Dwight Look College of ENGINEERING TEXAS A&M UNIVERSITY

Team 22: Smart Cricket Bat
Bi-Weekly Update 4

Team members: Pablo Barron
Gavin Dahl
Jiakai Hu
Nolapat Pipitvitayakul
TA: Fardeen Hasib Mozumder
Sponsor: Pranav Dhulipala

Project description

- Problem: “Cricket practice equipment lacks effectiveness without the assistance and guidance of a coach”

- Solution proposal: “Create a device that will mount on the cricket bat and, through a user friendly app, gives real time feedback on the user’s cricket swing i.e. efficiency and swing angle. Must be easy to use and set up”

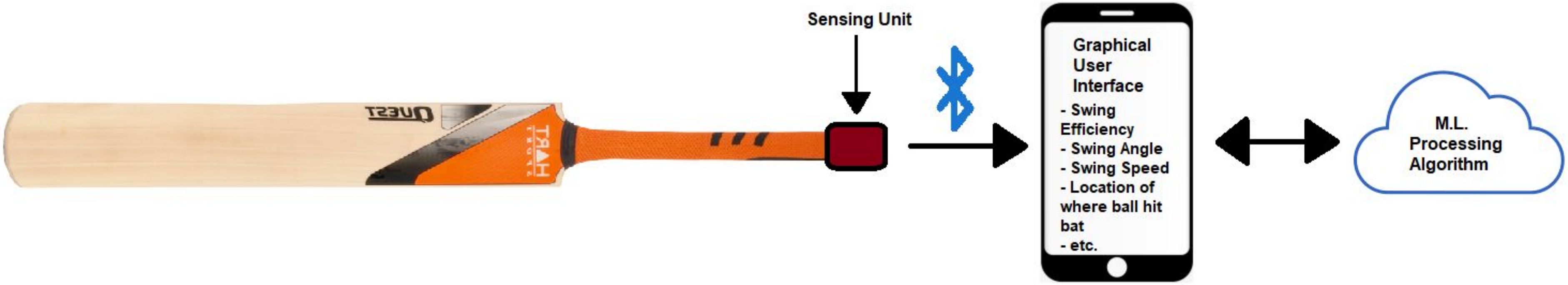
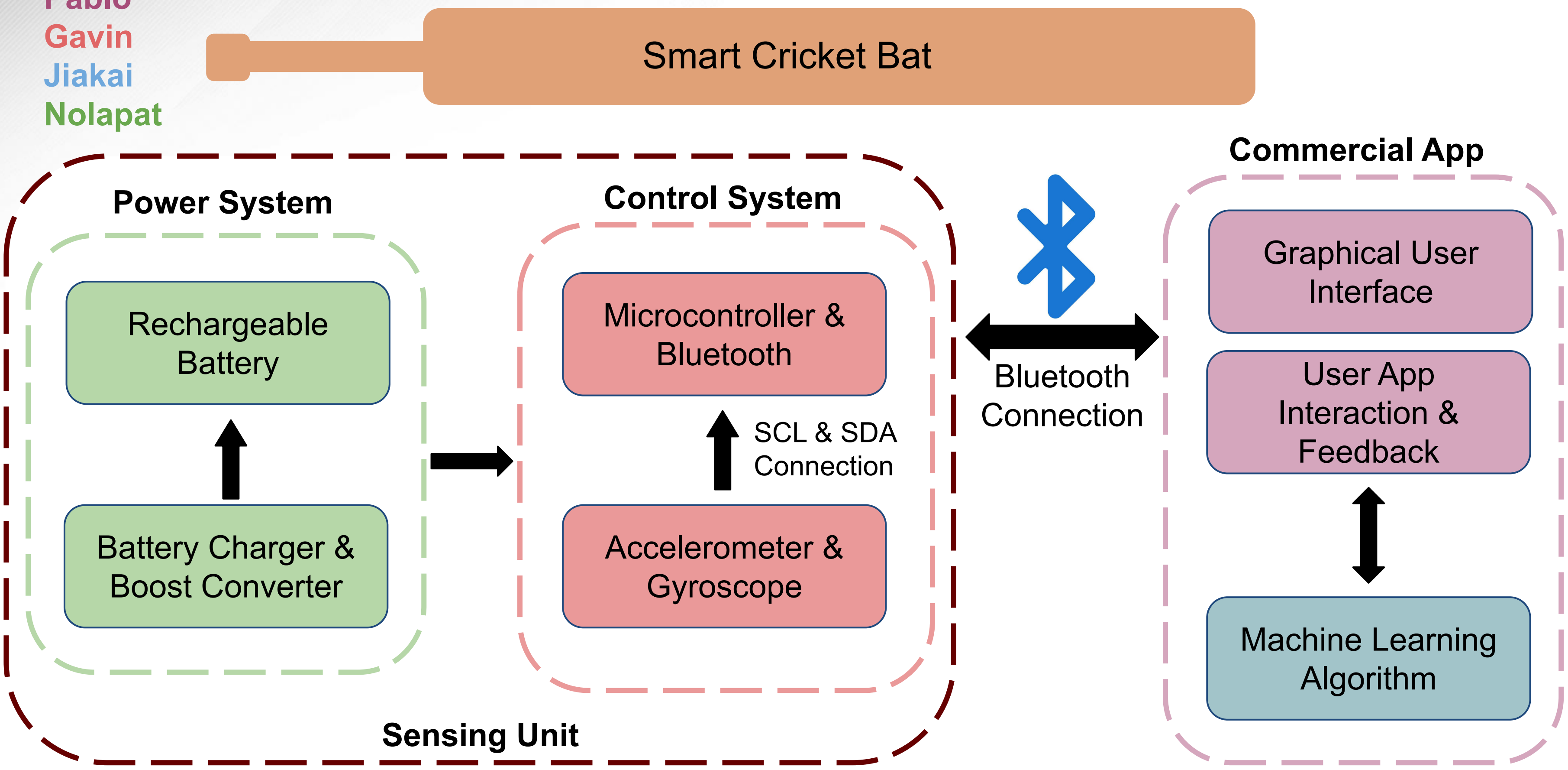


Diagram of subsystems and interface

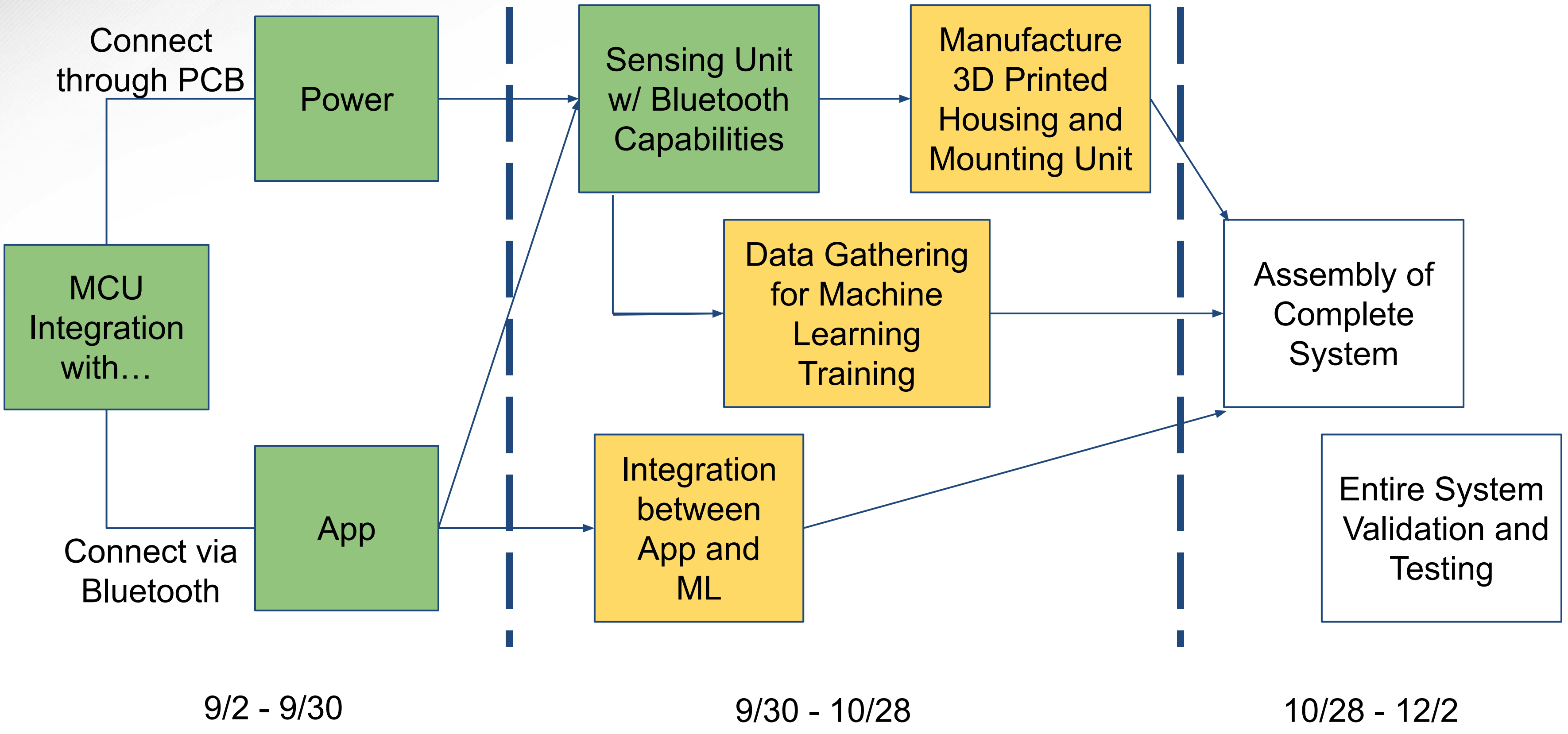
Pablo
Gavin
Jiakai
Nolapat



Subsystem Overview

- Android App: Will handle user interaction and communication between the MCU and the ML algorithm. The app receives data from the MCU via a bluetooth connection. The app will then send received data to ML algorithm to be processed
- Power: Contains the boost-converter, on-board lithium battery power supply, and recharging station
- ML Algorithm: Develop the ML algorithm for finding the specific characteristics of users swing
- Control : A sensing unit that will communicate the IMU values from the swing to the user app through an MCU with bluetooth capabilities

Project Timeline

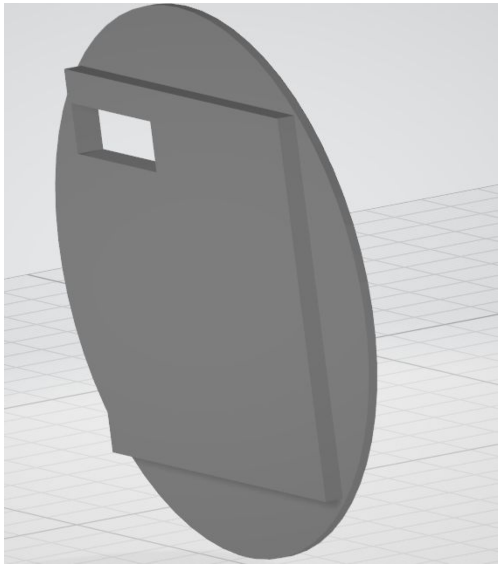


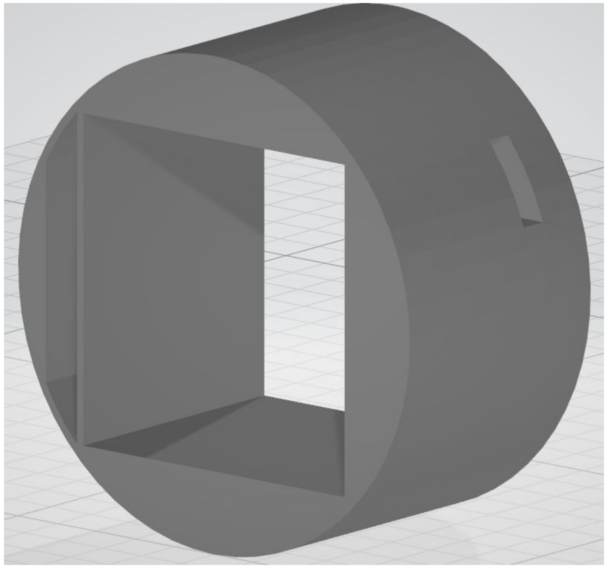
Connect through PCB
Power
MCU Integration with…
App
Connect via Bluetooth
Sensing Unit w/ Bluetooth Capabilities
Manufacture 3D Printed Housing and Mounting Unit
Data Gathering for Machine Learning Training
Integration between App and ML
Assembly of Complete System
Entire System Validation and Testing
9/2 - 9/30
9/30 - 10/28
10/28 - 12/2

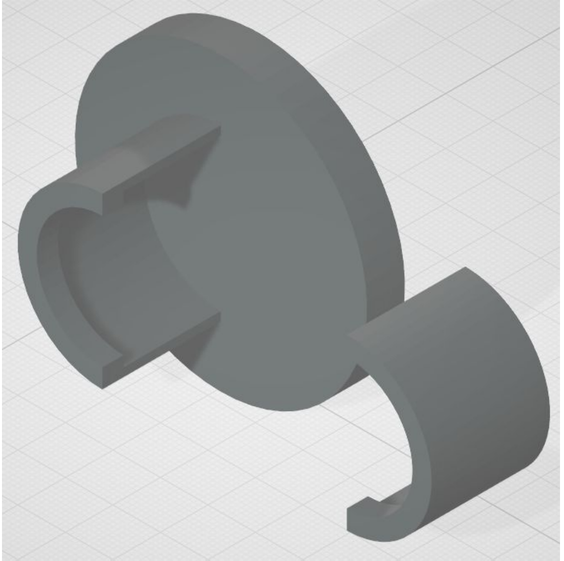
Control System

Gavin Dahl

Accomplishments since last presentation 20 hr of effort	Ongoing progress/problems and plans until the next presentation
● Housing and Mounting Unit Designed	● Finish final PCB assembly and validation ● Housing and Mounting Unit Printing ● ML data collection

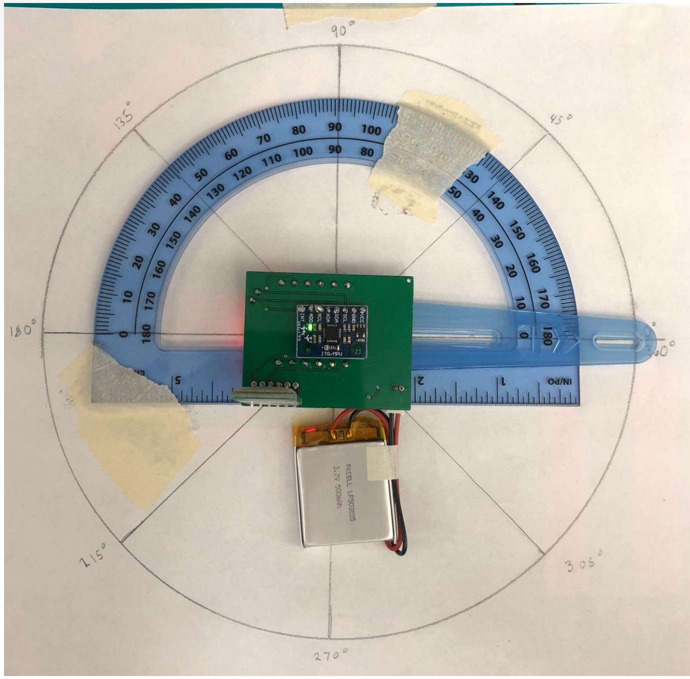


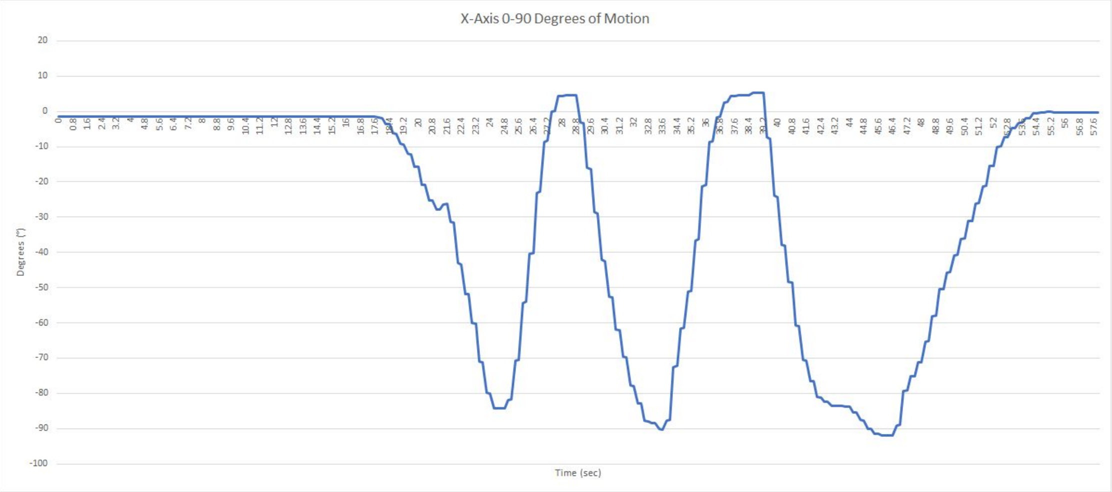


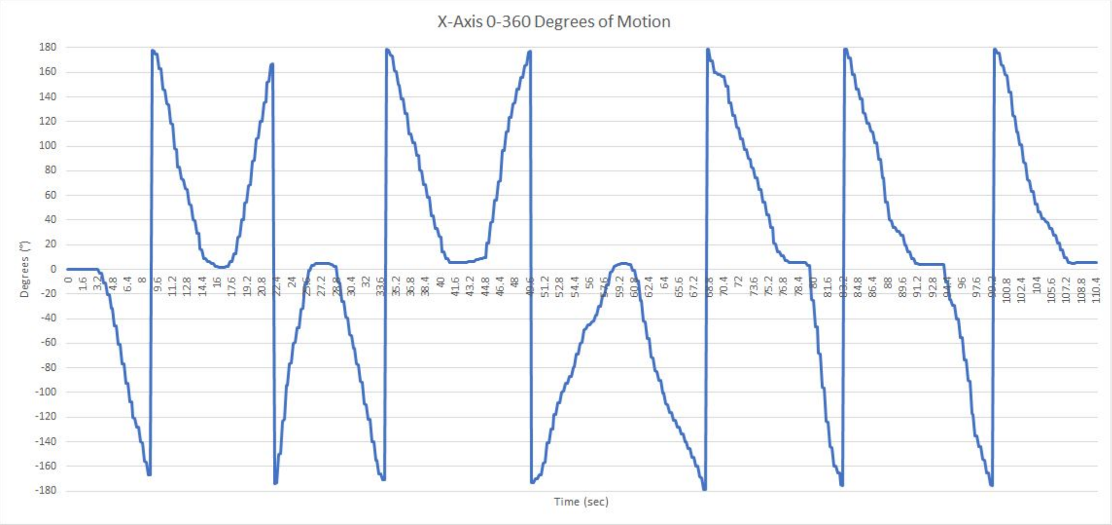


Control System

Gavin Dahl





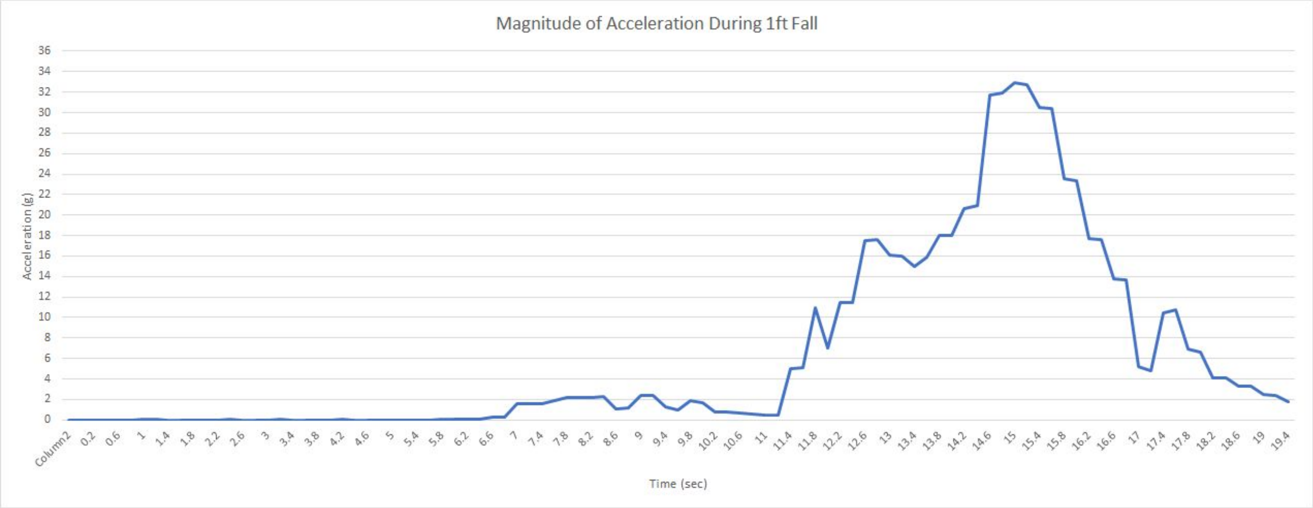


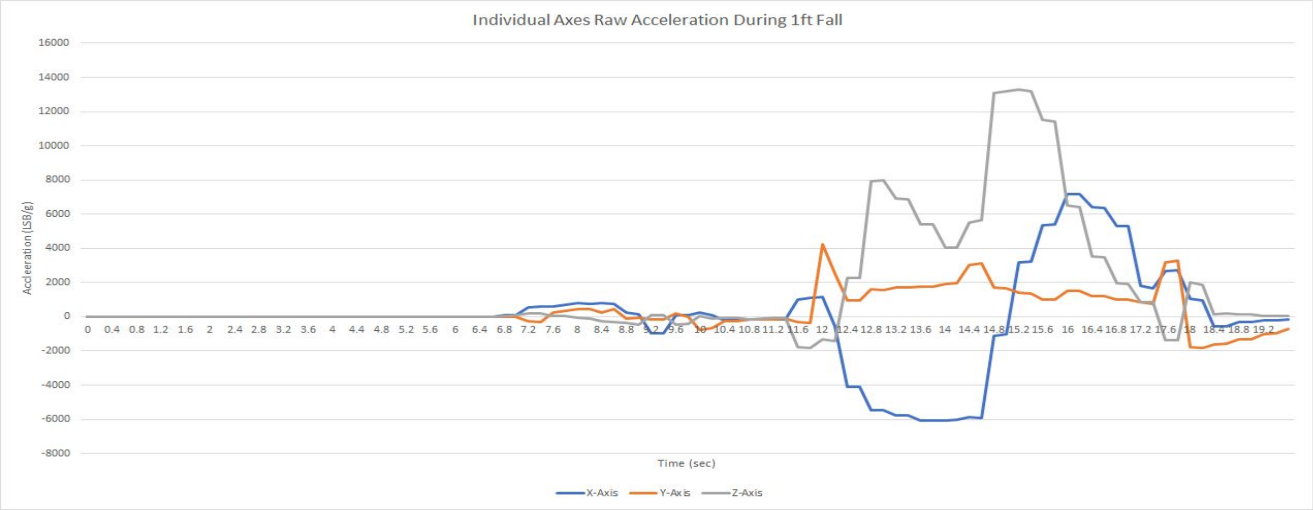
Control System

Gavin Dahl

- Calculated Expected Acceleration at Impact:
  31.16 g’s

- Measured Acceleration at Impact:
  32.883 g’s

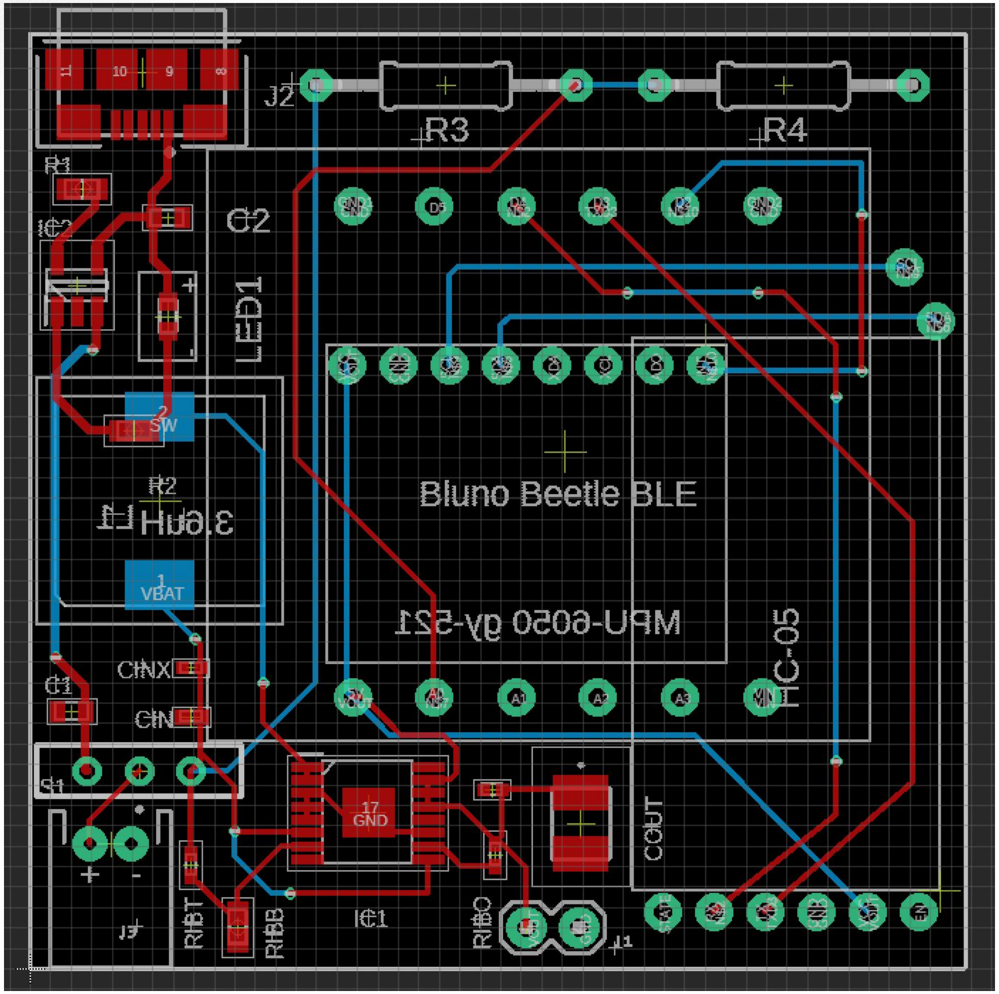




Power System

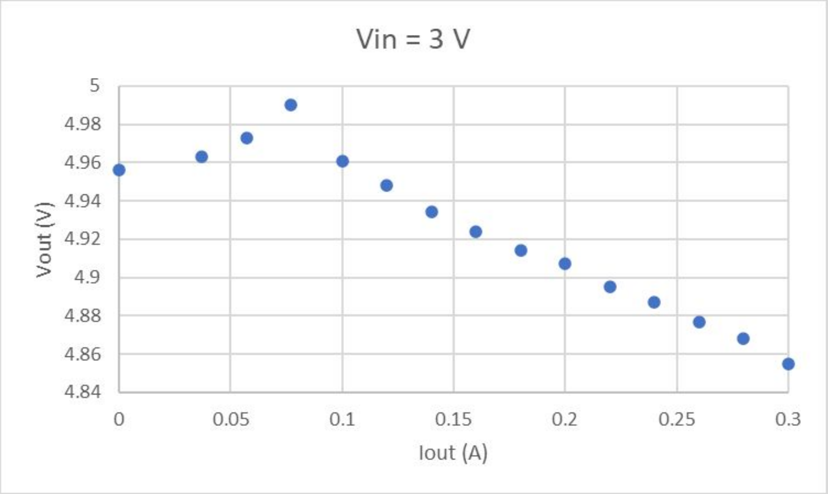
Accomplishments since 403 12 hr of effort	Ongoing progress/problems and plans until the next presentation
● Designed battery charging circuit ● Received final PCB	● Assemble final PCB ● Testing the final PCB ● Help with ML data collection

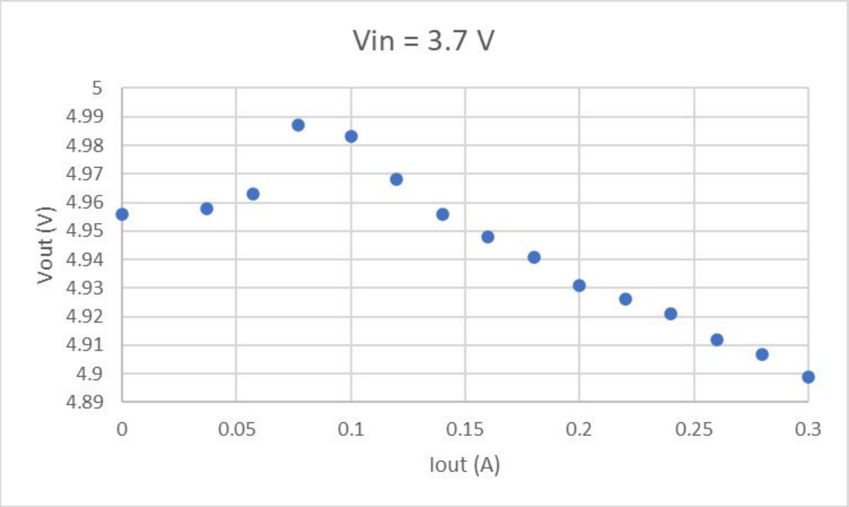
Power System

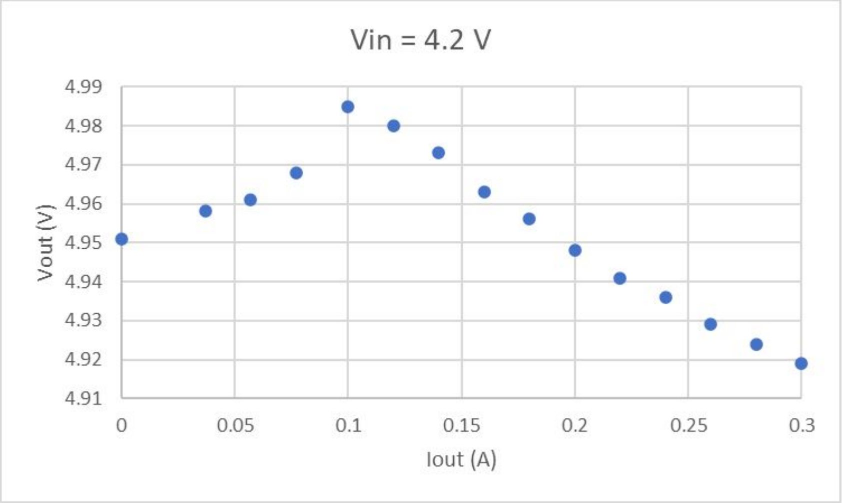


45.7x45.7 mm

Power System





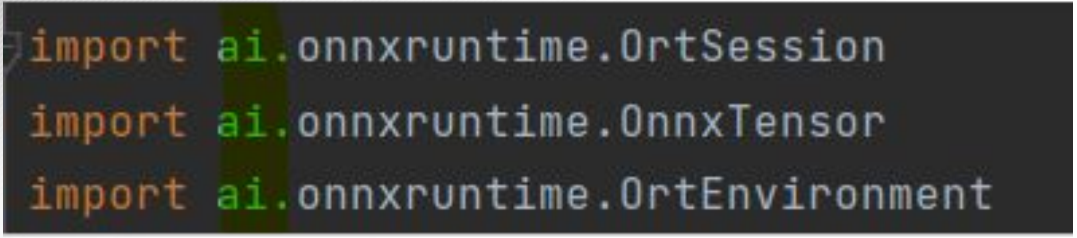


Android App

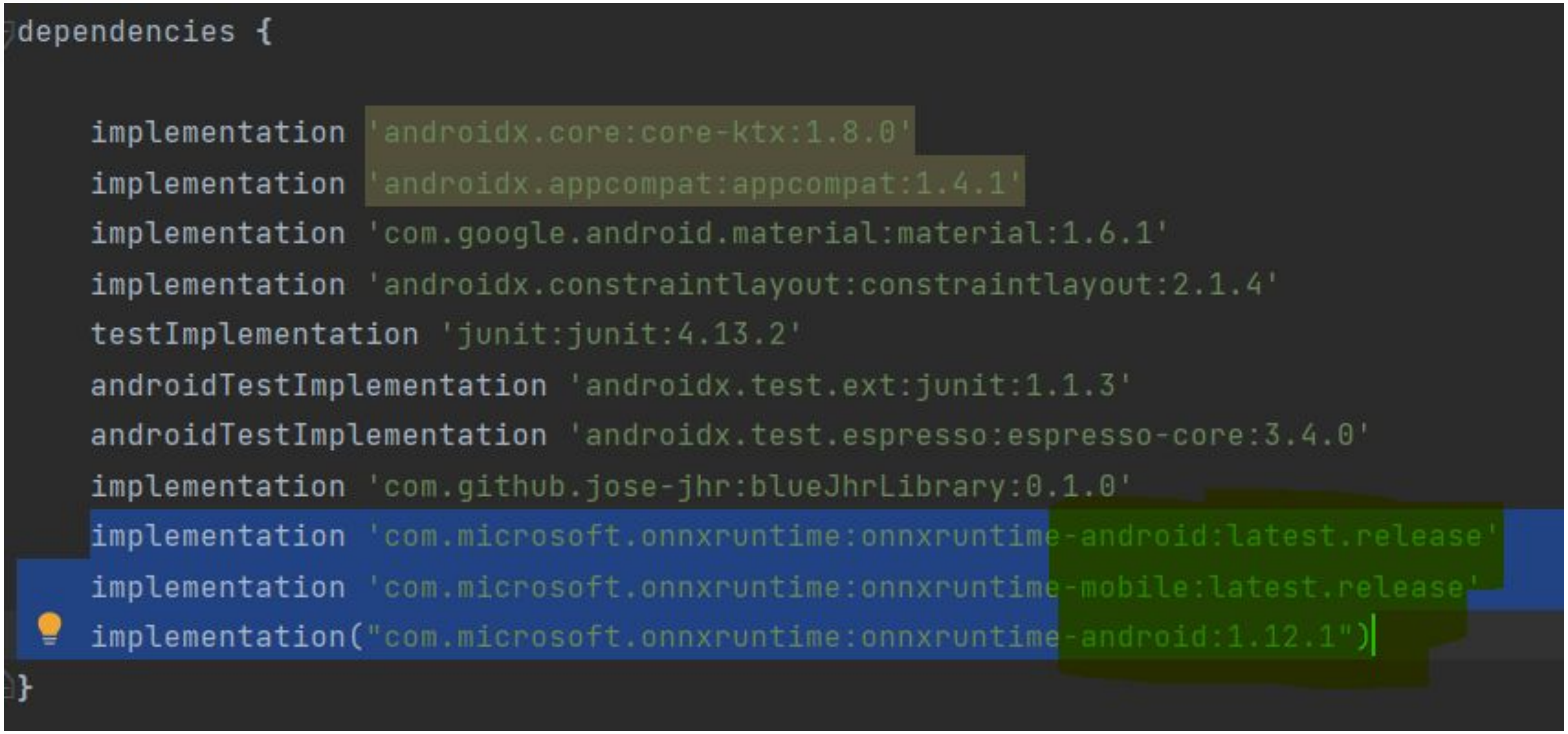
Pablo Barron

Accomplishments since last presentation: 16 Hrs	Ongoing progress/problems and plans until the next presentation
● No accomplishments still working on implementation of ML	● Currently integrating with ML, but encountering difficulty due to Android Studio only liking TensorFlow Models ● Tried Tensor Flow (.tflite and .pb) and skLearn (.sav and .h5) but encountered Converter errors ● Currently Switched to ONNX Model (.ort) using onnxruntime

Android App

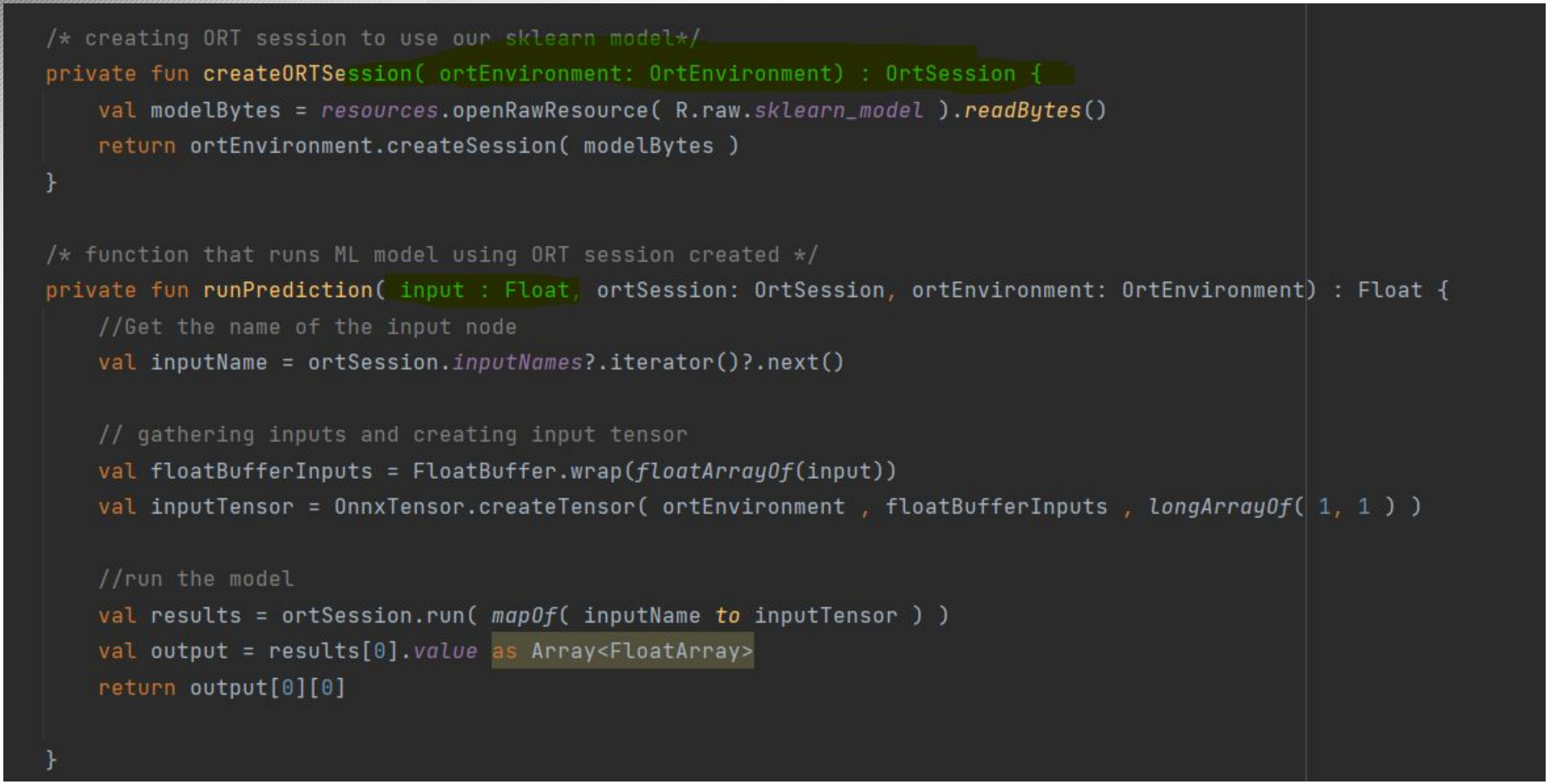


```
import ai.onnxruntime.OrtSession
import ai.onnxruntime.OnnxTensor
import ai.onnxruntime.OrtEnvironment
```



```
dependencies {

    implementation 'androidx.core:core-ktx:1.8.0'
    implementation 'androidx.appcompat:appcompat:1.4.1'
    implementation 'com.google.android.material:material:1.6.1'
    implementation 'androidx.constraintlayout:constraintlayout:2.1.4'
    testImplementation 'junit:junit:4.13.2'
    androidTestImplementation 'androidx.test.ext:junit:1.1.3'
    androidTestImplementation 'androidx.test.espresso:espresso-core:3.4.0'
    implementation 'com.github.jose-jhr:blueJhrLibrary:0.1.0'
    implementation 'com.microsoft.onnxruntime:onnxruntime-android:latest.release'
    implementation 'com.microsoft.onnxruntime:onnxruntime-mobile:latest.release'
    implementation("com.microsoft.onnxruntime:onnxruntime-android:1.12.1")
}
```

Android App



```
/* creating ORT session to use our sklearn model*/
private fun createORTSession( ortEnvironment: OrtEnvironment) : OrtSession {
    val modelBytes = resources.openRawResource( R.raw.sklearn_model ).readBytes()
    return ortEnvironment.createSession( modelBytes )
}

/* function that runs ML model using ORT session created */
private fun runPrediction( input : Float, ortSession: OrtSession, ortEnvironment: OrtEnvironment) : Float {
    //Get the name of the input node
    val inputName = ortSession.inputNames?.iterator()?.next()

    // gathering inputs and creating input tensor
    val floatBufferInputs = FloatBuffer.wrap(floatArrayOf(input))
    val inputTensor = OnnxTensor.createTensor( ortEnvironment , floatBufferInputs , longArrayOf( 1, 1 ) )

    //run the model
    val results = ortSession.run( mapOf( inputName to inputTensor ) )
    val output = results[0].value as Array<FloatArray>
    return output[0][0]

}
```

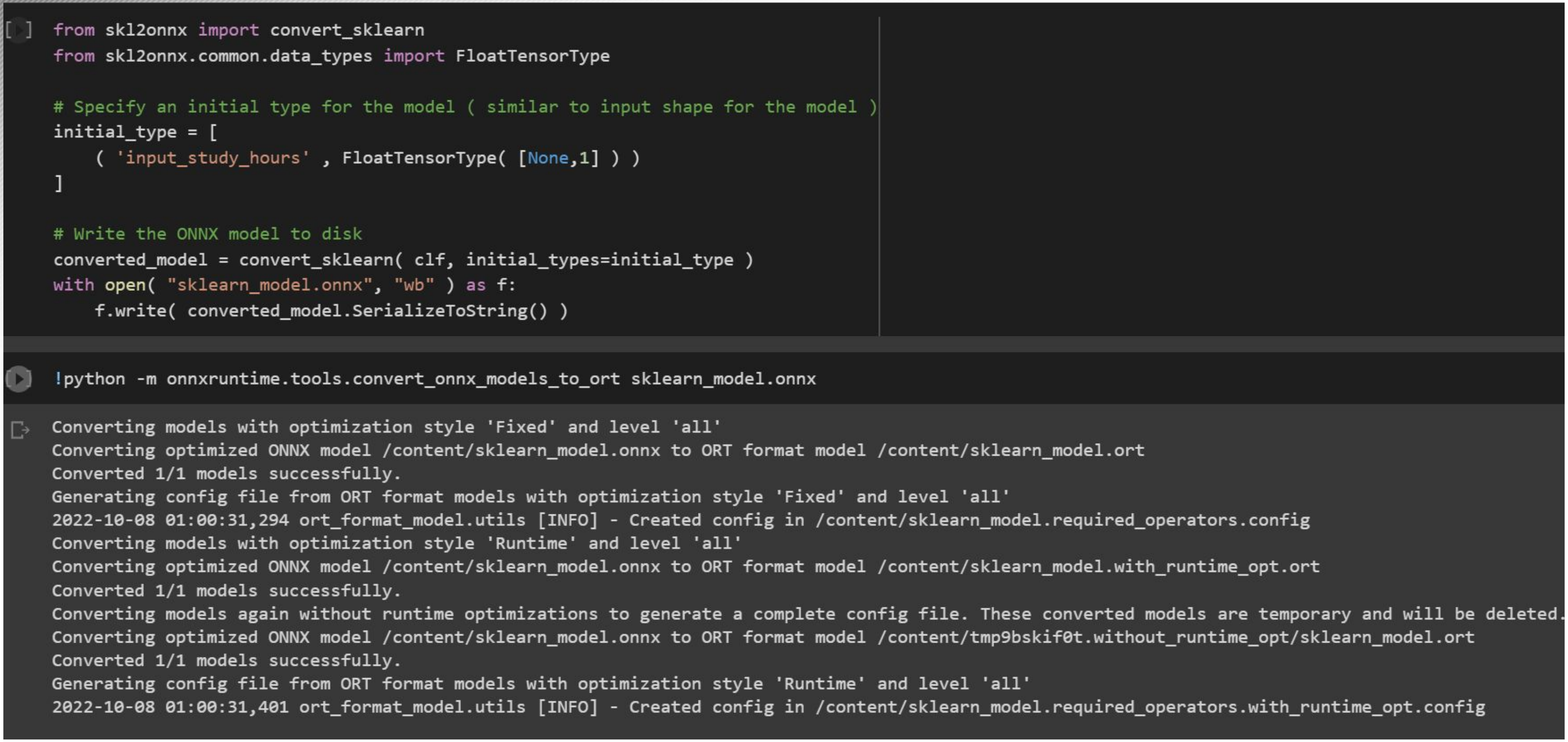
Machine Learning

Accomplishments since last presentation:

- Transforming the Scikit-Learn Random Forest Model to .ort then.onnx file to be used by Android App

Plans until next presentation:

- More data gathering to increase accuracy

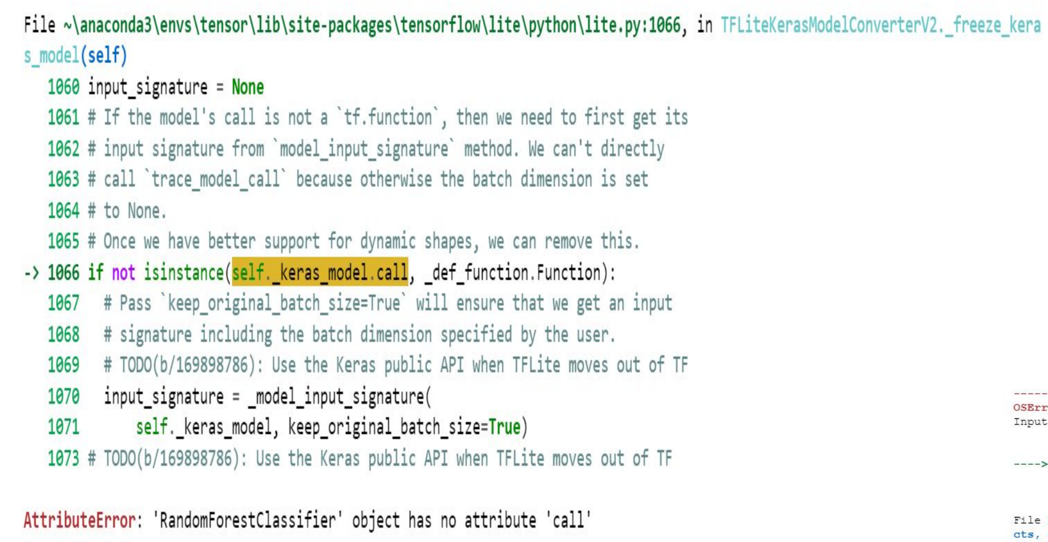


```python
from skl2onnx import convert_sklearn
from skl2onnx.common.data_types import FloatTensorType

# Specify an initial type for the model ( similar to input shape for the model )
initial_type = [
    ( 'input_study_hours' , FloatTensorType( [None,1] ) )
]

# Write the ONNX model to disk
converted_model = convert_sklearn( clf, initial_types=initial_type )
with open( "sklearn_model.onnx", "wb" ) as f:
    f.write( converted_model.SerializeToString() )
```

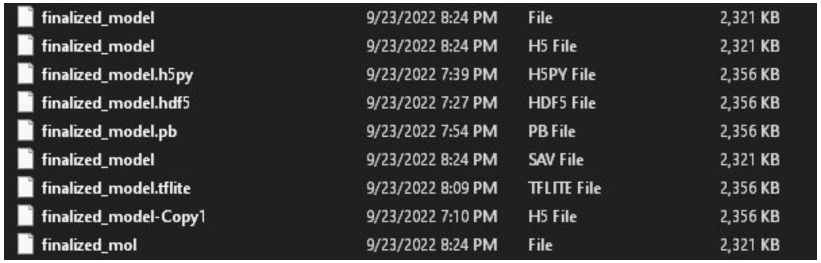
```
!python -m onnxruntime.tools.convert_onnx_models_to_ort sklearn_model.onnx
```

```
Converting models with optimization style 'Fixed' and level 'all'
Converting optimized ONNX model /content/sklearn_model.onnx to ORT format model /content/sklearn_model.ort
Converted 1/1 models successfully.
Generating config file from ORT format models with optimization style 'Fixed' and level 'all'
2022-10-08 01:00:31,294 ort_format_model.utils [INFO] - Created config in /content/sklearn_model.required_operators.config
Converting models with optimization style 'Runtime' and level 'all'
Converting optimized ONNX model /content/sklearn_model.onnx to ORT format model /content/sklearn_model.with_runtime_opt.ort
Converted 1/1 models successfully.
Converting models again without runtime optimizations to generate a complete config file. These converted models are temporary and will be deleted.
Converting optimized ONNX model /content/sklearn_model.onnx to ORT format model /content/tmp9bskif0t.without_runtime_opt/sklearn_model.ort
Converted 1/1 models successfully.
Generating config file from ORT format models with optimization style 'Runtime' and level 'all'
2022-10-08 01:00:31,401 ort_format_model.utils [INFO] - Created config in /content/sklearn_model.required_operators.with_runtime_opt.config
```

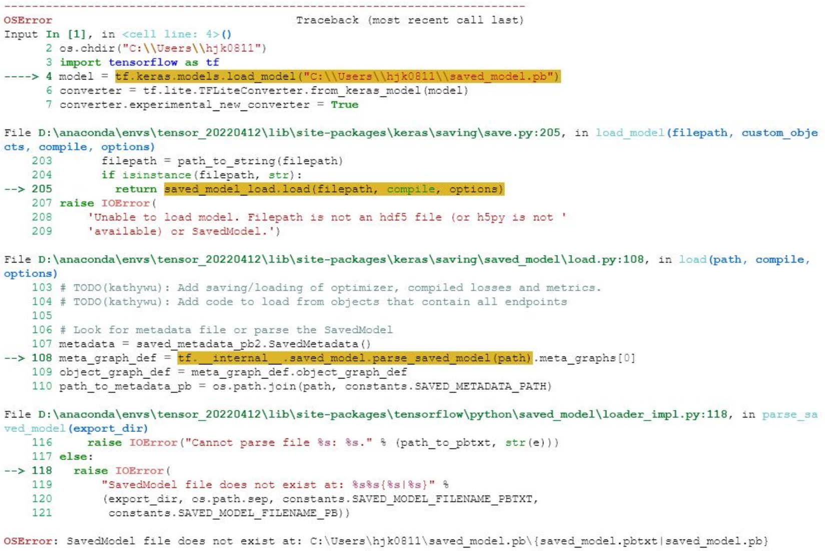
Errors when Converting sklearn model with all kinds of file type to .tflite



```
File ~\anaconda3\envs\tensor\lib\site-packages\tensorflow\lite\python\lite.py:1066, in TFLiteKerasModelConverterV2._freeze_kera
s_model(self)
   1060 input_signature = None
   1061 # If the model's call is not a `tf.function`, then we need to first get its
   1062 # input signature from `model_input_signature` method. We can't directly
   1063 # call `trace_model_call` because otherwise the batch dimension is set
   1064 # to None.
   1065 # Once we have better support for dynamic shapes, we can remove this.
-> 1066 if not isinstance(self._keras_model.call, _def_function.Function):
   1067   # Pass `keep_original_batch_size=True` will ensure that we get an input
   1068   # signature including the batch dimension specified by the user.
   1069   # TODO(b/169898786): Use the Keras public API when TFLite moves out of TF
   1070   input_signature = _model_input_signature(
   1071       self._keras_model, keep_original_batch_size=True)
   1073 # TODO(b/169898786): Use the Keras public API when TFLite moves out of TF

AttributeError: 'RandomForestClassifier' object has no attribute 'call'
```

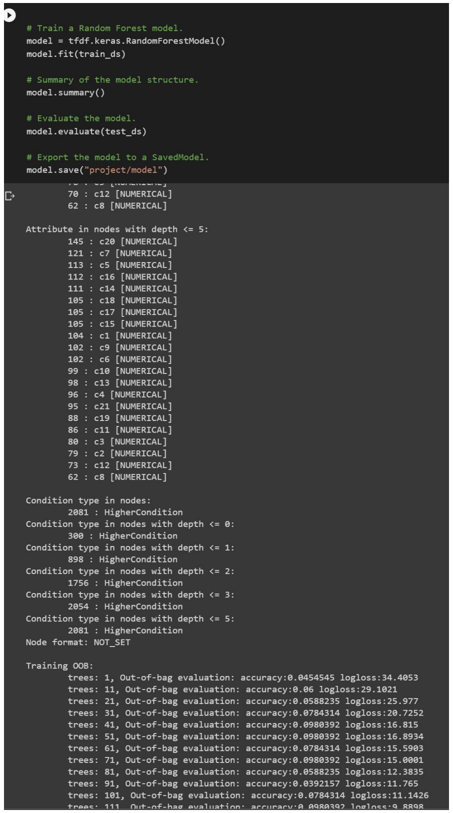


finalized_model	9/23/2022 8:24 PM	File	2,321 KB
finalized_model	9/23/2022 8:24 PM	H5 File	2,321 KB
finalized_model.h5py	9/23/2022 7:39 PM	H5PY File	2,356 KB
finalized_model.hdf5	9/23/2022 7:27 PM	HDF5 File	2,356 KB
finalized_model.pb	9/23/2022 7:54 PM	PB File	2,356 KB
finalized_model	9/23/2022 8:24 PM	SAV File	2,321 KB
finalized_model.tflite	9/23/2022 8:09 PM	TFLITE File	2,356 KB
finalized_model-Copy1	9/23/2022 7:10 PM	H5 File	2,356 KB
finalized_mol	9/23/2022 8:24 PM	File	2,321 KB



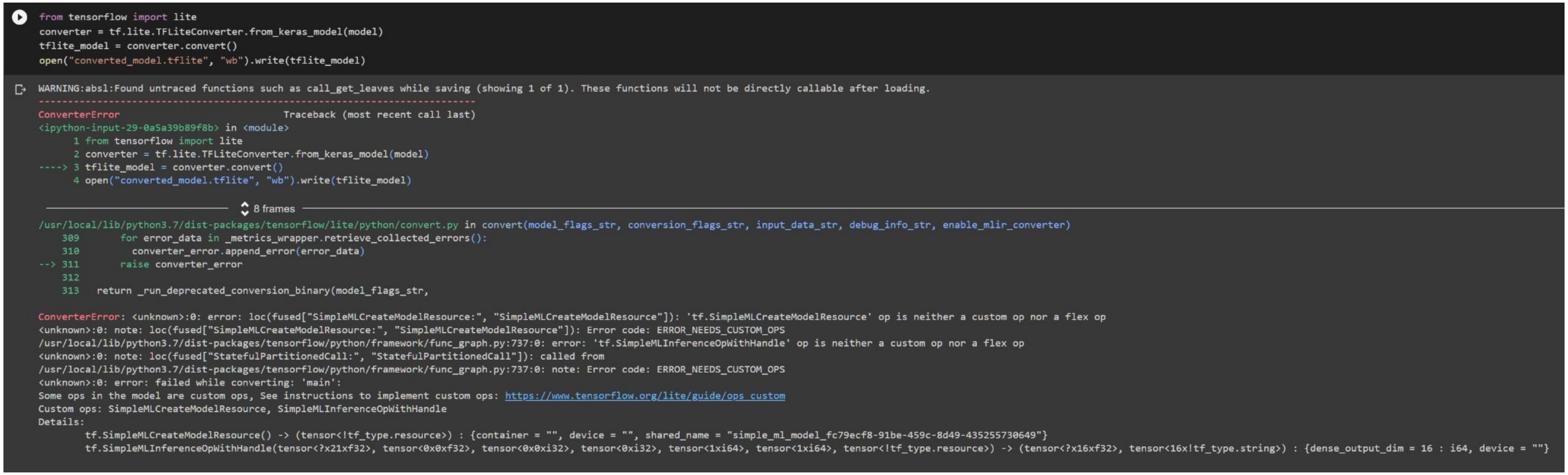
```
---------------------------------------------------------------------------
OSError                                   Traceback (most recent call last)
Input In [1], in <cell line: 4>()
      2 os.chdir("C:\\Users\\hjk0811")
      3 import tensorflow as tf
----> 4 model = tf.keras.models.load_model("C:\\Users\\hjk0811\\saved_model.pb")
      6 converter = tf.lite.TFLiteConverter.from_keras_model(model)
      7 converter.experimental_new_converter = True

File D:\anaconda\envs\tensor_20220412\lib\site-packages\keras\saving\save.py:205, in load_model(filepath, custom_obje
cts, compile, options)
    203       filepath = path_to_string(filepath)
    204       if isinstance(filepath, str):
--> 205         return saved_model_load.load(filepath, compile, options)
    207 raise IOError(
    208     'Unable to load model. Filepath is not an hdf5 file (or h5py is not '
    209     'available) or SavedModel.')

File D:\anaconda\envs\tensor_20220412\lib\site-packages\keras\saving\saved_model\load.py:108, in load(path, compile,
options)
    103 # TODO(kathywu): Add saving/loading of optimizer, compiled losses and metrics.
    104 # TODO(kathywu): Add code to load from objects that contain all endpoints
    105 
    106 # Look for metadata file or parse the SavedModel
    107 metadata = saved_metadata_pb2.SavedMetadata()
--> 108 meta_graph_def = tf.__internal__.saved_model.parse_saved_model(path).meta_graphs[0]
    109 object_graph_def = meta_graph_def.object_graph_def
    110 path_to_metadata_pb = os.path.join(path, constants.SAVED_METADATA_PATH)

File D:\anaconda\envs\tensor_20220412\lib\site-packages\tensorflow\python\saved_model\loader_impl.py:118, in parse_sa
ved_model(export_dir)
    116     raise IOError("Cannot parse file %s: %s." % (path_to_pbtxt, str(e)))
    117 else:
--> 118   raise IOError(
    119       "SavedModel file does not exist at: %s%s{%s|%s}" %
    120       (export_dir, os.path.sep, constants.SAVED_MODEL_FILENAME_PBTXT,
    121        constants.SAVED_MODEL_FILENAME_PB))

OSError: SavedModel file does not exist at: C:\Users\hjk0811\saved_model.pb\{saved_model.pbtxt|saved_model.pb}
```

Errors when Converting tensorflow model with all kinds of file type to .tflite



```
# Train a Random Forest model.
model = tfdf.keras.RandomForestModel()
model.fit(train_ds)

# Summary of the model structure.
model.summary()

# Evaluate the model.
model.evaluate(test_ds)

# Export the model to a SavedModel.
model.save("project/model")
```

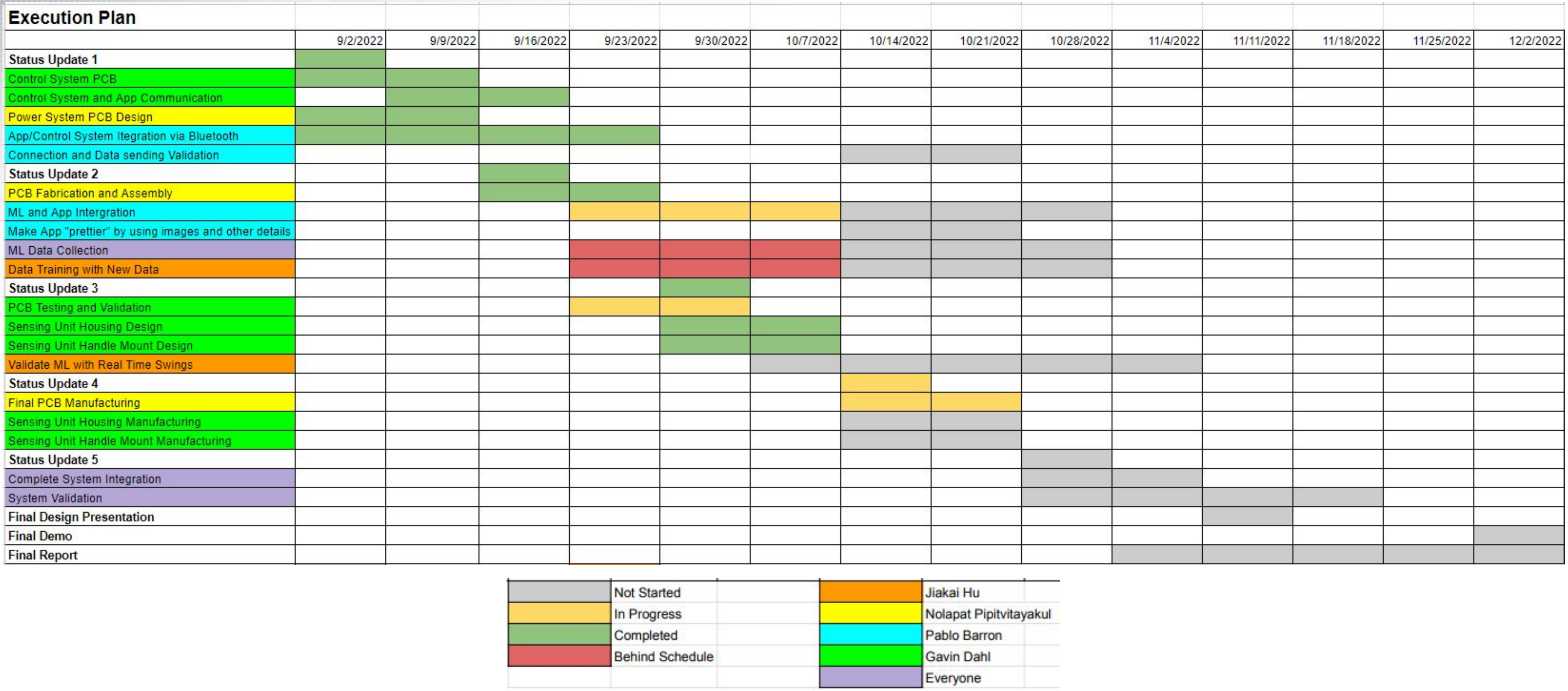
```
        70 : c12 [NUMERICAL]
        62 : c8 [NUMERICAL]

Attribute in nodes with depth <= 5:
        145 : c20 [NUMERICAL]
        121 : c7 [NUMERICAL]
        113 : c5 [NUMERICAL]
        112 : c16 [NUMERICAL]
        111 : c14 [NUMERICAL]
        105 : c18 [NUMERICAL]
        105 : c17 [NUMERICAL]
        105 : c15 [NUMERICAL]
        104 : c1 [NUMERICAL]
        102 : c9 [NUMERICAL]
        102 : c6 [NUMERICAL]
        99 : c10 [NUMERICAL]
        98 : c13 [NUMERICAL]
        96 : c4 [NUMERICAL]
        95 : c21 [NUMERICAL]
        88 : c19 [NUMERICAL]
        86 : c11 [NUMERICAL]
        80 : c3 [NUMERICAL]
        79 : c2 [NUMERICAL]
        73 : c12 [NUMERICAL]
        62 : c8 [NUMERICAL]

Condition type in nodes:
        2081 : HigherCondition
Condition type in nodes with depth <= 0:
        300 : HigherCondition
Condition type in nodes with depth <= 1:
        898 : HigherCondition
Condition type in nodes with depth <= 2:
        1756 : HigherCondition
Condition type in nodes with depth <= 3:
        2054 : HigherCondition
Condition type in nodes with depth <= 5:
        2081 : HigherCondition
Node format: NOT_SET

Training OOB:
        trees: 1, Out-of-bag evaluation: accuracy:0.0454545 logloss:34.4053
        trees: 11, Out-of-bag evaluation: accuracy:0.06 logloss:29.1021
        trees: 21, Out-of-bag evaluation: accuracy:0.0588235 logloss:25.977
        trees: 31, Out-of-bag evaluation: accuracy:0.0784314 logloss:20.7252
        trees: 41, Out-of-bag evaluation: accuracy:0.0980392 logloss:16.815
        trees: 51, Out-of-bag evaluation: accuracy:0.0980392 logloss:16.8934
        trees: 61, Out-of-bag evaluation: accuracy:0.0784314 logloss:15.5903
        trees: 71, Out-of-bag evaluation: accuracy:0.0980392 logloss:15.0001
        trees: 81, Out-of-bag evaluation: accuracy:0.0588235 logloss:12.3835
        trees: 91, Out-of-bag evaluation: accuracy:0.0392157 logloss:11.765
        trees: 101, Out-of-bag evaluation: accuracy:0.0784314 logloss:11.1426
        trees: 111, Out-of-bag evaluation: accuracy:0.0980392 logloss:9.8898
```



```
from tensorflow import lite
converter = tf.lite.TFLiteConverter.from_keras_model(model)
tflite_model = converter.convert()
open("converted_model.tflite", "wb").write(tflite_model)
```

```
WARNING:absl:Found untraced functions such as call_get_leaves while saving (showing 1 of 1). These functions will not be directly callable after loading.
---------------------------------------------------------------------------
ConverterError                            Traceback (most recent call last)
<ipython-input-29-0a5a39b89f8b> in <module>
      1 from tensorflow import lite
      2 converter = tf.lite.TFLiteConverter.from_keras_model(model)
----> 3 tflite_model = converter.convert()
      4 open("converted_model.tflite", "wb").write(tflite_model)

                                          8 frames
/usr/local/lib/python3.7/dist-packages/tensorflow/lite/python/convert.py in convert(model_flags_str, conversion_flags_str, input_data_str, debug_info_str, enable_mlir_converter)
    309       for error_data in _metrics_wrapper.retrieve_collected_errors():
    310         converter_error.append_error(error_data)
--> 311       raise converter_error
    312 
    313   return _run_deprecated_conversion_binary(model_flags_str,

ConverterError: <unknown>:0: error: loc(fused["SimpleMLCreateModelResource:", "SimpleMLCreateModelResource"]): 'tf.SimpleMLCreateModelResource' op is neither a custom op nor a flex op
<unknown>:0: note: loc(fused["SimpleMLCreateModelResource:", "SimpleMLCreateModelResource"]): Error code: ERROR_NEEDS_CUSTOM_OPS
/usr/local/lib/python3.7/dist-packages/tensorflow/python/framework/func_graph.py:737:0: error: 'tf.SimpleMLInferenceOpWithHandle' op is neither a custom op nor a flex op
<unknown>:0: note: loc(fused["StatefulPartitionedCall:", "StatefulPartitionedCall"]): called from
/usr/local/lib/python3.7/dist-packages/tensorflow/python/framework/func_graph.py:737:0: note: Error code: ERROR_NEEDS_CUSTOM_OPS
<unknown>:0: error: failed while converting: 'main':
Some ops in the model are custom ops, See instructions to implement custom ops: https://www.tensorflow.org/lite/guide/ops_custom
Custom ops: SimpleMLCreateModelResource, SimpleMLInferenceOpWithHandle
Details:
        tf.SimpleMLCreateModelResource() -> (tensor<!tf_type.resource>) : {container = "", device = "", shared_name = "simple_ml_model_fc79ecf8-91be-459c-8d49-435255730649"}
        tf.SimpleMLInferenceOpWithHandle(tensor<?x21xf32>, tensor<0x0xf32>, tensor<0x0xi32>, tensor<0xi32>, tensor<1xi64>, tensor<1xi64>, tensor<!tf_type.resource>) -> (tensor<?x16xf32>, tensor<16x!tf_type.string>) : {dense_output_dim = 16 : i64, device = ""}
```

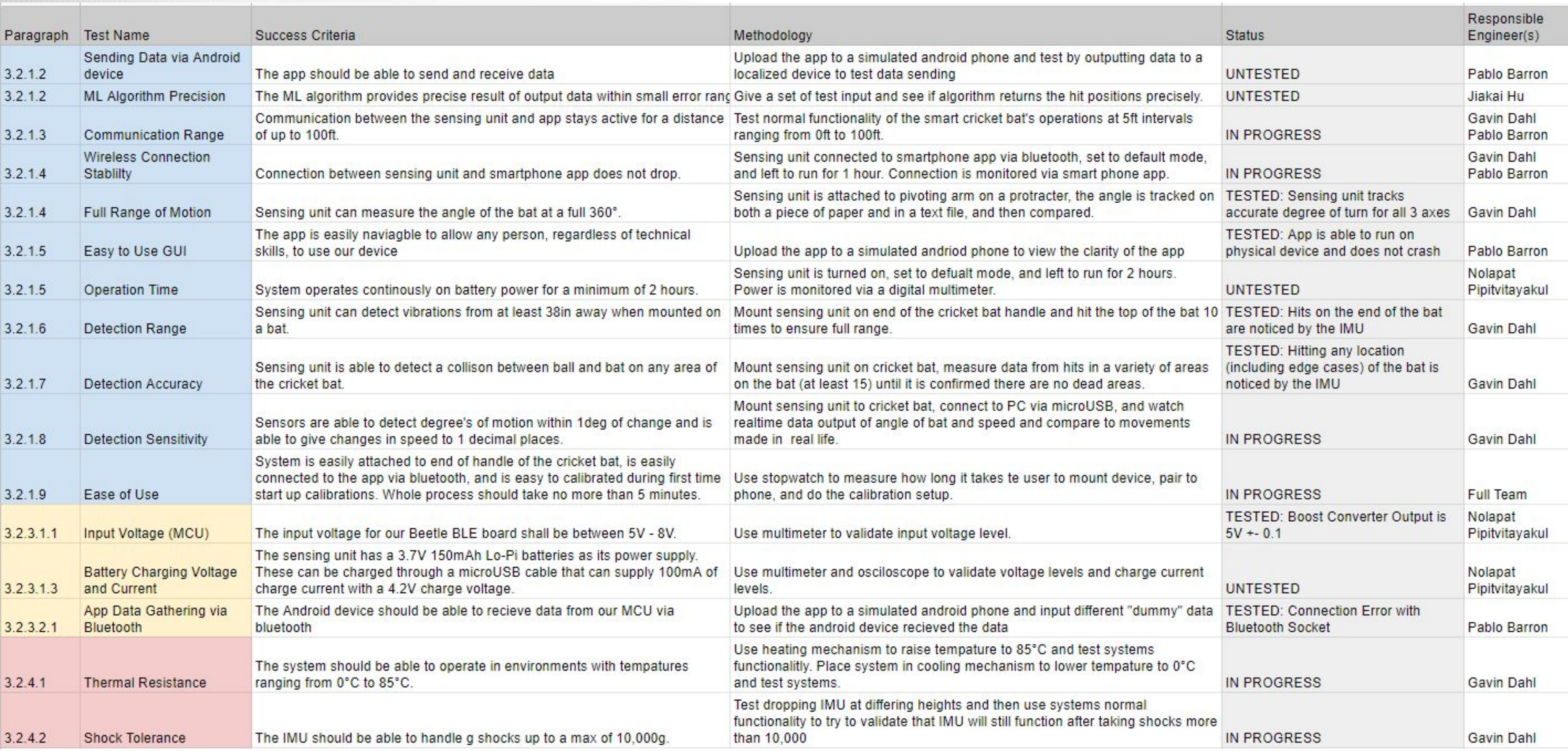
Execution plan



Execution Plan	9/2/2022	9/9/2022	9/16/2022	9/23/2022	9/30/2022	10/7/2022	10/14/2022	10/21/2022	10/28/2022	11/4/2022	11/11/2022	11/18/2022	11/25/2022	12/2/2022
Status Update 1	🟢													
Control System PCB	🟢	🟢												
Control System and App Communication		🟢	🟢											
Power System PCB Design	🟢	🟢												
App/Control System Itegration via Bluetooth	🟢	🟢	🟢	🟢										
Connection and Data sending Validation							▨	▨						
Status Update 2			🟢											
PCB Fabrication and Assembly			🟢	🟢										
ML and App Intergration				🟡	🟡	🟡	▨	▨	▨					
Make App "prettier" by using images and other details							▨	▨						
ML Data Collection				🔴	🔴	🔴	▨	▨	▨					
Data Training with New Data				🔴	🔴	🔴	▨	▨	▨					
Status Update 3					🟢									
PCB Testing and Validation				🟡	🟡									
Sensing Unit Housing Design					🟢	🟢								
Sensing Unit Handle Mount Design					🟢	🟢								
Validate ML with Real Time Swings						▨	▨	▨	▨	▨				
Status Update 4							🟡							
Final PCB Manufacturing							🟡	🟡						
Sensing Unit Housing Manufacturing							▨	▨						
Sensing Unit Handle Mount Manufacturing							▨	▨						
Status Update 5									▨					
Complete System Integration									▨	▨				
System Validation									▨	▨	▨	▨		
Final Design Presentation											▨			
Final Demo														▨
Final Report										▨	▨	▨	▨	▨

▨	Not Started	🟧	Jiakai Hu
🟡	In Progress	🟨	Nolapat Pipitvitayakul
🟢	Completed	🟦	Pablo Barron
🔴	Behind Schedule	🟩	Gavin Dahl
		🟪	Everyone

Validation plan



Paragraph	Test Name	Success Criteria	Methodology	Status	Responsible Engineer(s)
3.2.1.2	Sending Data via Android device	The app should be able to send and receive data	Upload the app to a simulated android phone and test by outputting data to a localized device to test data sending	UNTESTED	Pablo Barron
3.2.1.2	ML Algorithm Precision	The ML algorithm provides precise result of output data within small error rang	Give a set of test input and see if algorithm returns the hit positions precisely.	UNTESTED	Jiakai Hu
3.2.1.3	Communication Range	Communication between the sensing unit and app stays active for a distance of up to 100ft.	Test normal functionality of the smart cricket bat's operations at 5ft intervals ranging from 0ft to 100ft.	IN PROGRESS	Gavin Dahl Pablo Barron
3.2.1.4	Wireless Connection Stablilty	Connection between sensing unit and smartphone app does not drop.	Sensing unit connected to smartphone app via bluetooth, set to default mode, and left to run for 1 hour. Connection is monitored via smart phone app.	IN PROGRESS	Gavin Dahl Pablo Barron
3.2.1.4	Full Range of Motion	Sensing unit can measure the angle of the bat at a full 360°.	Sensing unit is attached to pivoting arm on a protracter, the angle is tracked on both a piece of paper and in a text file, and then compared.	TESTED: Sensing unit tracks accurate degree of turn for all 3 axes	Gavin Dahl
3.2.1.5	Easy to Use GUI	The app is easily naviagble to allow any person, regardless of technical skills, to use our device	Upload the app to a simulated andriod phone to view the clarity of the app	TESTED: App is able to run on physical device and does not crash	Pablo Barron
3.2.1.5	Operation Time	System operates continously on battery power for a minimum of 2 hours.	Sensing unit is turned on, set to defualt mode, and left to run for 2 hours. Power is monitored via a digital multimeter.	UNTESTED	Nolapat Pipitvitayakul
3.2.1.6	Detection Range	Sensing unit can detect vibrations from at least 38in away when mounted on a bat.	Mount sensing unit on end of the cricket bat handle and hit the top of the bat 10 times to ensure full range.	TESTED: Hits on the end of the bat are noticed by the IMU	Gavin Dahl
3.2.1.7	Detection Accuracy	Sensing unit is able to detect a collison between ball and bat on any area of the cricket bat.	Mount sensing unit on cricket bat, measure data from hits in a variety of areas on the bat (at least 15) until it is confirmed there are no dead areas.	TESTED: Hitting any location (including edge cases) of the bat is noticed by the IMU	Gavin Dahl
3.2.1.8	Detection Sensitivity	Sensors are able to detect degree's of motion within 1deg of change and is able to give changes in speed to 1 decimal places.	Mount sensing unit to cricket bat, connect to PC via microUSB, and watch realtime data output of angle of bat and speed and compare to movements made in real life.	IN PROGRESS	Gavin Dahl
3.2.1.9	Ease of Use	System is easily attached to end of handle of the cricket bat, is easily connected to the app via bluetooth, and is easy to calibrated during first time start up calibrations. Whole process should take no more than 5 minutes.	Use stopwatch to measure how long it takes te user to mount device, pair to phone, and do the calibration setup.	IN PROGRESS	Full Team
3.2.3.1.1	Input Voltage (MCU)	The input voltage for our Beetle BLE board shall be between 5V - 8V.	Use multimeter to validate input voltage level.	TESTED: Boost Converter Output is 5V +- 0.1	Nolapat Pipitvitayakul
3.2.3.1.3	Battery Charging Voltage and Current	The sensing unit has a 3.7V 150mAh Lo-Pi batteries as its power supply. These can be charged through a microUSB cable that can supply 100mA of charge current with a 4.2V charge voltage.	Use multimeter and osciloscope to validate voltage levels and charge current levels.	UNTESTED	Nolapat Pipitvitayakul
3.2.3.2.1	App Data Gathering via Bluetooth	The Android device should be able to recieve data from our MCU via bluetooth	Upload the app to a simulated android phone and input different "dummy" data to see if the android device recieved the data	TESTED: Connection Error with Bluetooth Socket	Pablo Barron
3.2.4.1	Thermal Resistance	The system should be able to operate in environments with tempatures ranging from 0°C to 85°C.	Use heating mechanism to raise tempature to 85°C and test systems functionalitly. Place system in cooling mechanism to lower tempature to 0°C and test systems.	IN PROGRESS	Gavin Dahl
3.2.4.2	Shock Tolerance	The IMU should be able to handle g shocks up to a max of 10,000g.	Test dropping IMU at differing heights and then use systems normal functionality to try to validate that IMU will still function after taking shocks more than 10,000	IN PROGRESS	Gavin Dahl

Parts Ordered

Part Description	Order Status
3 x Bluno Ble Microcontroller	Part Received
2 x HC-05 Bluetooth Module	Part Received
3 x gy-521 IMU	Part Received
1 x LiPo Battery 3.7V 150mah	Part Received
12 x Capacitor	Part Received
8 x Resistor	Part Received
3 x Inductor	Part Received
1 x Switching Voltage Regulators	Part Received
1 x JST Right-Angle Connector TH 2-Pin	Part Received

Questions?