CYBERSECURITY RESEARCH Consortium

NGCRC Project Proposal

for

Secure Intelligent Autonomous Systems with Cyber Attribution

October 6, 2018

Prepared for
The IS Sector Investment Program

Prepared by
Bharat Bhargava

Table of Contents

List of Tables

1 Executive Summary

Title	Secure Intelligent Autonomous Systems with Cyber Attribution		
Author(s)	Bharat Bhargava		
Project Lead	Bharat Bhargava		
University	Purdue University		
Requested Funding Amount	$160,000		
Period of Performance	November 1, 2018 - October 31, 2019		
Is this an existing Investment	Yes	**TRL Level of Project**	5
Key Words	Autonomous systems, anomaly detection, intrusion detection, malware detection sampling, data analytics, deep learning, neural networks, cyber attribution, data provenance, privacy preserving, ontological reasoning		
Key Partners & Vendors			
NGC projects you have collaborated with in the past	Intelligent Autonomous Systems based on Data Analytics and Machine Learning; Secure / Resilient Systems and Data Dissemination / Provenance; Context-based Adaptable Defense Against Collaborative Attacks in SOA; End-to-End Security Policy Auditing and Enforcement in Service-Oriented Architecture; Monitoring-Based System for E2E Security Auditing and Enforcement in Trusted and Untrusted SOA; Privacy-Preserving Data Dissemination and Adaptable Service Compositions in Trusted & Untrusted Cloud;		

Table 1: Executive Summary

1.1 Abstract

Intelligent Autonomous Systems (IAS) reconstruct their perception through adaptive learning and meet mission objectives. IAS are highly cognitive, rich in knowledge discovery, reflective through rapid adaptation, and provide security assurance. It is paramount to have effective reasoning, decision-making, and understanding of operational context since IAS are exposed to advanced multi-stage attacks during training and inference time. Advanced malware types such as file-less malware with benign initial execution phase can mislead IAS to accept them as normal processes and execute malicious code later. IAS are also exposed to adaptive poisoning attacks where adversary inputs malicious data into training/testing set to manipulate the learning. Hence it is vital to monitor IAS activities/interactions to conduct forensics.

This project will advance science of security in IAS through multifaceted advanced analytics, cognitive and adversarial machine learning, and cyber attribution. We plan to closely work with Paul Conoval, Robert Pike, and Jason Kobes and contribute to Cyber Resilience and Autonomy IRADs based on the following approaches.

(1) Implement deep learning based application profiling to categorize adaptive cyber-attacks and poison attacks on machine learning models using contextual information about the origin, trust, and transformation of data.

(2) Using HW/OS/SW data to develop perception algorithms using LSTM deep neural networks for detecting malware/anomalies and classifying dynamic attack contexts.

(3) Facilitate cyber attribution for forensics through privacy-preserving provenance structure for knowledge representation and perform intrusion detection sampling on HW/OS/SW data. We will collaborate with Dr. Lalana Kagal from MIT.

(4) Employ advanced data analytics to aid ontological and semantic reasoning models to enhance decision-making, attack adaptiveness, and self-healing.

1.2 Graphical Illustration

Our focus is on constraints, barriers and challenges such as poorly understood attack surfaces, data set training availability and biases, processing latency, human understanding of AI results, AI/ML countermeasures, human-machine disparity, measurement of effects. We propose novel approaches for privacy-preserving cyber attribution, intrusion detection, adversarial machine learning, malware/anomaly detection, reasoning, and decision-making. Cyber attribution involves extracting software, hardware, and operating system data to perform intrusion detection sampling (fixed or dynamic sampling), generating efficient provenance structure that is populated with specific data required for a particular analysis or learning, and labeling and tagging to properly represent the information obtained. The processed data is distributed to the cognitive module where the data is checked for any malicious data presence through poison attack filter. The filtered data is transmitted to cognitive computing module and knowledge discovery module, where the data is fed into supervised, unsupervised, and LSTM models to perform learning and advanced analytics. Based on multifaceted dimensions of data analytics, reasoning and decision-making ability of IAS are enhanced. The proposed comprehensive and unified architecture contributes to the holistic characteristics expected by NGC in their autonomous systems. The overall architecture of the proposed model—secure intelligent autonomous systems with cyber attribution—is demonstrated in figure 1.

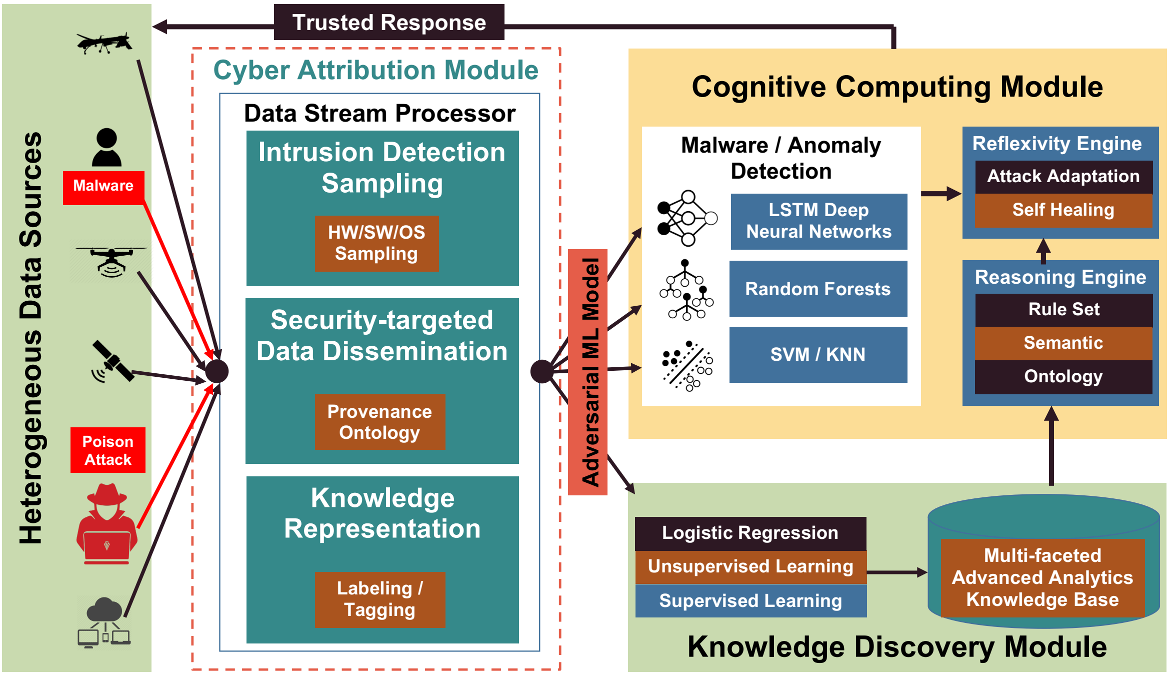

Fig. 1 Comprehensive Architecture of Secure Intelligent Autonomous Systems with Cyber Attribution

General characteristics of the proposed unified architecture are given as follows:

- Intelligent autonomous systems receive large amounts of diverse data from various data sources. In addition, they operate in a dynamic operational context and interact with numerous entities such as other IAS, UAVs, satellites, sensors, cloud systems, analysts, malicious actors, and compromised systems.
- Cyber attribution module constitutes a stream data processor where data streams are labeled / tagged on-the-fly for better knowledge representation and categorization. This data is stored as monitored or provenance data with its origin and historical information. For preserving privacy, detailed provenance data is reduced in its scope to include only necessary data for a particular analysis or learning. This module uses Provenance Ontology (PROV-O) structure (elaborated in a later section) to obscure unnecessary or privacy-compromising data. Furthermore, the attribution model monitors data generated by software (application parameters), hardware (memory bytes and instructions), and operating system (system calls). This data is used to conduct periodic sampling to identify signatures of intrusion activities.
- Once the data is processed, it goes through adversarial machine learning model. Attackers can insert malicious data into training and testing dataset to influence machine learning models. In order to isolate poisonous data, poison data filter performs methods such as classification of verified and unverified data as well as outlier extraction. Once the poisonous data is removed the data (raw or provenance data) is sent to Cognitive computing module and Knowledge discovery module.
- In Cognitive computing module, depends on the data and efficiency of machine learning methods, malware / anomaly detection is performed through either deep learning

methodologies such as Long short-term memory (LSTM) e.g. Recurrent Neural Networks (RNN) or Convolutional Neural Networks (CNN) or light-weight yet powerful machine learning methods such as Support Vector Machines (SVM), Random Forests (RF), and K-Nearest Neighbors (KNN). In addition, cognitive computing module consists of reasoning engine, which is driven by rule sets, semantic, and ontological reasoning. Both anomaly detection module and reasoning engine module influence the attack adaptiveness (reflexivity) and self-healing of IAS, where decisions obtained through reasoning and learning are turned into actions. With this extensive cognitive computing modules, the final response from IAS to other interacting entities will be a secure and trusted one.

- Knowledge discovery module facilitates multi-faceted dimensions of advanced data analytics including regression analysis, supervised learning, unsupervised learning, and pattern-recognition. Discovered knowledge is shared with cognitive computing module for further learning. The proposed structure provides robust cyber resilience and autonomous operation of the system.

2 Description of Project

The combination of advanced analytics, artificial intelligence and cyber security, will create a new paradigm in systems development and deployment of mission systems. The additional element of autonomous operations and the rapid timing dynamics required of system responses will also present new challenges. Gaining confidence in autonomous decision-making and trusting non-human responses will be a significant consideration in the deployment of practical systems that leverage artificial intelligence and advanced analytics. The technologies within this domain vary widely in levels of maturity and sophistication. However, as advances in machine learning, cognitive analytics and the enabling hardware computing platforms are rapidly being developed, there will be considerable new classes of capabilities and opportunities introduced by industry and government that will help serve the mission objectives. Multi-faceted dimensions of advanced analytics and artificial intelligence present new challenges and opportunities to mission systems development [24]. The project provides a unified and comprehensive architecture to enhance security in Intelligent Autonomous Systems with three major modules: (1) Cyber attribution module, (2) Knowledge discovery module, and (3) Cognitive computing module. The comprehensive solution provides privacy-preserving cyber attribution with efficient provenance data structure, sampling on SW/HW/OS data to profile the signatures of intrusion, removing poisonous data from training and testing datasets, detecting malware / anomalies by LSTM networks and light-weight machine learning techniques (SVM, KNN, RF, etc.), reasoning, and reflexivity combined with advanced analytics with supervised and unsupervised learning.

2.1 Statement of Problem

Intelligent Autonomous Systems (IAS) reconstruct their perception through learning and meet mission objectives. IAS are intelligent learning systems that are cognitive, perceptive through rich knowledge discovery, reflective, and assure security and privacy. When considering ML/AI decision making in autonomous systems, the security, trust, verification, reasonableness and understanding is paramount for practical utilization in real systems. AI systems and learned systems at training and inference time pose novel demands on compute infrastructure and present new risks. Advanced malware types such as file-less malware and ransom ware with benign initial execution phase can mislead IAS to accept them as normal processes and can execute malicious modules once the trust is established. Adversarial machine learning attacks through poisoning attacks where adversary inputs carefully crafted adversarial data into training and testing set to manipulate the learning pose a significant threat to model-oriented IAS. These types of cyber-attacks are adaptive and adjust their attack patterns to mask their intentions to exploit IAS.

Research problems are (a) adversarial machine learning, identifying threats to the training and deployment of learned systems, and develop corresponding defense strategies. (b) Develop verifiable explanation or proof to trust the activities of an IAS system and correctness of the system operations. 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. The self-monitoring capability for IAS systems requires machine learning to infer reasonableness of state. This will be achieved by maintaining detailed data provenance, track data as it flows through the system, enable lineage and trusted attribution, and enable auditing, (c) Cyber attribution that helps us gain insights into the source of the AI models and training sets and determining whether or not they can be trusted or are perhaps tainted by an adversary. Privacy issues factor in by inhibiting the system's ability to attribute sources of effects (positive or negative); (d) Deep learning based anomaly detection that can detect variable-length anomalous sequences that span for longer periods of time.

2.2 State of Current Technology and Alternatives

Autonomous systems must be highly cognitive, multitasking, reflexive, rich in knowledge discovery [1], and guarantee assurance of security and privacy. The convergence of analytics, artificial intelligence and advanced cyber security is presenting emerging opportunities, issues and threats, impacting mission deployable systems, operationalizing artificial intelligence into practical deployable systems requires an end-to-end, systems-level design approach, consideration of the multi-faceted dimensions of the AI/ML Enterprise allows for scaling the appropriate level of AI/ML for effectively and efficiently meeting mission objectives [24]. Our solution constitutes cyber attribution, knowledge engineering, and cognitive autonomy with machine learning. Over the years, a number of advances have been made in the specified topics.

Cyber Attribution:

Attribution for information assurance is nothing but tracing back attackers step and finding the attacker's origin as well as the attack's origin. In the process, cyber attribution has the ability to reveal the step-by-step progression of the attack and attacker's knowledge (security loopholes, external knowledge, system information, reconnaissance time etc.). Some of the traditional attribution techniques were proposed in [2]. 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. But the growth of sophistication of attacks and attackers are increasing exponentially [3], which demand adaptive and intelligent cyber attribution techniques. Intelligent Cyber Attribution (InCA) framework was proposed in [4]. The approach relies on two models: (a) environmental model (EM) to attribute the background knowledge and is probabilistic and (b) analytical model (AM) to analyze several hypotheses that can explain a particular event in cyberspace. The framework combines argumentation, probabilistic models, and logical programming to attribute a certain operation executed by a cyber system. Other non-adaptive yet effective methods are also available. Revere-engineering is proposed as a solution [5] to attribution problem where the old attacks are reproduced and new attacks are created to test the resiliency of the system. The author primarily focuses on developing offensive techniques first and design resilient modules to counter that offense. Source tracing through unsupervised learning (clustering) is proposed in [6]. The technique identifies effective features from the dataset and cluster the sources based on their attack resiliency, vulnerabilities, and computational capacity. Honeypots [7] are widely used to lure attackers to interact with the system and get information about the attacker and the planned attack. Value of the honeypot increases when an attacker finds it valuable to interact i.e. misleading the attacker to give his plan away. But the sophisticated attacks only focus on hands-off reconnaissance to observe the behavior of the system. Sinkholing [8] is a sophisticated attribution technique where the systems used in attacks are used as command and control centers to observe the incoming communication from other compromised systems. Present day attacks are sophisticated and adaptive in nature and they require an adaptive cyber attribution solution. This project includes proper knowledge representation techniques and privacy-preserving provenance data structure to conduct analytics on data to detect intrusion and profile attacks and attackers.

Adversarial Machine Learning:

Adversarial machine learning constitutes adaptive machine learning techniques used against dynamic attack techniques of malicious actors [9]. In recent development, adversarial machine learning is needed to defend against to adaptive Sybil attacks and poisoning attacks [10]. Several methods have been proposed in literature to mitigate Sybil attacks in learning poisoning. FoolsGold [11] is a novel defense proposed to thwart Sybil-based poisoning attacks. One of the main features of the design is that benign clients can be separated from duplicates based on the diversity of their gradient updates i.e. FoolsGold values clients that provide honest gradient updates. Similarly, authors in [12] proposed taxonomies to use machine learning classes to increase the cost of the attackers in Sybil attacks. They explore the efficiency of Reject On Negative Impact (RONI) defense's cost with the attack's cost on the SpamBayes—a famous spam filer. Another way of manipulating machine learning model is through causative or poisoning attacks where adversary inserts malicious data into training set. Using contextual information is about the origin and transformation of data expose poison data is proposed in [13]. Here the model conducts online detection and updates the model for every attack or attack-attempt. In this project, we propose adaptive machine learning techniques using LSTM networks to retrain the model. We propose a method that conducts online detection only when triggered, saving valuable computing resources.

Cognitive Autonomy and Knowledge Discovery for Security in IAS:

Cognitive computing is a vital part of security in autonomous systems. In particular, malware and anomaly detection has become a biggest challenge with increase in sophistication in attacks such as file-less malware [14] and ransomware [15]. Behavior-based malware detection system (pBMDS) was proposed in [16]. The technique observes unique behaviors of applications as well as users and leverages Hidden Markov Model (HMM) to learn application and user behaviors based on two features: process state transitions and user operational patterns. One of the drawbacks of the HMM model is that it has very limited memory thus cannot be used for sequential data. In this project, we leverage hardware, software, and operating system data and apply long short-term memory units to identify anomalous behavior. We will also profile applications and malware using HW data (memory bytes and instruction sequences) to whitelist benign processes and blacklist malicious processes. In order to enable better results for LSTM deep learning methodologies, knowledge discovery and representation are important. We proposed a metadata labeling scheme, BFC, for information tagging and clustering by reversing the error correction coding technique known as Golay coding [17]. The scheme utilizes 2^{23} number of binary vectors of size 23 bits to profile features and cluster the data items. Since the method is built based on error correction scheme, it exhibits fault tolerance in wrongly labeled data. Similarly, we perform privacy-preserving knowledge discovery through perturbed aggregation in untrusted cloud [18]. In this project, we will use advanced data analytics to enable reasoning module for assisting attack adaptation and reflexivity of the system.

Attack Adaptation (Reflexivity) of IAS:

Under ongoing attacks IAS should continue to fulfill mission objectives without shutting down completely. IAS should gracefully degrade i.e. continue to function in a degraded state and incrementally learn the new environment. It should adapt and self heal. We implemented a combinatorial balanced block design to implement incremental learning through graceful degradations [19]. The system uses efficient replacement of replicas when primary module is under attack and uses Bayesian inference to periodically update the replicas from primary module. In this way, replicas can take over when the primary module is undergoing self-healing.

2.3 Proposed Solution and Challenges

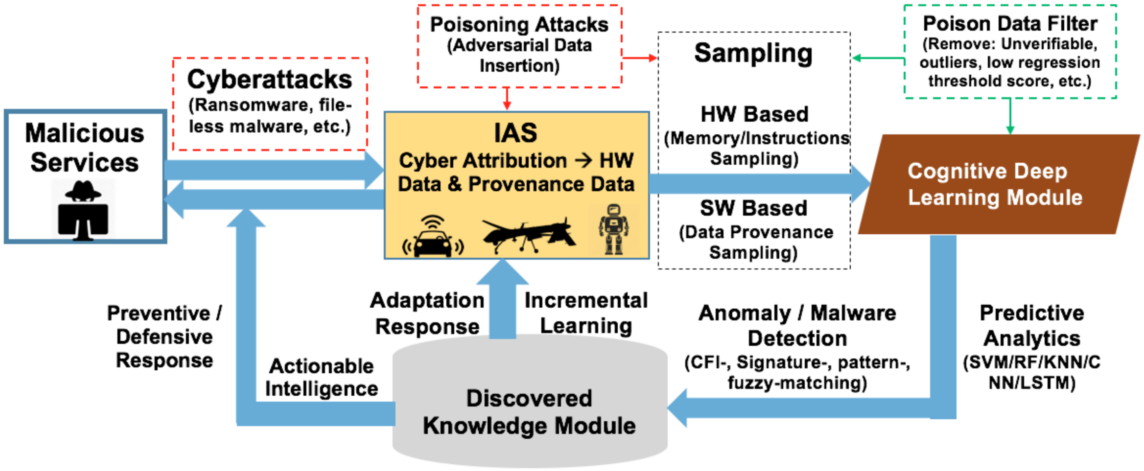
Poisoning Attacks
(Adversarial Data Insertion)
Sampling
Poison Data Filter
(Remove: Unverifiable, outliers, low regression threshold score, etc.)
Cyberattacks
(Ransomware, file-less malware, etc.)
Malicious Services
IAS
Cyber Attribution → HW Data & Provenance Data
HW Based
(Memory/Instructions Sampling)
SW Based
(Data Provenance Sampling)
Cognitive Deep Learning Module
Preventive / Defensive Response
Actionable Intelligence
Adaptation Response
Incremental Learning
Discovered Knowledge Module
Anomaly / Malware Detection
(CFI-, Signature-, pattern-, fuzzy-matching)
Predictive Analytics
(SVM/RF/KNN/CNN/LSTM)

Fig. 2 Proposed Approach for Secure Intelligent Autonomous System

The components of a secure IAS are shown in figure 2. We build IAS that provide security, learns from training data, understand its capabilities, and adapt to meet mission objectives with little or no human intervention, and prevent and defend against malware and poisoning attacks. The system employs incremental learning: progressive enhancements or graceful degradations to meet the mission objective under cyber-attacks. The following are research directions and approaches:

(a) Behavior-based analytics is used to categorize adaptive cyber-attacks and poison attacks on ML. We develop a methodology that uses contextual information about the origin and transformation of data points in the training set to identify poisonous data. This research considers both trusted test data set and partially trusted data set or untrusted data sets as needed in distributed learning models. While it is difficult to prevent adversaries from manipulating the environment around the source of information, IAS ensures that the provenance information is secured and cannot be tampered with and remains in an immutable storage system such as a block chain where the origin of data points cannot be faked. The detection service allows IAS to filter poisonous data with the help of provenance information. Poisoning attacks on ML will be removed by detection of outliers and unverifiable data from training and testing datasets. Further, we will quantify the usability of the sampled data through regression model and customized F-score. which will be used to filter the unwanted and adversarial data out. Using our past NGCRC research, collusive and Sybil attacks on ML are mitigated.

(b) Perception algorithms based on deep learning are developed to enable cognitive autonomy to classify and interpret dynamic contexts.

(c) 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. Since a clever attacker can provide misleading information to hide the true attacker, blocking an attack may be more effective if an intermediary is known.

(d) Since IAS can be infected by advanced malware such as ransom ware (encrypt IAS data without authorization) and file-less malware (a deceptive memory-based artifact without native application). 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.

2.3.1 Cognitive Autonomy:

Decentralized machine learning is a promising emerging paradigm in view of global challenges of data ownership and privacy. We consider learning of linear classification and regression models, in the setting where the training data is decentralized over many user devices, and the learning algorithm must run on device, on an arbitrary communication network, without a central coordinator. We plan to utilize and advance COLA, a new decentralized training algorithm [23] with strong theoretical guarantees and superior practical performance. This framework overcomes many limitations of existing methods, and achieves communication efficiency, scalability, elasticity as well as resilience to changes in data and participating devices. We will consider fault tolerance to dropped and oscillation of nodes from connected to disconnected and attacks on the nodes. The learning has to be communication-efficient decentralized framework and free of parameter tuning. COLA offers full adaptively to heterogeneous distributed systems on arbitrary network topologies, and is adaptive to changes in network size and data, and offers fault tolerance and elasticity. IAS should have clear understanding of its operational context, it's won processes, and its interactions with neighboring entities. In this project, the cognitive computing module consists of three major components: (1) Malware / anomaly detection module, (2) Reasoning engine, and (4) Reflexivity engine. Cyber attribution data (system monitoring data or provenance data) is sent to cognitive computing engine for analysis where the system profiles the applications based on machine learning models.

Malware / Anomaly Detection with Deep Learning Model:

As an anomaly is defined as a non-conforming pattern of events with respect to the learned model, it is possible to detect anomalies by analyzing the sequence of events resulting from system activities. For a specific system, each possible event must be uniquely identified to form the vocabulary of systems events. At any time t each possible event is assigned a probability estimated with respect to the sequence of events observed until time $t-1$. At classification time t, the decision is made with respect to a pre-defined threshold of the top-k most likely sequences as shown in Algorithm 1.

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

- **Step 1**: Define a finite set E of events $\{e_1, e_2, \ldots, e_N\}$ in the system. Events occur in a time-series fashion.
- **Step 2**: At time $t-1$ given an observed series of $\{{e_i}^1, {e_i}^2, \ldots, {e_i}^{t-1}\}$ (with $i = 1, 2, or\ N$), find the set K of the top k events to occur in time t. The size of the set K varies in each prediction and is determined by natural clusters in the output of the model.
- **Step 3**: At time t, the sequence $\{{e_i}^1, {e_i}^2, \ldots, {e_i}^{t-1},\ {e_i}^t\}$is non-anomalous if ${e_i}^t$ is in K, otherwise anomalous.

Algorithm 1. Anomaly Detection Algorithm

Fig. 3 shows the process of anomaly detection and an example how it can be applied to solve problems such as targeted information propagation. At any time t the probability of the new event is estimated using the RNN-based model. If the probability is low (determined by the natural clusters in the output of the model) the relevant subsequence (anomalous subsequence) formed by the new event and N predecessors (N determined by a probability threshold of the subsequence) is push to the interested parties for further analysis.

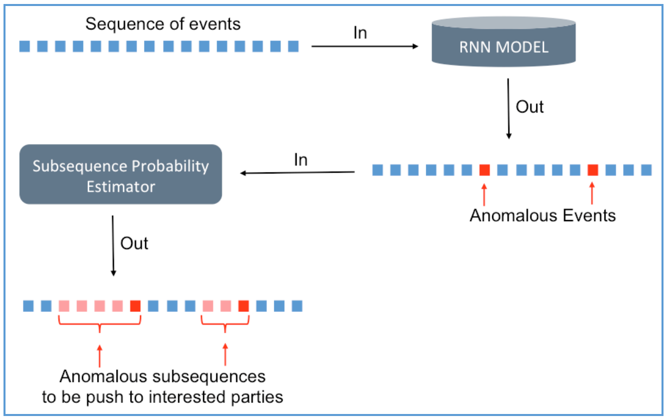
Sequence of events
In
RNN MODEL
Out
Subsequence Probability Estimator
In
Anomalous Events
Out
Anomalous subsequences to be push to interested parties

Fig. 3. Anomaly detection applied to targeted information propagation

LSTM networks, a type of recurrent neural network (RNN), have been successfully applied to many sequence learning tasks such as speech recognition, machine translation and natural language generation. The reason of this success is that the LSTM cell includes a special unit called *cell state (C)*, a vector with information that is passed as input in each iteration of the recurrent algorithm. Figure 3 shows the architecture of the LSTM cell in a simplified manner. The explanation that follows about how LSTM works is completed with the equations in Figure 4, which show the exact operations in each step.

- Input (x_t): Element of the sequence at time *t*. At time *t* all the other components, but the output, are values obtained at time *t-1*. Output (y_t): Estimated output at time *t*, which corresponds to the resulting hidden state at time *t*. See explanation below. Candidate vector ($\tilde{C}$): The input is combined with the previous hidden state (see below) to form the candidate vector, which is passed to the input gate.
- Input gate (i): An elementwise sigmoid operation is done over the elements of the candidate vector. The result is that some elements will become 0 and others will become 1 (i.e. only some elements of the input vector are selected to modify the state obtained until time *t-1*). Cell State (C): The vector in the cell state estimated at time *t-1* is passed to the forget gate.
- Forget gate (f): An elementwise sigmoid operation is done over the elements of the cell state vector. The result is that some elements will become 0 and others will become 1 (i.e. some elements of the cell state are discarded/forgotten and do not have an impact in the estimation of the new cell state). The output of the forget gate is combined with the output of the input gate to get the new cell state at time *t*. An elementwise hyperbolic than operation is then applied to the new cell state to keep values between -1 and 1 to prevent gradient vanishing and gradient explosion. The result is passed the output gate.
- Output gate (o): An elementwise sigmoid operation is done over received vector. Some elements of the vector will become 0 and others will become 1 (i.e. only some are considered to for the new hidden state). Hidden State: Considered the output at time *t* and used as hidden state to estimate the candidate vector at time *t+1*.

Malware / Anomaly Detection with Light-weight Machine Learning Models:

We will implement anomaly detection with hardware data (memory bytes and instructions) and light-weight machine learning algorithms (Fig. 4).

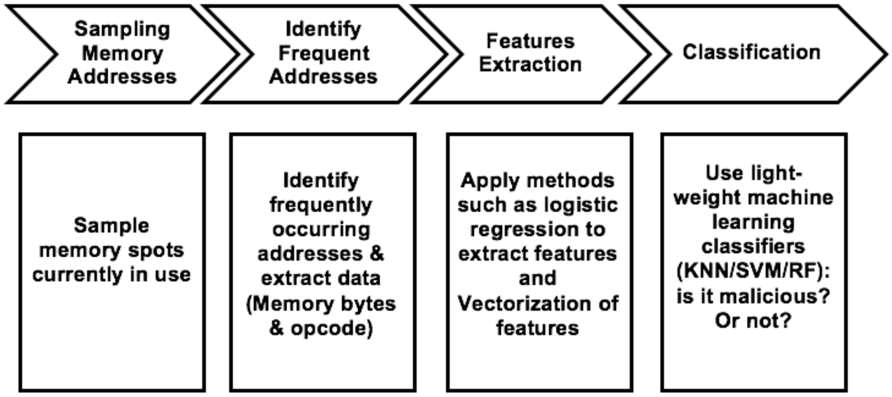
Sampling Memory Addresses
Identify Frequent Addresses
Features Extraction
Classification
Sample memory spots currently in use
Identify frequently occurring addresses & extract data (Memory bytes & opcode)
Apply methods such as logistic regression to extract features and Vectorization of features
Use light-weight machine learning classifiers (KNN/SVM/RF): is it malicious? Or not?

Fig. 4. Malware/anomaly Detection with Light-weight Machine Learning Methods

Advanced malware such as ransomware encrypts IAS data without authorization. Since it does not alter the system configurations and leave a footprint, it is difficult to detect them. But based on the executed instruction sequences and constants (also know as magic constants) used for encryption mechanism during malware execution, applications can be profiled. First, we will sample the address spots for every 1,000,000 instructions (fixed sampling). After a fixed period of time, we will calculate the frequently occurring addresses and their relevant process ids. A threshold T will be set for data extraction. For example, extract memory bytes and instructions from top T = 10% of the global list of sampled addresses (sorted in descending order based on their frequency of occurrence). Once opcode and memory bytes data is collected, we will extract features such as n-gram, bigram, unigram features, magic constants feature, cosine similarity with instructions occurrences, and standard deviation. Cosine similarity metric is one of the most efficient method to learn from large datasets [20]. It plays a crucial role in understanding similarity between two feature vectors when the magnitude of the vector is large or unspecified i.e., it can either be unigram, bigram, or n-gram features. Given two feature vectors $V_1 = \{f_{11}, f_{12}, \ldots\}$ and $V_2 = \{f_{21}, f_{22}, \ldots\}$, where $f_{11}, f_{21}, \ldots$ are values of a particular feature, the cosine similarity is given as,

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

The cosine similarity lies between 0 and 1. If the orientation of the two feature vectors is the same then the similarity between them is Cos 0 = 1 i.e., there is zero angle between them. But when the angle is 90^o (the orientation of the feature vectors is at an angle of 90) then the similarity is Cos 90 = 0. The similarity score varies between $[0, \frac{1}{2}\pi)$. Once the features are extracted, we will implement RF, SVM, and KNN learning models. K-NN is one of the simplest yet powerful classifier with high computational efficiency as well as accuracy [21]. It clusters raw data items based on majority of similar sampled data items among k-nearest neighbors that are very close to the test sample. The similarity distance between training samples and test samples is determined by the cosine similarity score. In training phase, the training samples and the class labels are stored. In case of streaming raw new data, the classifier can be retrained in a specified time interval with new training and testing data that represents the current context in which the data was obtained. In the classification phase, each test sample is classified with a majority vote of its nearest neighboring data items:

1. Cosine similarities from the testing sample to every single training sample are calculated.
2. The k-nearest neighbor of the testing sample is selected, where k is an integer that can be determined through elbow method [22].
3. 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.

The elbow method is can be an effective way of determining the value of k. The idea of the elbow method is to compute k-means clustering on the data set to compute the values of k and for each value of k computer the sum of squared errors. The elbow method shows the steep drop of the sum of squared errors. The ideal value for k can be found at the “elbow” bent in the curve. Elbow occurs at 3 for the given dataset. So k can be chosen as 3 to produce optimal accuracy in KNN classifier.

2.3.2 Cyber Attribution:

Our cyber attribution model has two major components: (1) Provenance ontology structure (PROV-O) and (2) Sampling for intrusion detection. Provenance Ontology (PROV-O) structure—a domain neutral and simple data model with a set of characteristics, restrictions, and classes, and used to interchange provenance data generated in different systems under different operational contexts—to disseminate security-targeted data and obscure unwanted details. It consists of Entity (it is about provenance about the cyber entity), Activity (creation and changes of entities), Agent (actor who is responsible for those changes and creation), and Usage. Restricted and simple structure like PROV-O allows limited and only needed information to be made available. Hence it is better to use this for privacy preserving. We will be closely collaborating with Dr. Lalana Kegal on this research direction using PROV-O and its extensions for trustworthiness in IAS.

We will also model Cyber attribution as graph where each node (sensors, systems, etc.) store 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 IAS communication network during real-time execution of system functionalities. The graphs will be predefined for the training model and updated each time when there is an entity (node) sending new data. 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. By computing fuzzy matching between the generated probabilities and predefined training graphs, we will determine if there are any poisoning attacks or tampering in the data, or the source itself. `

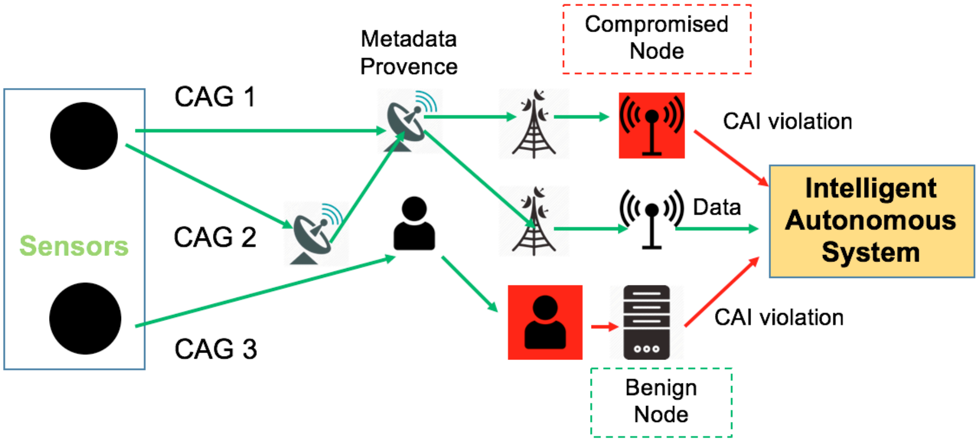

Fig. 5 Cyber Attribution in IAS

Fig. 5 shows the example scenarios of cyber attribution when a node is compromised as well as uncompromised. We will also employ machine learning to enhance cyber attribution in IAS. We will use the likelihood probabilities of CAI as a unigram feature (a histogram) light weight machine learning models such as one-class Support Vector Machine (SVM) and K-Nearest Neighbors (KNN) with cosine similarity to identify benign and malicious graphs i.e. CAGs that violate the CAI.

2.3.3 Adversarial Machine Learning Model:

IAS uses deep learning models and light-weight machine learning models to detect malware attacks and adapts to new attacks through its learning. Both models rely on training data sets to understand the underlying processes and classify them as benign or malicious. Attackers can insert carefully crafted data into the model training set that can corrupt the model and make the malicious process to deter the detection i.e. malicious actors can deploy evasive attacks and poisoning attacks. In order to be resilient against these attacks IAS needs to constantly update its training set and at the same time perform analytics using taxonomy and outlier detection to isolate poisoning data. We will work on offensive research where we create sophisticated attacks to infiltrate our model and design defense mechanisms.

- We will generate offensive models to infiltrate training set and observe the progression of the attacks in various states of learning.
- Based on the results, 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. and operating context awareness in terms of data admission i.e. a data item can be of expected type but the time of generation is not relevant.

2.4 Distinctive Attributes, Advantages, and Discriminators

We will employ robust learning models based on deep learning and reasoning for detecting insidious attacks on IAS and privacy-preserving cyber attribution.

- ***Cyber Attribution Module with PROV-O***: Cyber attribution model will be implemented with three major sub modules within the data stream processor. As the data is generated, knowledge representation module will use our bitwise fuzzy-based clustering scheme to label raw data. Similarly, depends on the type of interaction and requirements provenance data will be generated. Initially, provenance is captured raw without obscuring any details. Before the data is distributed for machine learning analytics, type of learning and data analytics are determined. We will use Provenance Ontology (PROV-O) structure—a data model with a set of characteristics, restrictions, and classes, and used to interchange provenance data generated in different systems under different operational contexts—to disseminate security-targeted data and obscure unwanted details. This mechanism provides privacy-preserving advantage by obscuring unnecessary provenance data and providing data specs needed for particular learning and analytics.
- ***Online Intrusion Detection***: Our cyber attribution model also provides online intrusion detection module using fixed sampling on hardware (memory bytes and opcode instructions), software (application and provenance data), and operating systems (system calls) data. The intrusion detection mechanism is very efficient since it gets triggered only when certain behaviors are exhibited in the system thus reducing the use of computing resources. For example, our intrusion detection model for ransomware only gets triggered when there are high CPU usage since ransomwares use extensive computing power to perform crypto calculations to encrypt IAS data without authorization. Another optimization utilized in our

online intrusion detection technique is fuzzy matching of intrusion malware signatures. Once the instructions are sampled from frequently used memory addresses, they are fuzzily matched with known signatures of previous intrusion to make a probabilistic decision.

- ***Adversarial Machine Learning Model:*** We will design data filtering mechanisms using logistic regression, clustering, and outlier prediction based on the origination of data (trusted data from trusted data source). Based on the outliers and clusters with a few anomalous data points are removed from the training set to make the machine learning model consistent. Our adversarial machine learning model also uses periodic adaptation technique: when the training set gets updated, outlier detection is executed and poisonous data is removed periodically. It incurs less computational cost since the model is trigger only when a new training set is obtained.
- ***Cognitive Computing for Malware / Anomaly Detection*****:** Our cognitive computing module uses both deep learning such as Recurrent Neural Networks and light-weight machine learning methodologies such as SVM/RF/KNN to conduct malware / anomaly detection. We use these methods depends on the features (unigram, bigram, and n-gram features) and unique malicious application feature (crypto constants in case of ransomware). If the features do not depend on expensive sequential data, then we use light-weight machine learning models. But if the data is streaming sequential data consisting of benign and new malicious sequences, we use LSTM methodologies. Hence our approach produces maximum detection accuracy with considerably less computational cost.
- ***Knowledge Discovery with Reasoning Engine*****:** Based on last year's NGC funding, we implemented bitwise fuzzy based clustering for effective knowledge discovery and scalable learning. In this project, we will enhance decision making by ontology and semantic reasoning on top of supervised and unsupervised learning. Once the data items are clustered, further reasoning is performed on top of each cluster to aid reflexivity of the IAS. Coupled with deep learning models and reasoning engine, this project produces a robust adaptive workflow for IAS under attacks.

2.5 Tangible Assets to be Created by Project

We will develop deployable machine learning models to achieve mission objective with limited data and under attacks.

2.5.1 Software

Cyber Attribution Module: This software prototype will consist of two parts: (1) knowledge representation and (2) provenance ontology (PROV-O) provenance structure. Based on the operational context and machine learning analytics requirement, PROV-O will be populated with specific data and sent for analysis and learning. Knowledge representation will include an efficient labeling scheme similar to error correcting code scheme that we proposed in [17]. In addition, metadata tagging will be introduced, where the analysis will just use the metadata to make predictions, which will preserve privacy.

Intrusion Detection Application: The application will have the implementation of fixed sampling of memory addresses, algorithm to identify most frequently used memory addresses (hotspots), extraction of memory bytes and opcode (instructions) from hotspot addresses, and fuzzy matching to match the intrusion instructions. The application conduct online detection of any intrusions. Windows kernel driver and its libraries will be used for data collections.

Cognitive Computing Module: This application leverages provenance data, hardware data (memory dump and instructions), and operating systems data (system calls of malicious and benign process) to conduct machine learning analytics. Initially, the model processes features extraction and conducts light-weight machine learning algorithms such as KNN. If the data is sequential and streaming, the model switches to deep learning model (LSTM recurrent neural networks). The application gives accurate predictions and detections with considerably less computational cost.

Adversarial Machine Learning Model: The demonstration will be a combination of taxonomies and outlier detection methodologies that are used to filter out the poisonous data.

Reasoning Engine: Reasoning engine application will enhance the reflexivity module and is supported by knowledge discovery module. First, reasoning engine will constitute rule set, ontology, and semantic based inferences leveraging provenance data. Using unsupervised and supervised learning results from knowledge discovery module, further learning through reasoning will be initiated. This gives NGC most effective reasoning mechanism for mission adaptation.

2.5.2 Documentation

We will provide four types of documentations that would help NGC researchers. They include:

1. **Source code*:*** Code for the software will be well self-documented for possible extensions/modifications by future developers.
2. **Deployment and user manuals:** All software components created in the project will be clearly documented with deployment guides and user guides on how to use each component separately as well as how to use the whole prototype.
3. **Reports:** We will provide mid-term and final reports that describe algorithm implementations, and the experimental results that characterize the performance of the presented solutions. These results will include both system performance and security evaluation of the system.
4. **Demonstrations:** To be made at NGC meetings and to NGC researchers.

We will provide high-quality documents adhering to the standards used at NGC.

3 Project Milestones

We will deliver quarterly results on cognitive computing engine, adversarial machine learning, long short-term memory based recurrent neural networks, and reasoning engine.

3.1 Statement of Work

3.1.1 Cyber Attribution

Cyber attribution model consists of three major parts: (1) knowledge representation, (2) security-related information dissemination, and (3) online intrusion detection through sampling hardware, software, and/or operating systems data.

- *Knowledge Representation:* Raw data needs to be labeled with unique features or tagged with meta information about a particular item. Each observation consists of several features (a particular type of provenance information or system parameter). We will focus our study in efficient labeling methods (learn with less labels) and conduct performance evaluation of methodologies.
- *Dataset*: A collection of observations, each containing values for each of the features. The dataset must be representative to guarantee generalization.
- *PROV-O Dataset*: Provenance data that is fit to serve a particular machine learning analytics task i.e. only necessary data will be available based on the operational context and data requirement for a particular data analysis or learning methodology. This comes under security-oriented targeted dissemination of data that guarantees privacy preservation.
- *Online Intrusion Detection:* When the autonomous system continues to function, it should be able to detect any intrusion that has already taken place or in the processes. We will implement the intrusion detection based on fixed sampling of hardware data and a fuzzy matching software.
- *Sampling and Frequent Addresses:* As a part of intrusion detection, we will conduct sampling of memory addresses used by the system for every *n* instructions. This creates a global table with process ids and addresses where the occurrences are counted to sort the list by most to least occurrences. Top occurring addresses are decoded and instructions will be extracted and these extracted instructions will be matched up with knowledge base on malware intruders to classify if a process benign or an intrusion.

The following experiment is planned for proof of concept and further tune our research ideas and approaches.

***Experiment 1**: Sampling-based Online Intrusion Detection*

- *Input*: A hardware dataset consisting of both memory bytes and instructions (opcode) extracted from the frequently used (hotspot) memory spots. Along with this data, we will also use time series provenance data, network sensor data, and system and software performance data. A significant amount of data generated about the system state is discrete in nature.
- *Output Parameters:* Model classifies top processes in the global descendingly sorted list and classifies whether the given instruction sequence is a benign sequence or if it belongs to a malicious process that has intruded IAS. Fuzzy pattern discovery provides information about potential malfunctions, security loopholes, insider attacks, and other failure events in

autonomous systems.

- *Experimental Setup:*
  - Build an application to sample and extract addresses and process ids for every 1,000,000 instructions. The raw data is stored on a physical file.
  - Hotspot identification algorithm will be implemented by counting the frequency of occurrences per address, and top T percent threshold addresses will be used for data extraction.
  - Use fuzzy matching techniques to compare the instruction sequences and memory bytes sequences with intrusion knowledge base.
  - Set up performance parameters such as detection rate (number of detections / memory extractions) between global hotspots methods (per address) and local hotspots method (per processes)

.

3.1.2 Adversarial Machine Learning

Adversaries can insert carefully crafted anomalous data into training set to influence the machine learning model. In order to filter out poisonous data, we will establish taxonomies for data before adding it into the training set and perform outlier detection once the data is in the training set. We will consider the following parameters in taxonomies for adding data items to the training set.

- Defining metrics to quantify trustworthiness of the data (originating from trusted or untrusted system, verified or unverified data format, etc.)
- Defining operating context awareness in terms of data admission i.e. a data item can be of expected type but the time of generation is not relevant.
- Developing models and mechanisms for optimized automated monitoring and reconfiguration of data collection to achieve *maximum* possible *trustworthiness* with *minimum* operational *cost*.
- Developing mechanisms to trim unnecessary data during run time so that training data set cannot be infiltrated when the data set is being fed in to the learning model.
- Developing risk and performance estimation models and optimization algorithms that will be integrated into the reconfiguration process to achieve optimal performance in applying taxonomy.
- Employ supervised and unsupervised learning to isolate the outliers and further run analysis on outliers to make a determination of their validity.

Experiment 2**: **Defending Against Poisoning Attacks

- *Input*: Software data (provenance data), hardware dataset consisting of both memory bytes and instructions (opcode) extracted from the frequently used (hotspot) memory spots, and operating system data (system calls sequence initiated by each process).
- *Output Parameters:* Two output parameters (1) a binary classification of whether the given data item matches the taxonomy described and (2) data outliers—identified by either supervised or unsupervised learning such as clustering.
- *Experimental Setup:*
  - Build a taxonomy model for admissible data points to the training set, which include verification parameters such as origin of the data, trustworthiness of the data source, and validity of the data given the current operational context.

- Develop an application that performs unsupervised learning (SVM/KNN/k-means clustering) to isolate the outliers based on the features extracted from training set.
- Build a program to verify further if the classified outlier data is useful to the model or malicious in nature that can adversely affect the model.
- Set up performance parameters for detection accuracy and detection rate (number of detections / n number of data points in training sets).

3.1.3 Malware / Anomaly Detection in Cognitive Autonomy

To apply right level of AI for meeting mission objectives, we consider the information from slide 9 in presentation of Paul Conoval at IEEE AIKE conference, October 2018 [24] in increasing level of cognitive autonomy and increasing level of cognition ranging from rule sets, semantics, machine learning to deep learning and general AI. For operationalization, we will consider mission requirements, constraints, validation, operational security to assessments and deployment as discussed in slide 10 [24]. A cognitive computing module for autonomous systems that utilizes both deep learning techniques such as long short-term memory units (recurrent neural network) and light-weight machine learning models such as SVM, KNN, and RF. This engine will create models for normal behavior (whitelisting) and malware / anomaly detection (blacklisting) in intelligent autonomous systems.

- *Observation selection:* The single data entity that represents the state of the autonomous system. Each observation consists of several features (a particular type of information). We will focus our study in data from the performance evaluation of the system (e.g. response time, CPU usage, memory usage, etc.).
- *Dataset*: A collection of observations, each containing values for each of the features. The dataset must be representative to guarantee generalization. We will be using hardware data (instructions), software data (provenance and other application parameters), and operating system data (system call sequence executed by each process).
- *Feature Selection:* Any creation of new features based on original or derived datasets. After having a comprehensive set of features we will select the subset that best fits our interest for system behavior modeling.
- *Method selection:* We will explore both supervised and unsupervised methods. With supervised methods a trained engine with labeled data will allow to classify an observation as either benign or malicious. On the other hand, unsupervised methods will allow clustering observations as benign or malicious without previous training.
- *Malware / Anomaly detection:* Depends on the features availability and size of the data, the malware detection module will be using deep learning or light-weight machine learning model.

The following experiment is planned for proof of concept and further tune our research ideas and approaches.

Experiment 3**: **Malware / Anomaly Detection using Light-weight Learning Models

- *Input*: A hardware dataset consisting of both memory bytes and instructions (opcode) extracted from the frequently used (hotspot) memory spots, and operating system data (system calls sequence initiated by each process). Along with this data, we will also use time series data, network sensor data, and system and software performance data. A significant amount of data generated about the system state is discrete in nature.

- *Output Parameters:* A binary classification result of whether the given process is malicious or benign (depends on the training based on features). Determine: is it likely malware? What specific known malware?
- *Experimental Setup:*
  - Build an application to sample and extract addresses and process ids for every 1,000,000 instructions. The raw data is stored on a physical file.
  - Hotspot identification algorithm will be implemented by counting the frequency of occurrences per address, and top T percent threshold addresses will be used for data extraction.
  - Extract features from decoded instructions: instruction features, common crypto features, bigram, unigram, and n-gram features.
  - Perform clustering and classification techniques: SVM, KNN, RF, XGBOOST, and k-means clustering.
  - Set up performance parameters such as accuracy and detection rate (number of detections / memory extractions) between global hotspots methods (per address) and local hotspots method (per processes)

Experiment 4**: **Malware / Anomaly Detection using Deep Learning Model

- *Input*: 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.
- *Experimental Setup:*
  - Implementation of the model: LSTM models and its variant Gated Recurrent Unite 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:
    - Dropouts in the hidden-to-hidden recurrent weights, embedding dropout, randomized-length backpropagation through time (BPTT), activation regularization (AR) and temporal activation regularization (TAR).
    - Cyclical learning rates.
  - How to effectively discriminate between events of low and high probabilities from the output array of the models without manually fixing a threshold?
  - *Evaluation:*
    - 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?

3.1.4 Integration with NGCRC researchers and NGC IRAD Projects

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 annual meeting for NGCRC and NGC TechExpo in May, 2018. We have benefited from the keynote talk by Paul Conoval at IEEE AIKE conference in September, 2018 in California.

- We plan to participate in the DARPA OFFensiveSwarm-Enabled Tactics (OFFSET) program where Northrop Grumman serves as a swarm systems integrator, is tasked with designing, and is developing and deploying a swarm-system, open-based architecture for swarm technologies in both a game-based environment and physical test bed.
- We plan to contribute to CAE, ACBM, ACES, and Cyber Resilient DevOps IRADs. We plan to coordinate with NG scientists and engineers and discuss with Darpa and AFRL for possible CRADS and BAA.
- Three research publications with NG coauthors (Kobes, Conoval, Steiner) were presented in IEEE Cloud and IEEE Cognitive Computing conferences in June 2018 and IEEE Artificial Intelligence and Knowledge Engineering (AIKE) conference in September, 2018. This direction of research and IAS project was originally inspired by Dr. Donald Steiner during Tech Fest in 2017.
- To fulfill the needs of potential customers (agencies such as AFRL, DoD), we will develop experiments and demonstrate the capabilities of Autonomous Systems, working closely with NGCRC researchers like Dr. Lalana Kagal from MIT on privacy preserving provenance ontology structure to increase the trustworthiness of autonomous systems and collaborate with and Chris Clifton at Purdue in one-class SVM models. It will include publishing joint work and exploring opportunities to respond to BAA in DARPA, AFRL, and NSF.

3.2 Milestones and Accomplishments

The following table shows the list of tasks to be accomplished during the project period, broken down in a quarterly basis. We plan to hold weekly meetings in Fall 2018 and Spring 2019 with NG researchers to accomplish the development of demos for Tech Expo 2019.

Task	Q1 (Oct - Dec)	Q2 (Jan - Mar)	Q3 (Apr - Jun)	Q4 (Jul - Sep)
Initial setup of autonomous system experiments with cyber attribution	X			
Implementation of knowledge representation through efficient labeling / tagging learning	X	X		
Integrating provenance data with Provenance Ontology (PROV-O) structure for privacy.	X	X		

Implementing a hardware data (opcode and memory bytes) collection application leveraging kernel drivers		X		
Implement online detection of intrusions through data sampling		X		
Development of data analytics models based on the collected data for reasoning engine		X	X	
Development and analysis of light-weight machine learning models for malware / anomaly detection		X	X	X
Development and analysis of adversarial machine learning model to filter poison data			X	X
Development of deep learning model with LSTM Recurrent Neural Network for malware / anomaly detection		X	X	X
Prototype demonstration at NGC TechFest 2019 (if approved)				X
Integration of developed autonomous framework with smart autonomy IRAD at NG		X	X	X

4 Project Budget Estimate

The project will involve one faculty, two Ph.D. students (one working on Ph.D. dissertations on intelligent autonomous systems and second one who will facilitate experiments and prototypes and demos for 2019 Tech Expo). Budget will consist of salary for the faculty and salary for Ph.D. students. The total budget including fringe benefits, tuition fees, and Purdue University overhead will be $160,000.

Project Budget Estimate		
Category	**Items**	**Cost**
Labor Hours	Total Hours: 1778	81,877.00
Materials and Equipment		5,000.00
Travel		3,000.00
ODCs		20,691.00
Other		49,432.00
Total		**$160,000**

Table 2: Project Budget Estimate

5 References

[1] "Program Solicitation NSF 16 - 608 for Smart and Autonomous Systems (S&AS) ", Retrieved on July 11, 2017. https://www.nsf.gov/pubs/2016/nsf16608/nsf16608.pdf

[2] Wheeler, David A., and Gregory N. Larsen. *Techniques for cyber attack attribution*. No. IDA-P-3792. INSTITUTE FOR DEFENSE ANALYSES ALEXANDRIA VA, 2003.

[3] Rid, Thomas, and Ben Buchanan. "Attributing cyber attacks." *Journal of Strategic Studies* 38, no. 1-2 (2015): 4-37.

[4] Shakarian, Paulo, Gerardo I. Simari, Geoffrey Moores, Simon Parsons, and Marcelo A. Falappa. "An argumentation-based framework to address the attribution problem in cyber-warfare." *arXiv preprint arXiv:1404.6699* (2014).

[5] Altheide, Cory, and Harlan Carvey. *Digital forensics with open source tools*. Elsevier, 2011.

[6] Thonnard, Olivier, Wim Mees, and Marc Dacier. "On a multicriteria clustering approach for attack attribution." *ACM SIGKDD Explorations Newsletter* 12, no. 1 (2010): 11-20.

[7] Kreibich, Christian, and Jon Crowcroft. "Honeycomb: creating intrusion detection signatures using honeypots." *ACM SIGCOMM computer communication review* 34, no. 1 (2004): 51-56.

[8] Espionage, Investigating Cyber. "SHADOWS IN THE CLOUD." (2010).

[9] Huang, Ling, Anthony D. Joseph, Blaine Nelson, Benjamin IP Rubinstein, and J. D. Tygar. "Adversarial machine learning." In *Proceedings of the 4th ACM workshop on Security and artificial intelligence*, pp. 43-58. ACM, 2011.

[10] Wang, Gang, Tianyi Wang, Haitao Zheng, and Ben Y. Zhao. "Man vs. Machine: Practical Adversarial Detection of Malicious Crowdsourcing Workers." In *USENIX Security Symposium*, pp. 239-254. 2014.

[11] Fung, Clement, Chris JM Yoon, and Ivan Beschastnikh. "Mitigating Sybils in Federated Learning Poisoning." *arXiv preprint arXiv:1808.04866* (2018).

[12] Barreno, Marco, Blaine Nelson, Anthony D. Joseph, and J. Doug Tygar. "The security of machine learning." *Machine Learning* 81, no. 2 (2010): 121-148.

[13] Baracaldo, Nathalie, Bryant Chen, Heiko Ludwig, and Jaehoon Amir Safavi. "Mitigating Poisoning Attacks on Machine Learning Models: A Data Provenance Based Approach."

In *Proceedings of the 10th ACM Workshop on Artificial Intelligence and Security*, pp. 103-110. ACM, 2017.

[14] Hopkins, Michael, and Ali Dehghantanha. "Exploit Kits: The production line of the Cybercrime economy?." In *Information Security and Cyber Forensics (InfoSec), 2015 Second International Conference on*, pp. 23-27. IEEE, 2015.

[15] Kharraz, Amin, William Robertson, Davide Balzarotti, Leyla Bilge, and Engin Kirda. "Cutting the gordian knot: A look under the hood of ransomware attacks." In *International Conference on Detection of Intrusions and Malware, and Vulnerability Assessment*, pp. 3-24. Springer, Cham, 2015.

[16] Xie, Liang, Xinwen Zhang, Jean-Pierre Seifert, and Sencun Zhu. "pBMDS: a behavior-based malware detection system for cellphone devices." In *Proceedings of the third ACM conference on Wireless network security*, pp. 37-48. ACM, 2010.

[17] Mani, Ganapathy, Bharat Bhargava, and Jason Kobes. "Scalable Deep Learning Through Fuzzy-based Clustering in Autonomous Systems." In *IEEE International Conference on Artificial Intelligence and Knowledge Engineering (AIKE), pp.* IEEE. 2018. http://www.cs.purdue.edu/homes/bb/aike2.pdf

[18] Mani, Ganapathy, Denis Ulybyshev, Bharat Bhargava, Jason Kobes, and Puneet Goyal. "Autonomous Aggregate Data Analytics in Untrusted Cloud." In *IEEE International Conference on Artificial Intelligence and Knowledge Engineering (AIKE), pp.* IEEE. 2018. http://www.cs.purdue.edu/homes/bb/aike1.pdf

[19] Mani, Ganapathy, Bharat Bhargava, Basavesh Shivakumar, and Jason Kobes. "Incremental learning through graceful degradations in autonomous systems." In *2018 IEEE International Conference on Cognitive Computing (ICCC)*, pp. 25-32. IEEE, 2018. http://www.cs.purdue.edu/homes/bb/iccc1.pdf

[20] Nguyen, Hieu V., and Li Bai. "Cosine similarity metric learning for face verification." In Asian conference on computer vision, pp. 709-720. Springer, Berlin, Heidelberg, 2010.

[21] Prasath, V. B., Haneen Arafat Abu Alfeilat, Omar Lasassmeh, and Ahmad Hassanat. "Distance and Similarity Measures Effect on the Performance of K-Nearest Neighbor Classifier-A Review." arXiv preprint arXiv:1708.04321 (2017).

[22] Bholowalia, Purnima, and Arvind Kumar. "EBK-means: A clustering technique based on elbow method and k-means in WSN." International Journal of Computer Applications 105, no. 9 (2014).

[23] He, Lie, An Bian, and Martin Jaggi. "COLA: Communication-Efficient Decentralized Linear Learning." *arXiv preprint arXiv:1808.04883* (2018).

[24] Conoval, Paul. Integrating Artificial Intelligence Capabilities into Deployable Systems, Keynote talk in IEEE Artificial Intelligence and Knowledge Engineering (AIKE), 2018. https://www.cs.purdue.edu/homes/bb/pc2018.pdf

[25] S. Hochreiter and J. Schmidhuber. "Long short-term memory." Neural computation. 1997.

[26] G. Mani, B. Bhargava, P. Angin, J. Kobes, "Machine Models to Enhance the Science of Cognitive Autonomy" To appear in the proceedings of IEEE Artificial Intelligence and Knowledge Engineering (AIKE), IEEE. 2018. http://www.cs.purdue.edu/homes/bb/aike3.pdf