Optimization of Biosensor Waveform Preprocessing

Siemens Healthineers

Weekly Report N°1

Dylan Longert
Jinxin (Jessie) Wang
Nan Tang
Nayeli Montiel Rodríguez
Han (Eden) Chen

1. Exploratory Data Analysis

In this section, we provide an overview of the characteristics of the six datasets provided by Siemens Healthineers, which will be the subject of our analysis. The EDA is divided into three subsections: first, the time series data, where four datasets consist of measurements for two different systems, allowing us to examine trends and patterns over time. Next, we describe the additional features related to the epoc® Blood Analysis System, including the specific conditions under which the data was collected. Lastly, we analyze the dataset containing values for specific time intervals (windows) within the signal data, representing periods of interest such as the calibration window and the sample window. By exploring these datasets, we aim to uncover key insights and patterns that will inform subsequent stages of our analysis.

1.1 Time Series Data

The time series datasets are each structured identically and contain multiple time-series signals, referred to as waveforms or readings. A sneak peek of the structure of these datasets can be seen in Table 1. In these files, the first column represents time in seconds, while the subsequent columns contain the target variable, which is the electrical signal values from the potentiometric sensors in millivolts (mV) for successful test runs. Specifically, two files belong to System 1 and two to System 2, corresponding to sensors A and B, respectively. There are a total of 7,039 time series for System 1 (3,524 for Sensor A and 3,515 for Sensor B) and 15,561 time series for System 2 (7,781 for Sensor A and 7,780 for Sensor B).

Missing values are apparent in the time series datasets, as shown in Figures 1 to 4. The counts are as follows 4,758,066 for System 1 - Sensor A, 7,121,978 for System 1 - Sensor B, 629,083 for System 2 - Sensor A, and 163,348,463 for System 2 - Sensor B. These can be attributed to differences in the lengths of time series across various tests.

Table 1. Structure of the Time Series Data

Time (s)	TestID_1	TestID_2	TestID_3	TestID_4
0.2	-1797.2	1744.735	-1797.314	-231.9601
0.4	-1797.224	1749.621	-1797.341	-240.4039
0.6	-1797.221	1756.121	-1797.333	-248.0002
0.8	-1797.198	1762.368	-1797.349	-254.0748

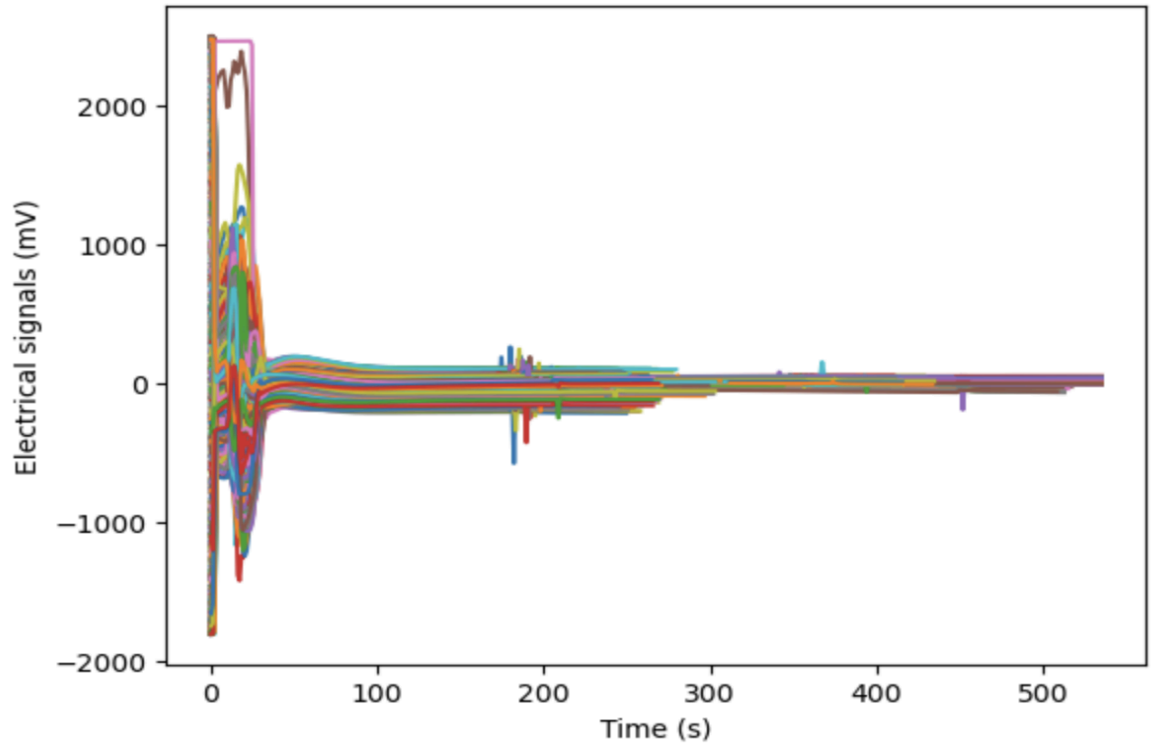

Figure 1. System 1 - Sensor A

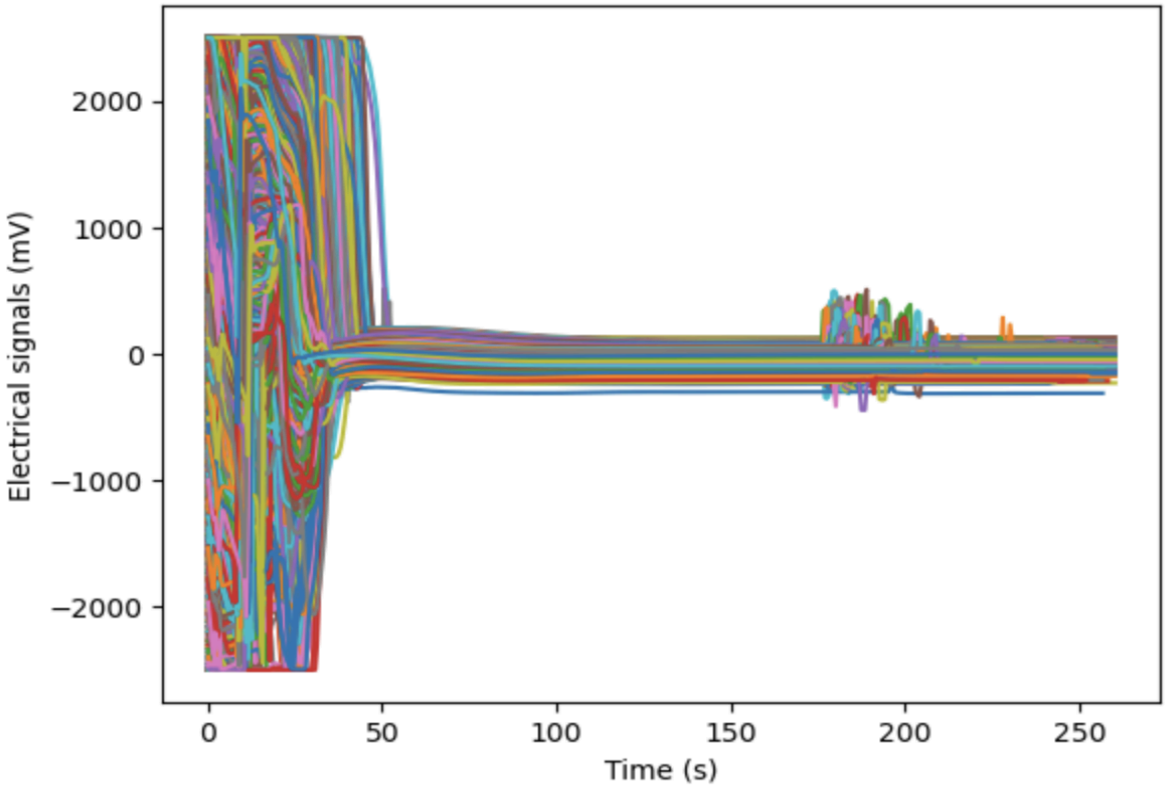

Figure 2. System 2 - Sensor A

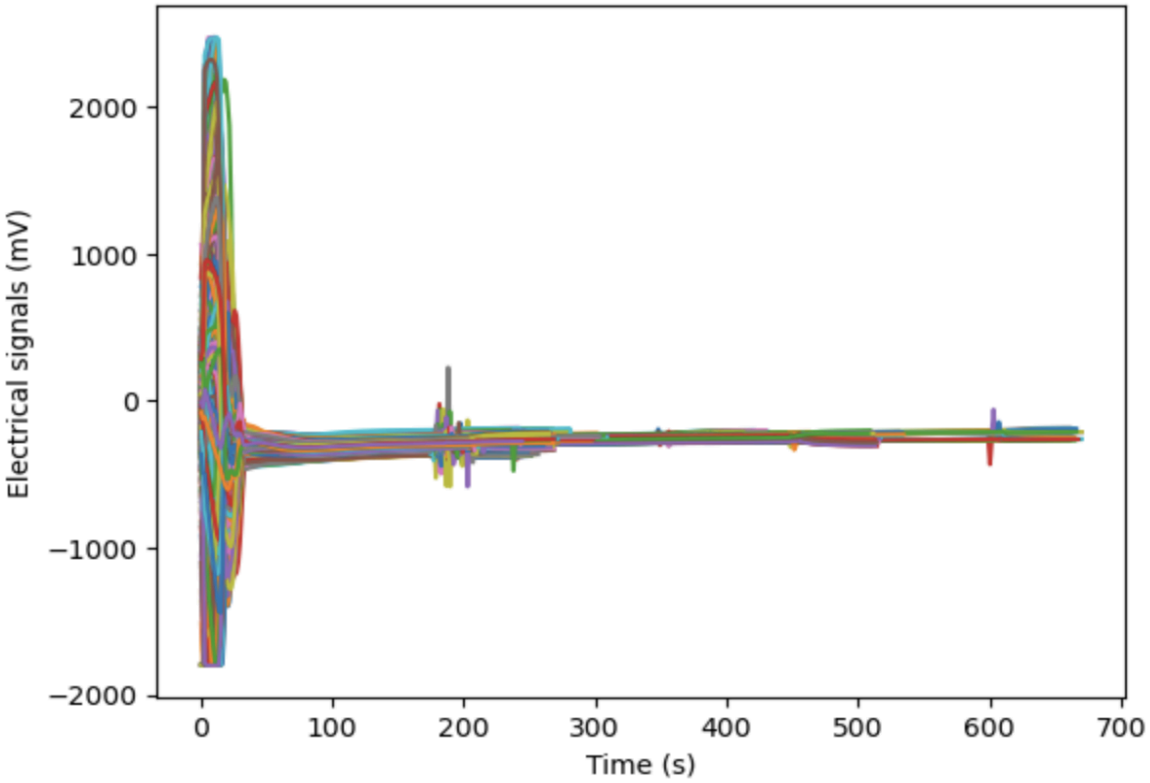

Figure 3. System 1 - Sensor B

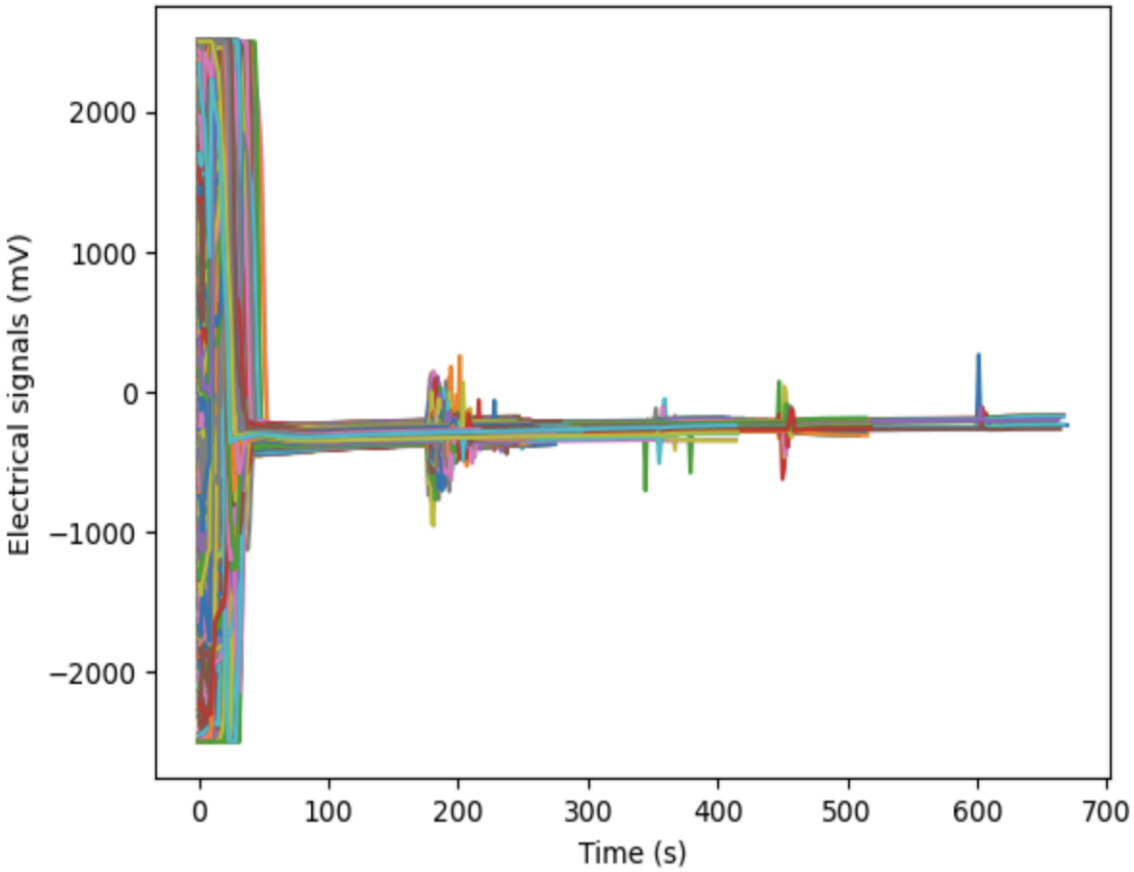

Figure 4. System 2 - Sensor B

1.2 Additional Features

The file with additional information of the time series contains 18 columns, each providing supplementary information for each unique test run. These columns include one date type variable (DateTime), one string variable (Lot), eight categorical variables (System, Sensor, FluidType, and four return codes derived from the readings), and eight numeric variables (TestID, CardNumber, AgeofCardInDaysAtTimeOfTest, ReaderSerialNumber, SampleDetectTime, BubbleDetectTime, AmbientTemperature, and FluidTemperature). Further details on these variables can be seen in Table 2.

To align the time series data with the additional features, we merged each time series dataset with the features dataset by TestID, revealing 117 unmatched TestIDs in both files for System 1 (Sensor A and Sensor B).

Moreover, missing values are evident in the additional features, including ‘Fluid Temperature’, ‘Device Return Code’, and ‘Bge Test ReturnCode’ for System 1 (Sensor A and Sensor B), with over 95% of values missing for ‘Fluid Temperature’ and 100% for both ‘Device Return Code’ and ‘Bge Test ReturnCode’. Furthermore, missing values were identified in ‘Fluid Temperature’ and ‘CardReturnCode’ for System 1 (Sensor A and Sensor B), with over 97% of values missing for ‘Fluid Temperature’ and 100% for ‘CardReturnCode’.

1.2.1. Date Features

For System 1 (Sensor A and Sensor B), tests were conducted between October 31st, 2023, and September 4th, 2024, with the majority taking place in February 2024. This is evident as both the mean and median fall within this month.

Similarly, for System 2 (Sensor A and Sensor B), tests were conducted from September 19th, 2023, to April 4th, 2024, with the majority also occurring in February 2024. This is indicated by both the mean and median aligning with this month.

1.2.2. String Features

The tests conducted for System 1 (comprising Sensor A and Sensor B) were sourced from 49 card lots, each with varying batch sizes ranging from 23 to 237 cards, all manufactured within a single day. Similarly, for System 2 (also featuring Sensor A and Sensor B), the tests were derived from batches with batch sizes ranging from 26 to 483 cards, all produced within a single day. Notably, cards within the same lot share identical sensor print mixes, thereby resulting in similar values on the response variable.

Table 2. Additional Features Description

Name of the variable	Description
TestID	Unique identifier of the test run created at the time of the test.
FluidType	The sample injected into the card.
DateTime	The date and time that the test was run.
Lot	Card lot. All cards built in one batch on one day.
CardNumber	Number that identifies the card that was used for running the test.
AgeOfCardInDaysAtTimeOfTest	Age of the card (in days) at the time of the test. Cards expire after 168 days.
ReaderSerialNumber	Unique identifier of the reader that ran the card.
SampleDetectTime	The time (in seconds) from the start of the test to when the sample fluid front is detected.
BubbleDetectTime	The time (in seconds) from the start of the test to when the bubble between calibration fluid and sample fluid is detected.
System	Identifies the system that runs the test.
Sensor	A or B.
AmbientTemperature	If fluid temperature is blank, assume this is the temperature of the entire system (card, reader, host, and fluid).
Fluid Temperature	Temperature of the sample fluid.
CardReturnCode	Determine whether or not the test was successful.
ReturnCode	Determine whether or not the test was successful.
Results Error Code	Determine whether or not the test was successful.
Device Return Code	Determine whether or not the test was successful.
Bge Test Return Code	Determine whether or not the test was successful.

1.2.3. Categorical Features

The tests were conducted using two different sensors, denoted as A or B, and two systems referred to as ‘System 1’ for the current system, and ‘System 2A’ and ‘System 2B’ both referring to the new system. Table 3 provides a summary of the number of test runs for each system and sensor.

The fluid samples injected onto the card fall into two categories: aqueous, represented in the dataset by Eurotrol L1, Eurotrol L3, Eurotrol L4, and Eurotrol L5, or blood, with values NB, AB, HNB, TB11, DB, SB, and SB-3 (Table 4). Experts have noted that the type of fluid significantly influences the readings of our response variable.

Table 3. Number of tests run by system and sensor after merging with additional features.

System	System 1		System 2	
	Sensor A	Sensor B	Sensor A	Sensor B
System 1	3,407	3,398	-	-
System 2A	-	-	5,365	5,365
System 2B	-	-	2,416	2,415

Table 4. Number of tests run by fluid type.

Fluid Type	Fluid Name	System 1		System 2		Total
		Sensor A	Sensor B	Sensor A	Sensor B	
Aqueous	Eurotrol L1	617	617	1,495	1,494	4,223
	Eurotrol L3	602	602	1,629	1,629	4,462
	Eurotrol L4	317	312	525	525	1,679
	Eurotrol L5	423	423	1,263	1,263	3,372
Blood	NB	594	593	1,008	1,008	3,203
	AB	232	232	728	728	1,920
	HNB	189	189	402	402	1,182
	TB11	175	174	225	225	799
	DB	117	117	210	210	654
	SB	111	109	260	260	740
	SB-3	30	30	36	36	132
	Total	3,407	3,398	7,781	7,780	22,366

Finally, there are five categorical variables featuring various return codes crucial for experts to determine test success. Tables 5 and 6 display the distribution of test runs for ‘CardReturnCode’ and ‘ReturnCode’. Additionally, among these codes, ‘Device Return Code’ solely contains ‘NoError’ values; however, all observations are absent for System 1 (Sensor A and Sensor B), while this code was returned for every test run on System 2 (7,781 for Sensor A and 7,780 for Sensor B). Similarly, ‘Bge Test ReturnCode’ exclusively comprises the value ‘Success’; yet, all observations are missing for System 1 (Sensor A and Sensor B), while this code was returned for all test runs on System 2 (7,781 for Sensor A and 7,780 for Sensor B).

Table 5. Number of tests run by Card Return Code.

Card Return Code	System 1		System 2	
	Sensor A	Sensor B	Sensor A	Sensor B
CalNotDetected	1	1	NaN	NaN
EarlyInjection	15	15	NaN	NaN
FluidicsFailedQCDuringCalibration	6	6	NaN	NaN
HematocritLowResistance	1	1	NaN	NaN
HumidityCheckFailedLooseLimit	3	3	NaN	NaN
NoError	3,288	3,281	NaN	NaN
RealtimeQCFailedDuringFluidics	32	31	NaN	NaN
SampleFailedQC	45	44	NaN	NaN
SampleInjectedTooFast	4	4	NaN	NaN
SampleInjectedTooSlowly	12	12	NaN	NaN
Total	3,407	3,398	NaN	NaN

Table 6. Number of tests run by Return Code.

Return Code	System 1		System 2	
	Sensor A	Sensor B	Sensor A	Sensor B
AdditionalDriftHigh	1	NaN	NaN	NaN
CalDriftQCHigh	2	1	NaN	NaN
CalNoiseQCHigh	3	7	2	4
CannotCalculate	15	3	34	10
EarlyDipInSample	1	4	NaN	NaN
EarlySpikeInSample	1	3	NaN	NaN
LateDipInSample	NaN	1	1	NaN
LateSpikeInSample	2	NaN	NaN	4
PostNoiseQCHigh	NaN	3	1	4
SampleNoiseQCHigh	NaN	1	NaN	13
Success	3,382	3,361	7,743	7,745
UnderReportableRange	NaN	14	NaN	NaN
Total	3,407	3,398	7,781	7,780

1.2.4. Numeric Features

In addition to the identifiers outlined in Table 1, our study encompassed various other noteworthy attributes. Beginning with card age in Figures 5 and 6, System 1, which employed Sensor A and Sensor B, was tested on cards ranging from 5 to 240 days old. The average age of cards processed by Sensor A within System 1 was 102.668 days, slightly higher than the average of 102.434 days for Sensor B. Interestingly, both sensors in System 1

recorded a median age of 92 days. In contrast, cards utilized in System 2, where Sensor A and Sensor B were also employed, exhibited an age range of 5 to 240 days, with average ages of 82.502 days and 82.500 days for Sensor A and Sensor B, respectively. Notably, the median age for both sensors in System 2 was 68 days.

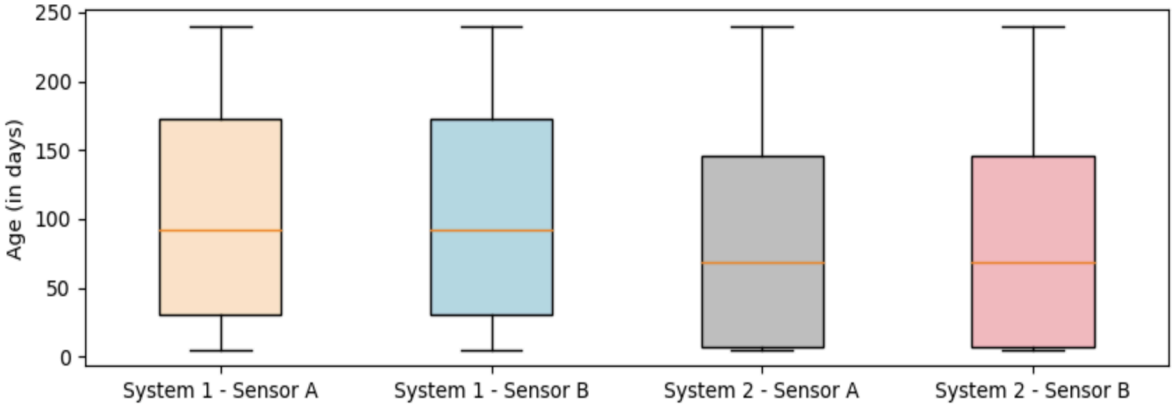

Figure 5. Boxplots of the age of the card.

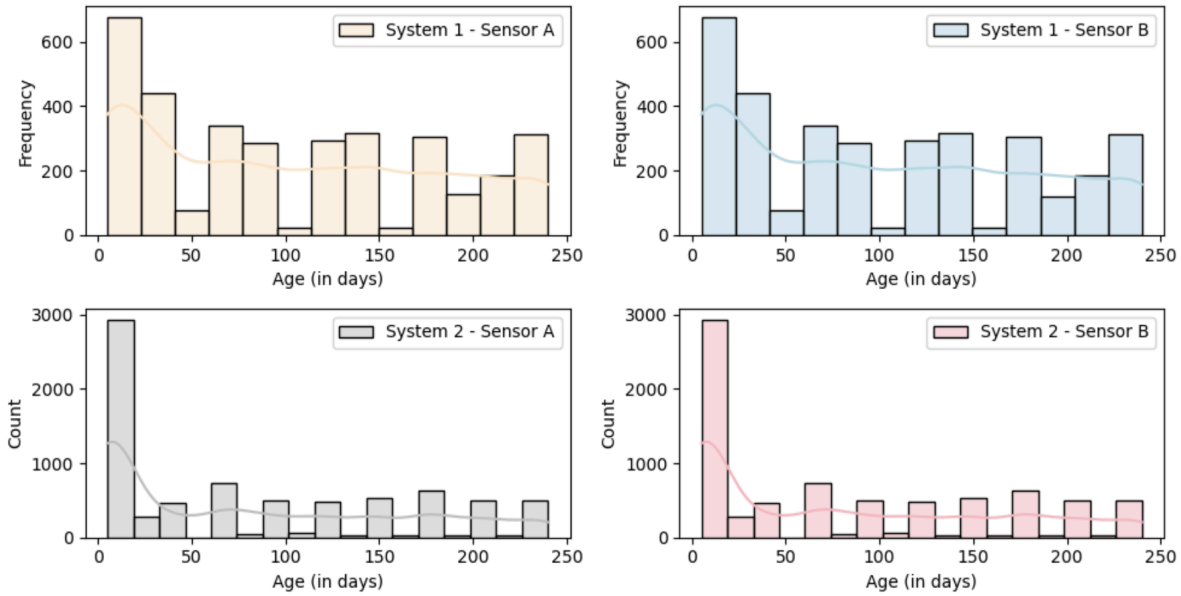

Figure 6. Histograms of the age of the card.

Moving on to the ‘SampleDetectTime’ in seconds which distribution can be seen in Figures 7 and 8. In System 1, Sensor A exhibits a mean of 202.464 seconds, while Sensor B has a slightly lower average of 202.456 seconds. However, the median for both sensors in System 1 was 186 seconds. On the other hand, System 2 shows an average of 189.729 seconds for Sensor A and 182.731 seconds for Sensor B, with the same median of 182 seconds.

Examining ‘BubbleDetectTime’ in System 1, Sensor A exhibits a mean of 201.427 seconds, while Sensor B has a slightly lower average of 202.419 seconds. However, the median for both

sensors in System 1 was 185 seconds. On the other hand, System 2 shows an average of 187.720 seconds for Sensor A and 187.722 seconds for Sensor B, with the same median of 179.6 seconds. Figures 9 and 10 show the distribution of bubble detect time, which by eyeball looks really similar to the distribution of sample detect time.

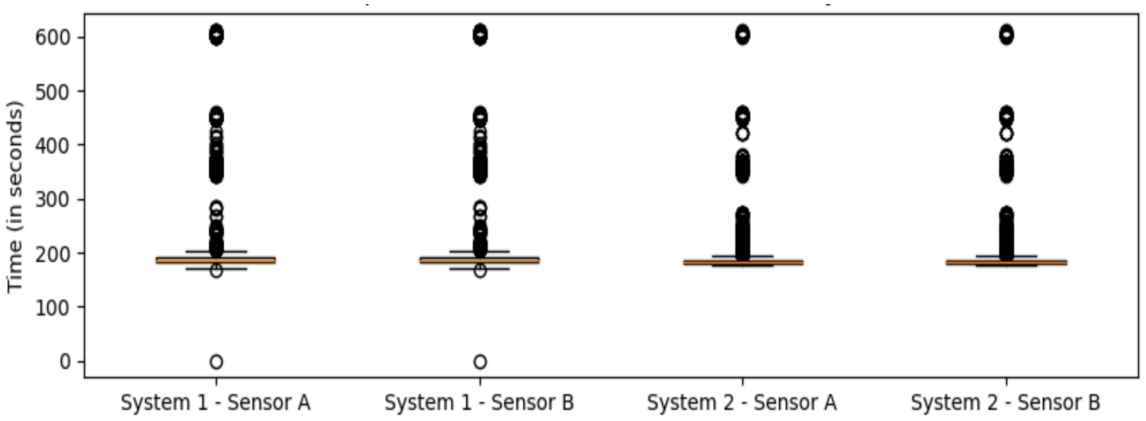

Figure 7. Boxplots of the sample detect time in seconds.

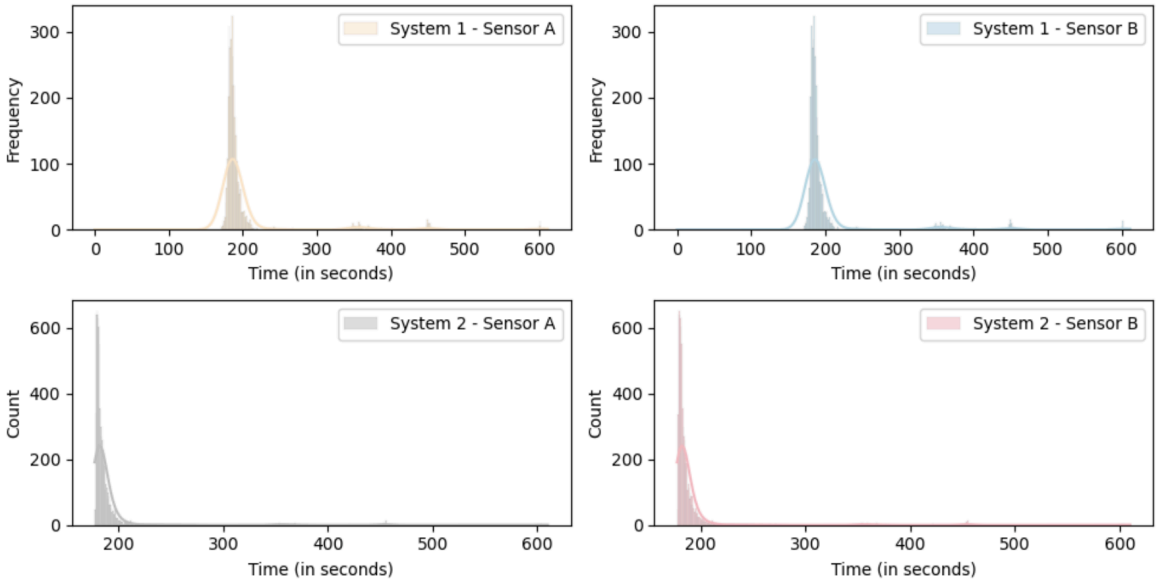

Figure 8. Histograms of the sample detect time in seconds.

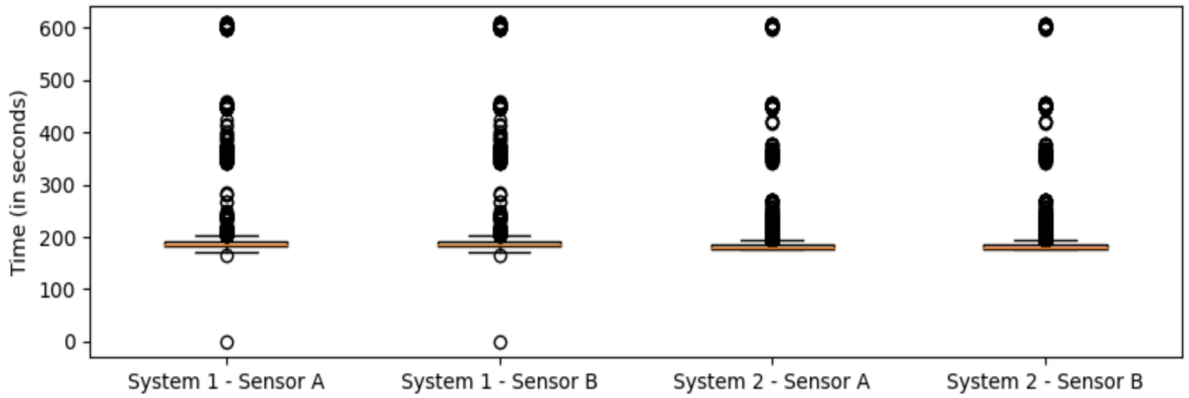

Figure 9. Boxplots of the bubble detect time in seconds.

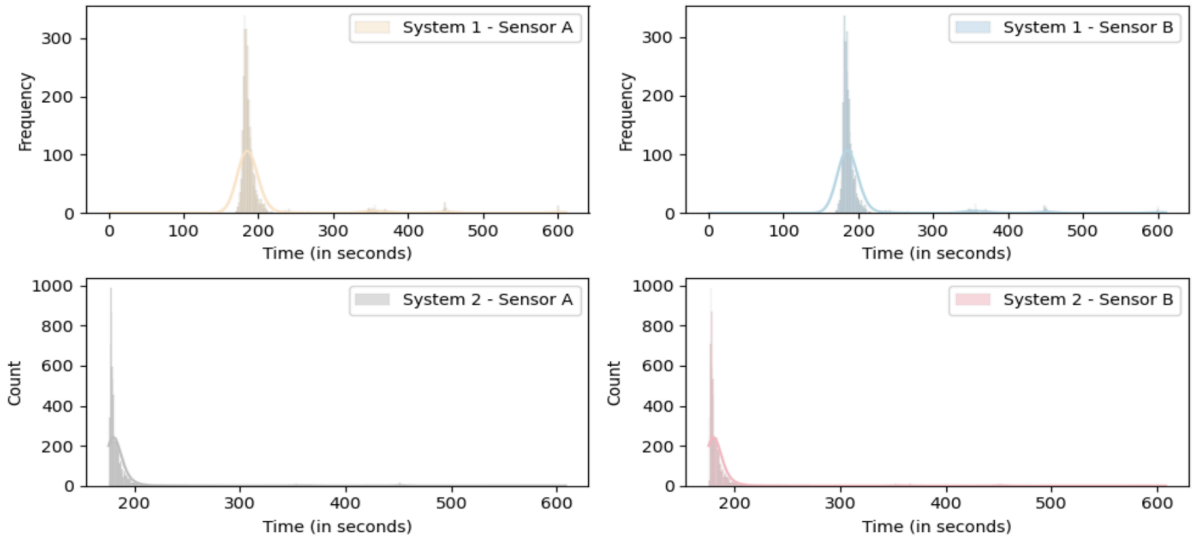

Figure 10. Histograms of the bubble detect time in seconds.

Finally, we examined ‘Ambient temperature’ and ‘Fluid Temperature’. Notably, ‘Fluid Temperature’ exhibited a significant number of missing values. As per expert recommendation, in cases where this value is absent, the ambient temperature represents the entire system temperature, encompassing the card, reader, host, and fluid components. Conversely, when ‘Fluid Temperature’ is available, it takes precedence.

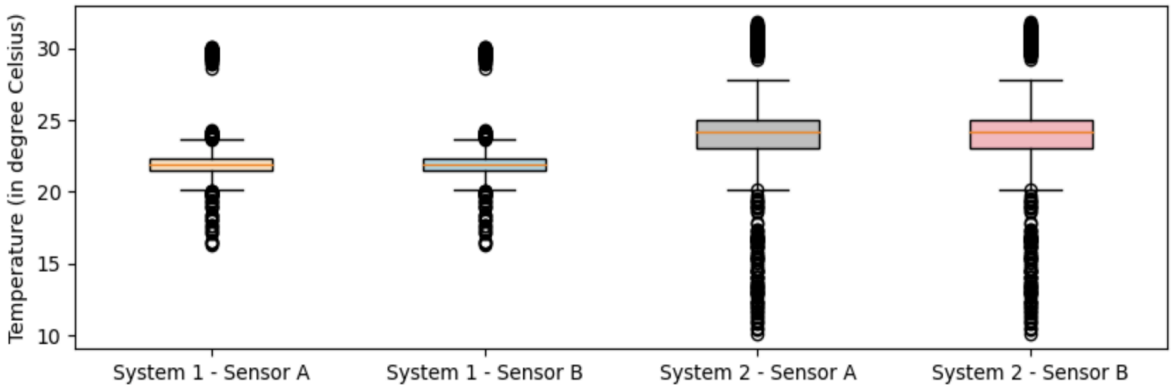

Figure 11. Boxplots of the ambient temperature in degree Celsius.

Interesting observations emerged from System 1, where the ‘Ambient Temperature’ ranged between 16.35 to 30.02 degrees Celsius and the ‘Fluid Temperature’ range was 15 to 30 degree Celsius. Remarkably, the average ‘Ambient Temperature’ and ‘Fluid Temperature’ were closely aligned, measuring 22.00 and 22.26 degrees Celsius for Sensor A and Sensor B, respectively. However, in System 2, the ‘Ambient Temperature’ exhibited a wider range from 10.12 to 31.85 degrees Celsius, while the ‘Fluid Temperature’ maintained a narrower range between 15 and 30 degrees Celsius. The mean values for ‘Ambient Temperature’ and ‘Fluid Temperature’ were approximately 23.96 and 22.27 degrees Celsius, respectively.

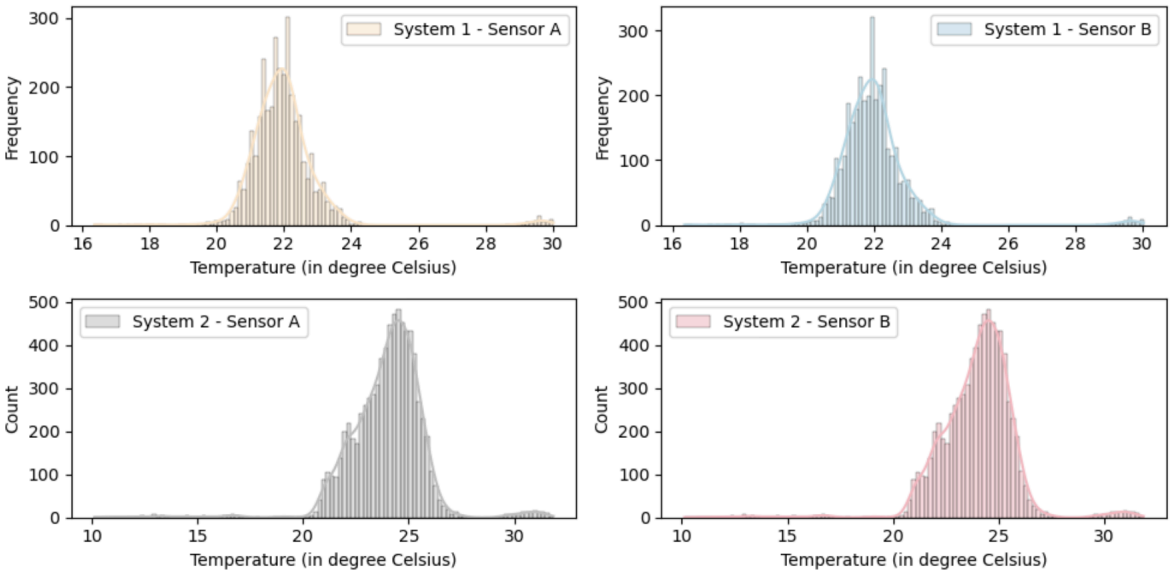

Figure 12. Histograms of the ambient temperature in degree Celsius.

1.3. Current Window Limits

This Excel file contains a table listing the fixed values for the boundaries currently used by Siemens Healthineers to determine the periods of interest in the time series. Specifically, different values are applied to readings from Sensor A, Sensor B with blood fluids, and Sensor B with aqueous fluids. Figure 13 presents a diagram of these periods of interest, which we will refer to as the CalWindow and Sample Window.

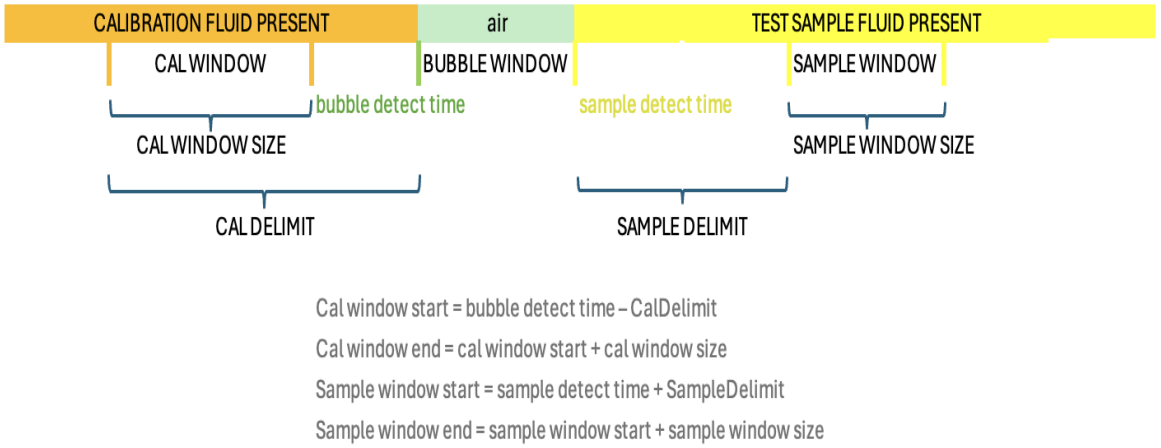

Figure 13. Current window partitioning.

2. Functional Data Analysis

2.1. Functional Principal Components

Functional principal component analysis (FPCA) is a statistical method used to analyze data that can be represented as functions rather than vectors. FPCA extends principal component analysis (PCA) to functional data, providing a way to identify the dominant modes of variation of the data throughout the visualization of the eigenfunctions [1]. In simple terms, the functional principal components show the patterns of variation of the data that differ from the mean function, which is the average curve of all the curves in the data.

Moreover, the Principal Component scores (PCscores) are numerical representations of each observation's contribution to the principal components extracted through functional PCA. These scores indicate the degree to which each observation aligns with the principal components and provides insights into the underlying patterns or structures within the data.

In this project, we employed the scikit-fda library [2] to implement FPCA on a grid representation of the data, where the grid consists of the number of points compressed by the time series extracted from the original data. The code can be seen in the file "Siemens_PCA_FPCA.ipynb" within the public GitHub repository.

The primary goal is to reduce the dimensionality of multiple time series for visualization of two curves that summarize the key features from all the time series from the windows. To illustrate this approach, we present our analysis divided into two subsections. The first section demonstrates Functional Principal Component Analysis (FPCA) on the dataset inclusive of erroneous error codes. In the second section, we provide an example post-removal of these erroneous error codes, which Siemens Healthineers confirmed during our recent meeting should be eliminated as part of the preprocessing phase. In both cases, the plots depict time series data for Sensor A. However, the plots for Sensor B data are also available in the code.

2.1.1 Analysis before removing bad codes

The figures (Figures 14 and 15) display the most important two features for Sensor A across System 1 and System 2 for the CalWindow and Sample Window respectively. When

comparing horizontally, a notable distinction between System 1 and 2 becomes evident in both Calibration (Cal) and Sample windows. Conversely, in the vertical direction, a divergent pattern emerges in the Cal and Sample windows of System 1, while they appear similar in System 2, yet with slight variations in scale.

Specifically, as we can see in Figures 14 and 15, the first principal component (illustrated by the Blue line) captures the primary behavior of the dataset. At initial observation, this component appears nearly flat, a characteristic that aligns seamlessly with the findings derived from window visualizations. Remarkably, in the four cases the first principal component explains an overwhelming majority of the dataset's variability, accounting for more than 99.99%, underscoring its significance in encapsulating the underlying structure. In contrast, the contribution of the second component is notably minor, explaining less than 0.01% of the variability, further highlighting the dominance of the first principal component in characterizing the dataset's variability.

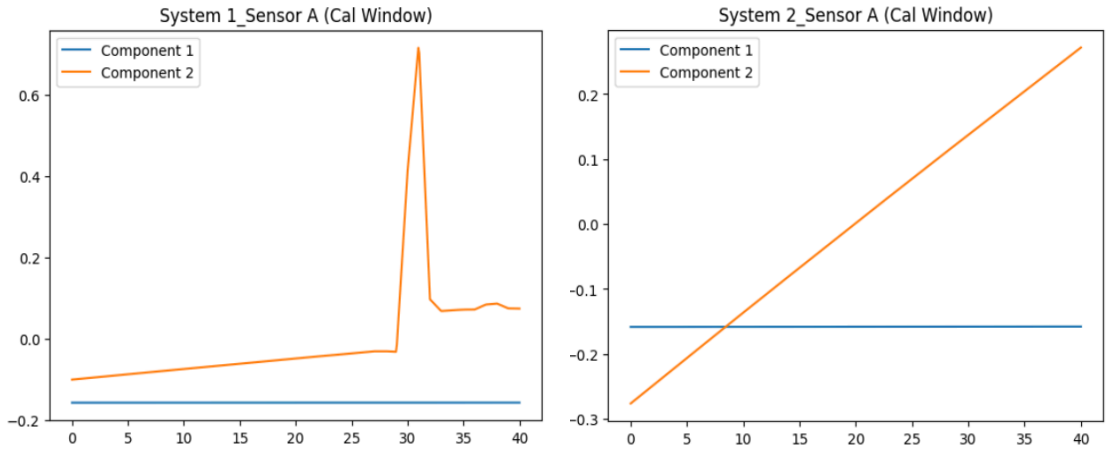

Figure 14. First two principal components for Sensor A from the Cal Windows.

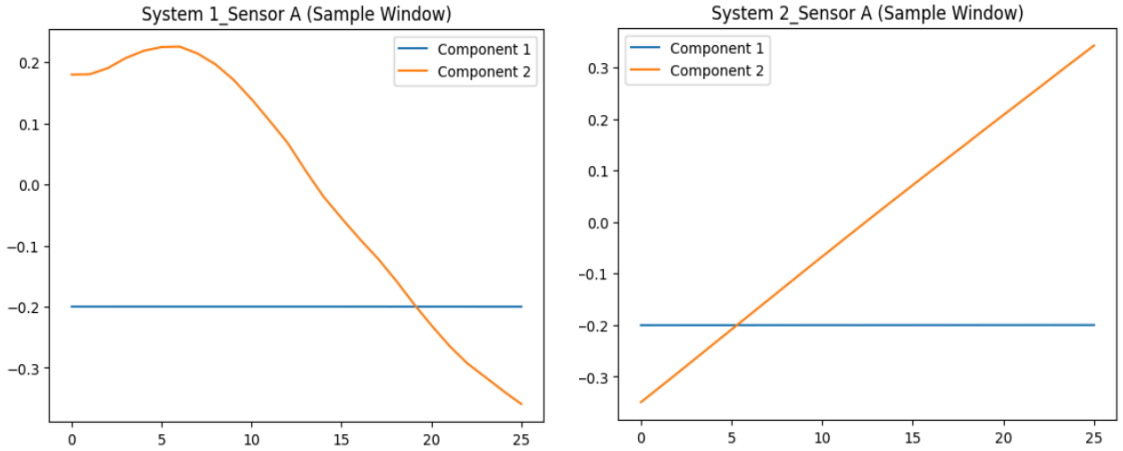

Figure 15. First two principal components for Sensor A from the Sample Windows.

Due to scale differences, distinguishing the flat lines between the two systems directly is challenging. Consequently, we zoom in on the first two plots. As depicted in Figure 16 and 17 below, the scales of the flat lines in both systems are identical. However, System 2 lacks the curved shape (e.g., trough) observed after 25 seconds, unlike System 1. Similar analysis for the Sample window is shown in Appendix 1.

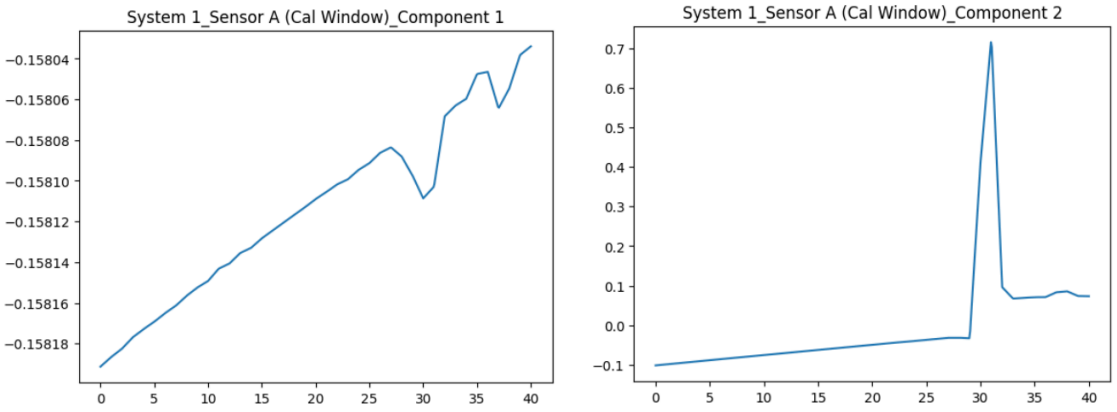

Figure 16. Zoom-in components in System 1 SensorA Cal Window.

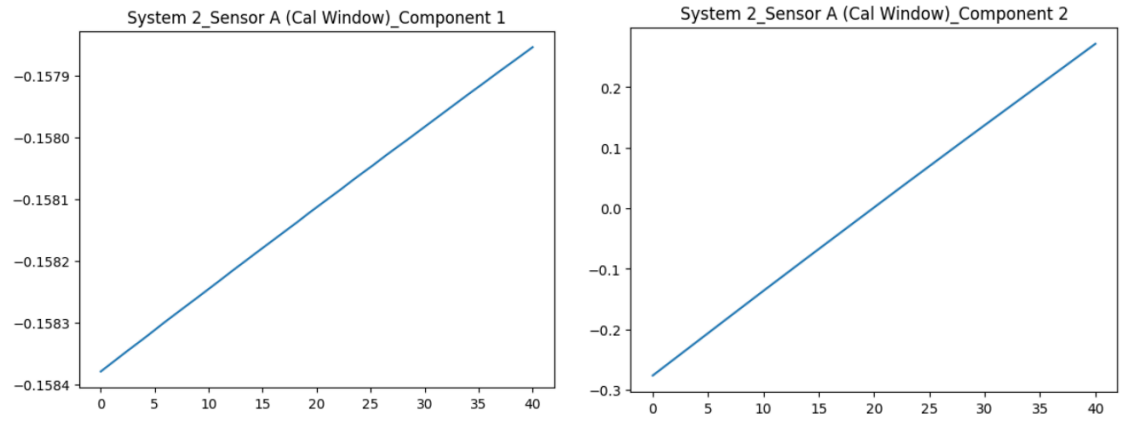

Figure 17. Zoom-in components in System 2 SensorA Cal Window.

Additionally, through the PC scores we identify the identifiers of the time series that contribute the most in the shapes of the principal components. Particularly, the time series with TestID 12515535 emerges as the key contributor to principal component two. After a careful verification, we found that this waveform had the error return code 'EarlyInjection'. Similarly, the waveform with TestID 12371902 which contributes the most in component two shapes had an error return code 'SampleFailedQC'.

2.1.2 Analysis without bad codes

In this section we show a similar analysis after the removal of the waveforms with error bad codes. For simplicity, we display plots just for System 1 - Sensor A and System 2 - Sensor A, but we can refer to the code for the analysis splitting the curves into Blood and Aqueous groups.

The figures (Figures 18 to 19) below display the results of FPCA after removing the 25 waveforms and the 38 waveforms with bad error codes from System 1 - Sensor A and System 2 - Sensor A, respectively. As we can see, when comparing Figure 14 with Figures 24 and 25 the new curves for component two depict a linear function.

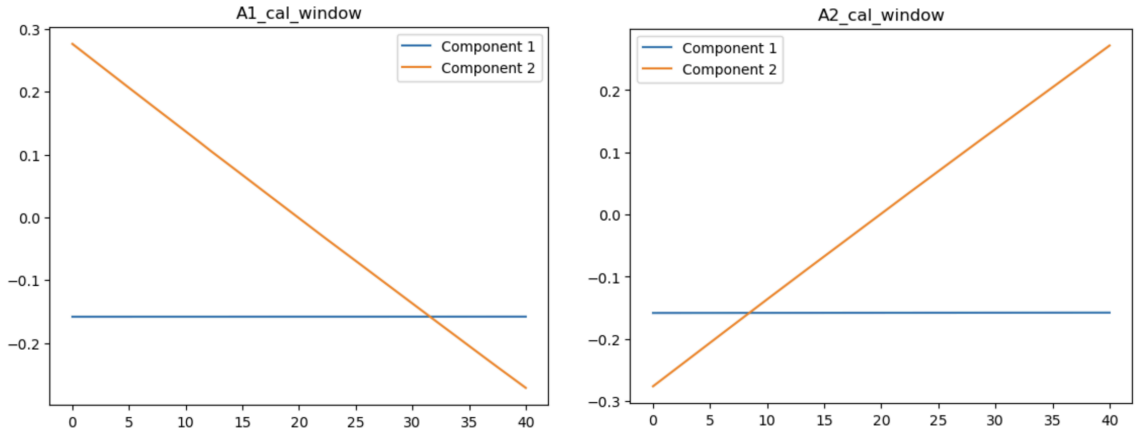

Figure 18. First two principal components for Sensor A from the Cal Windows.

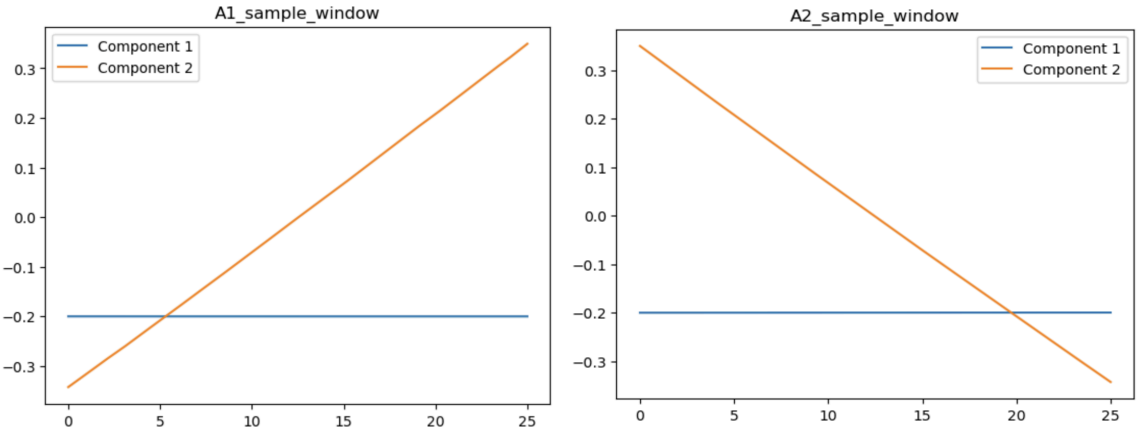

Figure 19. First two principal components for Sensor A from the Sample Windows.

Additionally, zooming in (Figures 20 and 21) reveals more regular linear shapes for both components.

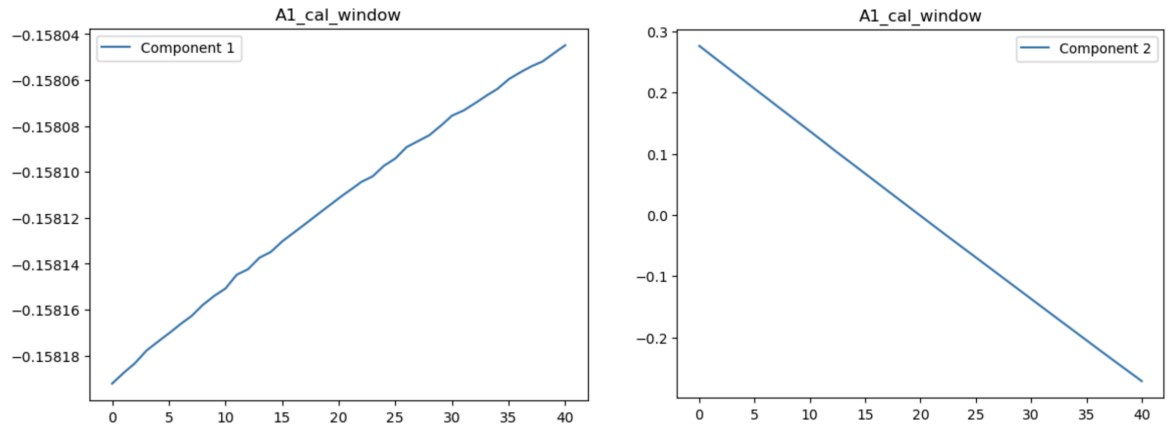

Figure 20. Zoom-in components in System 1 SensorA CalWindow.

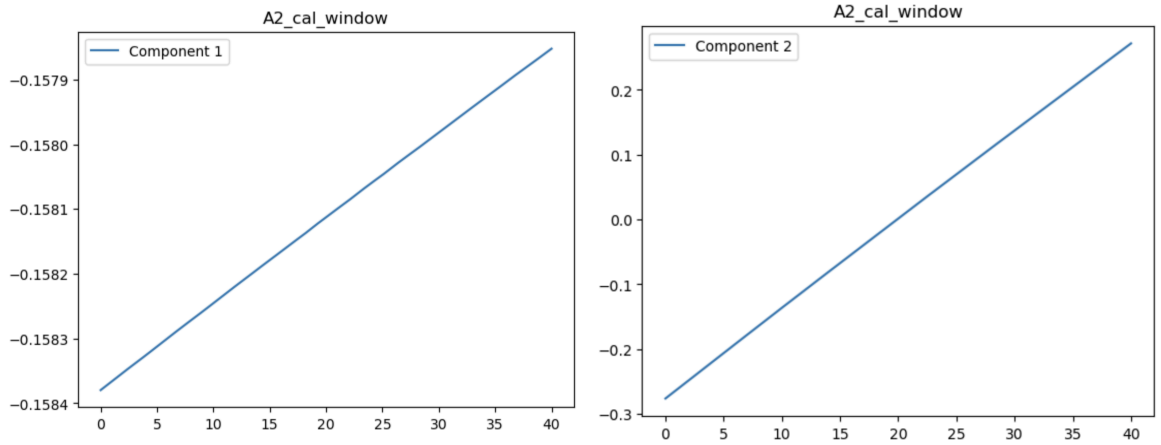

Figure 21. Zoom-in components in System 2 SensorA Cal Window.

Overall, this analysis helps us to identify one of the main benefits of functional principal components for outlier detection. As stated in [3] "FPCA methods are not robust against outliers because they involve the use of second order moments. Outliers for functional data have many different facets due to the high dimensionality of these data. They can appear as outlying measurements at a single or several time points, or as an outlying shape of an entire function".

Moreover, if the team decides to continue with this approach, FPCA can be employed in functional regression by projecting functional predictors to the first few principal components. However, guidance using functional data analysis is required due to the complex mathematics behind this statistical branch, which operates on Hilbert space instead of Euclidean space.

3. Time Series Clustering

To explore potential distinctions in biosensor waveforms between System 1 and System 2, we conducted two distinct analyses:

1. We applied k-means, hierarchical, and dynamic time warp (DTW) clustering to the sensor data values.

2. We applied k-means, hierarchical, and dynamic time warp (DTW) clustering to the sensor attributes, we focused on sensor fluid type and sensor card age.

Hierarchical clustering organizes data into clusters using a distance matrix, with the results often visualized via a dendrogram. K-means clustering, on the other hand, partitions data into clusters based on proximity, with the number of clusters predetermined or determined during analysis. DTW clustering leverages the DTW algorithm to measure the similarity between two given data sequences, making it particularly useful for comparing time-series data with different temporal patterns.

Following consultation with Siemens, it was advised to focus the clustering analysis solely on sensor data values rather than attributes. Therefore, the preliminary results presented here are exclusively derived from sensor data values.

Initially, data from both the calibration and sample windows of System 1 (Sensor A and B) and System 2 (Sensor A and B) were extracted. To ensure consistency in the analysis, the start time of each waveform was aligned to 0. This alignment was performed to retain only the curve pattern information for the sensor data. Subsequently, we applied each of the previously mentioned clustering algorithms. All clustering visualizations and code are available in the Siemens_data_clustering.ipynb file.

K-means Results:

Elbow plots were generated for each system's sensor, illustrating the sum of squared distances from 1 to 10 clusters. For both the calibration and sample windows of sensor A and sensor B in System 1, the optimal number of clusters was determined to be 3. Conversely, for System 2, the ideal number of clusters appeared more ambiguous, lying between 3 and 4. These plots effectively delineate the separation of each cluster.

Hierarchical Clustering Results:

Echoing the findings from K-means clustering, hierarchical clustering also yielded 3 clusters for the calibration and sample windows of sensor A and sensor B in System 1, and 3-4 clusters for System 2. The clustering plots demonstrate the distinctiveness of each cluster.

DTW Clustering Results:

Similarly to hierarchical clustering, DTW clustering produced 3 clusters for both System 1 and System 2. The clustering plots demonstrate the clear distinction of each cluster.

Since we have many plots for clustering results, we will show them during the meeting.

Appendices

Appendix 1: Plots before removing bad codes

First two components Sensor A Blood/Aqueous from CalWindow:

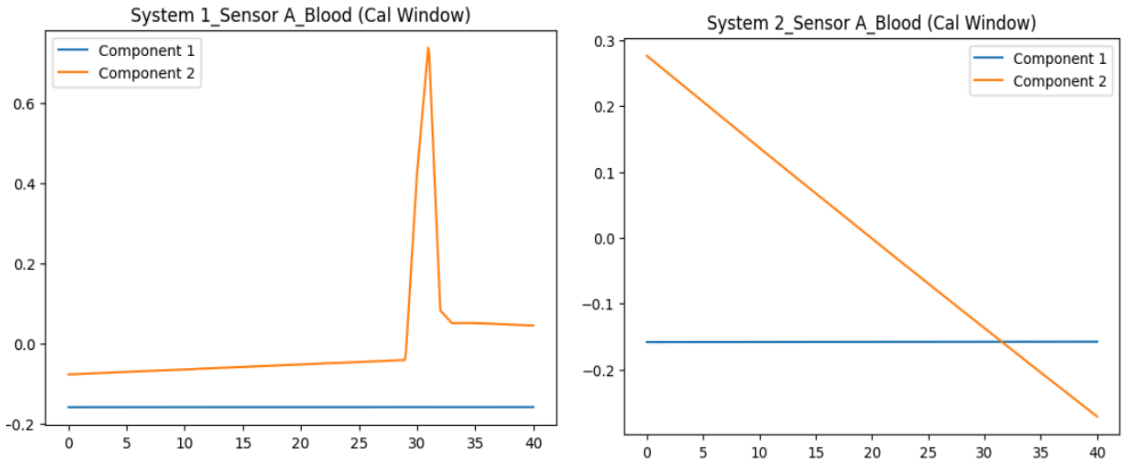

Figure 22. First two principal components for Sensor A Blood from the Cal Window.

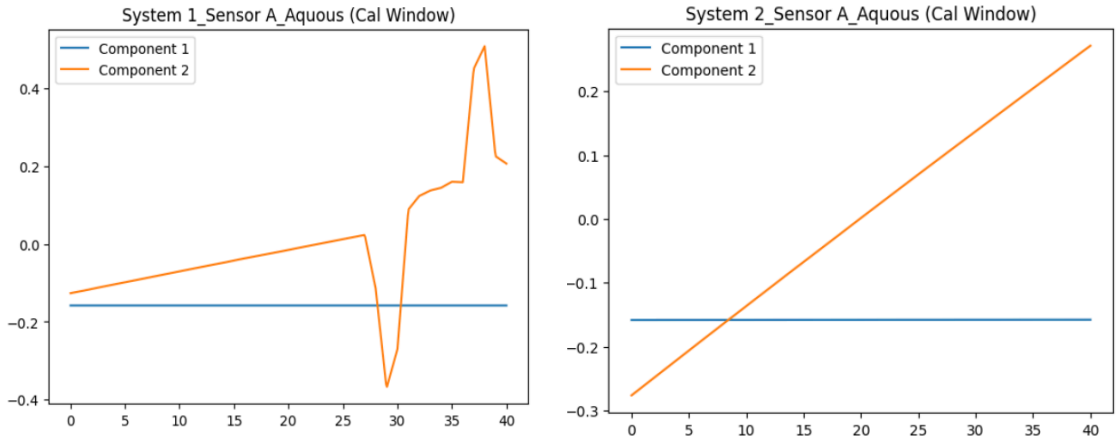

Figure 23. First two principal components for Sensor A Aqueous from the CalWindow.

Zoom in components in Blood CalWindow:

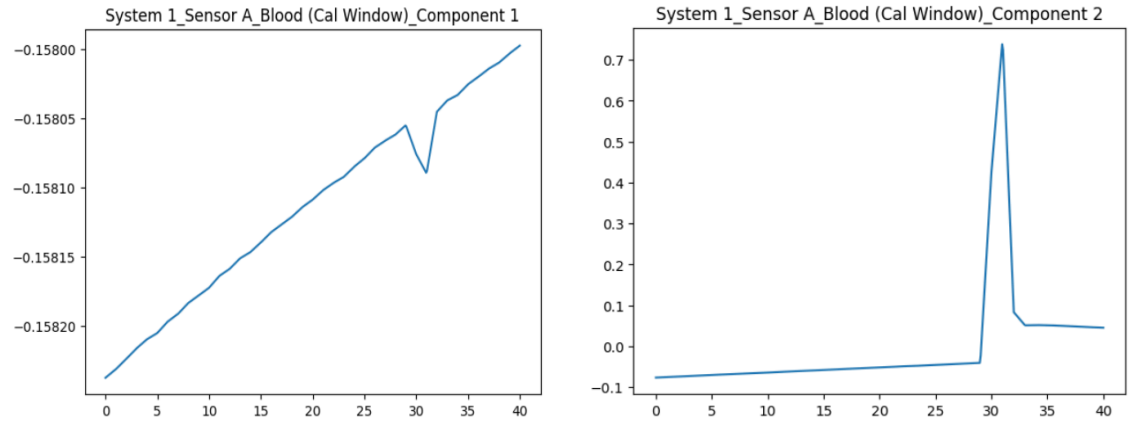

Figure 24. Zoom-in components in System 1 SensorA CalWindow Blood fluids.

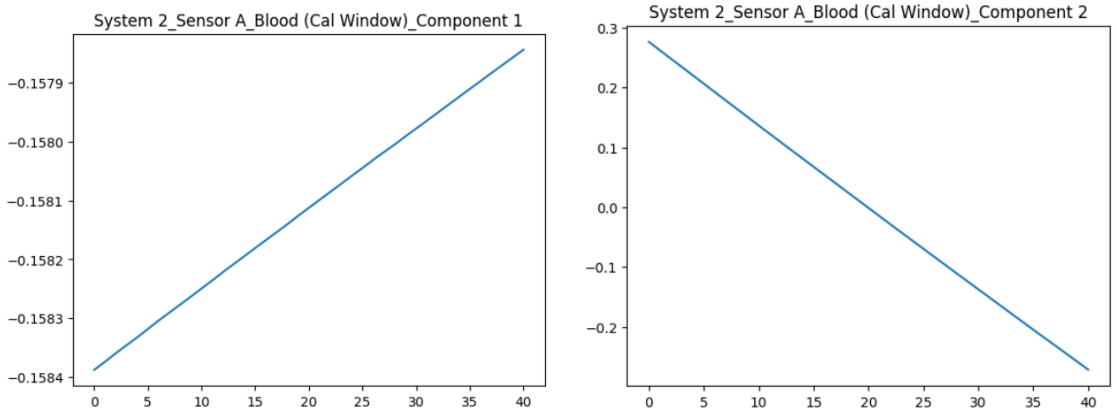

Figure 25. Zoom-in components in System 2 SensorA CalWindow Blood fluids.

Zoom in components in Aqueous CalWindow:

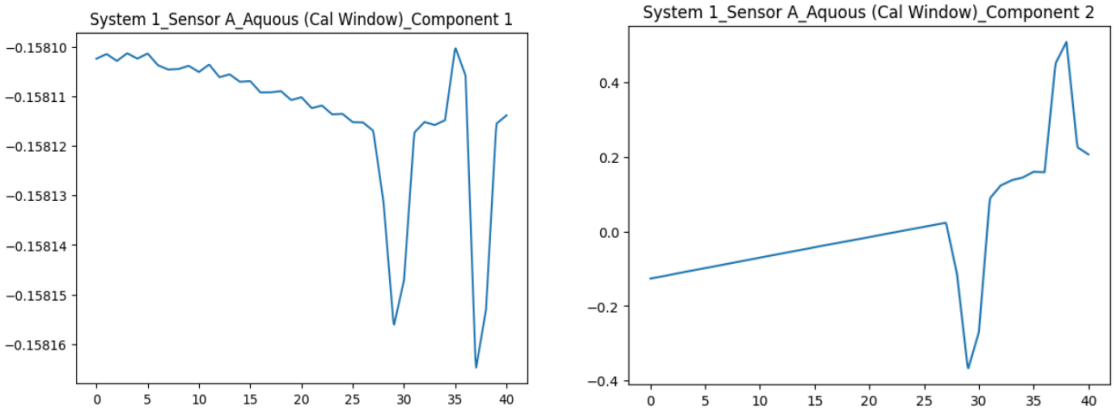

Figure 26. Zoom-in components in System 1 SensorA CalWindow Aqueous fluids.

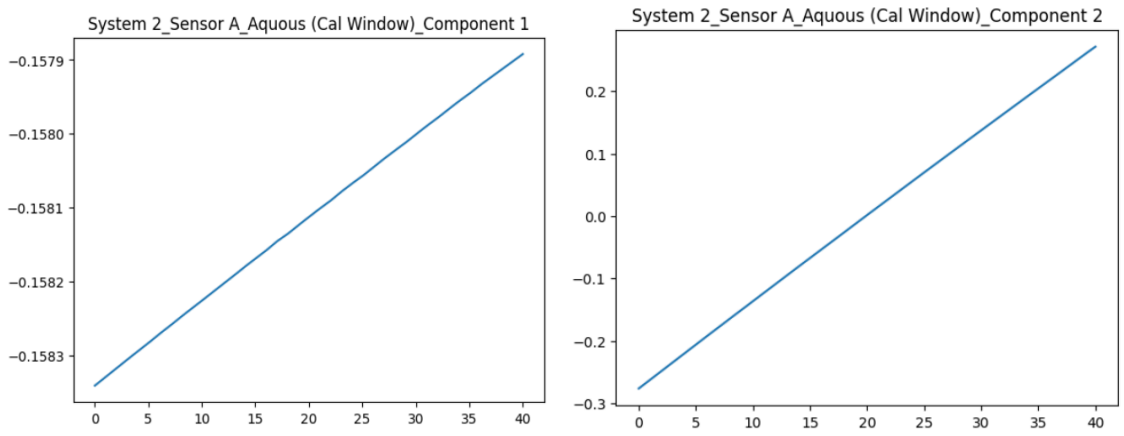

Figure 27. Zoom-in components System 2 SensorA CalWindow Aqueous fluids.

Zoom in components in general sample windows:

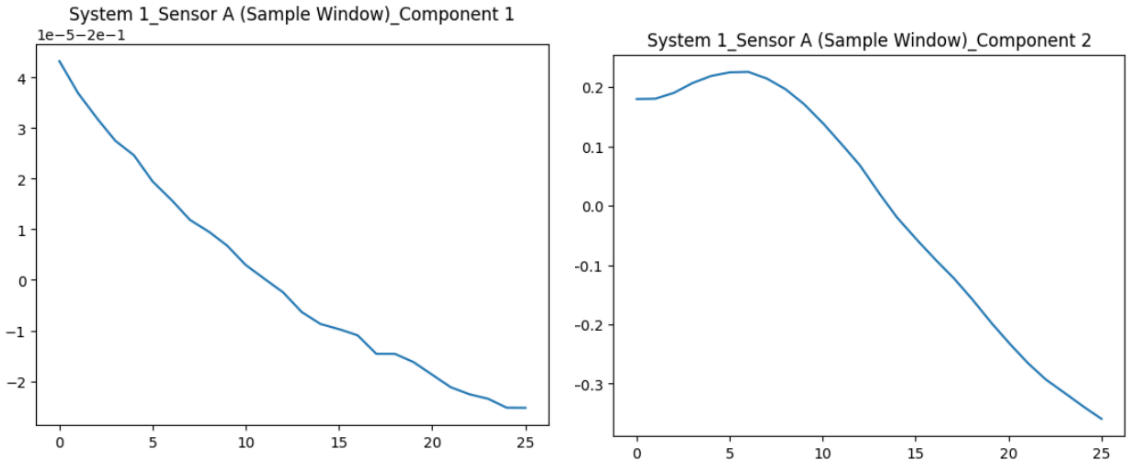

Figure 28. Zoom-in components in System 1 SensorA Sample Window.

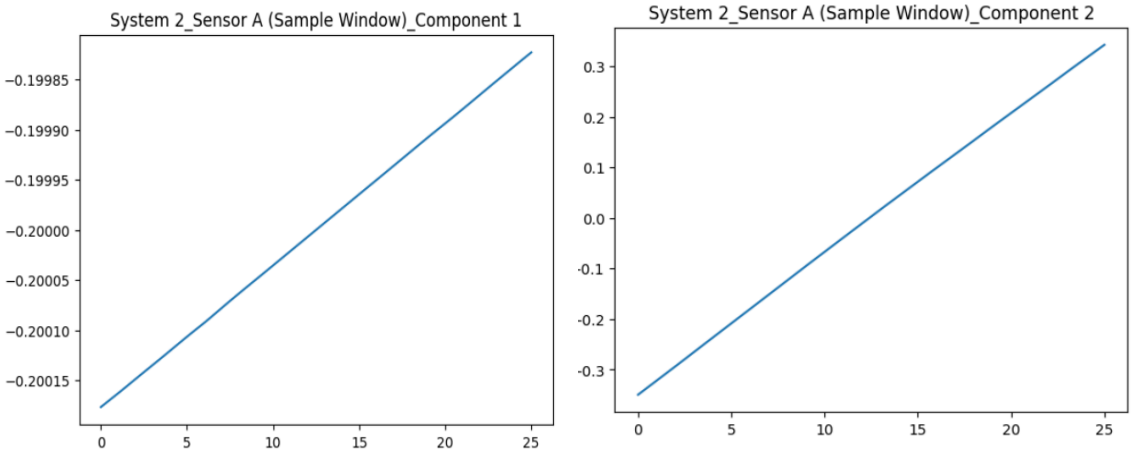

Figure 29. Zoom-in components in System 2 SensorA Sample Window.

Zoom in components in the sample windows of Blood:

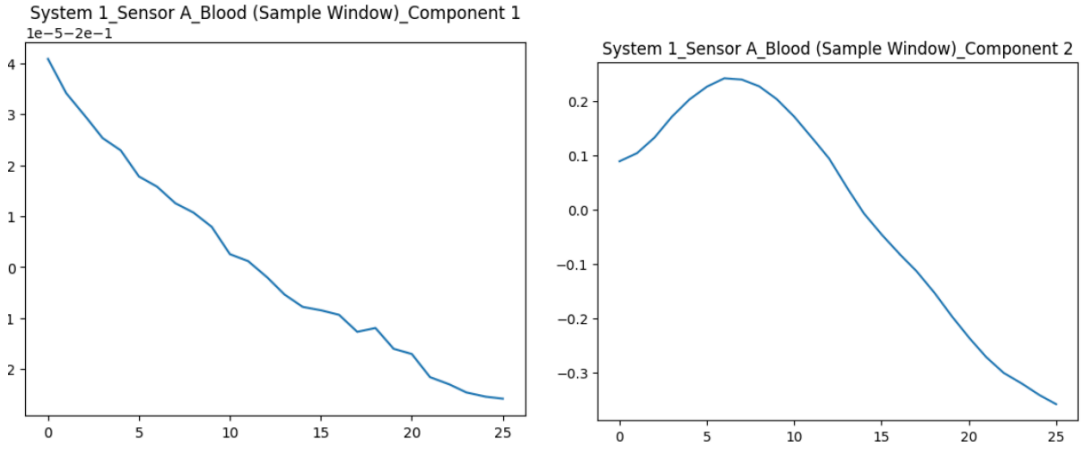

Figure 30. Zoom-in components in System 1 SensorA Sample Window Blood fluids.

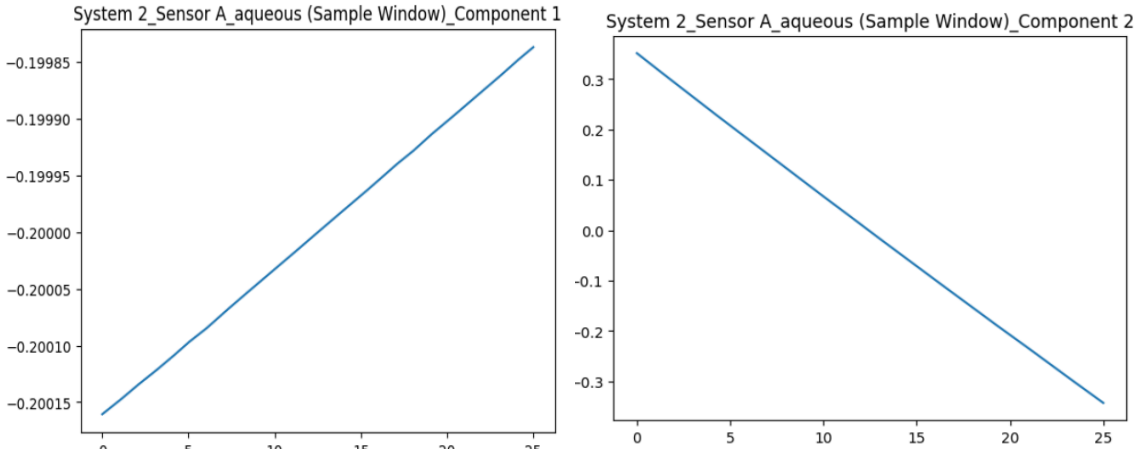

Figure 31. Zoom-in components in System 2 SensorA Sample Window Blood fluids.

Zoom in components in the sample windows of Aqueous:

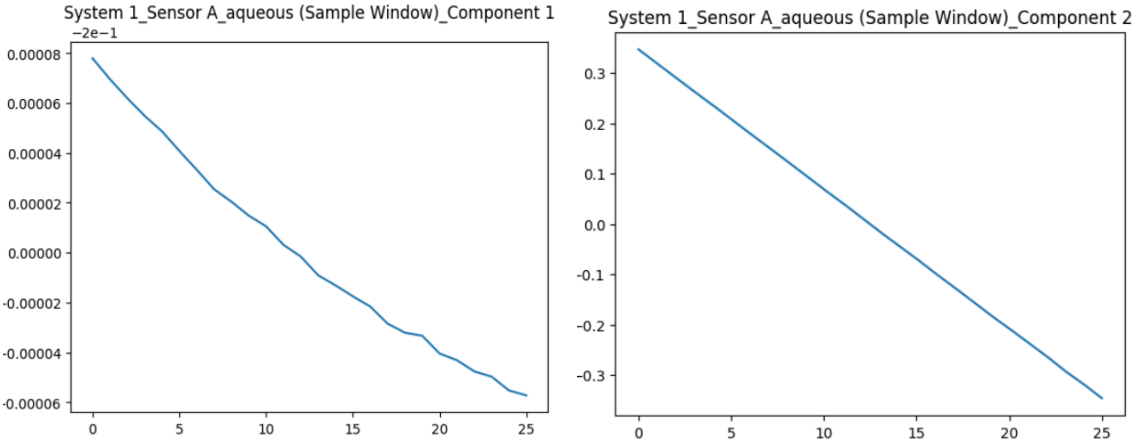

Figure 32. Zoom-in components in System 1 SensorA Sample Window Aqueous fluids.

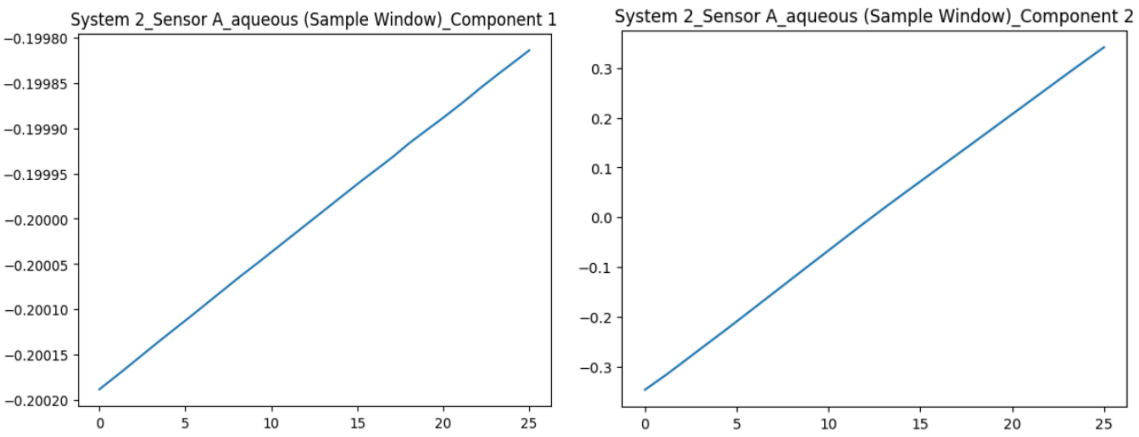

Figure 33. Zoom-in components in System 2 SensorA Sample Window Aqueous fluids.

References

[1] "Functional principal component analysis," Wikipedia, The Free Encyclopedia, https://en.wikipedia.org/wiki/Functional_principal_component_analysis (accessed May 19, 2024).

[2] "Functional Principal Component Analysis (FPCA)," scikit-fda documentation, https://fda.readthedocs.io/en/latest/modules/preprocessing/autosummary/skfda.preprocessing.dim_reduction.FPCA.html (accessed May 19, 2024).

[3] J.-L. Wang, J.-M. Chiou, and H.-G. Müller, "Functional Data Analysis," Annual Review of Statistics and Its Application, vol. 3, pp. 257-295, 2016, doi: https://doi.org/10.1146/annurev-statistics-041715-033624. [Online]. Available: https://www.annualreviews.org/content/journals/10.1146/annurev-statistics-041715-033624