Lt David Crow

Activity Report 5

30 June 2019 – 20 July 2019

Week 1

* Attended Maj Juan Jurado’s dissertation defense
  + Interesting stuff, but I’m not sure of its relevance to my own work
  + It concerns how a navigation system might filter our/modify/fix those sensors that seem to give incorrect information
    - I suppose this is kind of like an IDS
* Reviewed Capt Stone’s dissertation in greater detail
  + His semantic analysis process (chapter five) seems to be the same as the signal clustering CSCE 823 project
    - Will have to consider a different project
* Will speak with Dr. Borghetti about one-shot learning for the CSCE 823 project
  + Essentially, can it identify whether these two sets of data are the same or not?
    - We could modify this a bit — can one-shot learning identify whether these two sets of data come from the same car? Each example/test instance could be 100 bytes of data, for example

Week 2

* Reviewed a few articles
* Took Spanish test and completed LEAP application
* Wrote and submitted CSCE 823 CAN data project proposal
  + Can DL effectively determine which vehicle generated some segment of CAN data?

Week 3

* Signed up for a Cyber ANiMaL presentation slot on 19 August
  + Might have something substantial for a thesis update
  + Instead, likely going to present deep learning project
* Reviewed more articles (I’m currently at 11)
  + Identified a key article (Tyree et al, 2019)
* Cleaned/updated ORNL data parser
  + Creates all files in one pass
  + Formatted output to allow for easier information consumption
  + Spent significant time debugging Capt Stone’s reverse engineering pipeline
  + Currently doesn’t work on my Windows 10 machine
    - Didn’t work on my MacBook
  + Works correctly now — will clean modified/added code and submit pull request for Capt Stone’s benefit
  + Multi-pipeline seems to work fine now
  + Still need to copy bug fixes into single-pipeline before sending pull request
* Ran all ORNL log files (34 in total) through Capt Stone’s (now working) pipeline and saved nearly 10 GB of output