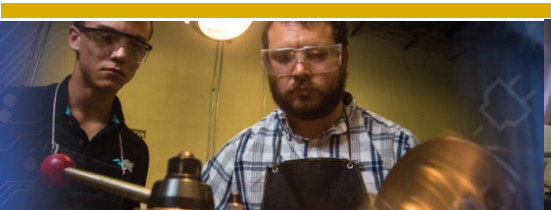
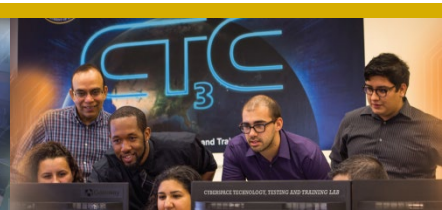
AI Platform for CTAM System

Dr. Himanshu Upadhyay
Co-Principle Investigator
Florida International University
upadhyay@fiu.edu

Research Funded by
OSD Test Resources Management Center
T&E S&T Program, Cyberspace Test Technology

FLORIDA INTERNATIONAL UNIVERSITY

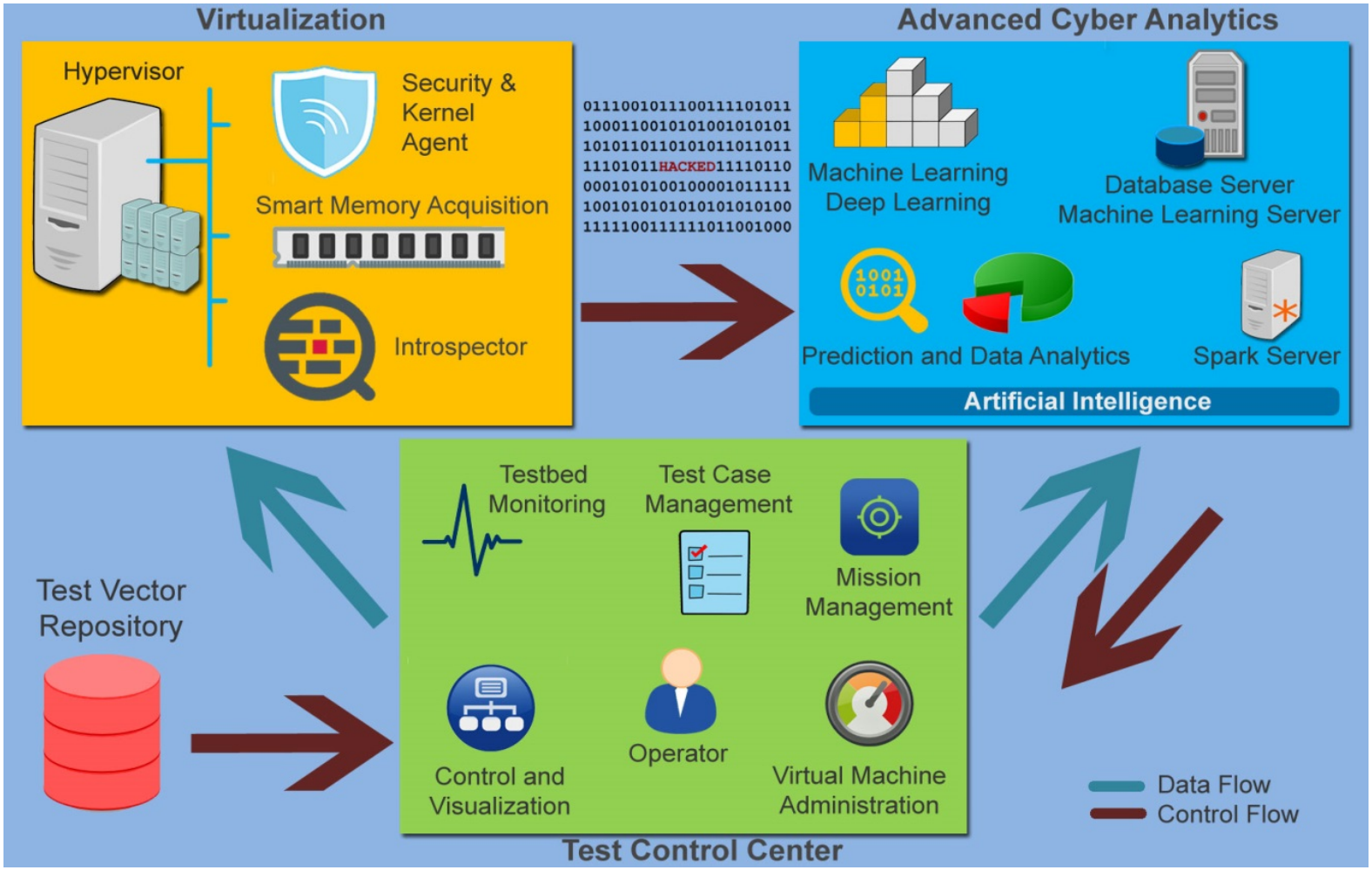




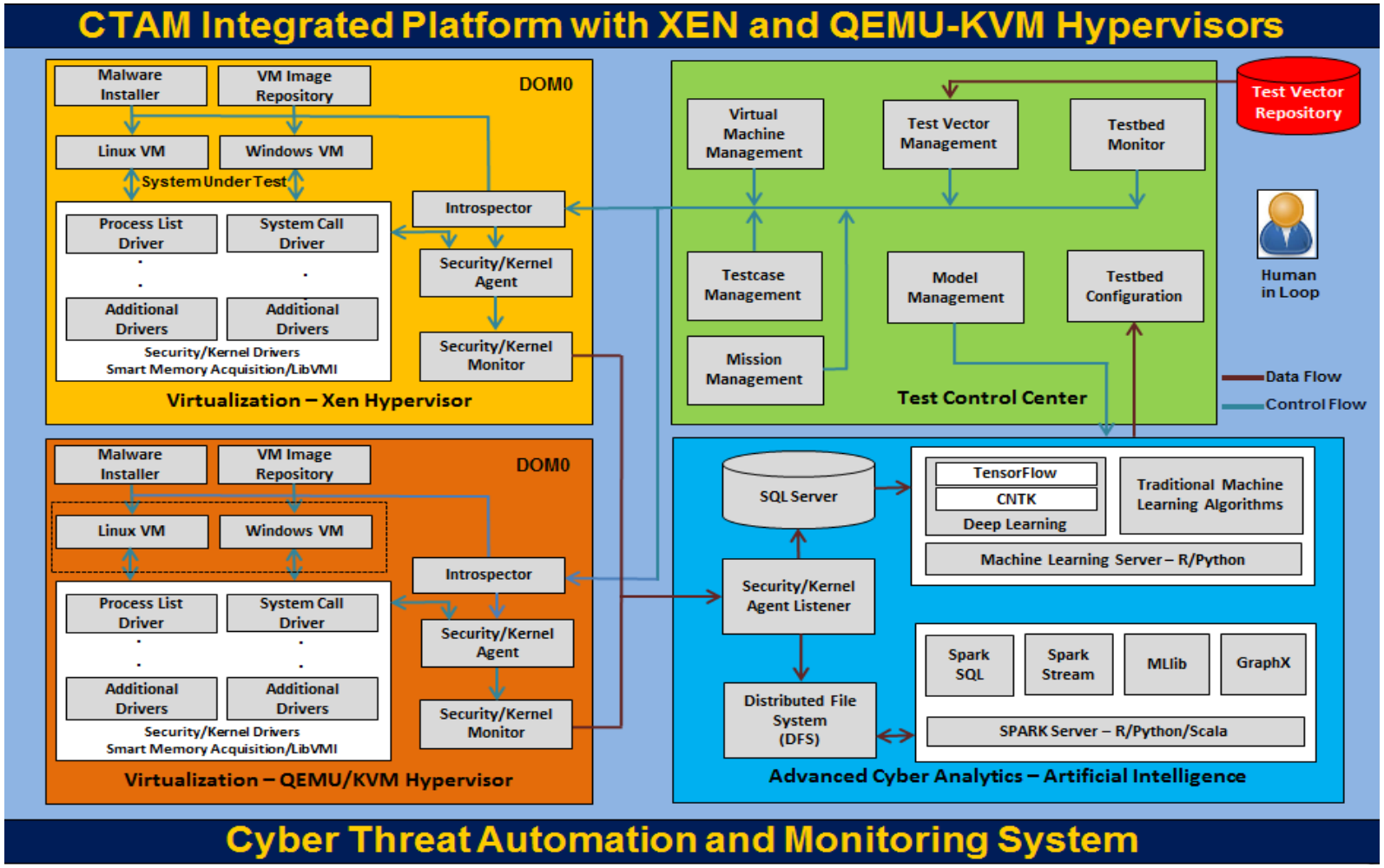
Project Description

- CTAM is a Cyberspace Test Technology for T & E purposes to monitor and analyze behavior during cyber attacks and also the impact on the current mission
- CTAM is based on fine-grained introspection of kernel data structures, data collection and advanced cyber analytics using Artificial Intelligence / Machine Learning techniques
- CTAM consists of three platforms:
  - Virtualization
  - Advanced Cyber Analytics
  - Test Control Center

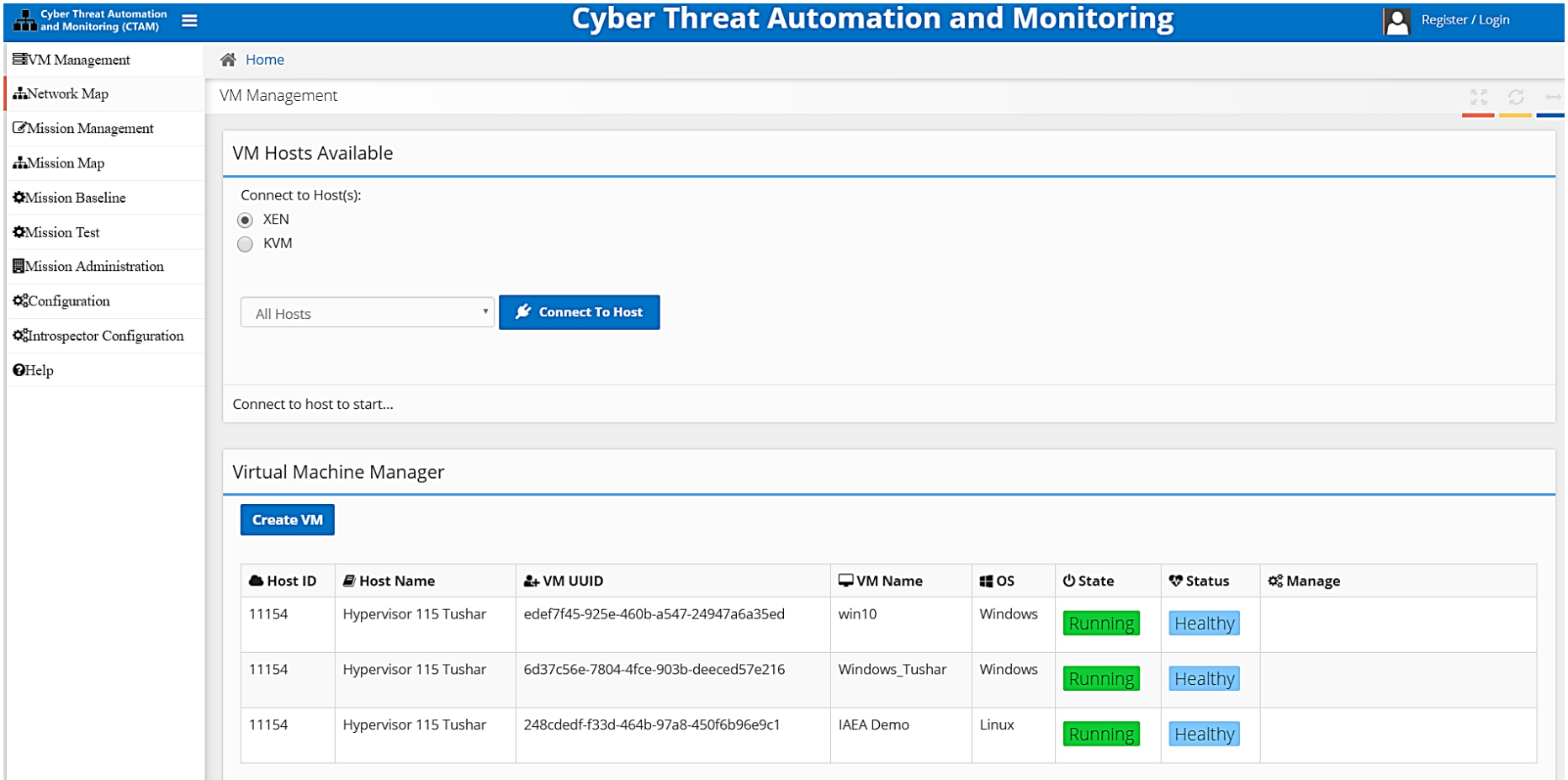
CTAM System Diagram



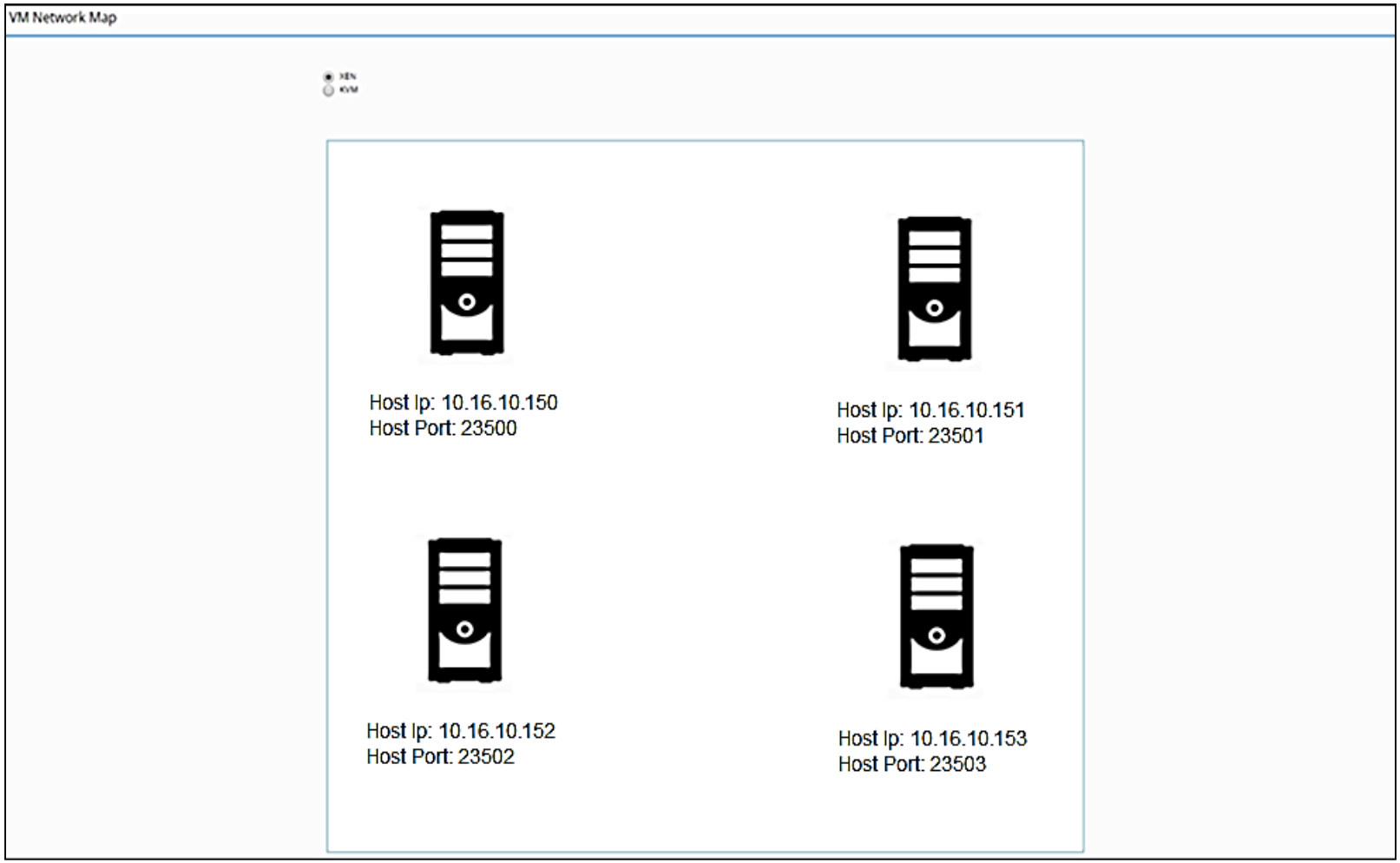
System Block Diagram



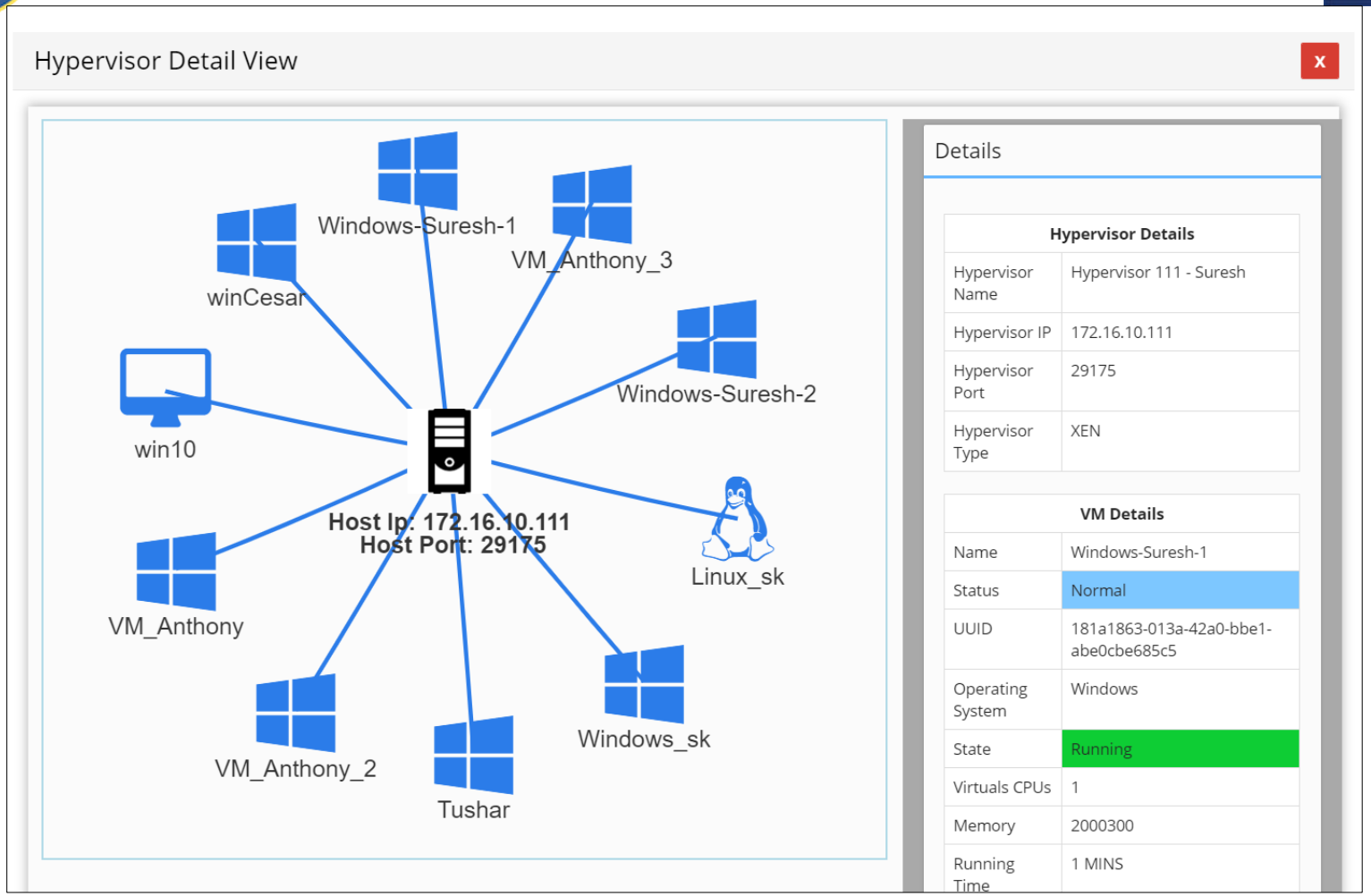
VM Management



VM Management



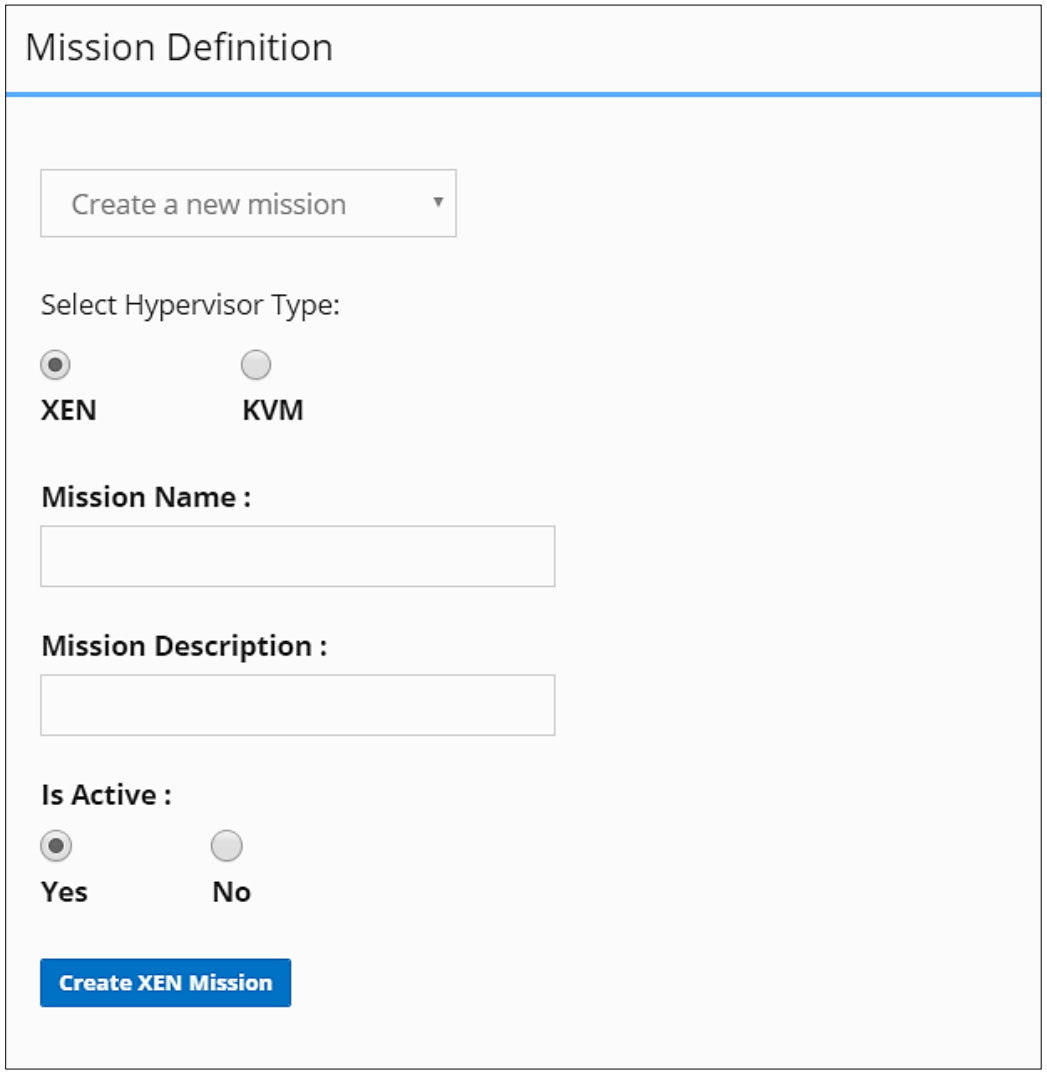
Network Map


Hypervisor Detail View
x
Windows-Suresh-1
VM_Anthony_3
winCesar
Windows-Suresh-2
win10
Host Ip: 172.16.10.111
Host Port: 29175
Linux_sk
VM_Anthony
Windows_sk
VM_Anthony_2
Tushar
Details
Hypervisor Details
Hypervisor Name
Hypervisor 111 - Suresh
Hypervisor IP
172.16.10.111
Hypervisor Port
29175
Hypervisor Type
XEN
VM Details
Name
Windows-Suresh-1
Status
Normal
UUID
181a1863-013a-42a0-bbe1-abe0cbe685c5
Operating System
Windows
State
Running
Virtuals CPUs
1
Memory
2000300
Running Time
1 MINS

SUT Mission

- **SUT Mission Management**
  - SUT Mission Definition
  - SUT Mission Subsystems
  - SUT Mission Module
- **SUT Mission Map**
  - SUT Map View
  - SUT Mission System Summary
- **SUT Mission Baseline**
- **SUT Mission Test**
- **SUT Mission Administration**

Mission Definition



Mission Subsystems

FIU Applied Research Center

Mission SubSystem

Add Mission SubSystem

	MissionSystemID	MissionID	MissionSystemName	MissionSystemDescription	HostID	VMStatusID	IsActive
Edit Delete	159	114	GPS	GPS	11155	806	1
Edit Delete	161	114	Application Server	Application Server	11155	807	1
Edit Delete	162	114	DBS	DBS	11155	808	1
Edit Delete	157	114	Flight Path Generator - SUT	Flight Path Generator - SUT	11155	809	1
Edit Delete	158	114	Weather	Weather	11155	810	1

Mission Modules

Mission Module

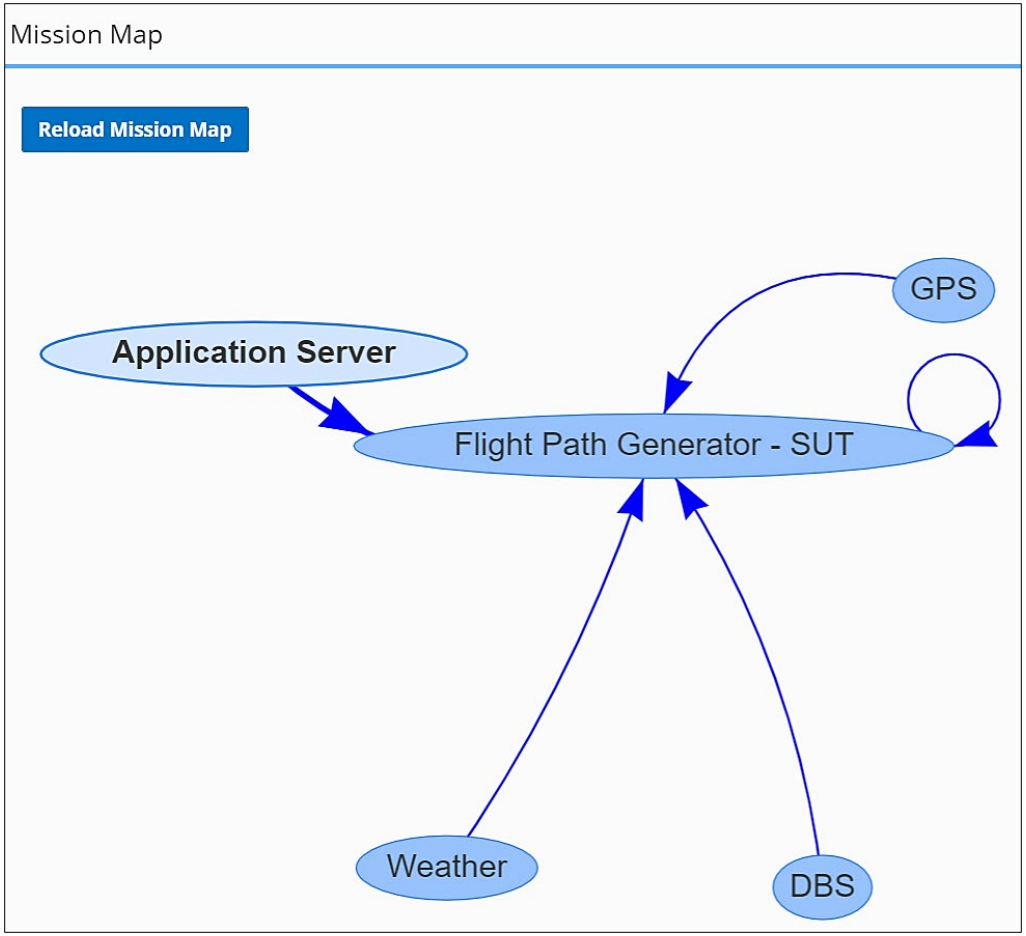
Select Mission Subsystem:

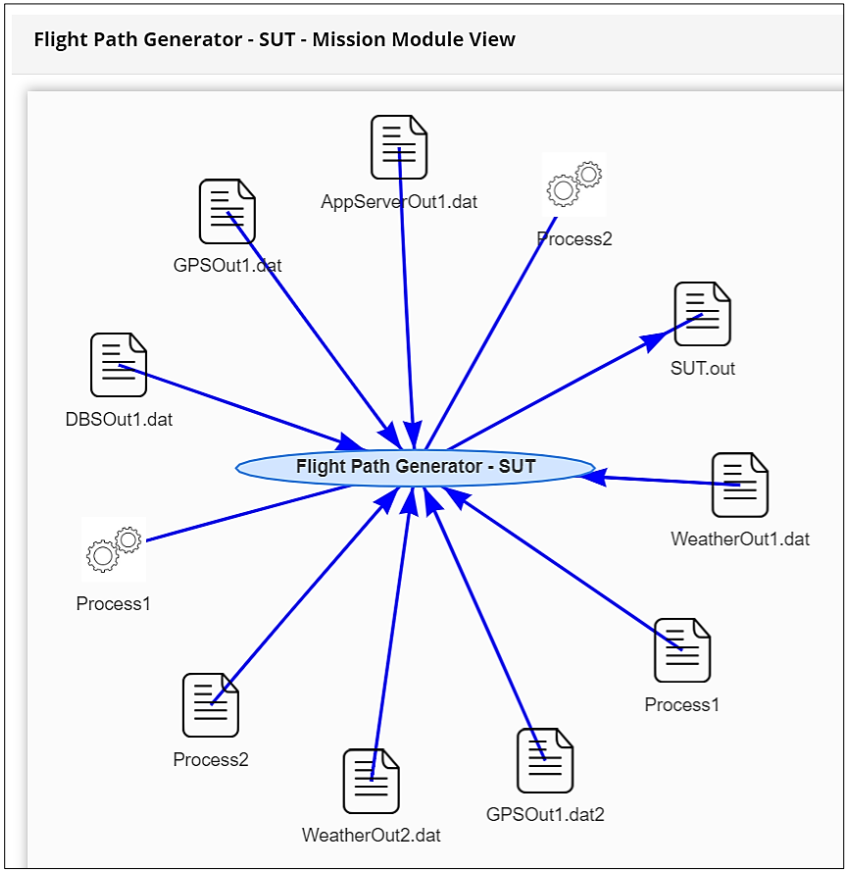
Flight Path Generator - SUT

Add Mission Module

	MissionModuleID	MissionSystemID	ModuleName	ModuleDescription	FileName	FilePath	ModuleTy
Edit Delete	169	157	Input	Input	GPSOut1.dat		1
Edit Delete	170	157	Input	Input	GPSOut1.dat2		1
Edit Delete	171	157	Input	Input	WeatherOut1.dat		1
Edit Delete	172	157	Input	Input	WeatherOut2.dat		1
Edit Delete	173	157	Input	Input	DBSOut1.dat		1
Edit Delete	174	157	Input	Input	AppServerOut1.dat		1

Mission Map


Mission Map
Reload Mission Map
GPS
Application Server
Flight Path Generator - SUT
Weather
DBS


Flight Path Generator - SUT - Mission Module View
AppServerOut1.dat
Process2
GPSOut1.dat
SUT.out
DBSOut1.dat
Flight Path Generator - SUT
WeatherOut1.dat
Process1
Process1
Process2
GPSOut1.dat2
WeatherOut2.dat

Mission Systems Summary

FIU Applied Research Center

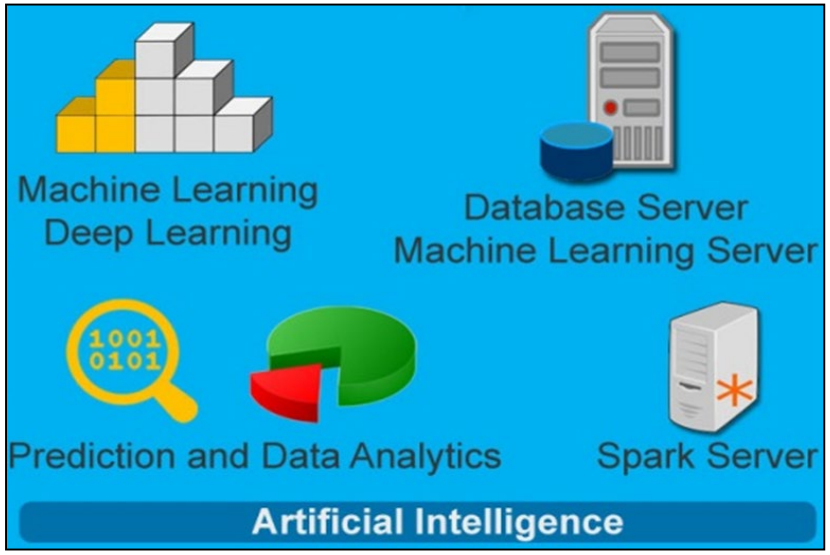
Mission Systems Summary

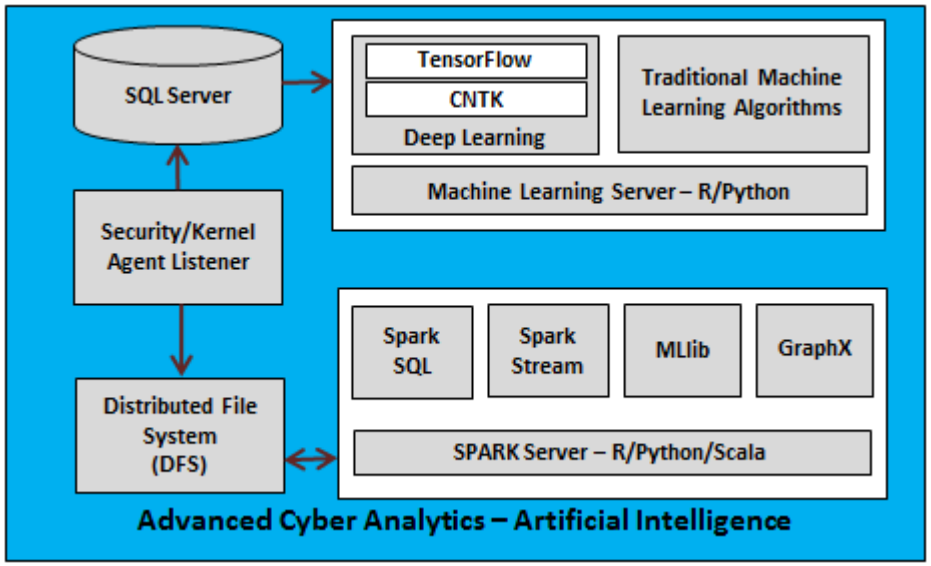
Module ID	Module Name	Module Description	Assigned VM Name	Assigned VM UUID
159	GPS	GPS	Mission_SubSystem_VM	0e74bc6b-9250-4feb-b985-270ea0ec5c2c
161	Application Server	Application Server	Mission_SubSystem_VM4	d429c12e-6a9a-4525-b731-cdeba644f4f9
162	DBS	DBS	Mission_SubSystem_VM3	616688b8-122f-41f4-86d5-7bc6f6766970
157	Flight Path Generator - SUT	Flight Path Generator - SUT	Mission_VM	ef8175c0-fe02-44ee-8d8b-2154569e1b28
158	Weather	Weather	Mission_SubSystem_VM2	7ef00d1c-c0d4-418e-b3dd-ba6ed026d460

MissionID : 114

Advanced Cyber Analytics

- Cyber Analytics module consists of Database server and Machine learning server for in-memory analytics
- Machine learning / Deep Learning models with different algorithms are built using the training data
- Models are used to predict the impact of the test vectors on a specific mission





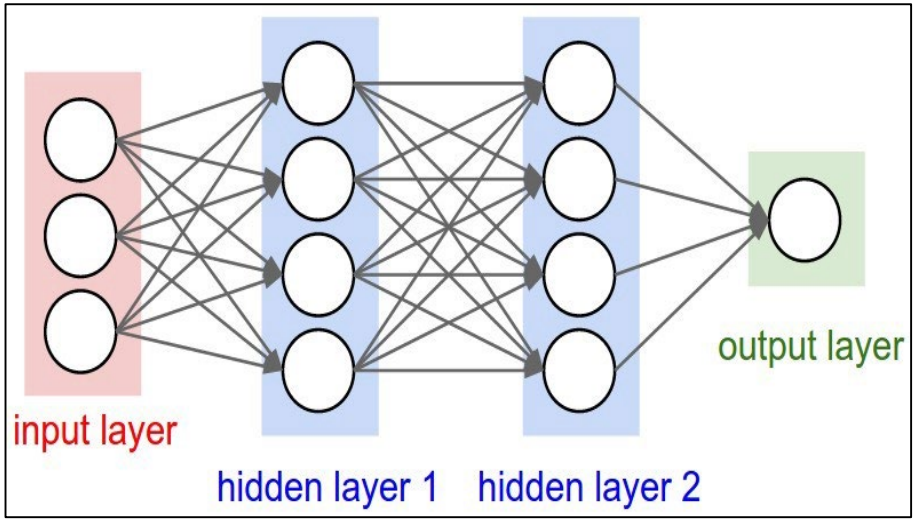
Machine Learning / Deep Learning Algorithms

FIU Applied Research Center

Traditional ML Algorithms	Deep Learning Algorithms
➢ Random Forest ➢ Support Vector Machine ➢ Logistic Regression ➢ Gradient Boosting ➢ Neural Network ➢ Ensemble ➢ One Class SVM ➢ Cosine Vector Similarity	➢ Recurrent Neural Networks ▪ Long Short Term Memory (LSTM) ▪ Auto-Encoders ▪ Bidirectional LSTM ▪ Generative Adversarial Network (GANs)

Deep Learning

- Weights are initialized to random values
- Data is propagated forward to produce a prediction
- The error of that prediction is propagated backwards
- The weights are corrected

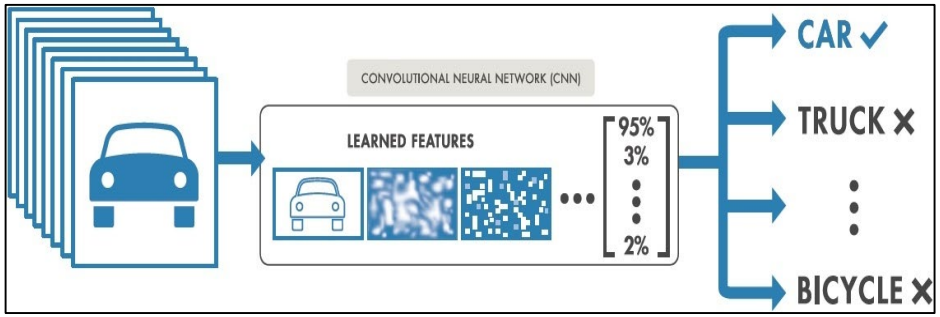


Types of Deep Neural Network

FIU Applied Research Center

Convolutional Neural Networks

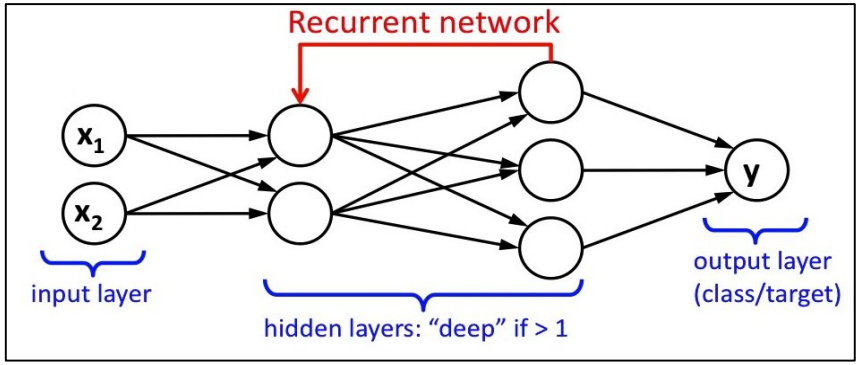
- Primarily used for image data
- Used for detecting features within an image dataset



Reference:
https://www.mathworks.com/solutions/deep-learning/convolutional-neural-network.html

Recurrent Neural Networks

- Primarily used for sequential data
- Used for identifying patterns in sequenced information

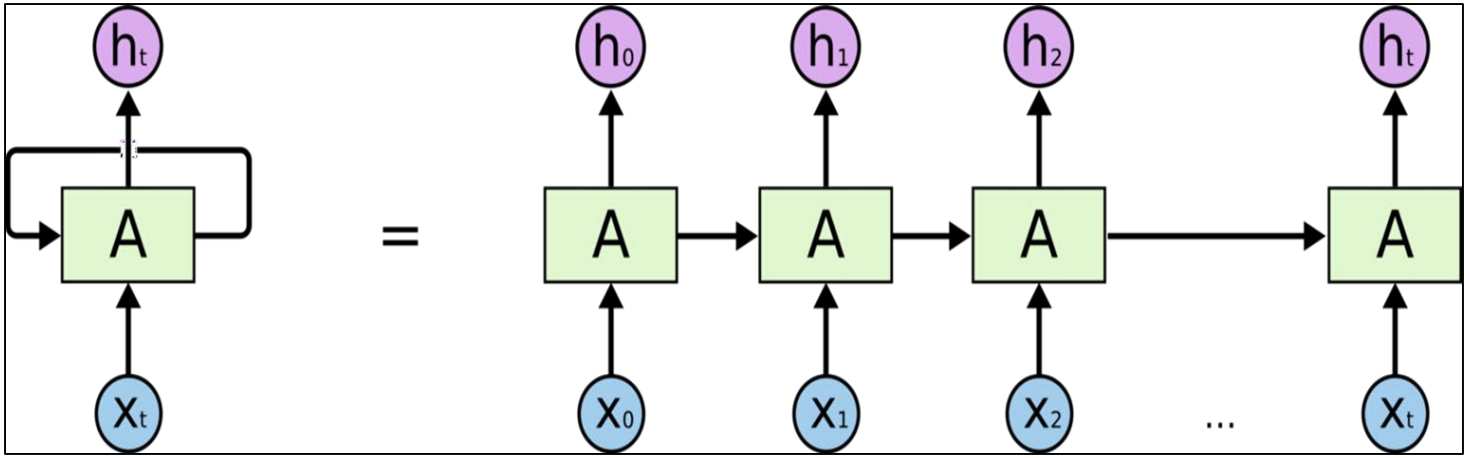


Reference:
https://leonardoaraujosantos.gitbooks.io/artificial-inteligence/content/recurrent_neural_networks.html

RNN – Long Short Term Memory (LSTM)

Recurrent Neural Networks - LSTM

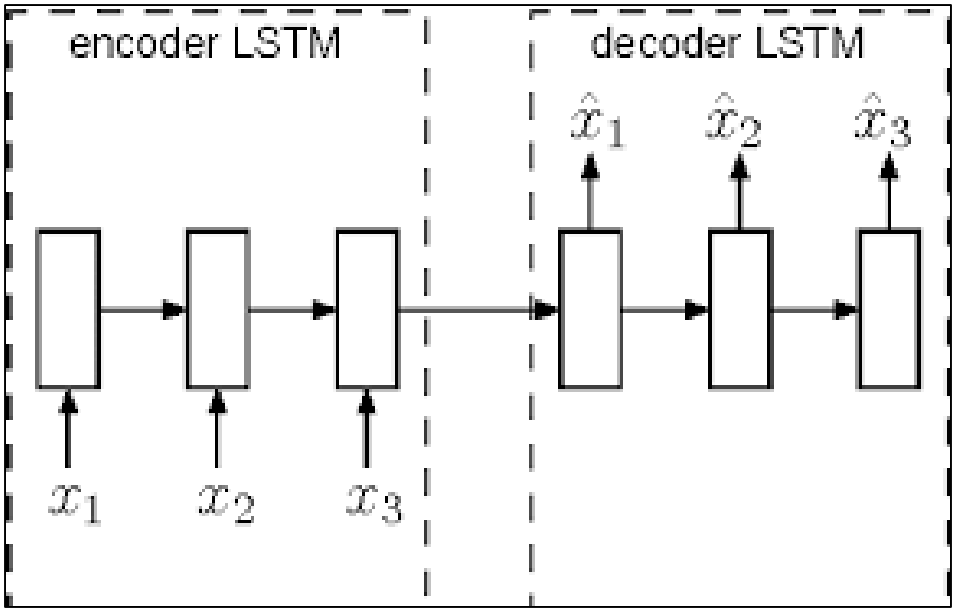
- Since we are dealing with a large amount of sequential data, it would make sense to use an RNN
- Past sequential information can be used to predict future time steps in the series



Reference: http://colah.github.io/posts/2015-08-Understanding-LSTMs/

LSTM - Autoencoders

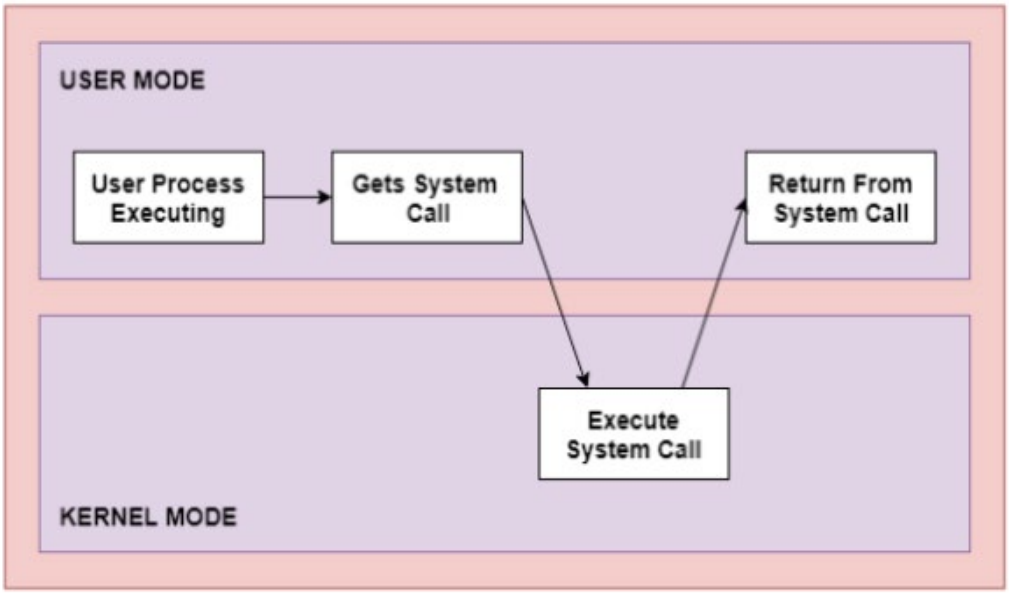
- Learns compressed representations of input data and attempts to reconstruct the original input.
- The encoder LSTM transforms an input sequence into a compressed latent space.
- The decoder LSTM tries to reconstruct the sequence from the encoder's latent vector



https://www.sciencedirect.com/science/article/pii/S0010482518300738

System Call

- System call is an OS kernel functions that work as an entry point to the kernel
- System call provides an interface between an application process and the operating system kernel
- Application programs invokes system calls to pass/retrieve data to the OS kernel



Reference: https://www.tutorialspoint.com/what-are-system-calls-in-operating-system

System Call

Types of System Calls:

- Process Control - These system calls deal with processes such as process creation, process termination etc.
- File Management - These system calls are responsible for file manipulation such as creating a file, reading a file, writing into a file etc.
- Device Management - These system calls are responsible for device manipulation such as reading from device buffers, writing into device buffers etc.
- Information Maintenance - These system calls handle information and its transfer between the operating system and the user program
- Communication - These system calls are useful for inter-process communication. They also deal with creating and deleting a communication connection

Reference: https://www.tutorialspoint.com/what-are-system-calls-in-operating-system

System Call Sequences

- System call sequence is the order in which the system calls are invoked by the application program
- Each application will invoke system calls in a particular sequence to accomplish a specific task
- Table shows the example of the system call sequence

System Call Sequence	System Call Name	System Call Number
1	NtQueryInformationProcess	25
2	NtOpenKey	18
3	NtQueryValueKey	23
4	NtOpenKey	18
5	NtOpenKey	18
6	NtQueryValueKey	23
7	NtClose	15

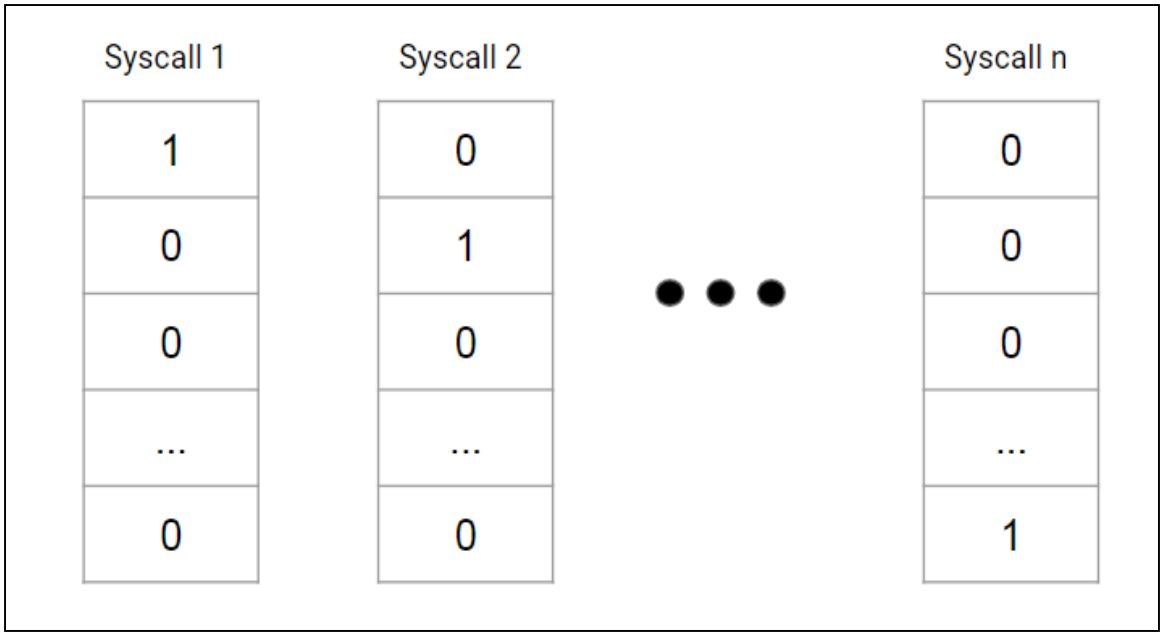
- Test vector can modify the system call sequence to change the application behavior
- Anomaly detection with Deep learning can be used to detect the change in the system call sequence

System Call Analysis

FIU Applied Research Center

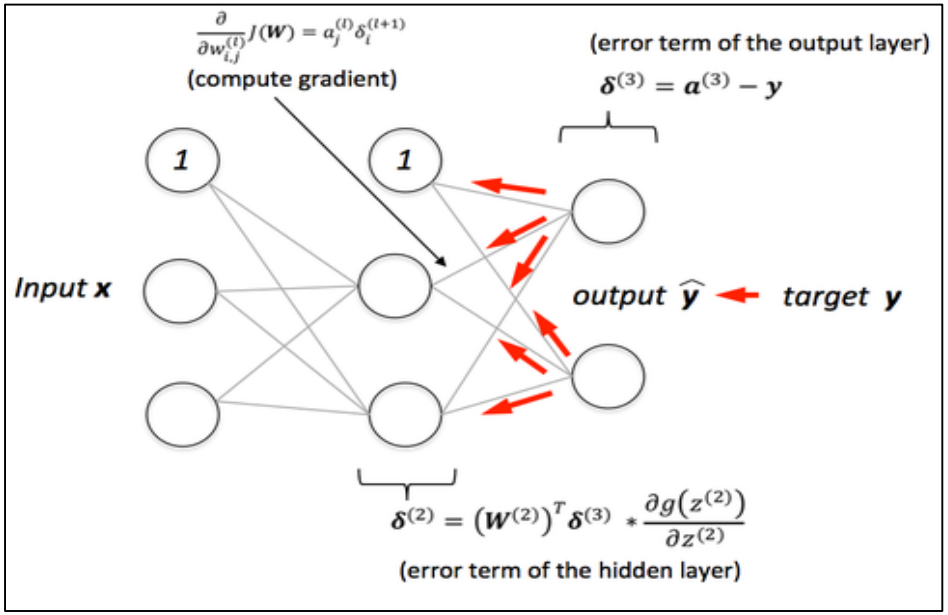
Data Pre-processing:

- Every system call is mapped to a number in a dictionary
- Represent every system call as a one-hot encoded sparse vectors so they may be used in the deep learning algorithm



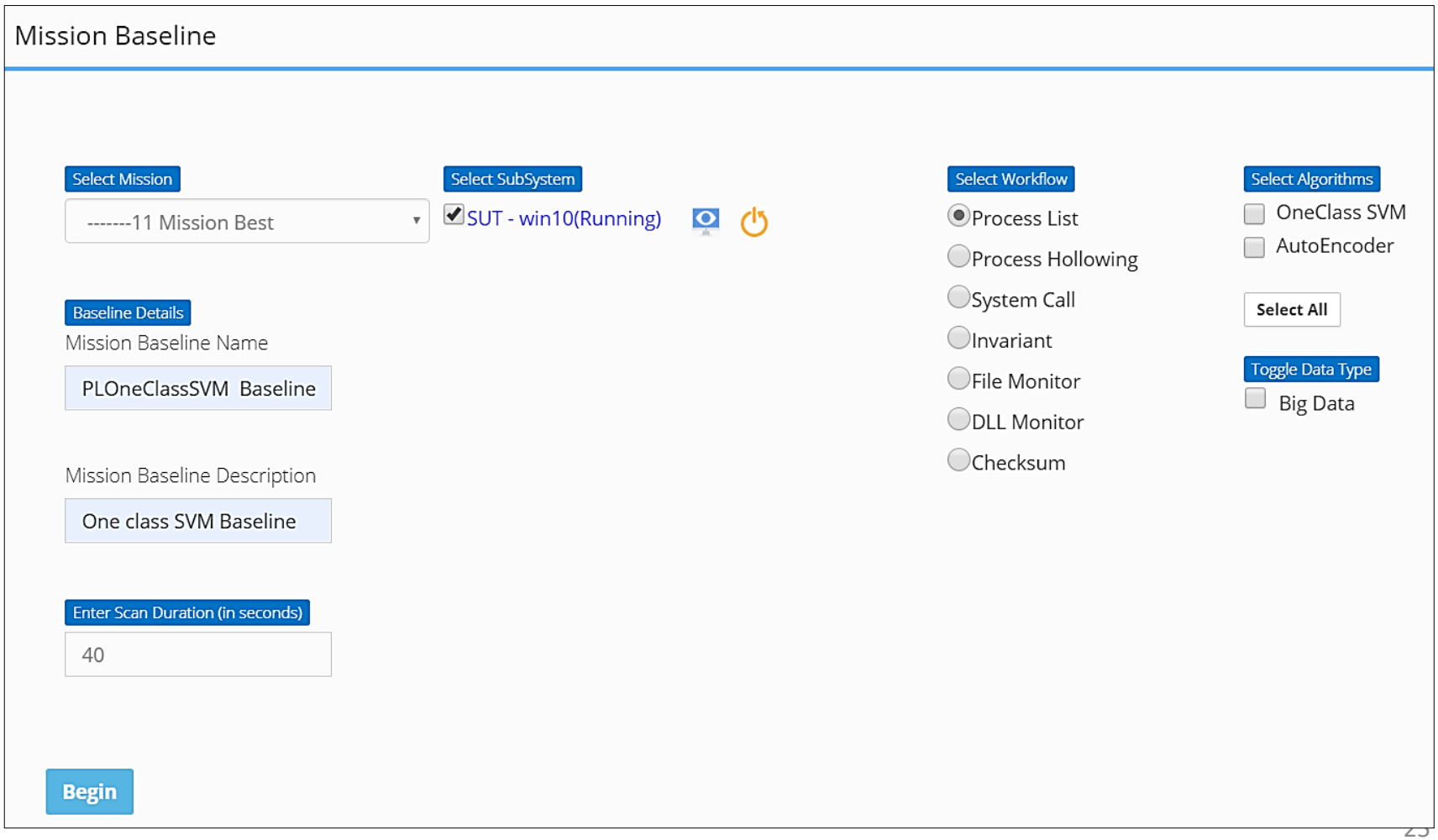
System Call Analysis

- The number of nodes and layers, along with many other hyper-parameters (learning rate, dropout, etc.)
- Weights and bias are initialized to be random
- Backwards propagation and enough training ensures that weights will converge to their optimal values



https://github.com/rasbt/python-machine-learning-book/blob/master/faq/visual-backpropagation/backpropagation.png

Process List – One Class SVM Mission Baseline



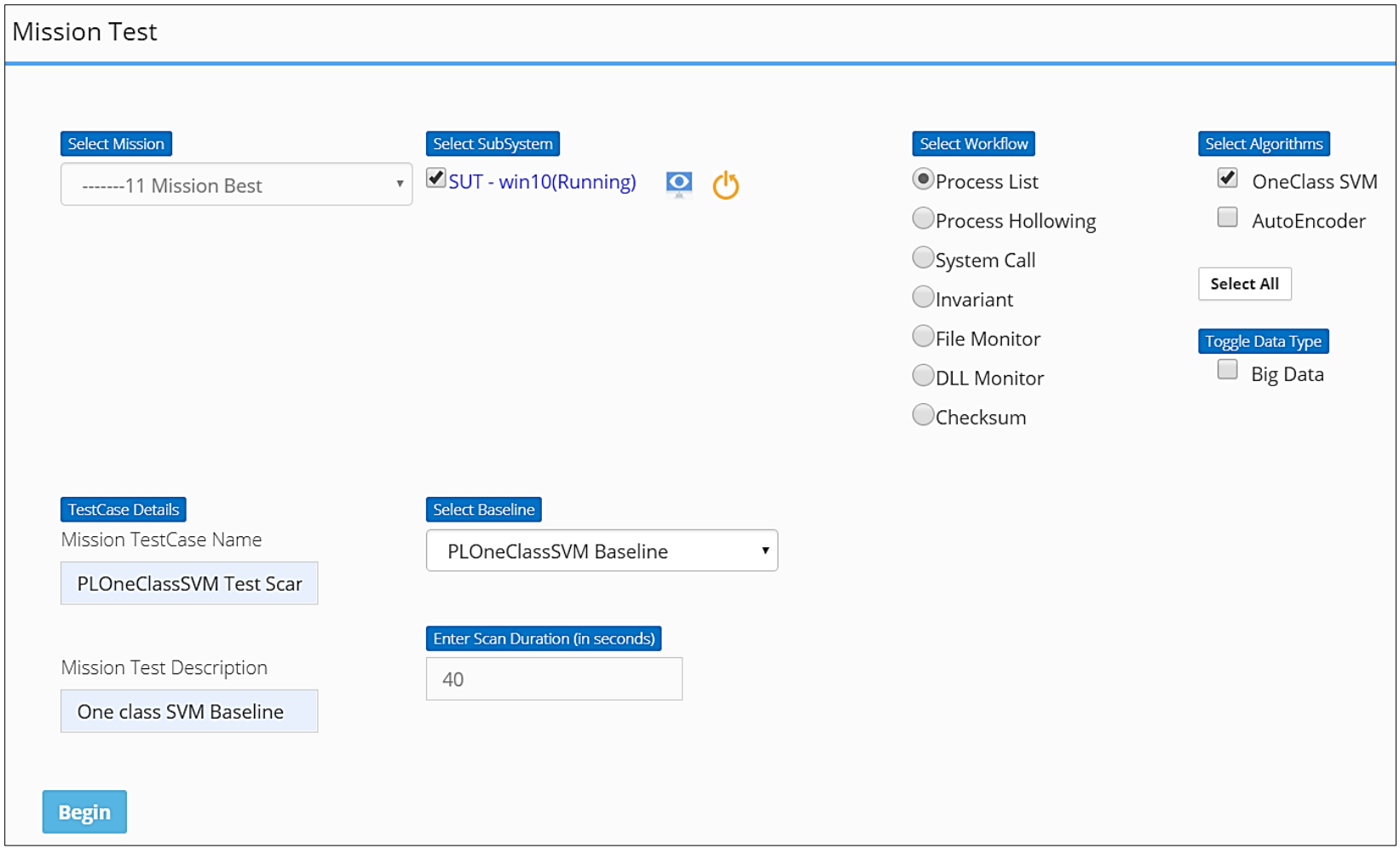
Process List – One Class SVM Mission Baseline Validation

6867	win10	Process List Baseline 08/29/2019	Full Process List Baseline	8/29/2019 4:04:01 PM	View Details

Model ID	Model Workflow	Baseline Scan Results
6867	Process List	

ProcessName	SimilarityScore
GPS.exe	83.9
Weather.exe	97.65
SUT.exe	97.68

Process List – One Class SVM Mission Test



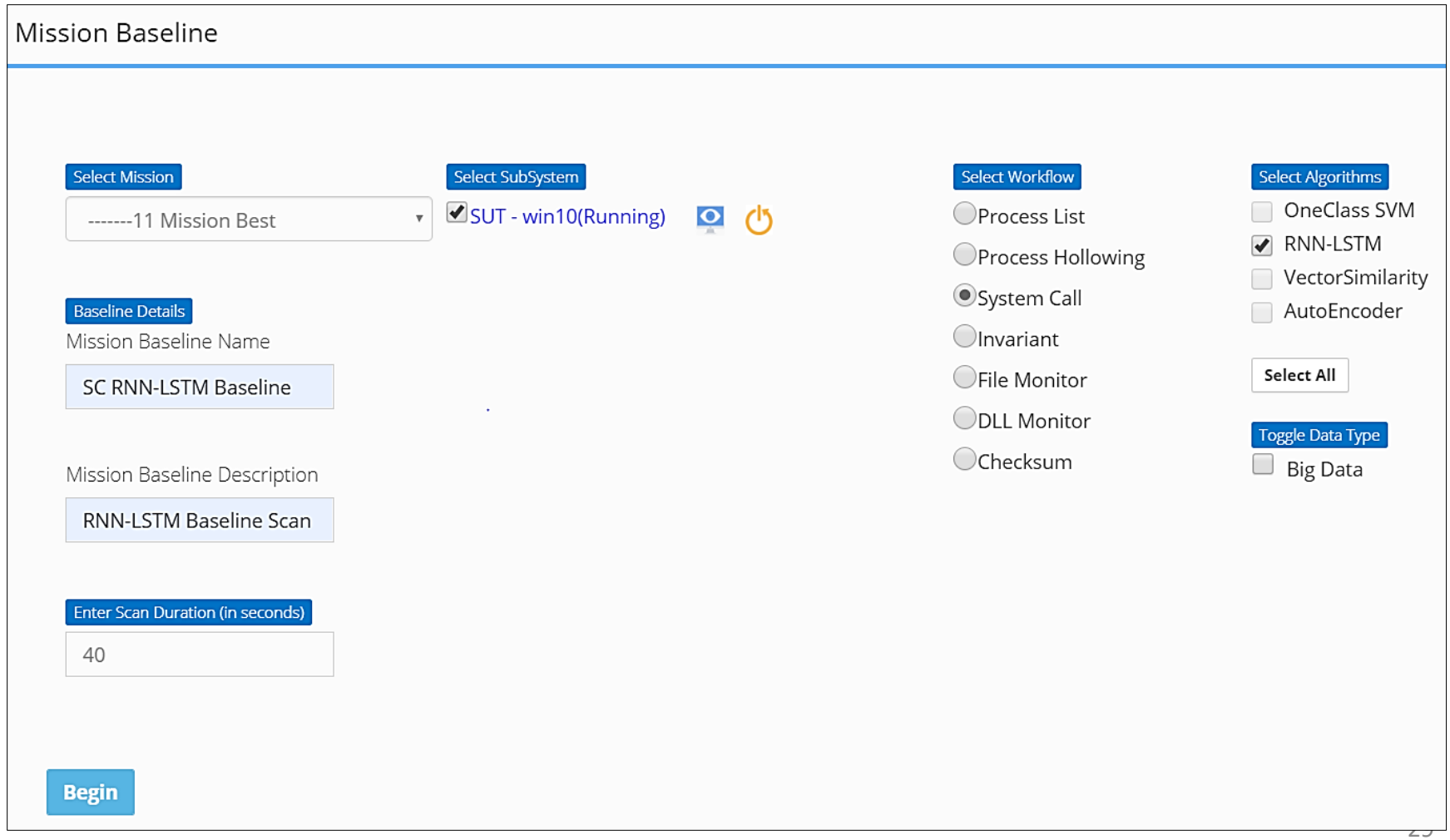
Process List – One Class SVM Mission Test Results

2358	win10	Process List Baseline 08/29/2019	Testing	8/29/2019 5:34:45 PM	View Details

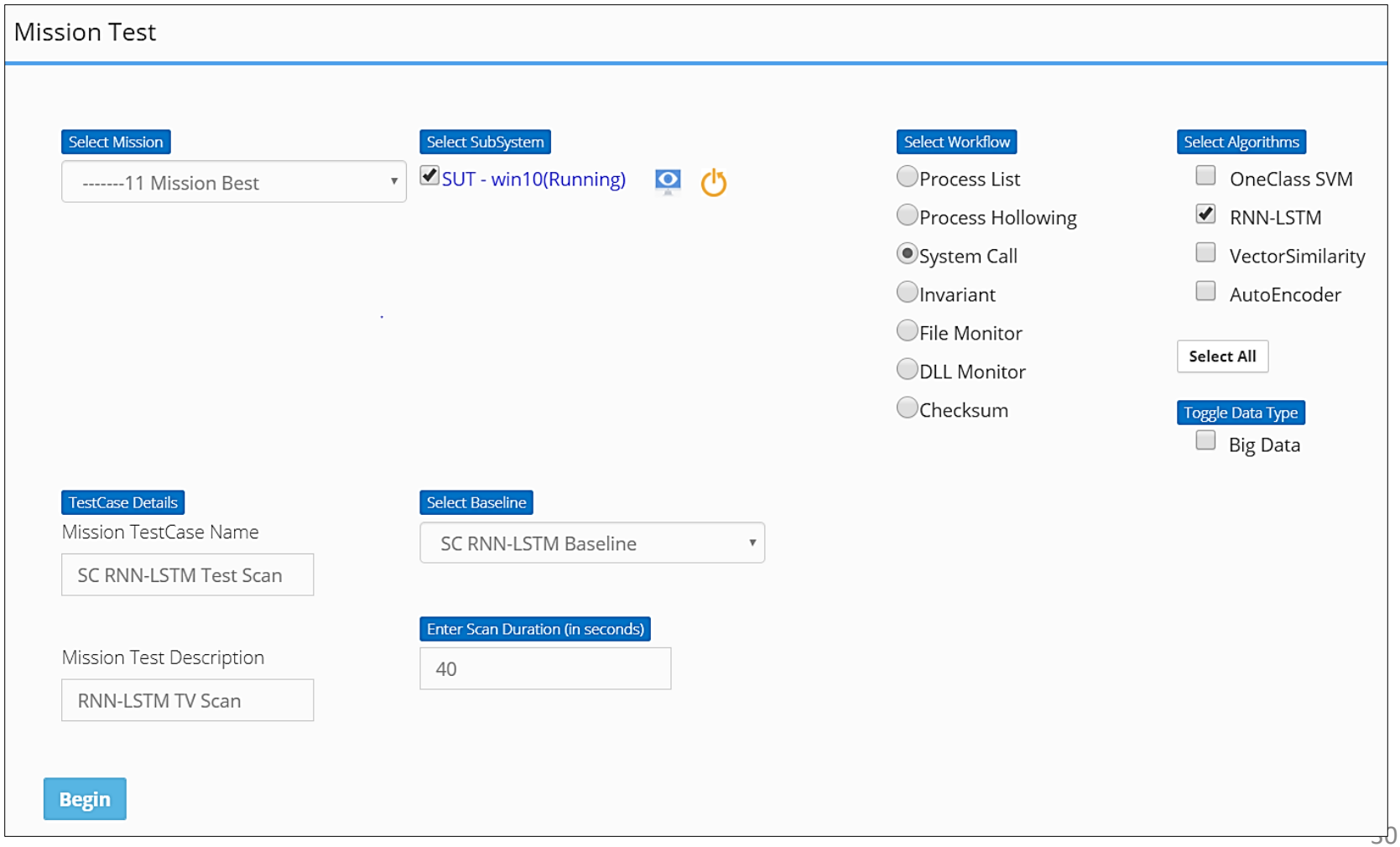
TestCase Result ID	Mission Workflow	Mission Result	
8137	Process List	Normal	
8138	Process List	Compromised	View Details

Baseline Scan Results		Test Vector Scan Results		Status
ProcessName	**SimilarityScore**	**ProcessName**	**SimilarityScore**	
GPS.exe	83.9	SUT.exe	76.44	Compromised
Weather.exe	97.65	Weather.exe	100.0	
SUT.exe	97.68	GPS.exe	94.21	

System Call – RNN – LSTM Mission Baseline



System Call – RNN – LSTM Mission Test



System Call – RNN – LSTM Mission Test Results

TestCaseID	VM Name	Model Name	TestCase Name	Inserted On	Detailed View
2431	win10	SC RNN-LSTM Baseline	SC RNN-LSTM Test Scan	9/12/2019 1:27:32 PM	View Details

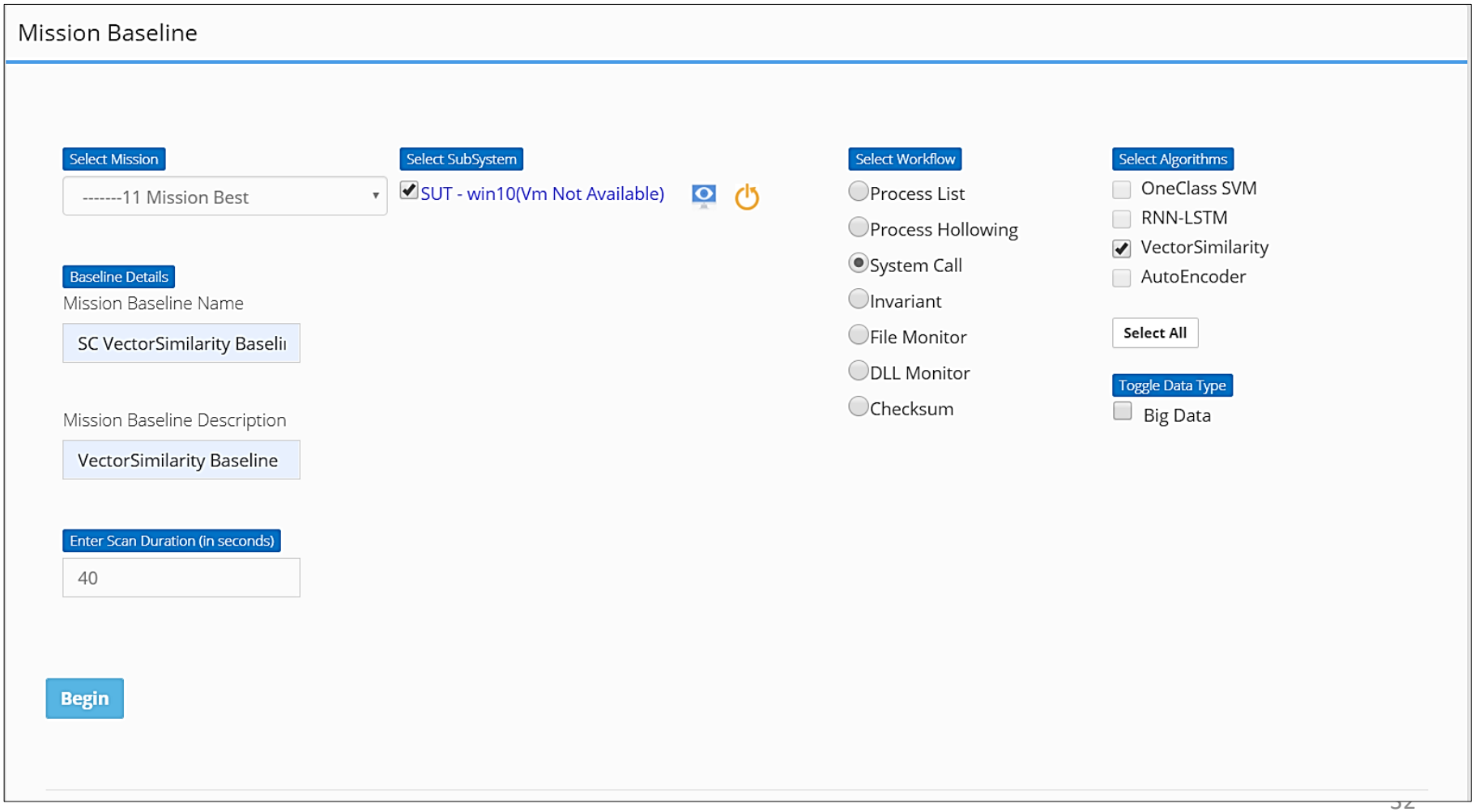
TestCase Result ID	Mission Workflow	Mission Result	
8195	System Call	Compromised	View Details

Result Summary

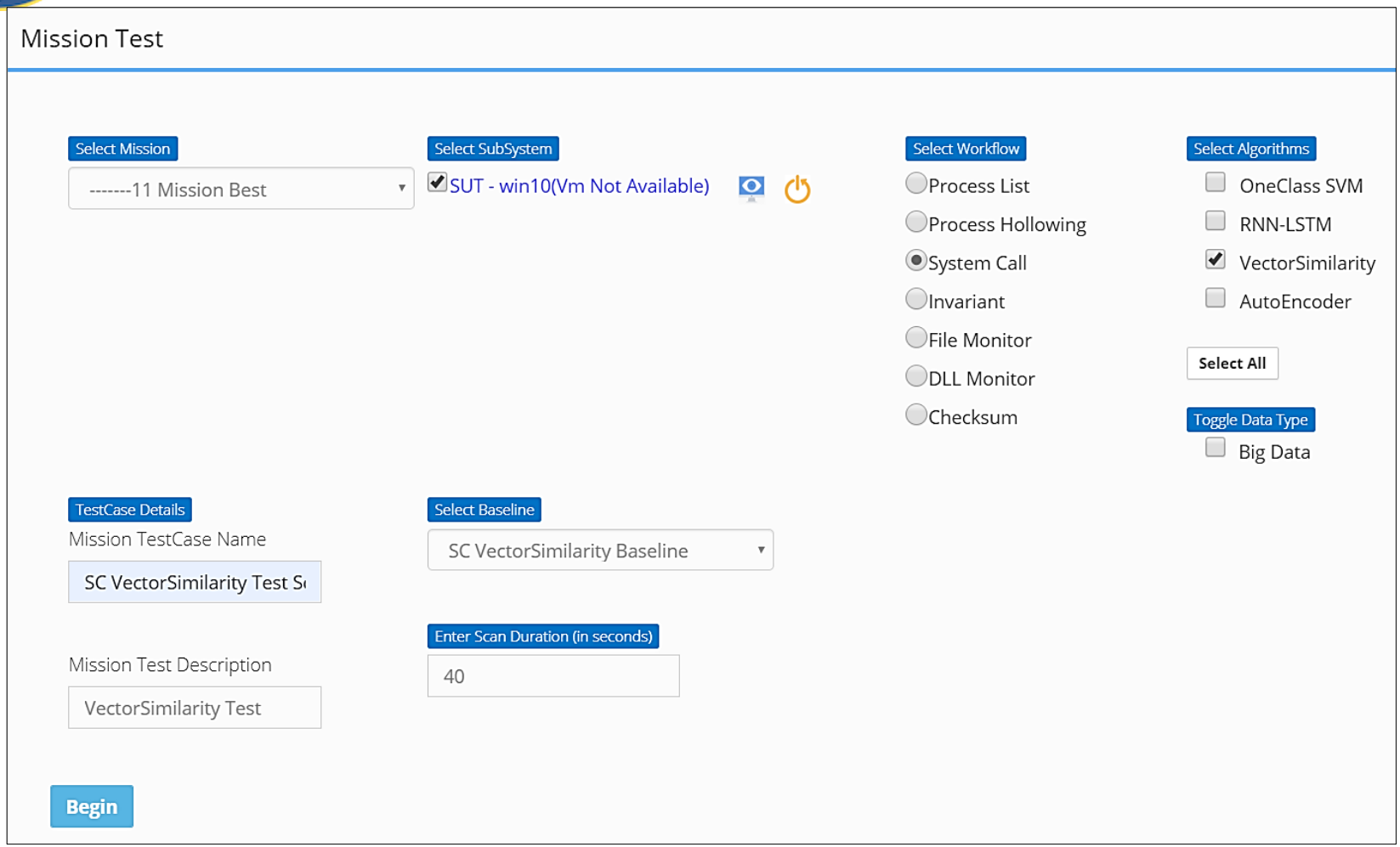
TestCase Details

Baseline Scan Results	Test Vector Scan Results	Status
97.0%	0.002%	Compromised

System Call – Vector Similarity Mission Baseline



System Call – Vector Similarity Mission Test



System Call – Vector Similarity Mission Test Results

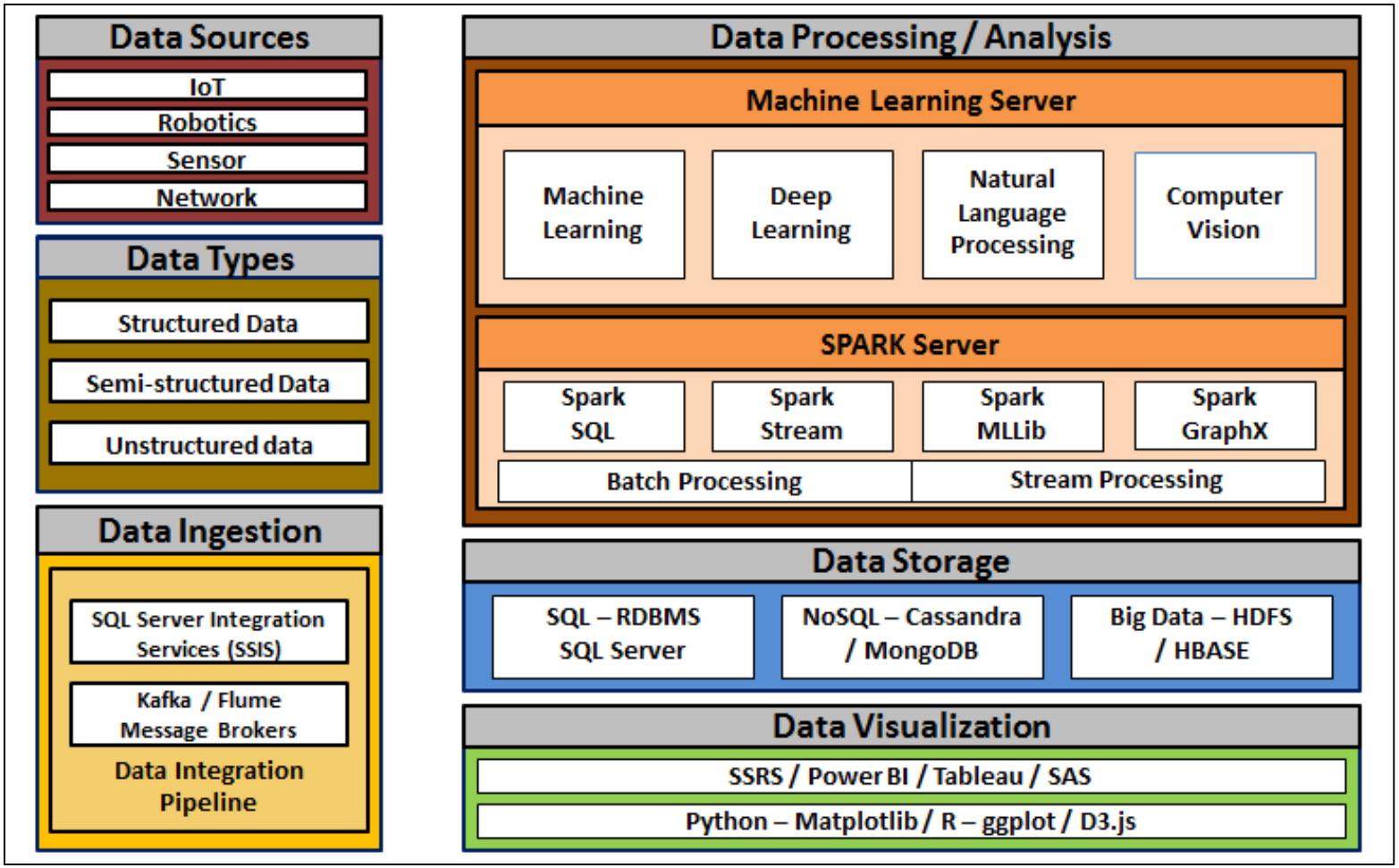
2428	win10	Vector Similarity Baseline	Vector Similarity Test	9/12/2019 12:27:01 PM	View Details

TestCase Result ID	Mission Workflow	Mission Result	
8190	System Call	Compromised	View Details

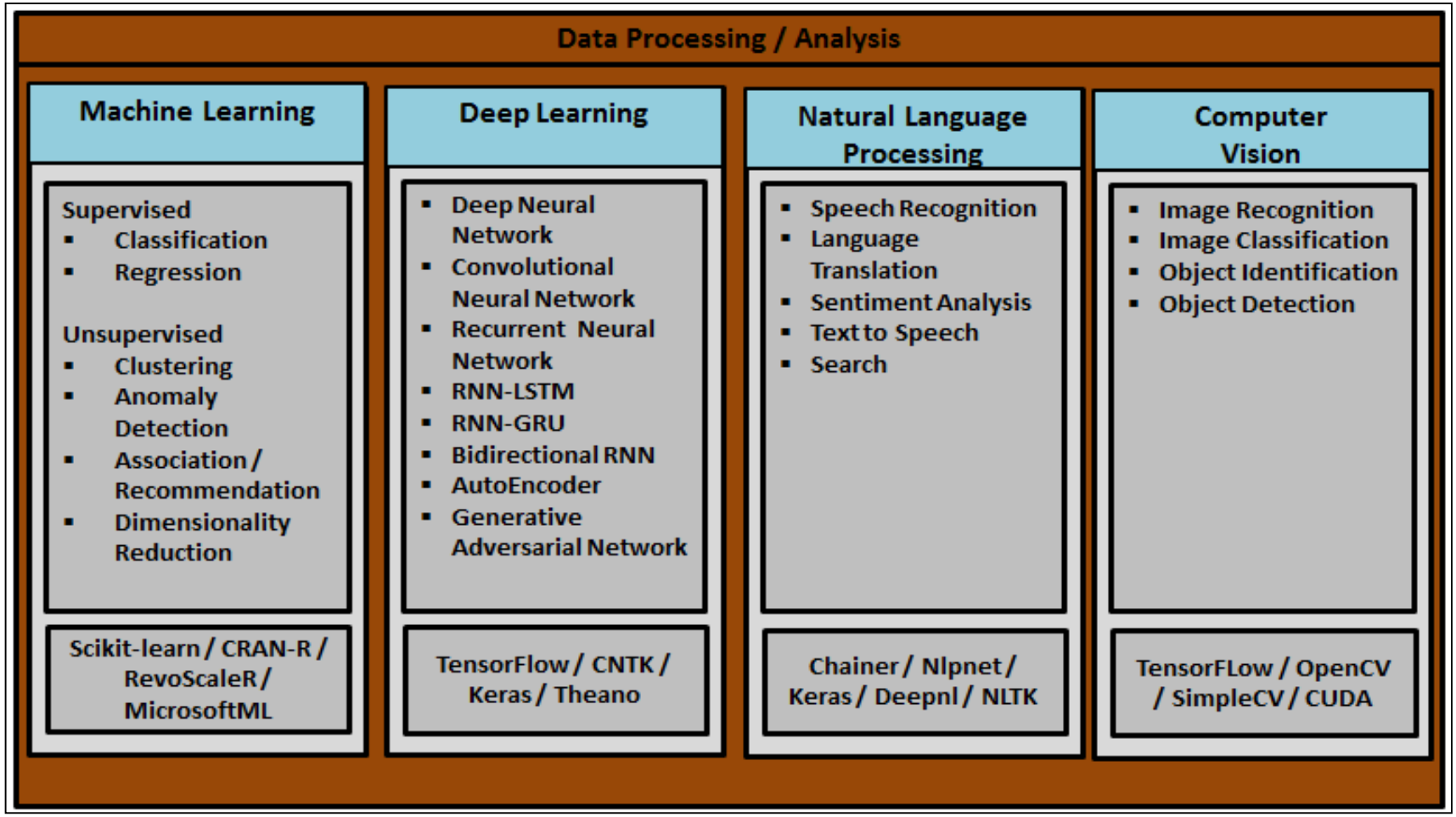
Baseline Scan	Test Scan	Status
Weather.exe, SUT.exe, GPS.exe	SUT.exe	Compromised

Artificial Intelligence & Big Data Hub
On-Premise | Cloud | Hybrid

FIU Applied Research Center



Artificial Intelligence Applications & Algorithms

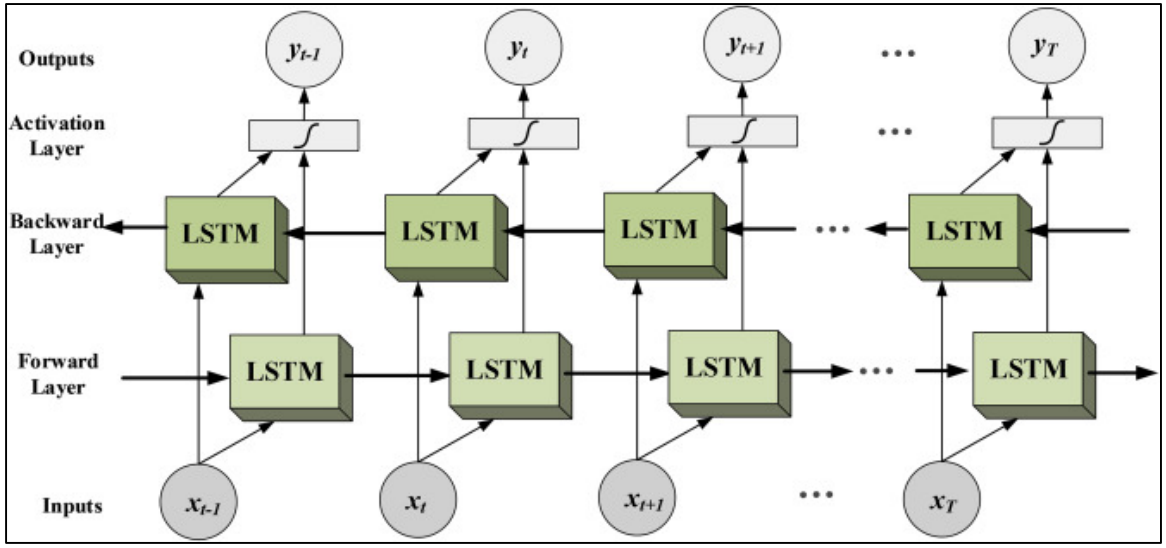


QUESTIONS AND DISCUSSION

Bidirectional RNN-LSTM

Bidirectional RNN-LSTM

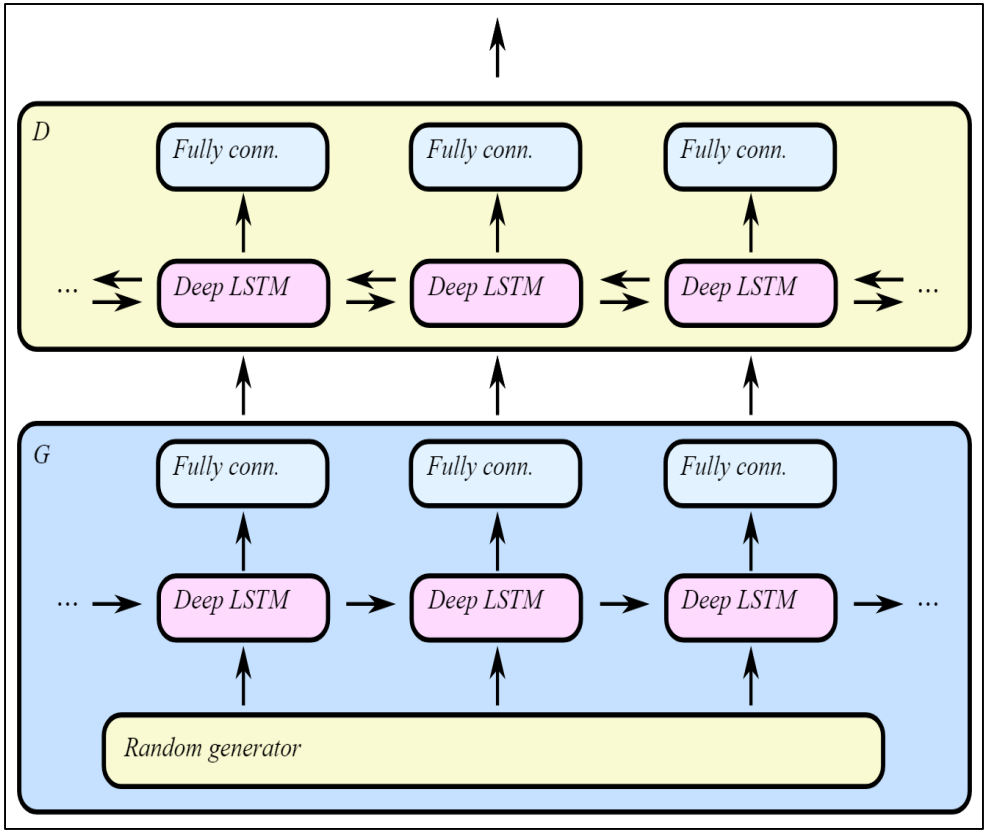
- Utilizes past and future sequence data.
- Gives "context" to our predictions.



https://www.sciencedirect.com/science/article/pii/S0010482518300738

LSTM - Generative Adversarial Networks (GANs)

- Comprised of a “generator” model and a “discriminator” model
- Generator creates benign sequences
- The discriminator determines if a given sequence is valid



https://www.sciencedirect.com/science/article/pii/S0010482518300738