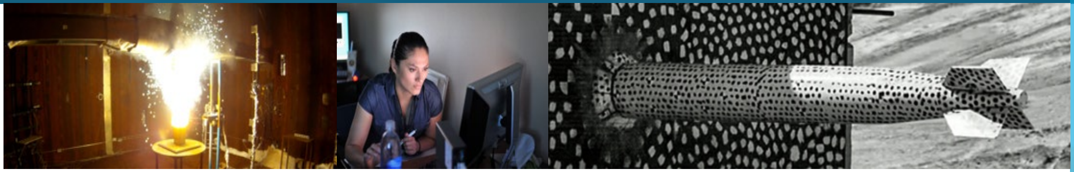
Monitoring DER Integrity using Machine Learning Algorithms on a Single Board Computer



PRESENTED BY

C. Birk Jones, PhD

Sandia National Laboratories is a multimission laboratory managed and operated by National Technology & Engineering Solutions of Sandia, LLC, a wholly owned subsidiary of Honeywell International Inc., for the U.S. Department of Energy's National Nuclear Security Administration under contract DE-NA0003525.

Overview

1. Motivation
2. Experiment Setup
3. Network Sensor
4. Intrusion Detection Analytics
5. Computer Utilization
6. Attack Scenarios

Motivation

U.S. Residential solar PV

- 1.9 Million
- 64.2 GW

PV Inverter Capabilities

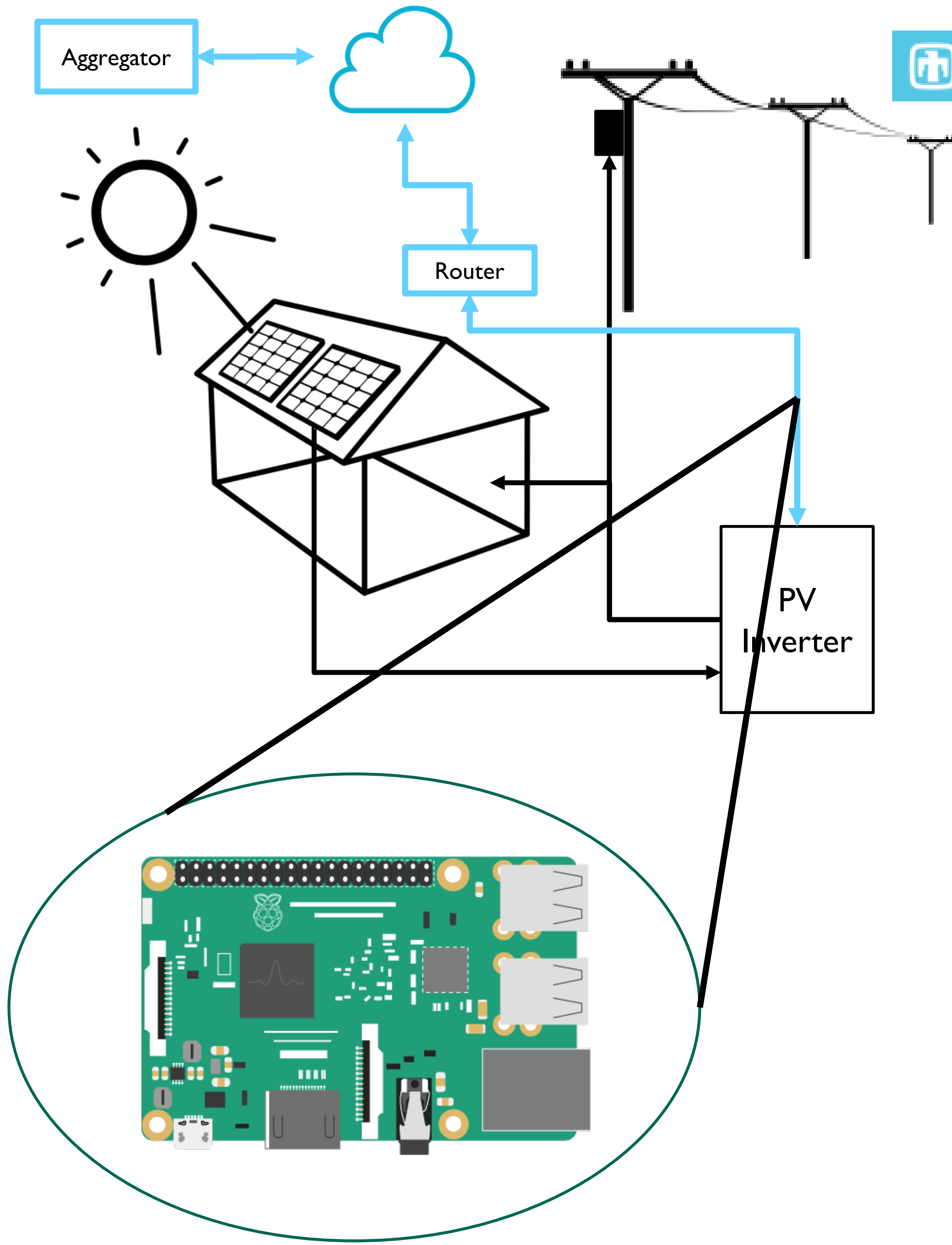
- Reactive/real power support
- Voltage Support
- Frequency support
- Ramp rate control

Centralized Control Issues

- Depend on 3rd party infrastructure
- Control signals are susceptible to:
  - Monitoring
  - Modifications
  - Blocking

Mitigation Strategy

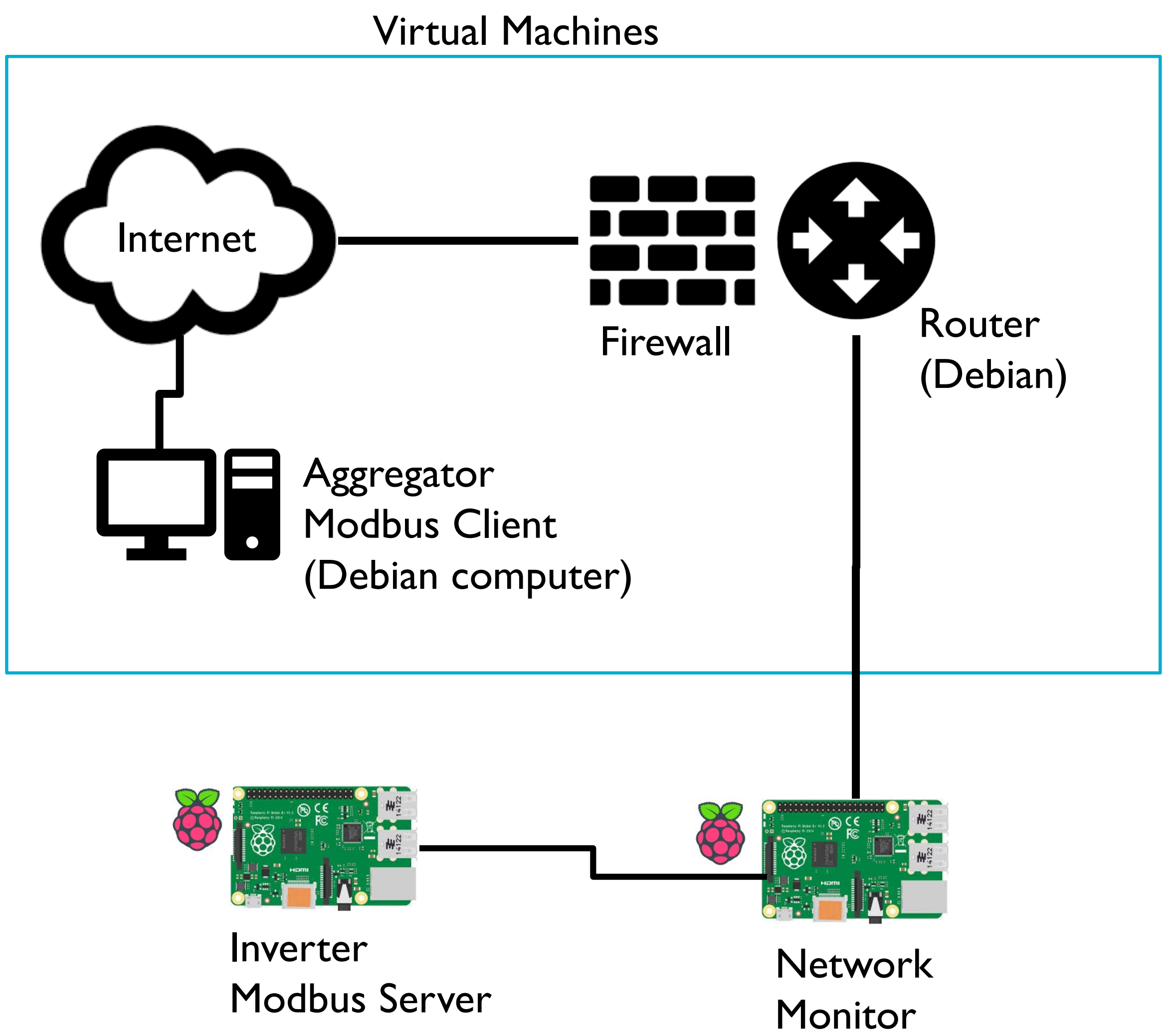
- Advanced monitoring and analytics at the grid edge
- Small, cheap single board computers



Experiment Description

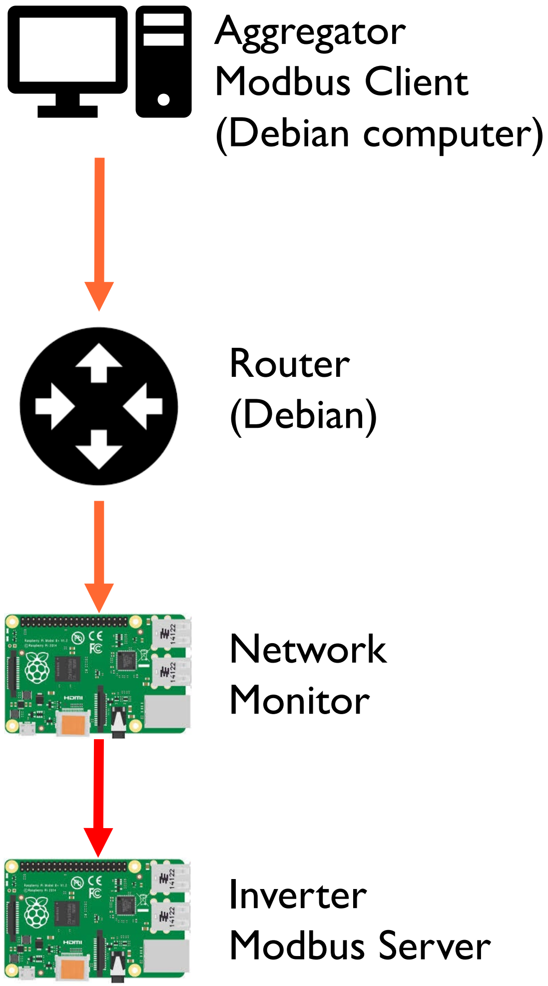
Experiment Setup & Procedures

1. Aggregator
    1. Modbus TCP/IP Client
2. Local Area Network Router
    1. Internet Connection
    2. Firewall
    3. Local Area Network Management
3. Network Monitor
    1. Packet Capture
    2. Intrusion Detection
4. Inverter
    1. Modbus TCP/IP Server



Experiment Procedures

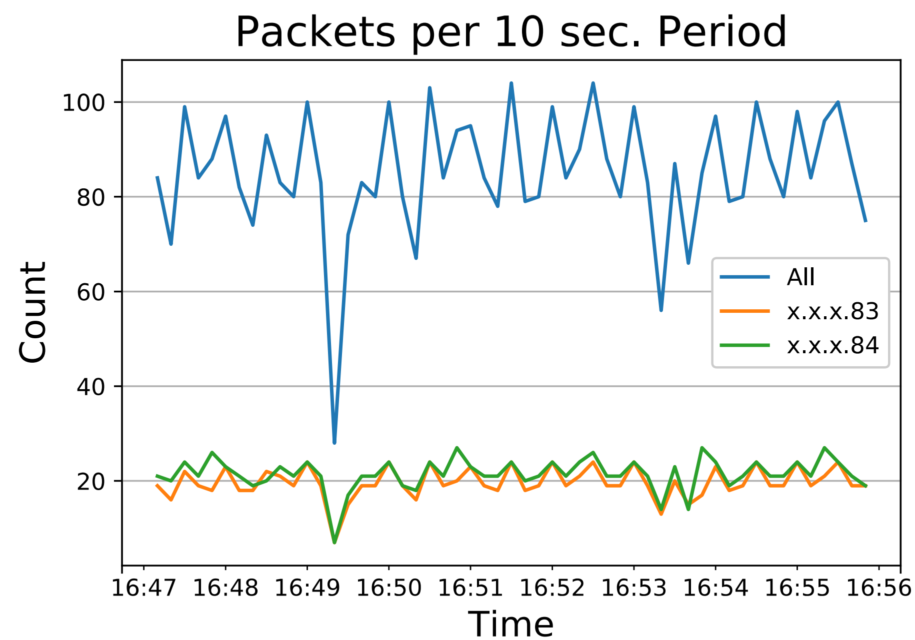
1. Send Messages
   1. Modbus TCP/IP Commands
2. Monitor Messages
   1. Capture Packets
   2. Storage Packet Information
3. Perform Analytics
   1. Intrusion Detection Algorithms
4. Evaluate Computer Operations
   1. Packet Capture
   2. Analysis –Training
   3. Analysis –Detection

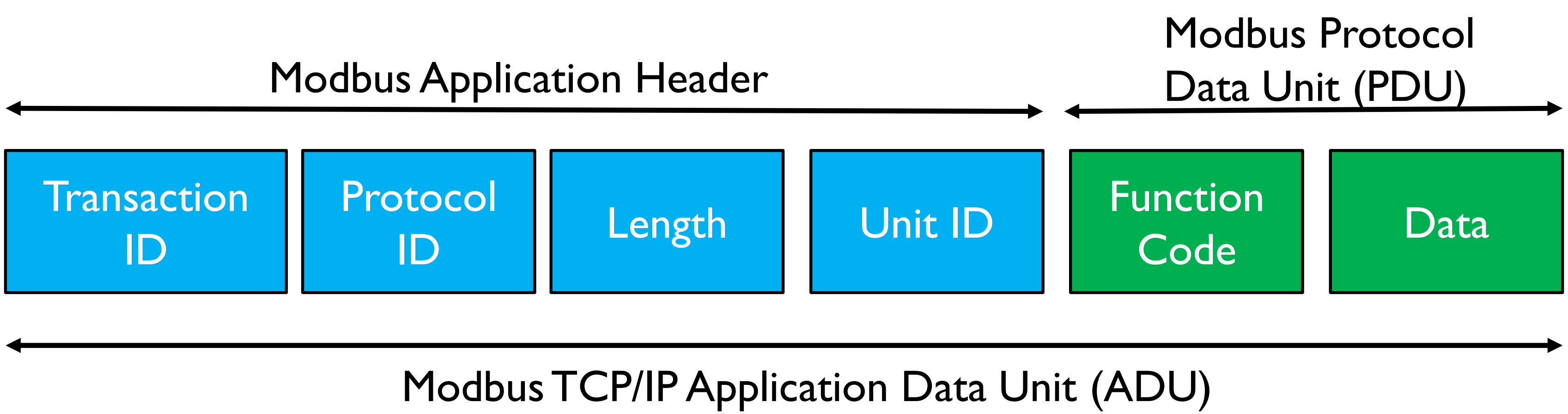


Network Sensor

Packet Capture & Inspection Tools

1. Python Packages
   1. scapy
   2. pcapy
2. Packet Types
   1. TCP/IP
   2. ICMP (ping)
   3. Address Resolution Protocol (ARP)
   4. Modbus TCP/IP

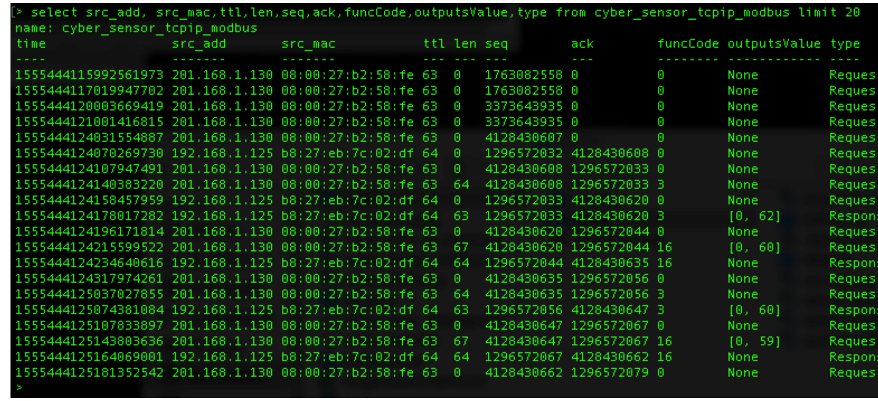




Packet Storage & Access

1. Database
   a. Influxdb (www.influxdata.com)
   b. Open-Source Time Series
   c. Written in Go
      a. High Availability
         ◦ Storage
         ◦ Retrieval
2. Python Queries
   a. Define Query:
      a. query = "select * from ‘xxx’ where time >= now() - 10s"
   b. Get Data
      a. df = client.query(query).get_points(measurement=‘xxx’)
3. Graphical Interface
   a. Grafana (grafana.com)
   b. Open-Source
   c. Graphs numeric time-series data

InfluxDB Terminal Query



```
> select src_add, src_mac,ttl,len,seq,ack,funcCode,outputsValue,type from cyber_sensor_tcpip_modbus limit 20
name: cyber_sensor_tcpip_modbus
time                src_add       src_mac           ttl len seq        ack        funcCode outputsValue type
----                -------       -------           --- --- ---        ---        -------- ------------ ----
1555444115992561973 201.168.1.130 08:00:27:b2:58:fe 63  0   1763082558 0          0        None         Request
1555444117019947702 201.168.1.130 08:00:27:b2:58:fe 63  0   1763082558 0          0        None         Request
1555444120003669419 201.168.1.130 08:00:27:b2:58:fe 63  0   3373643935 0          0        None         Request
1555444121001416815 201.168.1.130 08:00:27:b2:58:fe 63  0   3373643935 0          0        None         Request
1555444124031554887 201.168.1.130 08:00:27:b2:58:fe 63  0   4128430607 0          0        None         Request
1555444124070269730 192.168.1.125 b8:27:eb:7c:02:df 64  0   1296572032 4128430608 0        None         Request
1555444124107947491 201.168.1.130 08:00:27:b2:58:fe 63  0   4128430608 1296572033 0        None         Request
1555444124140383220 201.168.1.130 08:00:27:b2:58:fe 63  64  4128430608 1296572033 3        None         Request
1555444124158457959 192.168.1.125 b8:27:eb:7c:02:df 64  0   1296572033 4128430620 0        None         Request
1555444124178017282 192.168.1.125 b8:27:eb:7c:02:df 64  63  1296572033 4128430620 3        [0, 62]      Response
1555444124196171814 201.168.1.130 08:00:27:b2:58:fe 63  0   4128430620 1296572044 0        None         Request
1555444124215599522 201.168.1.130 08:00:27:b2:58:fe 63  67  4128430620 1296572044 16       [0, 60]      Request
1555444124234640616 192.168.1.125 b8:27:eb:7c:02:df 64  64  1296572044 4128430635 16       None         Response
1555444124317974261 201.168.1.130 08:00:27:b2:58:fe 63  0   4128430635 1296572056 0        None         Request
1555444125037027855 201.168.1.130 08:00:27:b2:58:fe 63  64  4128430635 1296572056 3        None         Request
1555444125074381084 192.168.1.125 b8:27:eb:7c:02:df 64  63  1296572056 4128430647 3        [0, 60]      Response
1555444125107833897 201.168.1.130 08:00:27:b2:58:fe 63  0   4128430647 1296572067 0        None         Request
1555444125143803636 201.168.1.130 08:00:27:b2:58:fe 63  67  4128430647 1296572067 16       [0, 59]      Request
1555444125164069001 192.168.1.125 b8:27:eb:7c:02:df 64  64  1296572067 4128430662 16       None         Response
1555444125181352542 201.168.1.130 08:00:27:b2:58:fe 63  0   4128430662 1296572079 0        None         Request
>
```

Grafana Visualization

Intrusion Detection Analytics

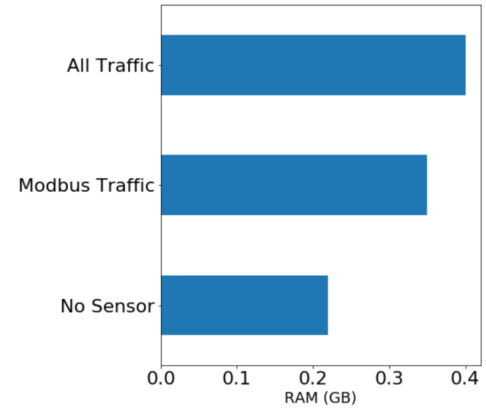
Machine Learning Algorithms

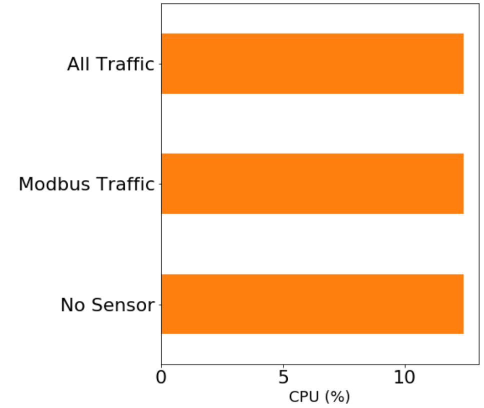
1. Adaptive Resonance Theory
   a. Unsupervised Artificial Neural Network
   b. Comparison and recognition layers
   c. https://github.com/cbirkj/art-python
2. One-Class Support Vector Machine
   1. Unsupervised Machine Learning
   2. Creates a multi-dimensional hyperplane
   3. https://scikit-learn.org/stable/modules/svm.html
3. Autoencoder
   1. Unsupervised Deep Neural Network
   2. Feedforward, non-recurrent neural network
   3. Implemented using:
      1. Keras
      2. Tensorflow

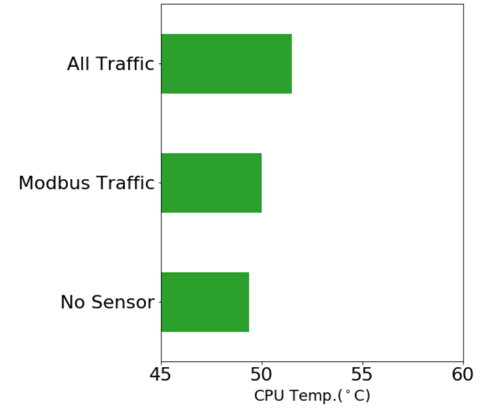
Computer Utilization

Computer Resources – Network Sensors

1. Random Access Memory
   a. Baseline = ~ 23%
   b. Max = ~40% of total
2. Central Processing Unit (CPU)
   a. Each use ~12%
3. CPU Temperature
   a. Baseline = 49.4°C
   b. Max = 51.5°C

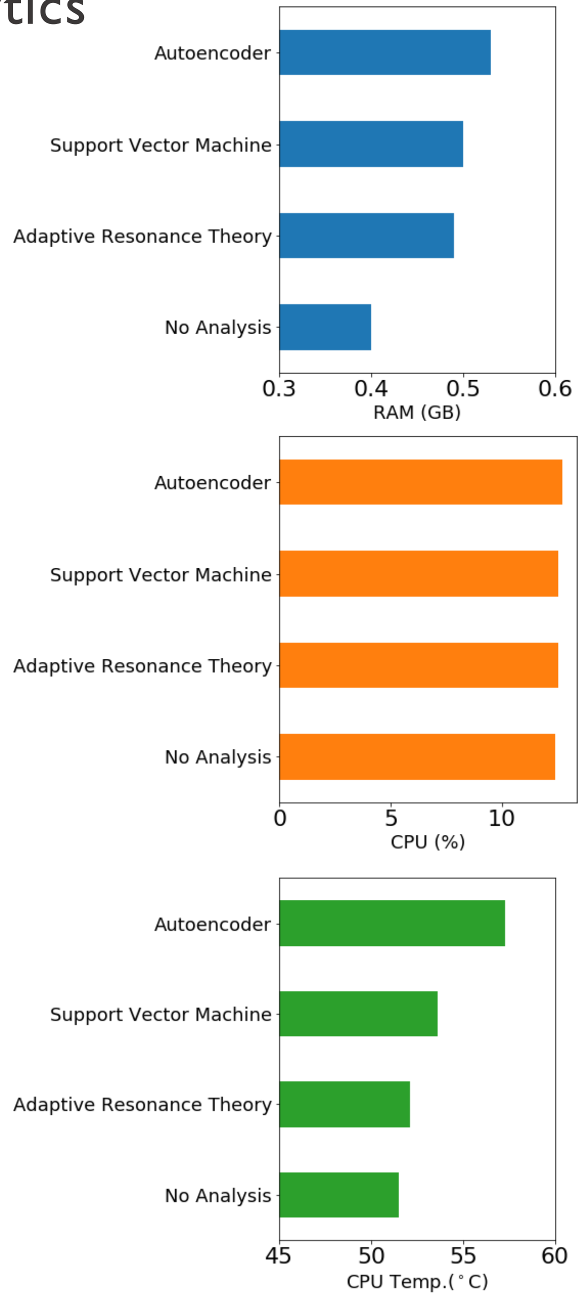






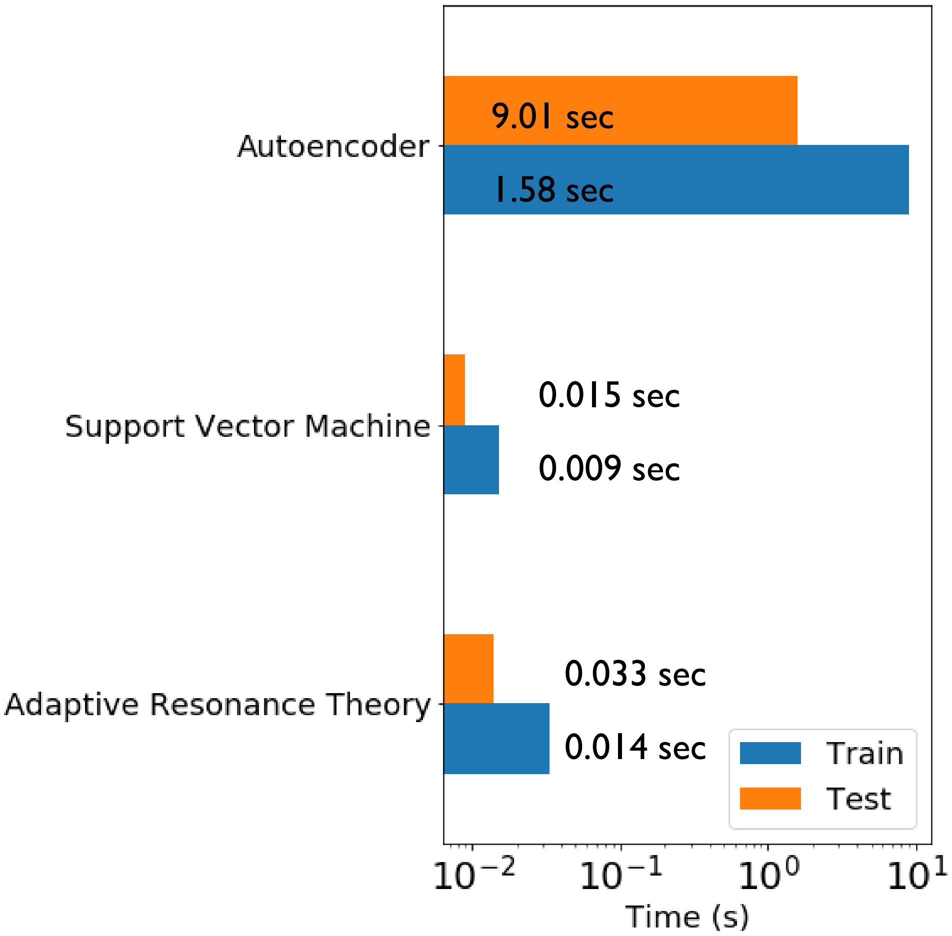
Computer Resource – Sensor + Analytics

1. Random Access Memory
   a. Min. = ~ 40%
   b. Max = ~55% of total
2. Central Processing Unit (CPU)
   a. Min = ~12.4%
   b. Max = 12.7%
3. CPU Temperature
   a. Min. = 51.5°C
   b. Max = 57.3°C



Algorithm Train & Test Time

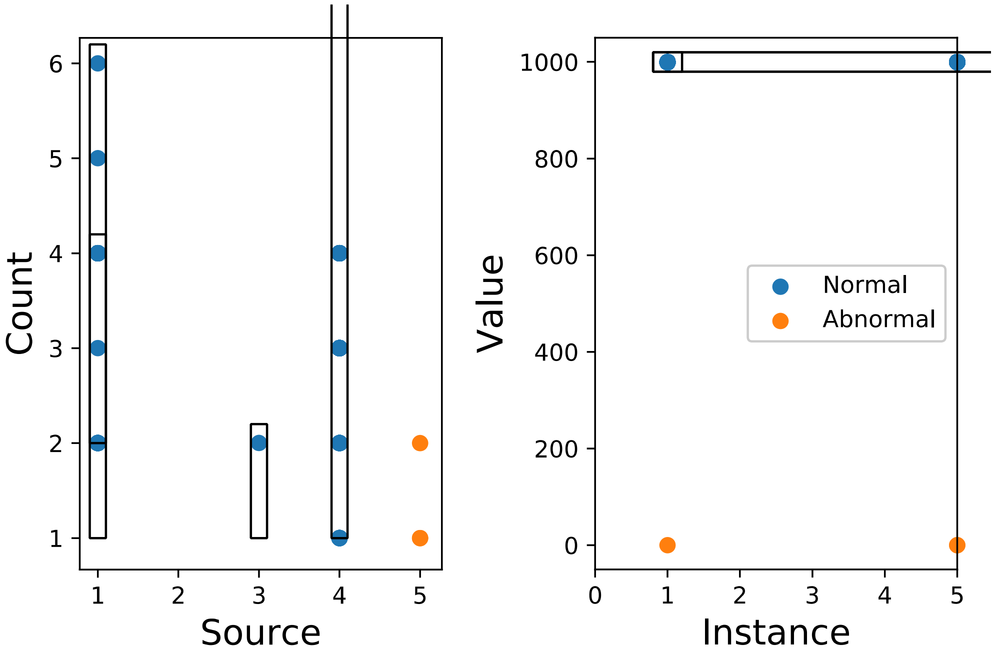
1. Batch Learning
    1. Learn on entire data set
2. On-Line Learning
    1. Learn when data available in sequential order
    2. Update predictor
3. Experiment used On-Line Learning
4. Adaptive Resonance Theory
    1. Performed well w/ On-Line Learning
5. Support Vector Machine
    1. Fast but hard to learn in on-line learning
6. Autoencoder
    1. Did not perform well
    2. Better with Batch Learning



Intrusion Detection

Network Based Intrusion Detection (Example)

1. Adaptive Resonance Theory
   a. Create hyperboxes around the data
   b. Violations/anomalies when data not inside boxes
2. Example Features
   1. Count - Frequency
   2. Source – IP address where signal originated
   3. Instance – Data point
   4. Value - Value of point



Summary

Conclusion

1. Single Board Computers
   a. Provide Bump-in-the-Wire Monitoring
   b. Capture Packets (multiple types)
   c. Inspect Packets
   d. Store & View Packets
   e. Analyze Packets
2. Sensor
   a. 40% of RAM
   b. 12% CPU
   c. 51.5 °C
3. Intrusion Detection Analytic
   a. Adaptive Resonance Theory
      - Lowest RAM, CPU, and Temp
      - Best on-line learner

Questions