how temperature fluctuations affect each other.

Our system architecture provides a convenient and scalable process to collect data, allowing numerous sensor nodes to function independently of each other. The data infrastructure is an extensible system which stores measurements in lightweight databases and allows monitoring in real time. Moreover, each sensor node can capture and report several environmental factors with as few as three components. Potential future work for our hardware solution includes completing a printed circuit board form factor, further reducing the node's footprint.

Chapter 7

Supplemental Work

In addition to the work detailed above, our research includes other explorations that looked to better our solution by lowering the cost, decreasing the footprint, as well as utilizing the state-of-the-art hardware. Though we did not manage to make the improvements we wanted, they are suggestions for possible future work related to the project.

7.1 Design

One of the original premises of our work was to create a small-footprint solution that would allow researchers to monitor the environmental factors in a lab setting. Our desire was that the solution would be compact and work out-of-the-box. We iterated on several printed circuit board (PCB) prototypes of the wireless sensor nodes. We managed to narrow down the exact chips and components necessary for the nodes to work and placed them all on one circuit board. Moreover, we attempted to power the nodes with a battery to facilitate placement unrestricted by power outlets. However, issues such as improper thermal isolation between the microcontroller and BME680 chip, long lead times, misconnected pins, and brownouts from improper current draw were unexpected delays in the project. Thus, we forewent the PCB design and proceeded to collect data with the breadboard design powered through a power outlet, depicted in Figure 3-3.

7.2 ESP32

Instead of using the ESP8266, we originally used its successor in Espressif's line of System on a Chip, the ESP32. With the ESP32, we attempted to program the sensor nodes with MicroPython, which is a microcontroller compatible version of the popular programming language, Python. However, due to limited robust functionality with the ESP32 at the time, we opted to use Arduino instead. When attempting to collect data with the ESP32, our sensor nodes occasionally malfunctioned and disconnected from Wi-Fi, which is undesired for frequent data collection. We suspected that the problem arose from package libraries that we used to interface the microcontroller with the BME680 environmental sensor. With these issues, we decided to switch from using the ESP32 to using the ESP8266 to progress the project.

Appendix A

Tables

Predictor	Least Significant	Average Significance	Most Significant
S_0	1.363	6.460	9.715
S_1	1.111	6.135	10.346
S_2	2.504	11.694	5.789
S_3	4.044	14.229	6.479
S_4	0.541	2.506	8.507
S_5	0.786	4.058	10.035
S_6	1.133	0.949	11.553
S_7	2.170	1.402	6.126

Table A.1: Clustered Setup Short Term Performance RMS Error

Predictor	Least Significant	Average Significance	Most Significant
S_0	95.172	85.189	207.25
S_1	162.334	141.002	62.743
S_2	113.854	88.775	175.905
S_3	572.923	479.900	437.840
S_4	136.238	127.436	69.028
S_5	144.440	133.596	72.353
S_6	146.903	148.660	75.969
S_7	148.279	154.841	104.199

Table A.2: Clustered Setup Long Term Performance RMS Error

Predictor	Least Significant	Average Significance	Most Significant
S_0	1.403	3.374	2.779
S_1	0.580	1.538	4.085
S_2	1.382	2.818	4.824
S_3	3.648	6.883	13.770
S_4	1.158	1.705	6.380
S_5	0.971	1.993	5.754
S_6	1.283	3.265	9.591
S_7	1.201	2.230	11.112

Table A.3: Semi-Clustered Setup Short Term Performance RMS Error Before Swap

Predictor	Least Significant	Average Significance	Most Significant
S_0	1.481	4.780	12.817
S_1	0.999	9.339	20.301
S_2	1.204	3.590	6.268
S_3	1.511	10.738	3.490
S_4	0.525	5.405	21.590
S_5	0.863	5.023	20.201
S_6	1.115	1.597	22.861
S_7	1.098	1.319	20.858

Table A.4: Semi-Clustered Setup Short Term Performance RMS Error After Swap

Predictor	Least Significant	Average Significance	Most Significant
S_0	229.912	95.250	87.805
S_1	127.680	39.825	108.222
S_2	87.969	148.043	92.400
S_3	519.961	59.561	176.414
S_4	146.112	38.642	104.918
S_5	153.971	40.362	103.731
S_6	225.767	56.747	145.479
S_7	198.227	49.431	143.418

Table A.5: Semi-Clustered Setup Long Term Performance RMS Error Before Swap

Predictor	Least Significant	Average Significance	Most Significant
S_0	232.535	382.020	184.754
S_1	52.678	67.607	148.925
S_2	95.259	123.542	111.625
S_3	137.940	85.009	103.216
S_4	69.134	55.396	117.018
S_5	67.990	54.459	111.267
S_6	143.036	74.412	69.390
S_7	150.323	61.586	56.492

Table A.6: Semi-Clustered Setup Long Term Performance RMS Error After Swap

Predictor	Least Significant	Average Significance	Most Significant
S_0	2.670	18.553	2.525
S_1	0.634	1.851	4.637
S_2	4.665	11.697	14.770
S_3	0.948	1.489	8.242
S_4	1.462	2.638	4.124
S_5	1.417	9.448	2.611
S_6	1.533	3.551	3.805
S_7	1.481	17.098	18.096

Table A.7: Spread Out Setup Short Term Performance RMS Error

Predictor	Least Significant	Average Significance	Most Significant
S_0	334.178	191.647	267.156
S_1	83.314	78.758	113.642
S_2	247.266	180.196	316.570
S_3	158.587	147.471	211.170
S_4	122.698	101.639	164.973
S_5	114.840	190.319	152.497
S_6	152.006	184.996	309.632
S_7	549.459	1157.583	1132.655

Table A.8: Spread Out Setup Long Term Performance RMS Error

Appendix B

Figures

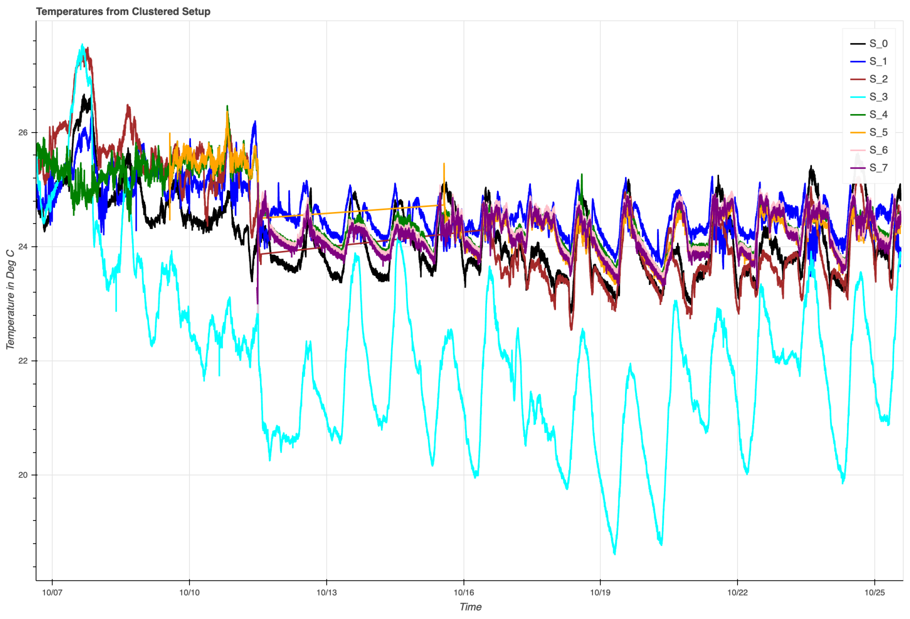


Figure B-1: Temperatures of Sensor nodes in Clustered Set up

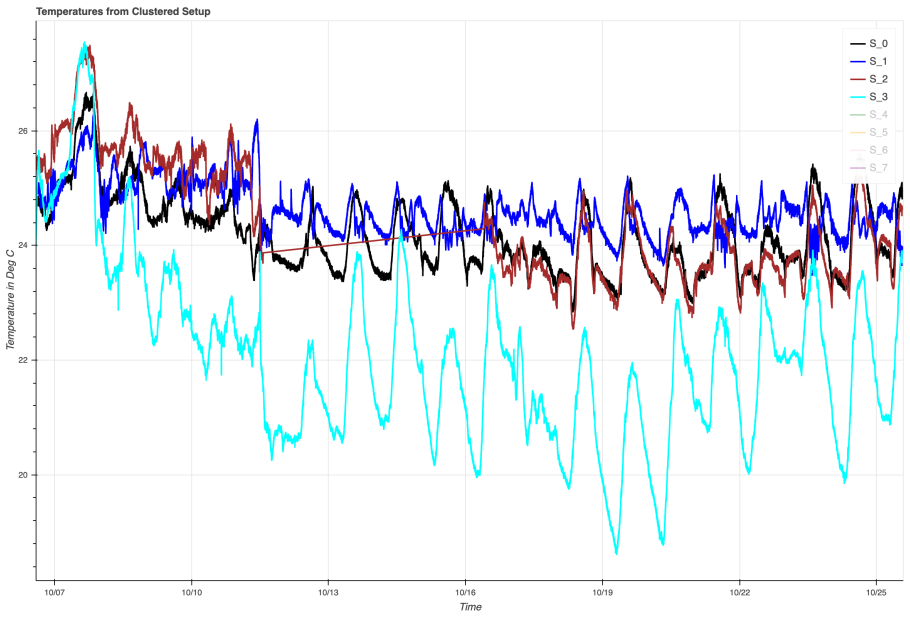


Figure B-2: Temperatures of Unclustered Sensor nodes in Clustered Set up

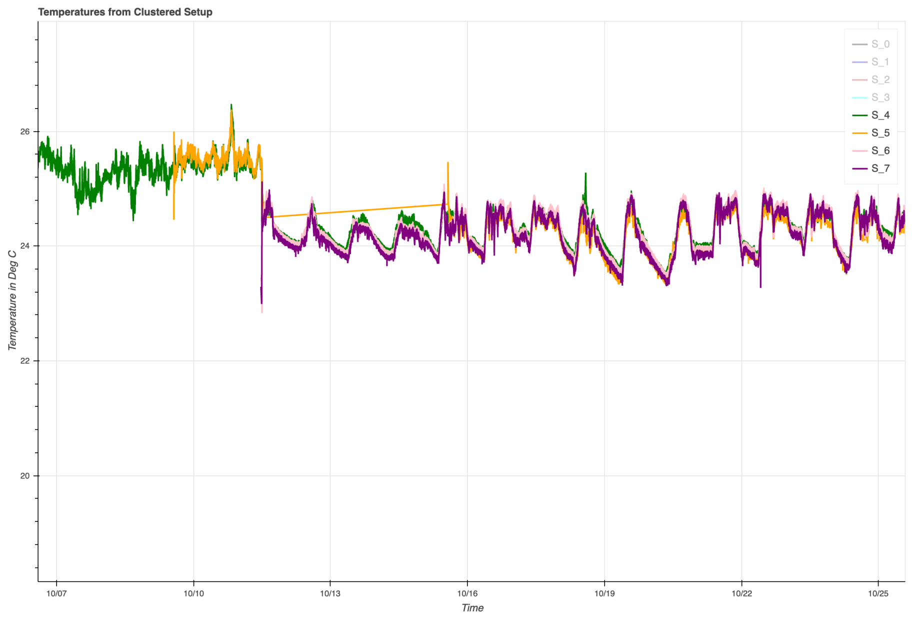


Figure B-3: Temperatures of Clustered Sensor nodes in Clustered Set up

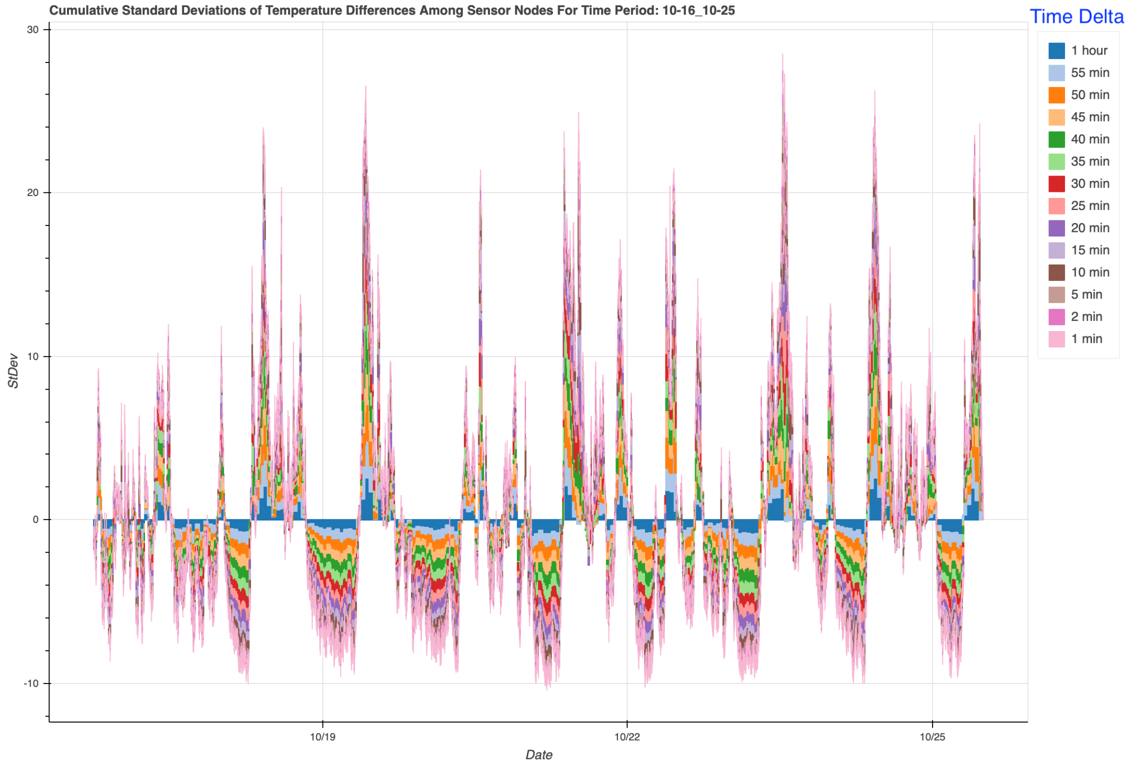
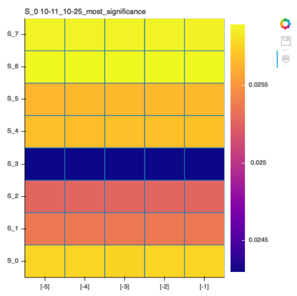
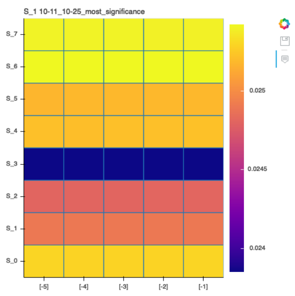


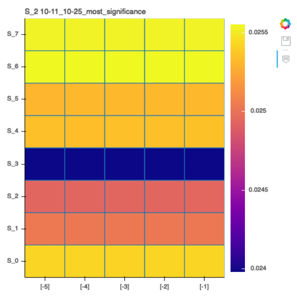
Figure B-4: Clustered Setup Cumulative Standard Deviations



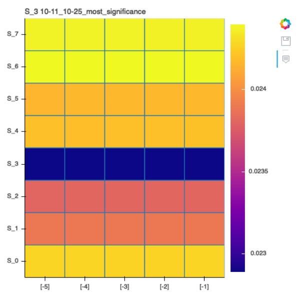
(a) S_0



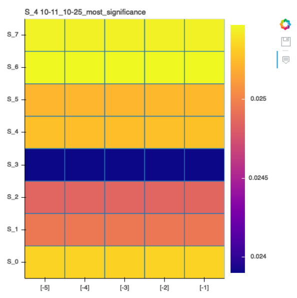
(b) S_1



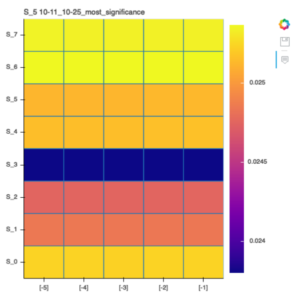
(c) S_2



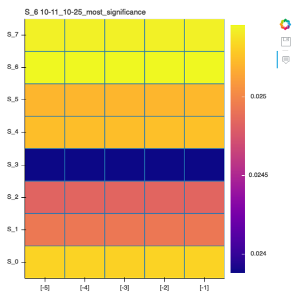
(d) S_3



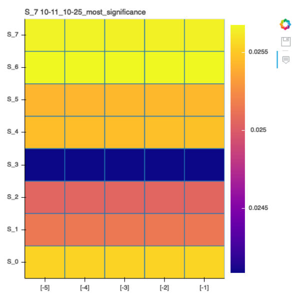
(e) S_4



(f) S_5

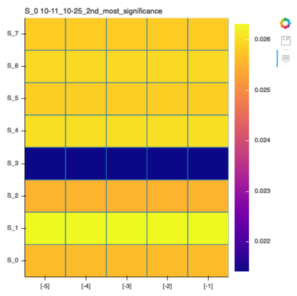


(g) S_6

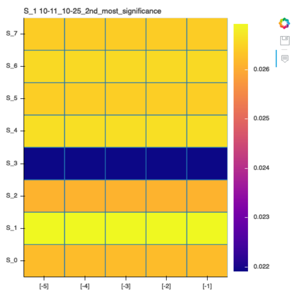


(h) S_7

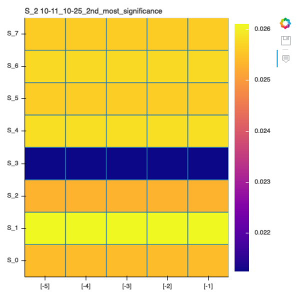
Figure B-5: Heatmaps from Most Significant Time Period



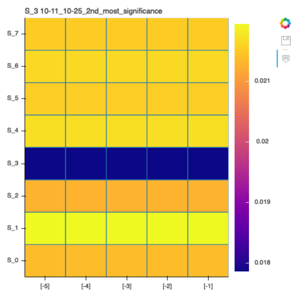
(a) S_0



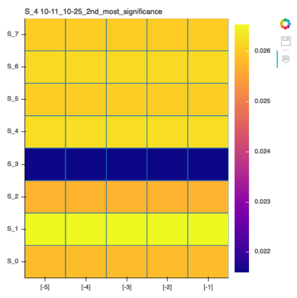
(b) S_1



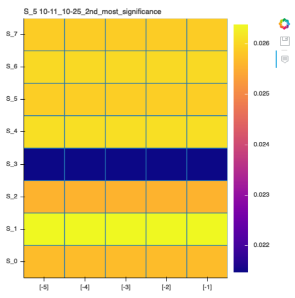
(c) S_2



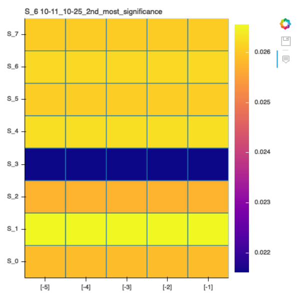
(d) S_3



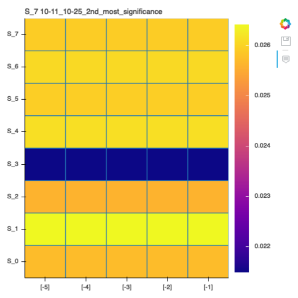
(e) S_4



(f) S_5

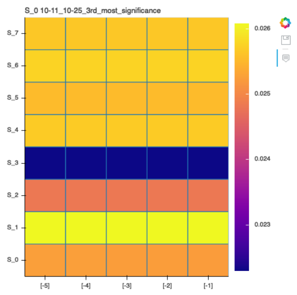


(g) S_6

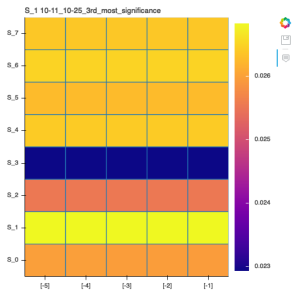


(h) S_7

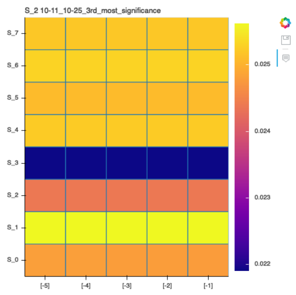
Figure B-6: Heatmaps from 2nd Most Significant Time Period



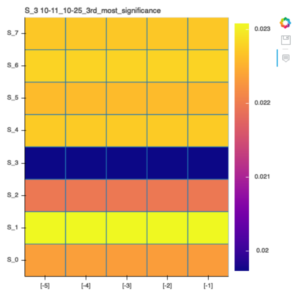
(a) S_0



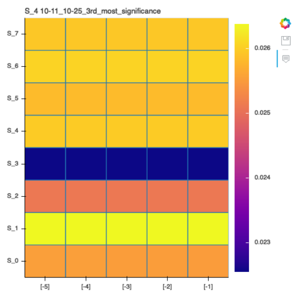
(b) S_1



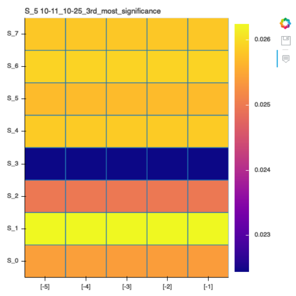
(c) S_2



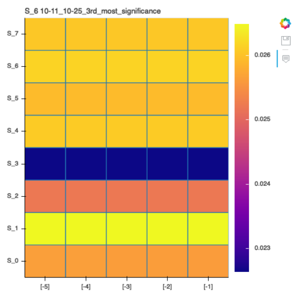
(d) S_3



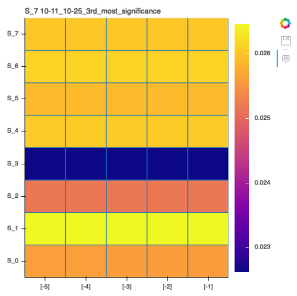
(e) S_4



(f) S_5

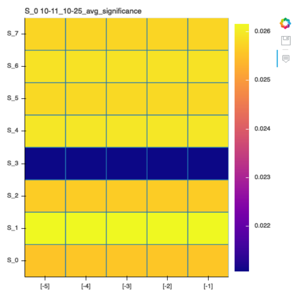


(g) S_6

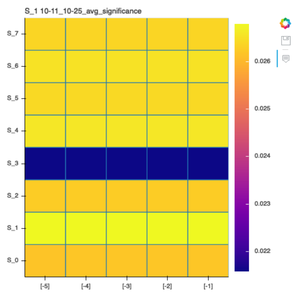


(h) S_7

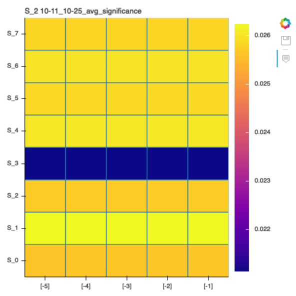
Figure B-7: Heatmaps from 3rd Most Significant Time Period



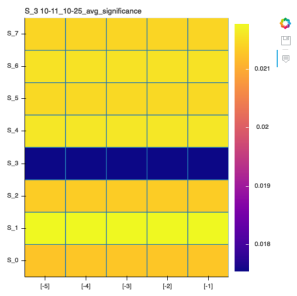
(a) S_0



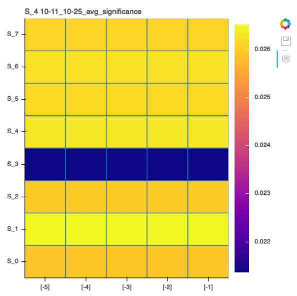
(b) S_1



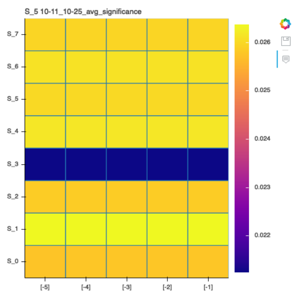
(c) S_2



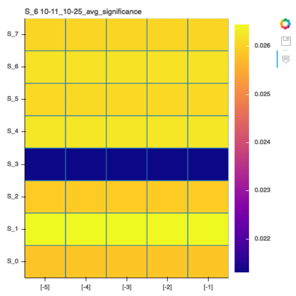
(d) S_3



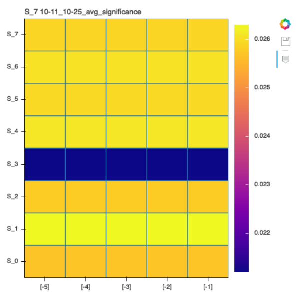
(e) S_4



(f) S_5

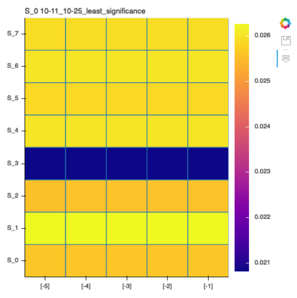


(g) S_6

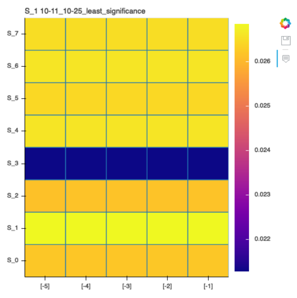


(h) S_7

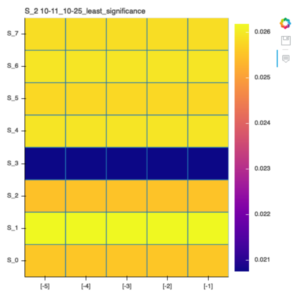
Figure B-8: Heatmaps from Average Significance Time Period



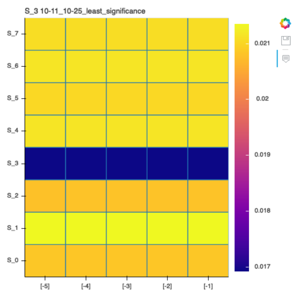
(a) S_0



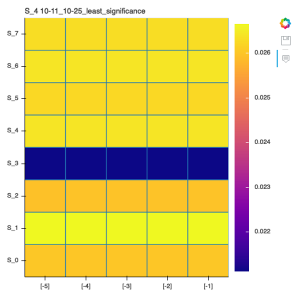
(b) S_1



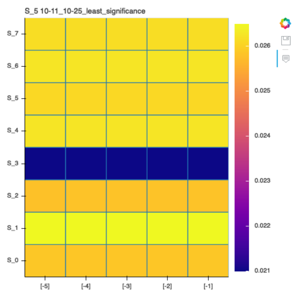
(c) S_2



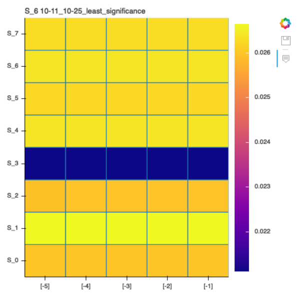
(d) S_3



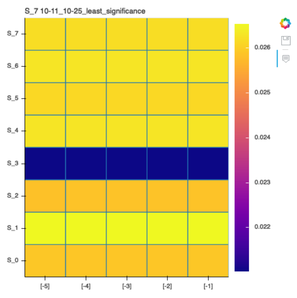
(e) S_4



(f) S_5

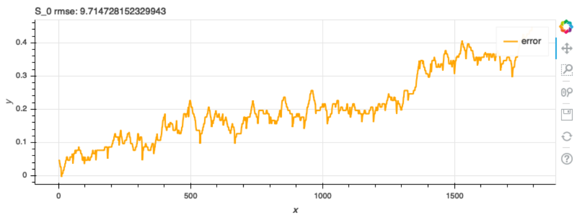


(g) S_6

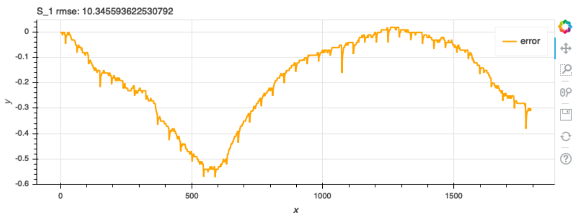


(h) S_7

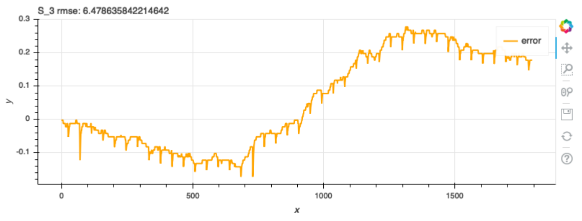
Figure B-9: Heatmaps from Least Significant Time Period



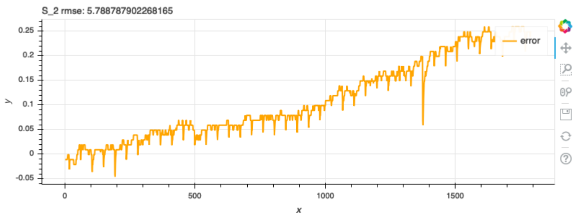
(a) S_0 Predictor



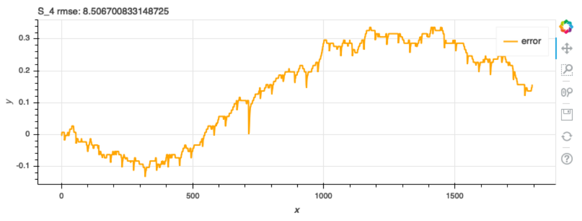
(b) S_1 Predictor



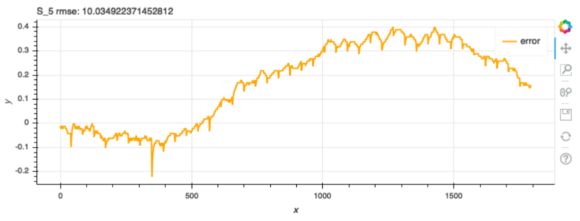
(c) S_2 Predictor



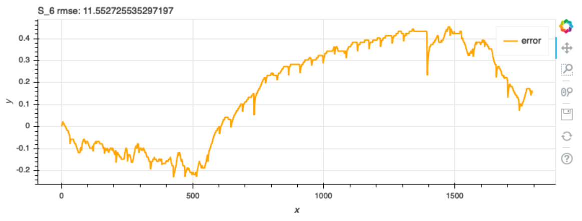
(d) S_3 Predictor



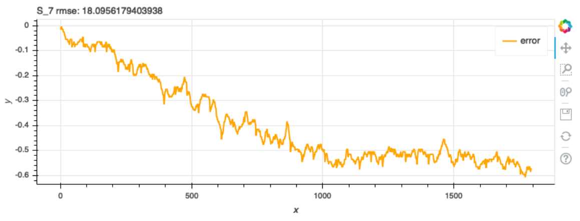
(e) S_4 Predictor



(f) S_5 Predictor

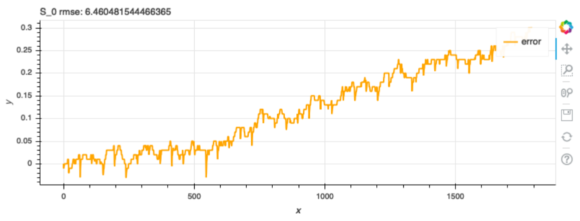


(g) S_6 Predictor

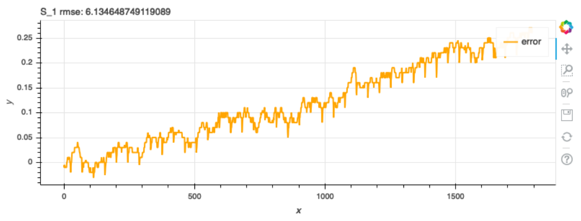


(h) S_7 Predictor

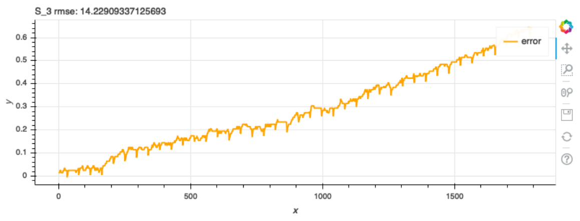
Figure B-10: Clustered Short Term Performances For Most Significant Time Period



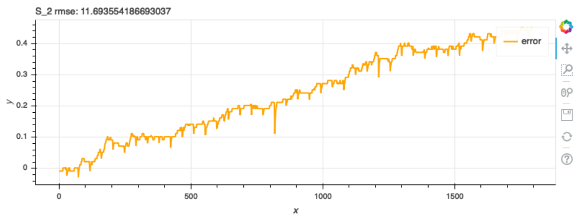
(a) S_0 Predictor



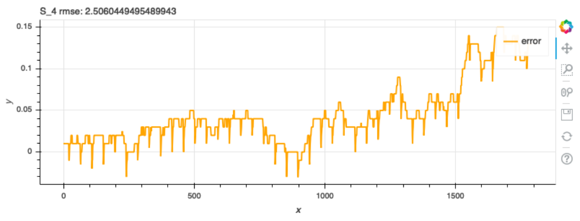
(b) S_1 Predictor



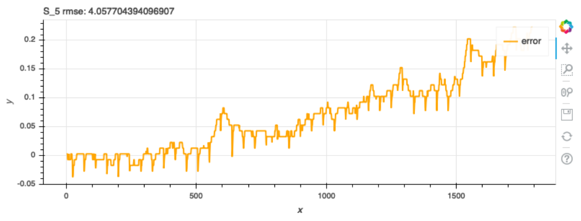
(c) S_2 Predictor



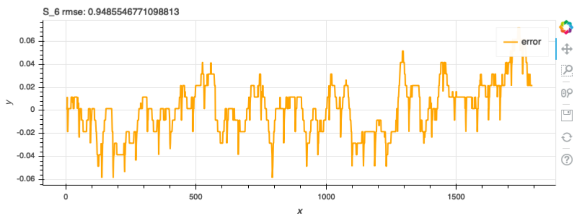
(d) S_3 Predictor



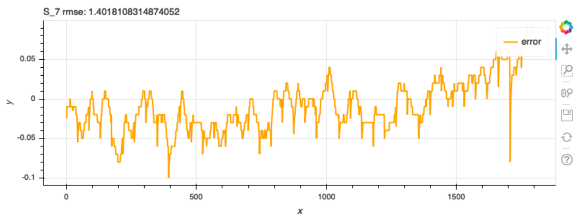
(e) S_4 Predictor



(f) S_5 Predictor

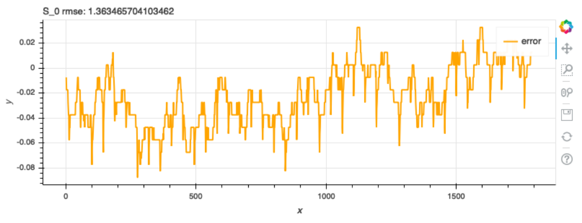


(g) S_6 Predictor

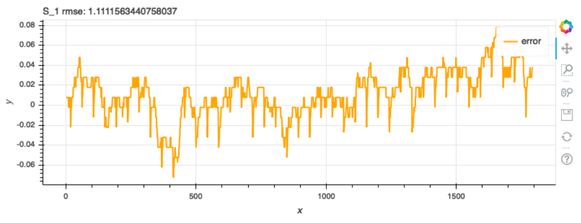


(h) S_7 Predictor

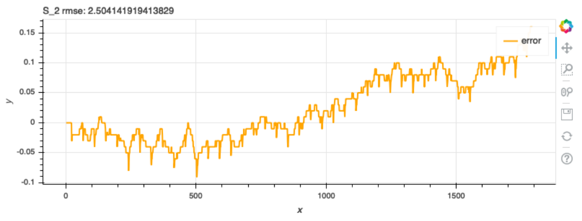
Figure B-11: Clustered Short Term Performances For Average Significance Time Period



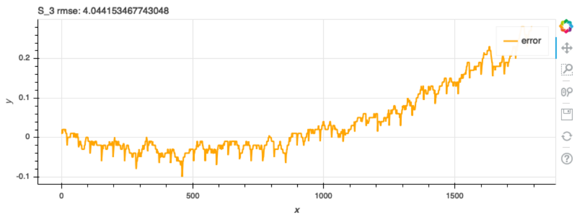
(a) S_0 Predictor



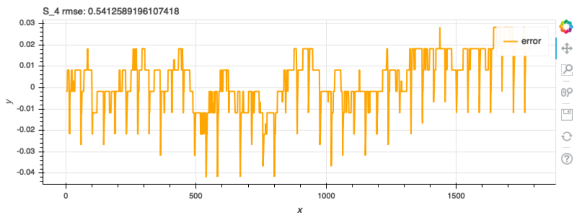
(b) S_1 Predictor



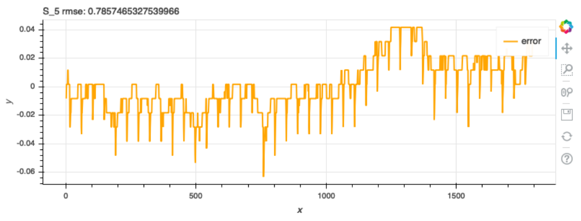
(c) S_2 Predictor



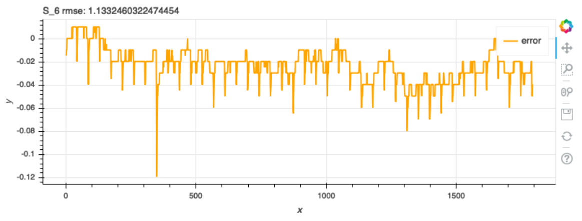
(d) S_3 Predictor



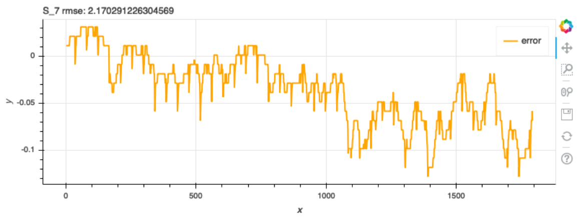
(e) S_4 Predictor



(f) S_5 Predictor

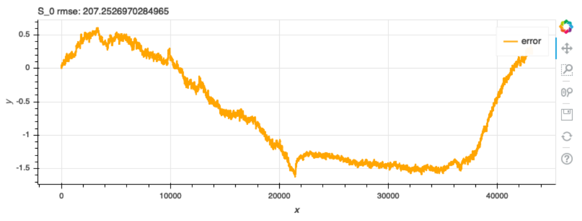


(g) S_6 Predictor

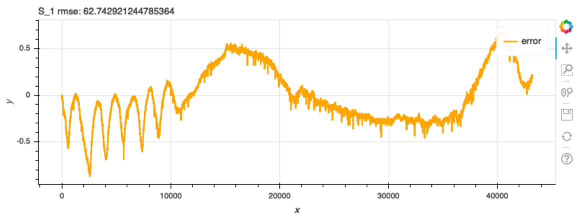


(h) S_7 Predictor

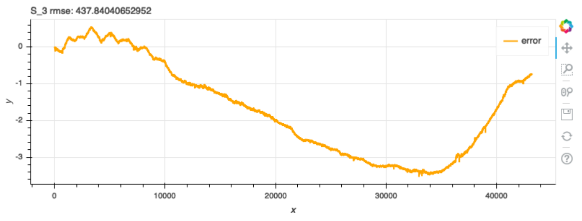
Figure B-12: Clustered Short Term Performances For Least Significant Time Period



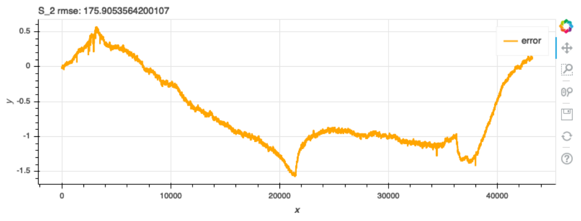
(a) S_0 Predictor



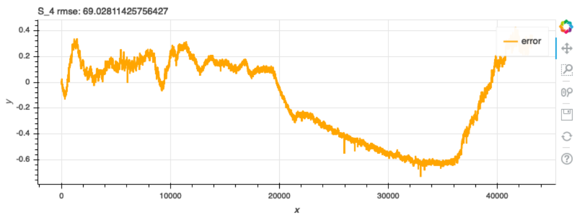
(b) S_1 Predictor



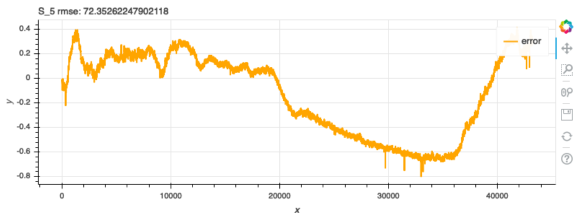
(c) S_2 Predictor



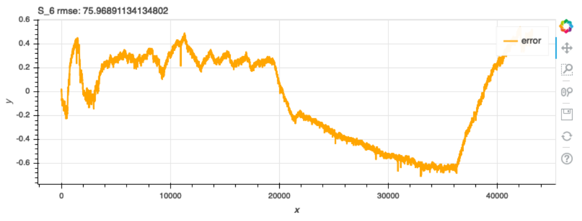
(d) S_3 Predictor



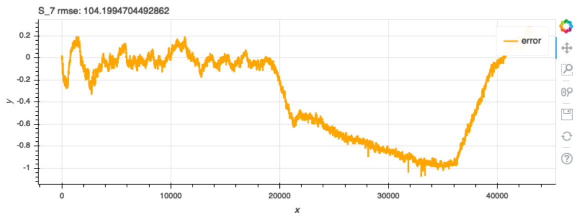
(e) S_4 Predictor



(f) S_5 Predictor

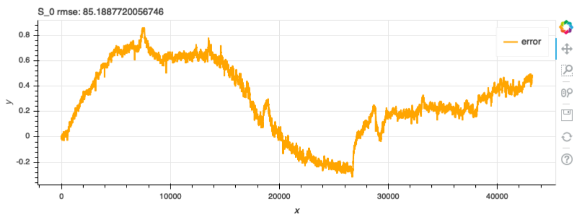


(g) S_6 Predictor

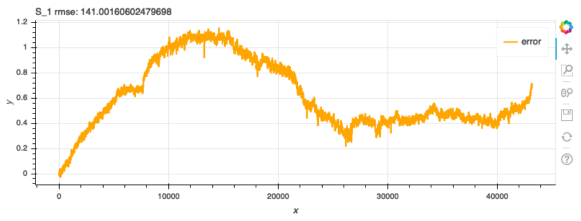


(h) S_7 Predictor

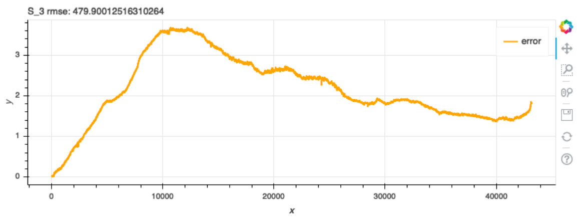
Figure B-13: Clustered Long Term Performances For Most Significant Time Period



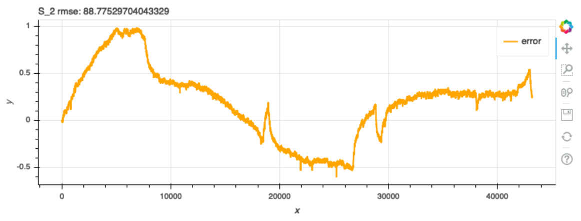
(a) S_0 Predictor



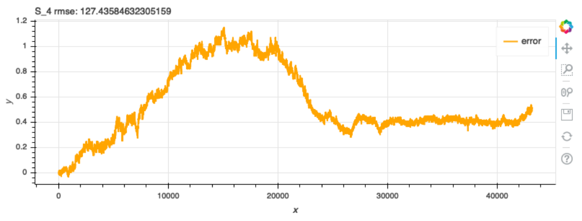
(b) S_1 Predictor



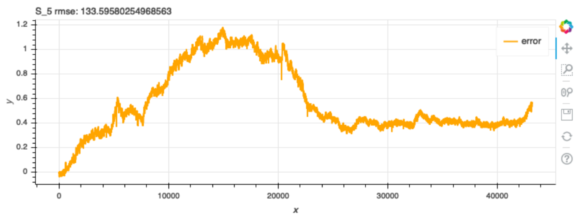
(c) S_2 Predictor



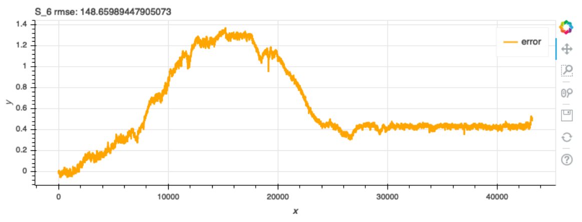
(d) S_3 Predictor



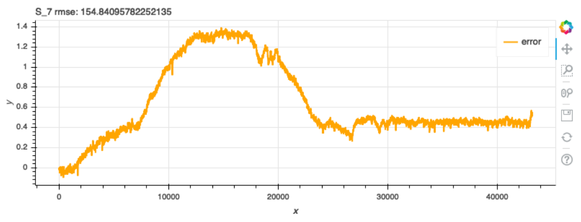
(e) S_4 Predictor



(f) S_5 Predictor

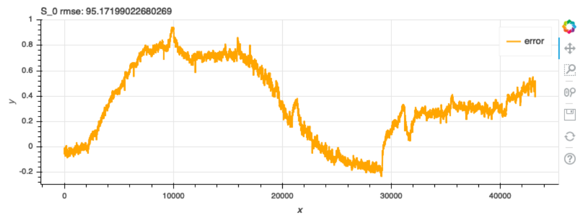


(g) S_6 Predictor

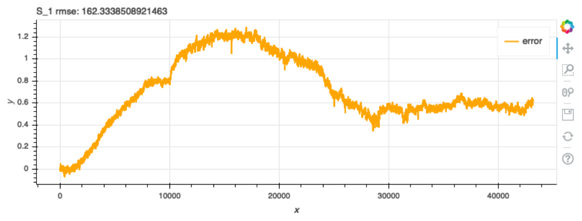


(h) S_7 Predictor

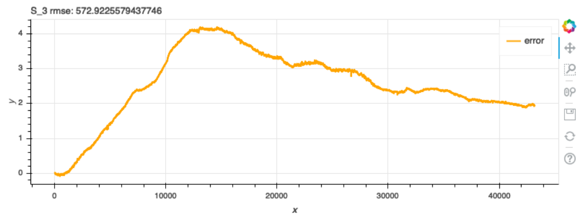
Figure B-14: Clustered Long Term Performances For Average Significance Time Period



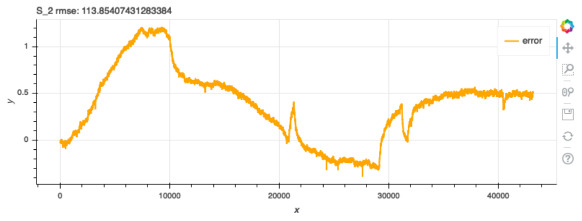
(a) S_0 Predictor



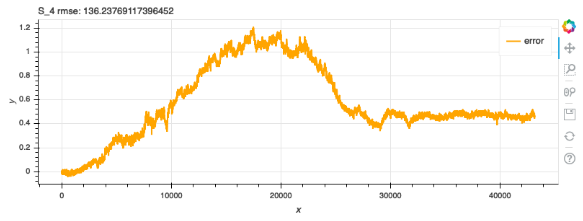
(b) S_1 Predictor



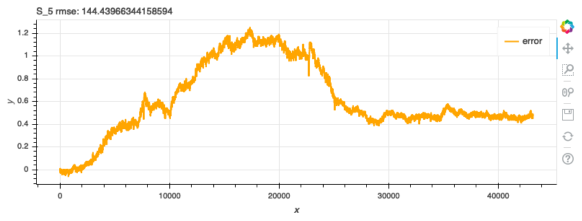
(c) S_2 Predictor



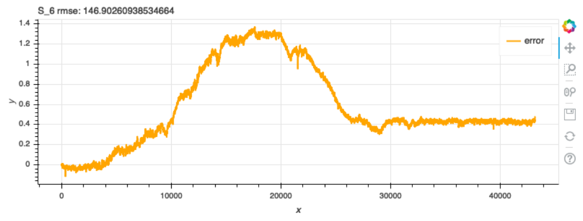
(d) S_3 Predictor



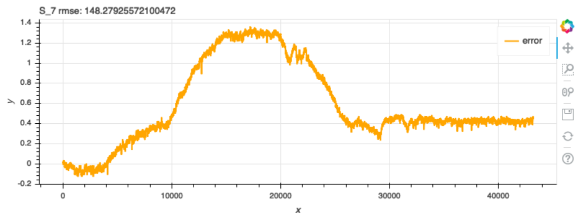
(e) S_4 Predictor



(f) S_5 Predictor



(g) S_6 Predictor



(h) S_7 Predictor

Figure B-15: Clustered Long Term Performances For Least Significant Time Period

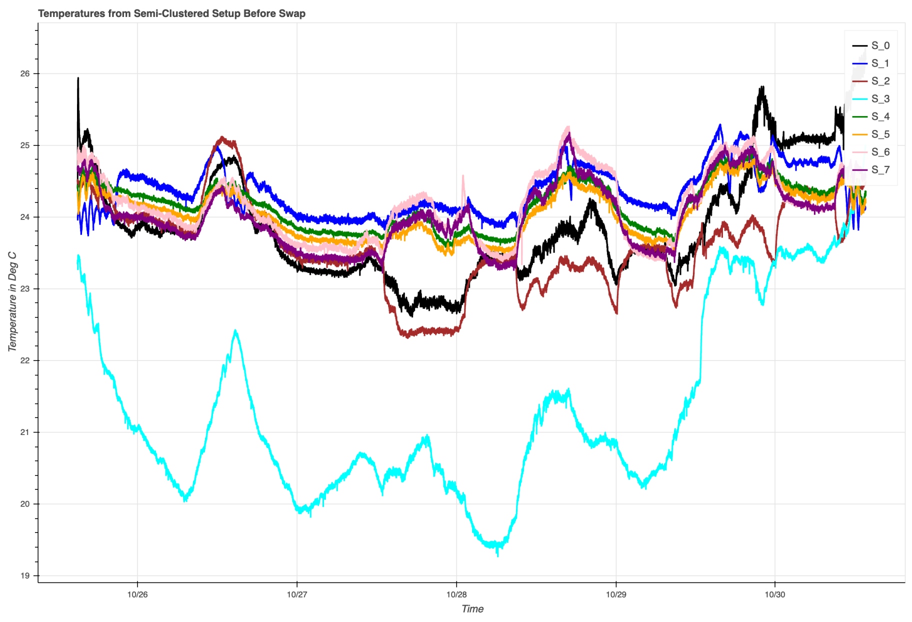


Figure B-16: Temperatures of Sensor nodes in Semi-Clustered Set up

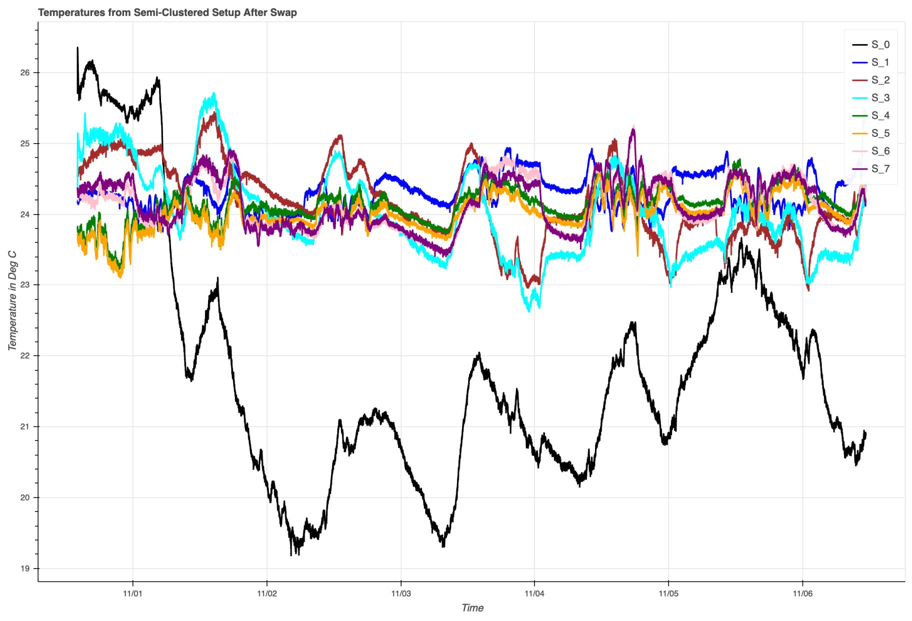


Figure B-17: Temperatures of Sensor nodes in Semi-Clustered Set up

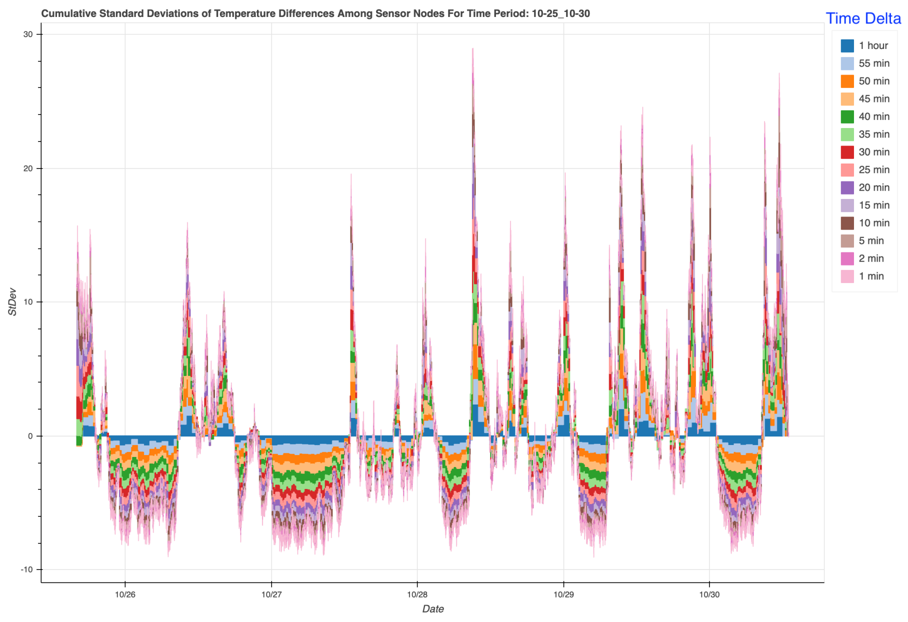


Figure B-18: Semi-Clustered Setup Cumulative Standard Deviations

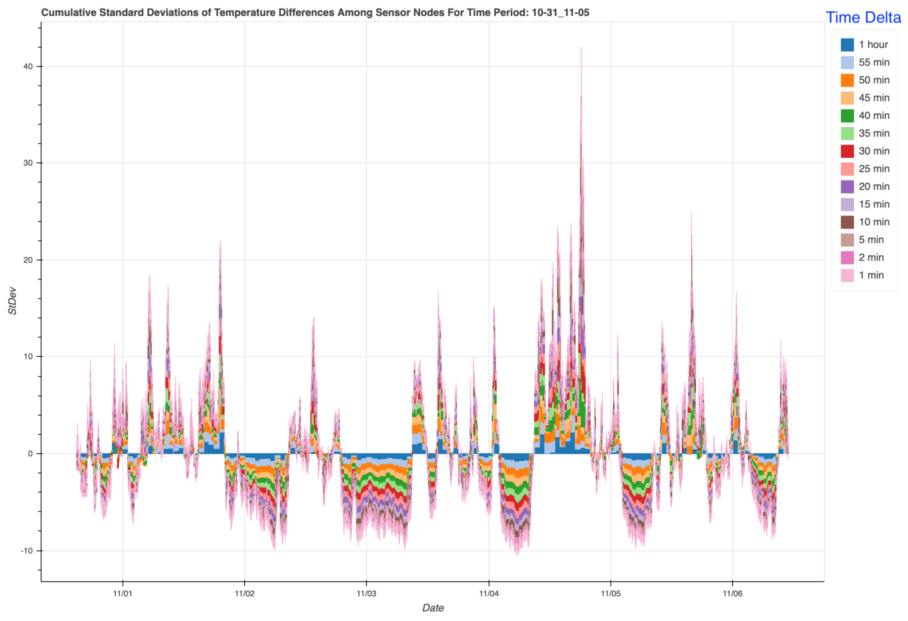
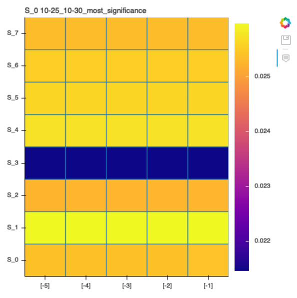
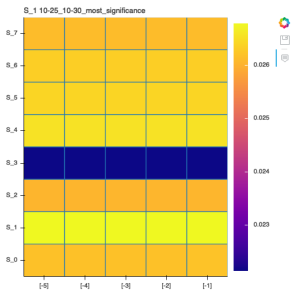


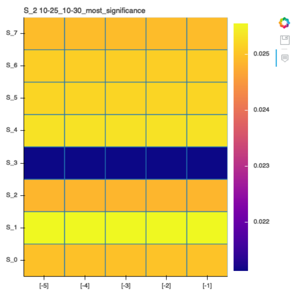
Figure B-19: Semi-Clustered Setup Cumulative Standard Deviations



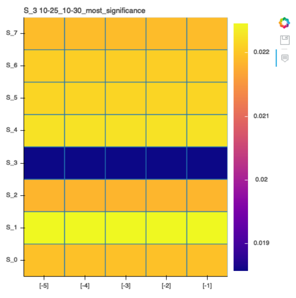
(a) S_0



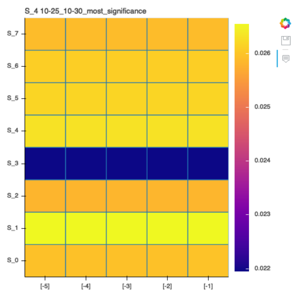
(b) S_1



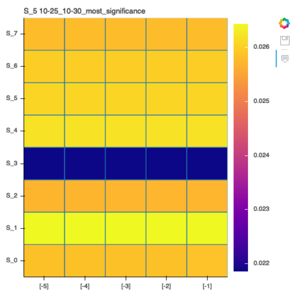
(c) S_2



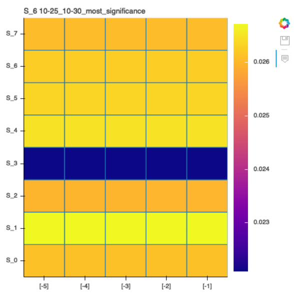
(d) S_3



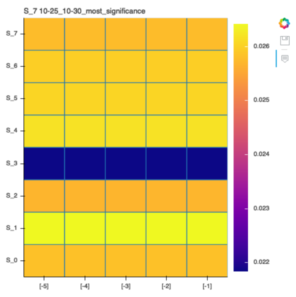
(e) S_4



(f) S_5

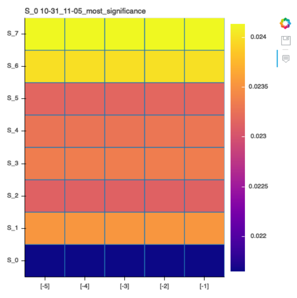


(g) S_6

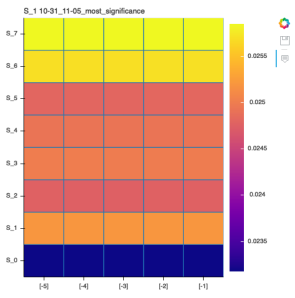


(h) S_7

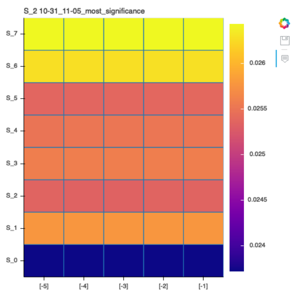
Figure B-20: Heatmaps from Most Significant Time Period Before Swap



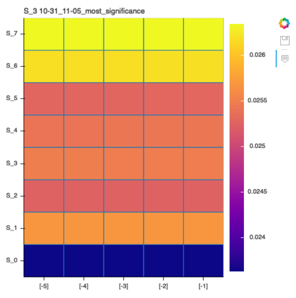
(a) S_0



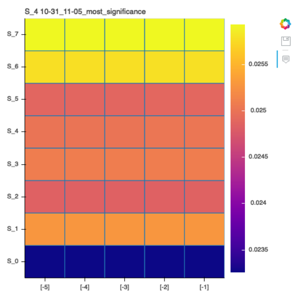
(b) S_1



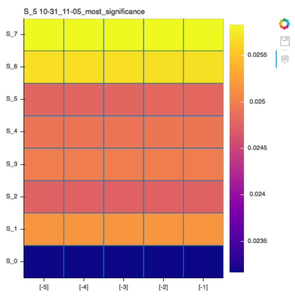
(c) S_2



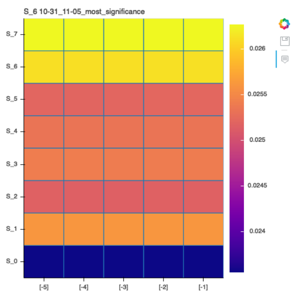
(d) S_3



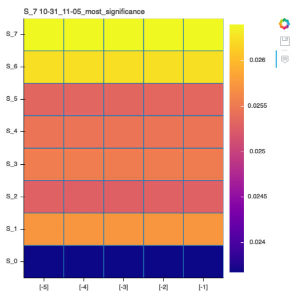
(e) S_4



(f) S_5

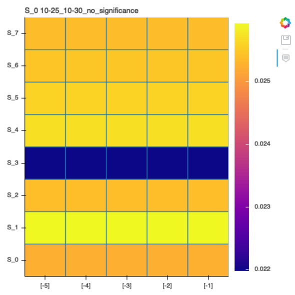


(g) S_6

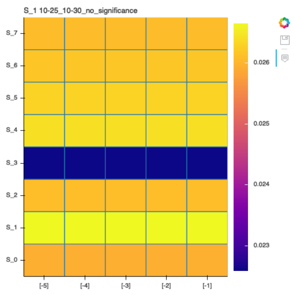


(h) S_7

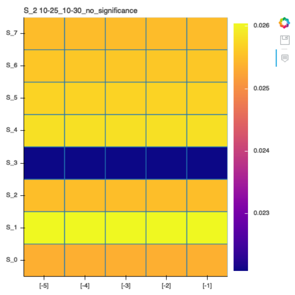
Figure B-21: Heatmaps from Most Significant Time Period After Swap



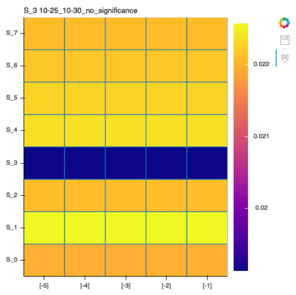
(a) S_0



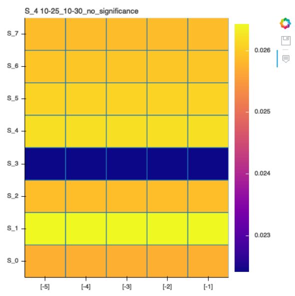
(b) S_1



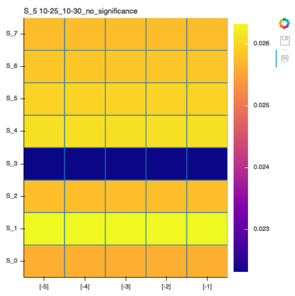
(c) S_2



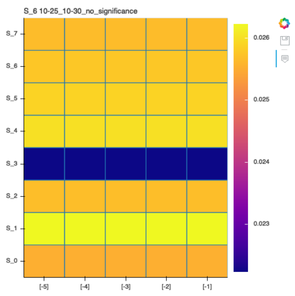
(d) S_3



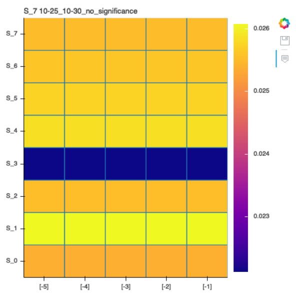
(e) S_4



(f) S_5

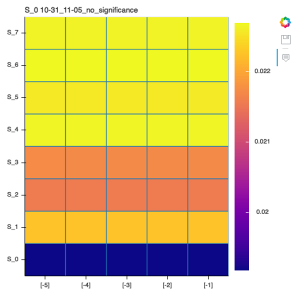


(g) S_6

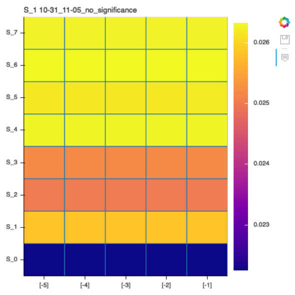


(h) S_7

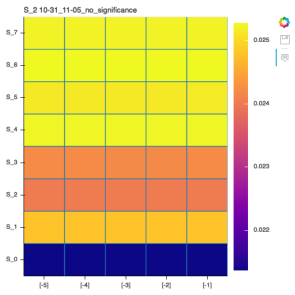
Figure B-22: Heatmaps from Average Significance Time Period Before Swap



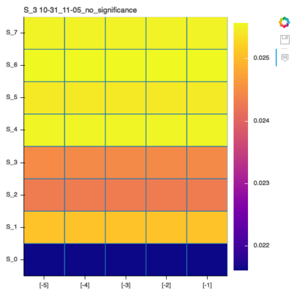
(a) S_0



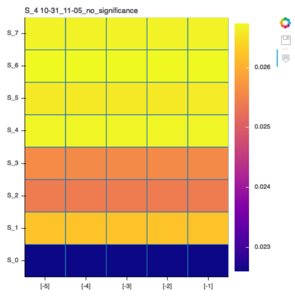
(b) S_1



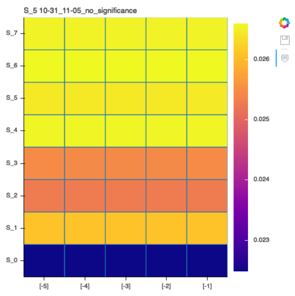
(c) S_2



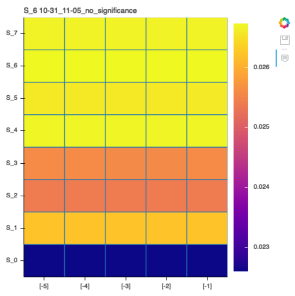
(d) S_3



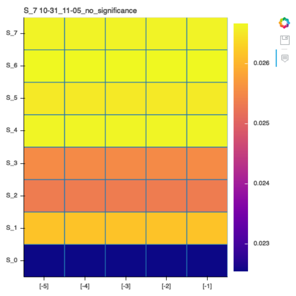
(e) S_4



(f) S_5

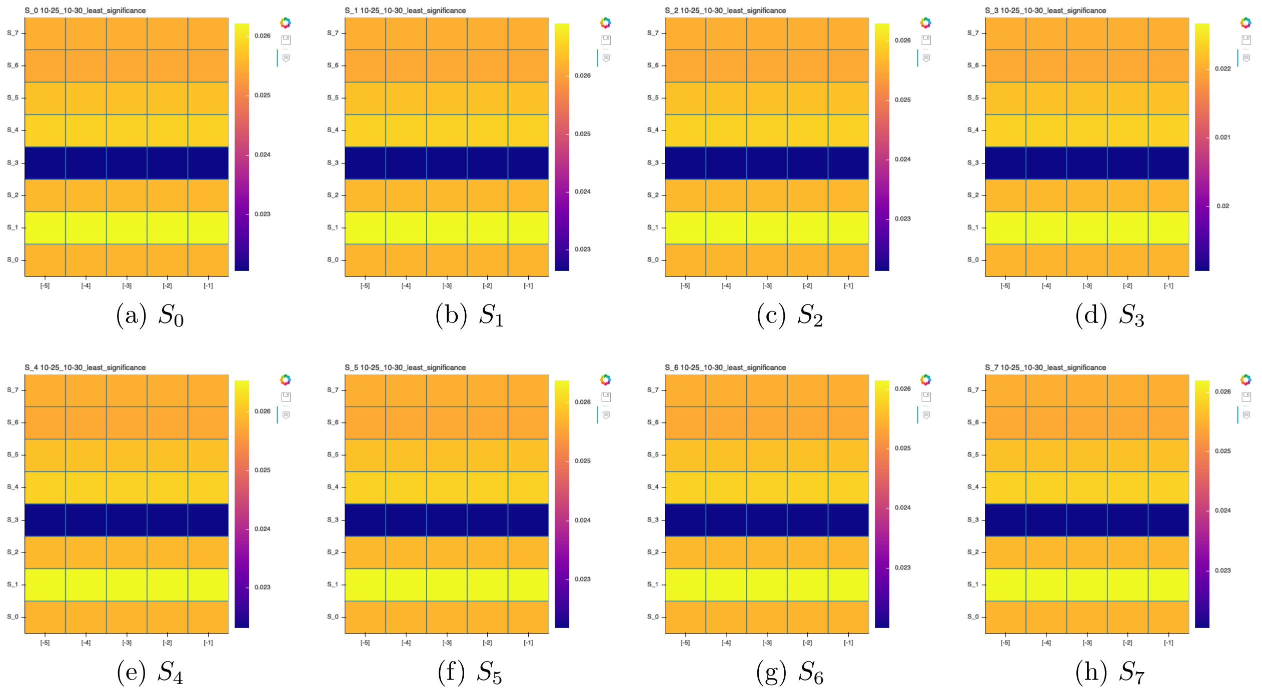


(g) S_6



(h) S_7

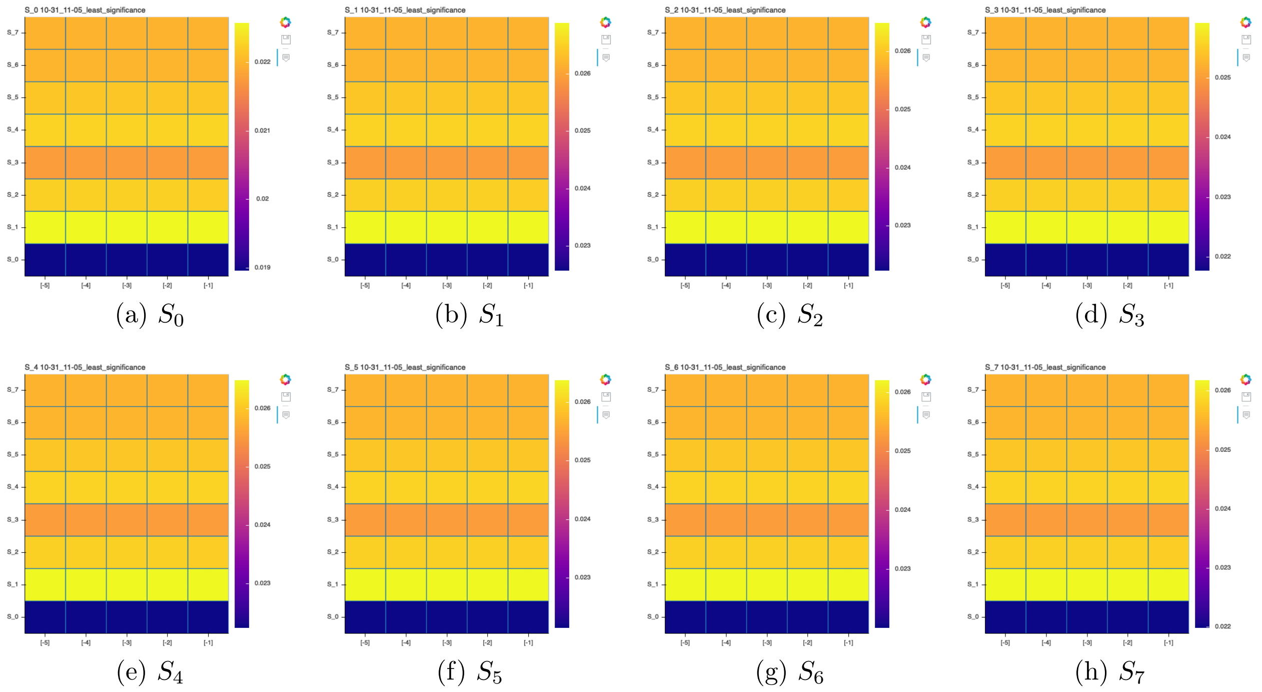
Figure B-23: Heatmaps from Average Significance Time Period After Swap



(a) S_0 (b) S_1 (c) S_2 (d) S_3

(e) S_4 (f) S_5 (g) S_6 (h) S_7

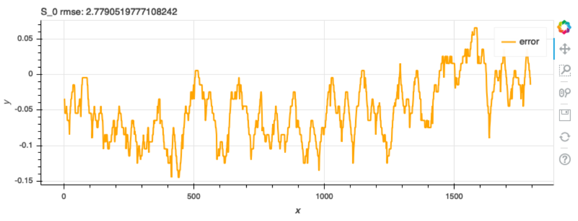
Figure B-24: Heatmaps from Least Significant Time Period Before Swap



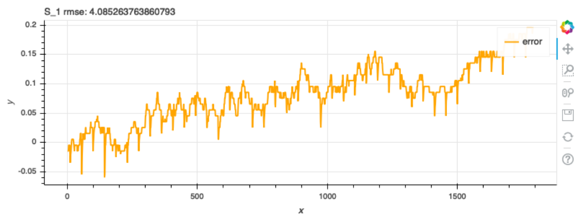
(a) S_0 (b) S_1 (c) S_2 (d) S_3

(e) S_4 (f) S_5 (g) S_6 (h) S_7

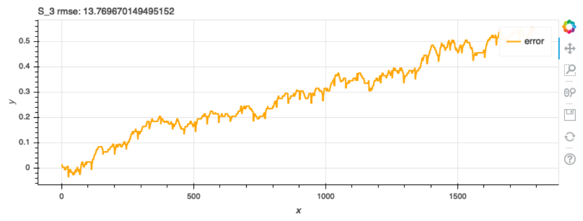
Figure B-25: Heatmaps from Least Significant Time Period After Swap



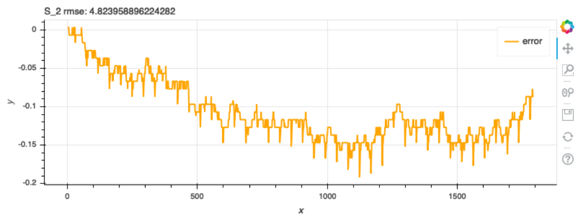
(a) S_0 Predictor



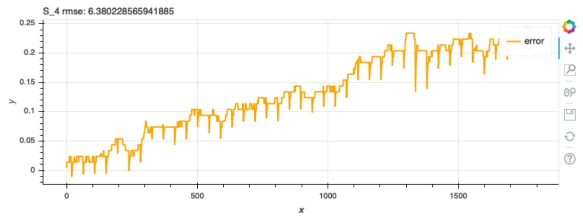
(b) S_1 Predictor



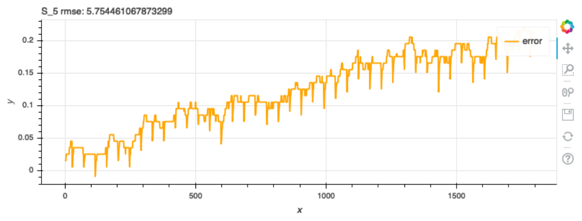
(c) S_2 Predictor



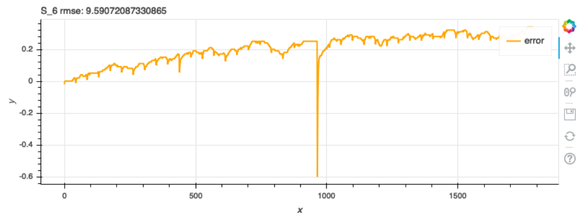
(d) S_3 Predictor



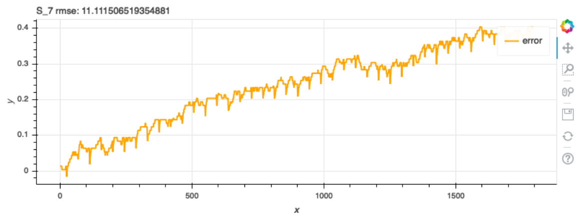
(e) S_4 Predictor



(f) S_5 Predictor

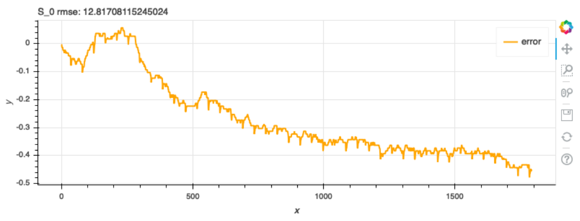


(g) S_6 Predictor

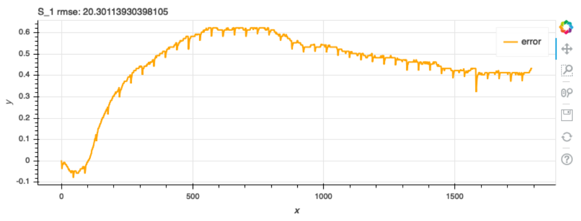


(h) S_7 Predictor

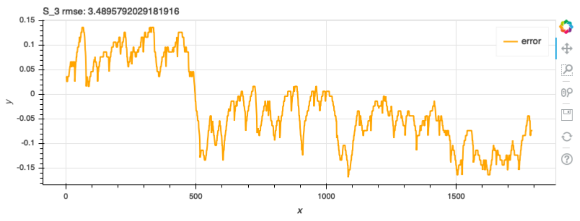
Figure B-26: Semi-Clustered Short Term Performances For Most Significant Time Period Before Swap



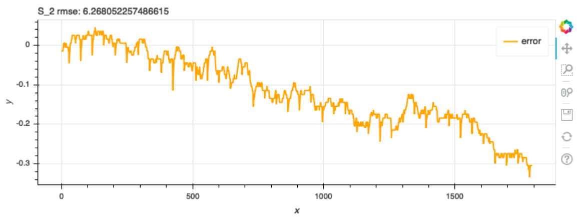
(a) S_0 Predictor



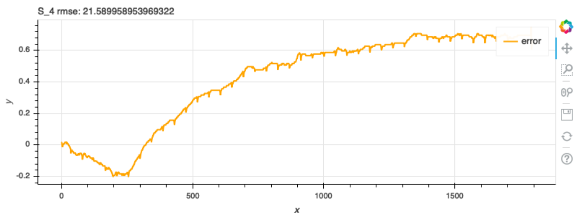
(b) S_1 Predictor



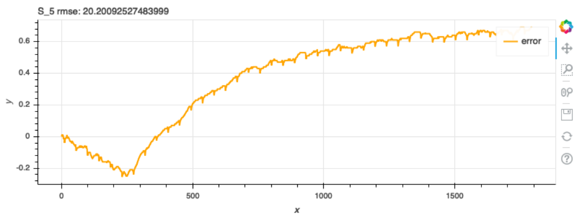
(c) S_2 Predictor



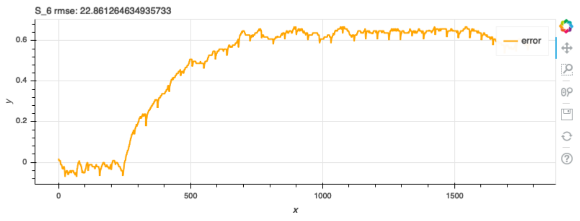
(d) S_3 Predictor



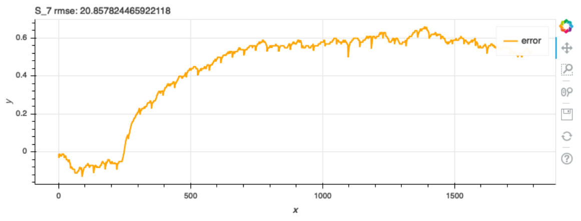
(e) S_4 Predictor



(f) S_5 Predictor

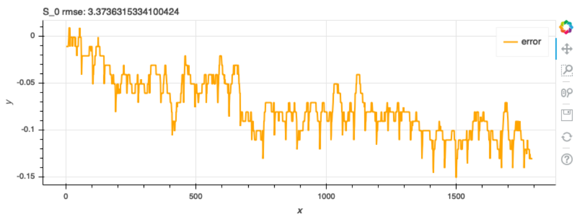


(g) S_6 Predictor

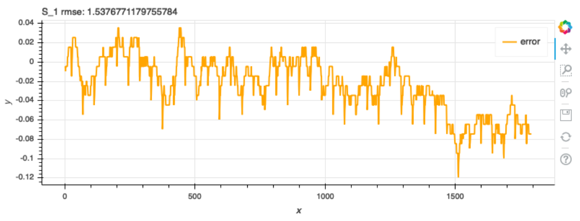


(h) S_7 Predictor

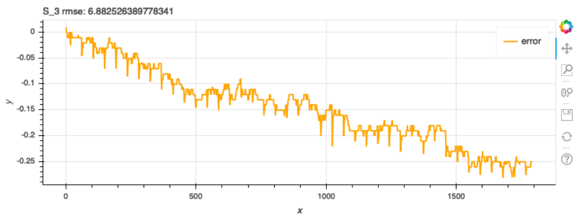
Figure B-27: Semi-Clustered Short Term Performances For Most Significant Time Period After Swap



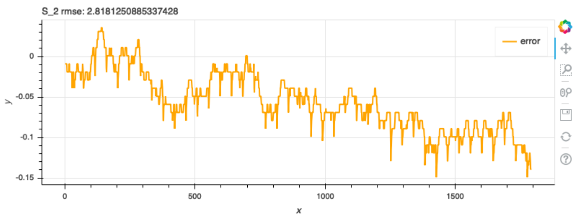
(a) S_0 Predictor



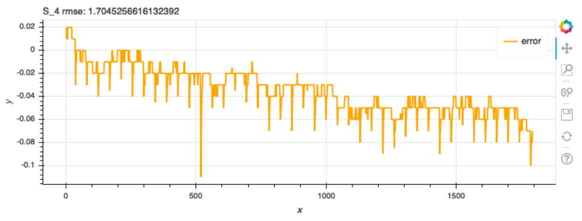
(b) S_1 Predictor



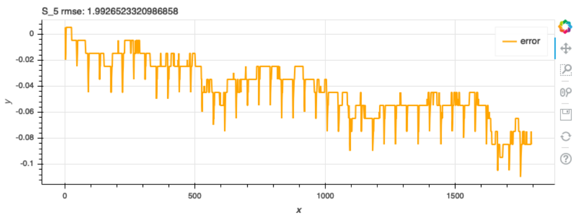
(c) S_2 Predictor



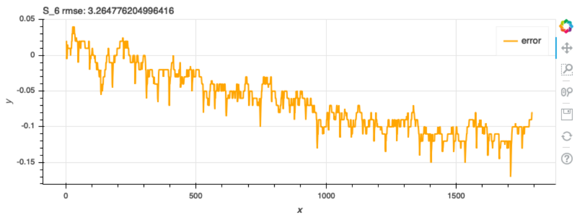
(d) S_3 Predictor



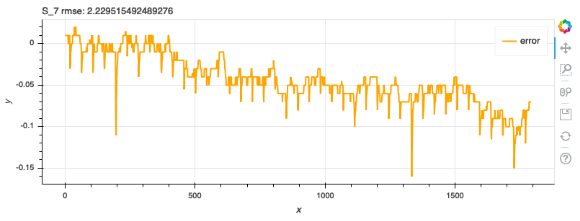
(e) S_4 Predictor



(f) S_5 Predictor

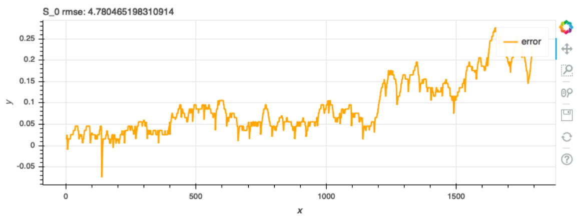


(g) S_6 Predictor

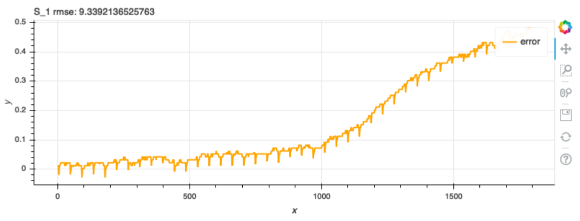


(h) S_7 Predictor

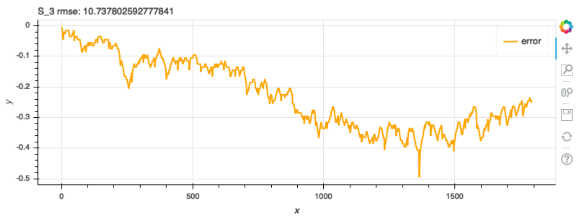
Figure B-28: Semi-Clustered Short Term Performances For Average Significance Time Period Before Swap



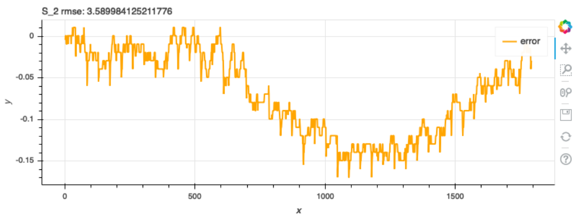
(a) S_0 Predictor



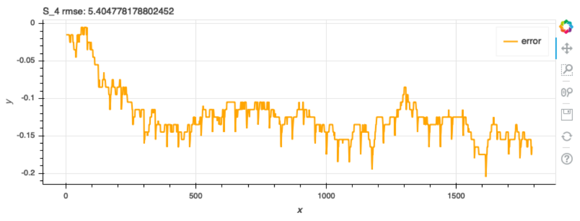
(b) S_1 Predictor



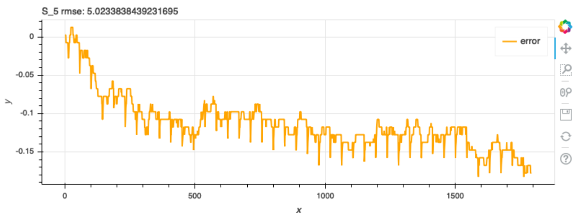
(c) S_2 Predictor



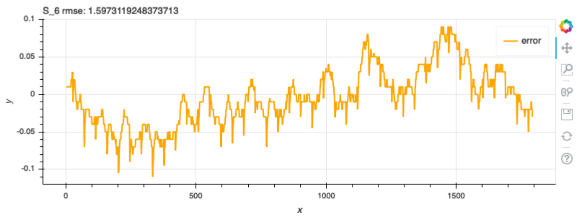
(d) S_3 Predictor



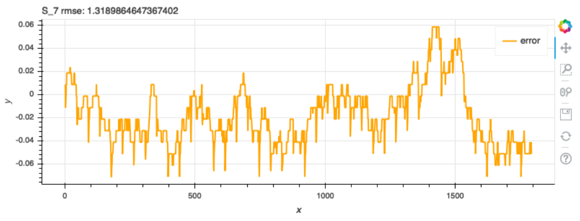
(e) S_4 Predictor



(f) S_5 Predictor

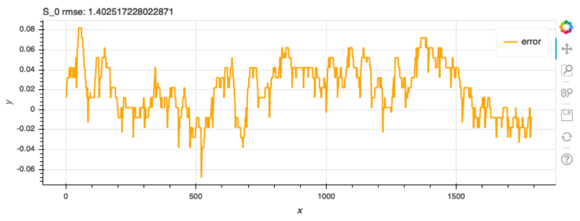


(g) S_6 Predictor

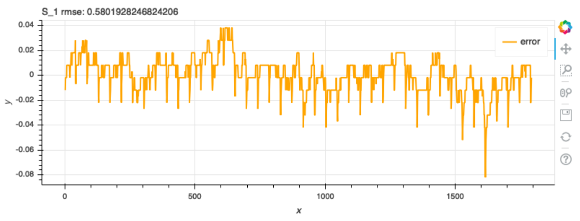


(h) S_7 Predictor

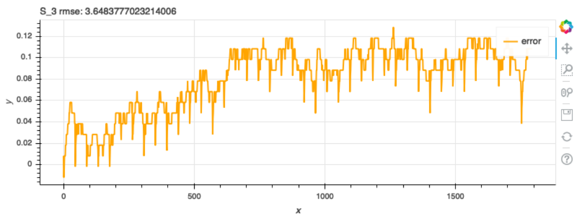
Figure B-29: Semi-Clustered Short Term Performances For Average Significance Time Period After Swap



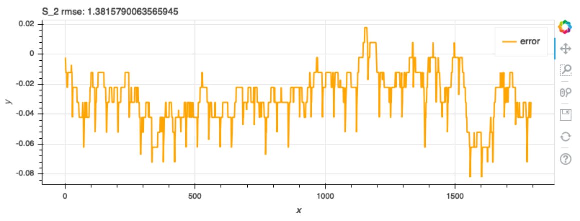
(a) S_0 Predictor



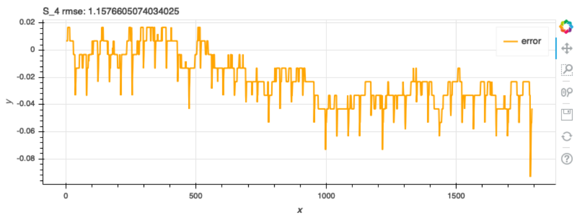
(b) S_1 Predictor



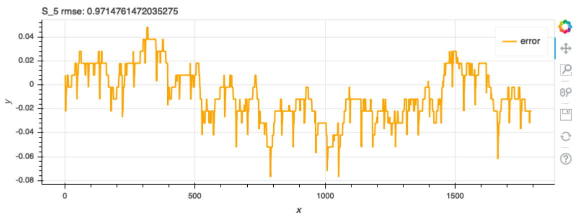
(c) S_2 Predictor



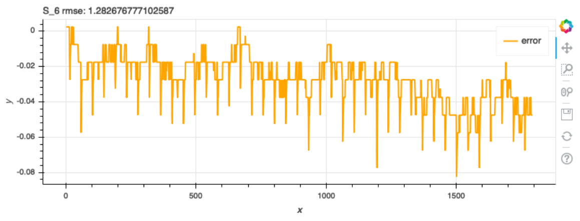
(d) S_3 Predictor



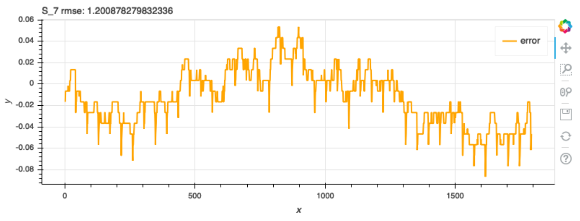
(e) S_4 Predictor



(f) S_5 Predictor

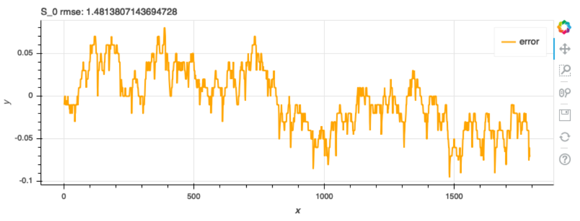


(g) S_6 Predictor

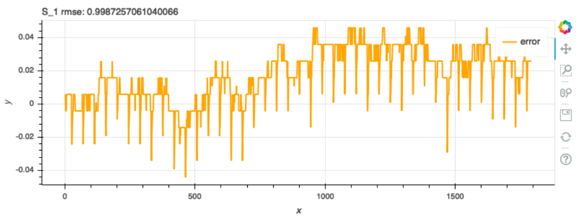


(h) S_7 Predictor

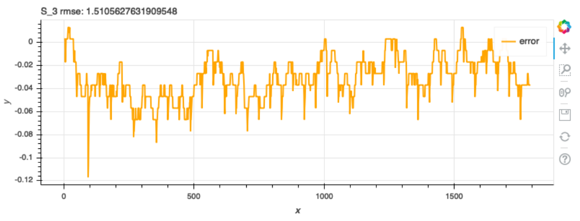
Figure B-30: Semi-Clustered Short Term Performances For Least Significant Time Period Before Swap



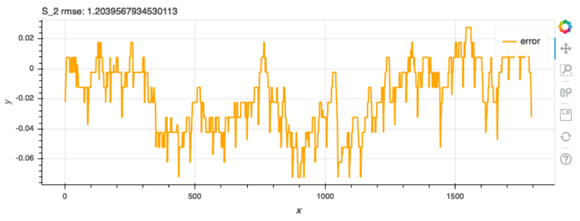
(a) S_0 Predictor



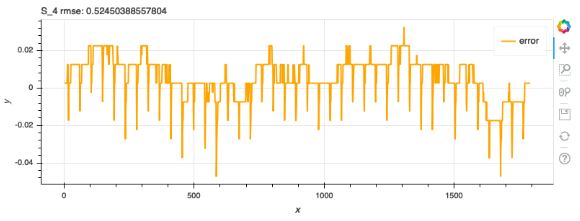
(b) S_1 Predictor



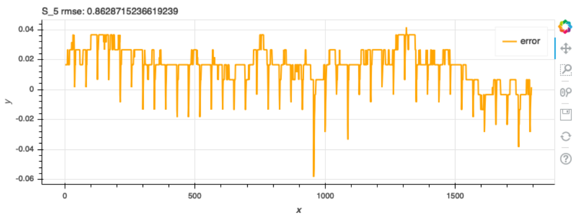
(c) S_2 Predictor



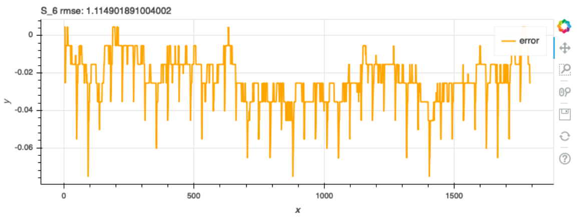
(d) S_3 Predictor



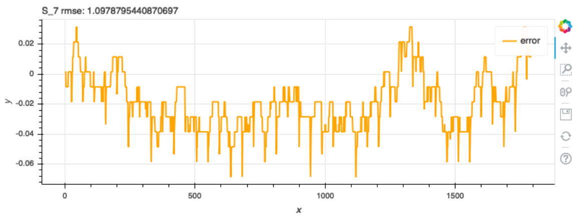
(e) S_4 Predictor



(f) S_5 Predictor

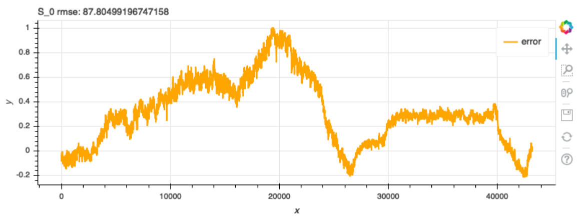


(g) S_6 Predictor

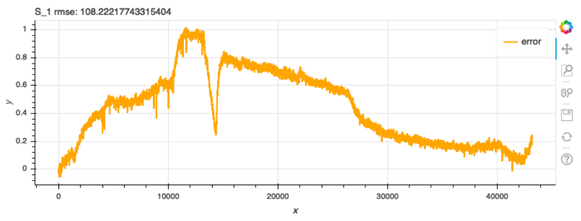


(h) S_7 Predictor

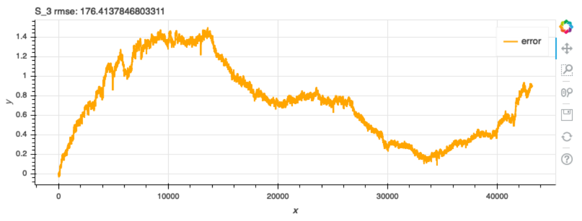
Figure B-31: Semi-Clustered Short Term Performances For Least Significant Time Period After Swap



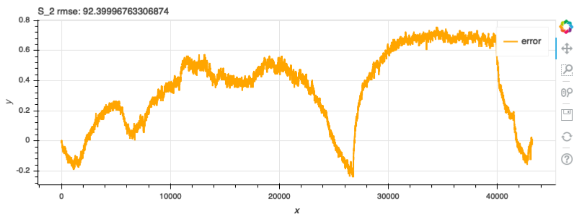
(a) S_0 Predictor



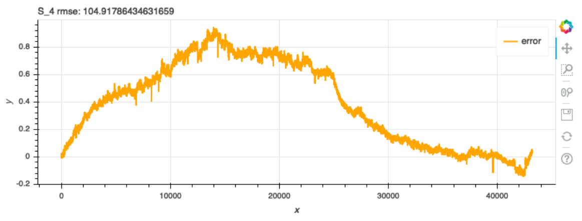
(b) S_1 Predictor



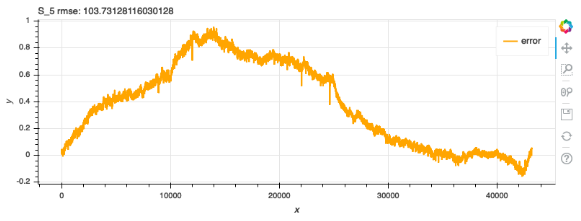
(c) S_2 Predictor



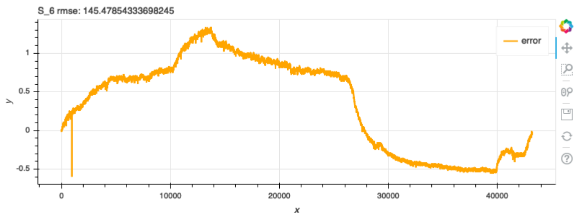
(d) S_3 Predictor



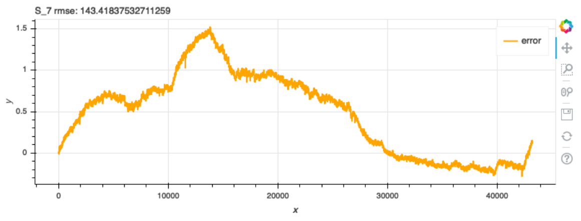
(e) S_4 Predictor



(f) S_5 Predictor

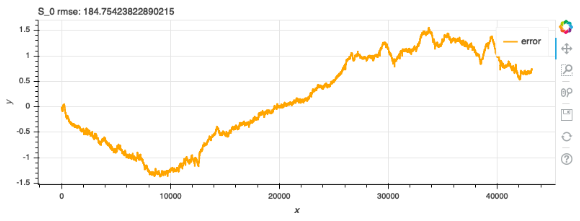


(g) S_6 Predictor

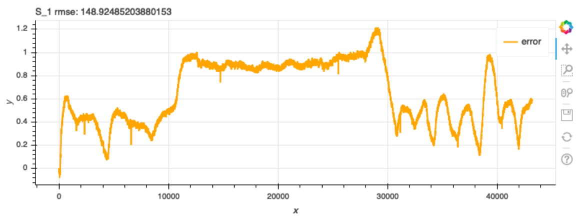


(h) S_7 Predictor

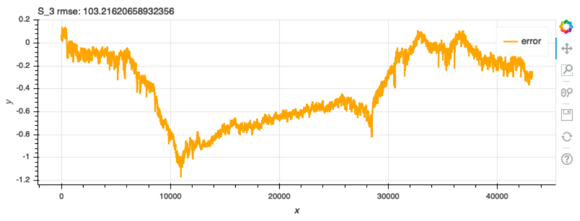
Figure B-32: Semi-Clustered Long Term Performances For Most Significant Time Period Before Swap



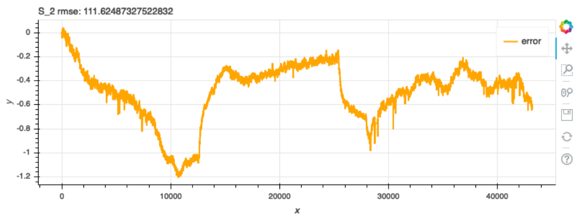
(a) S_0 Predictor



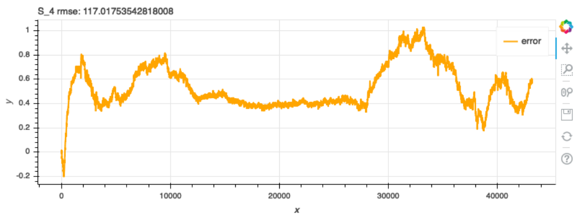
(b) S_1 Predictor



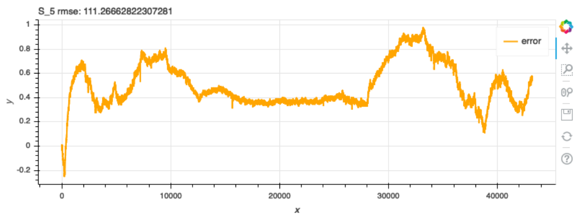
(c) S_2 Predictor



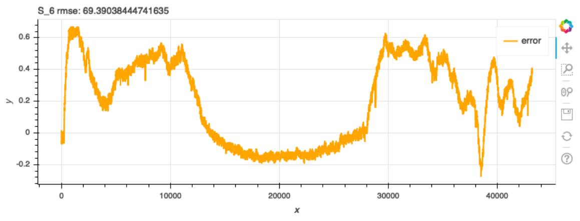
(d) S_3 Predictor



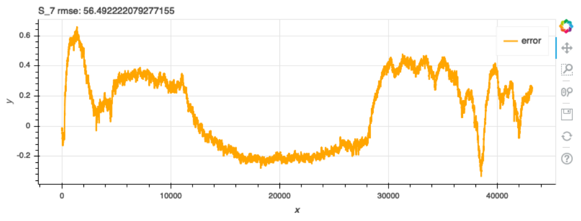
(e) S_4 Predictor



(f) S_5 Predictor

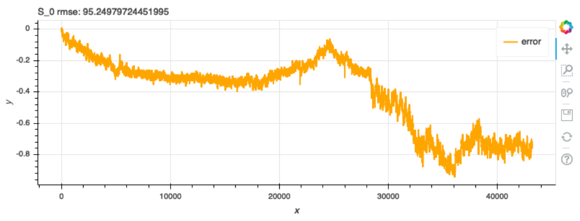


(g) S_6 Predictor

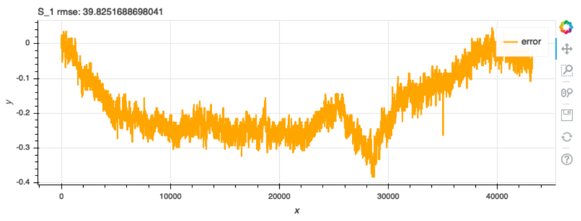


(h) S_7 Predictor

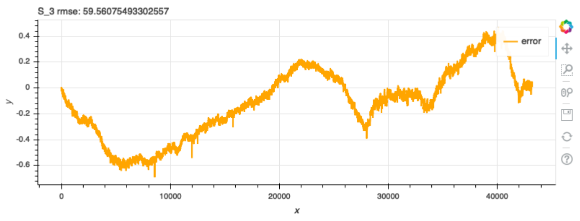
Figure B-33: Semi-Clustered Long Term Performances For Most Significant Time Period After Swap



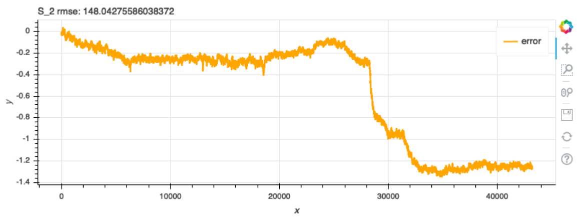
(a) S_0 Predictor



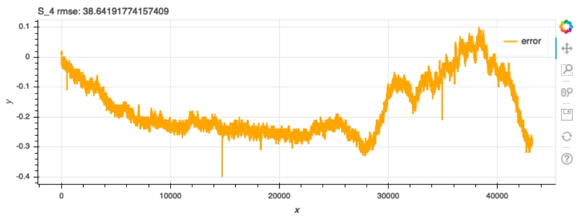
(b) S_1 Predictor



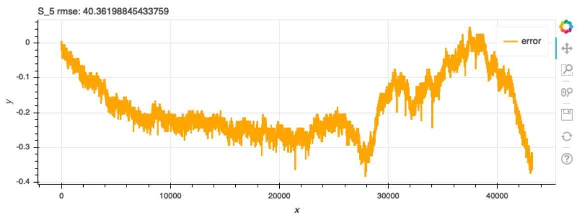
(c) S_2 Predictor



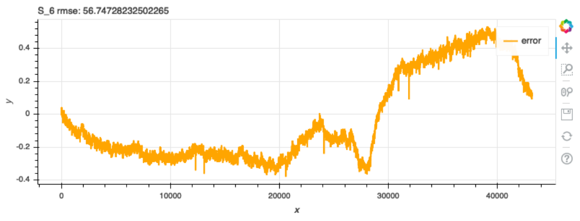
(d) S_3 Predictor



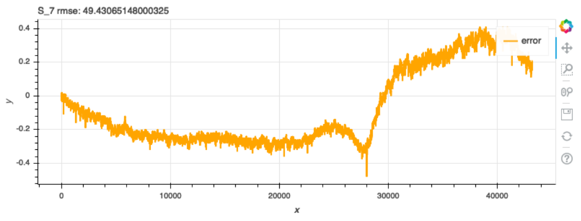
(e) S_4 Predictor



(f) S_5 Predictor

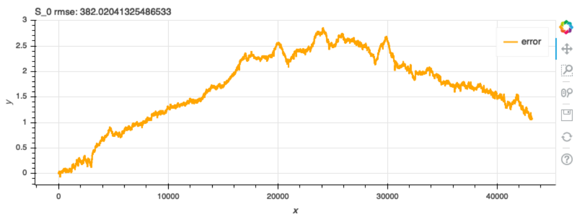


(g) S_6 Predictor

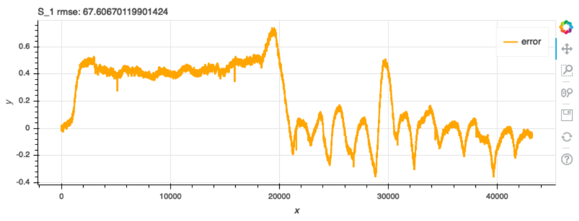


(h) S_7 Predictor

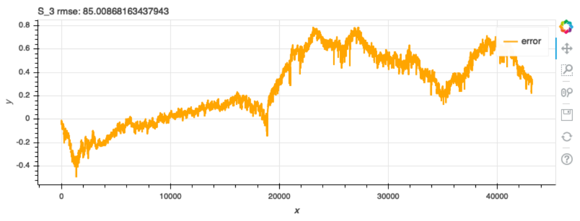
Figure B-34: Semi-Clustered Long Term Performances For Average Significance Time Period Before Swap



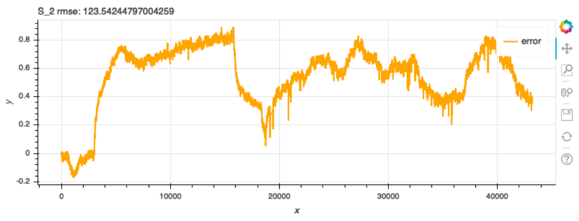
(a) S_0 Predictor



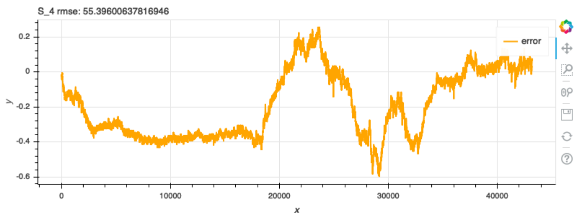
(b) S_1 Predictor



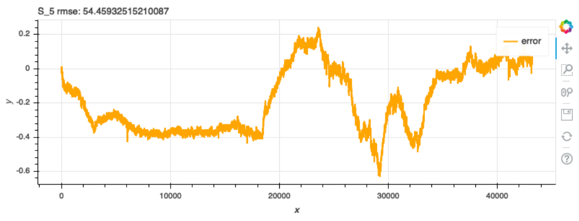
(c) S_2 Predictor



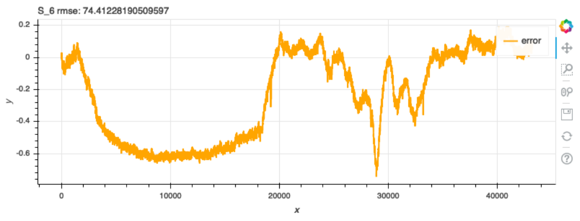
(d) S_3 Predictor



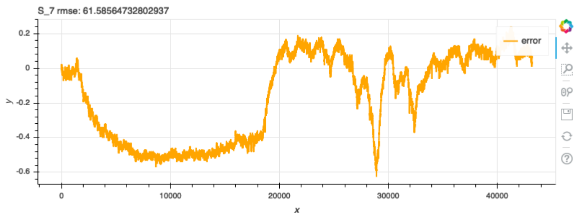
(e) S_4 Predictor



(f) S_5 Predictor

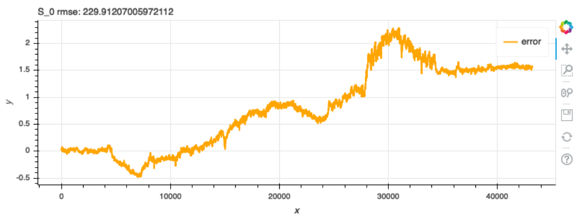


(g) S_6 Predictor

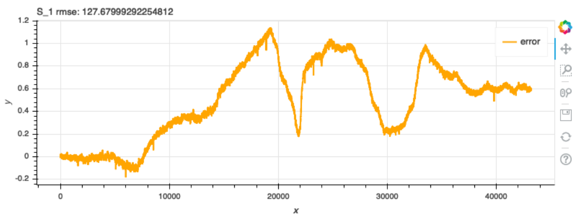


(h) S_7 Predictor

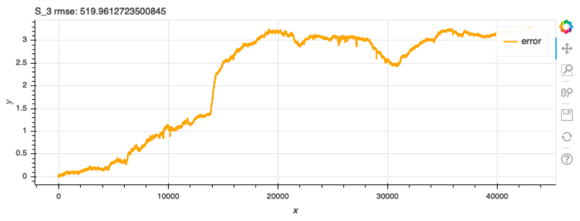
Figure B-35: Semi-Clustered Long Term Performances For Average Significance Time Period After Swap



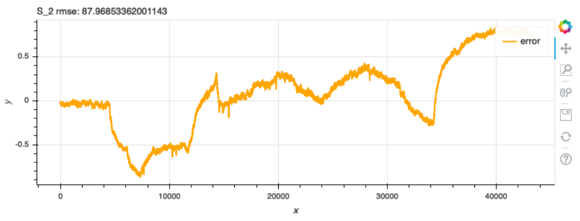
(a) S_0 Predictor



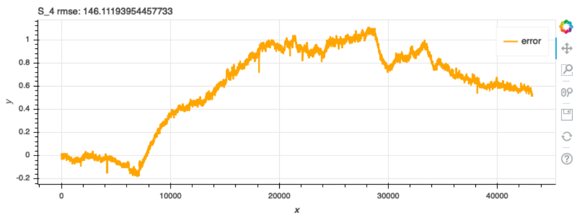
(b) S_1 Predictor



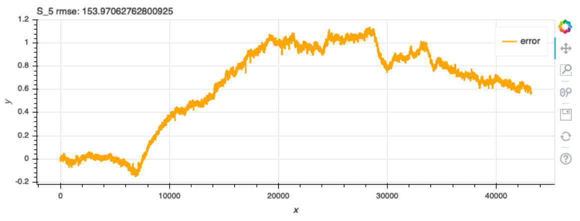
(c) S_2 Predictor



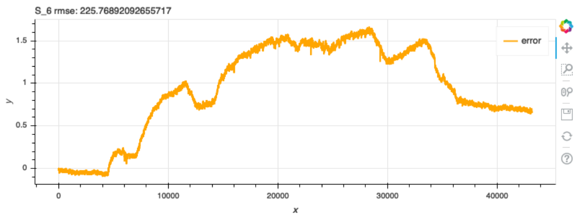
(d) S_3 Predictor



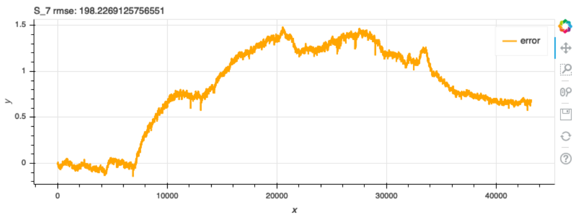
(e) S_4 Predictor



(f) S_5 Predictor

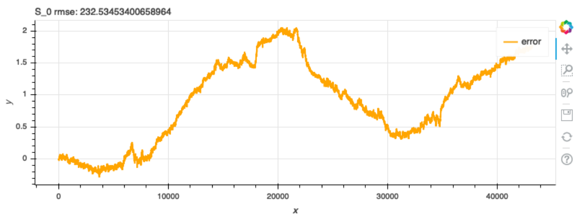


(g) S_6 Predictor

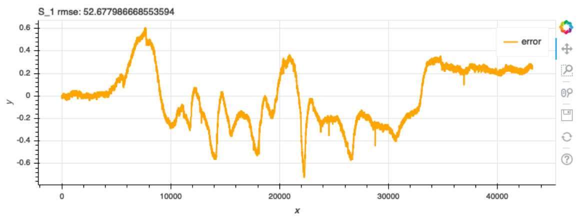


(h) S_7 Predictor

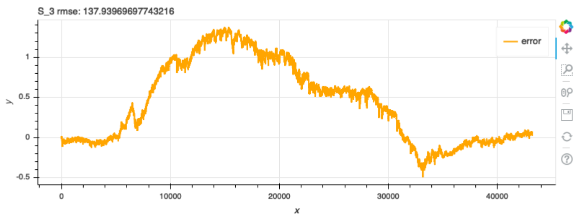
Figure B-36: Semi-Clustered Long Term Performances For Least Significant Time Period Before Swap



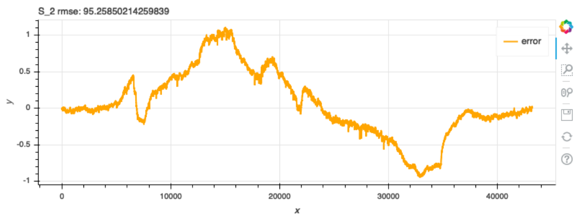
(a) S_0 Predictor



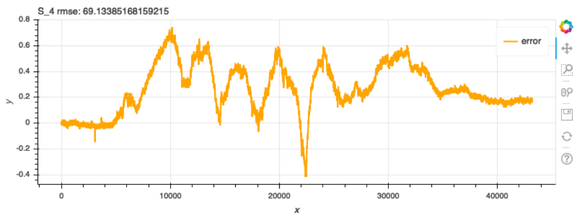
(b) S_1 Predictor



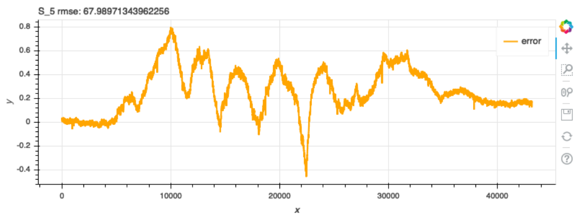
(c) S_2 Predictor



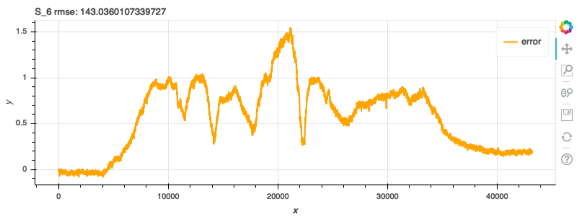
(d) S_3 Predictor



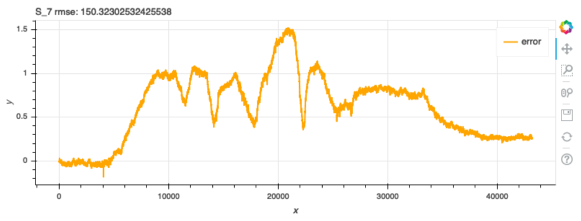
(e) S_4 Predictor



(f) S_5 Predictor



(g) S_6 Predictor



(h) S_7 Predictor

Figure B-37: Semi-Clustered Long Term Performances For Least Significant Time Period After Swap

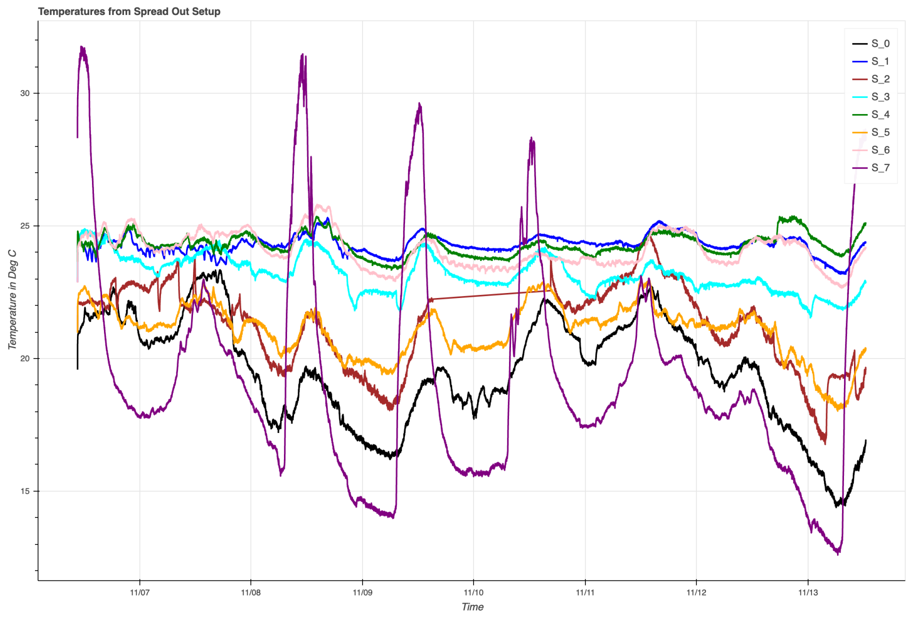


Figure B-38: Temperatures of Sensor nodes in Spread Out Set up

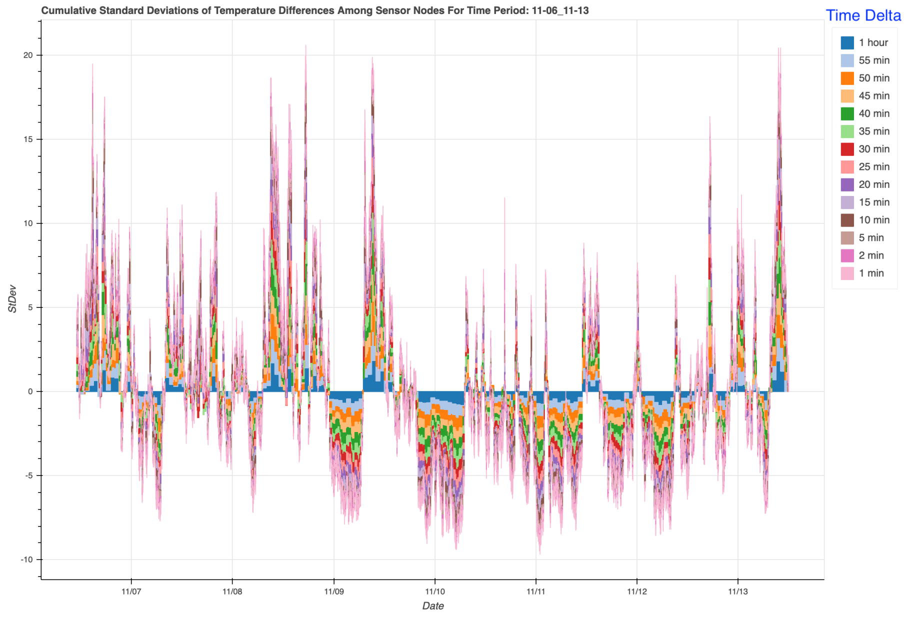
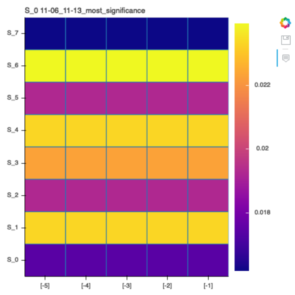
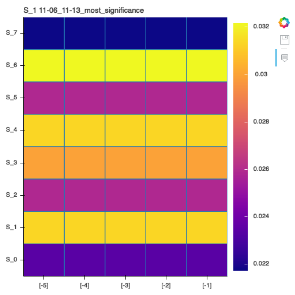


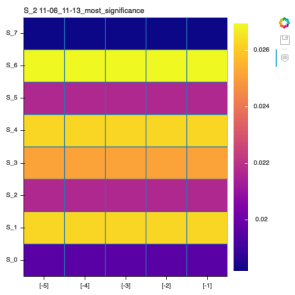
Figure B-39: Spread Out Setup Cumulative Standard Deviations



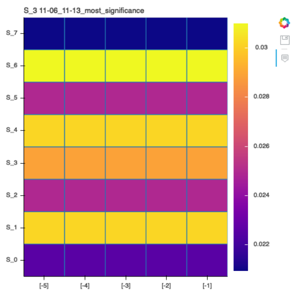
(a) S_0



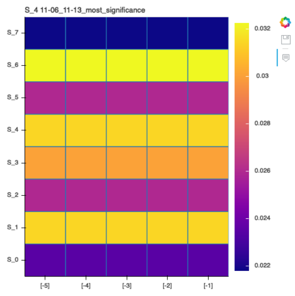
(b) S_1



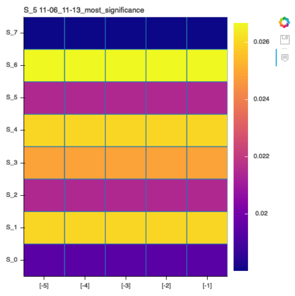
(c) S_2



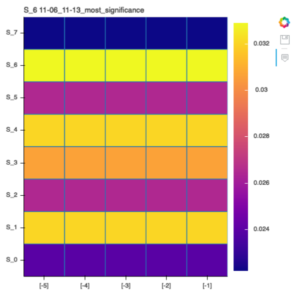
(d) S_3



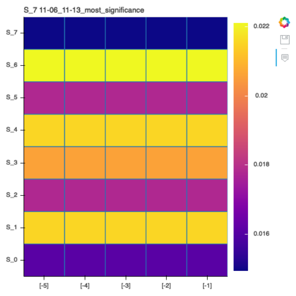
(e) S_4



(f) S_5

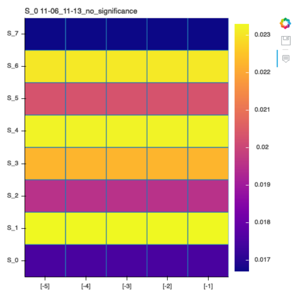


(g) S_6

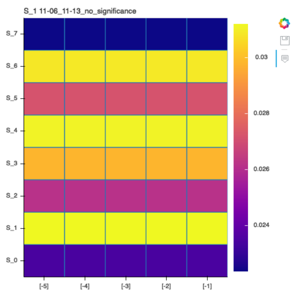


(h) S_7

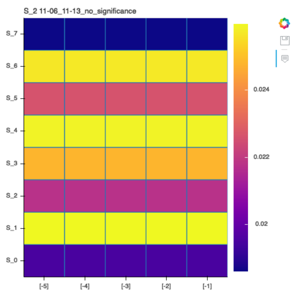
Figure B-40: Heatmaps from Most Significant Time Period



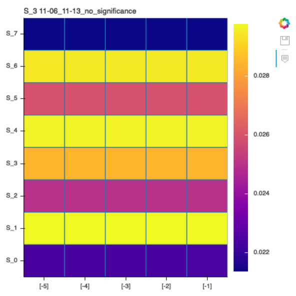
(a) S_0



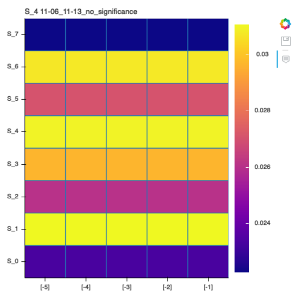
(b) S_1



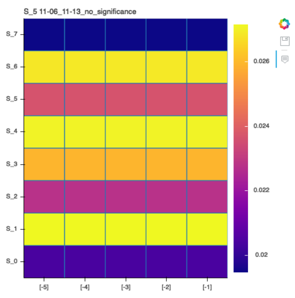
(c) S_2



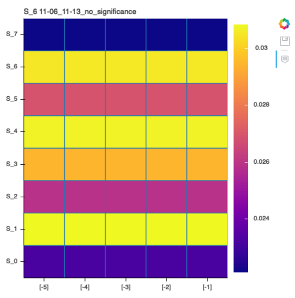
(d) S_3



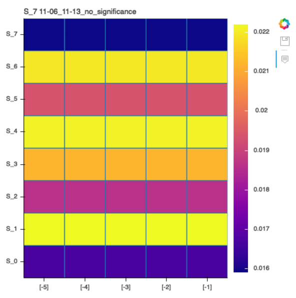
(e) S_4



(f) S_5

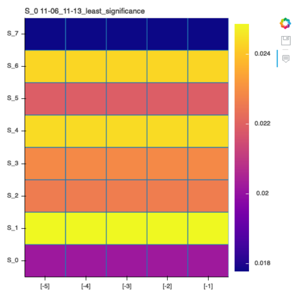


(g) S_6

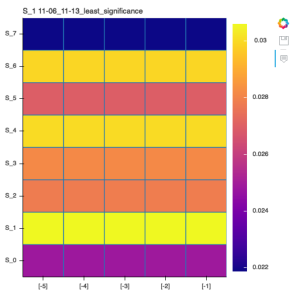


(h) S_7

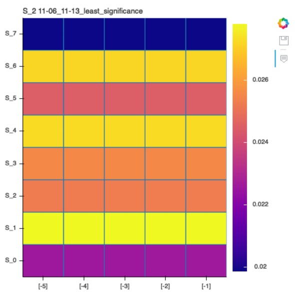
Figure B-41: Heatmaps from Average Significance Time Period



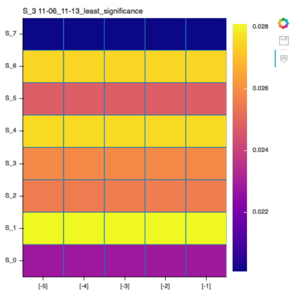
(a) S_0



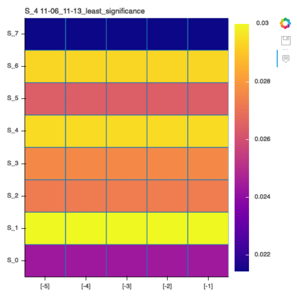
(b) S_1



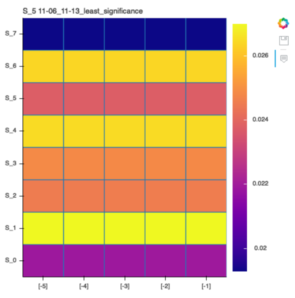
(c) S_2



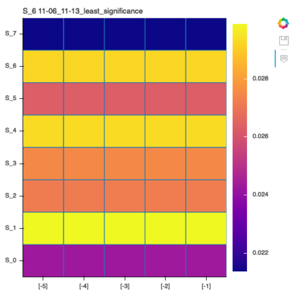
(d) S_3



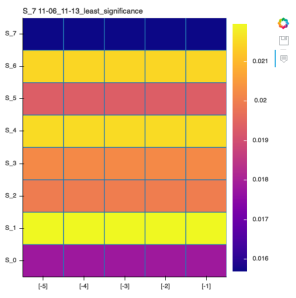
(e) S_4



(f) S_5

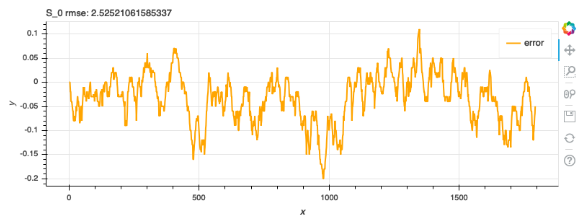


(g) S_6

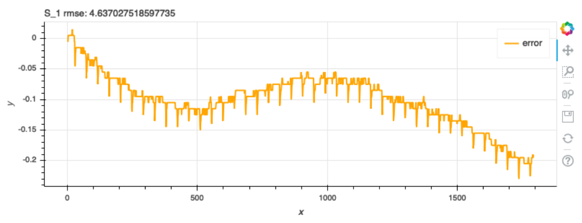


(h) S_7

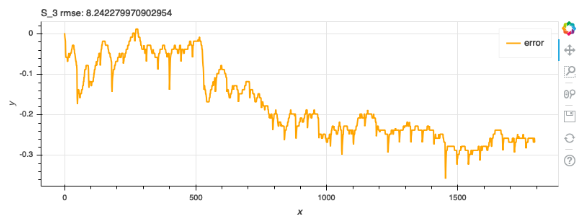
Figure B-42: Heatmaps from Least Significant Time Period



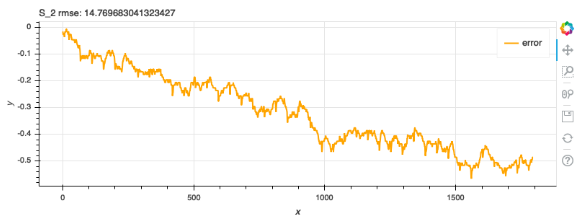
(a) S_0 Predictor



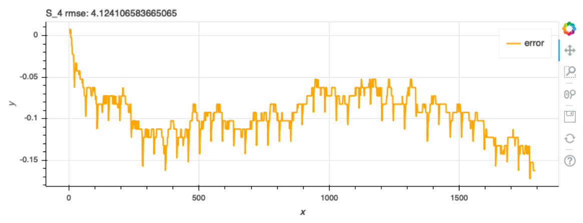
(b) S_1 Predictor



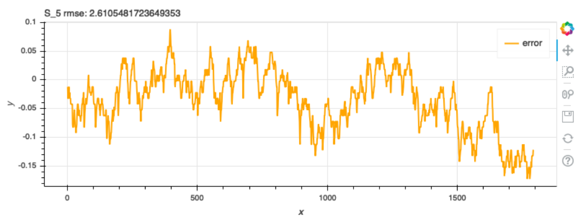
(c) S_2 Predictor



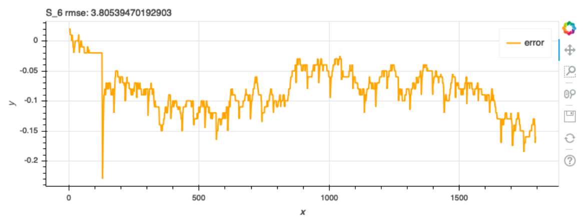
(d) S_3 Predictor



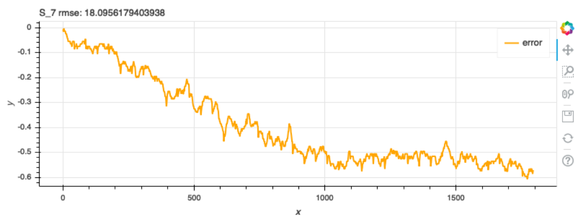
(e) S_4 Predictor



(f) S_5 Predictor

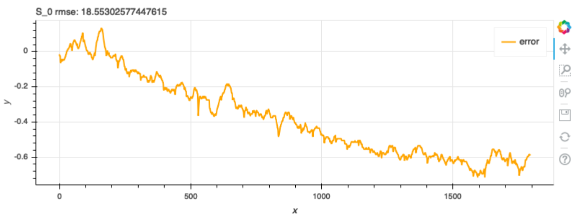


(g) S_6 Predictor

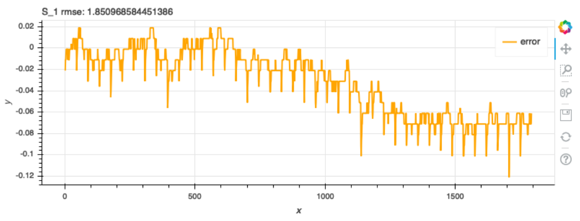


(h) S_7 Predictor

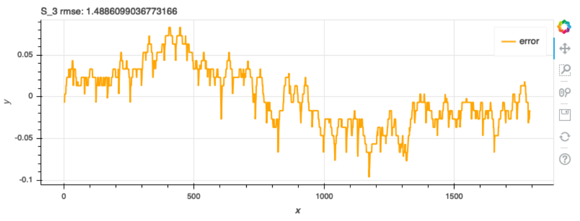
Figure B-43: Spread Out Short Term Performances For Most Significant Time Period



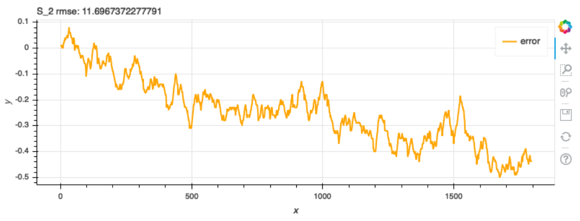
(a) S_0 Predictor



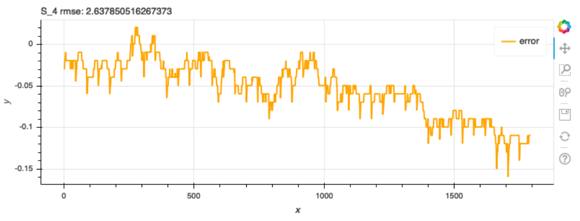
(b) S_1 Predictor



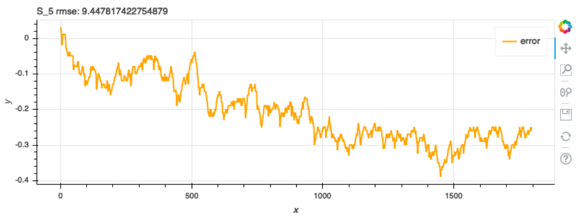
(c) S_2 Predictor



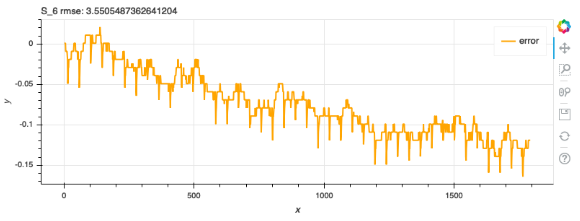
(d) S_3 Predictor



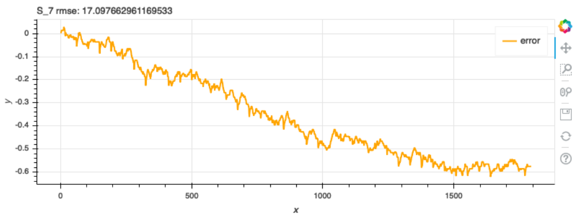
(e) S_4 Predictor



(f) S_5 Predictor

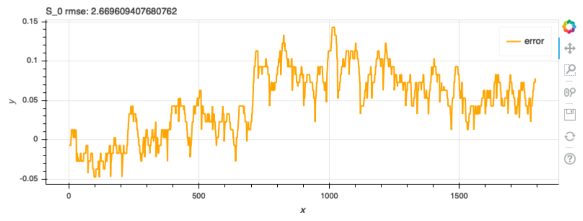


(g) S_6 Predictor

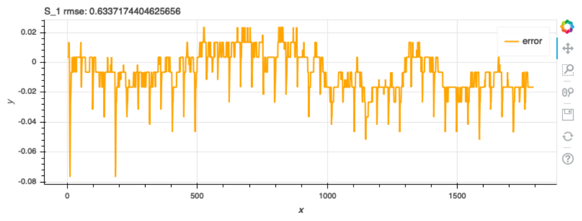


(h) S_7 Predictor

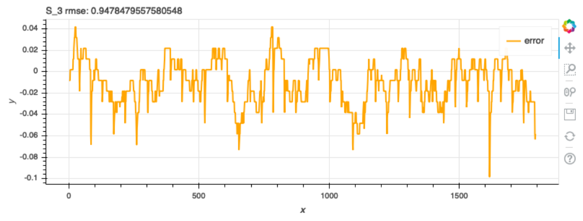
Figure B-44: Spread Out Short Term Performances For Average Significance Time Period



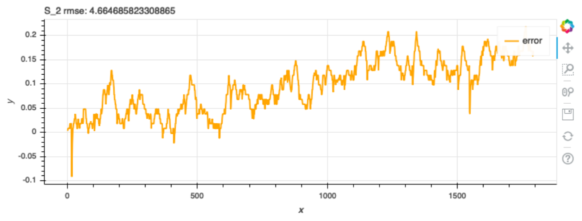
(a) S_0 Predictor



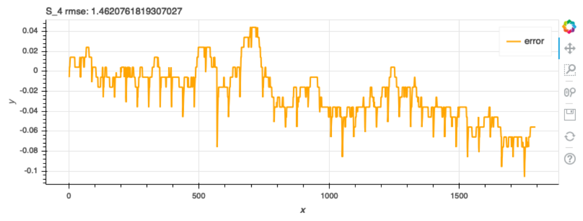
(b) S_1 Predictor



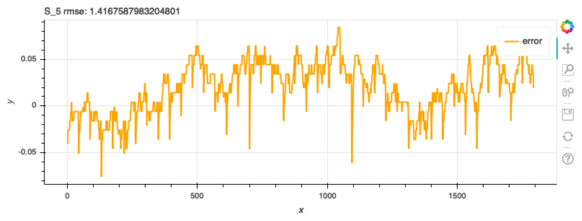
(c) S_2 Predictor



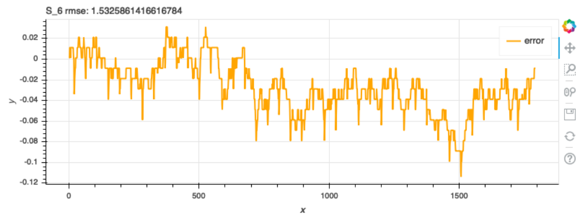
(d) S_3 Predictor



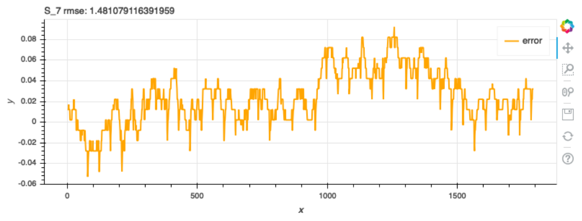
(e) S_4 Predictor



(f) S_5 Predictor

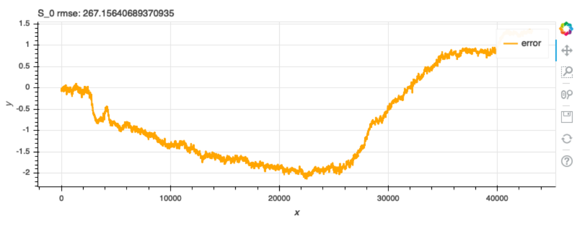


(g) S_6 Predictor

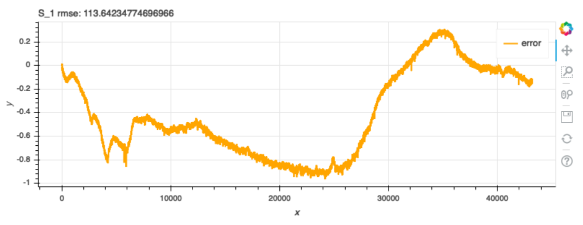


(h) S_7 Predictor

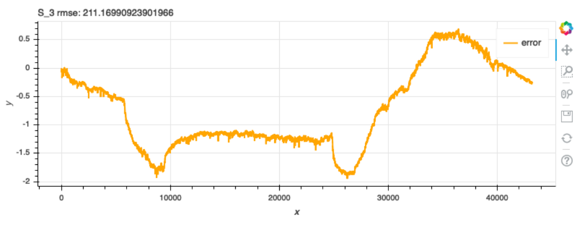
Figure B-45: Spread Out Short Term Performances For Least Significant Time Period



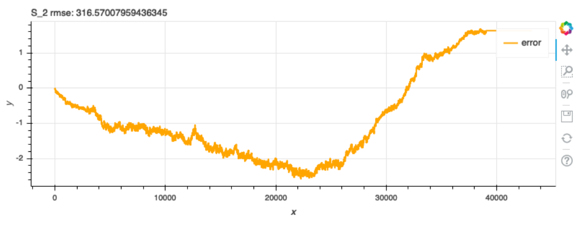
(a) S_0 Predictor



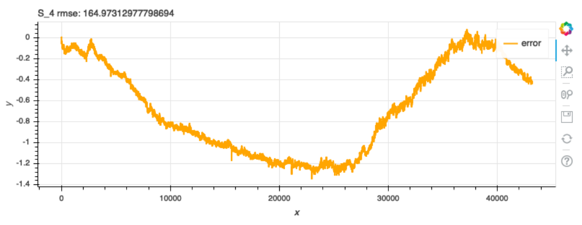
(b) S_1 Predictor



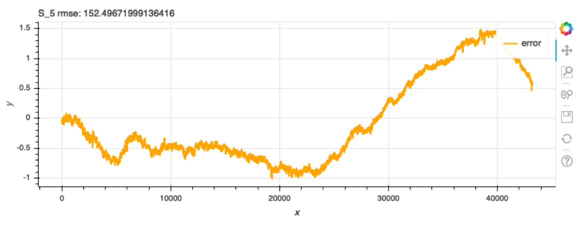
(c) S_2 Predictor



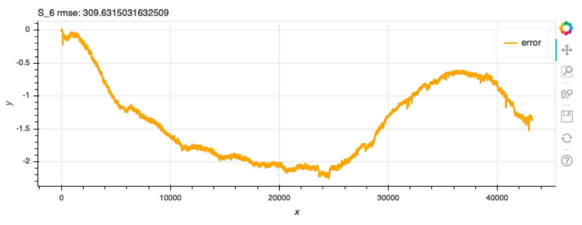
(d) S_3 Predictor



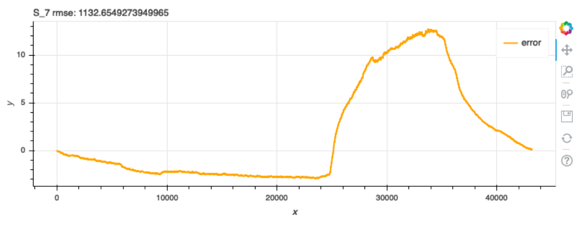
(e) S_4 Predictor



(f) S_5 Predictor

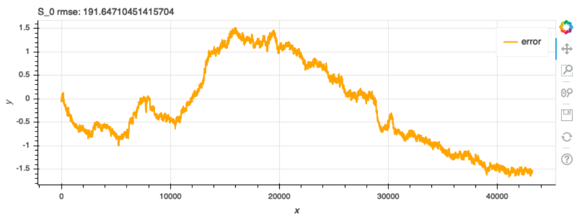


(g) S_6 Predictor

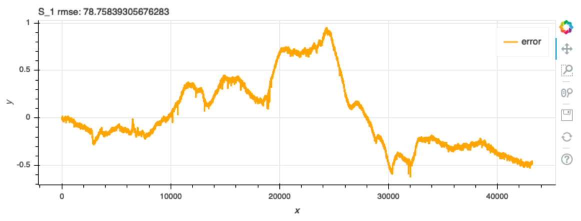


(h) S_7 Predictor

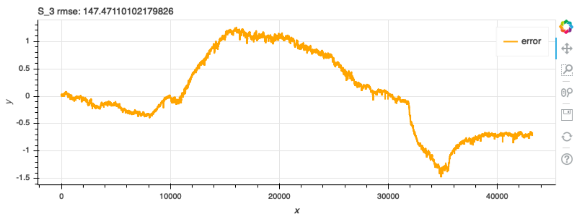
Figure B-46: Spread Out Long Term Performances For Most Significant Time Period



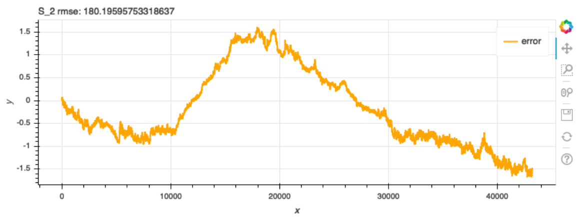
(a) S_0 Predictor



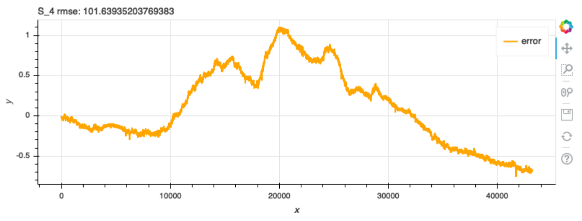
(b) S_1 Predictor



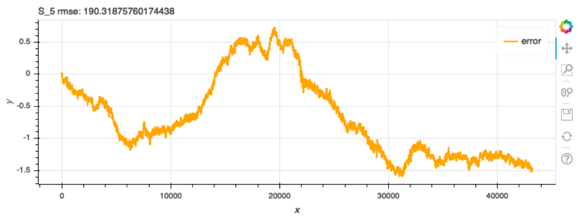
(c) S_2 Predictor



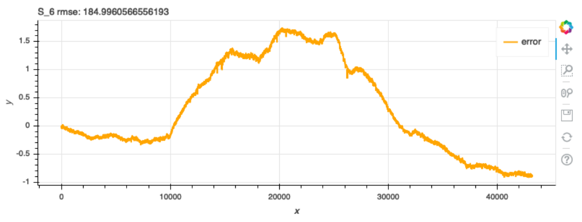
(d) S_3 Predictor



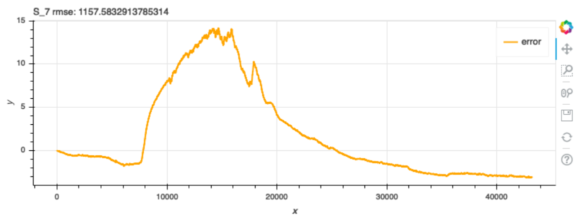
(e) S_4 Predictor



(f) S_5 Predictor

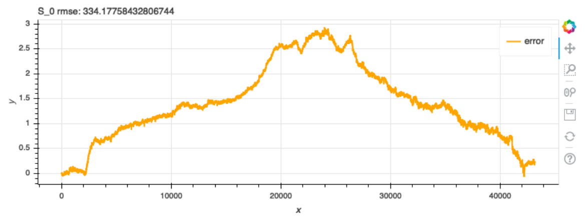


(g) S_6 Predictor

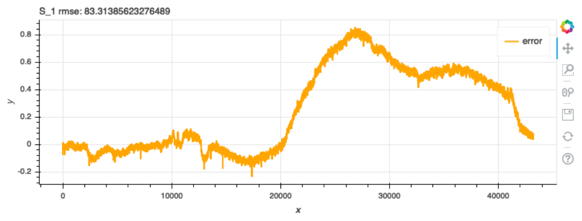


(h) S_7 Predictor

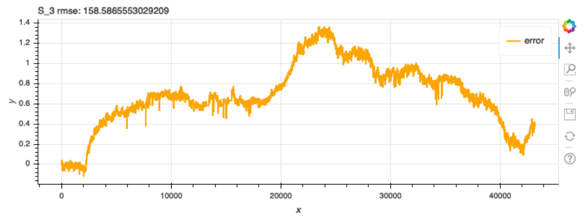
Figure B-47: Spread Out Long Term Performances For Average Significance Time Period



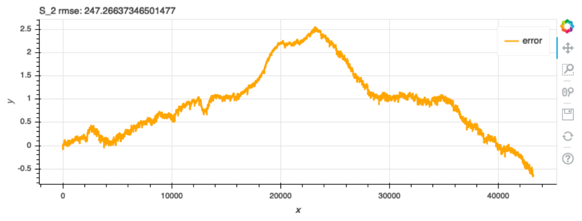
(a) S_0 Predictor



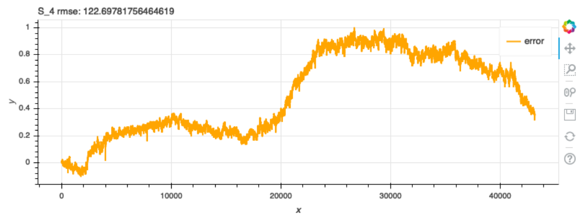
(b) S_1 Predictor



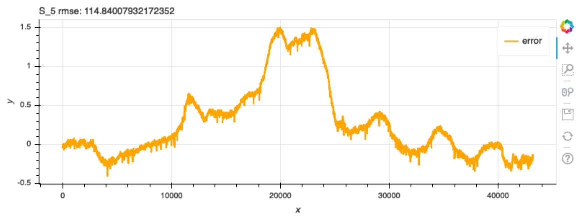
(c) S_2 Predictor



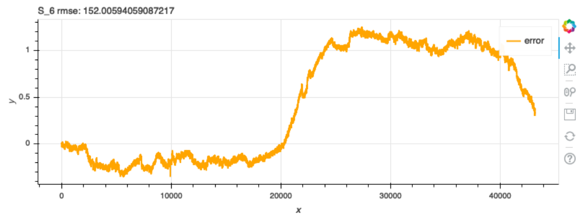
(d) S_3 Predictor



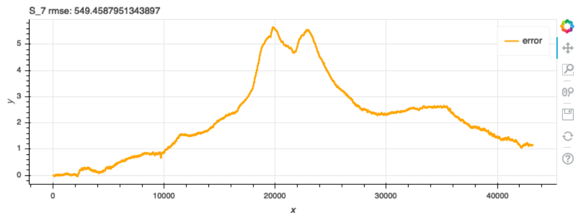
(e) S_4 Predictor



(f) S_5 Predictor



(g) S_6 Predictor



(h) S_7 Predictor

Figure B-48: Spread Out Long Term Performances For Least Significant Time Period

Bibliography

[1] B. Pratumvinit. The effects of temperature and relative humidity on point-of-care glucose measurements in hospital practice in a tropical clinical setting. Journal of Diabetes Science and Technology, 10(5):1094–1100, 2016.

[2] S. Li, S. Ren, and X. Wang. Hvac room temperature prediction control based on neural network model. In 2013 Fifth International Conference on Measuring Technology and Mechatronics Automation, pages 606–609, Jan 2013.

[3] R. I. Freshney. Culture of animal cells: a manual of basic technique. Wiley-Liss, 1994.

[4] C. Smith. Improvements in co2 incubators for cell cultures. Life Science Articles, 2015.

[5] D. Kattipparambil Rajan, J. Verho, J. Kreutzer, H. VÃďlimÃďki, H. Ihalainen, J. Lekkala, M. Patrikoski, and S. Miettinen. Monitoring ph, temperature and humidity in long-term stem cell culture in co2incubator. In 2017 IEEE International Symposium on Medical Measurements and Applications (MeMeA), pages 470–474, May 2017.

[6] S. Hempel, M. KÃűnig, C. Menz, D. Janke, B. Amon, T. M. Banhazi, F. EstellÃľs, and T. Amon. Uncertainty in the measurement of indoor temperature and humidity in naturally ventilated dairy buildings as influenced by measurement technique and data variability. Biosystems Engineering, 166:58–75, 2018.

[7] F. Jabeen and A. A. A. Fernandes. Distributed spatial analysis in wireless sensor networks. In 2010 IEEE 16th International Conference on Parallel and Distributed Systems, pages 558–567, Dec 2010.

[8] Alberto Cerpa, Jeremy Elson, Deborah Estrin, Lewis Girod, Michael Hamilton, and Jerry Zhao. Habitat monitoring: Application driver for wireless communications technology. Computer Communication Review, 31, 02 2001.

[9] Philo Juang, Hidekazu Oki, Yong Wang, Margaret Martonosi, Li-Shiuan Peh, and Daniel Rubenstein. Energy-efficient computing for wildlife tracking: Design tradeoffs and early experiences with zebranet. volume 37, pages 96–107, 10 2002.

[10] Nirupama Bulusu, Vladimir Bychkovskiy, Deborah Estrin, and John Heidemann. Scalable, ad hoc deployable rf-based localization. 02 2002.

[11] Tian He, Brian M. Blum, John A. Stankovic, and Tarek Abdelzaher. Aida: Adaptive application-independent data aggregation in wireless sensor networks. ACM Trans. Embed. Comput. Syst., 3(2):426–457, May 2004.

[12] L. Krishnamachari, D. Estrin, and S. Wicker. The impact of data aggregation in wireless sensor networks. In Proceedings 22nd International Conference on Distributed Computing Systems Workshops, pages 575–578, July 2002.

[13] Honglu Zhu, Jizhen Liu, Taihua Chang, and Liang Tian. Internal model control using lms filter and its application to superheated steam temperature of power plant. In 2010 The 2nd International Conference on Computer and Automation Engineering (ICCAE), volume 2, pages 135–138, Feb 2010.

[14] K. Chen, H. Li, and A. A. Wu. Lms-based adaptive temperature prediction scheme for proactive thermal-aware three-dimensional network-on-chip systems. In Technical Papers of 2014 International Symposium on VLSI Design, Automation and Test, pages 1–4, April 2014.

[15] T. Shu, J. Chen, V. K. Bhargava, and C. W. de Silva. An energy-efficient dual prediction scheme using lms filter and lstm in wireless sensor networks for environment monitoring. IEEE Internet of Things Journal, 6(4):6736–6747, Aug 2019.