EE / CprE / SE 491 – sdmay21-09

Instruction Level Reverse Engineering through EM Side Channel

Week 4 Report

3/2/2021 – 3/15/2021
Client: Akhilesh Tyagi
Faculty Advisor: Akhilesh Tyagi

Team Members:

Noah Berthusen — *Data Analysis Engineer*

Matthew Campbell — *Test Engineer*

Cristian George — *Meeting Scribe*

Jesse Knight — *Signals Processing Engineer*

Evan McKinney — *Integration Engineer*

Jacob Vaughn — *Report Manager*

Weekly Summary

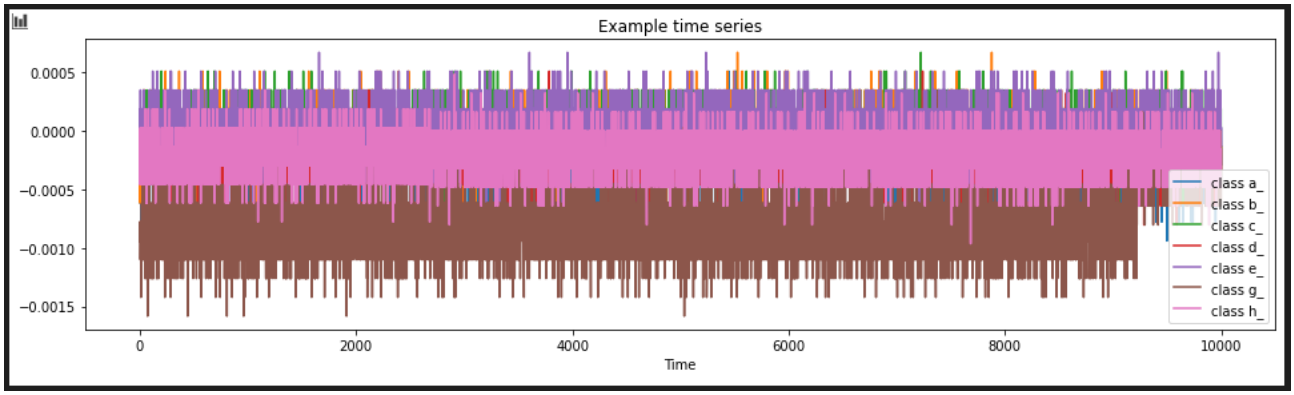
Our objective was to move forward past our ability to classify single instructions to being able to classify multiple instructions in random patterns in the timeseries. We want to be able to classify each instruction in a time series, but we aren't sure the best way to divide up the series into each instruction, so we are going to just start with experimenting on creating a training set containing a subset of instructions randomly distributed over the time series and testing our ability to identify which instructions were present. We changed the way we have been collecting data to accommodate this change, first by generating a text file of assembly instructions using a python script, and second by using our Matlab program to deploy each instruction set at a time.

Past Week Accomplishments

- New data
    - Created a new data capture framework to expand on the combinations captured up to 952 by including 2 opcodes that randomly alternate up to 12 times to train ML in order to recognize opcodes in context.
        - Facing issues and currently debugging.
        - Involved modifying MATLAB script to control new firmware setup
- Machine-Learning
    - Designed a new machine learning model that predicts whether or not a given training example contains a certain opcode. This model is different from previous models in that the output can have more than one opcode predicted. This is simple to implement by adding a

softmax as the output layer. Then the outputs for each opcode can just be interpreted as the probability of the training example containing the opcode.

- Python
    - Evan - There was an issue in the way the data was being processed for machine learning. The pandas dataframes were of unequal sizes (see how class g_ is shorter wave) and these should all be 2000 long, so there is another issue in the program (its being transposed somewhere but unclear why or where). However, the machine learning on this dataset performs very well with ~95% accuracy so that at least tells us that there is some amplitude difference between the first datapoint in each time series for different instructions, so after the issue is fixed we should be able to expect some success in the classification again).




```
[11]
from sklearn.pipeline import make_pipeline

# with sktime, we can write this as a pipeline
from sktime.transformations.panel.reduce import Tabularizer

classifier = make_pipeline(Tabularizer(), RandomForestClassifier())
classifier.fit(X_train, y_train)
classifier.score(X_test, y_test)

0.9908333333333333
```

Pending Issues

- Expanded data collection program needs to select data in a combination that's greater than what we can send in a single UART transaction.
    - Board can miss the second UART transaction causing failure.
    - Need to expand program to successfully capture both UART transmissions to expand data capture.

Individual Contributions

Team Member	Contribution	Weekly Hours	Total Hours
Noah Berthusen	Debugging in TLA and ML model designing	5	19
Matthew Campbell	Designing new training sets in TLA	1	9
Cristian George	UART Branch investigation	3	14
Evan McKinney	ML model debugging	3	18
Jacob Vaughn	Setting up more instructions to classify	3	13
Jesse Knight	Modifying STM firmware	6	22

Plans for Coming Week

- Jesse: Data Collection
    - Get UART data transfer working
    - Improve data collection speed
    - Build new antenna
- Cristian, Jake: UART Transmission Debug
    - Look through modified data capture code.
    - Identify method to ensure successful transmission of two UART Signals prior to execution.

- ML Team (Evan, Noah):
    - Achieve high accuracy classification on set of multiclass repeated instructions and multiclass randomly distributed instructions