A REPORT

ON

SENSORLESS ANOMALY DETECTION IN INDUSTRIAL MOTORS

BY

Mahalakshmi P.

2021HT01070

AT

MICROCHIP TECHNOLOGY INDIA PVT LTD, CHENNAI

333

BIRLA INSTITUTE OF TECHNOLOGY & SCIENCE, PILANI
APRIL 2023

A REPORT

ON

SENSORLESS ANOMALY DETECTION IN INDUSTRIAL MOTORS

BY

Mahalakshmi P.

2021HT01070

M. Tech. Embedded Systems

Prepared in partial fulfilment of the
WILP Dissertation/Project/Project Work Course

AT

Microchip Technology India Pvt Ltd, Chennai

BIRLA INSTITUTE OF TECHNOLOGY & SCIENCE, PILANI
APRIL 2023

Acknowledgements

I would like to express my profound gratitude to my Supervisor, Ineyaa N and my Additional Examiner, Enoch Richbert , for their continued support and guidance that helped me complete this project.

I would also like to thank my faculty mentor, Soumyabrata Barik, for his feedback on the project that helped me improve its quality.

BIRLA INSTITUTE OF TECHNOLOGY AND SCIENCE
PILANI (RAJASTHAN)
WILP Division

Organization: Microchip Technology India Pvt Ltd
Location: Chennai
Duration: 4 months
Date of Start: 10 Jan 2023
Date of Submission: 23 April 2023

Title of the Project: SENSORLESS ANOMALY DETECTION IN INDUSTRIAL MOTORS
ID No./Name of the student : Mahalakshmi P., 2021HT01070

Name (s) and
Designation (s) of
your Supervisor and Additional Examiner :

Supervisor: Ineyaa N, Principal Application Engineer
Additional Examiner: Enoch Richbert, Senior-II Application Engineer

Name of the
Faculty mentor:- Soumyabrata Barik

Key Words: Motor control, sensorless, Anomaly, Machine learning
Project Areas: Machine Learning

Abstract:
Objective of the project is to build cost effective sensorless industrial motor maintenance system and optimizing efficiency of the system by integrating Machine Learning algorithms to detect anomaly of the motor, focusing on Unbalanced load in both forward and reverse direction of motor rotation.

Pmala

Mahalakshmi P.

Signature of Student
Date: 21 APRIL 2023

Ineyaa N

Signature of your Supervisor
Date: 21 APRIL 2023

Contents

1. Introduction

Machine Learning is the technology for identifying the possibilities concealed in the data and converting them into real opportunities. In today's modern industries Machine Learning is used for various purposes, one of the major areas is Smart Predictive maintenance.

Adapting robust management systems for maintenance work can decrease unpredicted costs during equipment failures and shutdown periods. Smart Predictive Maintenance uses the analytic abilities of machine learning to monitor and predict machine failure thereby reducing maintenance costs, reducing unscheduled equipment downtime caused by equipment or system failure and increasing uptime. This also ensures safety by detecting anomalies in the earlier state.

One technique for controlling the speed and torque of a BLDC motor is sensorless field-oriented control (FOC). A 3-phase sinusoidal modulation is produced using a technique called field-oriented control, and its frequency and amplitude can then be adjusted. Instead of using position sensors, sensorless control employs electromotive force (EMF) to detect the position of the rotor.

Normally, anomaly detection is done using gyro or accelerometer sensors, using FOC eliminates sensor cost.

Efficient anomaly detection application in the industrial motors can be implemented by combining both Machine learning Predictive Maintenance and Motor control FOC algorithms.

2. Design Considerations

- Sensorless to avoid using gyro or accelerometer sensors.
- FOC to eliminate ssensor cost.
- 48MHz target device Operating frequency.
- 5V target device operating voltage.
- Motors :
  - Test motor : BLDC 075
  - DYNO Motor : BLDC 300 with encoder
  - Operating voltage : 24V
  - Current rating : 1 Amp
- Designed to detect two types for motor failures:
  - Unbalanced load
    - Forward direction
    - Reverse direction
- Data capture:
  - UART communication baud rate : 115200
  - Sample rate 250Hz
- Dyno side:
  - UART Communication baud rate : 115200

3. Software and Hardware requirements

The following table consists of hardware and software requirements for developing sensorless anomaly detection in industrial motor application.

Table 1: HARDWARE AND SOFTWARE REQUIREMENTS

Serial No	Hardware requirements	Software requirements
At Motor side:		
1	Short Hurst BLDC Motor	SensiML data capture LAB
2	DSPIC33CK LVMC Board	MPLAB Model builder
3	24V power connector	MPLAB X IDE
4	Motor to board power connector	
5	UART cable	
At Dyno side:		
1	Long Hurst BLDC with encoder Motor	SciLab/X2C
2	MCLV-2 board	MPLAB X IDE
3	SAME54 PIM	
4	24V power cable	
5	Motor to board power connector	
6	Motor to board Encoder connector	
7	UART cable	

3.1 Hardware Requirements

a. Short Hurst BLDC Motor: Hall-Effect sensors are included with this Hurst NT Dynamo 24-Volt 3-Phase BLDC permanent magnet motor for 6-step commutation. It can also be controlled with a Field Oriented Control algorithm. This will be used as Test industrial motor in this application.

Figure 1: Hurst 075 BLDC motor

b. DSPIC33CK LVMC Board: This development board is specifically developed for motor control applications that operates up to 48 Volts and maximum of 10 Amps of continuous current. This board has dsPIC33CK DSC and MIC4605 MOSFET gate drivers. LVMC board is used to control test short Hurst BLDC motor. This also has UART interface circuit to connect with SensiML software for data capturing.

Figure 2: DSPIC33CK LVMC Board

c. Long Hurst BLDC with encoder Motor: Hall-Effect sensors and a 250-line incremental encoder are included in this 24-Volt 3-Phase BLDC permanent magnet Hurst motor for 6-step commutation and position control applications, respectively. The failure pattern of the actual test motor will be induced using this motor.

Figure 3: Hurst 300 BLDC Motor

d. MCLV-2 board: This development board is specifically developed for motor control applications that operates up to 48 Volts and maximum of 10 Amps of continuous current. Has multiple communication channels such as USB, CAN, LIN and RS-232. This board is used to control Long hurst BLDC motor with SAME54 MCU. RS232 is used to connect the board to the SciLab in order to induce motor failure parameters.

Figure 4: MCLV-2 Board

e. SAME54 PIM: SAME54 PIM is mounted on the MCLV-2 board, main controller to control Long hurst BLDC motor.

f. 24V power connector: 24V power supply to power-up both LVMC and MCLV-2 motor control boards.

g. Motor connectors: Power connectors and encoder connectors are used to connect motors with control boards.

h. UART (RS-232) cable: To connect boards to PC to communicate with software Scilab and SensiML.

3.2 Software Requirements:

a. SensiML: SensiML is a machine learning software, will be used to capture the motor current and speed parameters when inducing failure pattern, and labeling the captured model according to failure type and building ML model. It will also be used to train the ML model. LVMC board will Communicate with SensiML through UART interface.

Figure 5: SensiML tool

b. SciLab/X2C: This software is used to build dyno model for Unbalanced load and Broken bearing failures. Using this dyno model, different failure patterns will be induced to the test motor through MCLV-2 board. The communication interface is UART.

Figure 6 :SciLab

c. MPLAB X IDE: This development environment is used to generate application firmware for dsPIC and SAME54 MCUs. The hex file will be generated and programmed into the corresponding LVMC and MCLV-2 boards.

Figure 7: MPLAB X IDE

4. Hardware Setup:

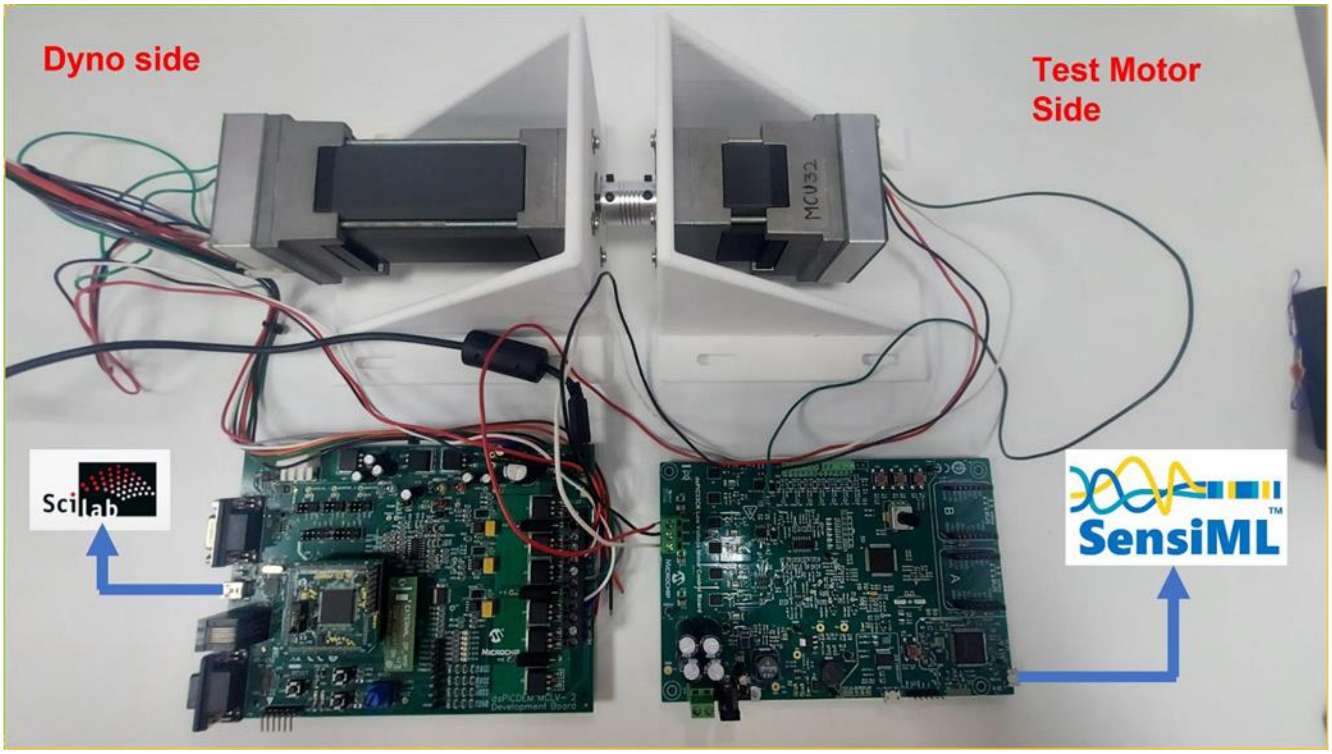


Figure 8: Hardware setup

hardware set-up for the anomaly detection in industrial motors is shown above Which majorly contains test motor side and dyno side. Both are detailed explained below.

4.1 Test Motor Setup :

This setup represents actual industrial motor application. In this module dsPIC motor control board LVMC is connected with short Hurst BLDC motor. Field Oriented Control (FoC) motor control algorithm is used to run the BLDC motor and to control motor's speed. FoC firmware is integrated with data streaming firmware and programmed into the dsPIC33CK245MP508 device.

Streaming firmware is implemented to stream out motor current iq, current id and speed parameters through UART interface to capture using SensiML software to build and train ML model.

4.2 Dyno setup:

This setup induces the motor failure pattern to the test industrial motor setup. This is a dynamometer control setup which uses MCLV2 board development board and SAME54 microcontroller along with long hurst BLDC motor which has Encoder.

With this setup we can apply dynamic loads such as Unbalanced load and Broken bearing to test motor. This setup is connected with Scilab and X2C software through UART interface to change load parameters on the run to induce varying failure patterns. Motor current values will vary for each of the different failure pattern values induced.

5. Technologies used:

5.1 Field Orientation Control(FOC)

Dyno project implemented based on an encoder-based Field Oriented Control (FOC) algorithm on the SAME54 32-bit micro-controller to track the position of the PMSM motor. The following section describes briefly about the FOC algorithm, software design and implementation.

FOC is an Algorithm used to the decoupled control of torque and flux. This is accomplished by transforming the stator phase currents from a stationary reference frame to torque and flux producing currents components in a rotating reference frame using mathematical transformations.

Following is Field Orientation Control Procedure :

1. Measure the motor phase currents.

2. Transform them into the two-phase system (a, b) using the Clarke transformation.[7]

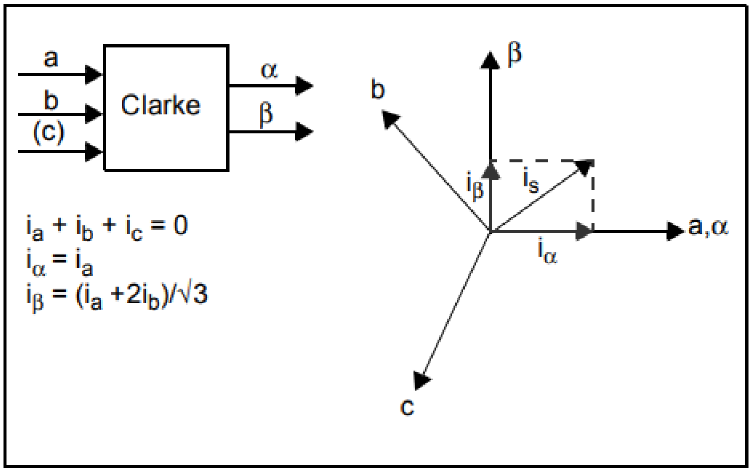


Figure 9: Clarke Transform

3. Calculate the rotor position angle.

4. Transform stator currents into the d,q-coordinate system using the Park transformation.[7]

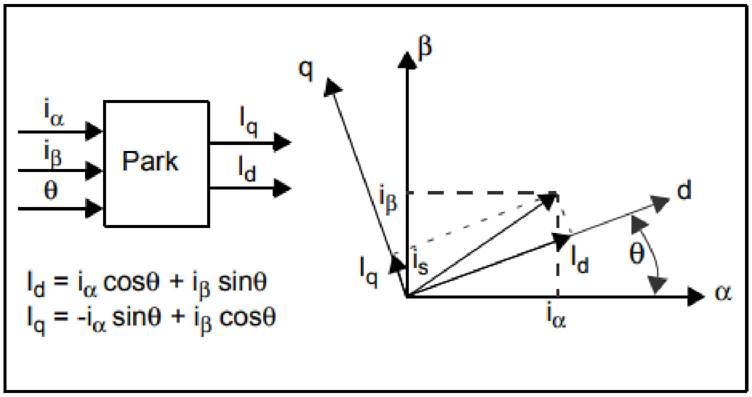


Figure 10: Park Transform

5. Position and speed are controlled by position and speed PI controllers respectively.[7]

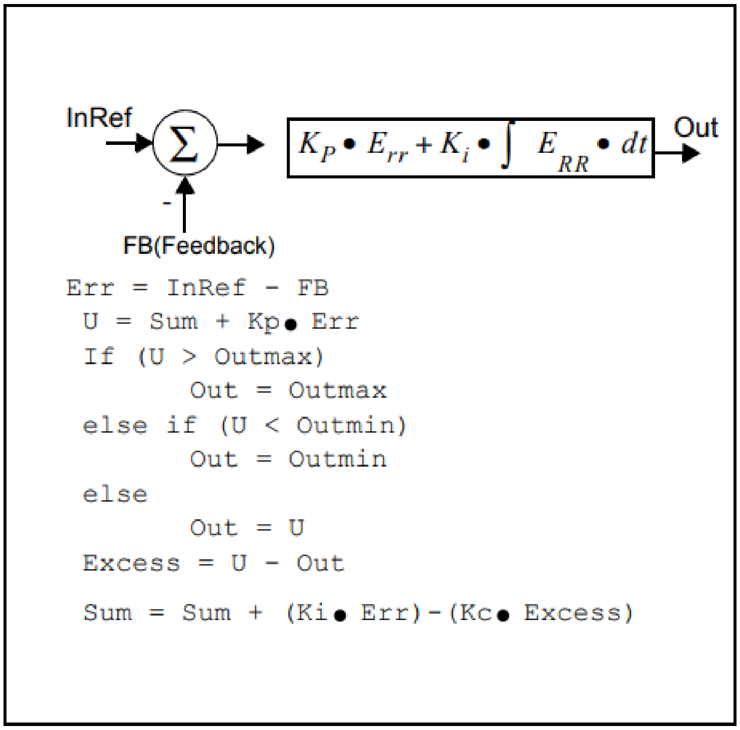


Figure 11: PI Control

6. The stator current torque (i_q) and flux (i_d) producing components are controlled separately by the corresponding PI controllers.

7. The output stator voltage space vector is transformed back from the d, q-coordinate system into the two-phase system fixed with the stator by the Inverse Park transformation.

8. Using the space vector modulation, the three-phase output voltage is generated.

The phase currents are measured using two shunt resistors connected to Phase U and Phase V inverter legs respectively. The W phase current is determined based on Kirchhoff's current law. The rotor position angle is determined from the Quadrature encoder.

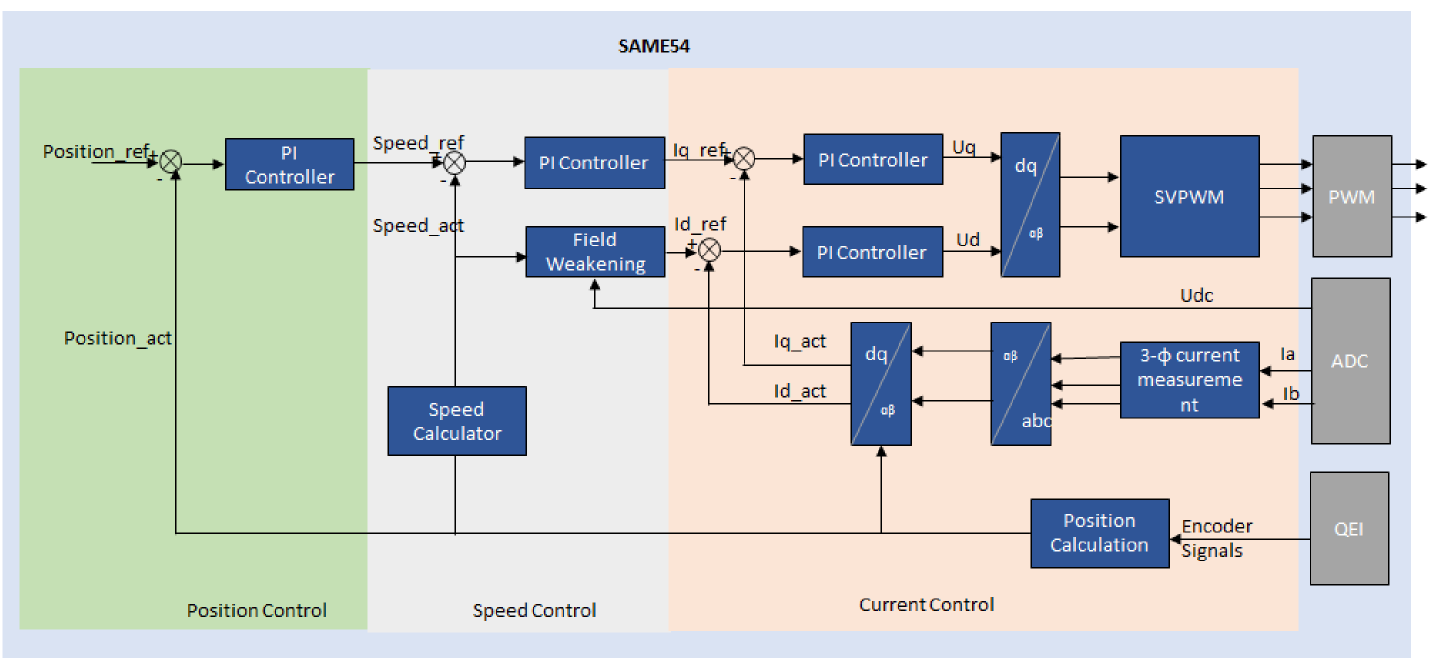


Figure 12: software realization of the FOC algorithm

5.2 Decision Tree Ensemble Machine learning classifier

A group of decision trees are assessed against an input vector to create the decision tree ensemble classifier. Each decision tree in the ensemble makes a single forecast, and the ensemble prediction is determined by the majority vote of all the trees.[6]

5.2.1 xGBoost

Decision Tree ensemble and boosted tree classifiers can be trained using the xGBoost training algorithm.[6]

Parameters:

- Input feature vectors with a label column -- input_data (DataFrame)
- The name of the column in input_data containing labels -- label_column (str)
- The max depth to allow a decision tree to reach -- max_depth (int)
- The number of decision trees to build -- n_estimators (int)

Returns:

- A trained model

5.2.2 Random Forest

Utilize the random forest training algorithm to train a group of decision tree classifiers. A random forest is a meta-estimator that employs averaging to increase predictive accuracy and reduce overfitting after fitting numerous decision tree classifiers to different dataset subsamples. Although the samples are drawn via replacement, the sub-sample size is always the same as the original input sample size[6].

Parameters:

- Input feature vectors with a label column -- input_data (DataFrame)
- The name of the column in input_data containing labels -- label_column (str)
- The max depth to allow a decision tree to reach -- max_depth (int)
- The number of decision trees to build -- n_estimators (int)

Returns:

- A set of models

6. Application Configurations

6.1 Dyno side configuration

a. TCC0:

- TCC peripheral is configured to generate 20 kHz frequency three pair of PWM complimentary signals at a in "Dual Slope PWM with interrupt or event when counter = TOP" i.e., "Center Aligned Mode"
- The PWM duty cycle has an inverse relation with the compare value in "Dual Slop PWM" mode. For example, for a Period Count (PER) = 3000, Compare Count (CCx) = 2000 (66%), would result in a PWM with a 33% duty cycle. In order to achieve a direct relation between comparing count and duty cycle, the PWMxH and PWMxL outputs are swapped.
- The phase current measurements are performed at the end of the PWM cycle in dual shunt configuration. The ADC conversion event is generated when the counter reaches TOP since the PWM outputs are swapped,

- 1uS dead time is configured and enabled.
- All PWM channels are held low when an event is detected on EV0. Non-recoverable Fault is enabled on EV0.

b. ADC0-ADC1:

- ADC1 is setup to operate in Client mode where ADC0 acts as a Host Mode.
- ADC0 samples and converts the Phase U current and ADC1 samples and converts the Phase V current. Both ADCs convert single ended inputs.
- At the end of each PWM cycle, an event created by TCC0 hardware triggers both ADCs simultaneously.
- ADC0 also generates a Conversion Ready interrupt. Both ADCs finish conversion simultaneously because they are both simultaneously triggered, have the same resolution, and sample at the same rate.

c. PDEC:

- To detect the quadrature encoder signals, PDEC is set to Quadrature decoder direction (X4) in QDEC mode.
- The angular position is measured using all 16 bits of the QDEC counter (CTRLA.ANGULAR = 0x7).

d. EIC:

- In the event of an over-current fault, External Interrupt Controller recognizes a hardware over-current fault input and generates a non-recoverable fault event for TCC0, shutting off the PWM.

e. EVSYS:

- EVSYS performs as a channel between event generator and event users.

- When the counter reaches TOP, the TCC0 generates an event that the ADC0 uses as a hardware trigger source through the Event System.
- The EIC's over-current fault event is used by the TCC0's Event System as a non-recoverable fault event.

f. SERCOM2:

- Peripheral SERCOM2 is configured in UART mode and is set to operate at 115200 bps.
- The X2CScope plugin uses UART channel to plot and watch global variables in run-time.

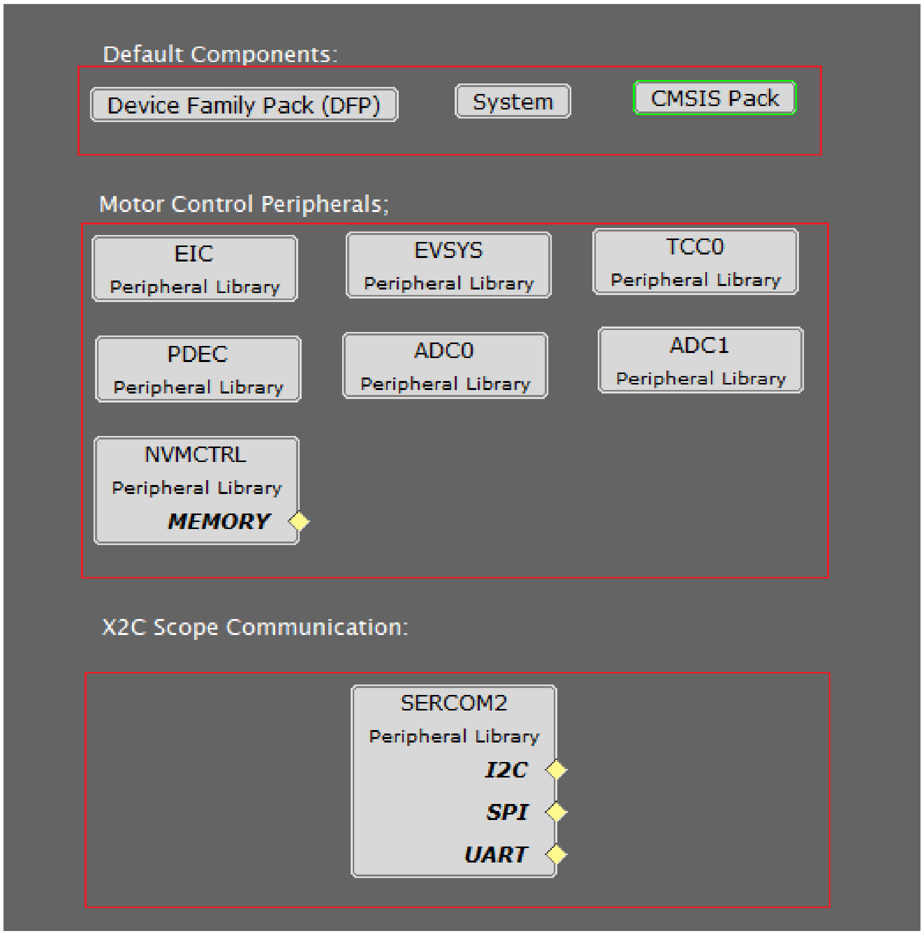


Figure 13: Dyno Project graph

6.2 Test Motor side configuration

Below Project graph shows required components and system configurations for dsPIC33CK256MP506 target device firmware.

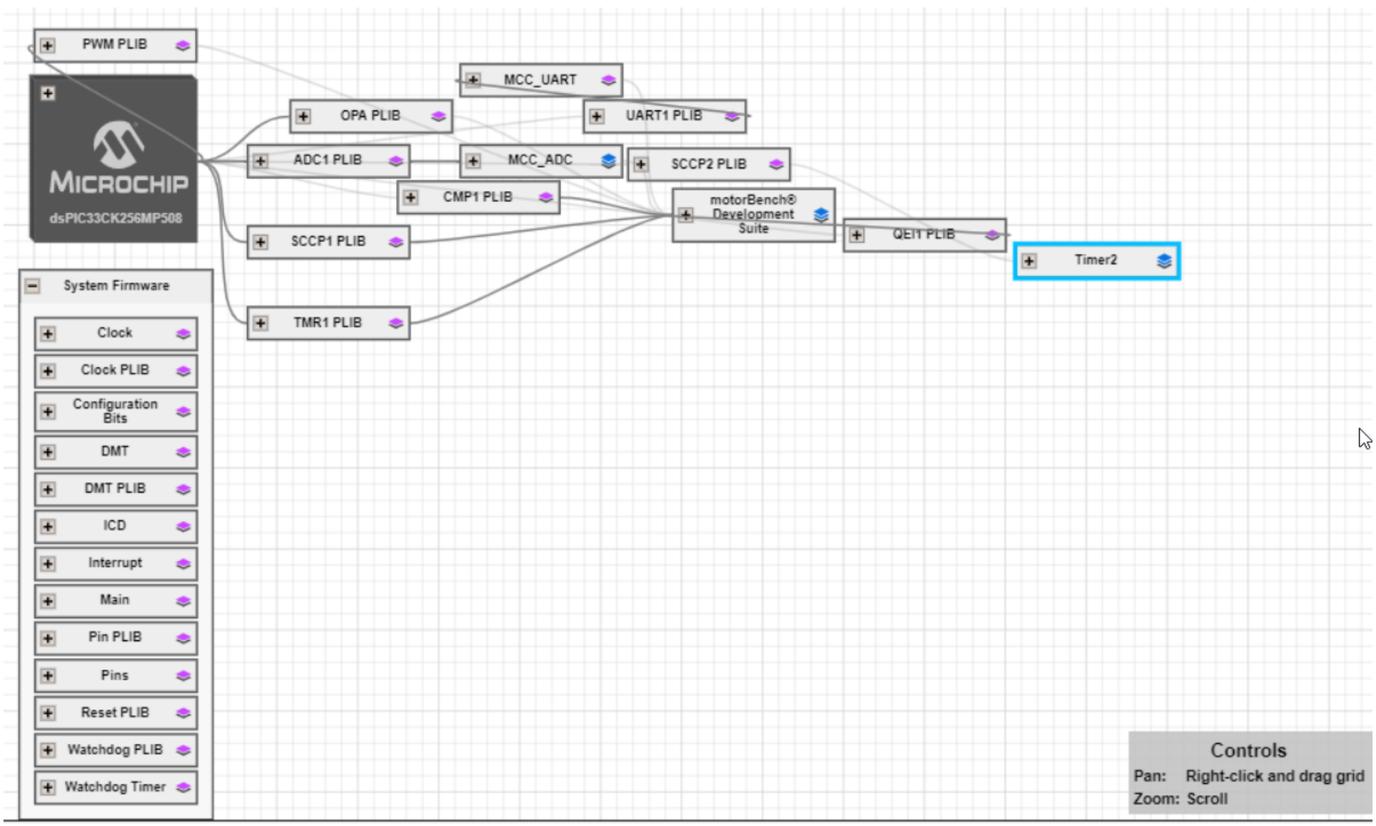


Figure 14: Test motor side Project graph

Following are BLDC Hurst motor parameters that will be fed into MCC motor bench suit.

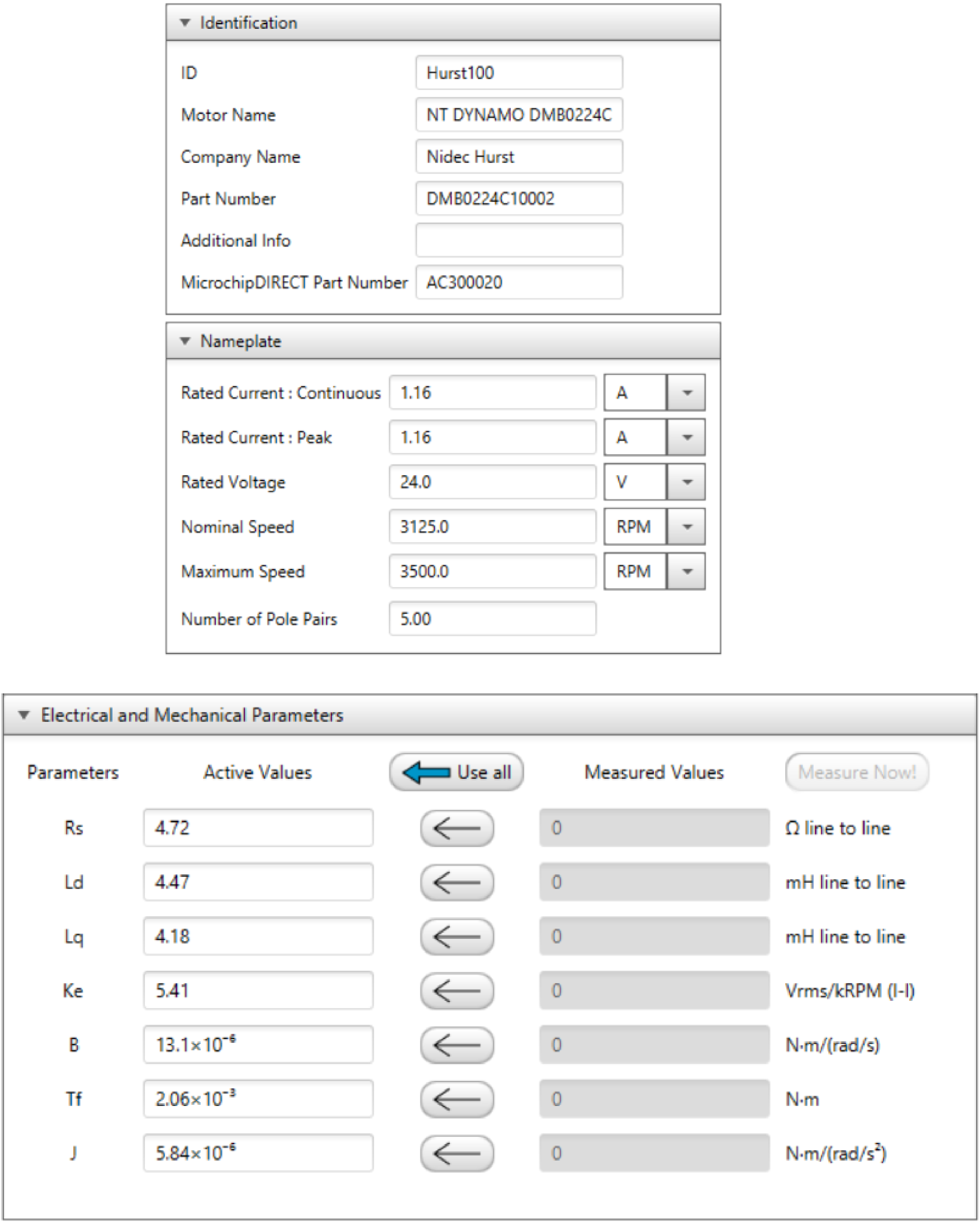


Figure 15: Hurst BLDC 075 Motor electrical Parameters

7. Software Design

The following figure shows the various state machines of the application software.

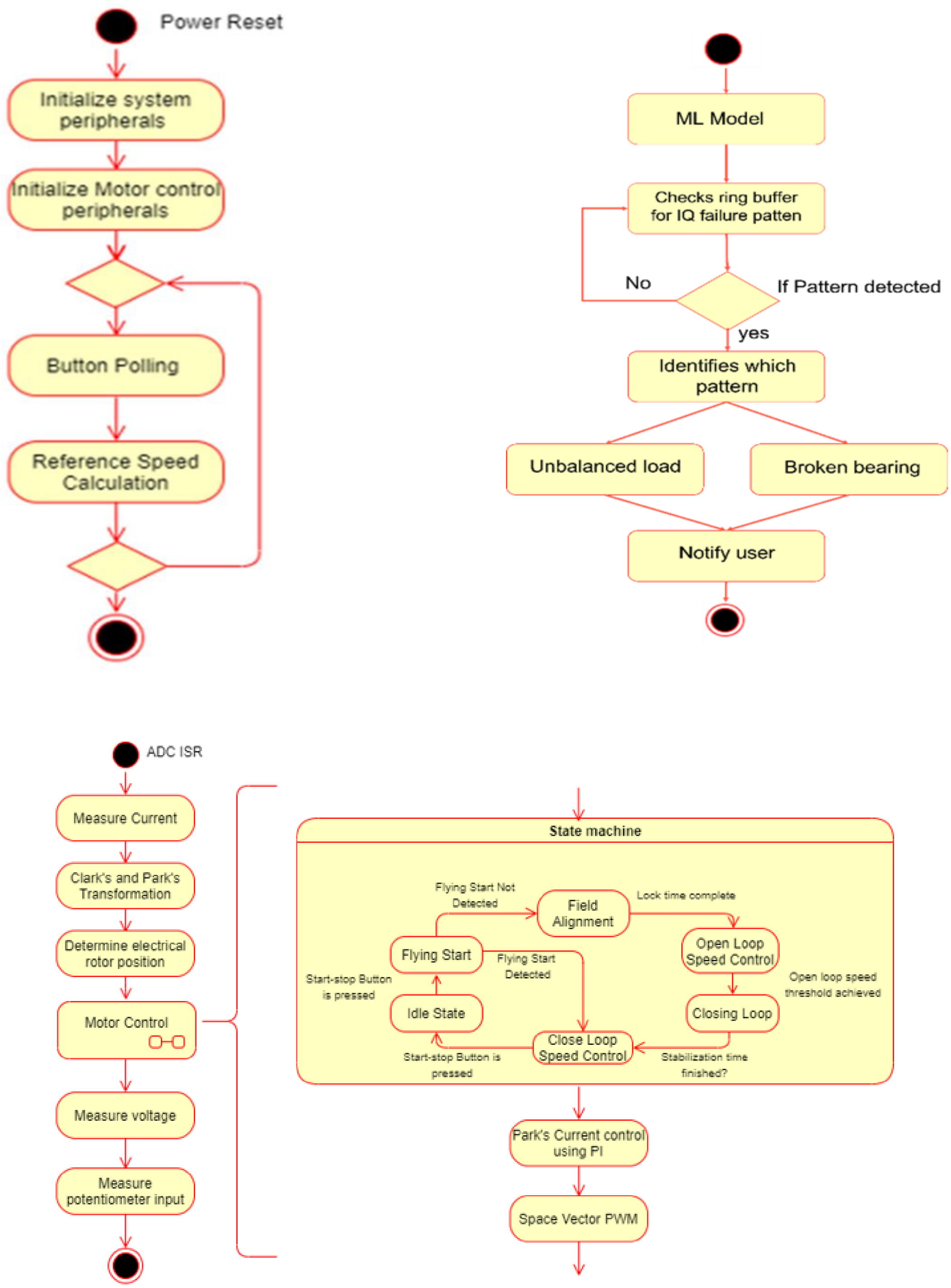


Figure 16: Application flow chart

8. Building Machine Learning Model

- Application used Supervised ML model.
- Application is implemented to detect unbalanced load on both forward and reverse direction of motor spinning direction.
- I_q current varies for both directions. Figure 17 shows I_q and I_d current for forward direction. Green color plot denotes I_q when motor spinning in the forward direction. Spikes occur when applying unbalanced load to the rotor.
- Similarly, figure 18 shows I_q and I_d current for reverse direction green color plot denotes I_q when motor running in reverse direction, where all the spikes occur when applying unbalanced load to the rotor.
- In this ML build has 4 labels as this is supervised ML model, names as below.
  - Normal_operation_forward
  - Normal_operation_reverse
  - Unbalanced_load_forward
  - Unbalanced_load_reverse
- Each signal is segmented with above mentioned labels, which can be seen in the right side of the image.

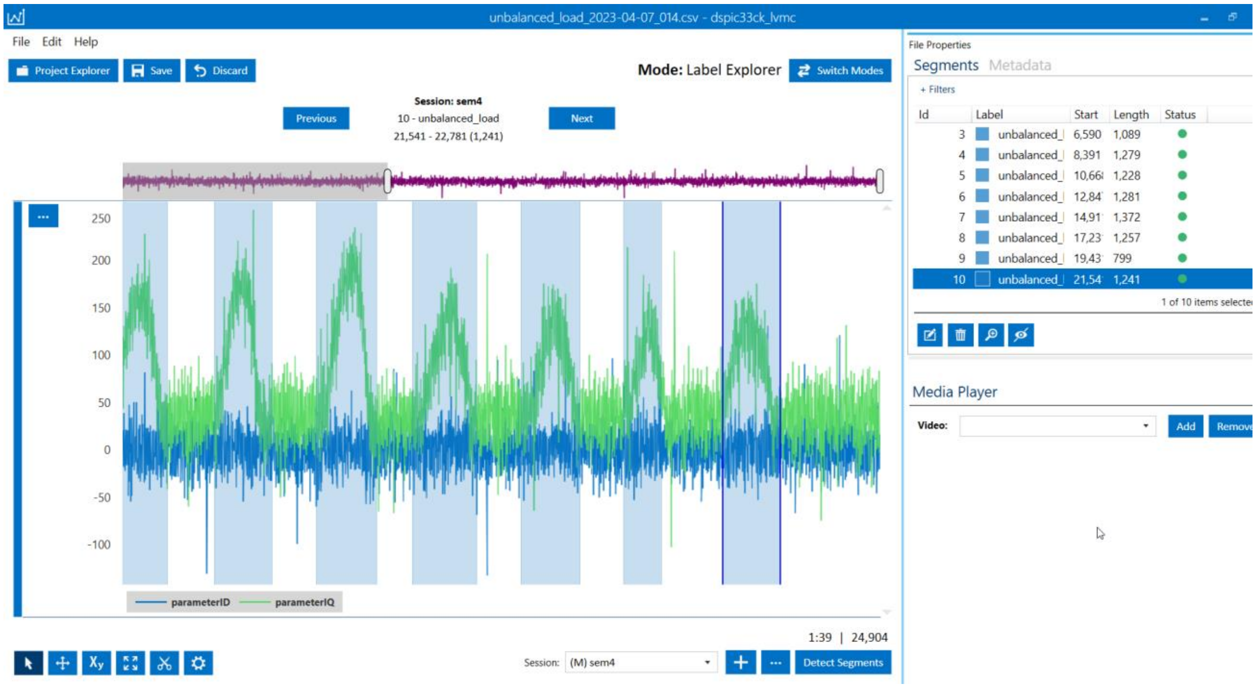


Figure 17: I_q value when motor forward direction

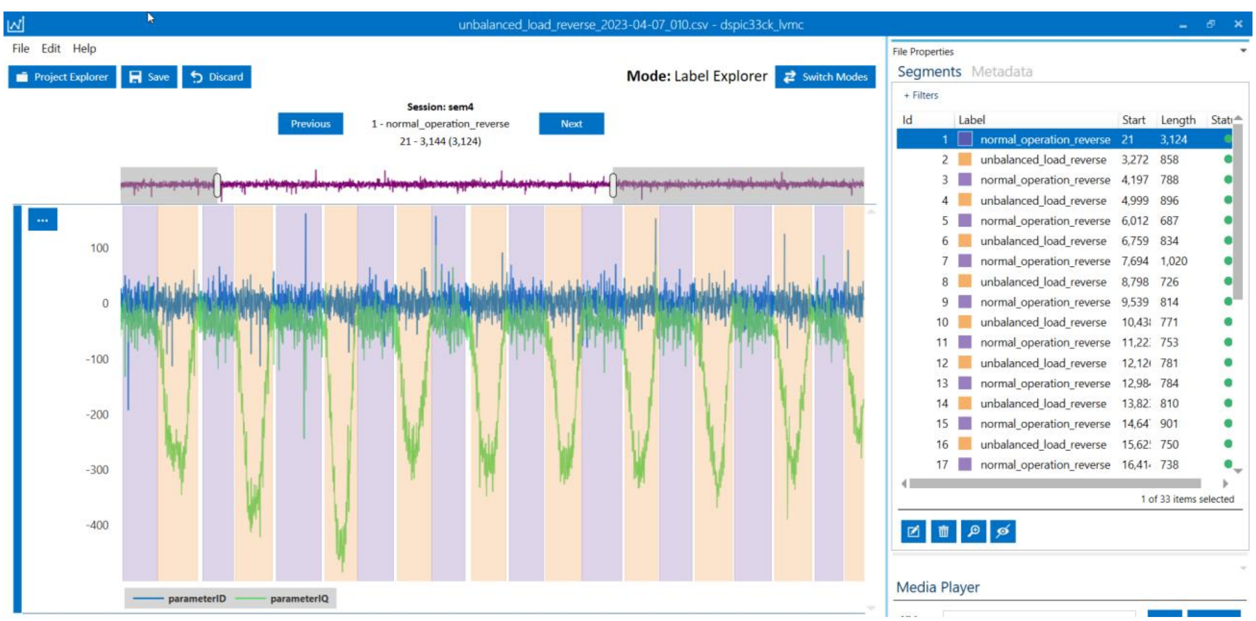
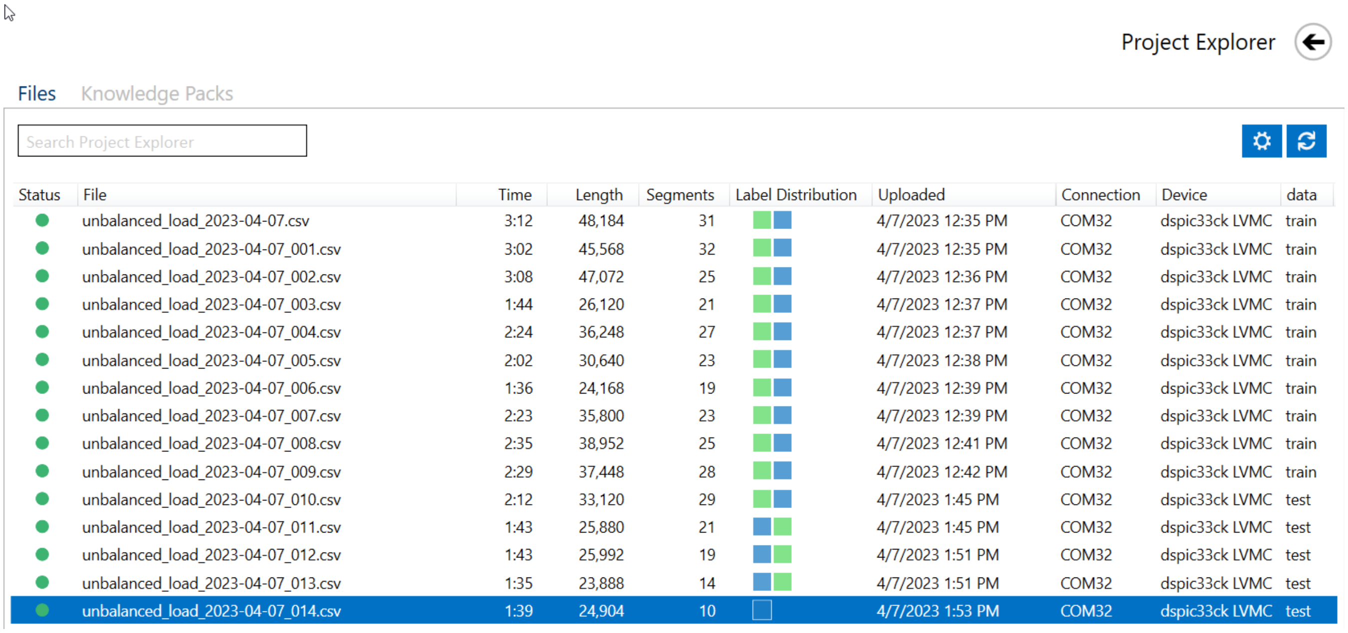


Figure 18: I_q value when motor reverse direction

- Below are datasets collected for both forward and reverse motor directions. No of segments for each signal is also listed.

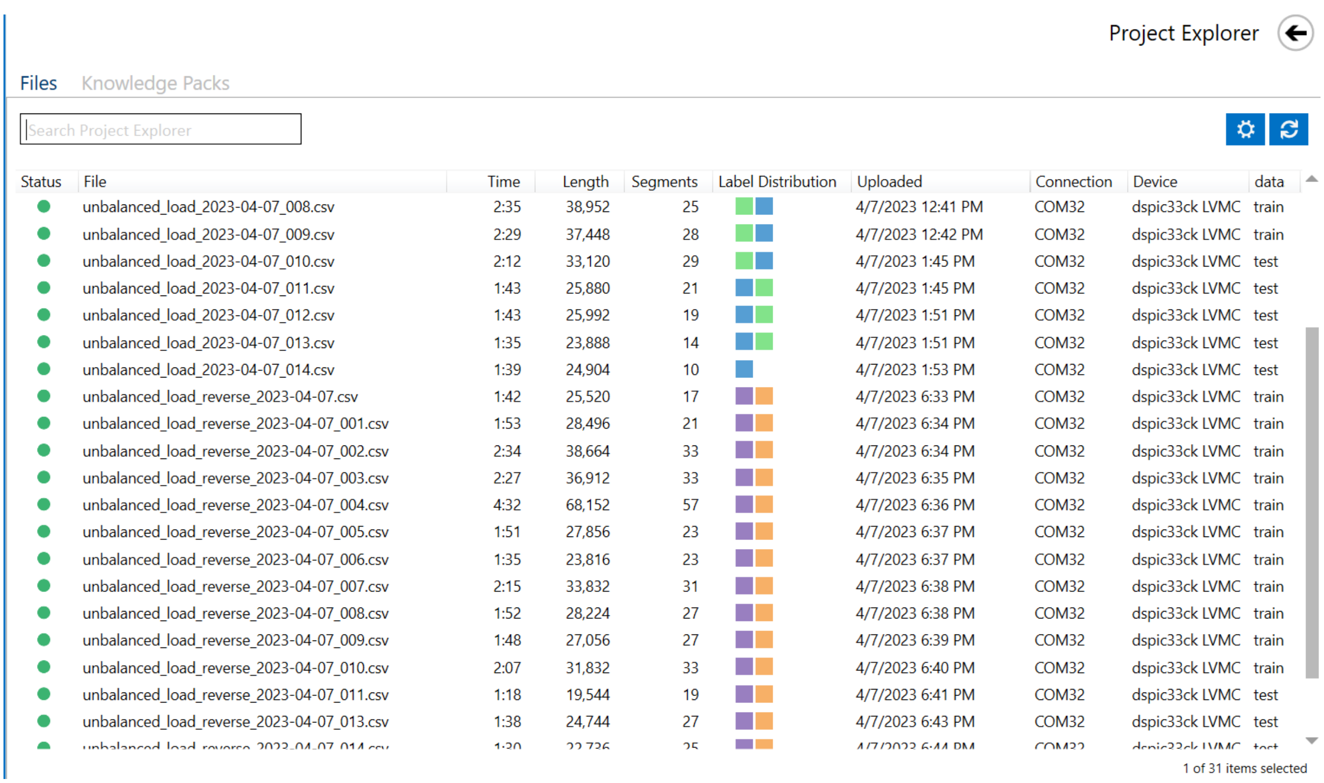


Project Explorer

Files Knowledge Packs

Search Project Explorer

Status	File	Time	Length	Segments	Label Distribution	Uploaded	Connection	Device	data
	unbalanced_load_2023-04-07.csv	3:12	48,184	31		4/7/2023 12:35 PM	COM32	dspic33ck LVMC	train
	unbalanced_load_2023-04-07_001.csv	3:02	45,568	32		4/7/2023 12:35 PM	COM32	dspic33ck LVMC	train
	unbalanced_load_2023-04-07_002.csv	3:08	47,072	25		4/7/2023 12:36 PM	COM32	dspic33ck LVMC	train
	unbalanced_load_2023-04-07_003.csv	1:44	26,120	21		4/7/2023 12:37 PM	COM32	dspic33ck LVMC	train
	unbalanced_load_2023-04-07_004.csv	2:24	36,248	27		4/7/2023 12:37 PM	COM32	dspic33ck LVMC	train
	unbalanced_load_2023-04-07_005.csv	2:02	30,640	23		4/7/2023 12:38 PM	COM32	dspic33ck LVMC	train
	unbalanced_load_2023-04-07_006.csv	1:36	24,168	19		4/7/2023 12:39 PM	COM32	dspic33ck LVMC	train
	unbalanced_load_2023-04-07_007.csv	2:23	35,800	23		4/7/2023 12:39 PM	COM32	dspic33ck LVMC	train
	unbalanced_load_2023-04-07_008.csv	2:35	38,952	25		4/7/2023 12:41 PM	COM32	dspic33ck LVMC	train
	unbalanced_load_2023-04-07_009.csv	2:29	37,448	28		4/7/2023 12:42 PM	COM32	dspic33ck LVMC	train
	unbalanced_load_2023-04-07_010.csv	2:12	33,120	29		4/7/2023 1:45 PM	COM32	dspic33ck LVMC	test
	unbalanced_load_2023-04-07_011.csv	1:43	25,880	21		4/7/2023 1:45 PM	COM32	dspic33ck LVMC	test
	unbalanced_load_2023-04-07_012.csv	1:43	25,992	19		4/7/2023 1:51 PM	COM32	dspic33ck LVMC	test
	unbalanced_load_2023-04-07_013.csv	1:35	23,888	14		4/7/2023 1:51 PM	COM32	dspic33ck LVMC	test
	unbalanced_load_2023-04-07_014.csv	1:39	24,904	10		4/7/2023 1:53 PM	COM32	dspic33ck LVMC	test



Project Explorer

Files Knowledge Packs

Search Project Explorer

Status	File	Time	Length	Segments	Label Distribution	Uploaded	Connection	Device	data
	unbalanced_load_2023-04-07_008.csv	2:35	38,952	25		4/7/2023 12:41 PM	COM32	dspic33ck LVMC	train
	unbalanced_load_2023-04-07_009.csv	2:29	37,448	28		4/7/2023 12:42 PM	COM32	dspic33ck LVMC	train
	unbalanced_load_2023-04-07_010.csv	2:12	33,120	29		4/7/2023 1:45 PM	COM32	dspic33ck LVMC	test
	unbalanced_load_2023-04-07_011.csv	1:43	25,880	21		4/7/2023 1:45 PM	COM32	dspic33ck LVMC	test
	unbalanced_load_2023-04-07_012.csv	1:43	25,992	19		4/7/2023 1:51 PM	COM32	dspic33ck LVMC	test
	unbalanced_load_2023-04-07_013.csv	1:35	23,888	14		4/7/2023 1:51 PM	COM32	dspic33ck LVMC	test
	unbalanced_load_2023-04-07_014.csv	1:39	24,904	10		4/7/2023 1:53 PM	COM32	dspic33ck LVMC	test
	unbalanced_load_reverse_2023-04-07.csv	1:42	25,520	17		4/7/2023 6:33 PM	COM32	dspic33ck LVMC	train
	unbalanced_load_reverse_2023-04-07_001.csv	1:53	28,496	21		4/7/2023 6:34 PM	COM32	dspic33ck LVMC	train
	unbalanced_load_reverse_2023-04-07_002.csv	2:34	38,664	33		4/7/2023 6:34 PM	COM32	dspic33ck LVMC	train
	unbalanced_load_reverse_2023-04-07_003.csv	2:27	36,912	33		4/7/2023 6:35 PM	COM32	dspic33ck LVMC	train
	unbalanced_load_reverse_2023-04-07_004.csv	4:32	68,152	57		4/7/2023 6:36 PM	COM32	dspic33ck LVMC	train
	unbalanced_load_reverse_2023-04-07_005.csv	1:51	27,856	23		4/7/2023 6:37 PM	COM32	dspic33ck LVMC	train
	unbalanced_load_reverse_2023-04-07_006.csv	1:35	23,816	23		4/7/2023 6:37 PM	COM32	dspic33ck LVMC	train
	unbalanced_load_reverse_2023-04-07_007.csv	2:15	33,832	31		4/7/2023 6:38 PM	COM32	dspic33ck LVMC	train
	unbalanced_load_reverse_2023-04-07_008.csv	1:52	28,224	27		4/7/2023 6:38 PM	COM32	dspic33ck LVMC	train
	unbalanced_load_reverse_2023-04-07_009.csv	1:48	27,056	27		4/7/2023 6:39 PM	COM32	dspic33ck LVMC	train
	unbalanced_load_reverse_2023-04-07_010.csv	2:07	31,832	33		4/7/2023 6:40 PM	COM32	dspic33ck LVMC	train
	unbalanced_load_reverse_2023-04-07_011.csv	1:18	19,544	19		4/7/2023 6:41 PM	COM32	dspic33ck LVMC	test
	unbalanced_load_reverse_2023-04-07_013.csv	1:38	24,744	27		4/7/2023 6:43 PM	COM32	dspic33ck LVMC	test
	unbalanced_load_reverse_2023-04-07_014.csv	1:30	22,736	25		4/7/2023 6:44 PM	COM32	dspic33ck LVMC	test

1 of 31 items selected

Figure 19: datasets

- Below images show meta data selected for building the model and filters data sets that labeled as train to train the model.

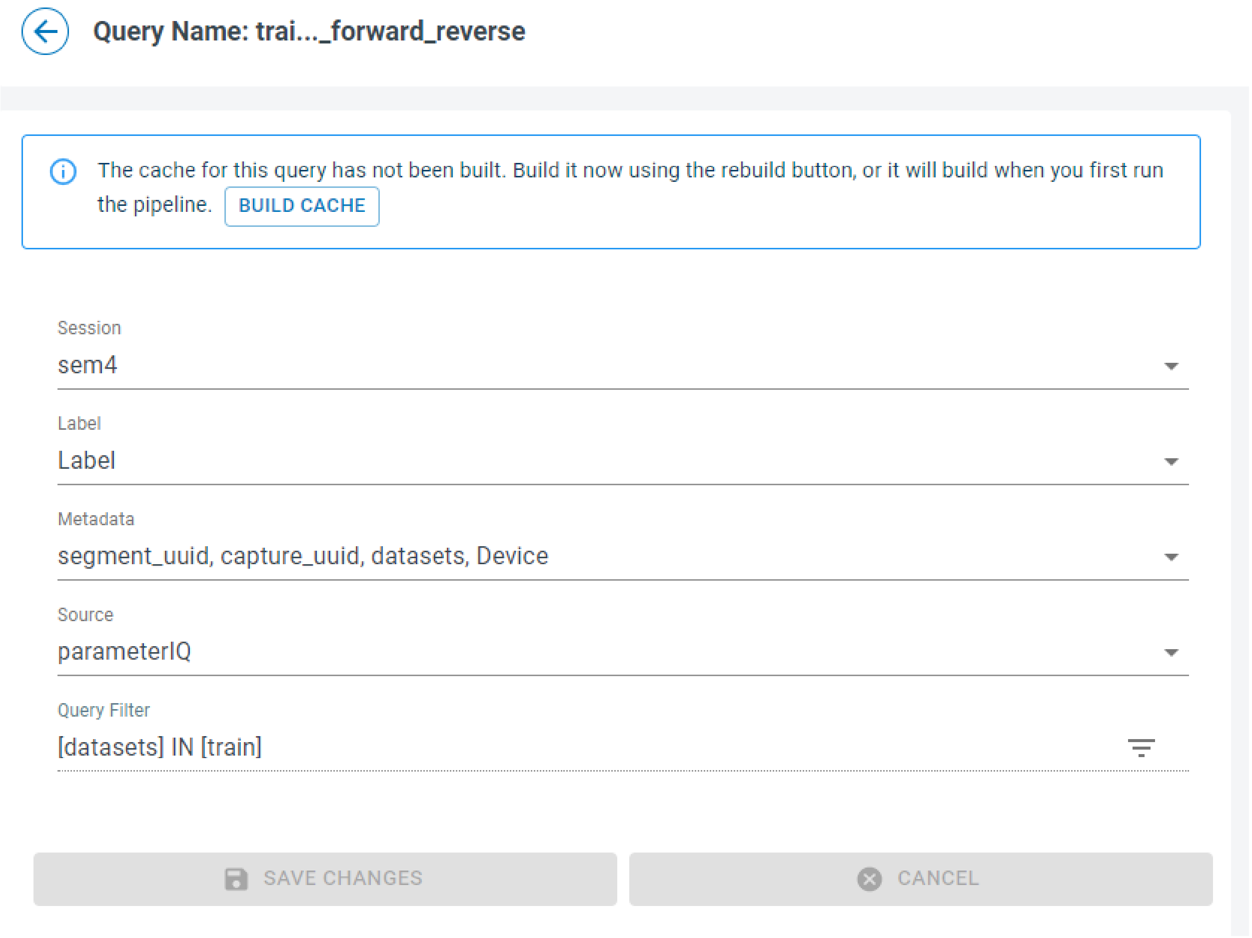


Figure 20: ML train config

- The following images show segment length distribution, display segment sample size variation for each label.

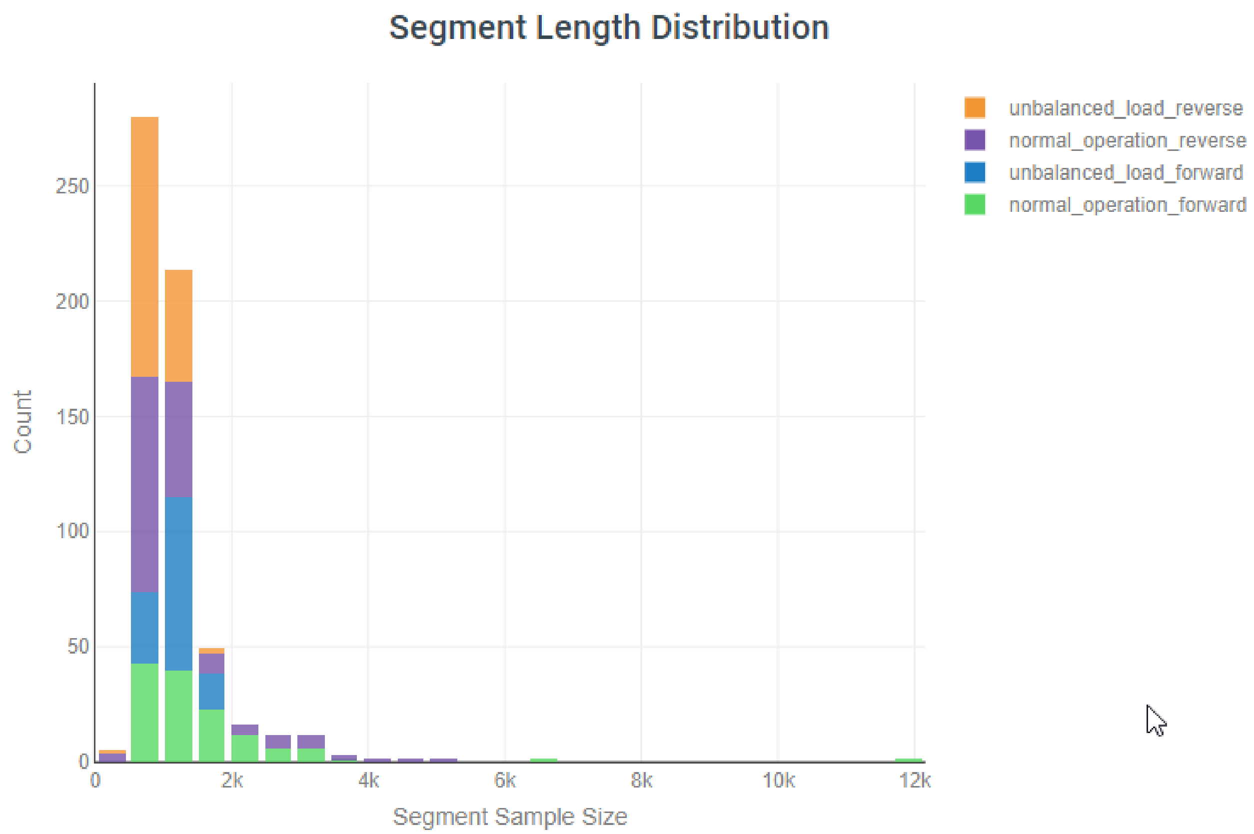


Figure 21:Segment length distribution

- Figure 22 displays the number of segments for each label and figure 23 shows how many samples each label contains.

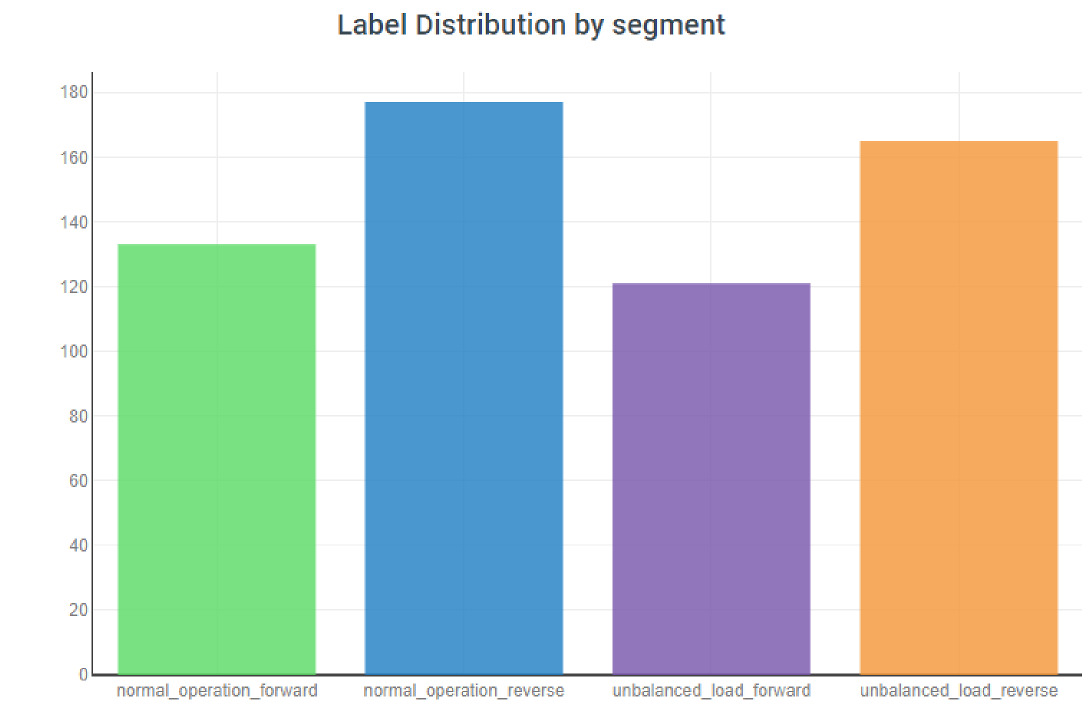


Figure 22:Label distribution by segment

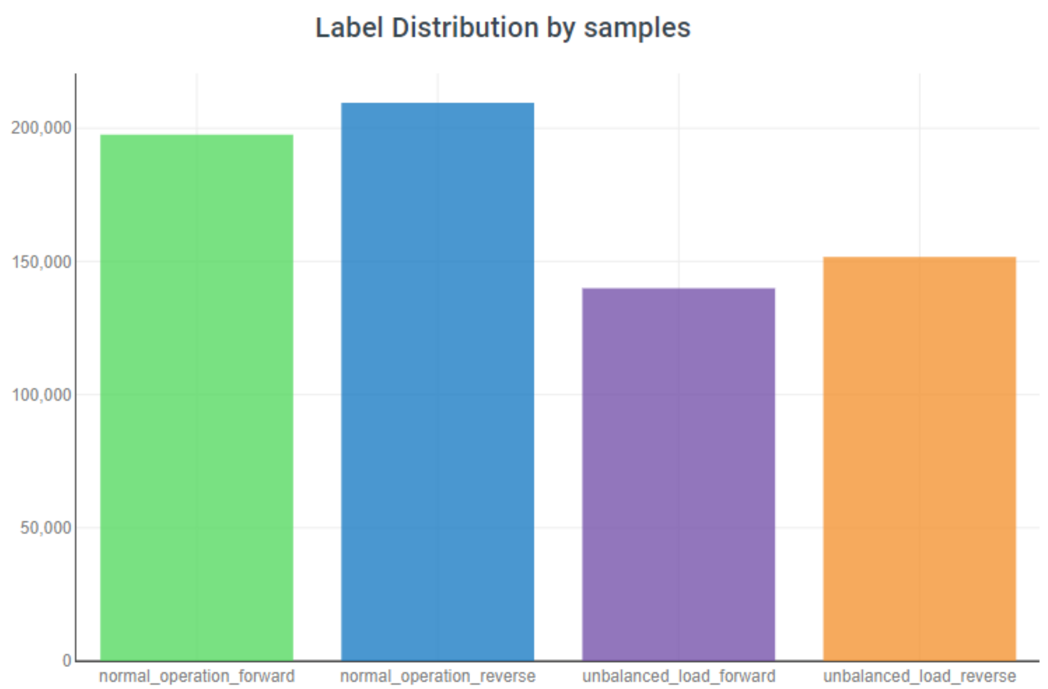


Figure 23: Label distribution by samples

• Flow chart shows pipeline configuration for the ML algorithm.

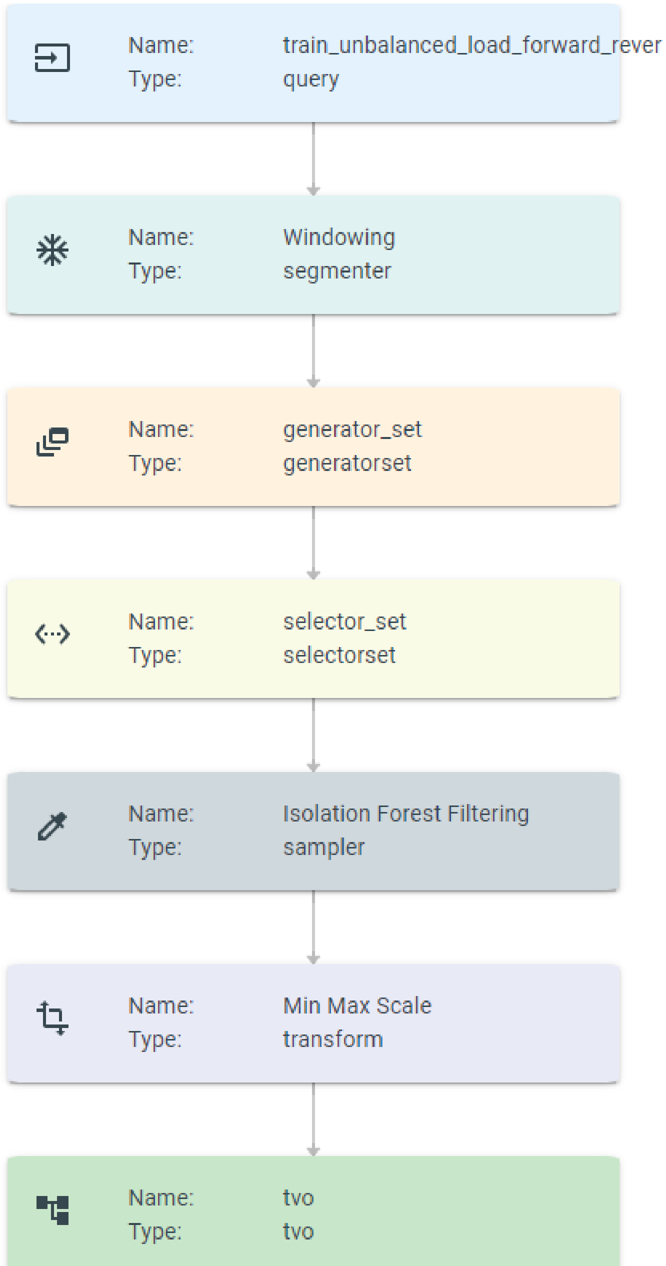


Figure 24: Pipeline configuration

- The data from the sensor transform/filter stage is fed into the segmenter, which buffers it until a segment is found. Windowing segmenter used to train data in this application.

  Parameters:

  - input_data (_type_) – _description_
  - group_columns (_type_) – _description_
  - window_size (_type_) – _description_
  - delta (_type_) – _description_
  - train_delta (int, optional) – _description_. Defaults to 0

  Return:

  - segment_index (bool, optional) – _description_. Defaults to False.

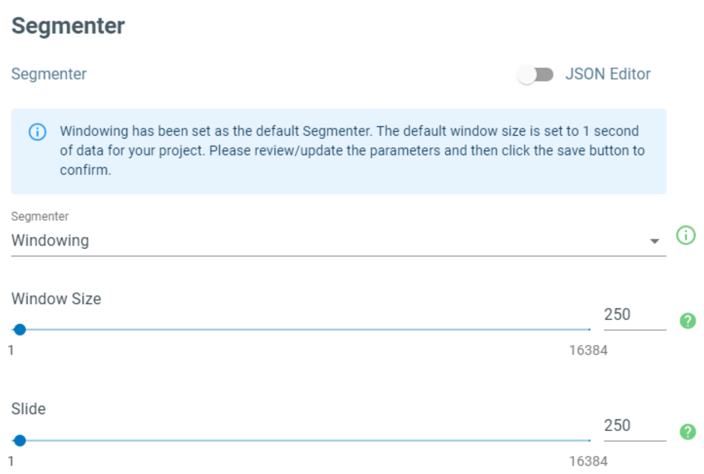


Figure 25: Segmenter configuration

- Choosing the accuracy optimization metric in this case would cause the model optimization to be biased towards the classes with more samples because of the imbalance in the dataset's class distribution; instead, choose the f1-score to provide a more accurate representation of the model performance.

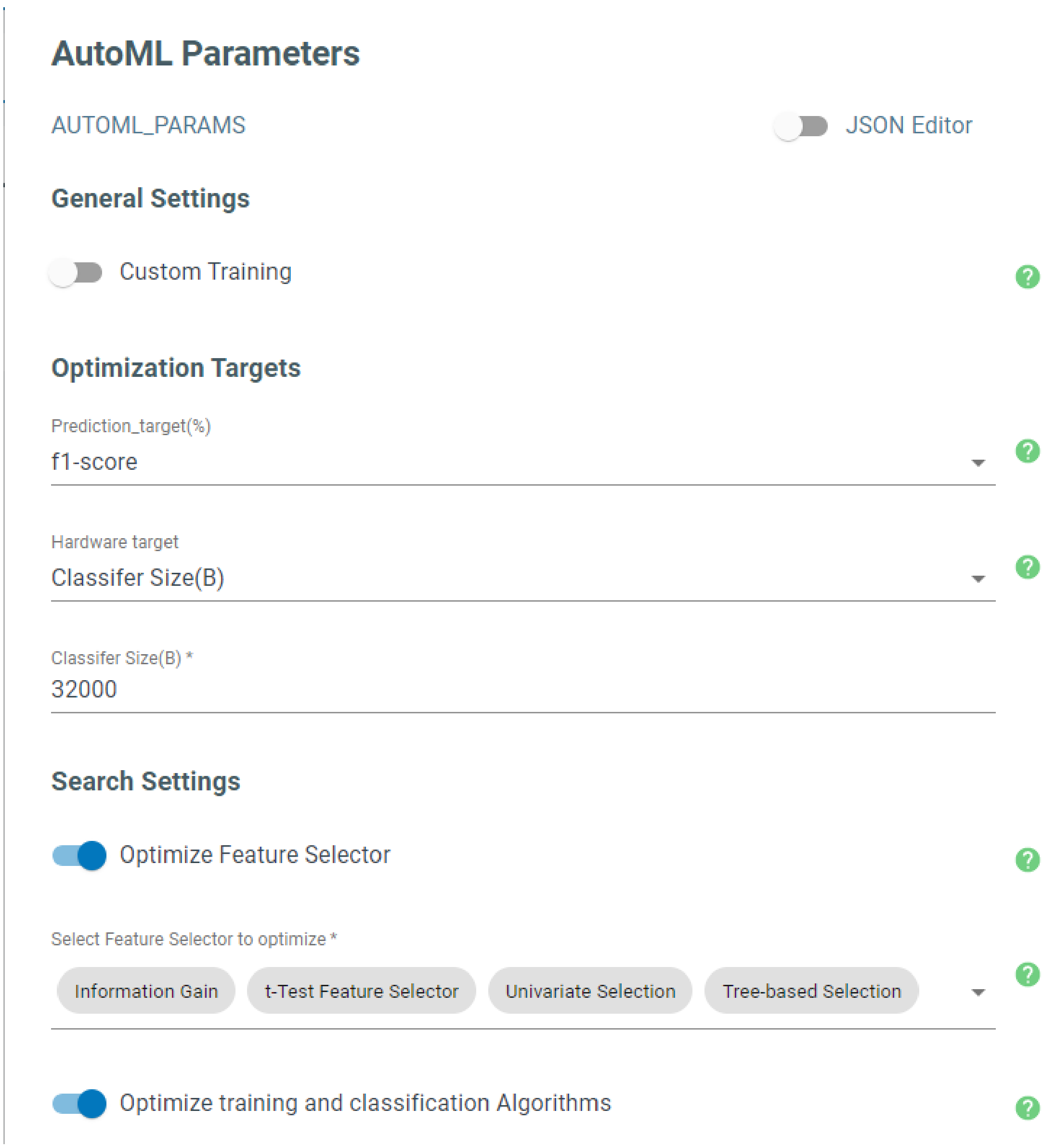


Figure 26: Segmenter configuration

9. ML model statistics

- Experimental with different ML algorithms names PME, Decision Tree Ensemble, Tensor flow , Bonsai, Boosted tree enable.
- As shown in the below image **Decision Tree Ensemble** Provides best solution with accuracy of **98.10%**
- Next PME produces the next best solution with 97.89 % accuracy.
- The best model generated has a size of **3746 Bytes**.

Select model

NAME	CLASSIFIER	ACCURACY	MODEL SIZE (BYTES)	FEATURE COUNT	PIPELINE	CREATED DATE	UUID	EXPLORE	TEST	DOWNLOAD	RENAME	DELETE
pipeline_forward_reverse_rank_4	PME	95.98 %	2149	12	pipeline_forward_reverse	4/7/2023 8:25:19 PM	ded5f3e9-7d00-440b-bf97-b823c0899842					
pipeline_forward_reverse_rank_3	Decision Tree Ensemble	96.62 %	6716	1	pipeline_forward_reverse	4/7/2023 8:25:19 PM	8783e69d-e0b0-4ed5-83d8-fbb2c5e7393f					
pipeline_forward_reverse_rank_2	PME	97.25 %	1468	6	pipeline_forward_reverse	4/7/2023 8:25:18 PM	a50965b2-9985-48ca-af01-71f113d41063					
pipeline_forward_reverse_rank_1	PME	97.89 %	1264	4	pipeline_forward_reverse	4/7/2023 8:25:18 PM	3570ce6a-787e-4baa-bd05-6938051cb058					
pipeline_forward_reverse_rank_0	Decision Tree Ensemble	98.10 %	3746	1	pipeline_forward_reverse	4/7/2023 8:25:18 PM	3e82b35a-542e-4f9d-9da9-a8f0cf2ec8fd					

Figure 27: ML results with accuracy and size

- Exploration on the best generated model named **"pipeline_forward_reverse_rank_0"** has been done.
- **Confusion Matrix** compares the predicted labels (columns) and true labels (rows) in the classification results. The Precision score (true positive predictions / total positive predictions) is displayed in the bottom-most row, and the Sensitivity (or Recall) score (true positive predictions / total true positives) is displayed in the right-most column for each class.

Model: pipeline_forward_reverse_rank_0

CONFUSION MATRIX FEATURE VISUALIZATION FEATURE SUMMARY MODEL SUMMARY PIPELINE SUMMARY KNOWLEDGE PACK SUMMARY

Validation

	normal_operation_forward	normal_operation_reverse	unbalanced_load_forward	unbalanced_load_reverse	UNK	Support	Sense %
normal_operation_forward	138	0	0	0	0	138.00	100.00
normal_operation_reverse	0	142	0	0	0	142.00	100.00
unbalanced_load_forward	9	0	85	0	0	94.00	90.43
unbalanced_load_reverse	0	0	0	100	0	100.00	100.00
Total	147.00	142.00	85.00	100.00	0	474.00	
Pos_Pred(%)	93.88	100.00	100.00	100.00		Acc(%)	98.10

Training

	normal_operation_forward	normal_operation_reverse	unbalanced_load_forward	unbalanced_load_reverse	UNK	Support	Sense %
normal_operation_forward	689	0	0	0	0	689.00	100.00
normal_operation_reverse	0	706	0	0	0	706.00	100.00
unbalanced_load_forward	57	0	414	0	0	471.00	87.90
unbalanced_load_reverse	0	5	0	495	0	500.00	99.00
Total	746.00	711.00	414.00	495.00	0	2366.00	
Pos_Pred(%)	92.36	99.30	100.00	100.00		Acc(%)	97.38

Fold 0 - validation

	normal_operation_forward	normal_operation_reverse	unbalanced_load_forward	unbalanced_load_reverse	UNK	Support	Sense %
normal_operation_forward	138	0	0	0	0	138.00	100.00
normal_operation_reverse	0	142	0	0	0	142.00	100.00
unbalanced_load_forward	9	0	85	0	0	94.00	90.43
unbalanced_load_reverse	0	0	0	100	0	100.00	100.00
Total	147.00	142.00	85.00	100.00	0	474.00	
Pos_Pred(%)	93.88	100.00	100.00	100.00		Acc(%)	98.10

Fold 0 - train

	normal_operation_forward	normal_operation_reverse	unbalanced_load_forward	unbalanced_load_reverse	UNK	Support	Sense %
normal_operation_forward	551	0	0	0	0	551.00	100.00
normal_operation_reverse	0	564	0	0	0	564.00	100.00
unbalanced_load_forward	48	0	329	0	0	377.00	87.27
unbalanced_load_reverse	0	5	0	395	0	400.00	98.75
Total	599.00	569.00	329.00	395.00	0	1892.00	
Pos_Pred(%)	91.99	99.12	100.00	100.00		Acc(%)	97.20

Figure 28: confusion matrix exploration

10. ML Model Testing

After building and exploring the model, the test has been done. Below image shows list of test files that were captured from sensiML. Test files were filtered from captures files pool and used for testing.

Project: dspic33ck_lvmc   mahalakshmiper

...	CAPTURE NAME	ACCURACY	CREATED	TOTAL EVENT COUNT	CAPTURE UUID	RESULTS	SIZE (MB)	CONNECTION	DATASETS	DEVICE	PIPELINEID
									test		
☑	unbalanced_load_2023-04-07_012.csv		2023-04-07T08:21:06.202Z	19	e584bb0b-b9bf-49f1-ad71-abb5f171a9c0		0.33	COM32	test	dspic33ck LVMC	
☑	unbalanced_load_reverse_2023-04-07_011.csv		2023-04-07T13:11:14.910Z	19	095f1e31-0c65-4376-8486-f9a85e949626		0.26	COM32	test	dspic33ck LVMC	
☑	unbalanced_load_reverse_2023-04-07_013.csv		2023-04-07T13:13:29.370Z	27	7e0d8d99-47a3-45cf-a141-376e935ed2bc		0.33	COM32	test	dspic33ck LVMC	
☑	unbalanced_load_2023-04-07_011.csv		2023-04-07T08:15:54.576Z	21	c6100549-35ea-48a0-949e-6363f35efbf1		0.32	COM32	test	dspic33ck LVMC	
☑	unbalanced_load_reverse_2023-04-07_014.csv		2023-04-07T13:14:12.924Z	25	b2d68acd-e315-4503-8d50-06824cc9d24c		0.30	COM32	test	dspic33ck LVMC	
☑	unbalanced_load_2023-04-07_013.csv		2023-04-07T08:21:37.424Z	14	b9238c68-2827-4cc4-b56b-391e20b5969f		0.30	COM32	test	dspic33ck LVMC	
☑	unbalanced_load_2023-04-07_010.csv		2023-04-07T08:15:32.171Z	29	adff90f9-273a-4e34-9bc4-e70aa5372f76		0.41	COM32	test	dspic33ck LVMC	
☑	unbalanced_load_reverse_2023-04-07_015.csv		2023-04-07T13:14:57.977Z	19	22eeff7b-f109-4281-bbde-816ab0f22501		0.30	COM32	test	dspic33ck LVMC	
☑	unbalanced_load_2023-04-07_014.csv		2023-04-07T08:23:04.896Z	10	30cfc4f8-1eb2-47da-bca5-bccf14faa419		0.31	COM32	test	dspic33ck LVMC	

Figure 29: Captured test datasets

Result of testing datasets against the model is shown below. All the files have higher test accuracy varied from 90% to 98%. This way we can confirm accuracy of the built ML model.

Captures

...	CAPTURE NAME	ACCURACY	CREATED	TOTAL EVENT COUNT	CAPTURE UUID	RESULTS	SIZE (MB)	CONNECTION	DATASETS	DEVICE	PIPELINEID	MODELID
									test			
☑	unbalanced_load_reverse_2023-04-07_011.csv	98.7	2023-04-07T13:11:14.910Z	19	095f1e31-0c65-4376-8486-f9a85e949626	RESULTS	0.26	COM32	test	dspic33ck LVMC		
☑	unbalanced_load_2023-04-07_012.csv	92.2	2023-04-07T08:21:06.202Z	19	e584bb0b-b9bf-49f1-ad71-abb5f171a9c0	RESULTS	0.33	COM32	test	dspic33ck LVMC		
☑	unbalanced_load_reverse_2023-04-07_015.csv	92	2023-04-07T13:14:57.977Z	19	22eeff7b-f109-4281-bbde-816ab0f22501	RESULTS	0.30	COM32	test	dspic33ck LVMC		
☑	unbalanced_load_reverse_2023-04-07_013.csv	91.7	2023-04-07T13:13:29.370Z	27	7e0d8d99-47a3-45cf-a141-376e935ed2bc	RESULTS	0.33	COM32	test	dspic33ck LVMC		
☑	unbalanced_load_2023-04-07_010.csv	91.7	2023-04-07T08:15:32.171Z	29	adff90f9-273a-4e34-9bc4-e70aa5372f76	RESULTS	0.41	COM32	test	dspic33ck LVMC		
☑	unbalanced_load_reverse_2023-04-07_014.csv	89.7	2023-04-07T13:14:12.924Z	25	b2d68acd-e315-4503-8d50-06824cc9d24c	RESULTS	0.30	COM32	test	dspic33ck LVMC		
☑	unbalanced_load_2023-04-07_011.csv	87.6	2023-04-07T08:15:54.576Z	21	c6100549-35ea-48a0-949e-6363f35efbf1	RESULTS	0.32	COM32	test	dspic33ck LVMC		

Figure 30: ML model test accuracy

11. Downloading ML model

ML Model can download as knowledge back from sensiML software. A Knowledge Pack takes the event detection model generated in pipeline and transforms it into a file that can be run on hardware device at the edge. Once the Knowledge Pack is on the device, it starts outputting classification IDs that correspond to events of interest.

Below are steps to be followed:

- Target device is dsPIC33CK256MP508, which is 16-bit device. XC16 is the supported compiler from the list, which can be used with MPLAB X IDE to compile the project.

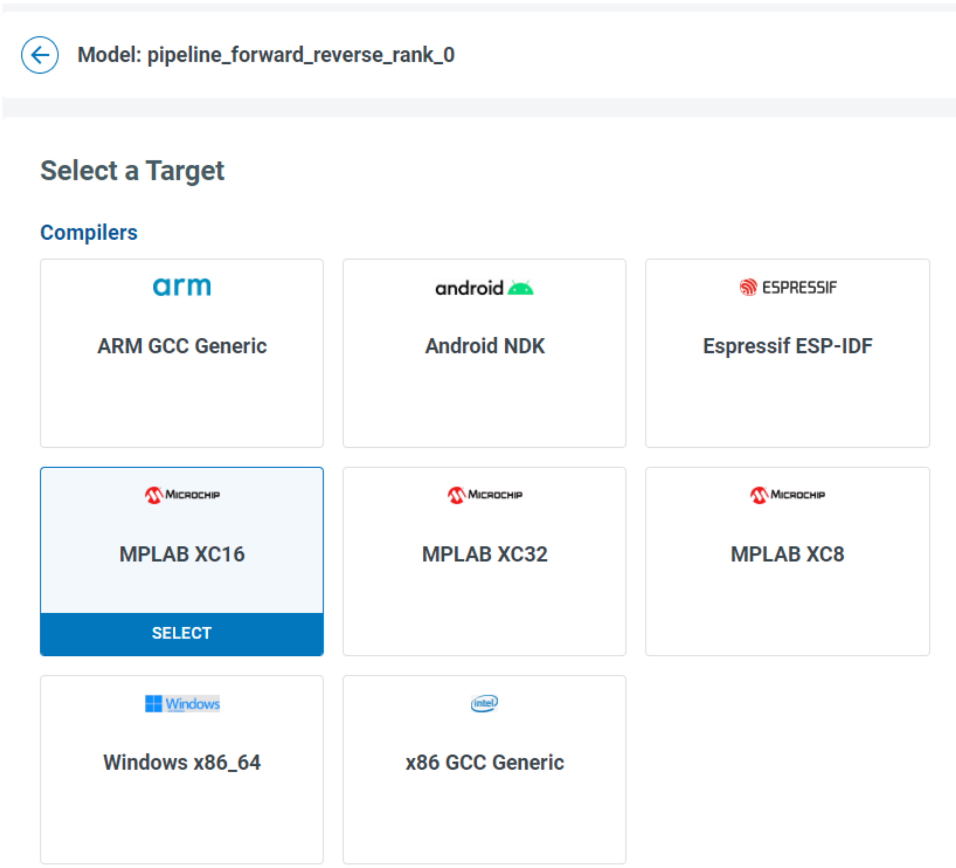


Figure 31: Compiler selection

- Configuring **Knowledge Pack** settings using the **Pipeline**, **Model**, and **Data Source** created in the previous steps, and selecting the **library** output format.
- The library format, available to all SensiML subscription tiers, will generate a pre-compiled library for the generated machine learning model, along with a header file defining the user API.
- Selecting target device as “dsPIC33CK256MP508” from the pool of devices

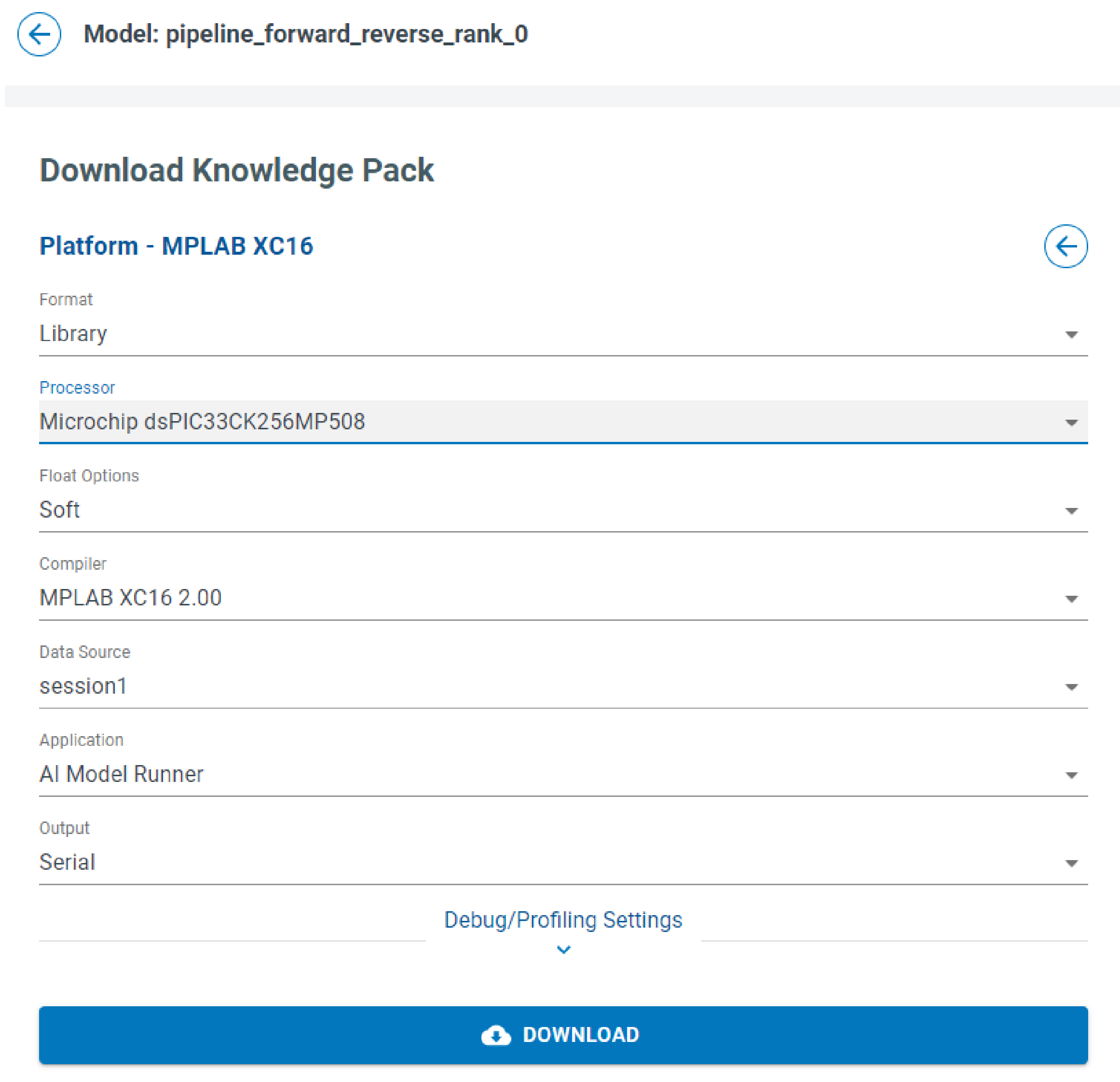


Figure 32: Knowledge pack setting

- Knowledge pack summary and estimations are listed below. Summary includes compiler platform, target device, plugin, sample rate, data source name and importantly memory usage estimation and latency.
- Can observe that the required SRAM would be 1368 Bytes, stack size is 852 Bytes and Flash size is 6554 Bytes.

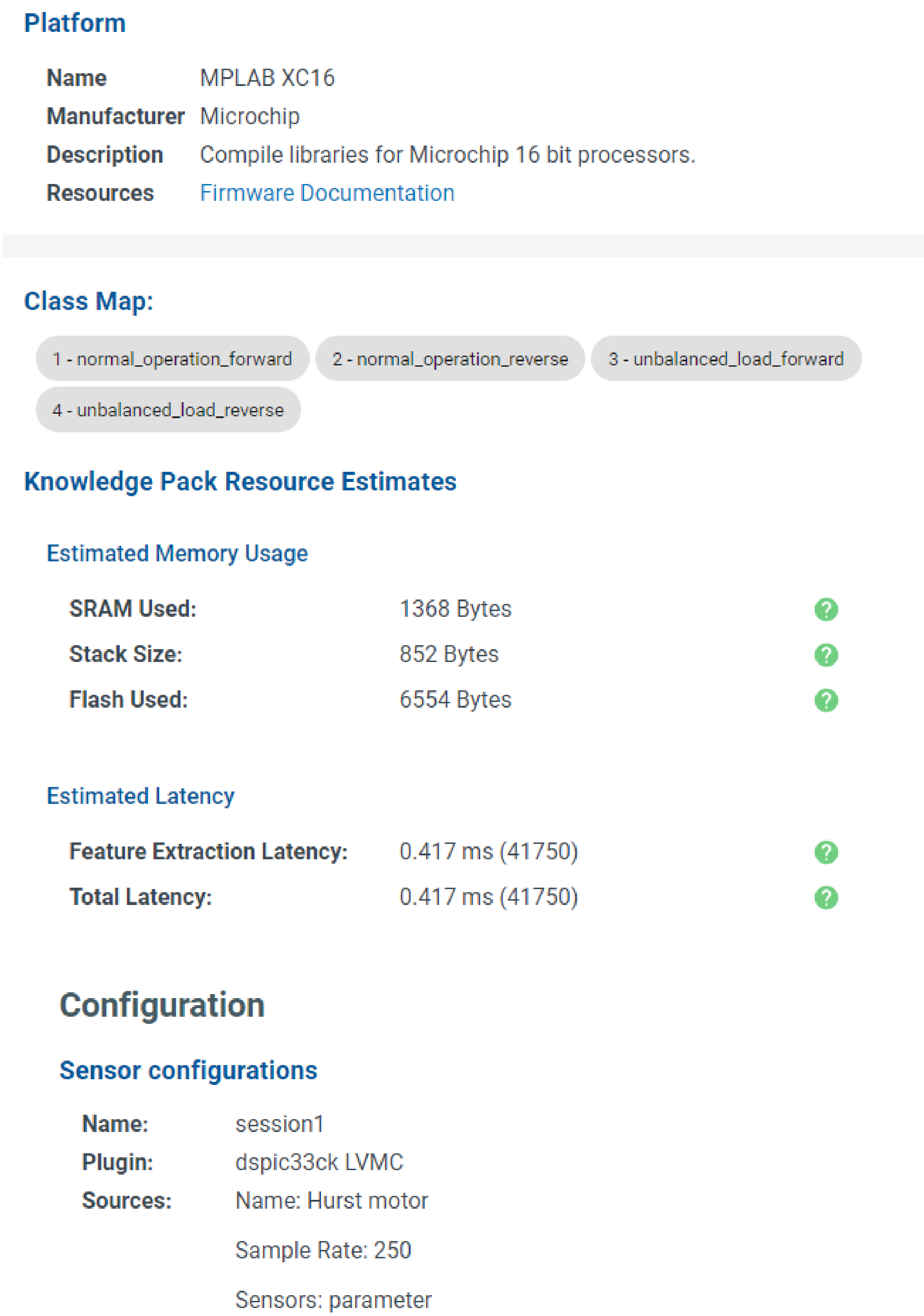


Figure 33: Knowledge pack summary

12. Application Results

Live testing on application with model has been done and observed expected results. Application tests were performed multiple times to confirm ML model working and tested to confirm all four functions monitoring.

a. Normal operation forward:

When the motor is running in a forward direction and no anomalies were observed. As shown in the image ML correctly detects the use case.

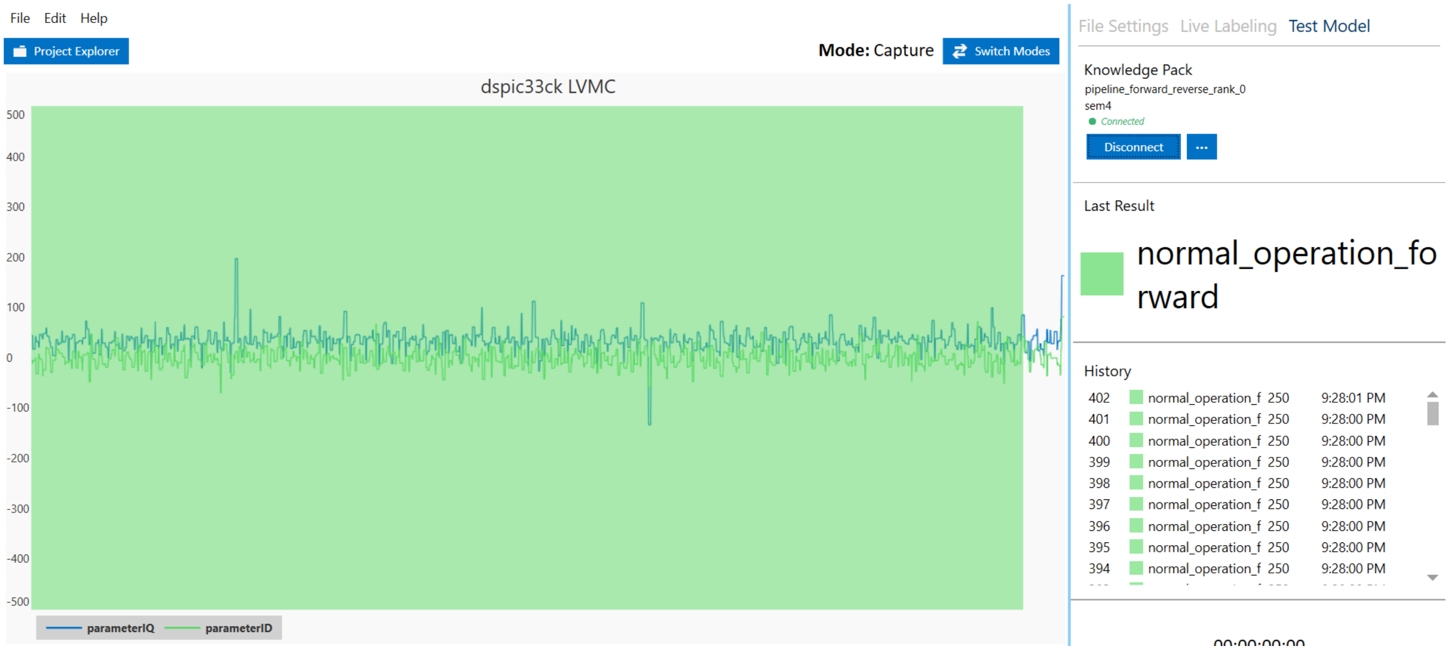


Figure 34: Normal_operation_Forward

b. Unbalanced load forward:

When the motor is running in a forward direction and unbalanced load were observed. As shown in the image ML correctly detects the unbalanced load. Since unbalanced load were applied for brief period, one can observe from below image that model detected that use case, displayed it.

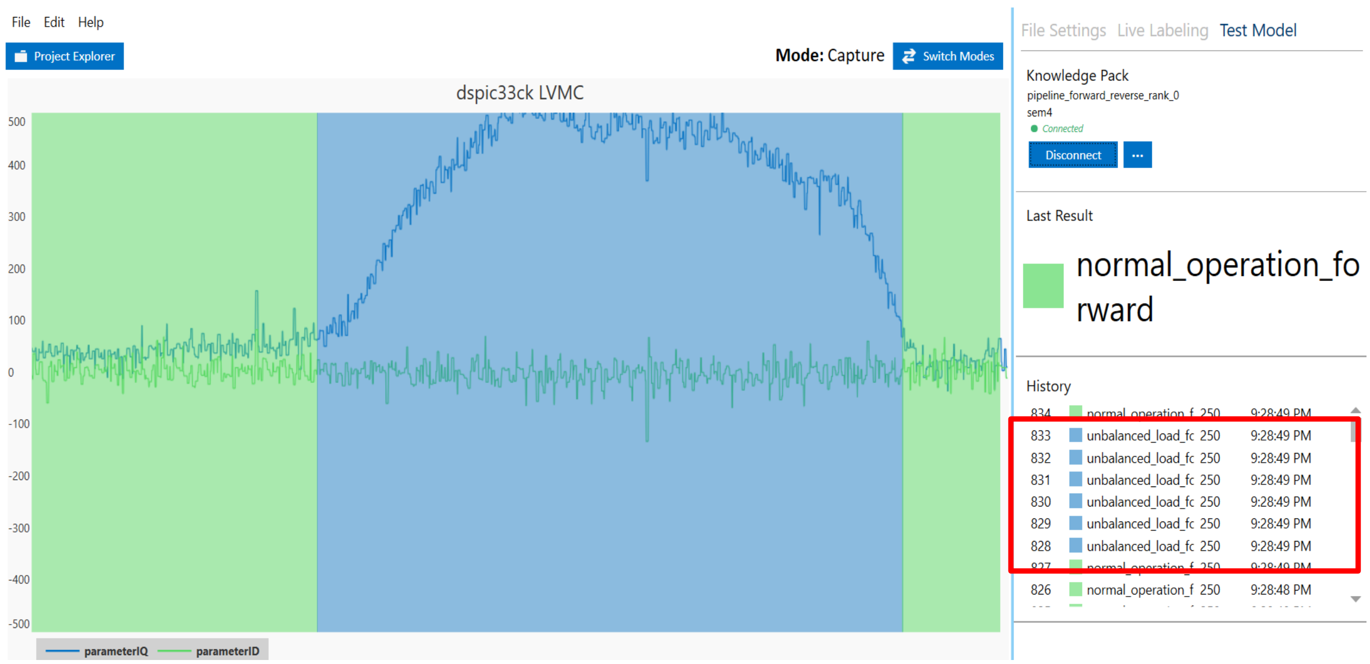


Figure 35: unbalanced_load_Forward

c. Normal_operation_reverse:

When the motor is running in a reverse direction and normal operation were observed. As shown in the image ML correctly detects that the motor running smoothly in reverse direction.

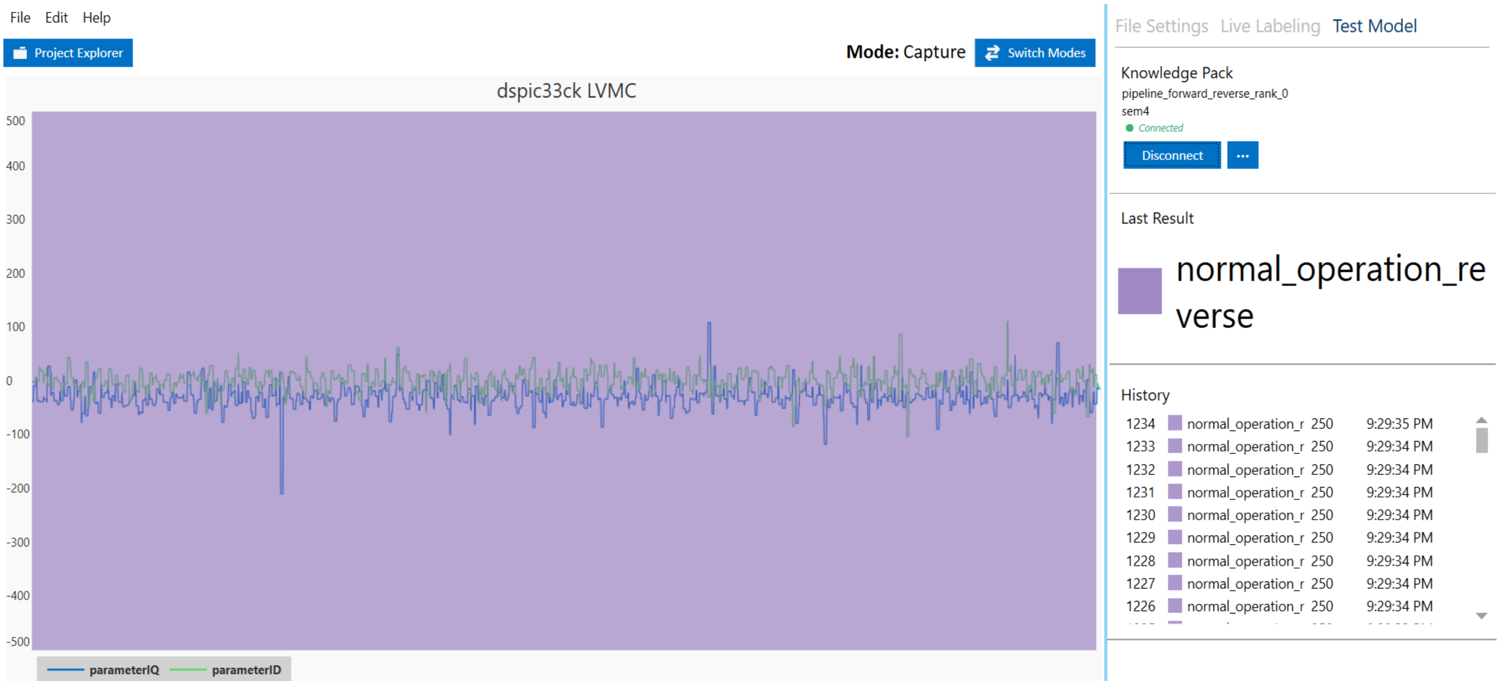


Figure 35: Normal_operation_reverse

d. Unbalanced_load_reverse:

When the motor is running in a reverse direction and an unbalanced load is induced at the rotor. As shown in the image ML correctly detects the unbalanced load and displays it.

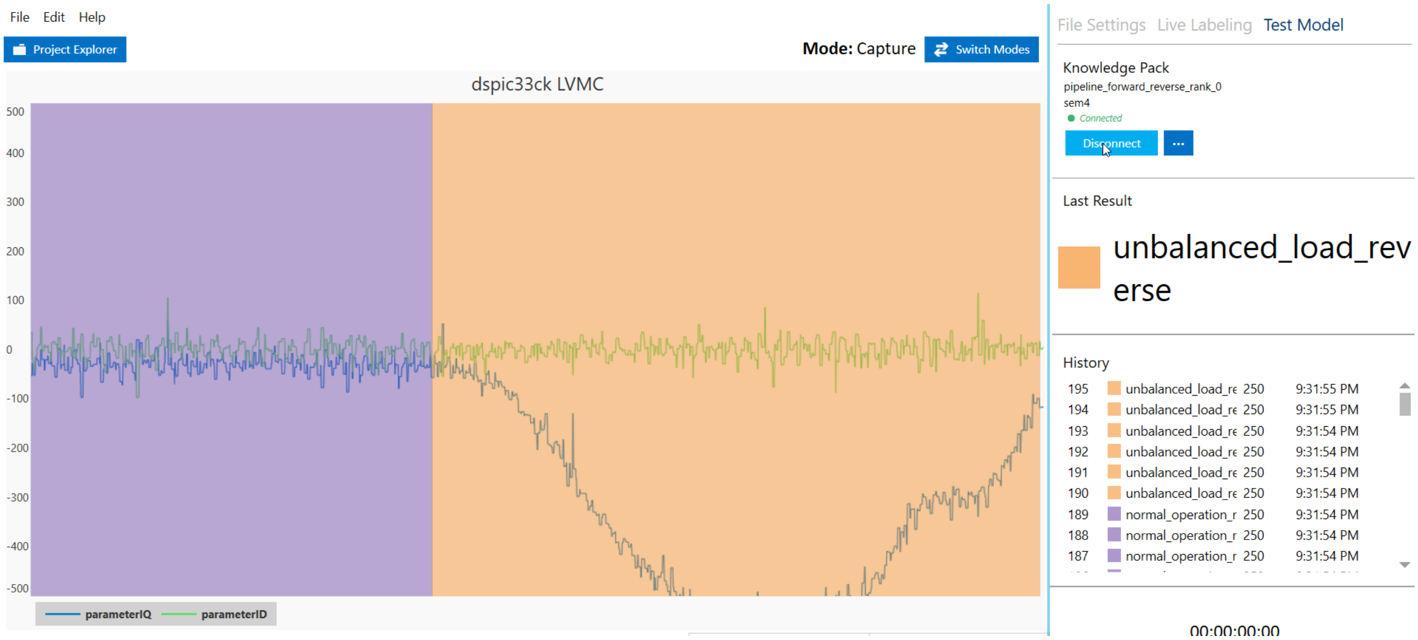


Figure 36: Unbalanced_load_reverse

13. Conclusion

A cost-effective predictive maintenance application to detect anomalies in the industrial motors would be a great addition to motor industries maintenance. We studied of Filed Orientation Algorithms and how to implement it on industrial motors with detailed peripheral configuration and block diagram. We examined different Machine Learning Algorithms, selected the best ML model algorithm that is Decision tree ensemble, based on accuracy. Finally, we discussed how to download model as knowledge pack from sensiML and tested application and shown results.

14. Abbreviations

Table 2: Abbreviations

Abbreviations	
FoC	Field Orientation Control
BLDC	Brushless DC
MCLV-2	Motor Control Low Voltage version 2
LVMC	Low Voltage Motor Control
UART	Universal Asynchronous Receiver Transmitter
TCC	Timer/Counter for Control Applications
ADC	Analog-to-Digital Converter
PDEC	Position Decoder
EIC	External Interrupt Controller
QDEC	Quadrature Decoder
EVSYS	Event System
SERCOM	Serial Communication
EMF	Electromagnetic Field
PIM	Plug-in Module
MCU	Microcontroller Unit
PMSM	Permanent Magnet Synchronous Motors
PWM	Pulse Width Modulation

15. References

[1] J. Brownlee, Master machine learning algorithms: discover how they work and implement them from scratch, 2016.

[2] B. Lu, D. B. Durocher, and P. Stemper, "Predictive maintenance techniques," IEEE Industry Applications Magazine, vol. 15, no. 6, 2019

[3] Evaluation of Machine Learning for Sensorless Detection and Classification of Faults in Electromechanical Drive Systems, October 2020Procedia Computer Science 176DOI:10.1016/j.procs.2020.09.170

[4] Krishna, Merugu Siva Rama and Ravi, Kiran S. "Fault diagnosis of induction motor using Motor Current Signature Analysis". In: 2013 International Conference on Circuits, Power and Computing Technologies (ICCPCT). Mar. 2013, pp. 180–186. DOI: 10.1109/ICCPCT.2013.6528849.

[5] Susto, Gian Antonio, Schirru, Andrea, Pampuri, Simone, Mcloone, Sean, and Beghi, Alessandro. "Machine Learning for Predictive Maintenance: A Multiple Classifier Approach". In: Industrial Informatics, IEEE Transactions on 11 (June 2015), pp. 812–820. DOI: 10.1109/TII.2014.2349359.

[6] SensiML Toolkit Documentation from SensiML website: https://sensiml.com/documentation/

[7] Sensorless Field Oriented Control of a PMSM Application note from Microchip: https://ww1.microchip.com/downloads/aemDocuments/documents/OTH/ApplicationNotes/ApplicationNotes/01078B.pdf.

16 . Glossary

1. Algorithm: A procedure used for solving a problem by performing a computation.
2. EMF: Electromotive force (EMF) is equal to the terminal potential difference when no current flows. EMF unit is volts.
3. Operating Frequency: Clock frequency at which target device CPU runs. Application speed depends on this.
4. Baud rate: At which rate data is transferred in a communication peripheral.
5. Encoder: Encoder is a sensing device that provides feedback. Encoders transform motion into an electrical signal that a control component, such a counter, can read in a motion control system. A feedback signal from the encoder can be used to calculate position, count, speed, or direction. BLDC Motor: Brushless DC motor with permanent magnet.
6. Phase Currents: Currently measured on each of the motor phases.
7. Kirchhoff's current law: Kirchhoff's Current Law states that" the algebraic sum of all the currents at any node point or a junction of a circuit is zero". That is $\Sigma I = 0$
8. TCC: Timer counter peripheral available in Microcontroller
9. ADC: Analog to digital converter module in Microcontroller that used to convert real life signal to digital form to process that in MCUs.
10. EVSYS: Event system is a core independent peripheral that is available in MCUs , can be used to reduce CPU load.

Checklist of Items for the Final Dissertation / Project / Project Work Report

This checklist is to be attached as the last page of the final report.

This checklist is to be duly completed, verified and signed by the student.

1.	**Is the final report neatly formatted with all the elements required for a technical Report?**	Yes / No
2.	Is the Cover page in proper format as given in Annexure A?	Yes / No
3.	Is the Title page (Inner cover page) in proper format?	Yes / No
4.	(a) Is the Certificate from the Supervisor in proper format? (b) Has it been signed by the Supervisor?	Yes / No Yes / No
5.	Is the Abstract included in the report properly written within one page? Have the technical keywords been specified properly?	Yes / No Yes / No
6.	Is the title of your report appropriate? **The title should be adequately descriptive, precise and must reflect scope of the actual work done.** Uncommon abbreviations / Acronyms should not be used in the title	Yes / No
7.	Have you included the List of abbreviations / Acronyms?	Yes / No
8.	Does the Report contain a summary of the literature survey?	Yes / No
9.	Does the Table of Contents include page numbers? i. Are the Pages numbered properly? (Ch. 1 should start on Page # 1) ii. Are the Figures numbered properly? (Figure Numbers and Figure Titles should be at the bottom of the figures) iii. Are the Tables numbered properly? (Table Numbers and Table Titles should be at the top of the tables) iv. Are the Captions for the Figures and Tables proper? v. Are the Appendices numbered properly? Are their titles appropriate	Yes / No Yes / No Yes / No Yes / No Yes / No Yes / No
10.	Is the conclusion of the Report based on discussion of the work?	Yes / No
11.	Are References or Bibliography given at the end of the Report? Have the References been cited properly inside the text of the Report? Are all the references cited in the body of the report	Yes / No Yes / No Yes / No
12.	Is the report format and content according to the guidelines? The report should not be a mere printout of a Power Point Presentation, or a user manual. Source code of software need not be included in the report.	Yes / No

Declaration by Student:

I certify that I have properly verified all the items in this checklist and ensure that the report is in the proper format as specified in the course handout.

Place: Chennai

Signature of the Student