Technology Final Report

Intelligent Autonomous Systems Based on Data Analytics and Machine Learning

November 18, 2018

Prepared for
The Northrop Grumman Cyber Research Consortium
As part of IS Sector Investment Program

Prepared by
Bharat Bhargava

CERIAS, Purdue University

Table of Contents

List of Figures

List of Tables

1 Executive Summary

Title	Intelligent Autonomous Systems based on Data Analytics and Machine Learning		
Author(s)	Bharat Bhargava		
Project Lead	Bharat Bhargava		
University	Purdue University		
Requested Funding Amount	$150,000		
Period of Performance	September 1, 2017 - August 31, 2018		
Is this an existing Investment Project?	No	**TRL Level of Project**	5
Key Words	Autonomous system, data provenance, reinforcement learning, cognitive autonomy, data analytics, machine learning analytics, ontological reasoning, blockchain, trust		
Key Partners & Vendors			

NGC projects you have collaborated with in the past	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 3: Executive Summary

1.1 Statement of Problem

Systems with smart autonomy should be capable of exhibiting high-level understanding of the system beyond their primary actions and their limitations and capacity. They should predict possible errors, initiate backup plans, and adapt accordingly. They should be able to multitask: collaborating with their human counterparts, communicating, and executing actions in parallel. A smart system is also required to monitor its interactions with the environment, find problems, optimize, reconfigure, and fix those problems autonomously, while improving its operations overtime. A comprehensive IAS should be rich in discovered knowledge on which it can reason with that knowledge at various levels of abstraction using several quantitative and qualitative models: semantic, probabilistic, ontological, symbolic, and commonsense. Hence, an IAS is contingent on its cognizance of its operational boundaries, operating environment, and interactions with clients and other services. An IAS should demonstrate reflexivity implying that it continuously adjusts its behavior and adapts to new unpredictable situations. It should have reasoning where it can introspect about its own reasoning limitations and capacity.

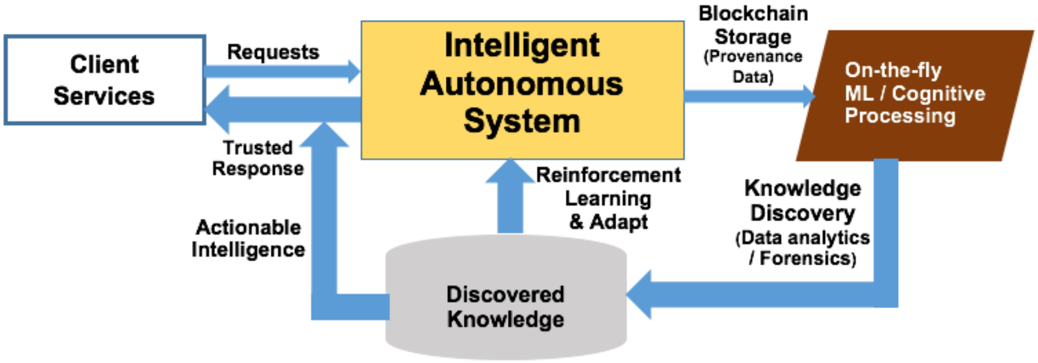
Client Services
Requests
Trusted Response
Intelligent Autonomous System
Blockchain Storage (Provenance Data)
On-the-fly ML / Cognitive Processing
Actionable Intelligence
Reinforcement Learning & Adapt
Knowledge Discovery (Data analytics / Forensics)
Discovered Knowledge

Figure 1. Conceptualization of Comprehensive Intelligent Autonomous Systems (IAS)

These characteristics lead to the following research problems and directions: (a) how to enhance the cognizance of IAS using novel cognitive processing approaches that enable the system to be aware of the underlying operating and client context where the data is being generated, (b) how to conduct distributed processing of streaming data on-the-fly (and in parallel) in order to apply advanced analytics techniques and machine learning models for knowledge discovery, (c) investigating new analytics techniques for finding underlying patterns and anomalies, thus increasing the value of the gathered data, (d) how to facilitate learning from data to improve the adaptability of the IAS, (e) how to innovatively apply blockchain technology in order to provide trust and verifiability to IAS, (f) how to contribute to representation and reasoning approaches based on both qualitative and quantitative models—probabilistic, ontological, semantic, and commonsense—to discover new knowledge, and finally, (g) how to advance science of learning algorithms to enable autonomy in self-optimization, self-healing, self-awareness, and self-protection, and to reason about making decisions under uncertainties.

1.2 Current State of Technology

According to NGC presentation at Kansas State University, an autonomous system should be able to act without the lapses of human judgment or execution inadequacies and provide the same level of concern as a human to a particular task. This is defined as cognitive autonomy [2]. A concept generation system for cognitive robotic entities is implemented by Algorithm of Machine Concept Elicitation (AMCE) [13]. AMCE enables autonomous concept generation based on collective intention of attributes and attributes elicited from formal and informal definitions in dictionaries. In [14], a bio-inspired autonomous robot with spiking neural network (SNN) is built with a capability of implementing the same SNN with five variations through conditional learning techniques: classical conditioning (CC) and operant conditioning with reinforcement or punishment and positive or negative conditioning. A wideband autonomous cognitive radio (WACR) has been designed and implemented for anti-jamming in [15]. The system has the collected data on spectrum acquisition as well as the location of the sweeping jammer. This information and reinforcement learning are used to learn the perfect communication mode to avoid the jammer. Here, the system is self-aware about the current context. We will investigate learning models and analytics to attain cognitive autonomy in IAS. To conduct data analytics on-the-fly and change the analytics techniques automatically, an instrumented sandbox and machine learning classification for mobiles is implemented in [16]. The analysis is conducted, adjusted, and readjusted based on the information of mobile applications submitted by the subscribers. There are well-known knowledge discovery mechanisms that can be applied on raw data to discover patterns. In [17], the authors outline scalable optimization algorithms and architectures encompassing advanced versions of analytics techniques such as principle component analysis (PCA), dictionary learning (DL), and compressive sampling (CS). We will be employing advanced data analytics techniques to discover patterns and anomalies from raw data.

Thomas E. Vice, corporate vice president of NGC, gave a talk at Purdue University about the future of autonomous systems [54]. He outlined the projects on autonomous systems and how Trusted Cognitive Autonomous Systems will be the future. Our project complements the vision of NGC. Through discovered knowledge, an IAS can continuously learn, reason, predict, and adapt to the future events. A lightweight framework for deep reinforcement learning is presented in [18]. The learning algorithm uses the asynchronous gradient descent for optimization of deep neural networks. In this paper [19], the authors introduce an agent that maximizes the reward function by continuous reinforcement learning with an unsupervised auxiliary task. Reinforcement learning is one of the major machine learning methods that is used primarily on automated cyber physical systems such as autonomous vehicles [20-22] and unmanned aerial vehicles (UAVs) [23-25]. Defender-and-attacker game, a game theoretic approach, is employed in general learning models of security as well. When the attacker information is very limited, and attacker persistently makes her moves (in the game) to affect the system, the defender needs to constantly adapt to the attackers' novel strategies. So, the defender constantly reinforces her beliefs based on the attacker moves and creates a robust defense strategy for future attacks [26]. We will use reinforcement learning algorithms to enhance automated decision making and dynamic reconfiguration capabilities to increase the reflexivity of the system.

Data provenance is used in forensics and security for providing robust support for the underlying systems, sometimes autonomous, through valuable meta-information about the system and its interactions [7]. Data provenance has been modeled for and used in autonomous systems in service-oriented architecture [3] [4] [12] and autonomous information systems [5] [6]. Further investigation is needed to model the use of provenance in enabling autonomy. The Database-Aware Provenance (DAP) architecture [8] provides a workflow that detects the addition of any new autonomous unit of work for fielding any service request and tracks its activities to extract the relevant operational semantics. Provenance data is also used to enhance trust and security in autonomous systems. Trust in information flow can be maintained and verified by provenance data [9], where trust of autonomous entities can be quantified by data provenance and internal values of the data items. Piercing perimeter defenses in autonomous systems can be resolved by provenance-aware applications and architectures [10]. To enable autonomy, systems must be able to reason about and represent provenance data at multiple levels of abstraction. Quantitative and qualitative reasoning can enable semantic knowledge discovery and predictable events. Semantic ontologies are widely used in autonomous cyber-physical systems (CPS) [27]. Ontology-like reasoning over several intelligence representations of new entities can enable the autonomous system to reason about unexpected entities present in their environment [28] [29]. A recent study [11] shows that trust and immutability are provided through provenance on blockchain technology, where smart contracts can be created. This increases trust, provides consensus, and reduces the need for third party intervention: creating a decentralized autonomous setting. *Provchain*—a blockchain-based data provenance architecture is proposed in [30] to provide enhanced availability and privacy in cloud environments. Blockchain provides integrity to provenance data through its immutable property [31]. Our research will utilize data provenance with blockchain technology for modeling autonomy in smart systems.

1.3 Proposed Solution

We developed a novel approach that performs *on-the-fly analytics* on data streams gathered from sensors/monitors of autonomous systems to *discover valuable knowledge*, *learn* from the system's interactions with the runtime environment and *adapt* its actions in a way to *maximize its benefits* over time for enhanced *self-awareness* and *auto-configuration capability,* and track the provenance of the data gathered/generated by the system to provide increased trust in the actions of the system. By integrating components for streaming data analytics, cognitive computing with deep reinforcement learning and knowledge discovery through unsupervised/supervised learning on streamed data, the proposed model aims to provide a unified architecture for smart autonomy, applicable to various systems that NGC is developing. The overall architecture of the model is demonstrated in Figure 1.

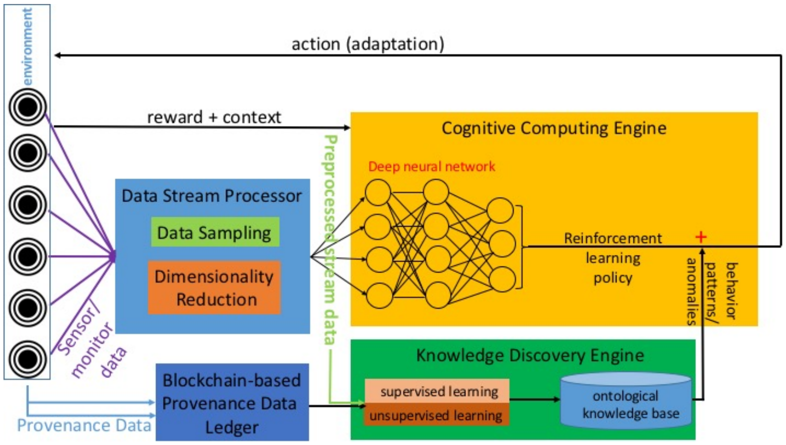

Figure 2. Intelligent Autonomous System (IAS) Architecture

General characteristics of the proposed solution are as follows:

- Data obtained through the sensors/monitors of the autonomous system are fed into data stream processor, which contains modules for pre-processing of the data to prepare it for analytics to derive valuable knowledge. The dimensionality of the data is reduced and data is sampled to allow for real-time processing.
- The pre-processed data is fed into the *data analytics* module (knowledge discovery engine), which applies unsupervised machine learning algorithms to detect deviations from the normal behavior of the system. The gathered data is used to build a model of the system's environment and actions by storing it in a knowledge discovery module, which is consulted repeatedly through the lifetime of the system, acting like the memory of a human-being to decide which actions to perform under different contexts.
- The provenance of the data gathered by the sensors/monitors of the system is logged in an immutable private ledger based on the blockchain technology. This provides verifiability of the data which is used in the knowledge discovery process. It helps in building and measuring the level of trust of an IAS.
- The data pre-processed by the data stream processor and the provenance data are fed into the cognitive computing engine, forming the observations for reinforcement learning in the system, so that the system gains self-awareness over time through a reward-based process. The reward can be based on the type of the system; for a UAV, it could be based on the quality of image processing, while for a missile defense system it could be accuracy and time needed to mitigate an attack. The reinforcement learning process utilizes deep neural networks to build a model of the big data gathered, rather than utilize a trial-and-error learning approach. This enables the system to gain increased self-awareness in time, and gain auto-configuration/self-healing abilities. The system acts upon its environment based on the outcomes of the reinforcement learning and knowledge discovery processes, keeping it in an action-value loop as long as it functions.

We developed a comprehensive approach to enable autonomy in smart systems by enhancing the following fundamental properties of IAS: *cognitive*—mindfulness of the current state of the system (self-awareness), *reflexivity*—ability of the system to monitor and respond to known and unknown scenarios, and adjust accordingly with limited or no human intervention (self-optimization and –healing), *knowledge discovery*—ability to find new underlying patterns and anomalies in system interactions through advanced data analytics techniques, *predictive*—learn and reason from the discovered knowledge, anticipate possible future events, and recalibrate corresponding actions, and finally *trust*—ability to provide verification and consensus for the clients as well as for the system (self-protection).

The quality and trustworthiness of data in an IAS is of prime importance for achieving the abovementioned goals. We utilize the following data storage/sharing technologies and data sources when modeling the system and conducting experiments.

NGC-WaxedPrune prototype system: Data are stored in the Active Bundle [39] [40] [41], which is a self–protected structure that contains encrypted data items, access control policies, and a policy enforcement engine. It assists in privacy preserving data dissemination. The design of this system received the first rank (voted by corporate partners) at the 2015 annual symposium competition of the Purdue CERIAS center. This system can be used to deal with all data generated and monitored in IAS and its interactions with outside entities.

Provenance data: In the Active Bundle scheme, provenance metadata is generated, attached to an Active Bundle and sent to a central monitor each time a service accesses data. Provenance metadata contains information on when data was accessed, where, by whom, as well as several execution environment parameters, such as OS version, Java version, libraries, CPU model at data recipient's side. Using provenance as a basis for decision making largely depends upon the trustworthiness of provenance [36]. We can deploy Active Bundle as used in WaxedPrune and blockchain storage for provenance data [33] in order to provide trust and integrity to IAS.

Monitoring Data: Log files are one of the most numerous data collection methods to record activities, user-and-system generated errors, notifications, transactions, interaction with third parties, etc., [31]. Employing advanced data analytics techniques can provide us with rich knowledge of patterns and anomalies. We intend to use the log files of the WaxedPrune system. Analytics on numerical data from sensors/monitors of autonomous systems can be used to verify the convergence of reinforcement algorithms [34]. We will use publically available data to test the proof of concept in terms of accuracy and convergence of machine learning techniques, reinforcement algorithms, and reasoning models for IAS.

The individual components of the proposed smart autonomy model are described in the subsections below.

1.3.1 Cognitive Autonomy:

An IAS in a distributed environment should be aware of its three major system, software, and interaction layers: (1) its own state of the system and software as well as operational parameters, (2) state of its neighboring systems, and (3) client or third-party services and their interactions with the system.

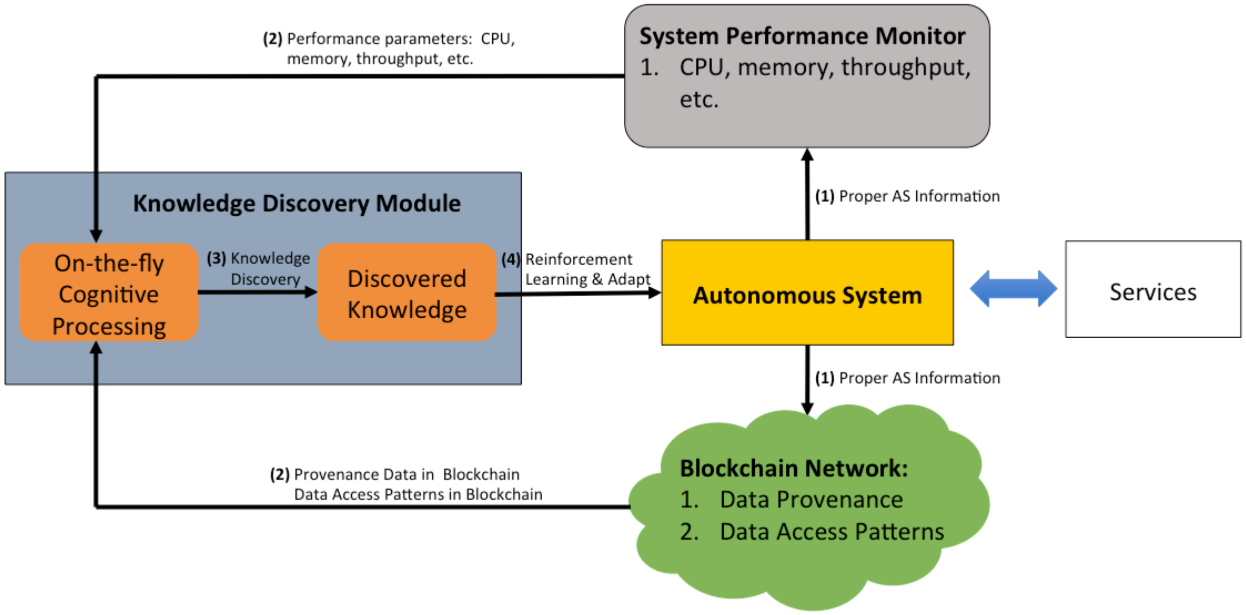
(2) Performance parameters: CPU, memory, throughput, etc.
System Performance Monitor
1. CPU, memory, throughput, etc.
Knowledge Discovery Module
On-the-fly Cognitive Processing
(3) Knowledge Discovery
Discovered Knowledge
(4) Reinforcement Learning & Adapt
(1) Proper AS Information
Autonomous System
Services
(1) Proper AS Information
(2) Provenance Data in Blockchain
Data Access Patterns in Blockchain
Blockchain Network:
1. Data Provenance
2. Data Access Patterns

Figure 3. Cognitive Computing Process for Autonomous Systems

We develop a novel approach that uses Artificial Intelligence (AI) techniques to monitor and learn the state of autonomous systems to automatically adapt to meet mission objectives with no human intervention. The main idea of the this research is actively monitoring the system to provide those results as inputs to decision-making machine learning algorithms that determine the new configuration of the system based on the resulting outputs. This research has a focus on the analysis of two types of data: (1) performance parameters, such as response time, CPU usage, memory usage, etc., and sensor data peculiar to the system and (2) data access patterns stored as data provenance in blockchain for misbehavior detection. By integrating system performance and either benign or malicious behavior data in making decisions from past experience the proposed model aims to provide a unified and comprehensive architecture for self-healing intelligent autonomous systems.

Deep reinforcement learning [18] is utilized as the primary machine learning technique for cognitive computing in the system to achieve adaptability to different environments, learn from previous vulnerabilities and maximize the security. As stated by Mnih et al. [39], reinforcement learning provides a way to model human behavior in terms of optimizing control of an environment of the agent, through an action-value feedback loop. Reinforcement learning is a difficult task due to the complexity of representing an environment with high-dimensional sensory data. Nevertheless, recent advancements in deep learning allow for building more abstract representations of data from sensors through utilizing multiple levels of nodes, which can be used as the model to optimize the action-value function in the reinforcement learning process. Deep reinforcement learning has recently been successfully applied for tasks like playing Atari games [18].

The deep neural network (DNN) component of the cognitive computing engine are used to approximate the optimal action-value function for the reinforcement learning model. Deep neural networks also solve the problems of adversarial search and Markov decision processes. The Markov property is nothing but the probability of the current event (E_i) depending on the probability of the previous event (E_{i-1}). With DNNs, we can store and build more memory in the previous state. Through this increased memory, we build effective Higher-order Markov models, which recollect more data history, enhancing more predictive capability of the system. We can represent the Markov decision process as follows: in the nth Markov model,

$$\Pr\,(E_i \mid E_{i-1}, E_{i-2}, \ldots, E_1) = \Pr(E_i \mid E_{i-1}, \ldots, E_{i-n})$$

We will employ customized *higher-order Markov Decision Processes (MDP)* to create novel *reinforcement algorithms*. For example, consider a smart system executing functionalities in a cloud environment. In the Markov model, there are states before (past, present, and future states) but currently the future states of the system are not only affected by the past state but also affected by the current actions of the client services and the system. There will be a reward function for the autonomous system, and in the transaction, the system must maximize the rewards. Given time (t), actions (A_t), rewards (R_t), and states (S_t), a reinforcement learning model is represented below,

$$\begin{array}{llllllllll} R_{t+1} & & & R_{t+2} & & & R_{t+3} & & & \\ S_t & \rightarrow\rightarrow\rightarrow & S_{t+1} & & \rightarrow\rightarrow\rightarrow & S_{t+2} & & \rightarrow\rightarrow\rightarrow & S_{t+3} & \ldots \\ A_t & & & & A_{t+1} & & & A_{t+2} & & \end{array}$$

Each state is combined with the actions and maximized reward function, so the system learns which actions to perform to gain more rewards and which actions to reduce the loss. The cognitive computing engine in the proposed research takes as input the data preprocessed by the data stream processor as well as provenance data, which represent the state/observations of the autonomous system for reinforcement learning. The task of the engine is to enable the system to make the best decision for the next action given the context of interaction, the current states of the various system parameters and the knowledge discovered through performing on-the-fly analytics on the streamed data. The overall goal is to select actions in a way to maximize the cumulative QoS parameters that include security and trust, performance, real-time response, and degradation. We will deploy NGCRC funded research on active monitoring tools for measurements of the performance and behavior of services, ideas from MTD [50] for switching replicas and will incorporate new tools for both supervised and unsupervised learning to allow dynamic reconfiguration under various unknown environments, context, and situations.

1.3.2 Knowledge Discovery: The knowledge discovery component of an IAS employs methodologies from pattern recognition, machine learning, and statistics to extract knowledge from raw, and sometimes unknown data. Knowledge discovery is an important element in supporting cognitive autonomy since new knowledge discovered can trigger changes to the smart system to adapt to the new parameters, thus enabling autonomy. Discovered knowledge constitutes the representation of unknown data, its form, and its degree of certainty. The generic process of knowledge discovery is shown in Figure 4 below.

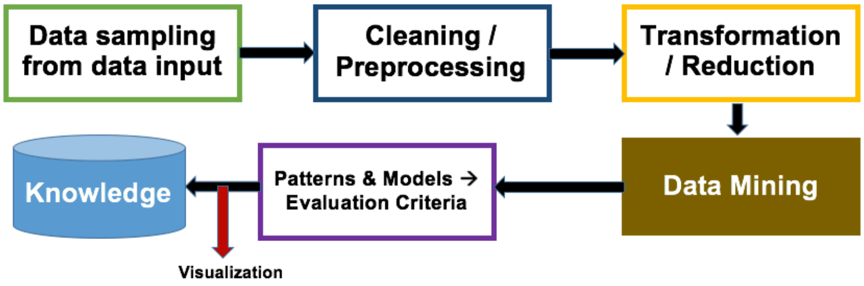
Data sampling from data input
Cleaning / Preprocessing
Transformation / Reduction
Data Mining
Patterns & Models → Evaluation Criteria
Knowledge
Visualization

Figure 4. Knowledge Discovery in Autonomous Systems

Knowledge discovery on large data, in particular streaming data, needs efficient data processing. Distributed data processing on streaming data becomes a necessity for faster classification and storage of data [52]. We will introduce a parallel processing of data items that can classify and categorize the streaming data considerably fast. The classification and clustering techniques must be capable of on-the-fly processing of data streams: distributed data processing can accommodate simultaneous processing of sequential/parallel data streams: the key idea behind the parallel processing is to host distributed data processing units (DDPU) that can (a) read (R) to load the data, (b) Analyze (A) to process and classify the data, and (c) toggle (T) to shift to/from read or analyze. For example,

DDPU 1: R item 1 T→ R item 2 T→ A item 2

DDPU 2:	A item 1	T→	R item 3	T→	A item 4
DDPU 3:	R item 4	T→	…		
	Cycle 1		Cycle 2		Cycle 3

The representation above shows a fundamental distributed data processing technique—*RAT*—to processes data on-the-fly, which is scalable to process Big Data streams. Depending on the priority and availability of the data items each processing unit prioritizes the RAT operation for each data item. In this way, instead of relying on static rules and heuristics to determine prioritization of data processing, we can compute the value of the data on-the-fly based on data's quantitative/qualitative system metrics such as sensitivity, dependence, and importance of the data, and process the data items accordingly. This distributed processing of data streams will contribute to the Distributed Data Processing IRAD of NGC.

The processed data contains both categorized (easy to label) such as data origin, time of creating, and modification, etc., and uncategorized data such as error logs (text). Hence, we will employ both customized—combination of *multi-level decision trees* and *Bayesian probabilistic methods—classification* and *regression* algorithms [53], and advanced *clustering* techniques to achieve high dimensionality and to label the data and prepare it for analysis. We will be using *Bayesian statistics* to estimate the reliability of the autonomous system and quantify the unknown due to lack of data (missing data). Bayes' theorem states that, given two data items D_1 and D_2,

$$\Pr(D_1 \mid D_2) \quad = \quad [\Pr(D_2 \mid D_1) / \Pr(D_2)] * \Pr(D_1)$$

The reliability of the autonomous system is measured using Bayesian statistical methods with conditional probability and prior distribution of the autonomous system's states. New knowledge can be discovered through reliability analysis of autonomous system, which will contribute to the self-awareness of the system, enabling smart autonomy. Our Bayesian statistics approach will contribute to the Reliability Analysis Data System (RADS) IRAD of NGC.

Quantitative and qualitative reasoning enable semantic knowledge discovery and predictable events. Semantic ontologies are widely used in autonomous cyber-physical systems. We apply ontologies to generate semantic reasoning over the provenance data. For example, semantic ontology reasoning is used to extract attributes of provider-client interaction such as: platform, data requested, update, and access. Applying semantic reasoning models to the log files of provenance data helps the system discover new knowledge about the client. This is stored and used to make decision and contribute to autonomy.

Of particular interest to the knowledge discovery process in the proposed system are the following methods that we will investigate and integrate into the knowledge discovery engine:

1. *Association Rule Mining:* Association rule mining discovers patterns of the form "if X then Y", where X and Y are item sets. This allows us to find frequent patterns of co-occurrence in large datasets. Typical algorithms for association rule mining include the *Apriori algorithm, sampling algorithm, frequent pattern tree* and *partition algorithm.* For IAS, we will utilize the mentioned association rule mining algorithms to discover system events that co-occur frequently under normal and anomalous circumstances (e.g. CPU and memory usage spiking up together). This will allow the system to have increased awareness of what environment and system conditions to expect when a certain event occurs and adapt itself

accordingly.

2. *Clustering:* Clustering allows us to partition data without having a training sample, which is useful in situations where the system has just started functioning and we need to discover groups of events/data similar to each other in terms of certain parameters, representing different states of the system. We will employ *k-means clustering*, a typical algorithm to cluster multi-dimensional data *D* consisting of *m* records $r_1 \ldots r_m$ into *k* clusters C_i with centroids m_i using the squared error criterion for optimization, such that each record is assigned to the cluster with the minimum distance to the centroid of that cluster. The error is measured as:

$$\sum_{i=1}^{k} \sum_{r_j \in C_i} Distance(r_j, m_i)^2$$

   Here the most effective distance function can depend on the nature of the data, therefore we will experiment with multiple distance functions. Finding clusters of IAS data along various dimensions will allow for detection of anomalies when incoming data does not belong to any of the previously built clusters. This is also useful for discovering cases like zero-day attacks, which have no known attack signature through detecting deviations from the normal behavior of the system.

3. *Sequential/Temporal Pattern Mining:* Sequential/temporal pattern mining discovers patterns in a dataset that occur frequently in a particular sequence. The gold standard for time series analysis is Hidden Markov Models (HMM), therefore we will utilize HMM to build a representation of IAS behavior through observation of the system states and state transitions over time.

   Based on HMM, the system can be in one of the N possible states $\{S_1, S_2, \ldots, S_N\}$, and undergoes a transition from one state to another at particular times. The state transition probabilities of the system depends on the immediate past, i.e.

$$P(q_t = S_j \mid q_{t-1} = S_i, q_{t-2} = S_k \ldots) = P(q_t = S_j \mid q_{t-1} = S_i)$$

   Additionally, the observations (data gathered through sensors/monitors) are a probabilistic function of each state, i.e.

$$P\ (o_t = v_k \mid q_t = S_j)$$

   where o_t is the data observed at time *t* and v_k is a distinct observation in the set of possible observations for the system. Using HMM, we will build a probabilistic model of the system from a sequential set of observations/data, which best explains the behavior of the system in terms of transitioning between different states and the data resulting from the transitions. For example, a low CPU usage *observation* can be associated with a *malfunctioning module state* with high probability, while an extremely high CPU usage observation can be associated with a *system under attack state*. Based on the knowledge discovered over time with HMM, the IAS is able to predict current and next states more accurately and take adaptability actions accordingly. Critical node analysis in higher order Markov models can lead to identifying critical steps in complex attack strategies of adversaries, reducing resource usage for target analysis. Once the pattern is discovered, the systems can reinforce its understanding and adapt to the new set up.

In addition to the abovementioned techniques, various models for detection of outliers in different types of data have been devised by the machine learning community. While supervised and unsupervised learning models have been applied with success to a variety of domains, robust models for detecting anomalies and failures in IAS operation are still lacking. The main shortcoming of supervised anomaly detection models is that they require a large amount of training data and can only provide accurate results on anomalies that were previously observed in the system. This makes such models unable to capture threats/anomalies that are completely new, which is essential in an environment of ever-growing security vulnerabilities and attacks. A significant advantage of unsupervised models is that the training data required is gathered from the behavior of services operating under normal conditions (possibly in an isolated environment/private cloud); i.e. no attack data is required to train these models. We will consider the advantages and disadvantages of existing models as listed in Table 2 and focus on the development of techniques that are both accurate and have low runtime overhead, possibly using an ensemble of models from the literature.

Method	**Advantages**	**Disadvantages**
K-means Clustering	Low complexity	Sensitive to noisy data
EM Meta Algorithm	Adaptable to different distributions	Converges slowly in some cases
One-Class Support Vector Machine (SVM)	Can handle very high-dimensional data, usually has high accuracy	High memory and CPU, needs positive and negative examples
Unsupervised Neural Network	Has learning capability	Long processing for big networks
Self-Organizing Map	High dimensionality reduction	Time consuming
Hidden Markov Models (HMM)	Representative of the time-based relations and states of services	Have scalability issues

Table 4: Machine learning techniques for outlier/anomaly detection

1.3.3 Reflexivity of the system: The goals of IAS in the proposed approach are (1) replacing anomalous/underperforming modules with reliable versions or adapting to a new mechanism to avoid anomalies, (2) reconfiguring system parameters to respond to anomalous system behavior, (3) swiftly self-adapting to changes in context, (4) enforcing proactive and reactive response policies to achieve performance and security goals, and (5) achieving continuous availability even under attacks and failures.

Providing adaptability in order to achieve increased autonomy in IAS relies on two main elements:

1. *Being cognitive and determining action:* Monitoring of systems is of utmost importance in achieving high self-awareness, as systems in environments with highly dynamic contexts may exhibit frequent changes in many QoS parameters. We measure the assurance level, (integrity/accuracy/trust) of the system from the performance parameters such as response time, throughput, packet loss, delays, consistency, acceptance test success, etc. Compliance with all the requirements of IAS is hard to achieve in such dynamic environments, making monitoring a must for accurate decision-making. The tasks involved in effective monitoring and analysis of the obtained data include the following: (a) identification of QoS metrics, such as response time, CPU usage, memory usage, etc., to determine the

performance and behavior of IAS; (b) development of models for identifying deviations from performance (e.g., achieving the total response time below a specific threshold) and security goals (e.g., having trust levels above a certain threshold).

2. *Autonomous system reconfiguration based on changes in context:* Changes in the context of IAS can affect system behavior, requiring autonomous reconfiguration. While changes in user context can result in updated priorities such as trading accuracy for lower response time in an emergency, changes in system context can result in failures requiring the restart of a component of the IAS. Dynamic reconfiguration of system modules based on the updated constraints and contexts enables success of mission objectives.

Adaptability allows dynamic configuration of software and execution to meet the changing demands of autonomous systems for performance, reliability, security, and resource utilization. Adaptable systems provide graceful degradation and can respond to the timing, duration, type, extent, severity of failures and attacks. Adaptation must satisfy the consistency and integrity constraints. The granularity of formally defined classes of algorithms will determine the overhead and benefits of adaptation. Experiments in adaptability allow systems to identify conditions for satisfying the Quality of Service (QoS) requirements of mission objectives and provide guidelines for reconfiguring algorithms, protocols, sites and associated servers, communication software and routers, and others components. We have explored many ideas about how to create new replicas and determine when to execute the replacement of nodes. One of them is based on graceful degradation. The main idea is having primary and alternate modules and using an acceptance test to validate their operation. Initially a primary module is used and constantly tested. In case of failure there are two alternatives: (1) weaken the acceptance test or (2) replace the primary module with the alternate/replica that can pass the acceptance test. Figure 5 illustrates the concept. In the case that an alternate module replaces the primary module of the IAS not able to pass an acceptance test, the composition of a process in the IAS can change as shown in the lower part of the figure (Note that here the system has two module alternatives for a process *A*, which invokes a process *B* with three module alternatives, that further invokes process *M* having three module alternatives chosen based on the acceptance test process in the upper part of the figure).

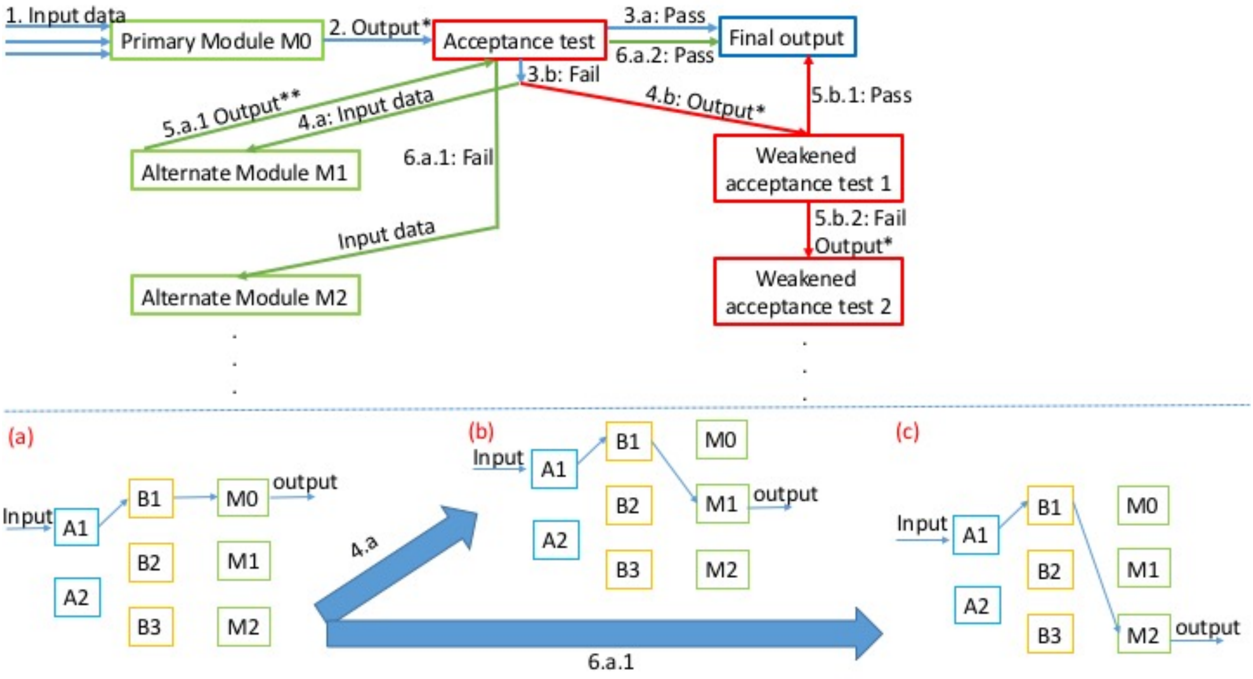

Figure 5. Dynamic Adaptation based on Recovery Block Scheme

Adaptable autonomous systems should be able change their system configuration to guarantee mission critical operations at the cost of sacrificing performance. Because some services may continue their effort to compromise these systems there exists a need for more adaptable solutions to protect systems. Our proposed Moving Target Defense (MTD)-type [50] is a defensive strategy that aims to reduce the need to continuously fight against attacks by decreasing the gain-loss balance perception of attackers. The framework narrows the exposure window of a node (module) to such attacks, which increases the cost of attacks on a system and lowers the likelihood of success and the perceived benefit of compromising it. The achieved reduction in the vulnerability window makes this strategy optimum for adaptable autonomous systems.

The proposed framework introduces reflexivity and adaptability to systems. Reflexivity has two main components: (1) continuing operation and (2) adapting to counter anomalies. The MTD-style approach takes into consideration these components since it transforms systems to be able to adapt and self-heal when ongoing anomalies are detected, which guarantees operation continuity. The initial target of the framework is to prevent successful compromises by establishing short lifespans for nodes/services to reduce the probability of attackers' taking over control. In case an attack occurs within the lifespan of a node, the proactive monitoring system triggers a reincarnation of the node.

1.3.4 Trust in Autonomous Systems: Self-protection (automatic identification and protection from security threats) and self-healing (automatic fault discovery and correction) are important properties of an IAS [43]. We propose an approach for *data provenance* with *blockchain-based* mechanisms to build trustworthiness of the data and ensure identities of network participants. Integrity of data will be guaranteed by blockchain technology. Data can be used for threat detection. Optimized access for transaction validation procedure allows to reduce number of blocks in the blockchain. There is one Merkle tree per Active Bundle and it gets updated with the hash of the data each time a transaction occurs, i.e. either data is read from Active Bundle or data inside Active Bundle gets updated by an authorized service. Provenance record contains information on what data type has been accessed / updated, by whom (by which service), when and who sent the Active Bundle to the service.

Challenges of blockchain technology deployment

1. *Performance:* Blockchain is replicated to all the network participants and this imposes a performance overhead. This was discussed with peter Meloy of NGC-UK.
2. *Access Control (Read):* In case of access revocation or subject's role change, access to data must be revoked immediately within an information system when authorization is no longer valid. However, revoked access to data on a blockchain can be bypassed in the following ways: (1a) by replaying old blocks against an empty blockchain and stopping before the revocation block is appended; (1b) An attacker holding a copy of a blockchain could use a modified client to just ignore the revocation block. Even if read access to local blockchain requires an off-chain token handshake with a centralized authority for authorization; then that token would continue to work forever in the future. The requirement to revoke previously granted access can be bypassed by rolling the local clock back and restoring unauthorized access to blockchain data.

We discussed and learnt many blockchain ideas from *Steve Seaberg (NGC)*. We plan to collaborate with *Peter Meloy, Steve Seaberg,* and *Vladimiro Sassone (University of Southampton, UK)* to work on blockchain – based methodology for provenance data storage and verification.

1.4 Technical Activities, Progress, Findings and Accomplishments

The project's main technical activities consisted of the following accomplishments through methodologies, algorithms, and procedures [55]-[60]:

1. Incremental learning through graceful degradations in autonomous systems using combinatorial balanced block designs.
2. Machine learning models to enhance cognitive autonomy
3. Scalable learning through clustering enabled by perfect error correcting codes.
4. Autonomous privacy-preserving aggregate analysis in untrusted cloud.

1.4.1 Incremental Learning Through Graceful Degradations in Autonomous Systems: Intelligent Autonomous Systems (IAS) are highly cognitive, reflexive, multitasking, trustworthy (secure and ethical), and rich in knowledge discovery. IAS are deployed in dynamic environments and connected with numerous devices of different types and receive large sets of diverse data. They receive new types of raw data that was not present in either training or testing data sets thus they are unknown to the learning models. In a dynamic environment, these unknown data objects cannot be ignored as anomalies. Hence the learning models should provide incremental guarantees to IAS for learning and adapting in the presence of unknown data. The model should support progressive enhancements when the environment behaves as expected or graceful degradations when it does not. In the case of graceful degradations, there are two alternatives for IAS: (1) weaken the acceptance test of data object (operating at a lower capacity) or (2) replace primary system with a replica or an alternate system that can pass the acceptance test. In this paper, we provide a combinatorial design-MACROF configuration-built with balanced incomplete block design to support graceful degradations in IAS and aid them to adapt in dynamic environments. The architecture provides stable and robust degradations in unpredictable operating environments with limited number of replicas. Since the replicas receive frequent updates from primary systems, they can take over primary system's functionality immediately after an adverse event. We proposed a Bayesian learning model to dynamically change the frequency of updates.

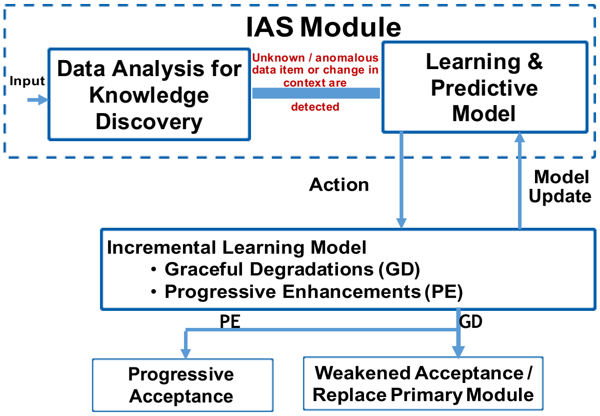
IAS Module
Input
Data Analysis for Knowledge Discovery
Unknown / anomalous data item or change in context are detected
Learning & Predictive Model
Action
Model Update
Incremental Learning Model
• Graceful Degradations (GD)
• Progressive Enhancements (PE)
PE
GD
Progressive Acceptance
Weakened Acceptance / Replace Primary Module

Figure 6. Reflexivity workflow with incremental learning in Intelligent Autonomous Systems (IAS)

Replica replacement by Combinatorial Balanced-blocks:

N systems (S1…S7) are split into M subset blocks (DAB1…DAB7) of size R (3: S1, S5, S7). Each system appears in C blocks (3 out of M). Each system pair appears in Δ blocks (only 1). We implemented (N, M, R, C, Δ) = (7, 7, 3, 3, 1). Each distributed block contains a subset of systems and their replicas that are mathematically distributed and connected, providing balanced resource usage. The replicas periodically receive updates from their primary modules. Update interval is set based on Bayesian inference. Replicas can be used to perform other tasks in parallel while primary module is functioning properly.

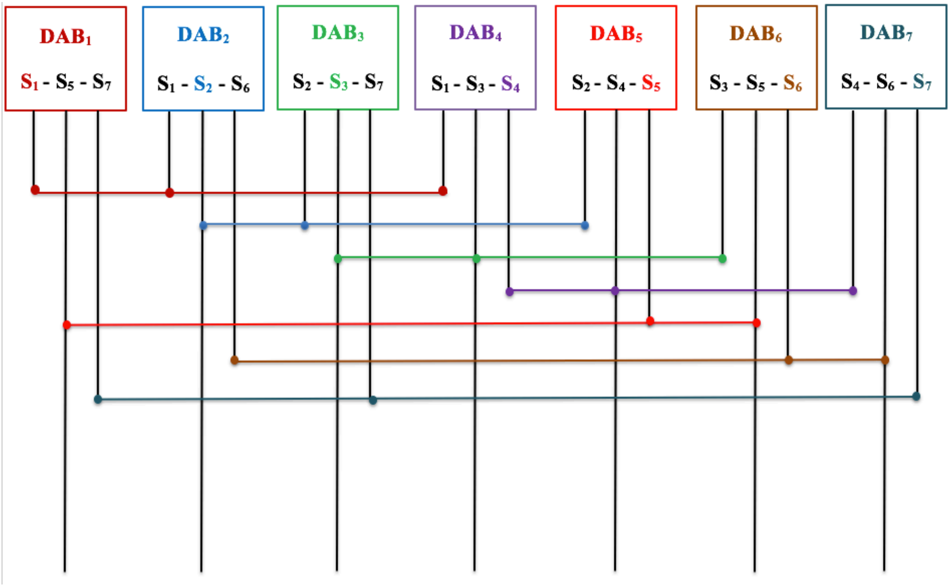

Figure 7. (7, 7, 3, 3, 1) configuration (DAB – Distributed Autonomous Block)

Experimental results:

We set *F* to clock tick of the underlying operating system. The simulator is built for extracting parallelism with sequential processing with replicated data storage elements such as *MACROF* configuration. It is equipped with storage elements (such as registers) to hold data objects. A sequential processing module is also embedded into the simulator to run similar processes that will be run by *MACROF* configuration. These storage elements act as independent autonomous modules. We measure the performance of the graceful degradation structure through number of updates it requires to complete a specific process and update the replicated systems. These number of updates are compared with a sequential processing module without the combinatorial structure. A compiler loads specified instructions for each program into both sequential processing and combinatorial processing structure. When a particular process is being run in the system, *MACROF* design enables the local elements to interact simultaneously without waiting for data from primary modules. This (a) avoids data dependence and (b) increases fault-tolerance—if data is corrupted in one storage element, its replica can replace it. Each storage element is updated with new values when the process progresses. These processes are written in assembly instructions and loaded into the simulator: sum (P1), binary search (P2), copying data (P3), print (P4), double copy (P5), moving data (P6), Fibonacci creation (P7), Fibonacci search (P8), and scalar product (P9). To complete a particular process, depends on the complexity of the processes, the distributed combinatorial *MACROF* model performs well with considerable speed, incurring a smaller number of updates than the sequential processing with replications. One of the main overheads in *MACROF* design is the replication cost when the system increases in scalability. We hypothesize that when the design is increased in its number of DABs and systems, the updates for the replicated systems can be reduce considerably. By keeping track of the failure rate of primary modules in a given time interval, the updates to replicas can be reduced. For example, if there are 700 DBAs with 300 replicas and the failure rate is 10 per day, then only 10-20 replicas may need updating where as other replicas can be updated in batches later on. The replicas that are not in use can be used for other purposes. The increase in number of interconnections may increase the complexity during the design phase but it is relatively easier to keep track once the construction is complete. Table II shows the overhead of updates incurred by non-*MACROF* processing for each process.

Process Type	Process Name	Speed Up Due to Combinatorial Replica Scheme
P1	FIBSEARCH	1.3
P2	DOUBLE MULT	1.4
P3	FIBB	1.5
P4	SEARCH	1.8
P5	COPY	1.8
P6	SCALAR	2
P7	SUM	2.1
P8	PRINT	3
P9	MOVEMENT	3.1

Table 3: Measurement of Various Process completion Speeds

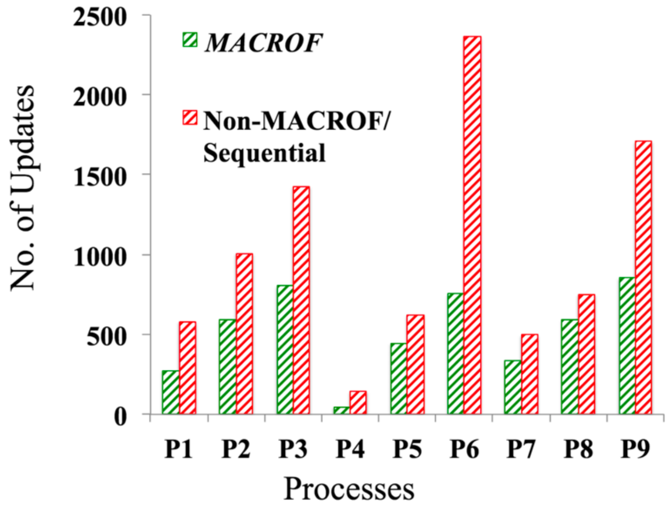

Figure 8. Clock ticks required for various process completions

1.4.2 Machine Learning Models to Enhance Cognitive Autonomy: Examples of IAS include software systems that are capable of automatic reconfiguration, autonomous vehicles, network of sensors with reconfigurable sensors platforms, and an unmanned aerial vehicle (UAV) respecting privacy by deciding to turn off its camera when pointing inside a private residence. Research contributes to build systems that can monitor their environment and interactions, learn their capability as well limitations, and adapt to meet the mission objectives with limited or no human intervention. The systems are fail-safe and allows for graceful degradations while continuing to meet the mission objectives. In this paper, we propose new methodologies and workflows, and develop approaches that can advance the science of autonomy in smart systems through enhancements in real- time control, auto-reconfigurability, monitoring, adaptability, and trust.

1.4.3 Scalable Deep Learning Through Fuzzy-based Clustering in Autonomous Systems: Autonomous cyber systems continuously receive large streams of diverse data from numerous entities operating and interacting in their environment. It is imperative that the learning models in autonomous systems to scale up to process the new and unknown data items. Scalable learning is a method to achieve maximum classification without rejecting any unknown data item that were not present in the training or testing datasets as anomalies. In this paper, we present Bitwise Fuzzy-based Clustering (BFC) technique through error-correcting codes to address the problem. Through BFC, we can approximate the classes of multidimensional features of data items by reversing standard forward error-correction coding. Approximating classes problems generally arise in autonomous systems that are processing fuzzily cataloged data items. These data items can be classified by applying binary vectors to their corresponding features (1: feature is present or 0: feature is absent) to obtain message words. These codewords are used as cluster centers. In BFC technique, binary vectors of 23 bits are mapped into codewords (labels or indices) of 12 bits. Two different 23-bit binary vectors with the Hamming distance of 2 will have a few common labels. This setting enables the clustering of neighboring 23-bit binary vectors with at most 2-bit variation (mismatch) from a given input. BFC technique has 223 codeword space, which makes it ideal for scalability in clustering of millions of categories and their associated features. With reasonable redundancy, the clustering can be accomplished in O (N) time.

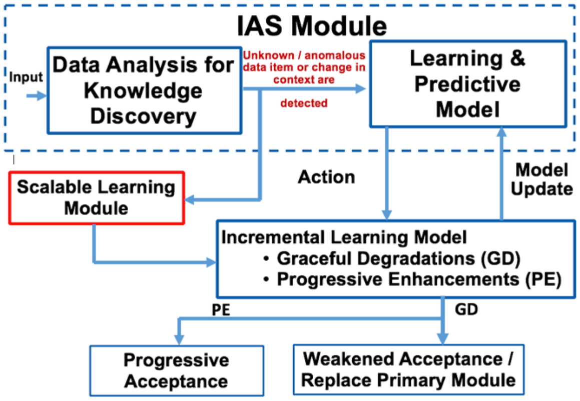
IAS Module
Input
Data Analysis for Knowledge Discovery
Unknown / anomalous data item or change in context are
detected
Learning & Predictive Model
Scalable Learning Module
Action
Model Update
Incremental Learning Model
• Graceful Degradations (GD)
• Progressive Enhancements (PE)
PE
GD
Progressive Acceptance
Weakened Acceptance / Replace Primary Module

Figure 9. Scalable learning module in IAS

Bitwise fuzzy-based clustering:

BFC is implemented through Perfect Error-correcting codes or Golay codes. Error Correcting Codes (ECC) are used for controlling errors in data (any information that could be represented in bits 0/1). When there is an error in data, the error correcting codes can approximately match the distorted data to the original. For example, take the message (m) = 000. Consider 1-bit distortions of m: 100, 010, 001. All three distortions are 1 hamming distance (it takes 1-bit flip to get to 000) away from 000. So, they can be easily corrected. BFC creates clusters based on fuzzily (approximately) matched data items similar to error correction. For example, take the message (m) = 000 as a data item. m’s 1-bit distortions (100, 010, and 001) will be clustered into one. 0/1 bits are used to label binary features. Assume that 0 – Absent and 1 – Present. Based on number of features of a data item, we can create a binary classification. For example, data item D has 3 features. Presence or absence of each feature creates a code word, say, 101. Code word such as 101 will be a **label** for that data item. Using ECC provides scalability (2^n combinations of clusters) and fault tolerance (distorted labels can be clustered correctly). A binary vector template is created. Based on the presence and absence of it, 0 or 1 is encoded. Consider an image data item:

Features	**YES/NO**
F1: is it red?	0
F2: is it female?	1
.	.
.	.
.	.
F22: is it tall?	0
F23: is it animal?	0

BFC follows Golay error-correction code for labeling which produces 2^{23} unique labels.

Experimental Evaluation:

Recall = (relevant items ∩ retrieved items) / relevant items. For higher hamming distances, the recall probabilities are small. 1—1 means direct matching with 1 hamming distance cluster with the same hamming distance.

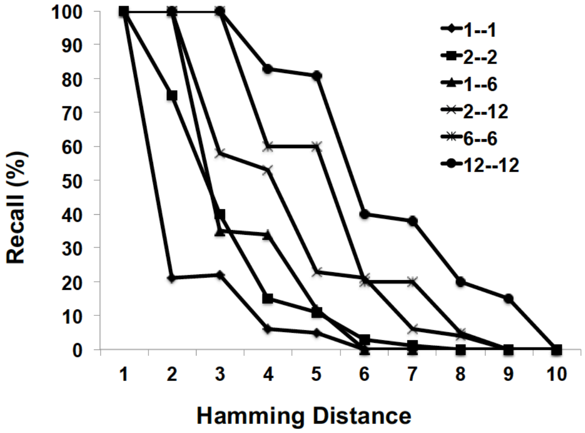

Figure 10. Recall Probabilities for BFC technique

Number of clock cycles for encoding. We used process thread API to collect data, sampled every 1000k instructions.

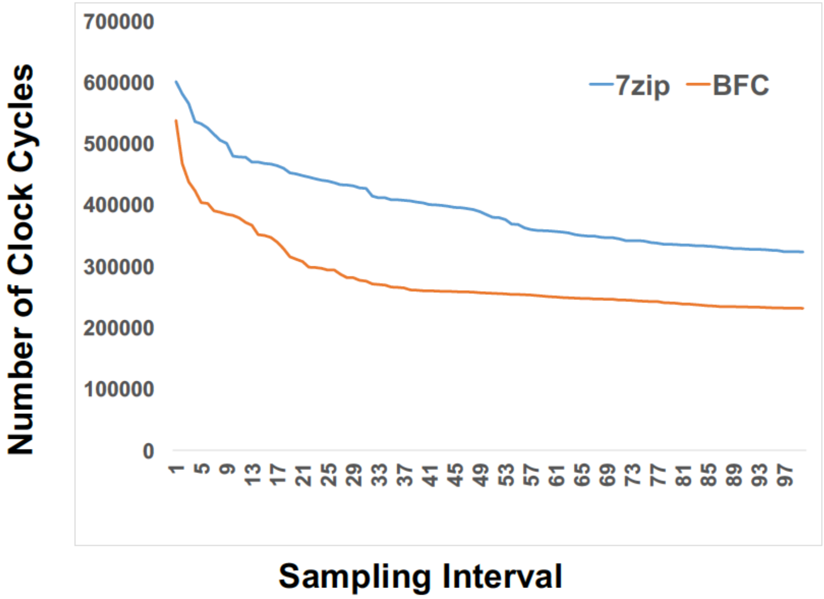

Figure 11. Number of clock cycles required for encoding compared to encoding software 7zip

Clustering Algorithms	Time Complexities
k-means	$\mathcal{O}(nkd)$
Hierarchical Clustering	$\mathcal{O}(n^2)$
Clustering using REpresentatives (CURE)	$\mathcal{O}(n^2\ log\ n)$
ROCK	$\mathcal{O}(min(n^2, nm_m m_a))$
CLICK	$\mathcal{O}(n\ log\ n)$
BFC	$\mathcal{O}(n)$

Table 4. BFC Time Complexity compared to other clustering algorithms

1.4.4 Autonomous Aggregate Data Analytics in Untrusted Cloud: Intelligent Autonomous Systems (IAS) are highly reflexive and very cognizant about their limitations and capabilities, interactions with neighboring entities, as well as the interactions with its operational environment. IAS should be able to conduct data analytics and update policies based on those analytics. These tasks should be performed autonomously i.e. with limited or no human intervention. In this paper, we introduce advanced aggregate analytics over untrusted cloud and autonomous policy updates as a result of those analytics. We used Active Bundle (AB), a distributed self- protecting entity, wrapped with policy enforcement engine as our implementation service. We proposed an algorithm that can enable individual ABs to grant or limit permissions to their AB peers and provide them with access to anonymized data to conduct analytics autonomously. When these processes take place, ABs do not need to rely on policy enforcement engine every time, which increases scalability. This workflow also creates an AB environment that is decentralized, privacy- preserving, and autonomous.

Using Active Bundle (AB), a distributed self-protecting entity with policy enforcement engine, we implement. One-time access certificate used to query other Abs. Privacy preserving aggregation analytics on numerical data. Instead of checking AB's authentication protocol every time, an AB can obtain a one-time pass to access other ABs data per aggregate query. Numerical data is perturbed for the analytics and at the end the perturbation is removed. After passing the authentication and policies enforced by AB's policy enforcement engine, aggregate data analytics can be performed. AB's provenance data is used for aggregated analytics such as *Count, Average, etc.* on qualified attributes. These aggregate analytics guarantee privacy of individual ABs. Consider an aggregation,

AB$_1$'s age attribute is perturbed = "Age (a) " + "Random Perturbation (R)"
= $2AB_1(a + r = a_n) + 2AB_2(a + a_n = a_{n1}) + \dots$
Final average = $(a_{nn} - R) / count(2AB)$.

Experiments:

We measure the latency of data request sent to AB, which is hosted by a local server, located in the same network with the client. As a latency parameter, we record Round-Trip Time (RTT) for the data request processing at the server side (Note: we do not consider network delays in this experiment). ApacheBench v2.3 is used to calculate RTT measurements. We run 50 requests in a row and compute RTT average. Our initial work shows that the policies enforced for each AB access raise the access time exponentially where as a simple python simulation of file access (one-time authentication example) stays almost constant for multiple entities.

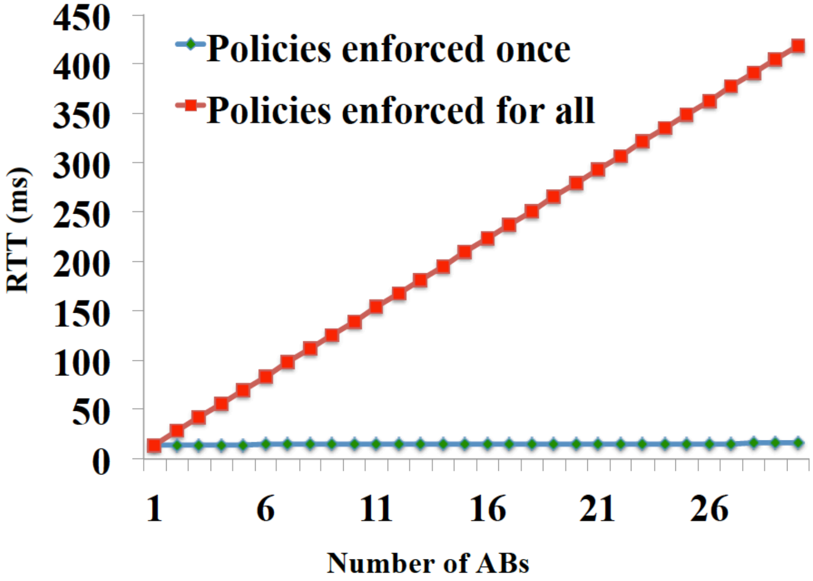

Figure 12. Round trip time for one-time policy enforcement vs. enforcement for each AB

1.4.5 'Blockhub': Blockchain-based Software Development System for Untrusted Environments: To ensure integrity, trust, immutability and authenticity of intelligent autonomous systems and information (cyber data, user data and attack event data) in a collaborative environment, research is needed for cross-domain data communication, global software collaboration, sharing, access auditing and accountability. Blockchain technology can significantly automate the software export auditing and tracking processes. It allows to track and control what data or software components are shared between entities across multiple security domains. Our blockchain-based solution relies on role-based and attribute-based access control and prevents unauthorized data accesses. It guarantees integrity of provenance data on who updated what software module and when. Furthermore, our solution detects data leakages, made behind the scene by authorized blockchain network participants, to unauthorized entities. Our approach issued for data forensics/provenance, when the identity of those entities who have accessed/updated/transferred the sensitive cyber data or sensitive software is determined. All the transactions in the global collaborative software development environment are recorded in the blockchain public ledger and can be verified any time in the future. Transactions cannot be repudiated by invokers. We also proposed modified transaction validation procedure to improve performance and to protect permissioned IBM Hyperledger-based blockchains from DoS attacks, caused by bursts of invalid transactions.

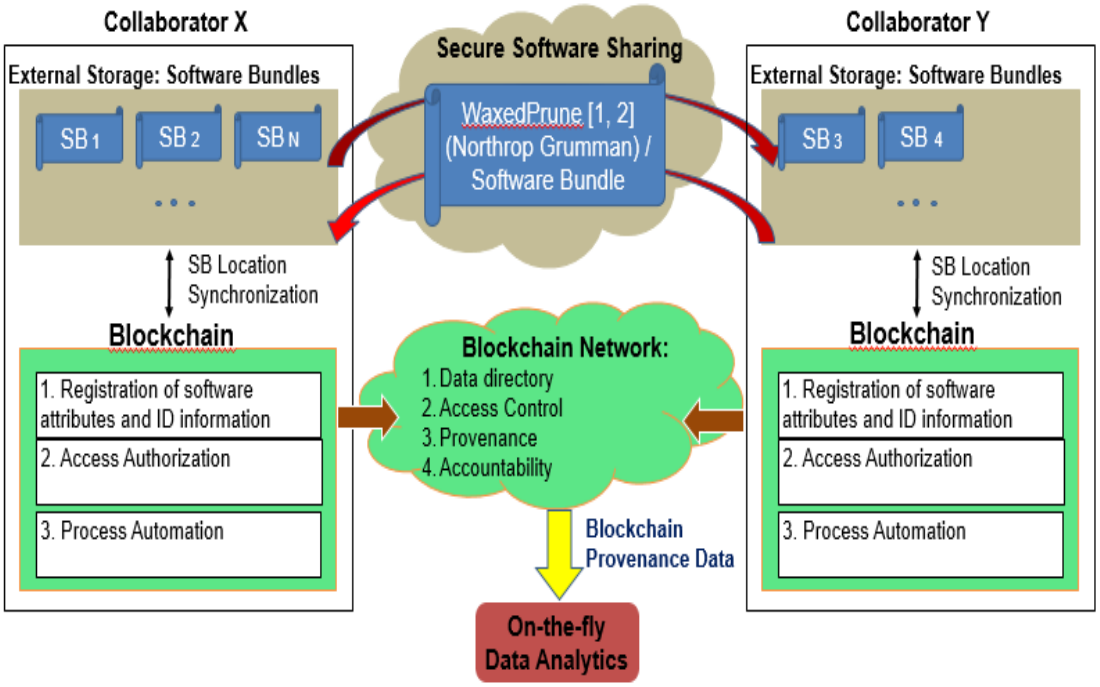
Collaborator X
External Storage: Software Bundles
SB 1
SB 2
SB N
• • •
SB Location Synchronization
Blockchain
1. Registration of software attributes and ID information
2. Access Authorization
3. Process Automation
Secure Software Sharing
WaxedPrune [1, 2] (Northrop Grumman) / Software Bundle
Collaborator Y
External Storage: Software Bundles
SB 3
SB 4
• • •
SB Location Synchronization
Blockchain
1. Registration of software attributes and ID information
2. Access Authorization
3. Process Automation
Blockchain Network:
1. Data directory
2. Access Control
3. Provenance
4. Accountability
Blockchain Provenance Data
On-the-fly Data Analytics

Figure 13. Blockchain-based software management system

In permissioned blockchain, there are three major nodes: (1) Client service invokes a transaction on behalf of a user, submits it to the transaction endorsers and broadcasts the transaction to the sorting module; (2) Committing peers commit the transactions and keep the copy of a blockchain. Some of these peers can have a special role of Endorsers. Before the commitment of a transaction, an endorsing peer checks the validity of the transaction; (3) Sorting module orders transactions in a chronological order and acts as a communication channel between client services and communicating/endorsing peers. This channel outputs the same message to all connected peers in the same chronological order. These communicated messages are nothing but transactions to be included in the blockchain. For efficiency, the blocks are created with a batch of transactions and sorting module imposes a deterministic ordering of transactions in each block. Clients can create a dense set of invalid transactions to make network busy. These bursts of transactions can create a bottleneck in Endorsers thus stopping the transmission of valid transactions from other clients. In the baseline (old) workflow the client service creates a transaction and broadcasts it to Tx Endorsers of its choice. The Endorsers simulate the invoked transaction, check if it is valid and if it adheres to the endorsement policies. After the validation procedure, each Endorser produces an endorsement signature. Client receives the endorsed transactions with Read/Write set and sends it to the Sorting Module. The Sorting Module delivers the transaction to Tx Committers. Tx Committers validate the transactions read set again with database before committing. The block is added to the blockchain with transaction marked as valid or invalid based on the validation. We created the new workflow which eliminates client communication. Permissioned Hyperledger-based blockchain platform architecture Sorting Module. The Tx Endorsers will directly send the endorsed transactions to the Sorting Module. In this way, we created a proper order when multiple clients create transactions at the same time. This order avoids confusion and excessive use of resources even if there was a burst of transactions. The sorting module knows the list of Tx Endorsers and waits for all of them to confirm the transaction. Here we endorsement principle. This solution is scalable, since Tx Endorsers and Sorting Module process clients' invoked transaction requests in parallel. Only transactions that fulfill the endorsement policies are passed directly to the Sorting Module (Ordering Service). Invalid transactions are discarded.

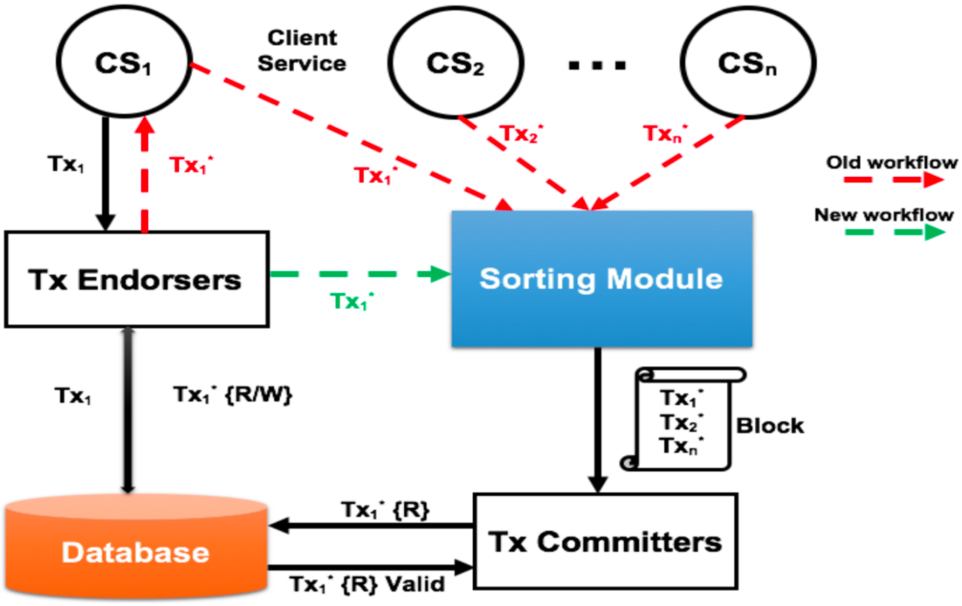

Figure 14. Permissioned Hyperledger-based blockchain platform architecture

Experiments:

We evaluated performance overhead of our 'Blockhub' prototype that combines IBM Hyperledger Fabric platform and 'WAXEDPRUNE' framework. In our experiment, there are three web services in IBM Hyperledger network.

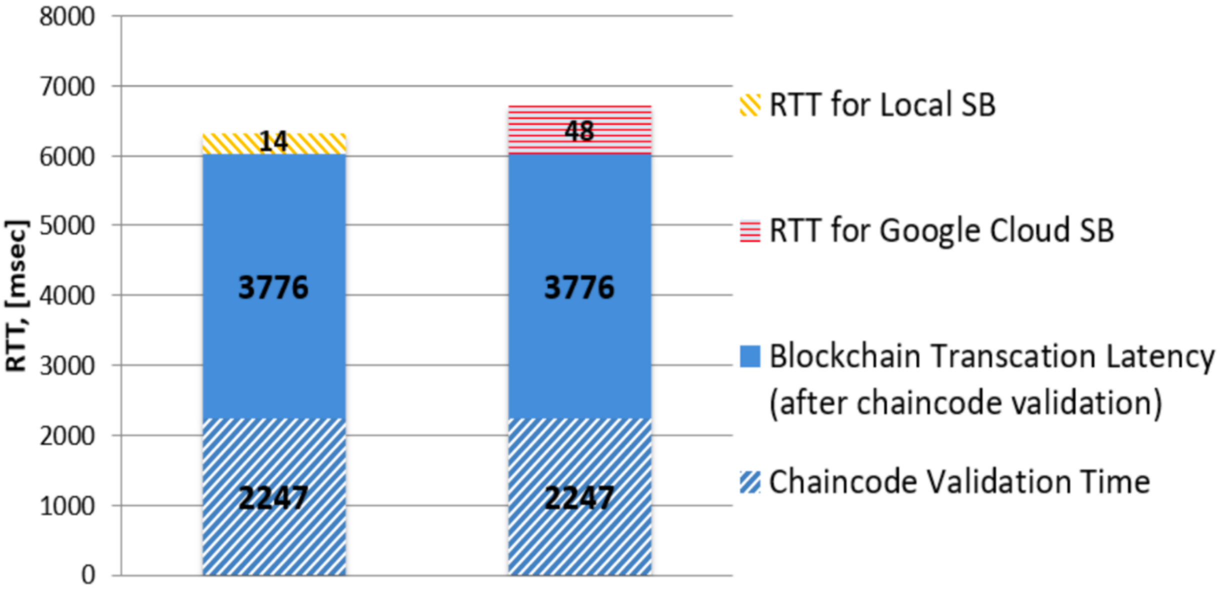

Figure 15. Transaction Latency for Local / Google cloud software bundle

Blockchain transaction latency is way greater than SB RTT. Overall blockchain transaction latency, including chain code validation time, is 6.02 [sec]. Benefits, provided by SB, such as attribute-based access control and leakage detection, impose only 0.23% overhead when SB is hosted by the service in the blockchain network, and 0.8% overhead when hosted by Google cloud instance. IBM announced that new Hyperledger Fabric version 1.1 will have way better performance. Microsoft Coco blockchain platform provides transaction latency 0.152 sec, but it is not open-sourced yet. Once it is open-sourced, we will be able to integrate it in our framework and improve its performance.

1.5 Distinctive Attributes, Advantages and Discriminators

- ***Cognitive autonomy with advanced data analytics:*** Cognitive processing is employed using customized activity pattern identification (multilevel decision trees and hierarchical Bayesian inference) models to understand the parameters of the environment such as processing platform, entities (both systems and humans), and their features to automatically detect and adjust the provenance metrics such as importance factor. If the data is accessed by the client in an unsecured or unverified environment the Bayesian inference to assess the probability of security risks is utilized. Once the risk is determined, the system can store additional provenance data points to monitor closely. If an anomaly is detected with on-the-fly provenance data analysis, then the system would take appropriate actions.
- ***Automated reconfiguration***: Existing approaches for autonomy in IAS lack robust mechanisms to monitor compliance of systems with security and performance policies under changing contexts, and to ensure uninterrupted operation in case of failures. The proposed work will demonstrate that it is possible to enforce security and performance requirements of IAS even in the presence of anomalous behavior/attacks and failure of system modules. The

self-healing will be accomplished through automated reconfiguration, migration, and restoration of modules.

- ***Reflexive systems****:* We designed and implement the reflexive machine learning model to make decisions and trigger corresponding actions for adaptations, while proactively monitoring (being cognitive) the system. We abstract the system runtime from autonomous system to formally reason about its correct behavior. This abstraction allows the framework to enable MTD-style capabilities to all types of systems regardless of its architecture or communication model (i.e. asynchronous and synchronous) on all kinds of platforms). The modules of cognitive monitoring, trust, and automated migration/reconfiguration can be easily integrated into NGC enterprise analytics flow. The modular architecture and use of standard software in the monitoring framework allows for easy plugin to IRAD software. The automation work will allow identification of NGC clients' requirements for building capabilities in prototypes for Air Force Research Lab (AFRL). We plan to work closely with NG on BAA proposals.
- ***Blockchain-based provenance for trust:*** We will use blockchain technology to store provenance data and utilize customized data structure for blockchain implementation. Merkle tree optimizations will be implemented by assigning quantitative measures to the provenance data points to decide the significance of each data point (or set of data points) so that they are sufficient for data analysis and making an informed decision. Only these significant data points will be stored in blockchain. These two techniques will increase the efficiency of implementation.

1.6 Tangible Assets Created by Project

Tangible assets created by this project include implementation of IAS components, demos, experiments, and software simulations. The following components of IAS are implemented:

Cognitive Autonomy & Knowledge Discovery:
Monitors and records system's activities (Data provenance and sequence of system calls)
Conducts privacy-preserving aggregated analytics on provenance data.
Utilizes Deep learning-based anomaly detection by analyzing sequence of system calls.

Reflexivity:
Adaptive actions are performed through graceful degradations without disrupting the ongoing critical processes by incremental learning.

Trust:
Uses blockchain to store provenance data for trust.

Implementations and demos:

Reflexivity prototype for combinatorial replica scheme:

Source code: Node.js implementation, Bayesian model, simulation software developed for combinatorial design, and Data used for simulation. Demo Link: https://goo.gl/M4rXCN

The prototype is built with FAYE framework (https://faye.jcoglan.com/node.html) with Node.js.

Replica updates are done through a combinatorial design simulator (https://goo.gl/pgVHdk).

Blockhub prototype for secure blockchain-based data distribution:
Source code: https://github.com/Denis-Ulybysh/Waxedprune2018 codebase is taken from open-source "Marbles" project https://github.com/IBM-Blockchain/marbles/tree/v4.0

Documentation: Demo video and User manual for running the prototype.

Demo description:
There are three components that are demonstrated.

Demo 1 (Cognitive Autonomy/Knowledge Discovery):

- System is monitored and its interactions with client services are recorded as provenance data.
- Privacy-preserving aggregated data analytics are performed on the provenance data.
- Sensitive data is perturbed with random noise and the noise is removed at the end to obtain aggregated result, protecting the privacy of individual entities.
- A Deep Learning based anomaly detection is implemented to protect against code-hijacking attacks.

Demo 2 (Reflexivity):

- Under anomalous operating contexts or attacks, the replicas in the replacement scheme based on Combinatorial balanced designs take over the processing from primary module.
- Replicas are updated with system states periodically (Update interval is determined through Bayesian inference of system's operating context).
- Unused replicas are used for other processes simultaneously, which makes the system faster and fault-tolerant.

Demo 3 (Trust):

- A scheme that guarantees the integrity of provenance data is implemented.
- Capability to verify every transaction in IAS.

1.7 Outreach Activities and Conferences

NGCRC support helped with progress on four Ph.D thesis. Two students proposed Ph.D thesis proposal to work on this project. One student got fellowship from Purdue to support us. Discussions with Paul Conoval and Jason Kobes during midterm review and meetings at Purdue have been productive. We have disseminated the results of the project with presentations at IEEE Cloud and AIKE (Artificial Intelligence and Knowledge Discovery) conferences in addition to invited keynotes in Zurich, India. The JPL presentation generated a lot of interest in Autonomous Systems work and has resulted in further collaboartions.The following list of publications resulted from the research work involved in this project:

1. G. Mani, B. Bhargava, P. Angin, M. Villarreal-Vasquez, D. Ulybyshev, D. Steiner, J. Kobes. “Machine Learning Models to Enhance the Science of Cognitive Autonomy.” In *IEEE International Conference on Artificial Intelligence and Knowledge Engineering (AIKE),* Laguna Hills, 2018
2. G. Mani, B. Bhargava, B. Shivakumar, J. Kobes. “Incremental Learning Through Graceful Degradations in Autonomous Systems.” In *IEEE International Conference on Cognitive Computing (ICCC),* San Francisco, 2018.
3. G. Mani, B. Bhargava, J. Kobes. “Scalable Deep Learning Through Fuzzy-based Clustering in Autonomous Systems.” In *IEEE International Conference on Artificial Intelligence and Knowledge Engineering (AIKE),* Laguna Hills, 2018
4. D. Ulybyshev, M. Villarreal-Vasquez, B. Bhargava, G. Mani, S. Seaberg, R. Pike, J. Kobes. “Blockhub: Blockchain-Based Software Development System for Untrusted Environments.” In *IEEE International Conference on Cloud Computing (CLOUD),* San Francisco, 2018.
5. G. Mani, D. Ulybyshev, B. Bhargava, J. Kobes, P. Goyal. “Autonomous Aggregate Data Analytics in Untrusted Cloud.” In *IEEE International Conference on Artificial Intelligence and Knowledge Engineering (AIKE),* Laguna Hills, 2018
6. D. Ulybyshev, B. Bhargava, A. Alsalem. "Secure Data Exchange and Data Leakage Detection in Untrusted Cloud." In *International Conference on Applications of Computing and Communication Technologies (ICACCT).* Zurich, 2018 Springer.

Keynote and Invited presentations:

1. Intelligent Autonomous Systems in 11th Central Area Networking and Security (CANSec) workshop, Rolla, Missouri, October 2017.
2. Artificial Intelligence Conference, Las Vegas, April 2018
3. Jet propulsion Laboratory, Pasadena, Ca, June 2018

1.8 Intellectual Property Accomplishments

None.

2 General Comments and Suggestions for Next Year

The TechFest is a great place to learn and develop ideas. We had the opportunity to participate in 2018. Collaboration and visits with other NGCRC universities is a great step forward for the future. The coordinators at NGC are very dedicated. We will appreciate any opportunity for CRADS.

3 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] “The Exciting Future of Autonomous Systems”, Retrieved on August 17, 2017. http://news.northropgrumman.com/news/presentations/wes-bush-addresses-kansas-state-university

[3] Miles, S., Munroe, S., Luck, M. and Moreau, L., 2007, May. Modelling the provenance of data in autonomous systems. In *Proceedings of the 6th international joint conference on Autonomous agents and multiagent systems* (p. 50). ACM.

[4] Tsai, W.T., Wei, X., Chen, Y., Paul, R., Chung, J.Y. and Zhang, D., 2007. Data provenance in SOA: security, reliability, and integrity. *Service Oriented Computing and Applications*, *1*(4), pp.223-247.

[5] Malik, T., Nistor, L. and Gehani, A., 2010, December. Tracking and sketching distributed data provenance. In *e-Science (e-Science), 2010 IEEE Sixth International Conference on* (pp. 190-197). IEEE.

[6] Thuraisingham, B., Cadenhead, T., Kantarcioglu, M. and Khadilkar, V., 2014. *Secure Data Provenance and Inference Control with Semantic Web*. CRC Press.

[7] Glavic, B., 2014. Big data provenance: Challenges and implications for benchmarking. In *Specifying big data benchmarks* (pp. 72-80). Springer, Berlin, Heidelberg.

[8] Bates, A., Hassan, W.U., Butler, K., Dobra, A., Reaves, B., Cable, P., Moyer, T. and Schear, N., 2017, April. Transparent Web Service Auditing via Network Provenance Functions. In *Proceedings of the 26th International Conference on World Wide Web* (pp. 887-895). International World Wide Web Conferences Steering Committee.

[9] Bertino, E., 2015. Data Trustworthiness—Approaches and Research Challenges. In *Data Privacy Management, Autonomous Spontaneous Security, and Security Assurance* (pp. 17-25). Springer, Cham.

[10] Moyer, T., Chadha, K., Cunningham, R., Schear, N., Smith, W., Bates, A., Butler, K., Capobianco, F., Jaeger, T. and Cable, P., 2016, November. Leveraging Data Provenance to Enhance Cyber Resilience. In *Cybersecurity Development (SecDev), IEEE* (pp. 107-114). IEEE.

[11] Gordon, G., 2017. Provenance and authentication of oracle sensor data with block chain lightweight wireless network authentication scheme for constrained oracle sensors.

[12] She, W., Zhu, W., Yen, I.L., Bastani, F. and Thuraisingham, B., 2016. Role-Based Integrated Access Control and Data Provenance for SOA Based Net-Centric Systems. *IEEE Transactions on Services Computing*, *9*(6), pp.940-953.

[13] Zatarain, O.A. and Wang, Y., 2016, August. Experiments on the supervised learning algorithm for formal concept elicitation by cognitive robots. In *Cognitive Informatics & Cognitive Computing (ICCI* CC), 2016 IEEE 15th International Conference on* (pp. 86-96). IEEE.

[14] Dumesnil, E., Beaulieu, P.O. and Boukadoum, M., 2017. Single SNN Architecture for Classical and Operant Conditioning using Reinforcement Learning. *International Journal of Cognitive Informatics and Natural Intelligence (IJCINI)*, *11*(2), pp.1-24.

[15] Machuzak, S. and Jayaweera, S.K., 2016, July. Reinforcement learning based anti-jamming with wideband autonomous cognitive radios. In *Communications in China (ICCC), 2016 IEEE/CIC International Conference on* (pp. 1-5). IEEE.

[16] Titonis, T.H., Manohar-Alers, N.R. and Wysopal, C.J., Veracode, Inc., 2017. *Automated behavioral and static analysis using an instrumented sandbox and machine learning classification for mobile security*. U.S. Patent 9,672,355.

[17] Slavakis, K., Giannakis, G.B. and Mateos, G., 2014. Modeling and optimization for big data analytics:(statistical) learning tools for our era of data deluge. *IEEE Signal Processing Magazine*, *31*(5), pp.18-31.

[18] Mnih, V., Badia, A.P., Mirza, M., Graves, A., Lillicrap, T., Harley, T., Silver, D. and Kavukcuoglu, K., 2016, June. Asynchronous methods for deep reinforcement learning. In *International Conference on Machine Learning* (pp. 1928-1937).

[19] Jaderberg, M., Mnih, V., Czarnecki, W.M., Schaul, T., Leibo, J.Z., Silver, D. and Kavukcuoglu, K., 2016. Reinforcement learning with unsupervised auxiliary tasks. *arXiv preprint arXiv:1611.05397*.

[20] Meyer, D., Feldmaier, J. and Shen, H., 2016. Reinforcement Learning in Conflicting Environments for Autonomous Vehicles. *arXiv preprint arXiv:1610.07089*.

[21] Kuderer, M., Gulati, S. and Burgard, W., 2015, May. Learning driving styles for autonomous vehicles from demonstration. In *Robotics and Automation (ICRA), 2015 IEEE International Conference on* (pp. 2641-2646). IEEE.

[22] Wu, Y., Zhang, Z., Yuan, J., Ma, Q. and Gao, L., 2016, November. Sequential game solution for lane-merging conflict between autonomous vehicles. In *Intelligent Transportation Systems (ITSC), 2016 IEEE 19th International Conference on* (pp. 1482-1488). IEEE.

[23] Tran, L., Cross, C., Montague, G., Motter, M., Neilan, J., Qualls, G., Rothhaar, P., Trujillo, A. and Allen, B.D., 2015. Reinforcement Learning with Autonomous Small Unmanned Aerial Vehicles in Cluttered Environments. *AIAA Paper*, (2015-2899).

[24] Rastgoftar, H. and Atkins, E.M., 2017, May. Unmanned vehicle mission planning given limited sensory information. In *American Control Conference (ACC), 2017* (pp. 4473-4479). IEEE.

[25] Zhang, T., Kahn, G., Levine, S. and Abbeel, P., 2016, May. Learning deep control policies for autonomous aerial vehicles with mpc-guided policy search. In *Robotics and Automation (ICRA), 2016 IEEE International Conference on* (pp. 528-535). IEEE.

[26] Hu, Z., Zhu, M., Chen, P. and Liu, P., 2016. On convergence rates of robust adaptive game theoretic learning algorithms. *arXiv preprint arXiv:1612.04724*.

[27] Endler, M., Briot, J.P., De Almeida, V., Dos Reis, R. and Silva, F.S.E., 2017. Stream-based Reasoning for IoT Applications–Proposal of Architecture and Analysis of Challenges.

[28] Amrouch, S., Mostefai, S. and Fahad, M., 2016. Decision trees in automatic ontology matching. *International Journal of Metadata, Semantics and Ontologies*, *11*(3), pp.180-190.

[29] Zhao, L., Ichise, R., Mita, S. and Sasaki, Y., 2014, November. An Ontology-Based Intelligent Speed Adaptation System for Autonomous Cars. In *JIST* (pp. 397-413).

[30] Liang, X., Shetty, S., Tosh, D., Kamhoua, C., Kwiat, K. and Njilla, L., 2017, May. Provchain: A blockchain-based data provenance architecture in cloud environment with enhanced privacy and availability. In *Proceedings of the 17th IEEE/ACM International Symposium on Cluster, Cloud and Grid Computing* (pp. 468-477). IEEE Press.

[31] Hu, H., Wen, Y., Chua, T.S. and Li, X., 2014. Toward scalable systems for big data analytics: A technology tutorial. *IEEE access*, *2*, pp.652-687.

[32] “UCI Machine Learning Repository: Data Sets”, Retrieved on August 23, 2017. https://archive.ics.uci.edu/ml/datasets.html

[33] Kim, H.M. and Laskowski, M., 2016. Towards an ontology-driven blockchain design for supply chain provenance.

[34] Zhu, M., Hu, Z. and Liu, P., 2014, November. Reinforcement learning algorithms for adaptive cyber defense against Heartbleed. In *Proceedings of the First ACM Workshop on Moving Target Defense* (pp. 51-58). ACM.

[35] Ram, S. and Liu, J., 2009, October. A new perspective on semantics of data provenance. In *Proceedings of the First International Conference on Semantic Web in Provenance Management-Volume 526* (pp. 35-40). CEUR-WS. org.

[37] Simmhan, Y.L., Plale, B. and Gannon, D., 2005. A survey of data provenance in e-science. *ACM Sigmod Record*, *34*(3), pp.31-36.

[38] Wang, J., Crawl, D., Purawat, S., Nguyen, M. and Altintas, I., 2015, October. Big data provenance: Challenges, state of the art and opportunities. In *Big Data (Big Data), 2015 IEEE International Conference on* (pp. 2509-2516). IEEE.

[39] Mnih, V. et al., 2015. Human-level control through deep reinforcement learning. *Nature*, 518, pp. 529-533.

[39] Lilien, L. and Bhargava, B., 2006. A scheme for privacy-preserving data dissemination. *IEEE Transactions on Systems, Man, and Cybernetics-Part A: Systems and Humans*, *36*(3), pp.503-506.

[40] Othmane, L.B. and Lilien, L., 2009, August. Protecting privacy of sensitive data dissemination using active bundles. In *Privacy, Security, Trust and the Management of e-Business, 2009. CONGRESS'09. World Congress on* (pp. 202-213). IEEE.

[41] Ranchal, R., 2015. Cross-domain data dissemination and policy enforcement.

[42] Ulybyshev, D., Bhargava, B., Villarreal-Vasquez, M., Alsalem, A.O., Halpin, H., Steiner, D., Li, L., Kobes, J. and Ranchal, R., Privacy–Preserving Data Dissemination in Untrusted Cloud. IEEE Cloud 2017.

[43] Terziyan, V., Shevchenko, O. and Golovianko, M., 2014. An introduction to knowledge computing. *Восточно-Европейский журнал передовых технологий*, (1 (2)), pp.27-40.

[44] Crosby, M., Pattanayak, P., Verma, S. and Kalyanaraman, V., 2016. Blockchain technology: Beyond bitcoin. *Applied Innovation*, *2*, pp.6-10.

[45] Greenspan, G., 2015. Avoiding the pointless blockchain project.

[46] Bechhofer, S., Buchan, I., De Roure, D., Missier, P., Ainsworth, J., Bhagat, J., Couch, P., Cruickshank, D., Delderfield, M., Dunlop, I. and Gamble, M., 2013. Why linked data is not enough for scientists. *Future Generation Computer Systems*, *29*(2), pp.599-611.

[47] “The Heart of the Elastic Stack”, https://www.elastic.co/products/elasticsearch

[48] “Centralize, Transform & Stash your Data”, https://www.elastic.co/products/logstash

[49] “Your Window into the Elastic Stack”, https://www.elastic.co/products/kibana

[50] M. Villarreal-Vasquez, P. Angin, B. Bhargava, N. Ahmed, D. Goodwin , K. Brin , J. Kobes, “An MTD-based Self-Adaptive Resilience Approach for Cloud Systems”. IEEE Cloud 2017.

[51] Gardiner, E.J. and Gillet, V.J., Perspectives on Knowledge Discovery Algorithms Recently Introduced in Chemoinformatics: Rough Set Theory, Association Rule Mining, Emerging Patterns, and Formal Concept Analysis. *Journal of chemical information and modeling*, *55*(9), pp.1781-1803, 2015

[52] Arasu, A., Babcock, B., Babu, S., Cieslewicz, J., Datar, M., Ito, K., Motwani, R., Srivastava, U. and Widom, J., 2016. Stream: The stanford data stream management system. In *Data Stream Management* (pp. 317-336). Springer Berlin Heidelberg.

[53] Witten, I.H., Frank, E., Hall, M.A. and Pal, C.J., 2016. *Data Mining: Practical machine learning tools and techniques*. Morgan Kaufmann.

[54] Vice, Thomas. "Future of Advanced Trusted Cognitive Autonomous Systems". September 6, 2016. https://engineering.purdue.edu/AAE/aboutus/lectures/rolls_royce/2016_Tom_Vice

[55] G. Mani, B. Bhargava, P. Angin, M. Villarreal-Vasquez, D. Ulybyshev, D. Steiner, J. Kobes. "Machine Learning Models to Enhance the Science of Cognitive Autonomy." In *IEEE International Conference on Artificial Intelligence and Knowledge Engineering (AIKE),* Laguna Hills, 2018

[56] G. Mani, B. Bhargava, B. Shivakumar, J. Kobes. "Incremental Learning Through Graceful Degradations in Autonomous Systems." In *IEEE International Conference on Cognitive Computing (ICCC),* San Francisco, 2018.

[57] G. Mani, B. Bhargava, J. Kobes. "Scalable Deep Learning Through Fuzzy-based Clustering in Autonomous Systems." In *IEEE International Conference on Artificial Intelligence and Knowledge Engineering (AIKE),* Laguna Hills, 2018

[58] D. Ulybyshev, M. Villarreal-Vasquez, B. Bhargava, G. Mani, S. Seaberg, R. Pike, J. Kobes. "Blockhub: Blockchain-Based Software Development System for Untrusted Environments." In *IEEE International Conference on Cloud Computing (CLOUD),* San Francisco, 2018.

[59] G. Mani, D. Ulybyshev, B. Bhargava, J. Kobes, P. Goyal. "Autonomous Aggregate Data Analytics in Untrusted Cloud." In *IEEE International Conference on Artificial Intelligence and Knowledge Engineering (AIKE),* Laguna Hills, 2018

[60] D. Ulybyshev, B. Bhargava, A. Alsalem. "Secure Data Exchange and Data Leakage Detection in Untrusted Cloud." In *International Conference on Applications of Computing and Communication Technologies (ICACCT).* Zurich, 2018 Springer.