ANOMALY DETECTION AND PREDICTIVE MAINTENANCE FOR INDUSTRIAL DEVICES

GROUP-18

Members:

- Davide Checchia - 2078232
- Jelin Raphael Akkara - 2072064
- Kiamehr Javid - 2084294

OUTLINE

I. SETTING UP THE VIRTUAL MACHINES

II. INTRODUCTION TO THE PROBLEM

III. TIME NORMALIZATION

IV. TASK 1: ANOMALY DETECTION

V. TASK 2: CORRELATIONS WITH TEMPERATURE

VI. TASK 3: ALARM CORRELATIONS AND PREDICTION

VII. BENCHMARKING

I. SETTING UP THE VIRTUAL MACHINES

1. INSTALLING LIBRARIES, ESTABLISHING CONNECTION
2. PREPARING DASK CLUSTERS
3. PREPARING CONNECTION TO BUCKETS

INSTALLING LIBRARIES, ESTABLISHING CONNECTION

- Installing the Dask library is required, from github repository and pip install command
- SSH Tunneling to connect ports 8888 from local machines to 8888 in VM
- Connection to VSCode and/or browser based on the forwarding

PREPARING DASK CLUSTERS

- Prepare cluster with dask-ssh command for single use, SSHCluster in jupyter for bulk analysis

- Forward port 8787 for Dashboard monitoring and data collection

PREPARING DASK CLUSTERS

```
kjavid@mapd-b-2023-gr18-1:~$ lscpu
Architecture:                    x86_64
  CPU op-mode(s):                32-bit, 64-bit
  Address sizes:                 46 bits physical, 48 bits virtual
  Byte Order:                    Little Endian
CPU(s):                          4
  On-line CPU(s) list:           0-3
Vendor ID:                       GenuineIntel
  Model name:                    Intel(R) Xeon(R) CPU E5-2670 v2 @ 2.50GHz
    CPU family:                  6
    Model:                       62
    Thread(s) per core:          1
    Core(s) per socket:          1
    Socket(s):                   4
    Stepping:                    4
    BogoMIPS:                    4999.96
```

PREPARING CONNECTION TO BUCKETS

- Install boto3 library, set up connection with credentials
- Create bucket, put parquet files in a folder, CSV outside of the folder
- Code snippet for data downloading

II. INTRODUCTION TO THE PROBLEM

1. OVERVIEW OF TASKS
2. THE DATASET

OVERVIEW OF TASKS

- **Main Objective:** Implement Distributed Analysis with Large Dataset (~5GB)
- Preparing Data: Normalizing Time
- Task 1: Anomaly Detection, Correlations
- Task 2: Correlations
- Task 3: Correlations, Prediction

THE DATASET

- The data is collected for four devices (refrigerators), each recorded for around 4-6 months, spanning a period from 1 October, 2020 to 31 March, 2021.

- Each device is measured with 132 metrics, some measured more frequently than others. The data types of the metrics vary: Integers, decimals and alarms.

- Alarms are recorded as 16 bit integers – the 16 bits corresponding to 16 different sensors.

id	when	hwid	metric	value
1	1601510485159	SW-065	SA4	0.0
2	1601510485159	SW-065	SA3	0.0
3	1601510485159	SW-065	SA2	0.0
4	1601510485159	SW-065	S34	0.0
5	1601510485159	SW-065	S33	1.0

III. TIME NORMALIZATION

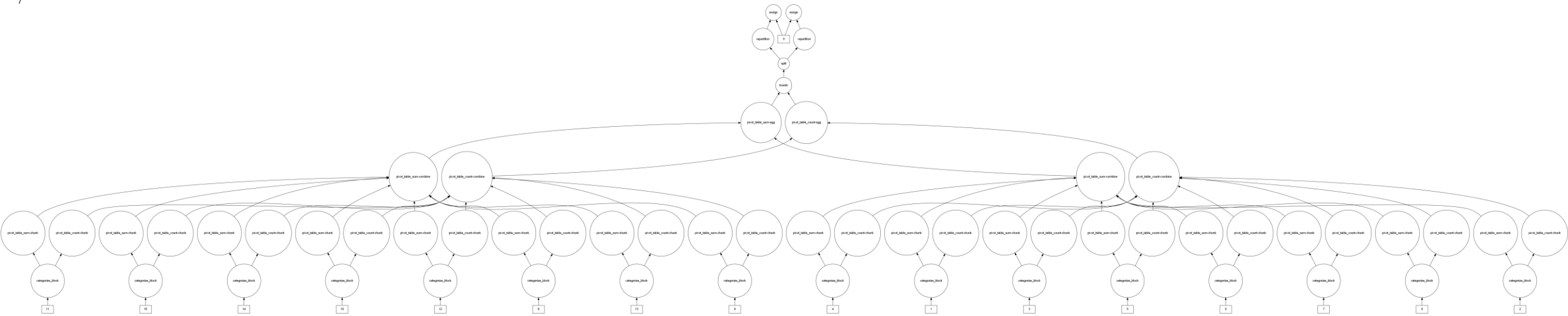
1. OBTAINING TIME INTERVAL
2. IMPLEMENTING MAP-REDUCE
3. GROUPING AND AGGREGATION

OBTAINING TIME INTERVAL

- **Main Objective:** Find optimal time interval and group dataframe in equally spaced intervals.
- **Approach:** For a given time interval, we check the percentage of occurrence of each metric per bin. If most metrics occur with a desired average occurrence (say 97%), we accept the time interval size.
- **Implementation:** In order to parallelize the code, we use the map-reduce framework.
  - Mapping Phase: For each partition, we define a pandas dataframe that records the presence of each metric in each bin.
  - Reduce Phase: Going through pairs of partition results, we combine the two dataframes into one by applying the logical OR function. At the end of this phase, we obtain a final dataframe with updated occurrence values for each bin.
- **Optimal Value(s):** 200 seconds for 97% occurrence, 600 seconds for 98% occurrence

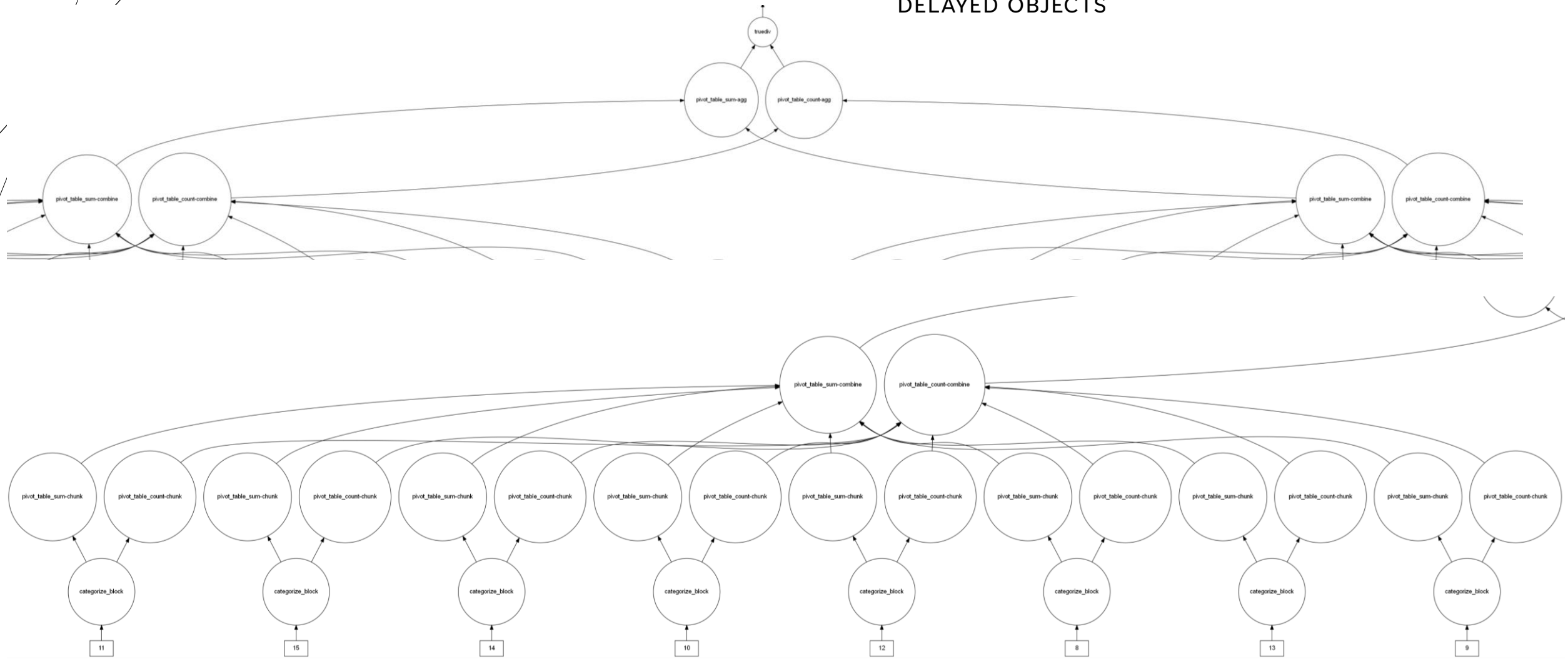
IMPLEMENTING MAP-REDUCE

DASK DATAFRAME → DELAYED OBJECTS → DASK BAG WITH DELAYED OBJECTS → FOLD



IMPLEMENTING MAP-REDUCE

DASK DATAFRAME → DELAYED OBJECTS → DASK BAG WITH DELAYED OBJECTS → FOLD



GROUPING AND AGGREGATION

- Once we have the optimal time interval size, we move on to group the metrics so that the dataframe has equally spaced readings.

- We define custom aggregation functions for different datatypes:
  - For integers: We take the maximum
  - For decimals: We take the mean
  - For alarms: We convert the integer value to its corresponding binary bits, and consider the logical OR of its 6^{th}, 7^{th}, 8^{th} bits.

III. TASK 1: ANOMALY DETECTION

1. APPROACH USED
2. IMPLEMENTATION
3. RESULTS

APPROACH USED

- **Main Objective:** Detect anomalies occurring in either of four engines, and find correlations to other metrics
- **Approach:** An anomaly is a period of high frequency fluctuations / flips in the binary states of either of the four engines in a device (S117, S118, S169, S170).
  - **Detecting a flip:**
    - All four engines are considered as one system.
    - Assign an integer value to the combined binary states of the engines.
    - A change in the integer value signifies a flip in either of the engines' states.
  - **Detecting an anomaly:**
    - Dynamic Window: Use a dynamic window to capture closely occurring flips.
    - Anomaly Threshold: Define a threshold to classify n number of flips as an anomaly event.

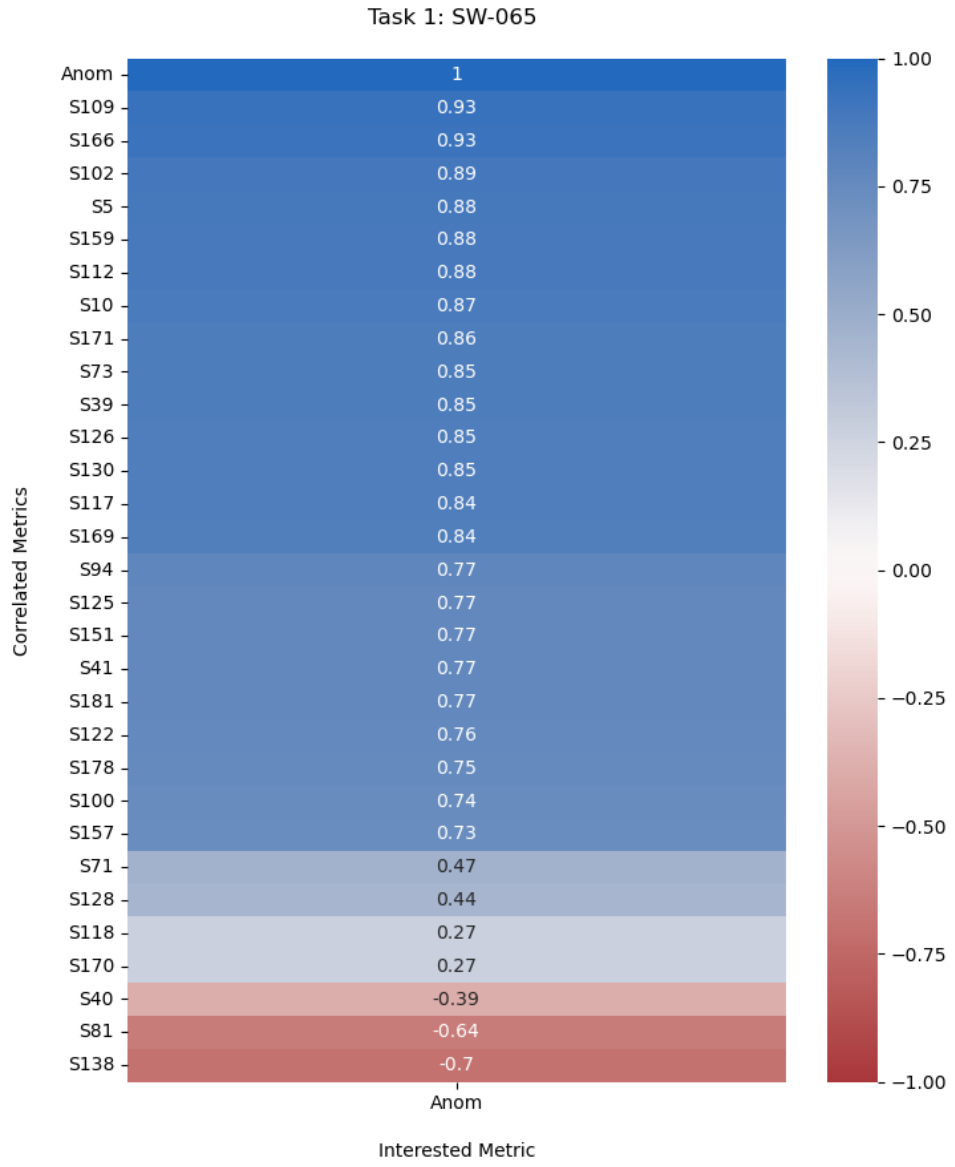
IMPLEMENTATION

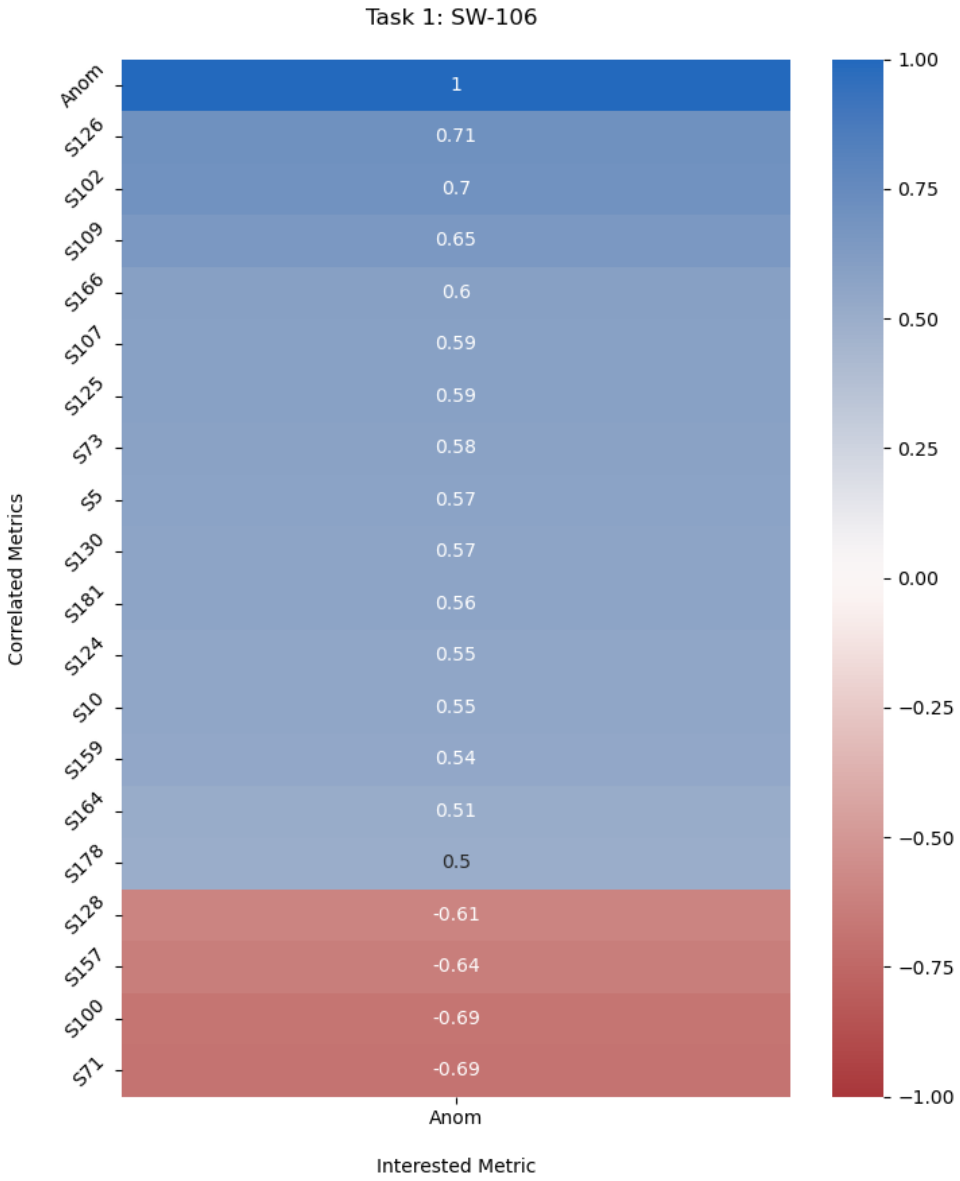
- This was implemented using a map-reduce framework:
  - **Mapping Phase:** For each partition, using the dynamic window and anomaly threshold parameters, anomalies are classified and the time periods noted.
  - **Reduce Phase:** Combines a pair of dataframes and returns a cleaned version. Cleaning includes removing duplicate time ranges, time ranges that may fall inside others and removing overlaps.

- For correlations, we compute the resulting dataframe to pandas and calculate the linear and non-linear correlations (Pearson, Spearman, Kendall correlations).
  - It is possible to find correlations in Dask, but it is only possible to find the linear correlations. Also, pivoting is costly in here.

RESULTS

Correlation: Pearson, **Threshold:** 20, **Window Size:** 1000 minutes





S109: Discharge Temperature, C1,1

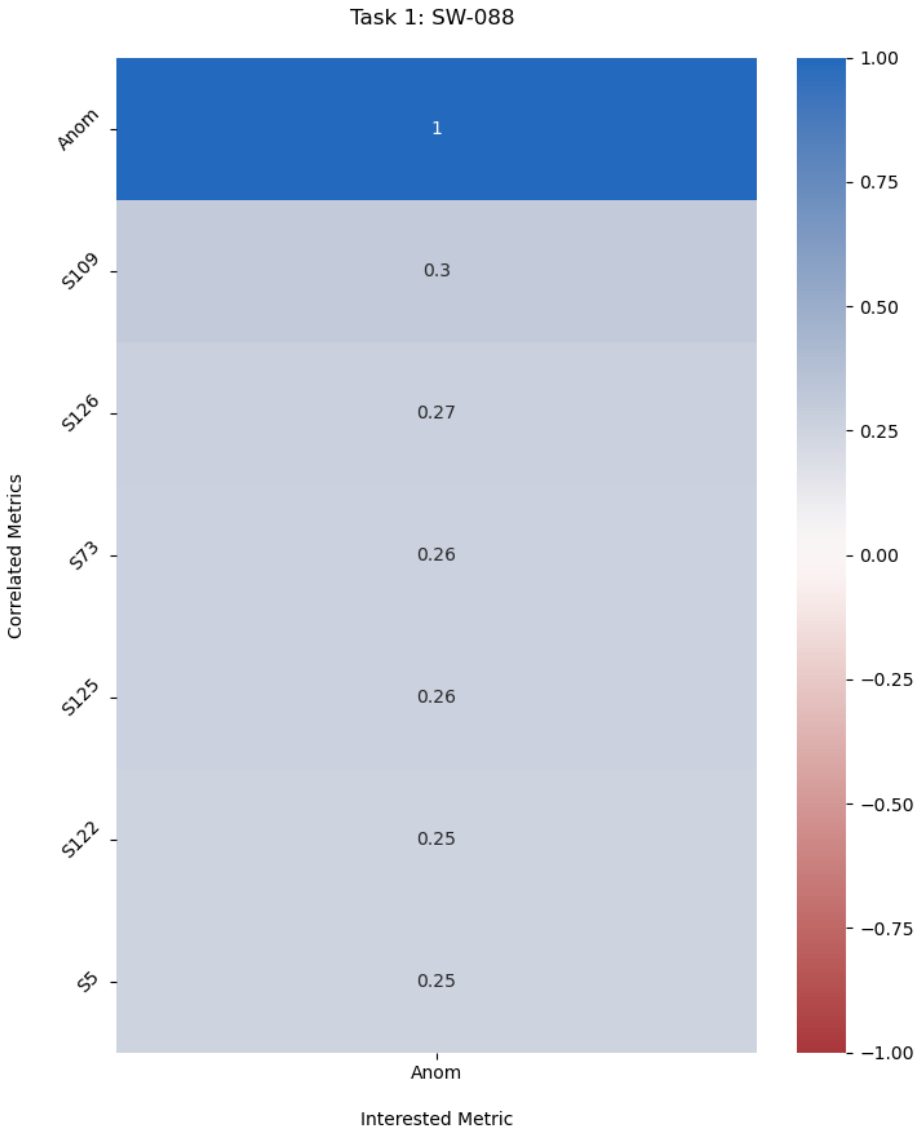
S166: Discharge Temperature, C1,2

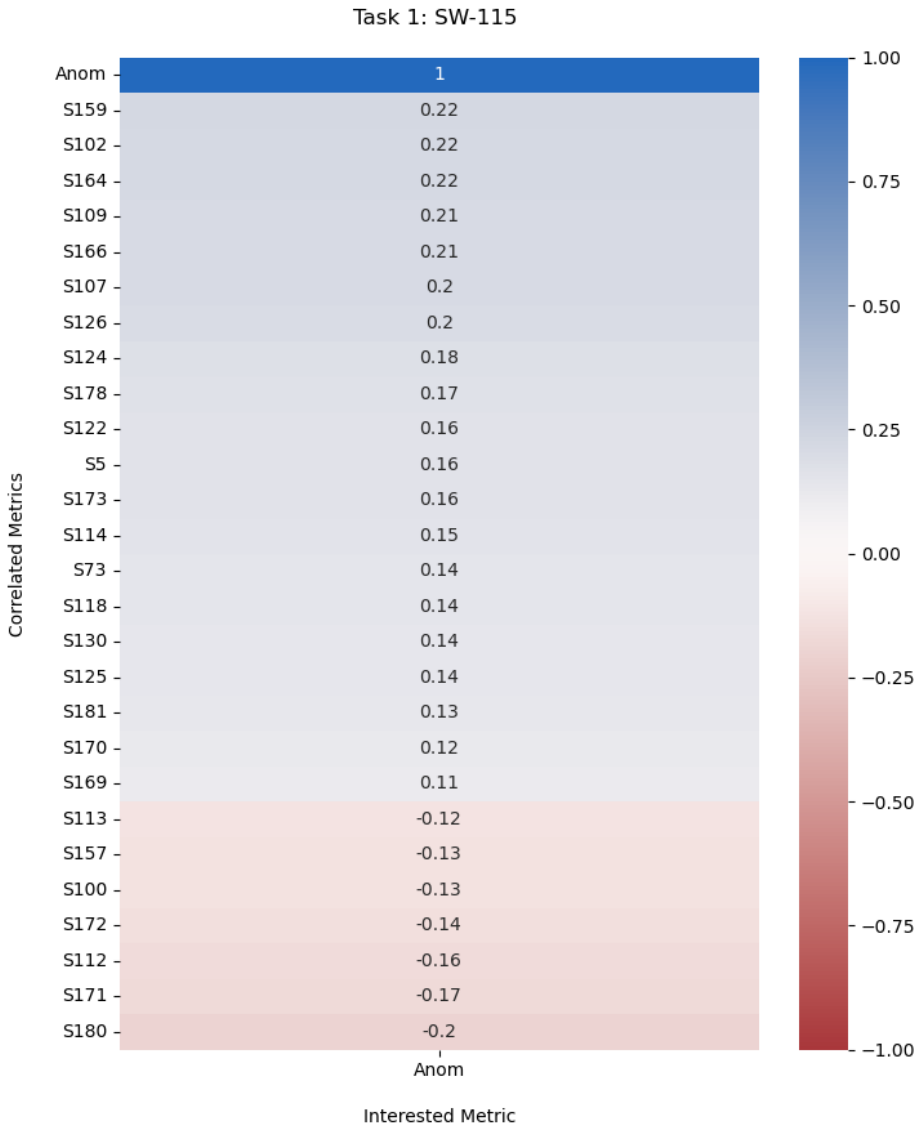
S102: Discharge Pressure Circ 1

S126: Pressure Ratio Circ 1

RESULTS

Correlation: Pearson, **Threshold:** 20, **Window Size:** 1000 minutes





S109: Discharge Temperature, C1,1

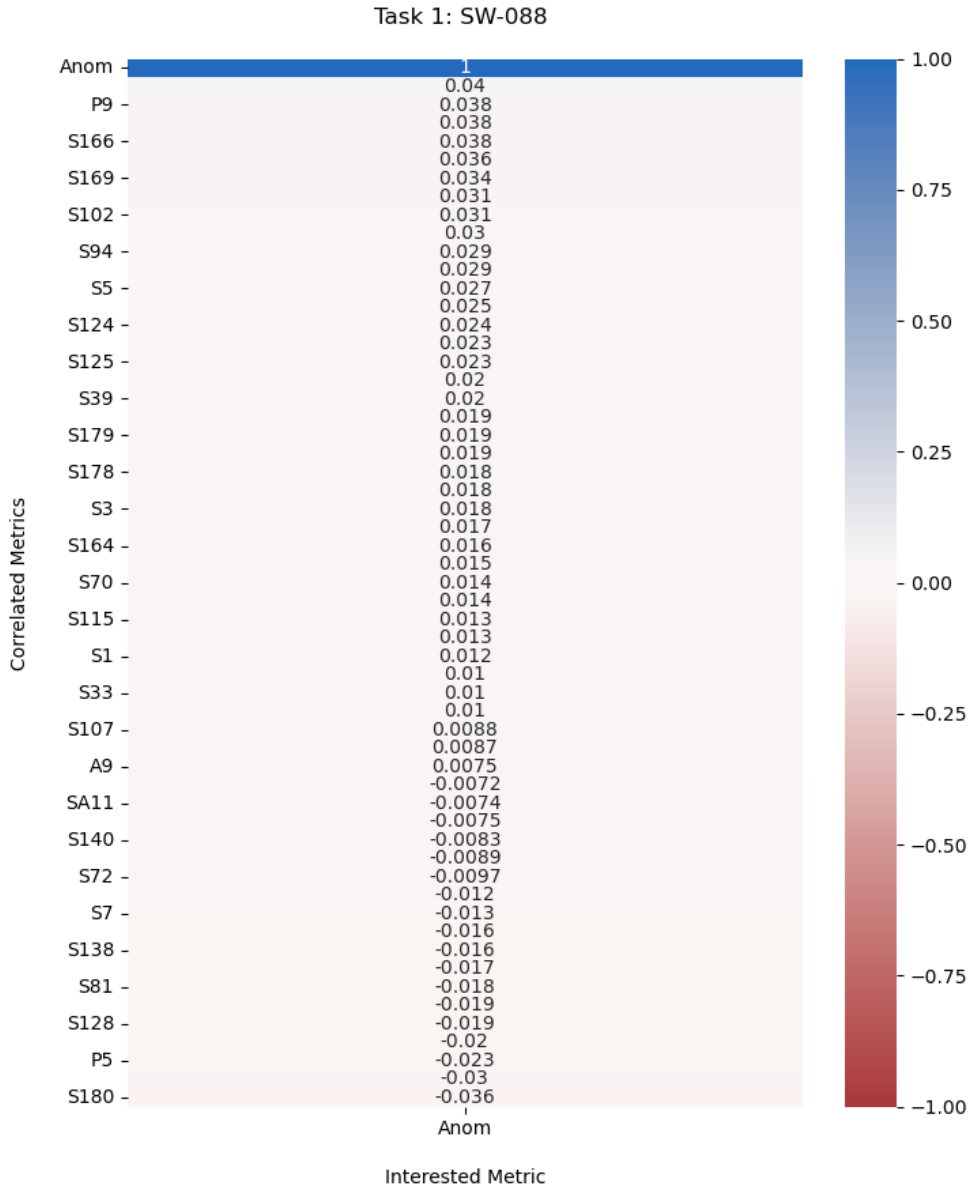
S159: Discharge Pressure Circ 2

S102: Discharge Pressure Circ 1

S126: Pressure Ratio Circ 1

RESULTS: Whole

Correlation: Pearson, **Threshold:** 20, **Window Size:** 1000 minutes



IV. TASK 2: CORRELATIONS WITH TEMPERATURE

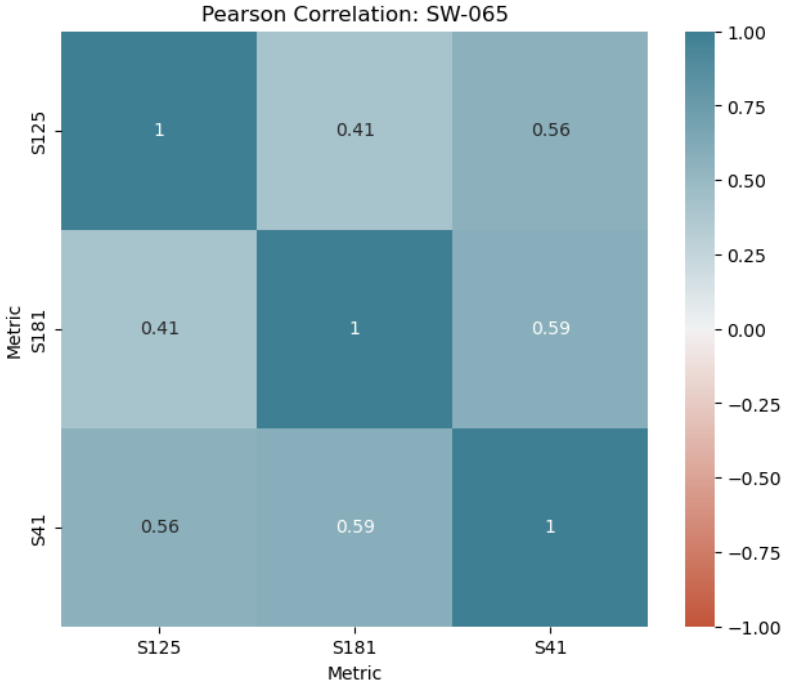
1. OBJECTIVE AND APPROACH
2. RESULTS

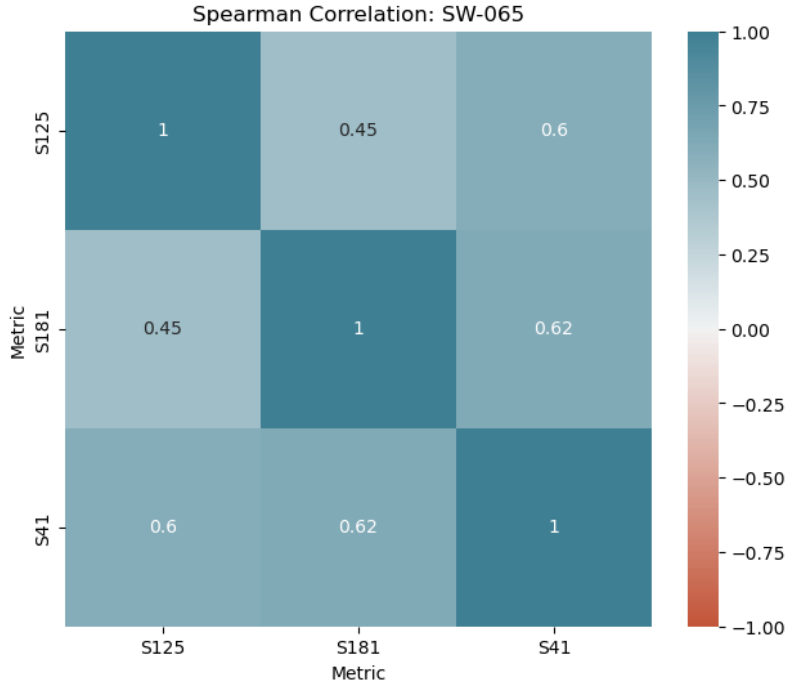
OBJECTIVE AND APPROACH

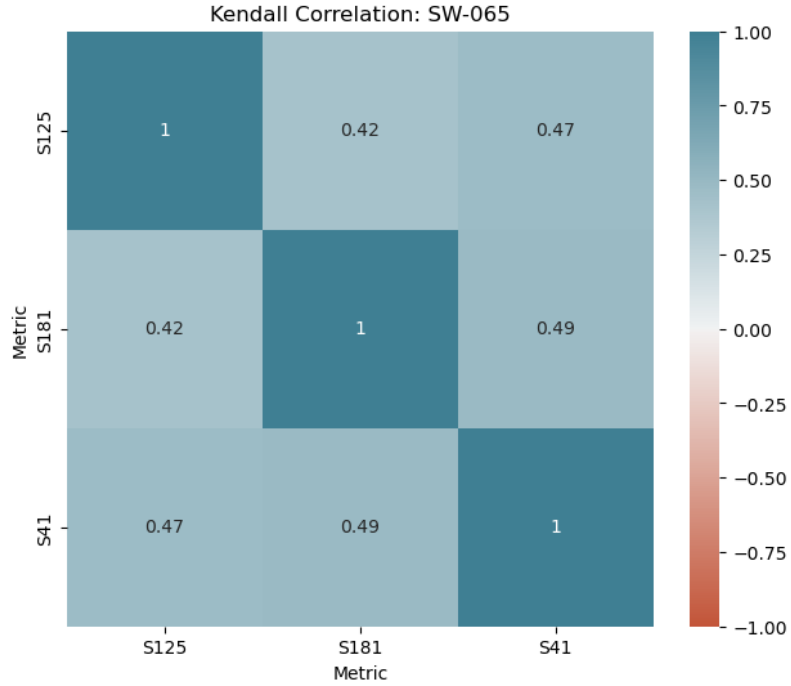
- **Main Objective:** Calculate correlations of two metrics (S125, S118) , referring to percentage of device loading, against external temperature (S41)

- **Approach:** Using the grouped dataframe (Time Normalized), we filter to the metrics and calculate the correlations.

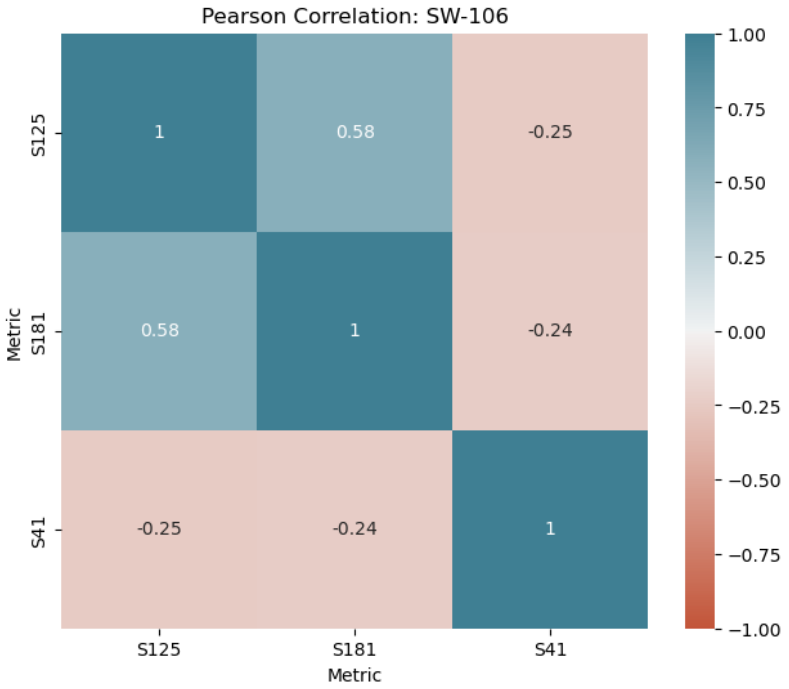
RESULTS: SW-065

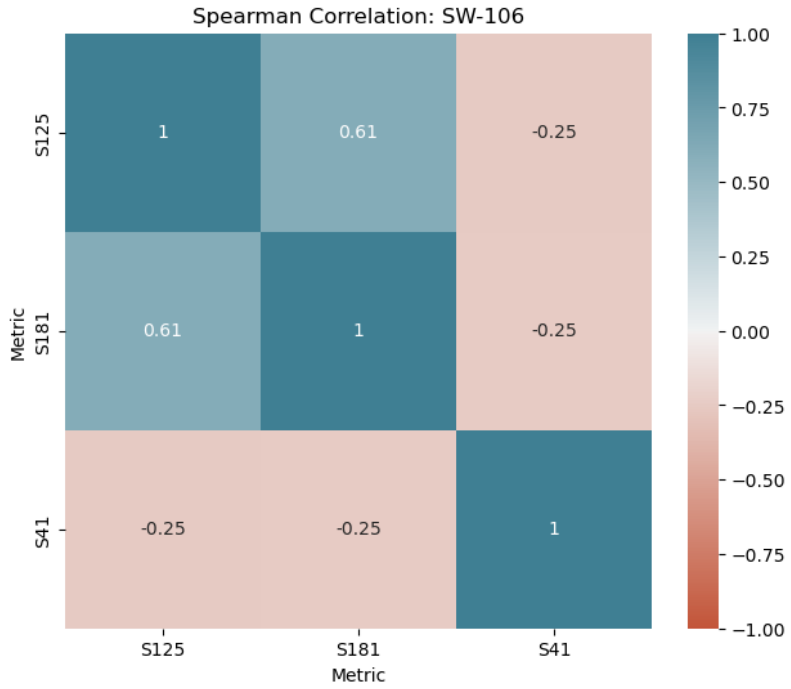


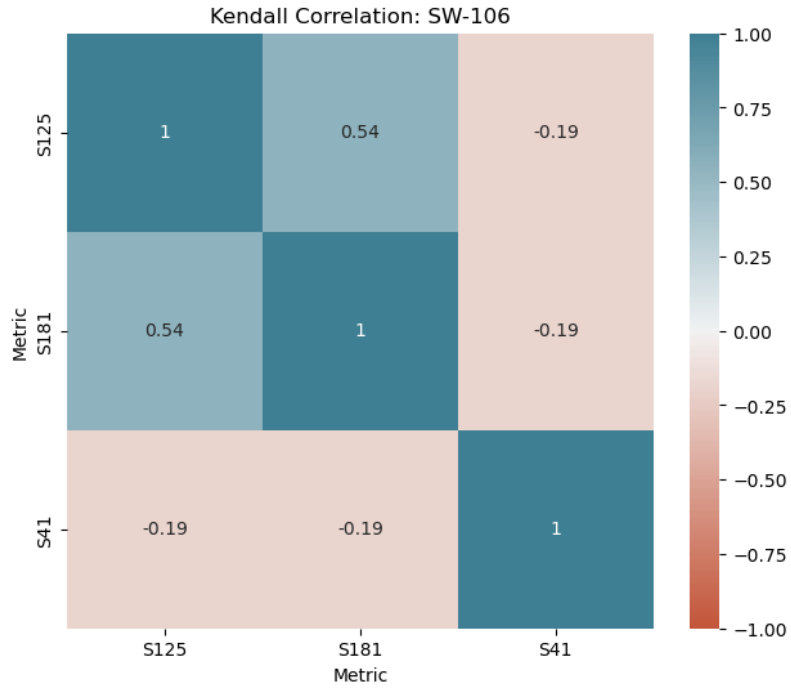




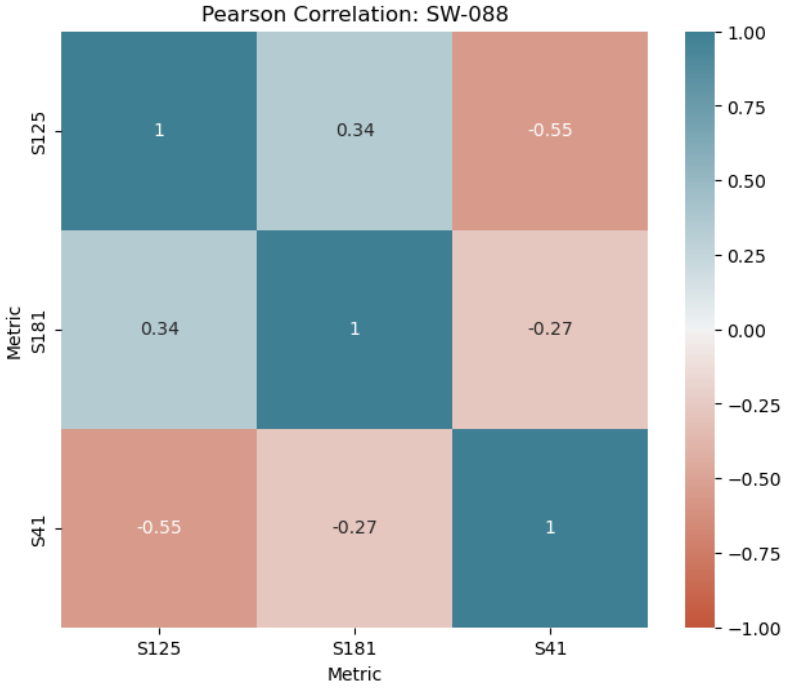
RESULTS: SW-106

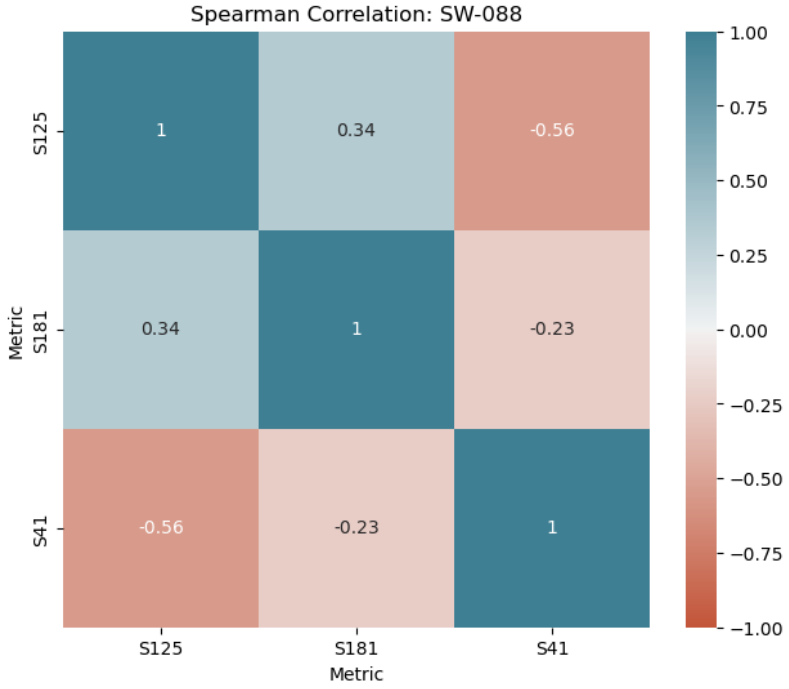


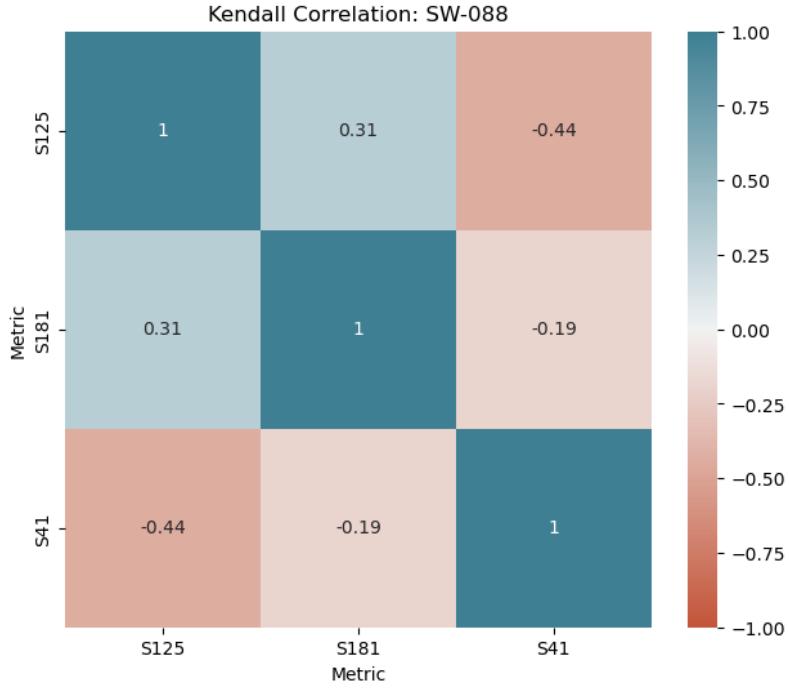




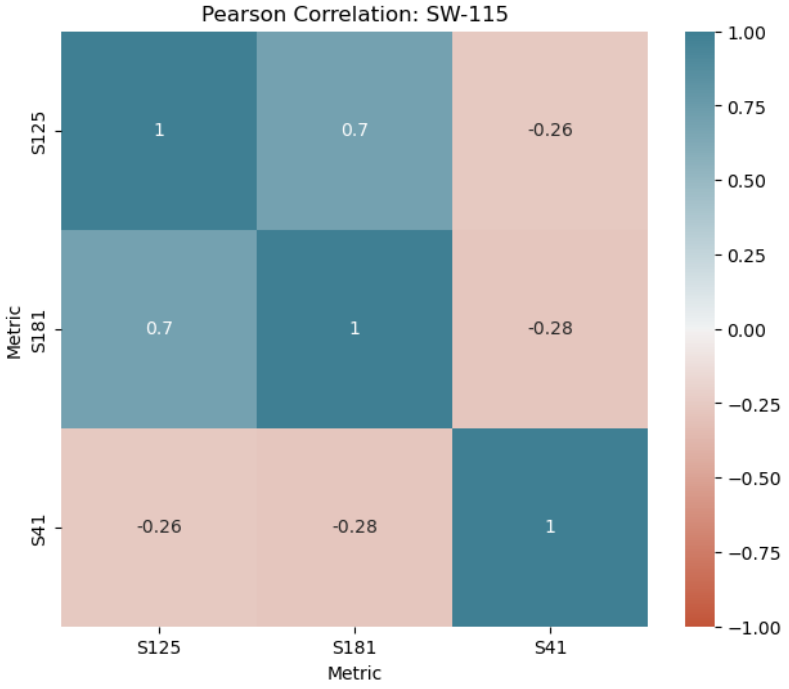
RESULTS: SW-088

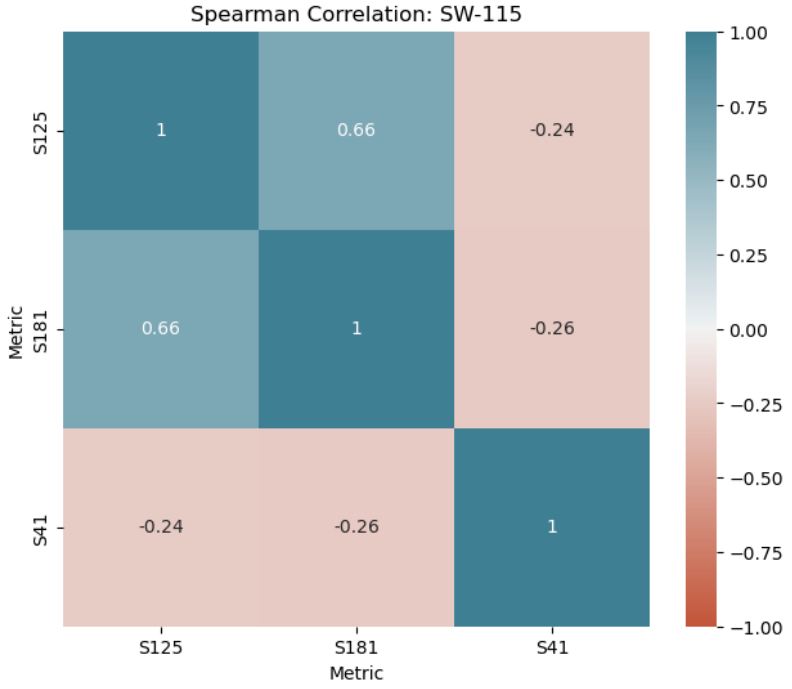


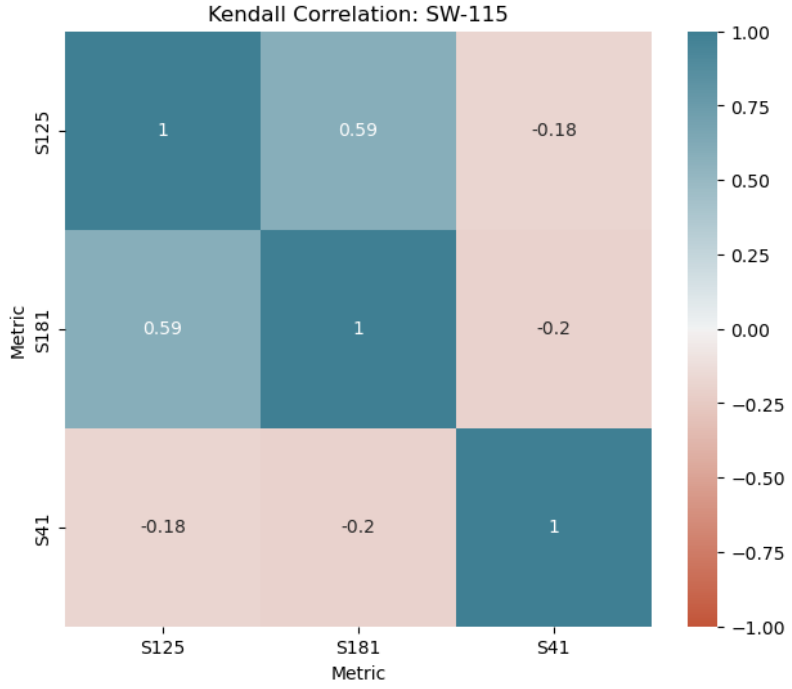




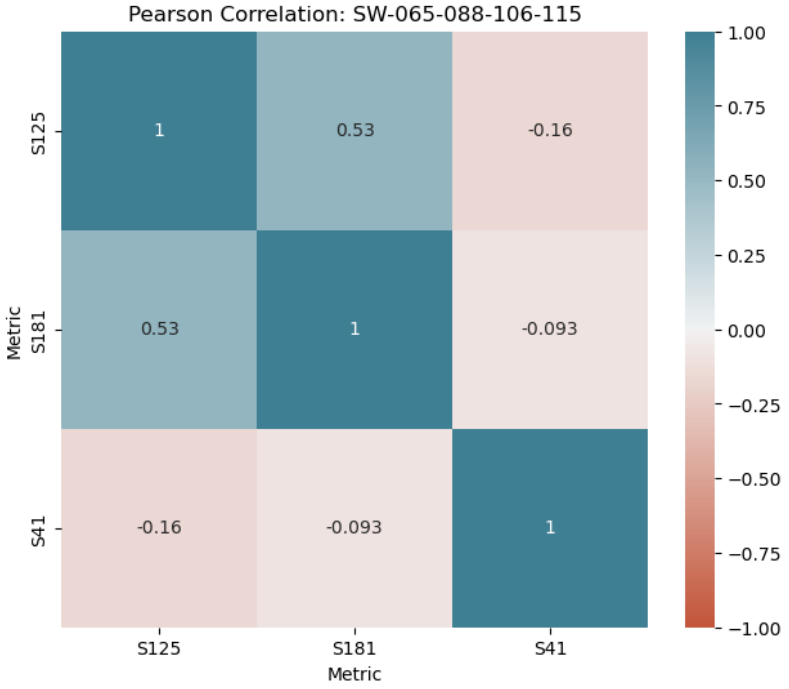
RESULTS: SW-115

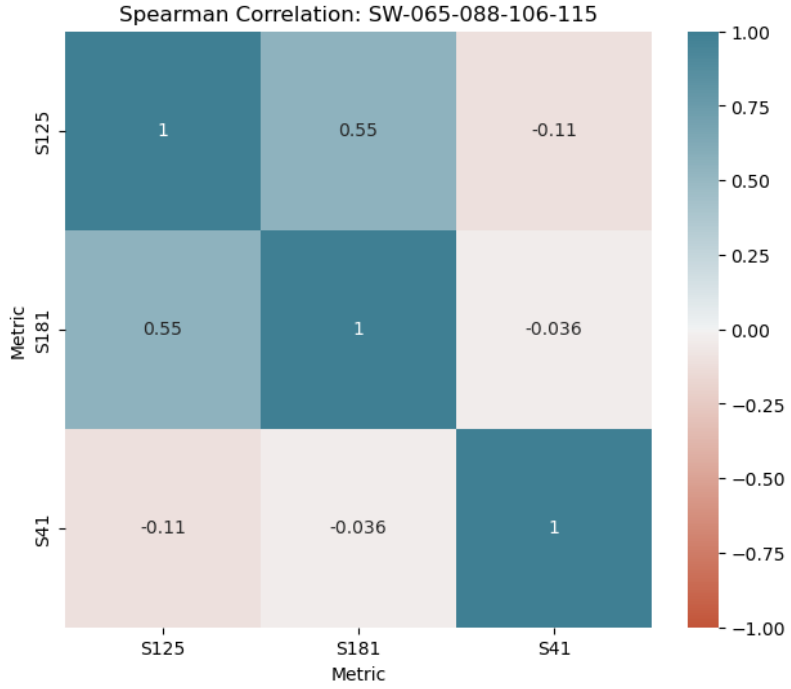


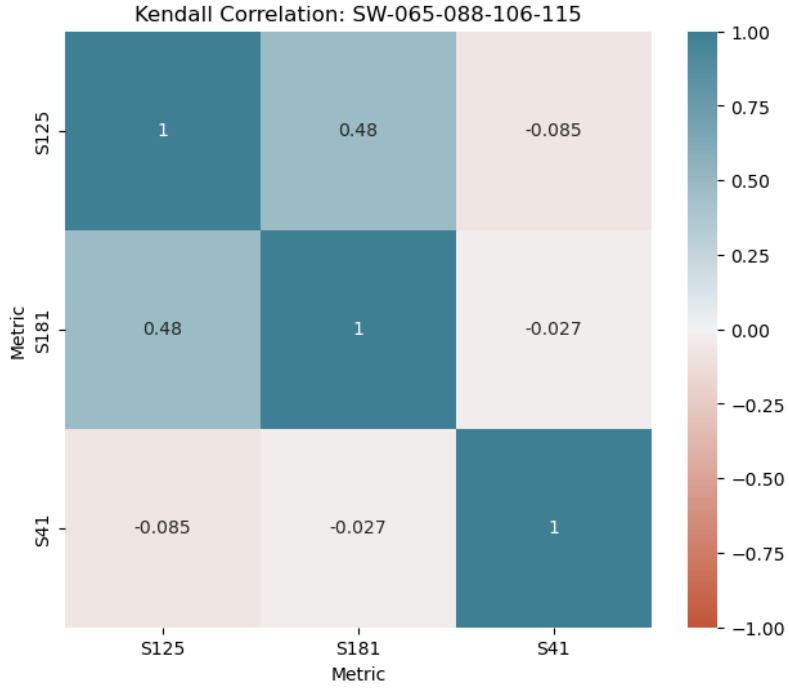




RESULTS: Whole Data







IV. TASK 3: ALARM CORRELATIONS AND PREDICTION

1. OBJECTIVE AND APPROACH
2. RESULTS

OBJECTIVE AND APPROACH

- **Main Objective:** Find correlations for alarms and develop an algorithm for predicting an alarm state based on previous values of metrics.

- **Approach:**
  - **Correlations:** Define a new column, which is the logical OR of the two alarm columns (A5, A9), and get the correlations with respect to the new column.

  - **Prediction:** We get the most correlated metrics, and assign weights and thresholds to each of them. The weight is a normalized value that is proportional to the corresponding correlation value. The threshold is obtained by taking the mean of the average values when the alarm is one and zero. This gives a sense of each metric's position depending on the alarm's value.

OBJECTIVE AND APPROACH

- **Prediction Example:**
  - Corr(param_1) = 0.7
  - Corr(param_2) = 0.68
  - Corr(param_3) = 0.62
  - Corr(param_4) = 0.6

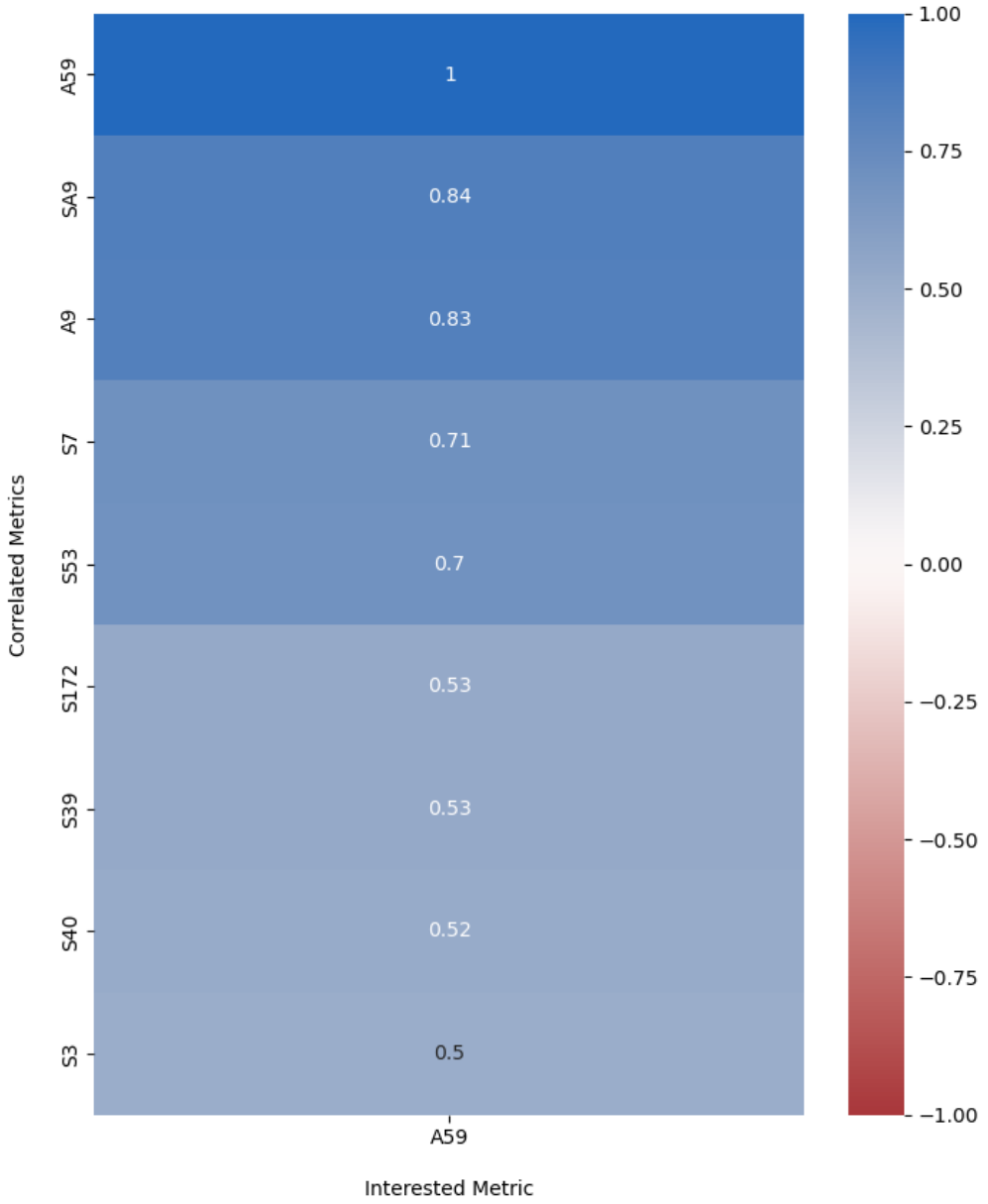
metric	corr	weights	threshold	new_vals	check	weighted_check
SA9	0.840873	0.194690	0.446240	1.0	True	0.194690
S7	0.707179	0.163735	0.638514	0.0	False	0.000000
S53	0.695330	0.160992	0.645270	1.0	True	0.160992
S172	0.527290	0.122085	0.670094	1.0	True	0.122085
S39	0.525779	0.121735	191.013921	200.0	True	0.121735
S40	0.521872	0.120830	190.094907	145.0	False	0.000000
S3	0.500721	0.115933	178.839948	NaN	False	0.000000

  - Sum(Corr(param)) = 2.6

  - **Probability =** $\sum_i \ Corr(param_i) \ * (\ param_i > Threshold_i) \ / \ 2.6$

RESULTS: SW-088

- The alarms are activated in only one device (SW-088). Out of a total of 1332 alarm measurements (in the whole data), there are only 161 instances where either of the alarms are active, all of which occur in a single device.
- All the highly correlated metrics correspond to other alarm metrics. However, there are weak correlations with metrics involving the water flow.
- Correlated Metrics:
  - **SA9, S7, S53:** Circuit 2 Alarm 1 (status), Blocked Alarm, Comulative Alarm
  - **S172:** Chiller Valve Circ 2
  - **S39, S40:** Utility Water IN, OUT

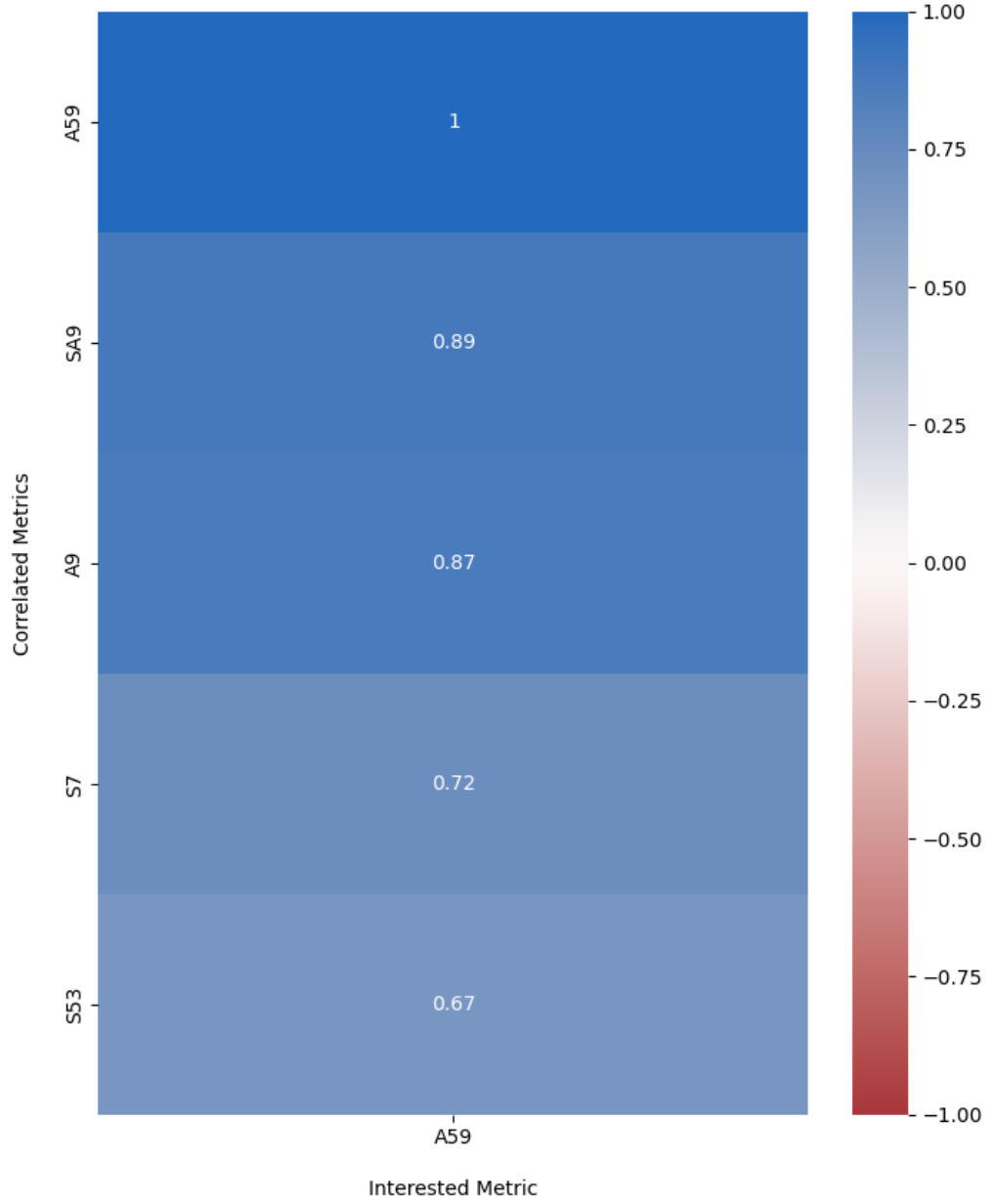


RESULTS: Whole Data

- In this case, there are no metrics, other than alarms, with greater than 0.5 correlations.

- The lack of correlations from relevant metrics proves, again, that the devices behave separately.

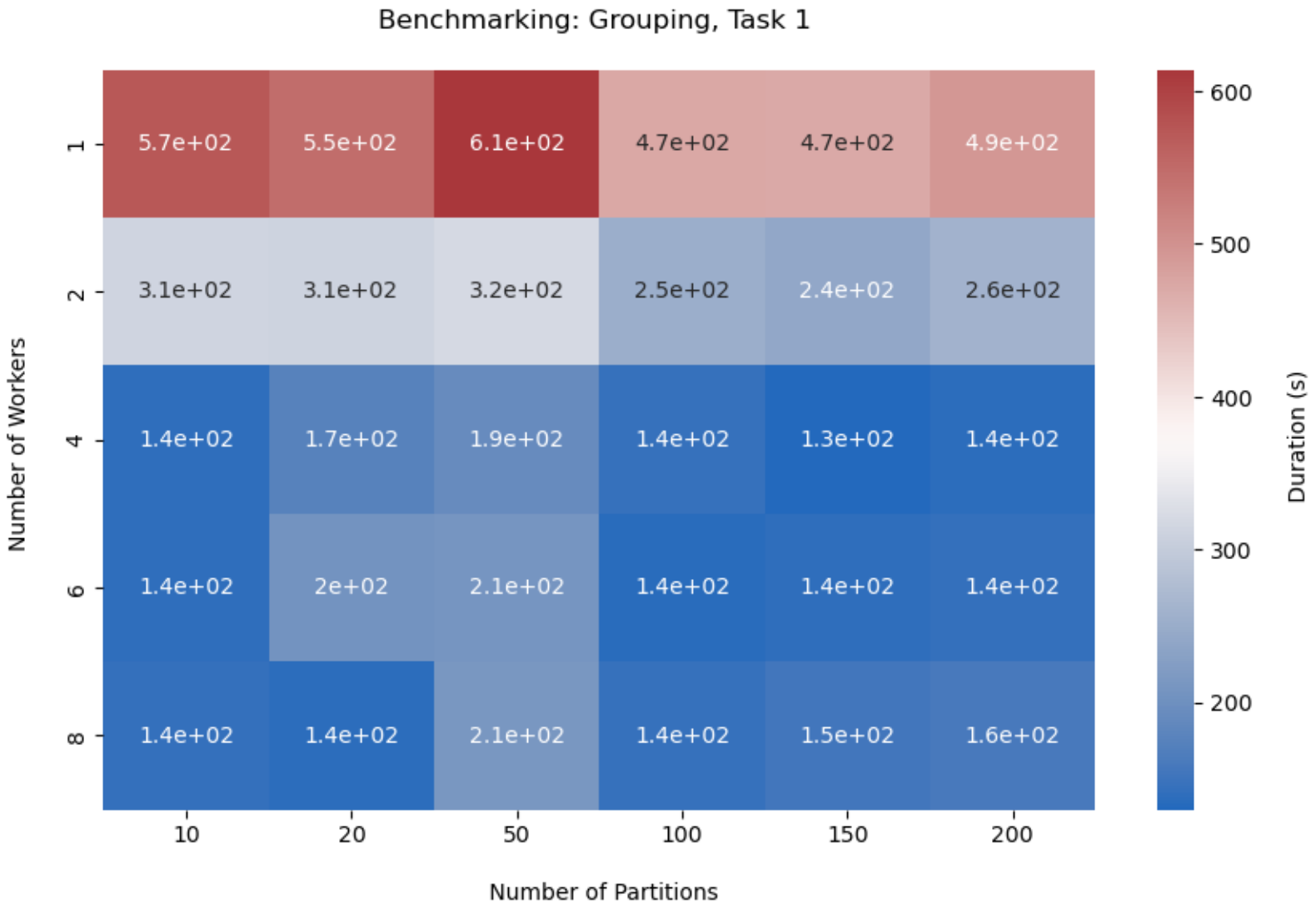
- Correlated Metrics:
  - **SA9:** Circuit 2 Alarm 1 (status)
  - **S7:** Blocked Alarm
  - **S53:** Comulative Alarm



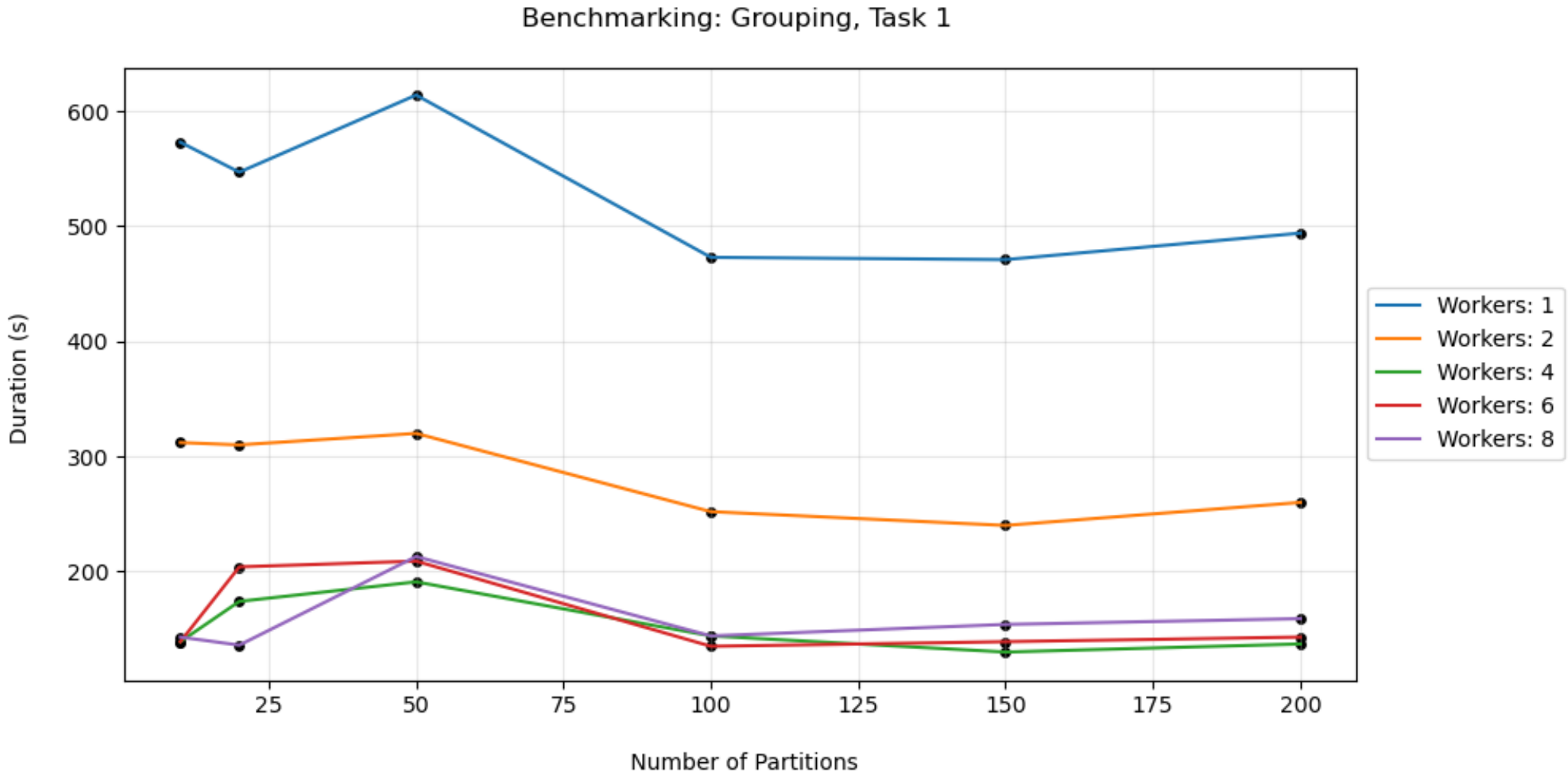
V. BENCHMARKING

1. WORKERS VS PARTITIONS
2. WORKERS VARYING
3. PARTITIONS VARYING
4. TASK STREAMS
5. WORKER BANDWIDTHS
6. CSV VS PARQUET

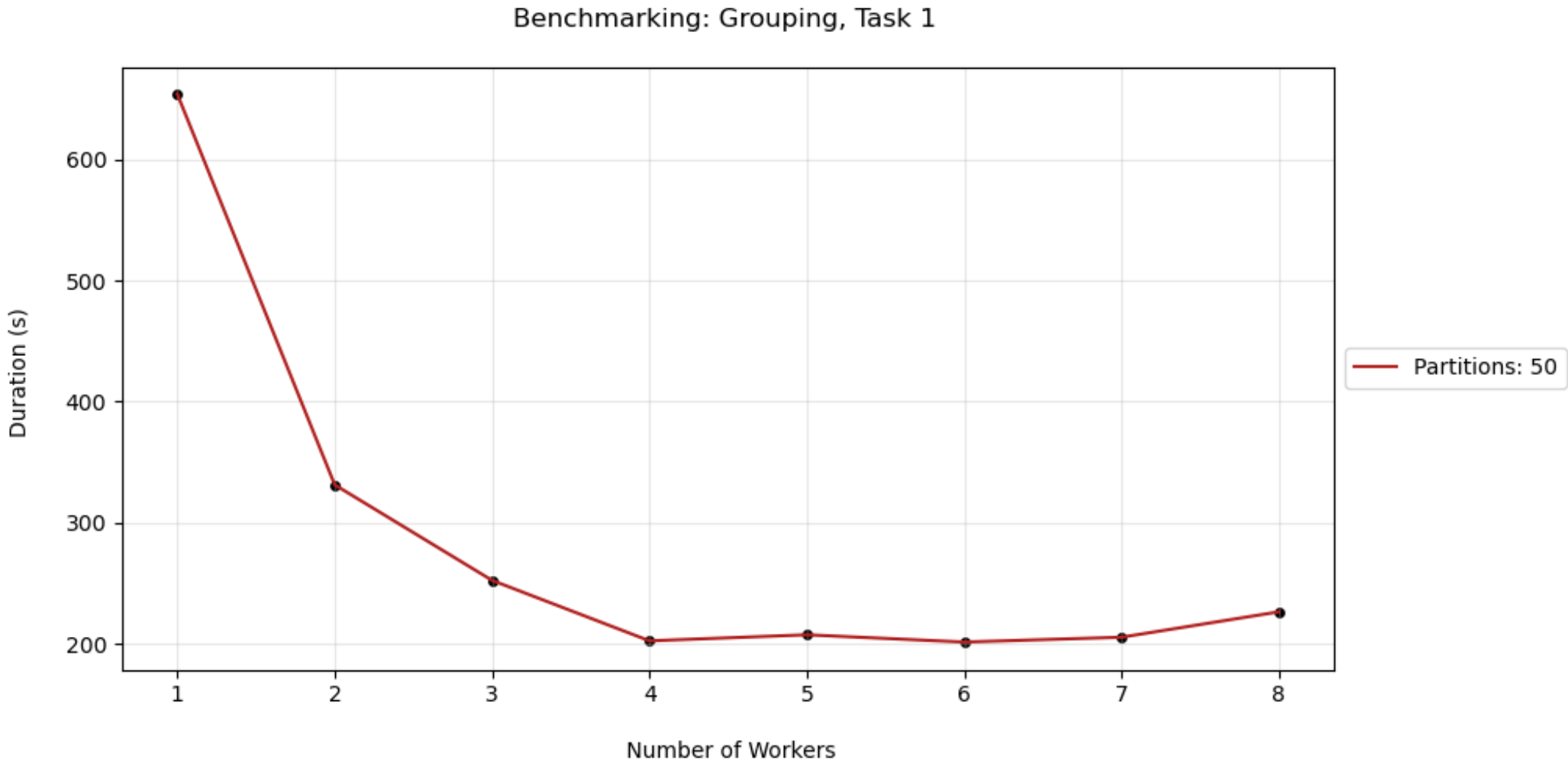
WORKERS VS PARTITIONS



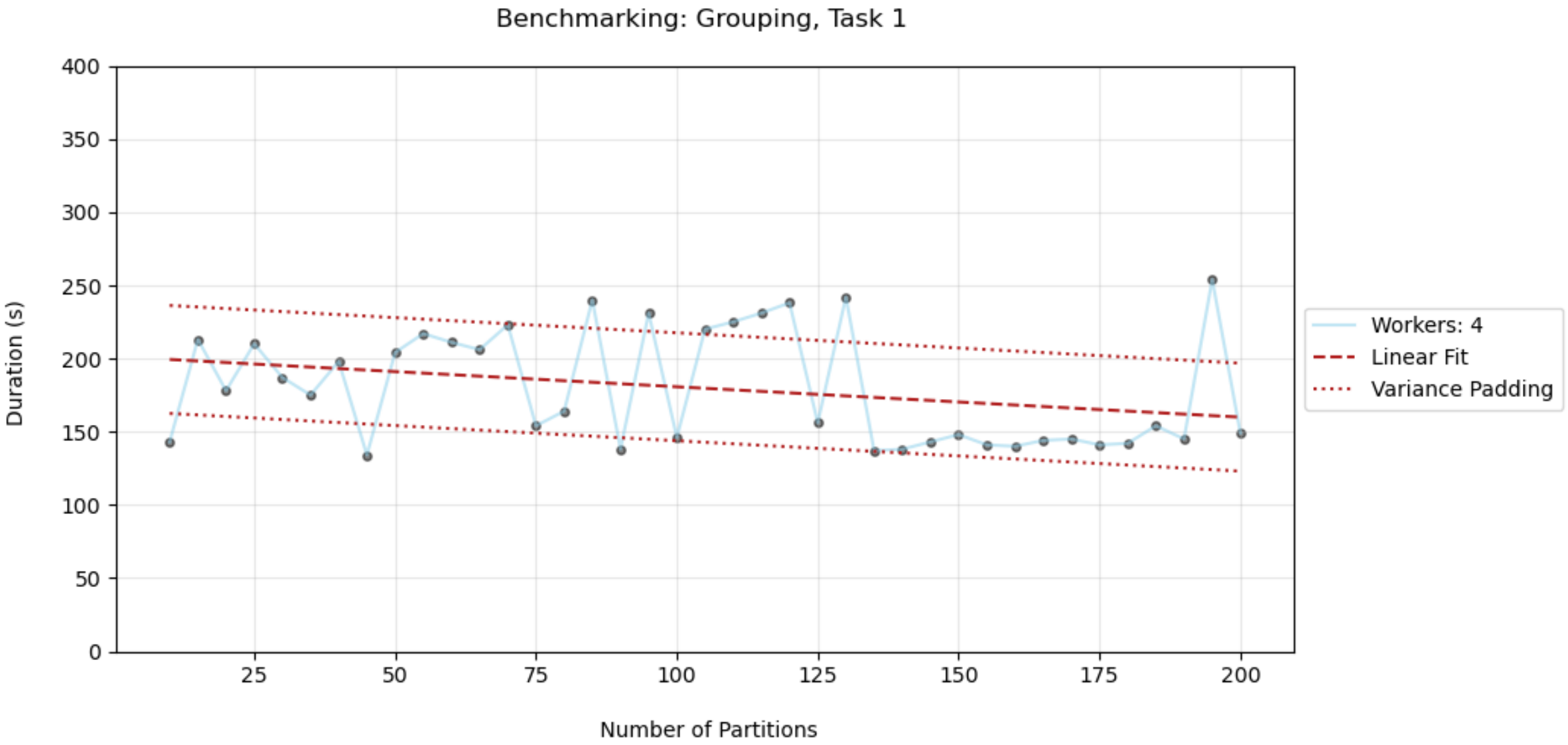
WORKERS VS PARTITIONS



WORKERS VARYING



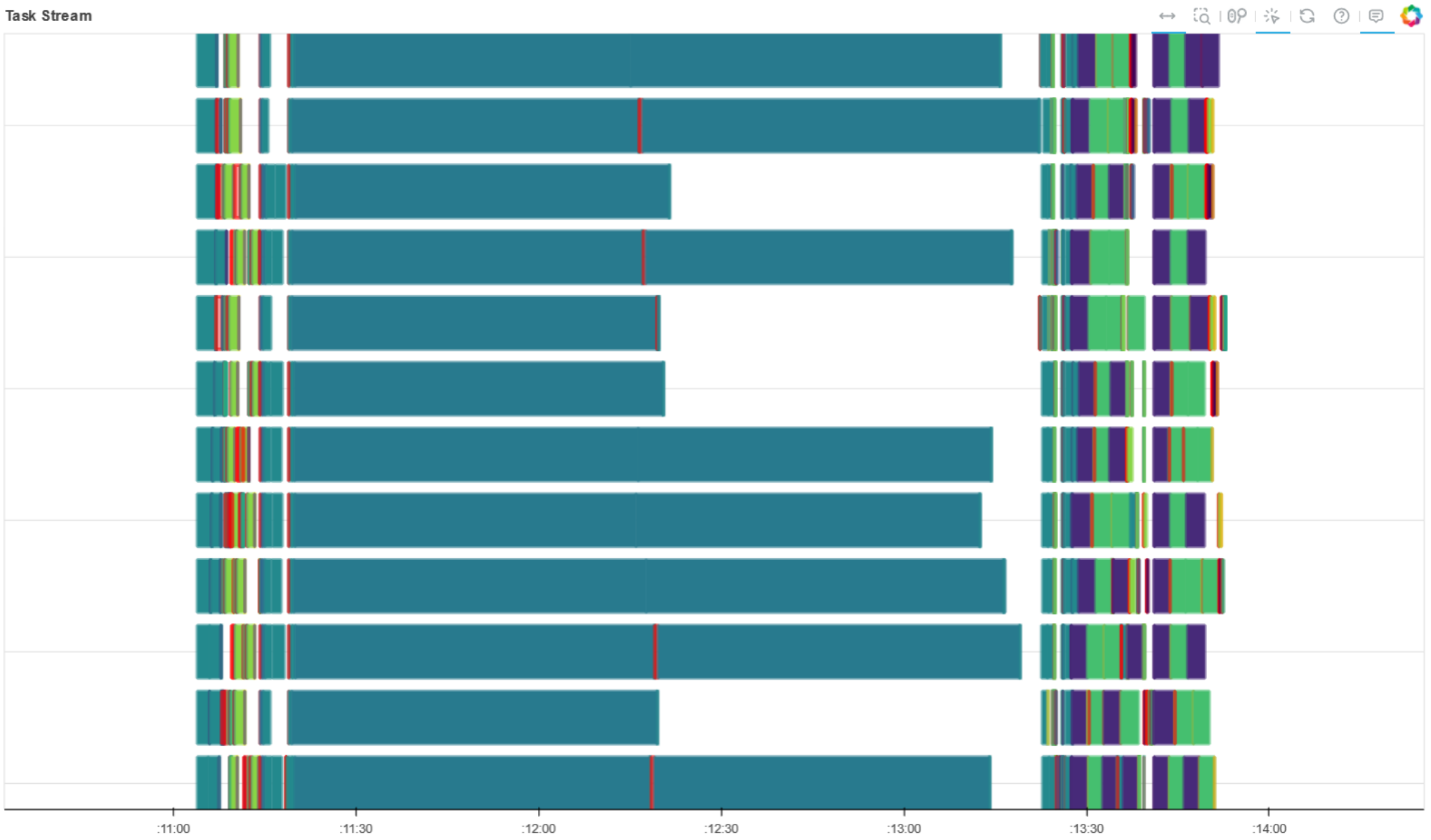
PARTITIONS VARYING



TASK STREAM:

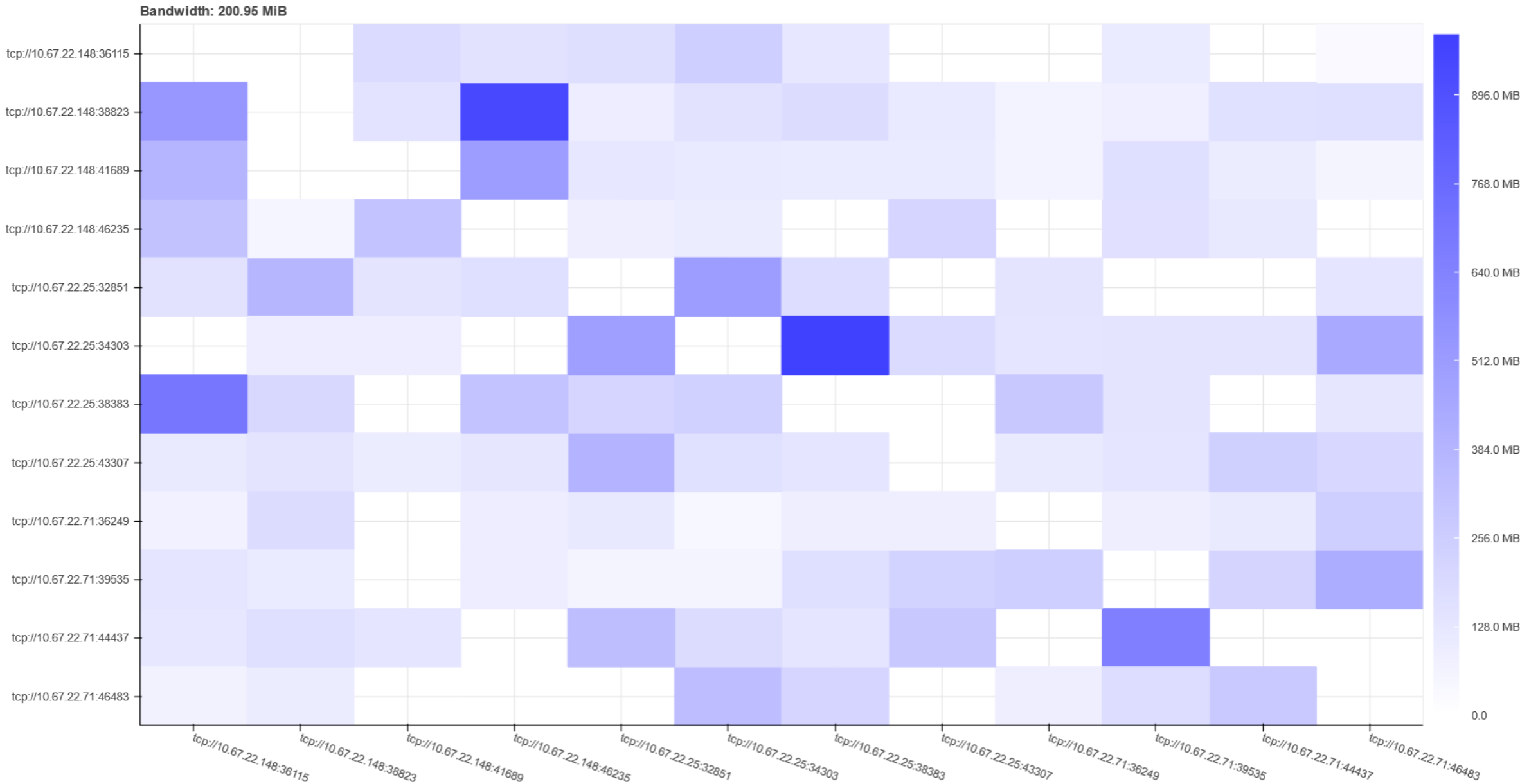
NW = 4, NP = 50, Grouping+Task_1

TASK STREAM:
NW = 4, NP = 20, Grouping+Task_1



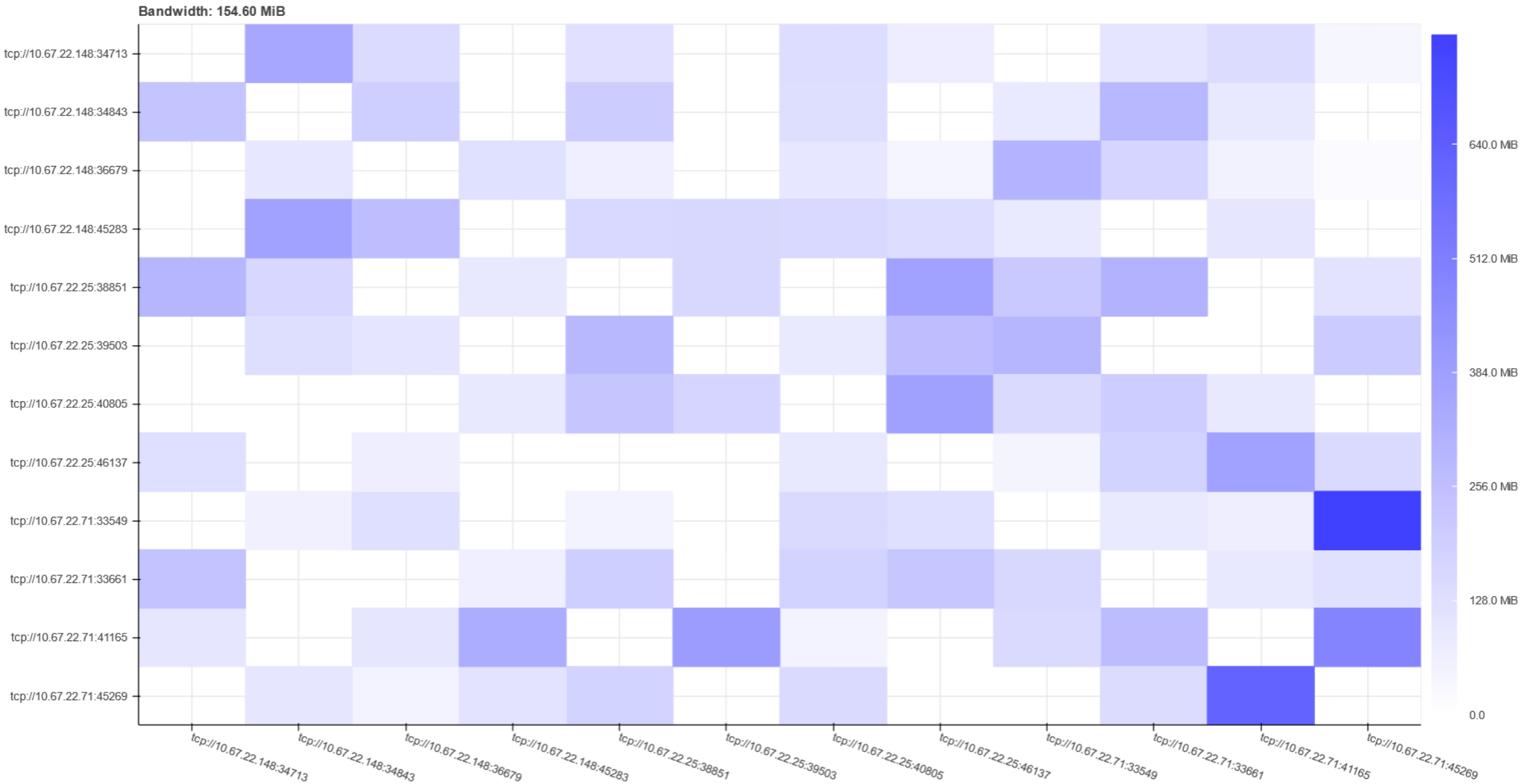
WORKER BANDWIDTHS:

NW = 4, NP = 50, Grouping+Task_1

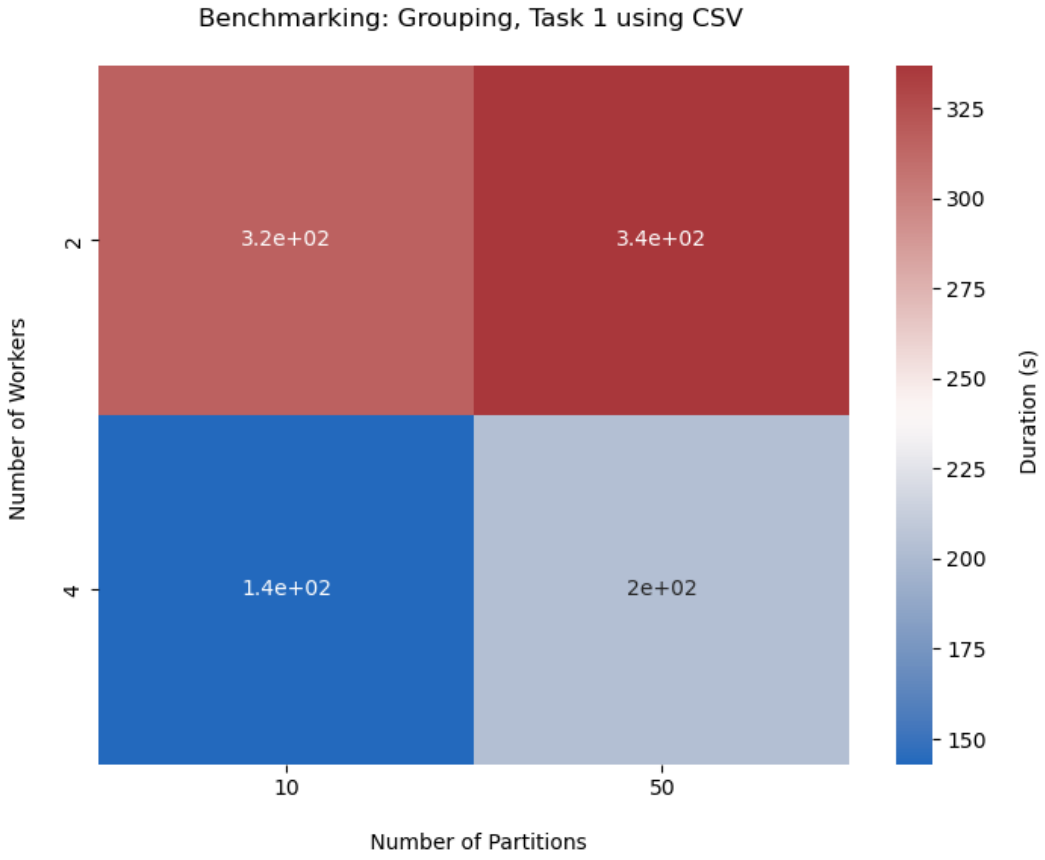


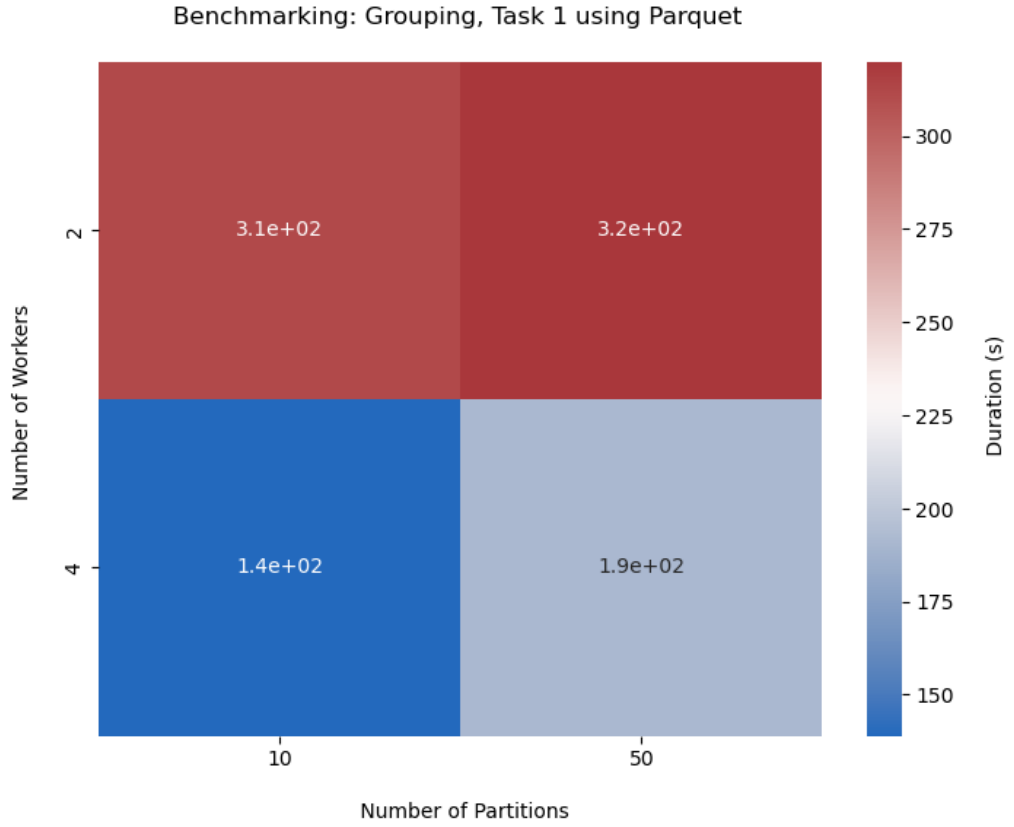
WORKER BANDWIDTHS:

NW = 4, NP = 20, Grouping+Task_1



CSV VS PARQUET





THANK YOU

GROUP 18