Northrop Grumman Cybersecurity Research Consortium (NGCRC)

Secure Intelligent Autonomous Systems with Cyber Attribution

08 November 2018

Bharat Bhargava
Purdue University

Technical Champions: Paul Conoval, Jason Kobes, Jeffrey Ciocco, Robert Pike, Will Chambers, Miguel Ochoa, Steve Seaberg, Peter Meloy, Jessica Trombley-Owens, Brock Bose, Sam Shekar, Roderick Son

Motivation from NGC: Securing Intelligent Autonomous Systems

NORTHROP GRUMMAN

- This project achieves NGC's goal on secure mission-deployable autonomous systems: develop and continuously monitor mission plans, react based on mission profile, and achieve mission objectives even under attacks[1].

- Autonomous operations and the rapid timing dynamics required of system responses present new challenges in security and cyber attribution.

- The project operationalizes AI into practical deployable systems through an end-to-end systems-level design approach for security.

- We implement multi-faceted dimensions of AI/ML, and scaling the appropriate level of AI/ML for effectively and efficiently meeting mission objectives under attacks.

[1] **Paul Conoval,** "*Integrating Artificial Intelligence Capabilities into Deployable Systems*", Keynote given in IEEE AIKE 2018, Laguna Hills, California, USA

Collaboration with NGC, MIT, and Purdue Faculty*

NORTHROP GRUMMAN

- This project's goals are developed based on the discussions with Paul Conoval, Jason Kobes, and Bob Pike. We will continue update them about the projects.
- We will collaborate with Dr. Lalana Kegal from MIT on developing provenance ontology to enhance security and trustworthiness of IAS.
- We will work with Prof. Christopher Clifton* on poisoning attacks on machine learning model—offensive and defensive strategies.
- We will work with Prof. Christopher Yeomans* from Philosophy Department in cognitive autonomy.

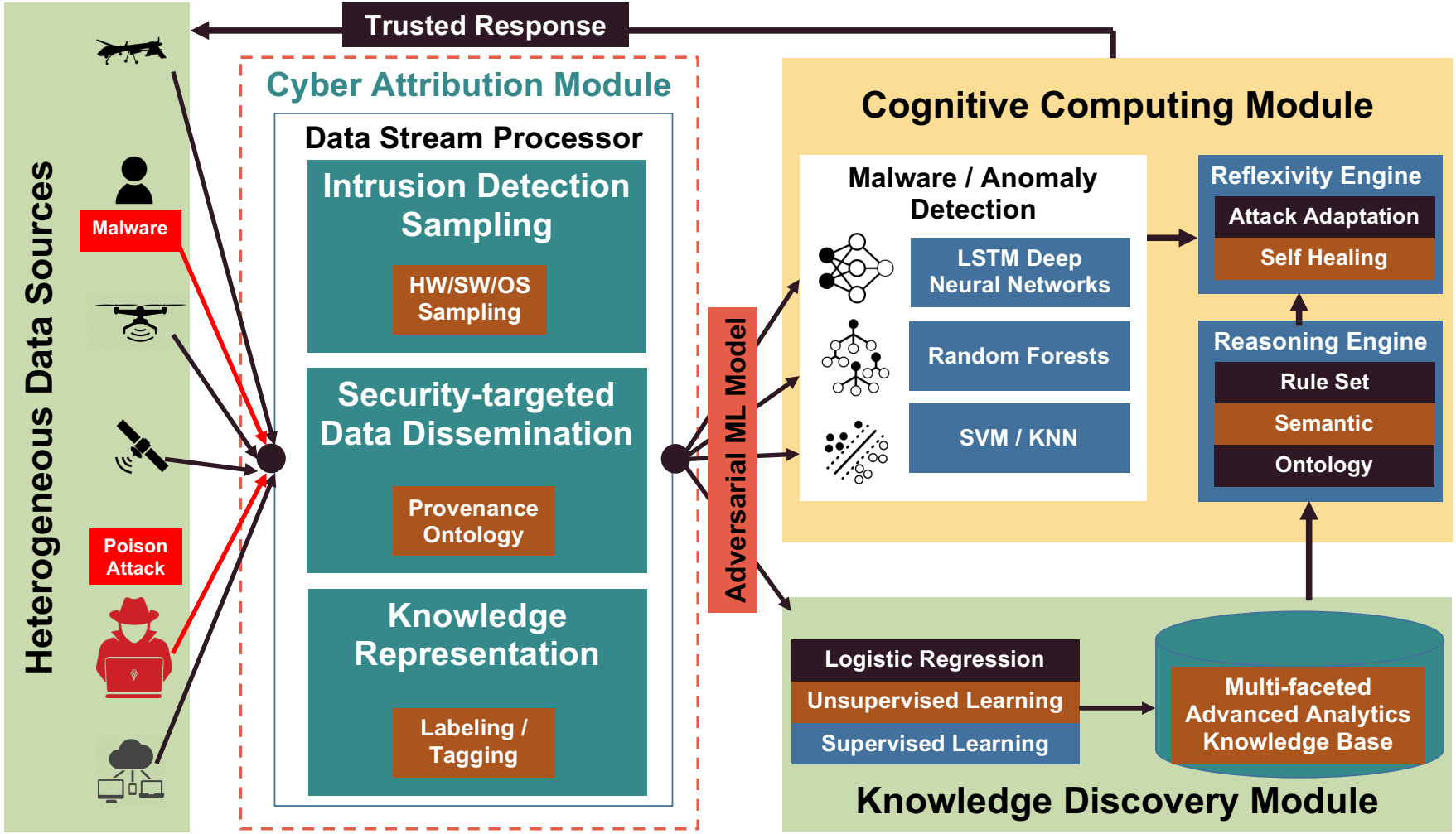
Secure Intelligent Autonomous Systems

NORTHROP GRUMMAN

- Autonomous Systems should be
  - Able to continually perform complex tasks under attacks without or with limited ongoing connection to humans.
  - Cognitive enough to act without a human's judgment lapses or execution inadequacies.
- Intelligent Autonomous Systems (IAS) are highly **Cognitive** to predict attackers next steps, effective in **Knowledge Discovery** for attack patterns, **Reflexive** to adapt and self-heal, and **Trusted**.
- We'll leverage convergence of advanced analytics, cyber security, and AI, presenting new opportunities, issues, and threats, impacting mission deployable systems.

Comprehensive Secure IAS with Cyber Attribution Architecture

NORTHROP GRUMMAN



Secure IAS: Overview

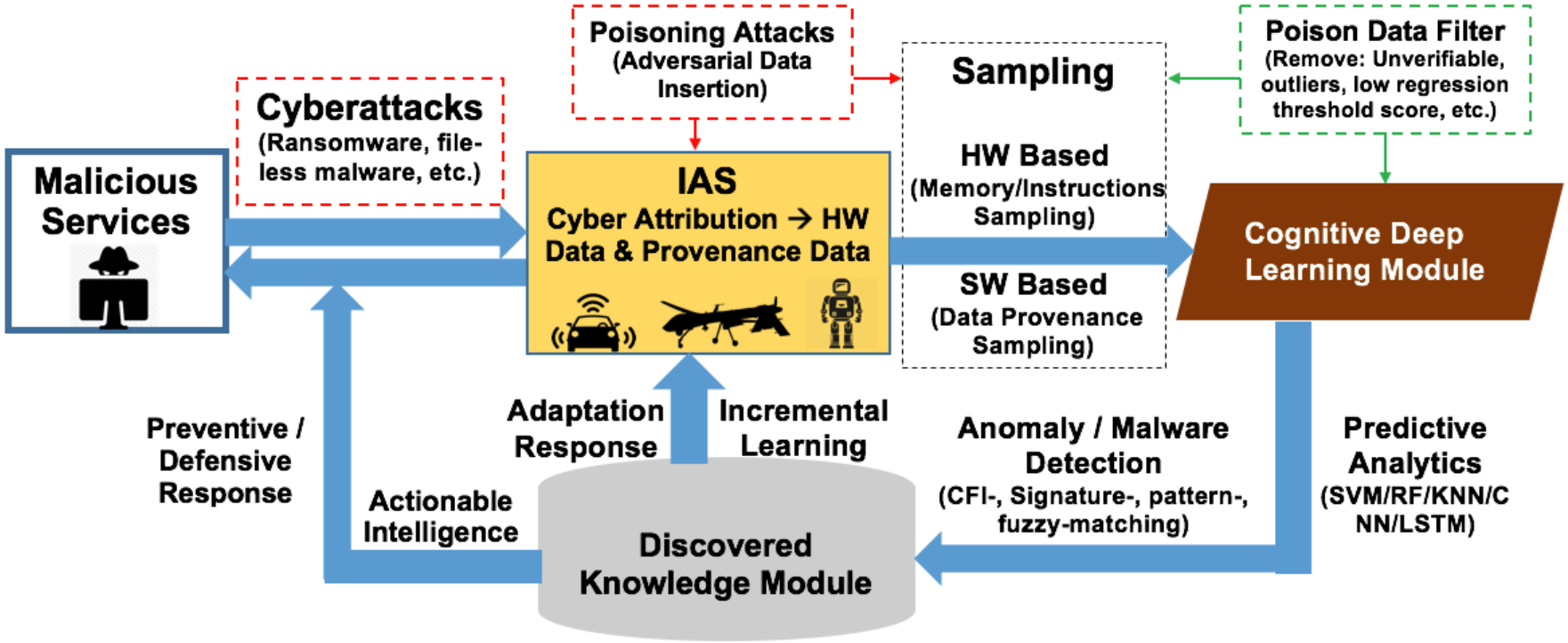
NORTHROP GRUMMAN

- This project will advance science of security in IAS through multifaceted advanced analytics, cognitive computing, adversarial machine learning, and cyber attribution.

- We will develop
  - A cyber attribution module that monitors threats and performs intrusion detection sampling on system data.
  - A cognitive computing module with reasoning engine & anomaly / malware detection through light-weight ML algorithms and LSTM deep neural networks.
  - A secure information dissemination model with provenance ontology.
  - Reflexive module to adapt and self-heal during attacks.

Secure IAS: Overview

NORTHROP GRUMMAN



Secure IAS: Overview

NORTHROP GRUMMAN

- Behavior-based analytics is used to categorize adaptive cyber-attacks and poison attacks on ML. We will develop methodologies that use contextual information about the origin and transformation of data points in the training set to identify poisonous data.

- The research in trace back, source routing, ideas on onion router (TOR), and pre-positioned instrumentation for cyber attribution model to identify source as well as intermediary of attack are the ideas for cyber attribution.

- To enable cognitive autonomy for security, large data will be generated to discover patterns and identify signatures for malicious and benign processes.

- The classification algorithms (RF/SVM/KNN) will be applied to the sampled data and discover patterns to detect likely anomalies and malwares.

NORTHROP GRUMMAN

Cyber Attribution & Monitoring

Solutions based on PROV-O and Graph model of Cyber Attribution

Cyber Attribution

NORTHROP GRUMMAN

- Attribution for information assurance is nothing but tracing back attackers step and finding the attacker's origin as well as the attack's origin.

- Cyber attribution techniques include periodic trace back querying, maintaining logs, debugging, host monitoring, transmitted message observation, forcing attacker to self-identify through validation and access requirements, intrusion detection by periodic sampling, and combination of these techniques.

- Our cyber attribution model has two major components: (1) Provenance ontology structure (PROV-O) and (2) Sampling for intrusion detection.

NORTHROP GRUMMAN

Cyber Attribution & Monitoring

Solutions based on **PROV-O** and Graph model of Cyber Attribution

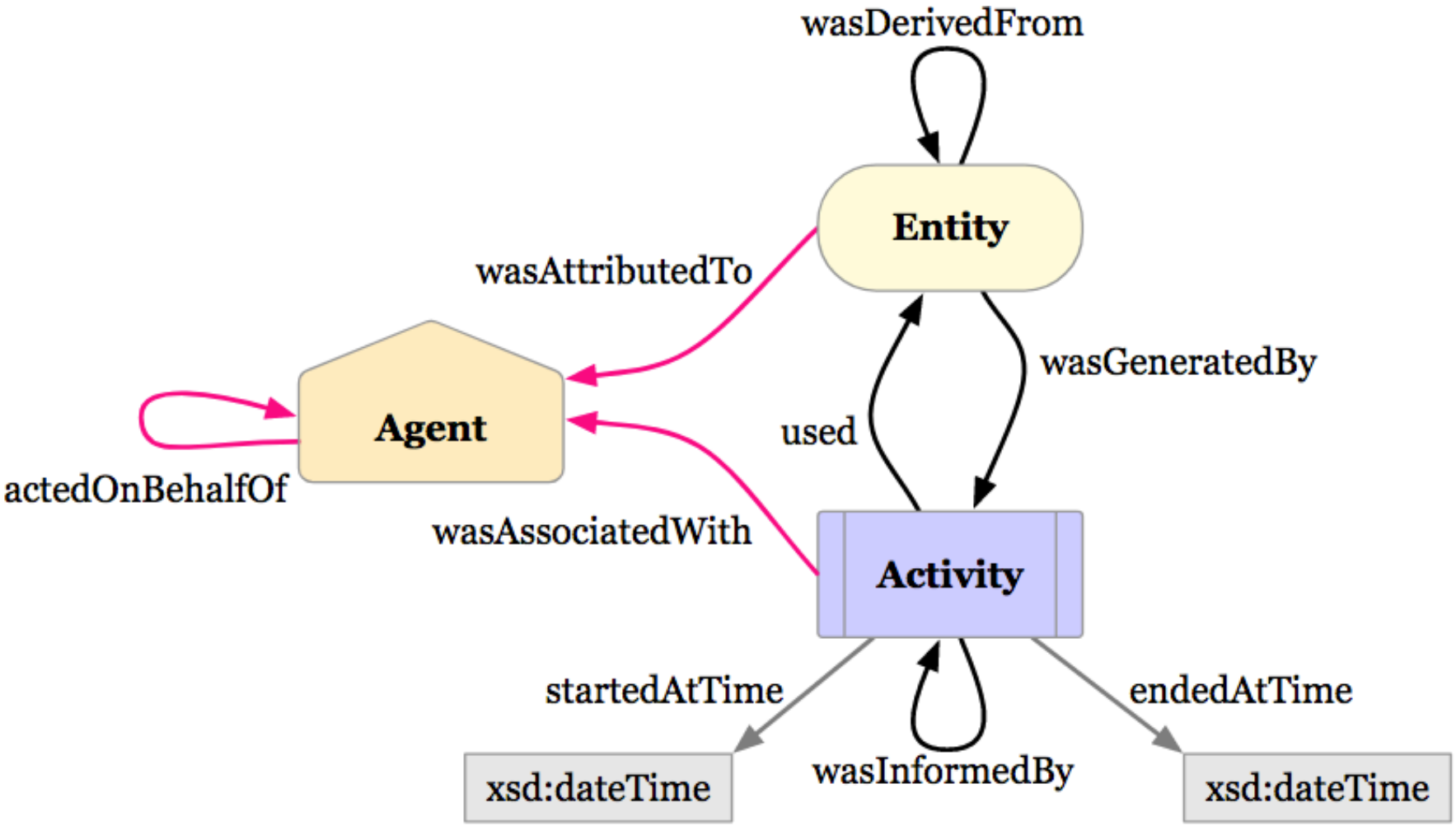
Knowledge Representation by PROV-O (Collaboration with Dr. Lalana Kagal, MIT)

NORTHROP GRUMMAN

- Provenance Ontology (PROV-O) provides set of classes, properties, restrictions, and reasoning rules to represent and interchange provenance information gathered in different systems.
- This ontology specification provides the foundation to implement provenance applications in different domains.
- PROV-O can represent, exchange, and integrate provenance information generated in different systems and under different contexts.
- PROV-O is a lightweight ontology that can be adopted in a wide range of applications.
- A new component of an IAS is a reasoned module in addition to machine learning and model training. Our collaborator at MIT, Dr. Lanana Kagal has developed explanation mechanisms for reasoners and we will work together to advance it for machine learning.

PROV-O Structure

NORTHROP GRUMMAN



Source: https://www.w3.org/TR/prov-o/

PROV-O Structure

NORTHROP GRUMMAN

- An **prov:Entity** is a physical, digital, conceptual, or other kind of thing with some fixed aspects; entities may be real or imaginary.
- An **prov:Activity** occurs over a period of time and acts upon or with entities—consuming, processing, transforming, modifying, relocating, using, or entities.
- An **prov:Agent** is something that bears some form of responsibility for an activity taking place, for the existence of an entity, or for another agent's activity.
- Properties such as prov:used, prov:wasDerivedFrom, prov:startedAtTime etc. are used to complete the structure.

Experiment: PROV-O Structure Integration with NGC-WaxedPrune

NORTHROP GRUMMAN

- **Objective:** Integrate PROV-O structure with Active Bundle (AB) implementation of NGC-WaxedPrune.

- **Input:**
  - Role-based access control policies
  - Cryptographic capabilities of client browsers
  - Authentication techniques

- **Output:** PROV-O integrated AB with representation of different roles, client browsers, and authentication techniques.

- **Experimental Setup:** Active Bundle is already exists with NGC-WaxedPrune set up and experiments will be conducted on that.

Cyber Attribution & Monitoring

Solutions based on PROV-O and **Graph model** of Cyber Attribution

Secure Monitoring: Cyber Attribution Graphs (CAG)

NORTHROP GRUMMAN

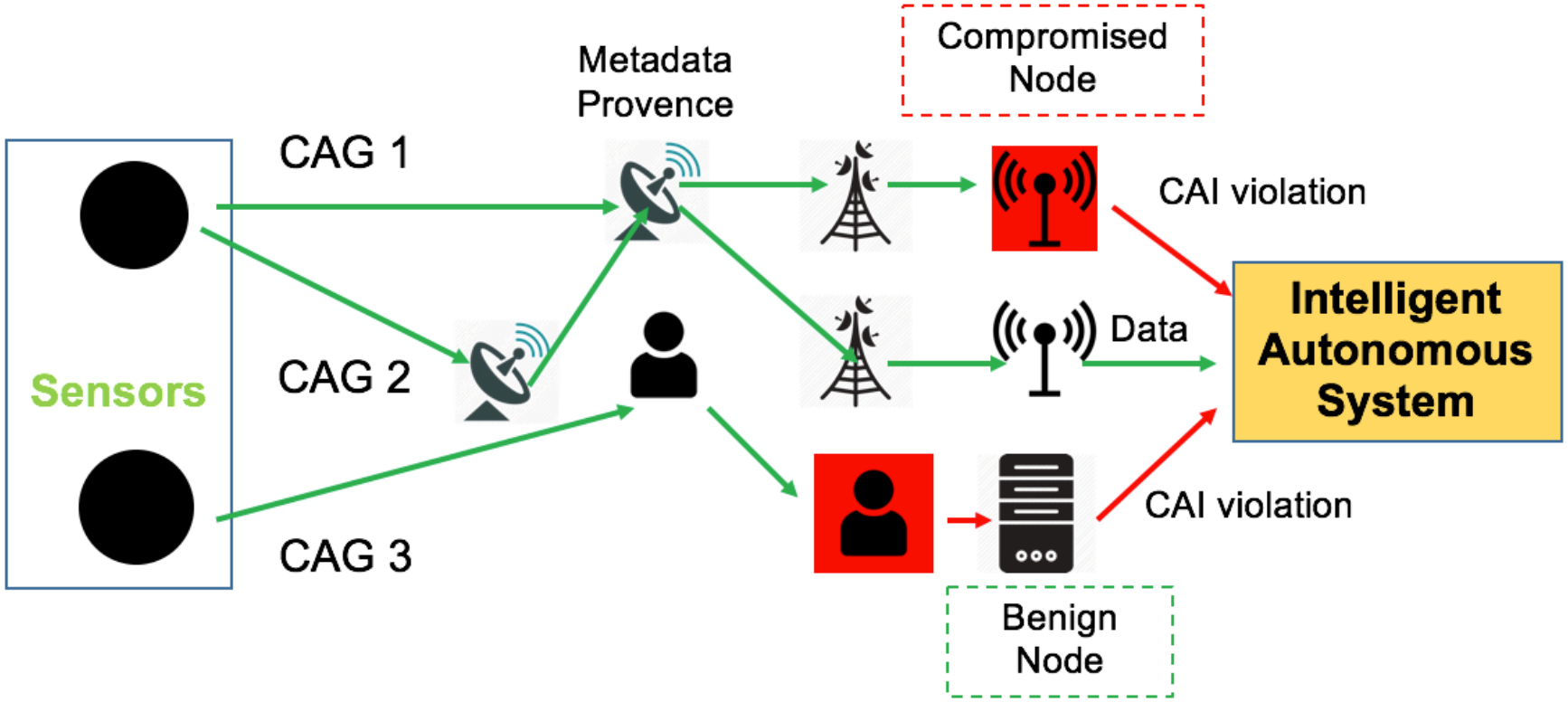
- We will model Cyber attribution as a graph where each node (sensors, systems, etc.) stores metadata of provenance—data source id, user access-level, user id, etc.
- These Cyber Attribution Graphs (CAG) are a representation of all of the paths that might traverse through the IAS communication network.
- The graphs will be predefined for the training model and updated each time when there is an entity (node) sending new data.

Secure Monitoring: Cyber Attribution Graphs (CAG)

NORTHROP GRUMMAN

- We will model Cyber attribution as a graph where each node (sensors, systems, etc.) stores metadata of provenance—data source id, user access-level, user id, etc.
- These Cyber Attribution Graphs (CAG) are a representation of all of the paths that might traverse through the IAS communication network.
- The graphs will be predefined for the training model and updated each time when there is an entity (node) sending new data.

Cyber Attribution in IAS with CAG



Malicious Activity Detection with CAGs

NORTHROP GRUMMAN

- The new data traversal graph will be compared to the known training graphs to find out any malicious activity.
- We will create a new quantitative measure—Cyber Attribution Integrity (CAI)—that predicts the likelihood of the given data path of a graph being benign or malicious.
- CAI will consist of the predictive probabilities of each edge in CAG from one node to another.
- Fuzzy matching between the generated probabilities and predefined training graphs used to determine if there are any poisoning attacks or tampering in the data.

Experiment: Malicious Activity Detection with CAGs

NORTHROP GRUMMAN

- **Objective:** Using supervised learning, monitor and generate cyber attribution graphs with different entities, simulate known malware, and detect malicious behavior.
- **Input:** Monitored and recorded graph data where each node (sensors, systems, etc.) stores metadata of provenance—source id, user access-level, user id, etc.
- **Output:** A binary classification-–whether the given graph is malicious or benign.
- **Experimental Setup:** Fuzzy matching between the generated predictive probabilities and training graphs used to determine if there are any poisoning attacks or tampering in the data.

Adversarial Machine Learning

Poisoning Attacks: Offensive & Defensive Strategies

Adversarial Machine Learning: Challenge

NORTHROP GRUMMAN

- Attackers poison training data to Misclassify inputs: the attackers tries to shift the decision of the model so that a specific class is misclassified.
- Another insidious attack is the *backdoor or Trojan attack*: the attackers make sure that the training data and validation performs normally but the model misbehaves only when backdoor key is present.
  - E.g. a backdoor causes the model to misclassify a road sign as a different sign whenever there is a post-it note on top it. This type of backdoor can result in disastrous outcome for autonomous vehicles.
- Poisoning data also reduces model performance: attacker tries to minimize the accuracy of the model, rendering it unusable in real-time scenarios.

Adversarial Machine Learning: Solution Overview

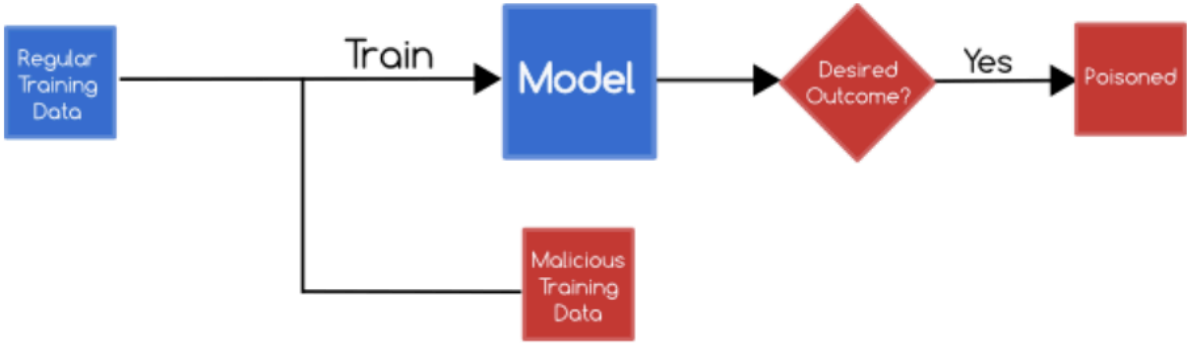
NORTHROP GRUMMAN

- We will generate offensive models to infiltrate training set and observe the progression of the attacks in various states of learning.
- We will design counter measures to isolate the poisonous data by using outlier detection algorithms and clustering.
- We will implement a fuzzy matching technique with knowledge base of training set manipulations to identify potential intrusions in training sets.
- We will define metrics to quantify trustworthiness of the data originating from trusted or untrusted system, verified or unverified data format, etc.

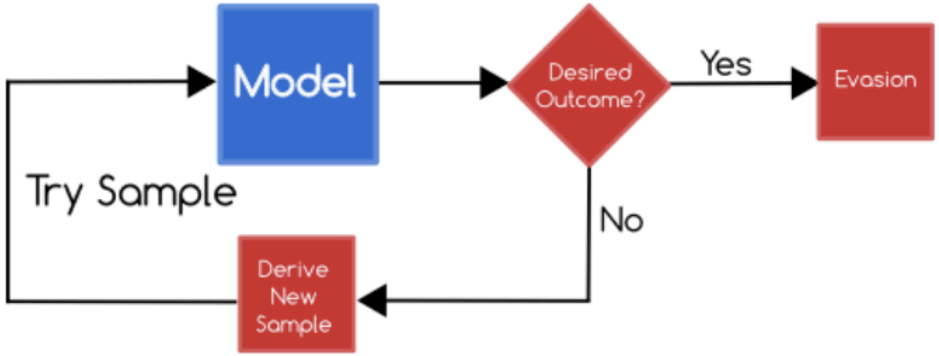
Poisoning & Evasion Attacks (Collaboration with Prof. Clifton)

NORTHROP GRUMMAN

- In poisoning attacks, attackers try to influence, learn, and corrupt the machine learning model through **training** data.



- In evasion attacks, the attacker does not tamper with the machine learning model but they produce adversary selected inputs for **testing** phase.



Poisoning & Evasion Attacks: Offensive Strategy

NORTHROP GRUMMAN

- In order to facilitate an effective defense against poisoning attacks, we will first implement an attack: optimize the data perturbation to the minimum to corrupt the machine learning model.

- Optimization-based poisoning attacks: We will characterize the attack strategy with ***bilevel optimization*** problem and we will solve it with iteratively optimizing one poisoning sample at a time through gradient ascent.

- With this, we will also implement Baseline Gradient Decent (BGD) attack where poisoning attack will take place in regression setting.

Poisoning & Evasion Attacks: Defensive Strategy

NORTHROP GRUMMAN

- We will employ k-Nearest Neighbors (kNN) algorithm to identify outliers in testing and training data.
- Cosine similarities from the testing/training samples to every single new training/testing samples are calculated.
- The k-nearest neighbor of the testing sample is selected, where k is an integer that can be determined through elbow method.
- The most frequent classes of these K neighbors is assigned to the test sample i.e., the testing sample is assigned to the class D if it is most frequently occurring label in k nearest training samples.

Poisoning & Evasion Attacks: Defensive Strategy for Partially Trusted Data

NORTHROP GRUMMAN

- When training data is obtained from untrusted sources (customer behavior profile and crowdsourced data), it may be prone poisoning attacks.
- It is vital to detect when models have been poisoned or tampered with when they are trained by untrusted sources.
- Overview of the poisoning attacks detection

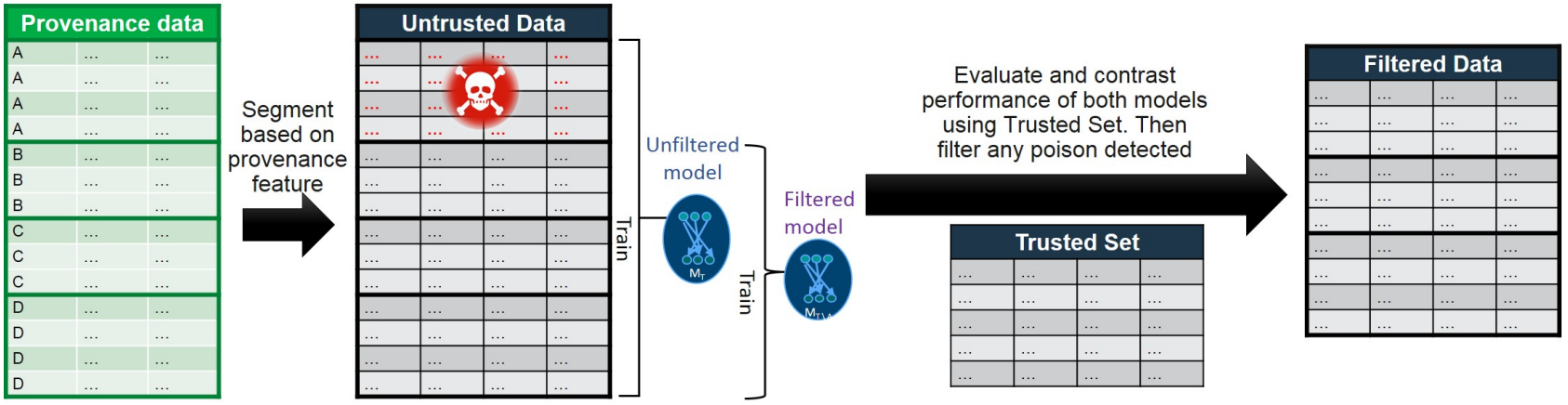


 Image source: IBM Adversarial Machine Learning

Poisoning & Evasion Attacks: Defensive Strategy for Partially Trusted Data

NORTHROP GRUMMAN

- Segment the training data into several groups based on provenance data. The probability of poisoning is highly correlated in across samples in each group.
- Data points in each segment are evaluated by comparing the performance of the classifier trained with and without that group.
- Using Reject on Negative Impact (RONI), we evaluate the effect of individual data points on the performance of the final classifier.
- By using provenance data, our method can properly group data points together and compute their cumulative effect on the classifier. It increases detection rates and reduces computational costs.

Experiment: Poisoning & Evasion Attacks

NORTHROP GRUMMAN

- **Objective:** Create an offensive mechanism to efficiently poison training data and design a defensive mechanism for detecting poisoning and evasion attacks.
- **Input:** Featured datasets (datasets that are already categorized with features) with training and testing samples.
- **Output:** Percentage of poisonous samples required to poison training data, accuracy of techniques in detecting poisonous and evasion attacks.
- **Experimental Setup:** Data collection kernel drivers for windows and LiME for Linux to collect benign and malicious samples.

Cognitive Computing

Solutions based deep learning & light-weight Machine Learning models

Cognitive Computing

Solutions based **deep learning** & light-weight Machine Learning models

Problem Statement

NORTHROP GRUMMAN

- Order-aware Recognition Problem: Refers to incapacity of distinguishing sequences after certain length.
- Different ordered sequences *abcba*, *cbabc* and *bcbab produce the same* set of 2-tuple adjacent events.
- Clearly, methods able to analyze sequences of length 2 or less can not discern among these three different ordered sequences. Obvious if we consider the set of 3-tuple adjacent events.

Sequence	2-tuple	3-tuple
abcba	{ab, bc, cb, ba}	{abc, bcb, cba}
cbabc	{cb, ba, ab, bc}	{cba, bab, abc}
bcbab	{bc, cb, ba, ab}	{bcb, cba, bab}

Sequences of length 2 might be considered non-anomalous, while sequences of length 3 are anomalous

- N-gram and N-order HMM based approaches can distinguish among sequences of length less than or equal N. Due to limitations on their algorithms N is usually small.

Types of attacks

NORTHROP GRUMMAN

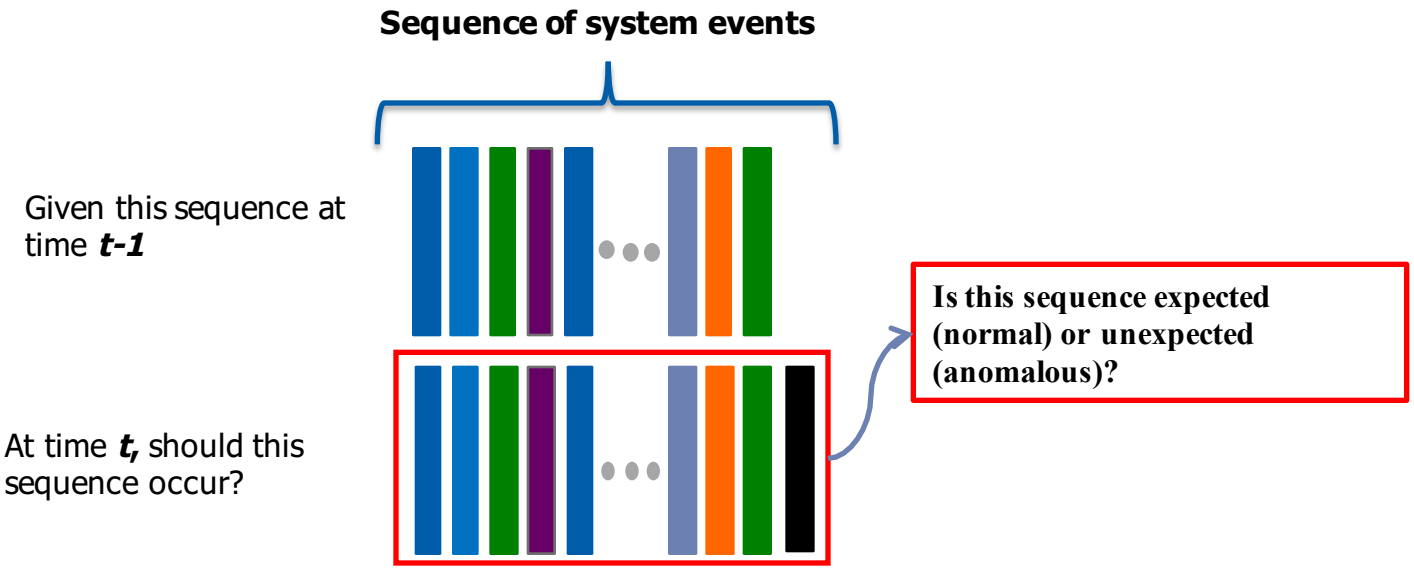
- **Mimicry attacks:**
  - Attacker has knowledge of the detection algorithm and attempts to avoid detection while accomplish their attack goals.
  - Event sequences are crafted to conduct attacks. For example, the shortest six-event (system calls) malicious sequence*: namely*, *chroo*t, *chdir*, *chroot*, *open*, *write*, *close*[1]*.*
  - Adversaries can insert No Operation (NOP) events to make sequences of variable length and longer than expected.
- **Advanced Persistent Threats:**
  - Requires a high degree of covertness over a long period of time.
  - Involves an external command and control system that continuously monitors and extract data from the target.
  - The events of the attack form variable length sequences that span for long periods of time.

[1]D. Yao. "Anomaly Detection As a Service: Challenges, Advances, and Opportunities." Morgan and Claypool Publishers. 2018.

Approach Overview

NORTHROP GRUMMAN

- Deep Learning based approach for the detection of unexpected sequences of events in the system.
- Answer **the anomaly detection problem** of given a sequence of system events $e_1e_2e_3\ldots e_{k-1}$ **whether or not the sequence** $e_1e_2e_3\ldots e_k$ **should occur**?



Approach Overview

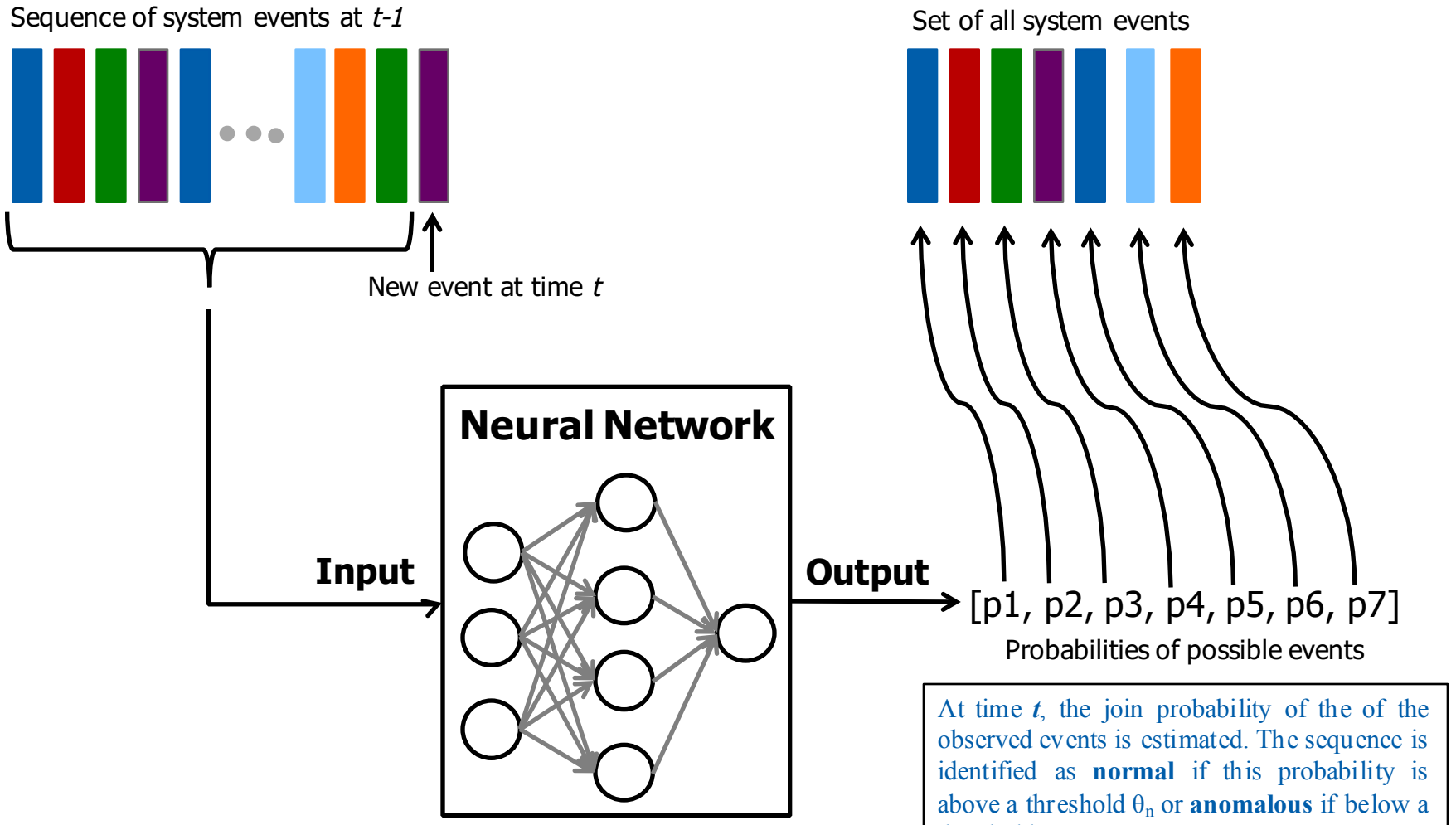
NORTHROP GRUMMAN

- An event in the sequence can either be a network message, a system call or even a HTTP GET request.
- The actual definition of **“a system event”** is given by our subject of study.
- **Defining the vocabulary of system events is one of our main research tasks.**

What is an event?

Set of all possible system events?

Approach Details



Approach Details

NORTHROP GRUMMAN

Input: Sequence of events in the system
Output: normal or anomalous

- **Step 1:** Define a finite set E of events $e_1, e_2, ..., e_N$ in the system. Events occur in a time-series fashion.
- **Step 2:** At time $t-1$, given an observed series of events $\{e_i{}^1, e_i{}^2, ..., e_i{}^{t-1}\}$ (with $i = 1, 2, ...orN$) find the set K of the top k events to occur in time t.
- **Step 3:** At time t, the sequence $\{e_i{}^1, e_i{}^2, ..., e_i{}^{t-1}, e_i{}^t\}$ is non-anomalous if $e_i{}^t \in K$, otherwise anomalous.

Algorithm 1: Anomaly detection algorithm

Research Tasks

NORTHROP GRUMMAN

- **Selection of data**: The data to be used will either have CVE (Common Vulnerabilities and Exposures) vulnerabilities or allow the insertion of these vulnerabilities. This is a requirement to obtain quantitative results and being able to compare different models.
- **Preprocessing of dataset**: Data will be represented as time-series. The required pre-processing will be conducted to obtain sequences of events organized by time. The definition a system event is crucial in this task.
- **Implementation of the model**: LSTM models and its variant Gated Recurrent Unit without and with an attention layer will be implemented for comparison purposes.
- **Optimization of the models with the state-of-the-art regularization techniques for RNN-based models**:
  - Implementation of: dropouts in the hidden-to-hidden recurrent weights, embedding dropout, randomized-length backpropagation through time (BPTT), activation regularization (AR) and temporal activation regularization (TAR)[1].
  - Implementation of Cyclical learning rates as suggested[2].

[1]S. Merity, N. Shirish, R. Soccer. "Regularizing and Optimizing LSTM Language Models." CoRR. 2017.
[2]L. Smith. "Cyclical Learning Rates for Training Neural Networks." In IEEE Winter Conference on Applications of Computer Vision. 2017.

Research Tasks

NORTHROP GRUMMAN

- **Evaluation of the models and answer the following model-related questions**:
  - What is the minimum number of events previously observed required to have an acceptable rate of detection?
  - What impact does the length of the previous sequences have over the detection during the evaluation phase? Is the detection of anomalies given the observation of a short event sequence as good as the detection made given an observation of a long event sequence?
  - How to effectively discriminate between events of low and high probabilities from the output array of the models without manually fixing a threshold?
  - What is the True Positive Rate (TPR) and the False Positive Rate (FPR) of the RNN-based (LSTM and GRU with and without attention) anomaly detection solution.
  - How much accurate are RNN-based models with respect to traditional solutions such n-gram and Hidden Markov Model?

Model Implementation

NORTHROP GRUMMAN

MODEL

```
# -*- coding: utf-8 -*-
"""
Deep Learning Based Anomaly Detection Model
"""
# Imports
from __future__ import print_function
import torch
from   torch.autograd import Variable
```

python

PYTORCH

theano

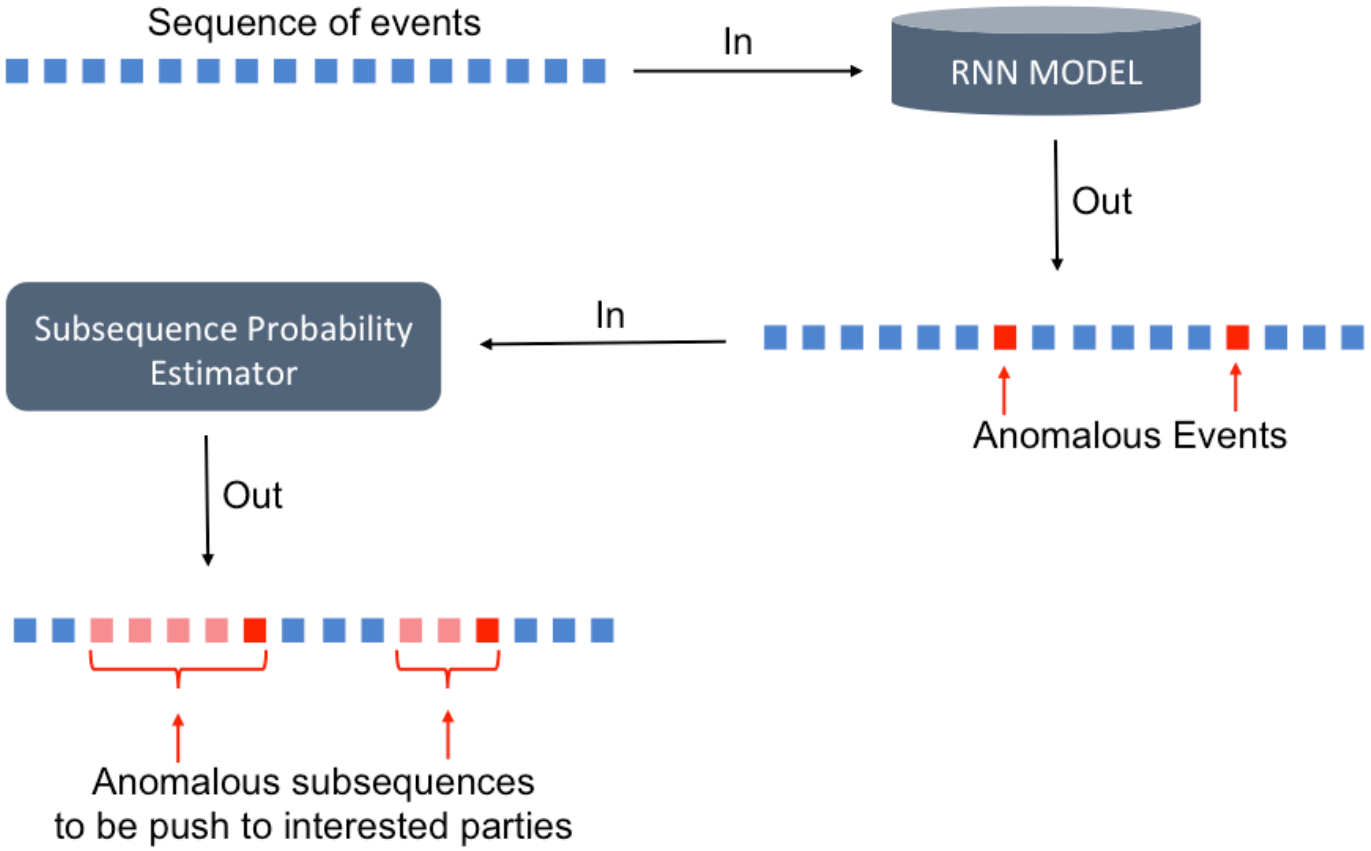
TensorFlow™

CUDA

GPU

- ❑ **Model**
  - ✓ Based on Recurrent Neural Networks (RNN): Long-Short Term Memory (LSTM) and Gated Recurrent Unit (GRU)
- ❑ **Programming Language: Python**
  - ✓ Compatible with several numerical computing libraries suitable for the use of GPUs
  - ✓ Some examples: Pytorch, TensorFlow and Theano
- ❑ **Computing Library: PyTorch**
  - ✓ Scientific computing package
  - ✓ Replacement of Numpy to take advantage of the power of GPUs
  - ✓ Python-friendly platform for neural networks (Deep Learning)
- ❑ **Parallel Computing Platform: CUDA**
  - ✓ Programming model to use GPUs for general purpose computing

Use case: Anomaly detection applied to Targeted Information Propagation



Cognitive Computing

Solutions based deep learning & **light-weight Machine Learning** models

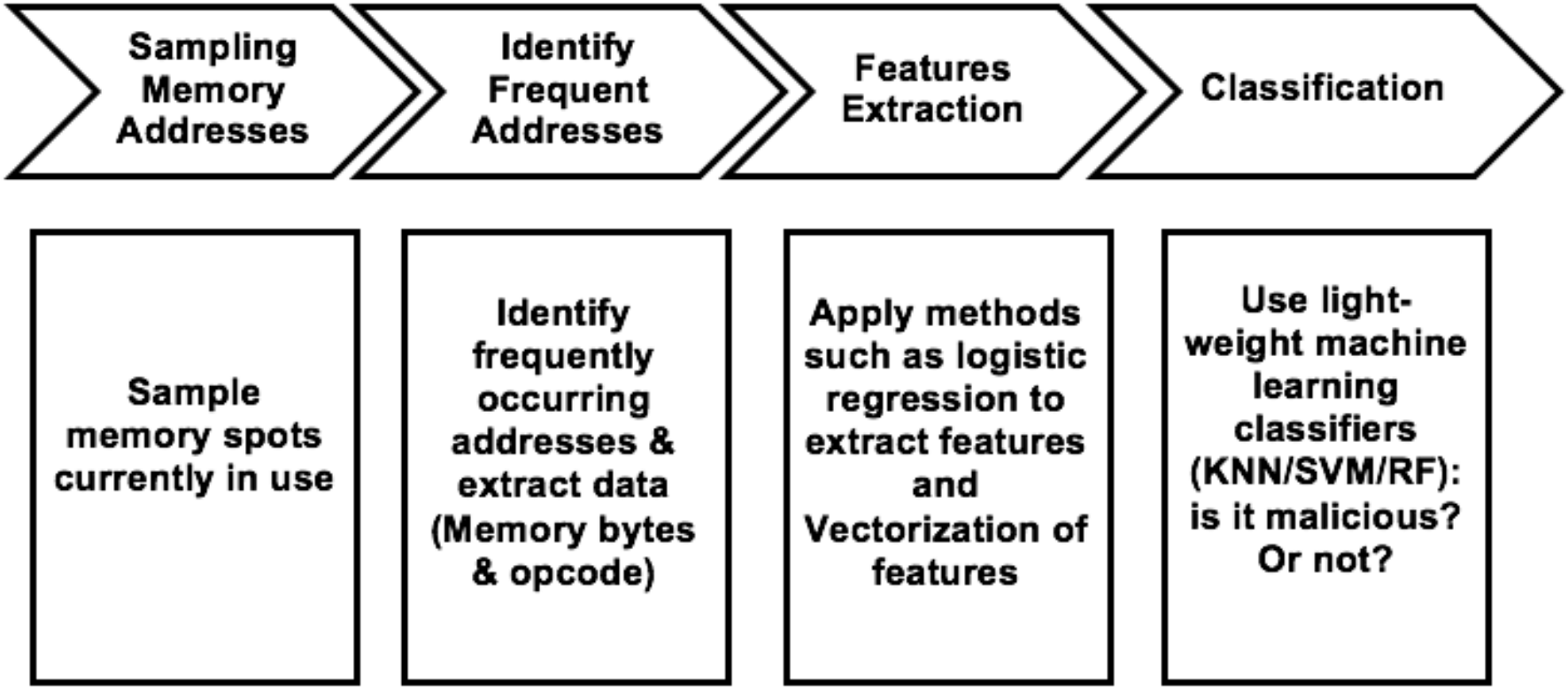
Malware / Anomaly Detection with Light-weight Machine Learning Models

NORTHROP GRUMMAN

- **Problem Statement:**
  - In mission critical systems, it is paramount that the system is mindful of it's resources.
  - The autonomous system should spend minimalistic amount of computing power to detect malware and anomalies.
  - IAS should conduct efficient featurization to increase the accuracy of machine learning models.
  - IAS should be able to switch it computation to secondary module when dealing with attacks.

Malware / Anomaly Detection: Workflow

NORTHROP GRUMMAN



Malware / Anomaly Detection: Intrusion Detection Sampling

NORTHROP GRUMMAN

- Sampling parameters: hardware parameters (addresses, instructions, memory bytes, etc.), software parameters (provenance, operational attributes, etc.), and operating system parameters (system calls, memory reads, etc.).

- Fixed sampling (e.g. once every 10 million instructions) and variable sampling will be conducted to sample data items in order to extract features.

- Sampling interval as well as detection interval will be varied based on the intrusion detection accuracy and computational context.

Malware / Anomaly Detection: Algorithm

NORTHROP GRUMMAN

- Conduct Fixed or Variable Sampling of hardware, software, and/or operating system data.
- Identify frequently appearing data items: number of times a particular memory address or system call is used.
- Extract information (e.g. opcode from frequent addresses).
- Perform feature extraction (unigram, bigram, n-gram, unique features, or cosine similarity).

$$\text{Similarity}\ (V_1,\ V_2) = \frac{V_1 \cdot V_2}{||V_1||\,||V_2||} = \frac{\sum_{i=0}^{n} V_1^i\, V_2^i}{\sqrt{V1_i^2}\sqrt{V2_i^2}}$$

- Use k-means, SVM, RR, and kNN for classification.

Experiment: Malware/Anomaly Detection

NORTHROP GRUMMAN

- **Objective:** To detect malware and anomalous behavior in autonomous cyber systems.
- **Input:**
  - HW / SW / OS data.
- **Output:**
  - Binary classification of whether a program is malicious or benign based on the features.
  - Performance parameters: CPU Usage, detection accuracy.
- **Experimental Setup:** Kernel drivers will be installed to extract data. Known malwares will be installed and tested.

Distinctive Attributes and Advantages

NORTHROP GRUMMAN

- ***Cyber Attribution Module with PROV-O*:** The cyber attribution model will be implemented with three major sub modules within the data stream processor.

- ***Online Intrusion Detection*:** Our cyber attribution model also provides online intrusion detection using sampling on hw/sw/os data.

- ***Adversarial Machine Learning Model:*** We will design data filtering mechanisms using logistic regression, clustering, and outlier prediction based on the origination of data.

Distinctive Attributes and Advantages

NORTHROP GRUMMAN

- ***Cognitive Computing for Malware / Anomaly Detection*:** Our cognitive computing module uses both deep learning such as RNN and light-weight machine learning methodologies such as SVM/RF/KNN to conduct malware / anomaly detection.

- ***Knowledge Discovery with Reasoning Engine*:** The objective is to learn about data about attacks and anomalies.

Tangible Assets Created by Project

NORTHROP GRUMMAN

- **Cyber Attribution Module:** This software prototype will consist of two parts: (1) knowledge representation and (2) provenance ontology (PROV-O) structure.
- **Intrusion Detection Application:** Application profile will be created through instructions based on supervised learning to classify benign and malicious applications.
- **Cognitive Computing Module:** This module leverages provenance and HW/SW/OS data to conduct ML analytics.
- **Reasoning Engine:** Reasoning engine module will enhance the reflexivity module and is supported by the knowledge discovery module.

Tangible Assets Created by Project

NORTHROP GRUMMAN

- **Cyber Attribution Module:** This software prototype will consist of two parts: (1) knowledge representation and (2) provenance ontology (PROV-O) structure.
- **Intrusion Detection Application:** Application profile will be created through instructions based on supervised learning to classify benign and malicious applications.
- **Cognitive Computing Module:** This module leverages provenance and HW/SW/OS data to conduct ML analytics.
- **Reasoning Engine:** Reasoning engine module will enhance the reflexivity module and is supported by the knowledge discovery module.

Integration with NGCRC & IRADs Projects

NORTHROP GRUMMAN

- Our research exclusively contributes to both Cyber Resilience IRADs and Data Analytics and Autonomy IRADs.

- We have discussed these projects and research with Paul Conoval and obtained feedback from Jason Kobes during the annual meeting for NGCRC and NGC TechExpo in May, 2018.

- We plan to participate in the DARPA OFFensive Swarm-Enabled Tactics (OFFSET) program where Northrop Grumman serves as a swarm systems integrator.

Plan for 2019 Meeting

NORTHROP GRUMMAN

- Prototype and demonstration of the workflow of the property of autonomy with
  - Anomaly Detection
  - Cognitive Autonomy
  - Reflexivity
  - Knowledge Discovery

- Collaboration and Integration with NGC Projects.

- We plan to respond to the NSF Grant proposal on Operationalizing Machine Learning for Command & Control (OMLC2) Program: https://www.fbo.gov/index.php?s=opportunity&mode=form&id=1547d6c1d89a67e76432ae67e918c732

Thank you!!!