Project Title: Broad Advanced Intelligent Networked (BRAIN) System (SBIR/STTR: NASA)
Date: 09/24/2019
Sub-contract: Dr. Pradeep Chowriappa, and Dr. Michael O'Neal in collaboration with AGNC.

Problem statement: Analyzing changes in evolving data from advanced habitation systems. The goal of this task is to create a framework that monitors and provides change detection in a multimodal system. **Our objective** is two-fold (a) To provide the mining of patterns from older data as changes in data could reflect long term/ previous trends. (b) Mining for patterns over near data – as recent changes in data could indicate the recurrences of the previously known pattern or an upcoming event.

Task 1.1: An exhaustive survey of related datasets that capture multi-modal environments/systems, whose specific aims are as follows.

1. Conduct exploratory data analysis of the following datasets
    a. NASA Bearing Data Set
    b. NASA Battery Data Sets
2. In addition, investigate related datasets specifically
    a. NASA Randomized Battery Usage Data Sets
3. Extract relevant data from these datasets

Task 1.2: Data Streaming platforms - to provide a continuous processing system (CPS) using Spark SQL engine for structural streaming data, whose specific aim is as follows

1. Investigate the use of Apache Spark for streaming analytics

Task 2.1: Investigating the Statistical based approaches of Concept drift detection. The specific aims of this task are as follows:

1. Understanding the types of concept drift and its patterns.
2. Groundwork for the generic schema for an online adaptive learning algorithm.

Task 2.2: Retrospective Change point detection. The specification aims are as follows:

1. Implementation of proposed algorithm for Change point detection using SVD using multi-modal data.
2. Integrating with data streaming platform

Observed Red Flags:

1. **Yet to work with Relevant Multi-Modal Steaming data**
    1.1. After wrangling the datasets, we found that the NASA Battery Data Sets is not what we are looking for. The batteries are still useful after the experiments
    1.2. Need to identify how apache spark streaming is integrated with Kafka (input source).
2. **We are yet to integrate change detection with the data stream framework.**
    2.1. We are yet to fix the errors while integrating MOA to R.

Timeline (tentative timeline for the upcoming week)

Task 1.1: Datasets and CNN	**10/02**	**10/03**	**10/04**	**10/07**	**10/08**	**10/09**
1.1.1 Looking at CNNs in TensorFlow						
1.1.2 Explore relevant datasets						
Task 1.2: Streaming Data Platforms						
1.2.1 Identifying the approach for Apache Spark Streaming with Kafka Integration						
1.2.2 Implementation of change detection based on non-parametric approaches						
Task 2: Concept Drift Detection						
Task 2.1: Integrating the R interface to MOA.						
Task 2.2: Investigate the sequential analysis and window-based concept drift detection methods.						
Task 2.3: Integrating the bearings dataset to the drift detection methods.						

References:

[1] Brian Bole (2015). Description of Room Temperature Random Walk Charging and Discharging Data Sets
[2] Brian Bole (2015). Description of Room Temperature Random Walk Discharging Data Sets
[3] Brian Bole (2015). Description of Room Temperature Random Walk Discharging Experiments with Variable Recharging Periods
[4] Brian Bole (2015). Description of Right Skewed Random Walk Discharging Data Sets at 40C
[5] Brian Bole (2015). Description of Right Skewed Random Walk Discharging Data Sets
[6] Brian Bole (2015). Description of Left Skewed Random Walk Discharging Data Sets at 40C
[7] Brian Bole (2015). Description of Left Skewed Random Walk Discharging Data Sets
[8] Hai Qiu, Jay Lee, Jing Lin. "Wavelet Filter-based Weak Signature Detection Method and its Application on Roller Bearing Prognostics." Journal of Sound and Vibration 289 (2006) 1066-1090
[9] S. Aminikhanghahi and D. J. Cook, "A survey of methods for time series change point detection," *Knowledge and Information Systems*, vol. 51, no. 2, pp. 339–367, Aug. 2016.
[10] A. Bifet, G. Holmes, and B. Pfahringer, "MOA-TweetReader: Real-Time Analysis in Twitter Streaming Data," *Discovery Science Lecture Notes in Computer Science*, pp. 46–60, 2011.
[11] "Analysis of real-time data with spark streaming," *Journal of Advances in Technology and Engineering Research*, vol. 3, no. 4, 2017.
[12] Baena-Garcıa, M., del Campo-Ávila, J., Fidalgo, R., Bifet, A., Gavalda, R., & Morales-Bueno, R. (2006, September). Early drift detection method. In *Fourth international workshop on knowledge discovery from data streams* (Vol. 6, pp. 77-86).
[13] Liu, Song & Yamada, Makoto & Collier, Nigel & Sugiyama, Masashi. (2012). Change-Point Detection in Time-Series Data by Relative Density-Ratio Estimation. Neural Networks. 10.1016/j.neunet.2013.01.012.
[14] Sugiyama, M. & Suzuki, T. & Kanamori, Takafumi. (2010). Density Ratio Estimation: A Comprehensive Review. RIMS Kokyuroku. 10-31.

Appendix A

Results from Task 1.1

Three datasets that were found had the potential to be explored as a multi-modal system.

One of them may be promising, since it has samples in throughout the lifespan of at least a bearing at a time.

Some plots were done to see how the data would behave.

Besides having the complete lifespan (test-to-failure) of at least one bearing per experiment, it also is a multi-modal dataset since all the bearings of each test were running and being measured at the same time, inside the same system.

The first test had the first 43 measurements taken every 5 minutes but besides that, all the other measurements were taken every 10 minutes.

Each sample (file) had a sampling rate of 20kHz and a one second duration with 20480 points. Each row is a data point. All the files are in ASCII format, with no extension.

Test 1 has 2,156 files and each bearing has 2 channels. At the end of the test-to-failure experiment, inner race defect occurred in bearing 3 and roller element defect in bearing 4.

Test 2 has 984 files and each channel represents a bearing. At the end of the test-to-failure experiment, outer race failure occurred in bearing 1.

Test 3 has 4,448 files and each channel represents a bearing. At the end of the test-to-failure experiment, outer race failure occurred in bearing 3.

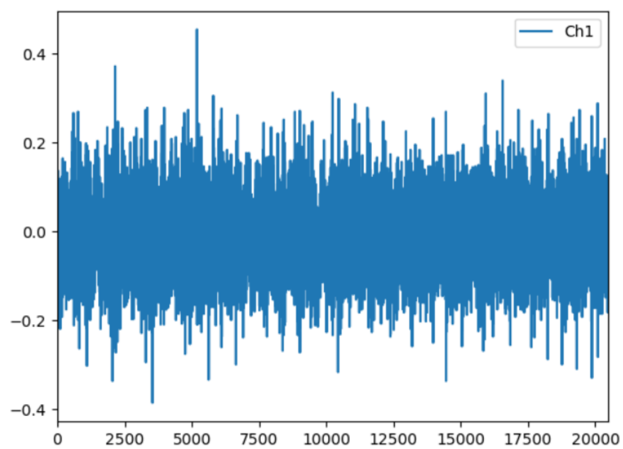


Fig 1 – Beginning of the experiment, bearing 1 (the bearing that failed) at the second test

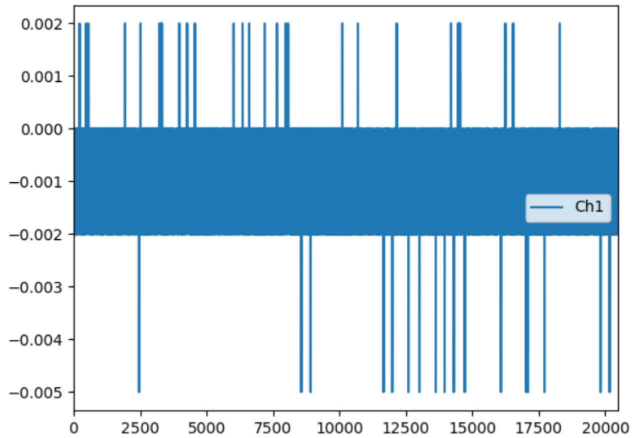


Fig 2 – End of the experiment, bearing 1 (the bearing that failed) at the second test

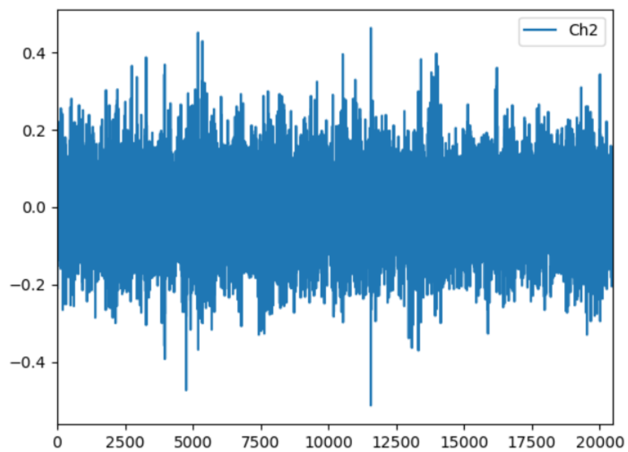


Fig 3 – Beginning of the experiment, bearing 2 (for comparison) at the second test

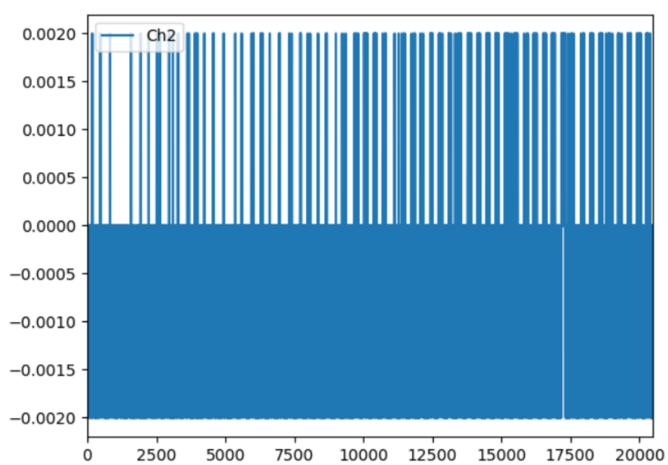


Fig 4 – End of the experiment, bearing 2 (for comparison) at the second test

Project Title: Broad Advanced Intelligent Networked (BRAIN) System (SBIR/STTR: NASA)
Date: 09/24/2019
Sub-contract: Dr. Pradeep Chowriappa, and Dr. Michael O'Neal in collaboration with AGNC.

The NASA Battery Data Sets looked promising with tests with different room temperatures:

- 43oC
- 24oC
- 4oC
- multiple (24oC and 44oC)

different discharge types:

- CC, 4A
- CC, 2A
- Multiple (1A, 2A & 4A) 0.05Hz square wave load
- 4A and 50% cycle

Stopping criteria:

- 20% fade in capacity
- 30% fade in capacity
- Until the experiment control software crashed

*Note that there are several discharge runs where the capacity was very low. The cause for this is unknown.

Each test was made with 3 or 4 batteries at a time.

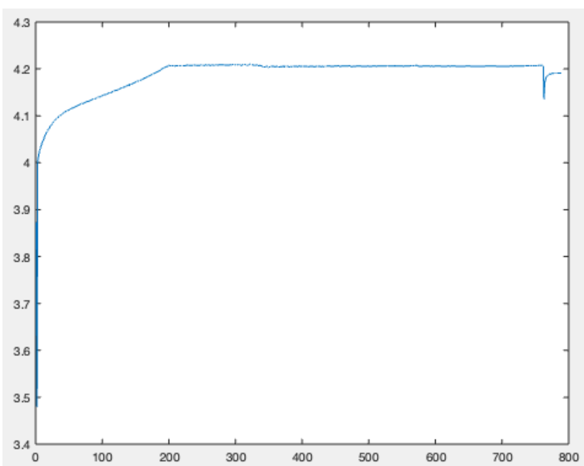


Fig 5 – Measured voltage of a battery in one of the datasets (while charging)

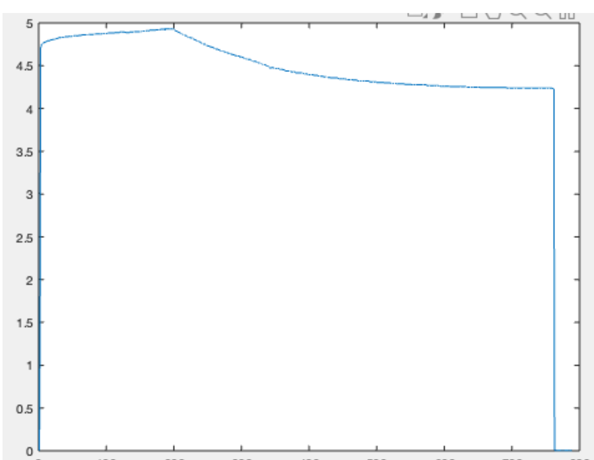


Fig 6 – Charged voltage of a battery in one of the datasets (while charging)

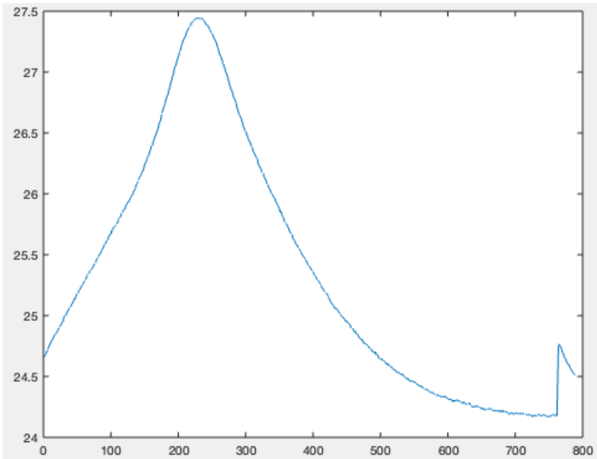


Fig 7 – Measured temperature of a battery in one of the datasets (while charging)

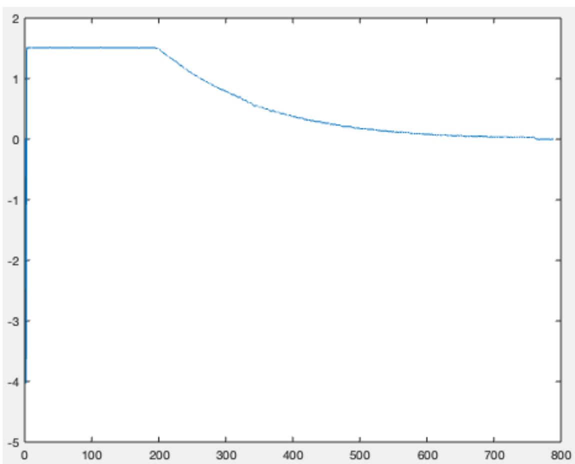


Fig 8 – Measured current of a battery in one of the datasets (while charging)

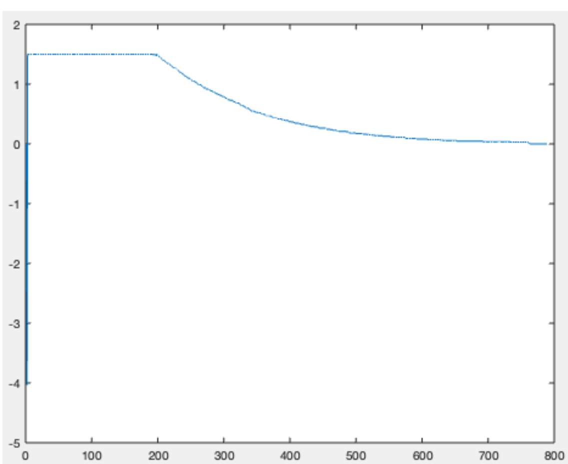


Fig 9 – Charged current of a battery in one of the datasets (while charging)

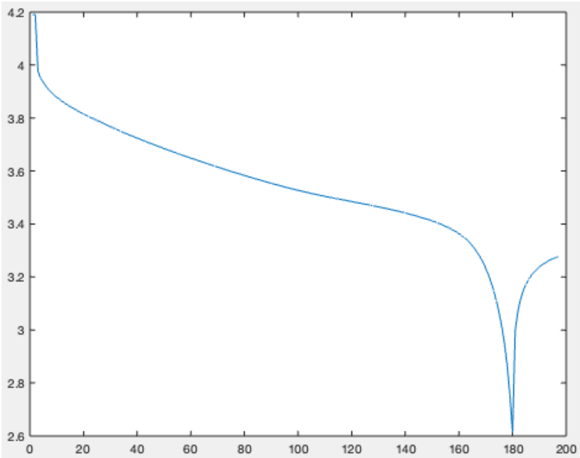


Fig 10 – Measured voltage of a battery in one of the datasets (while discharging)

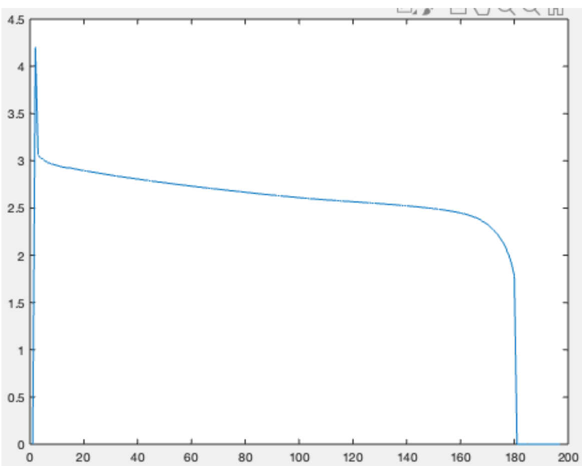


Fig 11 – Voltage load of a battery in one of the datasets (while discharging)

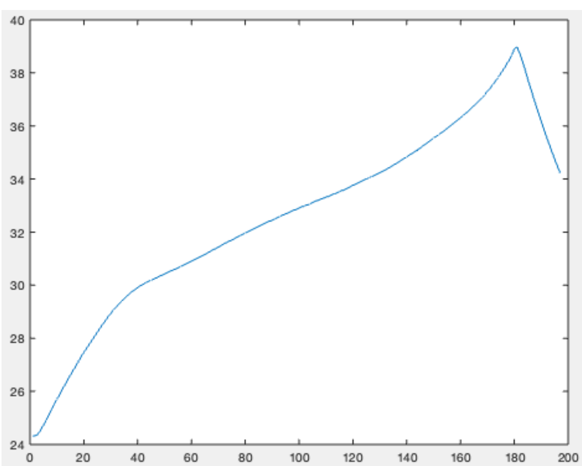


Fig 12 – Measured temperature of a battery in one of the datasets (while discharging)

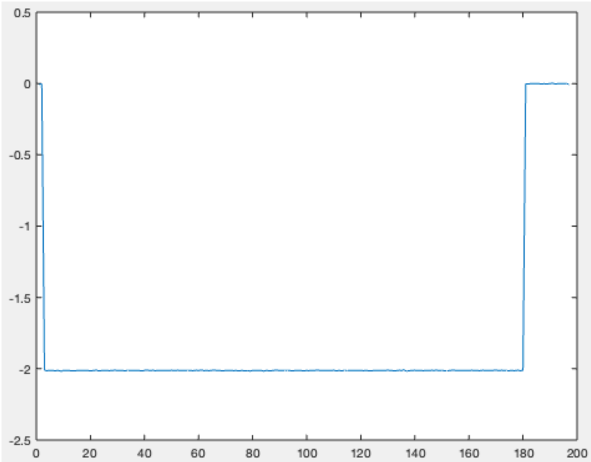


Fig 13 – Measured current of a battery in one of the datasets (while discharging)

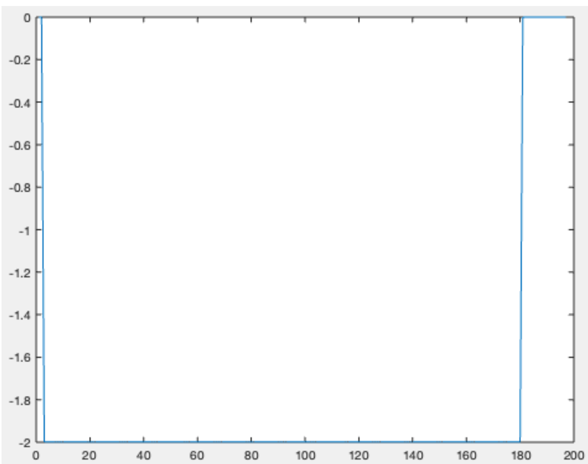


Fig 14 – Charged current of a battery in one of the datasets (while discharging)

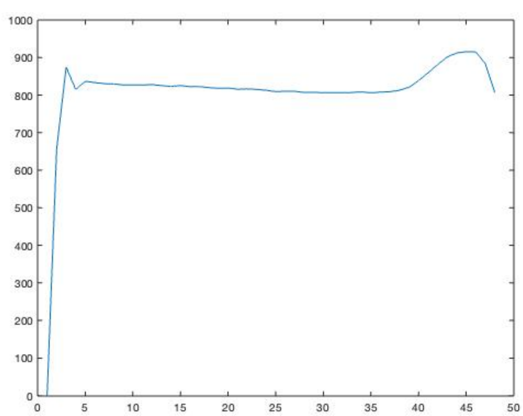


Fig 15 – Sense current of a battery in one of the datasets (impedance, real numbers)

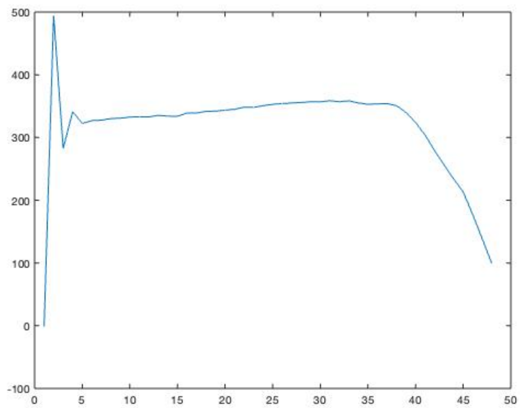


Fig 16 – Battery current of a battery in one of the datasets (impedance, real numbers)

Along with each one of the NASA Randomized Battery Usage Data Sets, there is a HTML file with a complete description of the dataset and some graph plots.

Even though the tests were not done until failure, these datasets have some very interesting data.

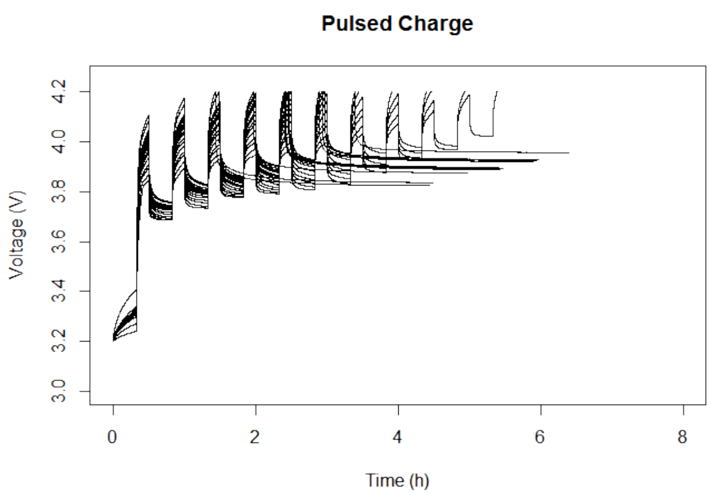


Fig 17 – Pulsed charge on Room Temperature Random Walk Charging and Discharging Data Sets

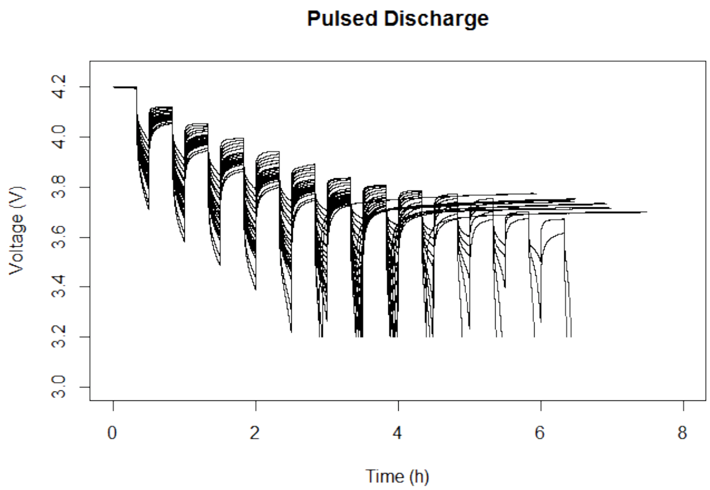


Fig 18 – Pulsed discharge on Room Temperature Random Walk Charging and Discharging Data Sets

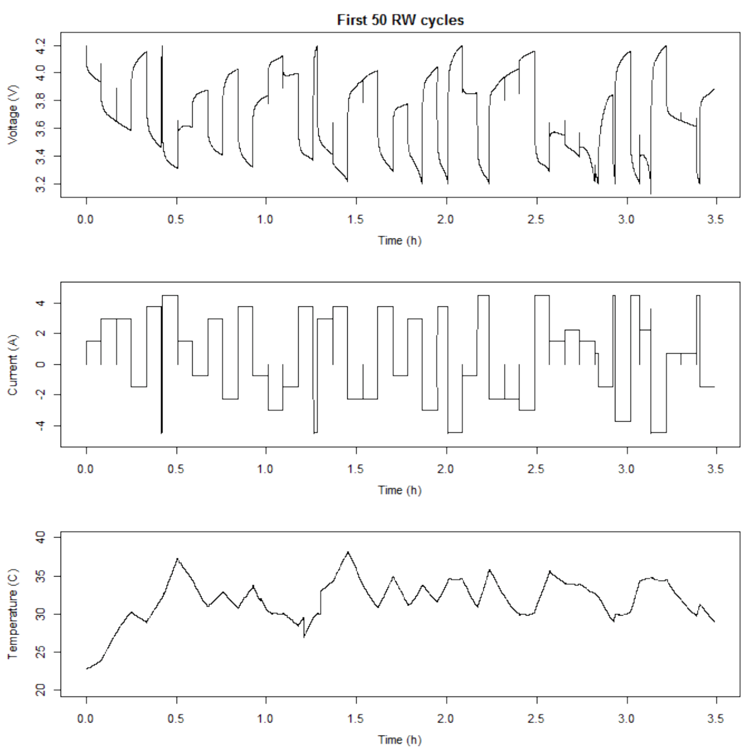


Fig 19 – First 50 RW cycles on Room Temperature Random Walk Charging and Discharging Data Sets

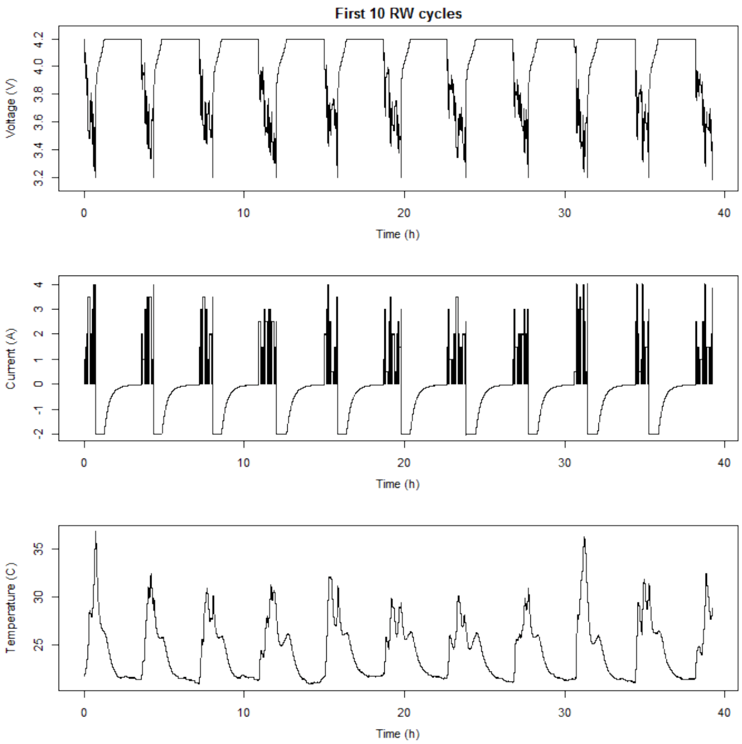


Fig 20 – First 10 RW cycles on Room Temperature Random Walk Discharging Data Sets

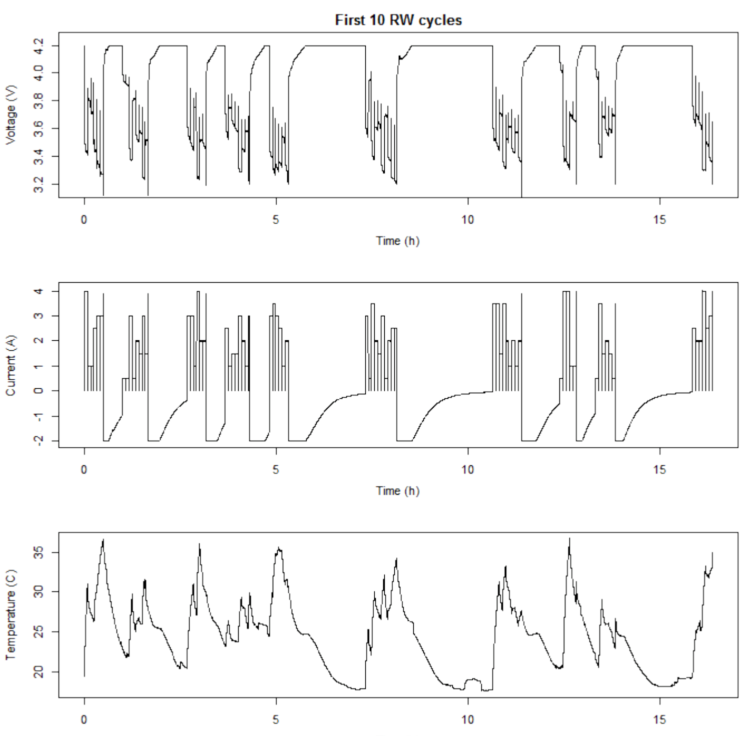


Fig 21 - First 10 RW cycles on Room Temperature Random Walk Discharging Experiments with Variable Recharging Periods

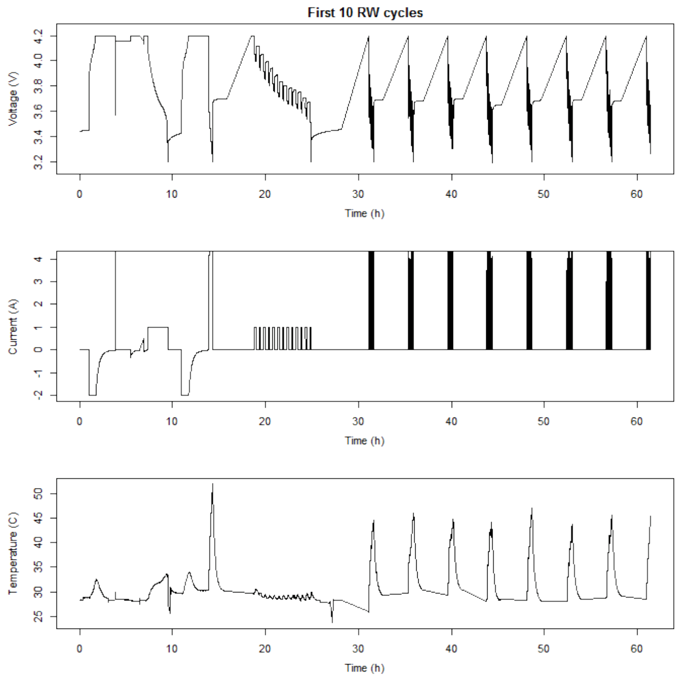
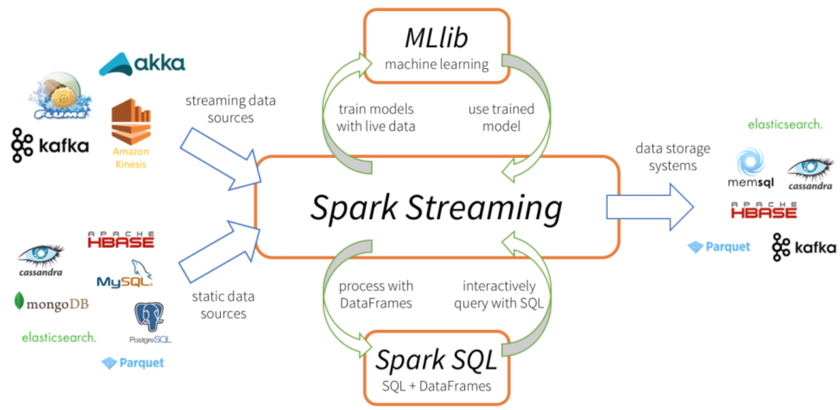


Fig 22 - First 10 RW cycles on Right Skewed Random Walk Discharging Data Sets at 40C

These plots show that these datasets can be promising with the change point detection.

Results from Task 1.2



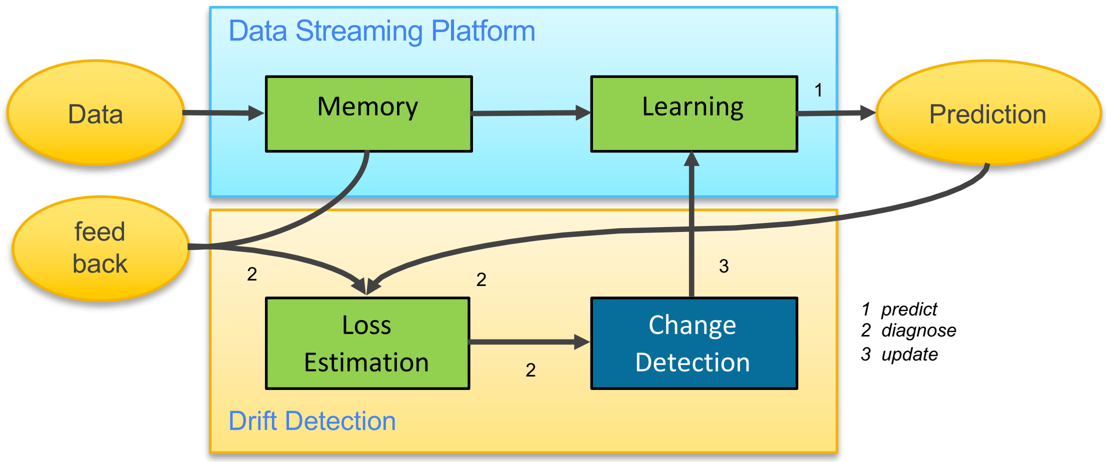
Schematic representation of Spark Steaming ecosystem for Continuous Processing System (CPS).

PREREQUISITES:

1. Setting path for latest Java version (1.8.0) and spark Hadoop (2.4.4) are done in same folder, where Jupyter python files are saved.
2. A list of packages are imported for apache spark streaming:
   a. pyspark: Python API written in python to support Apache Spark. It requires python to be available on the system path.
   b. findspark: It helps to find the location of apache spark Hadoop.
   c. SparkContext: It is the entry point to any spark functionality.
   d. StreamingContext: It represents the connection to a spark cluster and can be used to DStream various input sources such as Kafka, Flume and Kinesis.
   e. SQLContext: It is the entry point into all SQL functionality in spark.

Results from Task 2.1

Architecture



Change detection: This component refers to the techniques and mechanisms for **explicit drift** and **change detection**. It characterizes and quantifies concept drift by identifying change points or small-time intervals during which changes occur.

Learning

The Learning component refers to the techniques and mechanisms for generalizing from examples and updating the predictive models from evolving data.

Active learning algorithms rely on immediate arrival of feedback (true labels).

In reality labels may become known immediately in the next time step after casting the prediction. However, feedback may come with an uncontrollable delay, be unreliable, biased or costly. Labels may arrive within a fixed or variable time lag.

Drift Detection Method

The Drift Detection Method (DDM), monitors the error-rate of the classifier to detect concept drift. On the basis of the probably approximately correct (PAC) learning model, the classification error-rate decreases or stays constant as the number of instances increases. Otherwise, it suggests the occurrence of a drift.

Let p_t be the error-rate of the classifier with a standard deviation of $s_t = \sqrt{p\left(p_t \frac{1-p_t}{t}\right)}$ at time t. As instances are processed, DDM updates two variables $p_{\min}$ and $s_{\min}$ when $p_t + s_t < p_{min} + s_{\min}$.

DDM warns for a drift when $p_t + s_t \geq p_{min} + 2 \times s_{\min}$, and alarms for a drift when $p_t + s_t \geq p_{min} + 3 \times s_{\min}$. The variables $p_{\min}$ and $s_{\min}$ are reset when a drift occurs.

Types of drifts

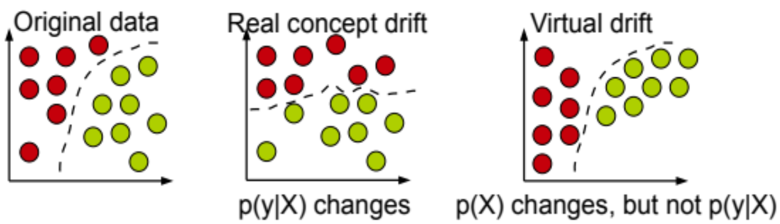


Fig. 1. Types of drifts: circles represent instances, different colors represent different classes.

<u>**Patterns of changes**</u>

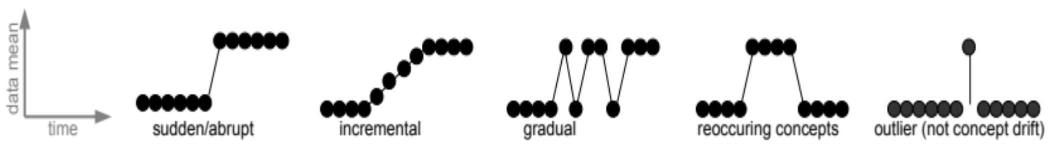
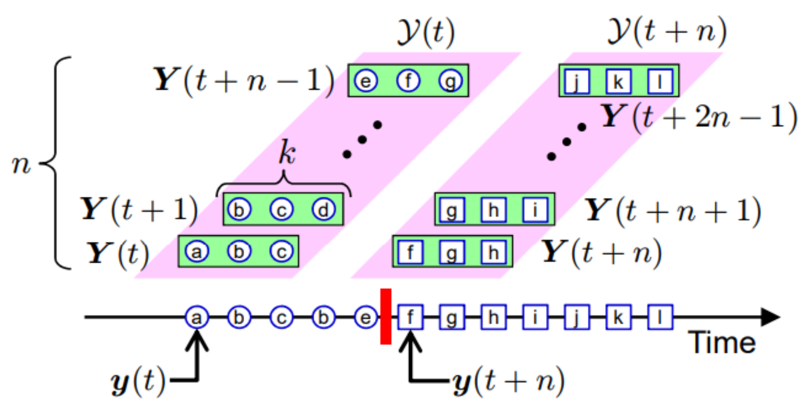


Fig. 2. Patterns of changes over time (outlier is not concept drift).

Results from Task 2.2

Foundations of the Methods Used for Change Detection



- We take d-dimensional data and encapsulate it into an array, Y, at time t, $Y(t)$ of size k.
- We then take all of the $Y(t)$'s up until $Y(t + n - 1)$ and call that $y(t)$.
- We take all of the $y(t)$'s up until a certain time point.
- We take that $y(t)$ and we compare it to all of the $y(t)$'s after that time point, $y(t + n)$.