COMPUTERS & SECURITY 73 (2018) 439–458

Available online at www.sciencedirect.com

ScienceDirect

journal homepage: www.elsevier.com/locate/cose

Computers & Security

Intelligent agents defending for an IoT world: A review

Rory Coulter [a,*], Lei Pan [b,*]

[a] Swinburne University of technology, School of Software and Electrical Engineering, Faculty of Science, Engineering and Technology, Hawthorn, VIC 3122, Australia

[b] School of Information Technology, Faculty of Science, Engineering and Built Environment, Geelong, VIC 3220, Australia

ARTICLE INFO

Article history:
Received 17 January 2017
Received in revised form 4 September 2017
Accepted 20 November 2017
Available online 5 December 2017

Keywords:
Intrusion detection
Artificial intelligence
Machine learning
Internet of Things
Cyber security
Data driven cyber security

ABSTRACT

Transition to the Internet of Things (IoT) is progressing without realization. In light of this securing traditional systems is still a challenging role requiring a mixture of solutions which may negatively impact, or simply, not scale to a desired operational level. Rule and signature based intruder detection remains prominent in commercial deployments, while the use of machine learning for anomaly detection has been an active research area. Behavior detection means have also benefited from the widespread use of mobile and wireless applications. For the use of smart defense systems we propose that we must widen our perspective to not only security, but also to the domains of artificial intelligence and the IoT in better understanding the challenges that lie ahead in hope of achieving autonomous defense. We investigate how intruder detection fits within these domains, particularly as intelligent agents. How current approaches of intruder detection fulfill their role as intelligent agents, the needs of autonomous action regarding compromised nodes that are intelligent, distributed and data driven. The requirements of detection agents among IoT security are vulnerabilities, challenges and their applicable methodologies. In answering aforementioned questions, a survey of recent research work is presented in avoiding refitting old solutions into new roles. This survey is aimed toward security researchers or academics, IoT developers and information officers concerned with the covered areas. Contributions made within this review are the review of literature of traditional and distributed approaches to intruder detection, modeled as intelligent agents for an IoT perspective; defining a common reference of key terms between fields of intruder detection, artificial intelligence and the IoT, identification of key defense cycle requirements for defensive agents, relevant manufacturing and security challenges; and considerations to future development. As the turn of the decade draws nearer we anticipate 2020 as the turning point where deployments become common, not merely just a topic of conversation but where the need for collective, intelligent detection agents work across all layers of the IoT becomes a reality.

* *Corresponding authors.*
E-mail addresses: research@coulter.io (R. Coulter), l.pan@deakin.edu.au (L. Pan).

https://doi.org/10.1016/j.cose.2017.11.014
0167-4048/

1. Introduction

The approach of defending Information and Communications Technology (ICT) resources is a continually developing landscape that requires the attention of both researchers and professionals alike. No one system is foolproof or immune to the innumerable variance of attack and exploitation. With the development of information systems, security mechanisms have fought to keep in touch with actors that seek to exploit not only device or data, but also the fabric of computer systems. The nature of computer systems tread a fine line between security, functionality and ease of use; whereby shift only a little in favor of one, and risk the impedance of others. Intruder Detection/Prevention Systems (IDS/IPS) are but one mechanism that can aid in strengthening cyber-defenses, providing a means to monitor or constrain malicious network interactions (Sobh, 2006).

A significant drawback of detection systems is intrusions deemed to be false positives (FP), where a determined intrusion results in being false. FPs generate noise within the environment of positive occurred attacks. Several approaches exist in a means to deploy detection by affording intelligence mechanisms in reducing FP noise, *Misuse*, *Anomaly* and *Behavioral*. Misuse compares activity to rules or known attack signatures, anomaly seeks to divide unknown traffic of normal and malicious classes, while behavioral, or specification, is concerned with operational patterns. Of these means misuse detection is mostly employed in live deployments, yet suffers from zero-day, or unknown attacks. Yet in contrast to intruder detection, the use of intelligence has been successful within other computing domains such as sign language recognition (Yang et al., 2015), improved robot planning (Galindo et al., 2004), facial (Hsu et al., 2002) and sketch to photo recognition (Wan and Panetta, 2016), real-time object tracking (Stauffer and Grimson, 2000), visualization in chess (Lu et al., 2014) and multi-agents for traffic signaling improvements (Balaji and Srinivasan, 2010). To better determine the current approach of defense systems with intelligence, we present detection aligned with the intelligent agent framework defined by Russell et al. (2003).

A new challenge is faced with the development of the Internet of Things, or everything (IoT), considered a new communication direction in aiming to bridge the physical with the cyber world. Whereby the integration of connected systems, objects and devices, homo- and heterogeneous alike, provides access to untold services, information and application (Perera et al., 2014; Xu et al., 2014; Zanella et al., 2014). Given the increased connection of devices, and the generation of large sums of data, both personal and system, previous security methodologies require adaptation in order to maintain defensive expectations. The structure of an IoT environment sees communication and cooperation across many different system levels; the evolution of computing structures requires adaptive and self-adaptive technologies to maintain affordable security. Faith to garner its potential ability to operate and provide a level of expected security go hand in hand, as suggested by Stankovic (2014), considerations are needed due to the capacity of devices from a security perspective.

This paper is concerned with the current approaches of intrusion detection, its modeling from an intelligence perspective, and the security challenges for defense systems in the IoT. Contributions made within this review are the review of literature of traditional and distributed approaches to intruder detection, modeled as intelligent agents, for an IoT perspective; defining a common reference of key terms between fields of intruder detection, artificial intelligence and the IoT, identification of key defense cycle requirements for defensive agents, relevant manufacturing and security challenges; and considerations to future development.

The rest of the paper is organized as followed: Section 2 provides an overview of each domain and defines a collective context definition. In Section 3 we discuss agent models and their intelligence with respect to research of IDS systems. In Section 4 we discuss the use of intelligence, limitations and future challenges. Summaries of sections are added where appropriate, finally the paper is concluded in Section 5.

2. Background and related work

With three distinct fields, one almost within its infancy, we provide a summary or definition of key attributes relevant to each field in Section 2.1–2.3. A brief summary of literature is provided in Section 2.4, identifying the key topics previously engaged. Importantly, we position a definitive intelligence model to draw and compare aspects for intrusion detection, IoT systems and intelligence terms into one collective reference in Section 2.5.

2.1. *Intrusion detection*

In securing networks and devices from attacks, several mechanisms are available. *Firewalls* are software or hardware devices that permit or deny traffic within the boundaries of a network or device. Their logic is based on predetermined conditional statements (Protocol, IP, Port etc.), and operating in either of two distinct operational modes, Stateless or Stateful. Stateless action is determined based on packet headers against defined rules, i.e. Drop or Allow. Stateful action concerns the *state* of the connection, which is further considered. For example, if an incoming connection was not initiated from an internal host, but does not violate any rules, a stateful inspection would drop the connection based on this state condition.

Intruder Detection System (IDS) monitors the flow of network traffic to determine intrusion attempts, possible attempts may be logged or trigger an alert. *Intruder Prevention System* extends detection capabilities to actively alter connections that do not meet predefined rules, i.e. dropping traffic. Detection methods can be classified within one of three classes: *Signature* based, *Anomaly* based and *Specification* based (Sobh, 2006). Signature detection compares attacks/traffic against known signature patterns of malicious behavior, they possess knowledge of previous attacks and a means to mitigate in the forms of rules or signatures. Anomaly detection classifies possible attacks against what is perceived normal, and perceived anomalous; it classifies patterns deviated from its normal traffic baseline.

Specification based detection, or *Stateful Protocol Analysis* (SPA), detects possible malicious activity by pairing protocol states; requests with replies or protocol metrics in determining activity that does not fit the prescribed behavioral model (Sobh, 2006).

Environmental positioning of either of these models can either be Network based or Host based. Network based, or *NIDS*, seeks to monitor the flow of network traffic, while Host based, *HIDS*, is concerned with monitoring a single host's activity, logs and system calls (Sobh, 2006).

2.2. *Intelligent agent*

The field of artificial intelligence has had many contributions, which convey the principles, theories or practical methods of intelligent systems or applications. In particular, the contribution of Russell et al. (2003), *Artificial Intelligence: A Modern Approach*, has seen a wide use in both university education, and has provided a unified reference to the field of Artificial Intelligence (AI). This source is chosen as which to conceptualize the two supporting fields for discussion within this review by mapping intelligent functionality of IoT devices, or intruder detection systems within the defined model of *intelligent agents*. Particularly representative of IoT devices, we see a scale of simple low processing actions of sensors to complex smart devices that act autonomously; we feel this model appropriately complements the agent model presented. The basic form and functionality of devices themselves can be seen exhibiting levels of design, engineering and operational ability; if we consider the traditional autonomous disk vacuum cleaner, a simple reflex agent might proceed left, right, up and down, however, smart devices may coordinate with other sensors within a smart home and determine more attention be applied to a given location. In this example, as the ability increases so too does the intelligence, and this is representative of agents presented in Russell et al. (2003).

An intelligent agent comprises of several characteristics, respective of its name. An *agent*, be it hardware or software, is recognized through perceiving the environment in which it resides through sensors. An agent through effectors undertakes actions within this environment. For what an agent has perceived is regarded as its percept sequence, an agent that has as complete percept sequence is possibly considered optimal. Through this function, agents should return with the most correct action, maximizing a determined performance measure. The combination of these elements, performance measure, percept sequence, environment knowledge, and determining which actions to take lead to the idea of an ideal rational agent.

An *intelligent* agent encompasses the previous elements, while also firstly affirming rational action, choosing the most logical action in response to its known percept sequence. Secondly an ideal mapping of percept to action, invoking the most correct action further contributes to our intelligent tag. The most ideal mapping of percept to action could simply be preprogrammed into each agent for each given state, but this would require an indefinite list of actions requiring an indefinite period of attention; what then we require is a sense of autonomy. We simply cannot expect an agent to function without some prior knowledge, if we look at human agents, we spend years fostering and teaching until autonomous actions can overtake guided or instructed action, thus a sense of autonomy finalizes what we consider an intelligent agent. We require autonomy for our agents, agents need to be adaptive and flexible, as we cannot prescribe them with all knowing actions. The perfect rational agent is yet to exist, if it did, we would not be required to develop agents, rather reassign their functionality. Our IoT environments will require many different types of autonomous agents (Desai et al., 2015; Rivera et al., 2015), importantly those which can defend, coordinate, develop and heal their environment. The model of intelligent agents as selected from Russell et al. (2003), displayed in Fig. 1, shows functionality increasing from Reflex to Leaning.

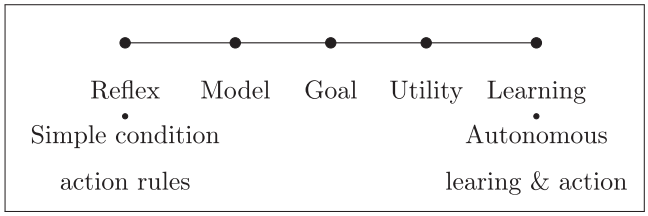

Fig. 1 – Agents characteristics enable categorization into one of five classes. The ability, or intelligence increases through the agent models. Reflex agents reside at the lower end of the scale, while Learning agents pose the highest level of intelligence and ability to act autonomously.

Reflex agents operate by simple means of matching condition-action rules based on the current percept information. Stateless in nature, a lookup table would be impractical and exceedingly large for reflex agents. Thus action is taken by the first matched rule based on the perceived environment.

If we consider our simple IDS installations that run based on rules, we could substitute a lookup table, whereby for each conditional environment element, an existing rule exists to counter the occurrence. This may be a too costly operation and detection agents will soon become inefficient with a very large number of entries.

As seen bellow, **if** the condition that the **source** matches the IP address 10.0.0.1 on **port** 3001, the **alert** action is to be taken:

if *src* 10.0.0.1 *port* 3001; *alert*
if *src* 10.0.0.1 *port* 3002; *alert*
if *src* 10.0.0.1 *port* 3003; *alert*

Rules allow greater overreaching control and efficiency, summarizing many lookup entries.

if *src* 10.0.0.1 *port* 3000–3003; *alert*

Information is gained through their percept sequence and acted based on their prescribed knowledge and intelligence of their condition action rules.

State or Model agents build upon conditional statements with the inclusion of state based history, that is, the state of the environment around them and how it changes. An agent's state is generated through their percepts, and retained in keeping track of how their world evolves.

IDS models that make use of machine learning (ML) techniques, be it unsupervised, semi-supervised or supervised, help

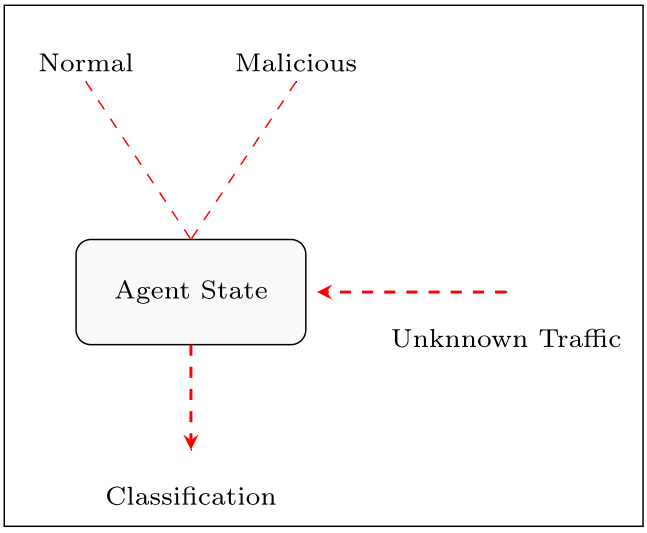

Fig. 2 – An agent's *State*, or perceptually perceived history, builds an internal memory that enables agents to determine actions. As an example, detection agents are trained through various machine learning techniques with normal and malicious traffic. This enables agents to build an internal state for classification of traffic or attack characteristics. These then allow new, or unknown traffic to be categorized as either normal or malicious.

agents retain an internal state. If we consider supervised ML involving classification methods as an example, training data consisting of either normal or malicious traffic are classified by their respected labels. For any new data that are perceived by the agent, traffic is compared and classified within its respective class based on training representation as seen in Fig. 2.

Knowledge about agents environment is not necessarily enough in making the best decision, though one or several actions maybe decided upon, the addition of *goals* allows support in reaching the desired results or state where situations arise and agents consider, what is required of me now? In combination with environment knowledge, goals allow consideration of possible actions and their results, allowing decisions that fulfill an agent's aims to be chosen.

For instance, IDS may achieve goals through monitoring the state of their environment, maintaining repeated information based on node behavior. While Reflex agents may act unnecessary to every environmental occurrence, goal based agents seek to maintain an optimal environment through reaching a desired state.

This idea of a desired state, or goals, may be set through organizational security policy in determining actions based in environment instances, or traffic that require attention, decisions are then made reaching these goals.

Utility agents help establish higher quality action through association of a utility value to one goal over another. That is, one goal state maybe preferred over another, to distinguish between, a utility function may specify a goal and its actions result in a better environment state, thus utility helps map a state to a given function. Furthermore a utility value may also aid goal decision-making, as goals may be conflicting, and where a collection of goals do not necessarily guarantee success in a certain state, utility may increase the likelihood of success being achieved through logical decision. Thus agents can prioritize their action to reach an optimal environment state and utility determination helps with agent functionality in enforcing a prioritized decision.

Learning agents encapsulate all previous functionality of our agents, and seek to grant further logical reasoning and independence from previous agents. Their make-up consists of:

- *Critic* pertains to a means of providing feedback to the Learning agent. By doing so, we can validate the perceptual input, as percepts do not guarantee indication of success. A measure of feedback is provided via a fixed standard outside the control of an agent, so that it maintains performance standards.
- *Performance* element pertains to action selection, this may also be considered as previous agents.
- *Learning* element determines requirements to improve the performance element.
- *Problem* generator suggests alternative action that provides a means of obtaining new information and a learning experience. Detection agents can act autonomously with learning functionality and increasing their performance.

We will model IDS systems, their processes and functionality with respect to these model structures. By aligning the current research with this model, we aim to identify what elements contribute to a logical IDS agent.

2.3. *Internet of Things*

The Internet of Things (IoT) is seen as the next progression of the Internet with the interconnection of homo- and heterogeneous devices allowing autonomous and smart applications by connecting devices and sensors to Internet applications or services (Desai et al., 2015; Rivera et al., 2015).

The physical environment is perceived and measured through nodes that aid in providing services or information to smart applications. The architecture consists of a Perception, Network, Application layers (Gou et al., 2013; Zhao and Ge, 2013). Other architecture modeling supports a Perception, Network, Middleware, Application and Business layer (Khan et al., 2012). We will consider the middleware and application layer as one by example only, and follow a three-layered model as in Zhao and Ge (2013).

The Perception layer consists of small form-factor computing, sensors and radio frequency devices that collect data from or interact with the physical environment. Interpreted data are passed to the Network layer for transmission, which is then passed to the Middleware/Application layers for service processing and decision-making.

Intelligent agents, whatever form they may be, are required for all level of the IoT domain, conducting many different types of operations and services, and facing several security and privacy challenges at each perspective layer, including traditional and new security challenges (Zhao and Ge, 2013).

Fig. 3 highlights the communication that may be required between agents and their given surrounding levels. Perception layer nodes interact with other nodes at the same level, while either forwarding or receiving traffic from a sink or cluster

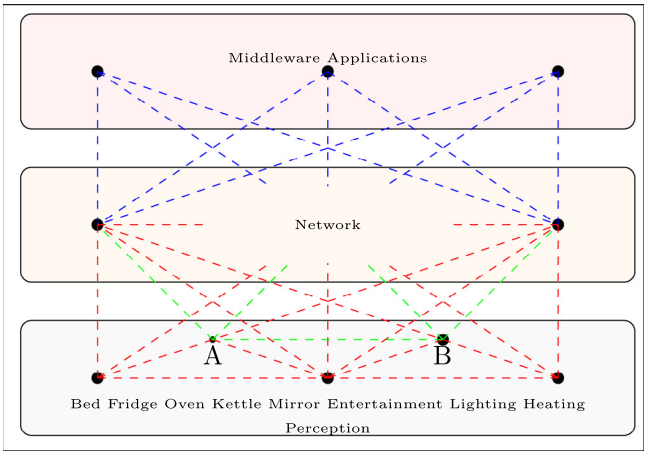

Fig. 3 – Agent interaction: communication requires agents interaction not only at their given level, but surrounding levels. Disruptions, compromise or the removal of nodes require reaching coordination among peers. Routing protocols establish path and communication overhead reduction, defense mechanisms should allow inter-layer coordination.

head (*A* or *B*), or the required network layer node. Network layer agents communicate between both layers. This result highlights the need for agents to not only consider their own layer, but also the supporting layer either above or below.

Consider a scenario of living immersed within an IoT world. Prior to waking up in a morning, average waking patterns are interpreted to begin heating systems so that a desired household temperature is achieved. Upon waking, sensors within the bed record that its occupant has woken and this data is used by smart systems to prepare morning behaviors. This information is conveyed by smart applications to prepare the kettle to begin boiling water for morning coffee, entertainment systems are engaged to begin broadcast of news radio, lighting is triggered as the occupant moves around the premises. The fridge begins to adjust department temperature settings for the morning meal in time with the coffee. Morning tasks in front of the mirror are achieved with interactive display showing the weather, calendar appointments and news items. Pattern analysis is continually conducted to match behavioral profiles to pre-emptive disable device functions, save power or improve learning for optimizing object use.

Traveling to work tire sensors update safety requirements for air pressure and road conditions are relayed to city maintenance facilities. Traffic management information is shared to provide real time traffic conditions, while monitoring aids observe driver behavior in case of fatigue.

Drone mail services operate in various roles, one of which mail is delivered to recipients. Within operation, sensors and smart decisions are made to best conserve battery consumption respective to perceived environment conditions (Moisture, Light, Temperature) and those aggregated by other drones.

Collectively, these agents greatly optimizes functionality, consumption or costs however, the potential of abuse is alsoe increased. Behind the scenes of these services reveals network layer devices transporting data to middleware and application level services to interpret and analyze. Considering this scenario, our expectation for intrusion detection generally resides within the network layer and potentially cluster head control devices. Considering intrusion detection functionality enabled within a kitchen toaster, a household mirror, or a bed seems far-fetched. Everyday devices are now sensing their environment and generating data for service functionality. The condition of this data may be sensitive in nature, enough for motivation to illegally obtain or abuse. Furthermore, the increase of device capable of IP communication poses a threat for potential abuse or attack. Detection systems are required within each end device node, each level of network segmentation and perimeter boarders.

2.4. *Defense cycle*

Much has been made in incorporating intelligence into defense models in detecting anomalous behavior; indeed several reviews have been conducted in that regard. Security controls of Software Defined Network such as Firewalls, Access control, IDS/IPS, Policy, Monitoring and auditing are reviewed in Alsmadi and Xu (2015), cloud intrusion detection techniques in Modi et al. (2013) and alarm management in Patel et al. (2013), IDS/IPS technologies, approaches and methodologies in Liao et al. (2013) and the use of Ensemble techniques in Folino and Sabatino (2016). Computational intelligence concerned with associated algorithms is surveyed in Wu and Banzhaf (2010). Of these reviews, it can be concluded that insider threats, issues of adaptation of detection systems, scalability of nodes and associated data amounts are but a few of the surrounding issues. We also determine that a large portion of the literature surveyed is concerned with the optimization of data and analysis of results, represented as the sensing portion of Fig. 4.

Interestingly real-world intruder systems face similar problems (Keung et al., 2012; Mohapatra et al., 2016), yet if we are to draw any conclusions from the reviewed literature we can see a common approach to intelligent intrusion detection.

Deployment cycles follow four key processes in determining anomalous behavior as seen in Fig. 4. From monitoring, information is extracted of key features and aggregation processes are conducted in helping to better determine points of interest to aid anomaly detection. Much is afforded toward information gained through handling the surrounding data used to test and train IDS agents, yet the make-up of these systems face inherent drawbacks for an IoT environment, namely autonomous action and security analysis. For an IDS system in monitoring the perimeter edge traffic, merit can be found, but will still operate traditional functionality lacking the required intelligence and autonomy that is needed (Rivera et al., 2015).

2.5. *Definition*

The intersection of *Intruder Detection*, *Artificial Intelligence* and the *Internet of Things* domains share common attributes. In unifying further development of smart security in an IoT context, a common context definition is presented in Table 1.

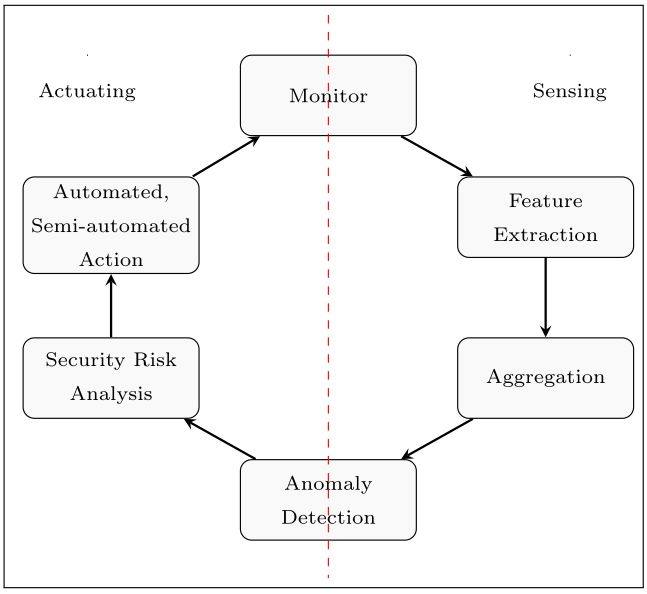

Fig. 4 – Autonomous process: considerable effort has been applied to the Sensing operations in establishing IDS systems. Attacks and threats continually evolve, so too should defensive agents. Greater effort is required for autonomous action and support through agents. (For interpretation of the references to color in this figure, the reader is referred to the web version of this article.)

3. Models and intelligence

Section 3.1 outlines the fitting of traditional approaches to intrusion detection (single instance, non distributed) approaches within our agent model. We identify characteristics unique to each deployment or research type and match these with agent qualities. Section 3.2 continues much like Section 3.1 yet attention is turned to distributed approaches (multiple instances, collaboration) and again their given agent assignment.

The defense of infrastructure, or determining actions that occurred within networks is a cycle that mirrors the adaptation of new attacks faced daily. The cycle of defensive strategies or countermeasures is intertwined with attack methodologies employed by villains. Of the time given to detection efforts, both within industry and academia, it is clear that malicious activity simply cannot be completely eliminated, and will forever plague systems. There is a clear intersection between the two paths of defense and attack; where monitoring of the network is conducted to determine an attack. In addition, continual work is given to keep up with the evolving attack landscape; with the increase of connectivity, the amounts of personal data generated and the potential for widespread abuse, defensive mechanisms need to be evolving in time with wrongdoers. We apply our formal model of an agent against attempts to enable this evolution within the reviewed literature; in doing so, we classify the attempts into either one of two categories, traditional or distributed modeling. Traditional approach is concerned with single agents following a single reference point like a firewall within a network perimeter; while a multi-agent approach is for distributed agents where several agents coordinate either among nodes, as cluster heads or base stations for a collection of nodes, or as autonomous agents.

3.1. *Traditional defense mechanisms*

The knowledge defined within traditional detection system consists of either *misuse* or *anomaly* detection, with respect to the intelligence of these agents we can see implementations of reflex and model-based agent instances.

Security professionals, as experts, have previously interpreted environment conditions into conditional rules for agents to use in monitoring, these deployments fit a reflexive model. ML integrates memory functionality that allows agents to remember instance or occurrence and an associated class, thus maintaining an internal state aligned with model agents. Typically there have been single instance deployments that either sniff, log or alert to real-time or previous environment conditions based on a threat cost.

Defining their mechanisms we categorize them based on increasing intelligence ability: Reflexive process, and Model process in terms of *soft*, *medium* and *hard* states.

3.1.1. *Reflex*

Simple reflexive agents model instances of Snort, Bro deployments, consist of predefined intelligence or programmed logic, where no knowledge of state factors are retained. Configurable extras such as preprocessor, retain limited knowledge per goal instance based on programmed requirements. We will consider rule based detection approaches as opposed to signature libraries. Signature detection is constructed after an attack has occurred; we seek to aid the development of more reactive approaches. A collaborative effort based on user cases and given qualities has seen rule recommendation (Sonchack et al., 2015) via a defined knowledge base. Data processing (cleaning, reduction, standardization, etc.) is combined with k-Means clustering for abstraction via data mining techniques (Haque et al., 2012), numerical and binary processing is conducted in Ji et al. (2016) in establishing a greater model understanding, aided with Support Vector Machine (SVM), and rule pattern analysis for dynamic construction in reducing rule libraries (Chen et al., 2009). Dataset analysis with regards to required state information for rules construction (Lin et al., 2012), uses Simulated Annealing (SA) and SVM for parameter optimization, then SA and Decision Tree (DT) for decision rule formulation. The result of these processes allows the configuration of rules based on expert or machine intelligence.

3.1.2. *State representation*

Various machine learning methods are used to build an agent's state knowledge through learning techniques; unsupervised, semi-supervised and supervised learning. Depending on the level of complexity added to this process, analysis or further maximization of the training data, we define a subset of state leveling. Soft states, Fig. 5, are defined through data normalization and feature selection; Medium state, Fig. 6, is defined through data clustering, data profiling or optimization and information gain before training; and Hard state, Fig. 7, is a representation via ensemble techniques, parameter tuning, higher level agent ability incorporated for state adjustment.

Table 1 – Shared contexts: common items that overlap between the Artificial Intelligence, Intrusion Detection and the Internet of Things domains are referenced. This table establishes a collective description that is unique to the union of the respective fields of study.

Item	Intelligent agent (Russell et al., 2003)	IDS (Lugo-Cordero and Guha, 2013; Sobh, 2006)	IoT (Desai et al., 2015; Rivera et al., 2015; Zanella et al., 2014)	Context definition
Agent	Physical or software instance that interacts logically with its environment	IDS device or software installation of varying operational ability and autonomy	IoT device with capabilities of intelligence and reasoning	Instance of an IDS that possess a given level of intelligence
Sensor	Means for input of agent's environment. Imagery, Audible, Environments, etc.	An IDS devices Network Interface Card (NIC) or simple IP device (fixed or wireless)	Small wireless IP device of limited processing, energy; either IP or combined with environmental sensing abilities	Small form factor device that transmits via IP, fixed or wireless communications. Combined with IDS and possibility for environment sensory input
Effector	Provides the ability to interact with environment	Decision that results in alteration of network characteristics or traffic	Physical or software that grants ability to interact	Element for agent to interact with environment
Percept (sequence)	Perceived history	Perceived history	Perceived history	The observed history of an agent
Autonomy	A means to act independently	A means to act independently	A means to act independently	Ability to act independent
State	Understanding of environmental condition	Operational mode, condition of its being or state analysis of connection	Operational mode or condition of its being	Conditional being and environment conditional memory
Goal	Objective of ideal state	Achievement of given action	Achievement of given action	Task, Ideal state, Objective, etc. to achieve
Utility	Performance measure for achieving goal	Value associated to goal, or purposefulness	Value associated to goal, or purposefulness	Achievement measure:
Critic	External party, determine success and means providing feedback	External expert or node (i.e. specification based)	Node or control head	External expert or master node/device
Node	Data reference within search tree	Reference to device or sensor	Reference to device or sensor	Reference to device or sensor
IDS	N/A	Intruder Detection System	Intruder Detection System	Intruder Detection System
Multi-agent	Multiple version of Agent, may be independent of each other	Multiple version of Agent, may be independent of each other	Multiple version of Agent, may be independent of each other	Multiple version of Agent, may be independent of each other

For example, a simple installations state is as representative as a hard line model is unfair and untrue, as experts we control what knowledge is provided, thus we can categorize this level as a hard state.

3.1.3. *Soft*

State representation, in which light data preparation is applied, is not well reflected in the research; the quest of optimization has spurred further preparation techniques. Moreover, Fang and Liu (2011) reflect real-world application of incorporated intelligence. Classification of probe attacks is achieved through numerical processing of data in training an Artificial Neural Network (ANN), (Elman NN), incorporated through a Snort preprocessor.

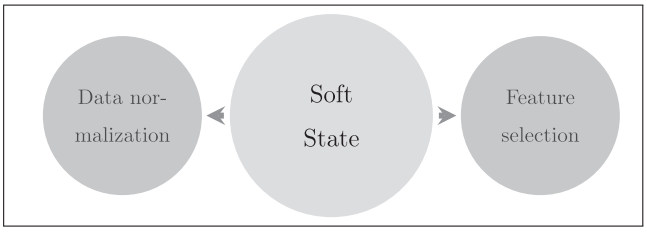

Fig. 5 – Soft state: light data analysis, adjustment and optimization are conducted.

3.1.4. *Medium*

Approaches as seen in Fig. 6 attempt to optimize the data used to define the state view. Outcomes of ML are reliant on the data representation used to train the state space, good representations encourage better results and feature selections allow removing noise from data to better enable learning (Armanfard et al., 2016; Bengio et al., 2013).

Information gain of features is conducted with classification and processing combined with DT in Sangkatsanee et al. (2011), clustering and neighbor distance with Known Nearest Neighbor (k-NN) in Lin et al. (2015), and area triangulation of k-Means clusters is used for training (k-NN) classifier in Tsai and Lin (2010). Extreme Learning Machine (ELM) is used in combination with batch training over large training samples in Creech and Jiang (2012), MapReduce for data processing in Xiang et al. (2014), and alpha and beta profiling combined with 10-fold validation in Online Sequential-ELM in Singh et al. (2015). A feature

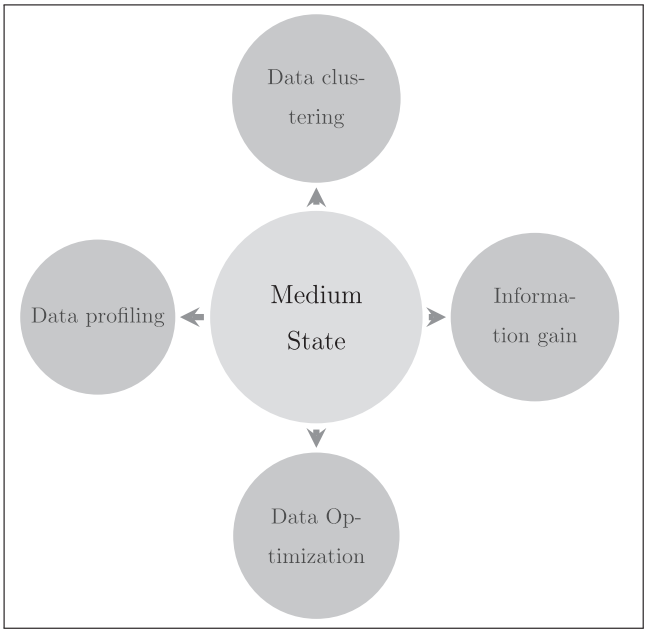

Fig. 6 – Medium state: data optimization is focused in best representing features, elements or representation of attacks.

selection algorithm is proposed in Ambusaidi et al. (2016) and combined with Least Square Support Vector Machine (LSSVM).

3.1.5. *Hard*

Hard state awareness, Fig. 7, allows classifiers to best determine the nature of intrusion attempts; in having as complete percept sequence as possible, the most rational decision may be made (Bengio et al., 2013). We look at Ensemble and Hyperparameter approaches as a means to best provide representation. These methods provide an optimized approach, whereby multiple parameters are individually tuned, or the approach itself, say at a kernel level, is optimized. Hyperparameter optimization has shown to improve ML classification (Thornton et al., 2012), while ensemble learning has shown to increase state representation (Zhang et al., 2015; Zhou et al., 2002).

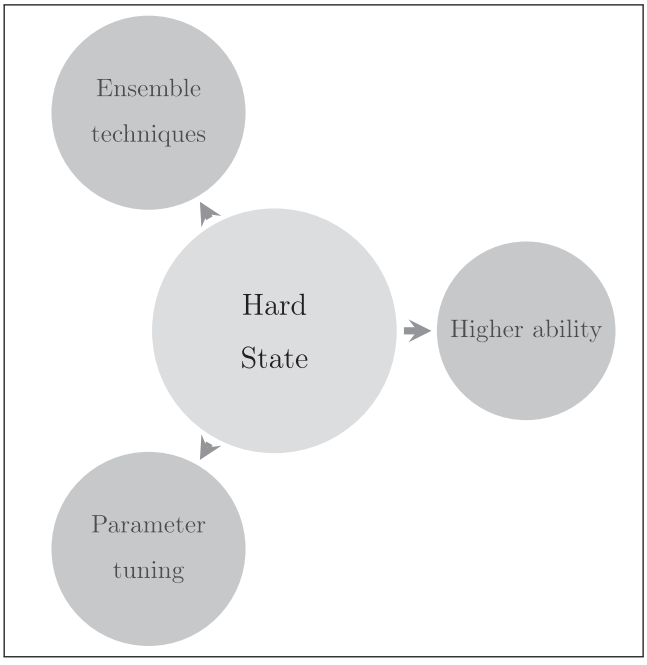

Fig. 7 – Hard state: considerable steps are undertaken in improving the data used for sensing by agents. With combination of previous state processing and adjustments the machine learning techniques themselves are applied.

A two level hybrid detection system is proposed in Guo et al. (2016) involving both anomaly and misuse detection. Traffic is divided via a clustering methodology that is then fed for further detection. Based on the clustering result, the instance are compared again either against misuse detection or anomaly detection through k-NN.

Ensemble approaches used in Aburomman and Reaz (2016) and Kuang et al. (2014) seek to increase state knowledge. A combination of SVM with Kernel Principle Component Analysis (KPCA) is used for optimum feature extraction, while SVM optimized with Genetic Algorithm (GA) and radial based kernel function (N-RBF) based on Gaussian kernel functions is used within Kuang et al. (2014). SVM and six k-NN classifiers are trained per attack class (Aburomman and Reaz, 2016) (i.e. Probe, DoS, etc.), with results from the classifiers then passed to three expert classifiers consisting of Particle Swarm Optimization (PSO) and Weighted Majority Voting (WMV), Local Udinmodal Sampling (LUS) Optimized PSO weights and WMV, and a single iteration of WMV. Within Veeramachaneni et al. (2016), a big data machine is used for anomaly detection by combination of unsupervised and supervised learning. Determination of unsupervised results is presented for feedback to a critic, which then are fed into a supervised module aiding in learning development. We have classified this agent featuring elements of both state and learning abilities through the use of a feedback look, represented in Table 2, yet we regard this deployment as predominately state based.

In drawing a bridge between ensemble methods and large volumes of training data, Huang et al. (2016) present a parallel ensemble learning method using OS-Extreme Machine Learning, *PEOS-ELM*. A MapReduce like method has been incorporated, supporting a mix of bagging, subspace partitioning and cross validation. Further parallel developments regarding EML are explored in Wang et al. (2016), presenting a Parallel-Regularized ELM (PR-ELM). PR-ELM trains nodes on data chunks and sub-models. Firstly, data are split into chunks and trained on a per-node basis that are then combined. Secondly, the model is then divided into subspaces where again a per-node instance takes place, finally combining by communicating data between nodes.

Orientated specifically toward Botnet detection, Al-Jarrah et al. (1796) employs a novel means in improving detection at the transport level. State representation is determined through a modified forwarding selection algorithm, while also a data reduction technique utilizes Voronoi based data partitioning. Random sampling of data is used in building ensemble DT classes.

Conversely, Ali et al. (2013) looks toward adaptive threshold tuning as a means to improve anomalous detection. Identified within anomalous detection system, conditions of hosts and traffic adjust overtime; observations are modeled with

Table 2 – Traditional summary: for traditional, anomaly detection features higher in agents that seek to detect unseen zero-day attacks.

Traditional	Detection			Attacks		Data		Agent					AI Algorithm(s)
	A	M	S	Common	Targeted	Known	Self	Reflex	State	Goal	Utility	Learning	
Rule													
Haque et al., 2012; Ji et al., 2016; Sonchack et al., 2015		√		√		√		√					KD SVM Naïve Bayes NN
Chen et al., 2009; Lin et al., 2012		√		√		√		√					DT Simulated Annealing
Soft State													
Fang and Liu, 2011	√	√			√		√		√				ENN
Medium State													
Ambusaidi et al., 2016; Creech and Jiang, 2012; Lin et al., 2015; Tsai and Lin, 2010; Xiang et al., 2014	√			√		√			√				LSSVM CANN
Singh et al., 2015													ELM OS-ELM
Sangkatsanee et al., 2011	√	√		√		√			√				DT
Hard State													
Aburomman and Reaz, 2016; Ali et al., 2013; Guo et al., 2016; Huang et al., 2016; Kuang et al., 2014; Song et al., 2013	√			√			√		√				KNN-SVM-PSO KPCA-GA-SVM ONE-CLASS SVM MARKOV-CHAIN PEOS-ELM
Veeramachaneni et al., 2016	√				√		√		√			√	-
Al-Jarrah et al., 1796	√				√	√			√				DT

A, Anomaly; M, Misuse; S, Specification; –, Unavailable.
The majority of datasets used feature common datasets in training and testing agents, as well as common attacks that are representative in the datasets. The presence of a tick represents the classification of that category matches the reviewed paper. In the absence of a tick, that paper reviewed does not meet this classification. Should no entry exist, the authors feel it did not contribute, or was not relevant to the context of this paper. The AI Algorithm(s) columns refer to the algorithms used within the papers reviewed.

Markov chains, which are used in a stochastic monitoring framework in adapting thresholds with real-time measurements. Furthermore, Song et al. (2013) identifies network characteristics that also require considerations. As ratios and cluster amounts may become irrelevant with changing environments, a method of parameter tuning is presented as attack to normal data ratios differ greatly, which is combined with one-class SVM.

3.1.6. *Impact*

Considering an IoT context, State and Reflex methods will pose little internal impact to increased numbers of perceptual layer devices and security. Their operations will remain consistent for current network deployments among domain perimeters. Of the two models, current Reflex methods can be considered for lower computation cost devices and environments, until state development for small form factor devices is improved. Stacking of reflex models, whereby optimized rules are tailored to a given sub-domain might provide extra defense following traditional modeling.

3.2. *Distributed defense mechanisms*

Distributed approaches to intruder detection rely on multiple agents for detection or processing on end devices. Agents operate within increasingly uncertain environments; i.e. Mobile Ad-hoc Networks (MANET) and Wireless Sensor Networks (WSN). Approaches consist of monitoring local devices clusters, scattering of agents throughout networks (as opposed to domain perimeters) or employing node collections for data or processing.

We categorize two levels of intelligence complexity based on our formal model: *goals* and *utility*. We insert *parallel* for discussion purposes in reviewing parallel approaches for techniques, data collection or processing. Specification based agents continue operation requirements while also monitoring the network state, the afforded rules of ML techniques allow an optimal state to be maintained through detecting malicious uncharacteristic behavior. We regard agents at a goal level where the decision process is passed to a cluster head or perhaps more traditional NIDS. Utility is granted when nodes employ decision-making themselves as a given metric prompts action over another and learning when greater autonomy is used for self-formulation without expert help. The defining elements for *Parallel*, *Goal* and *Utility* are: Parallel, central agent that employs mobile agents for computing and local data instances; Goal, which conducts detection action in fulfilling model state (report to central agent for decision process), Utility, where agent conducts detection and uses voting or associated metric for utility determination to reach model state (autonomous) and Learning, autonomous detection, action and development (Table 3).

3.2.1. *Parallel*

Parallel computing of ML allows advantages in processing larger amounts of data, with lower computational cost and time with parallel kernel methods (Díaz-Morales and Navia-Vázquez, 2016). Furthermore, approaches have been made in improving scalable for high data dimensions (Muja and Lowe, 2014)

Table 3 – Distributed summary: anomaly detection against defined profiles features prominently, with also an increase in targeted attacks outside of common datasets.

Distributed	Detection			Attacks		Data		Agent				AI Algorithm(s)
	A	M	S	Common	Targeted	Known	Self	Reflex	State	Goal	Utility	Learning	
Parallel													
Macia-Perez et al., 2011; O'Reilly et al., 2016	√			√		√		√					SOM-NN MVE-PCA
Goal													
Le et al., 2016			√		√		√			√			Self: RPL & Topology attacks detection
Paschalidis and Chen, 2010	√		√		√		√			√			Markov Models
Utility													
Seresht and Azmi, 2014	√		√	√		√					√		Immune network NS-CS-DTh
Sreelaja and Pai, 2014			√		√		√				√		ACO-AD
Lauf et al., 2010	√				√		√				√		MDS CCDS
Liu et al., 2012	√			√	√	√	√				√		Isolation Forest
Learning													
Igbe et al., 2016	√			√		√						√	Immune system NS-GA

A, Anomaly; M, Misuse; S, Specification.
Self-generated or acquired datasets also feature higher. The presence of a tick represents the classification of that category matches the reviewed paper. In the absence of a tick, that paper reviewed does not meet this classification. Should no entry exist, the authors feel it did not contribute, or was not relevant to the context of this paper. The AI Algorithm(s) columns refer to the algorithms used within the papers reviewed.

and multi-agent approaches in granting time to agents for better decision-making (Archibald et al., 2016) resulting in planning time improvements.

O'Reilly et al. (2016) introduce a distributed algorithm for anomaly detection based on PCA and soft-margin minimum volume ellipse, where analysis is shared across nodes. Macia-Perez et al. (2011) presents an embedded smart sensor like device within service-oriented architecture, where anomalous intrusions are detected and alerts may be shared with response and managements system. A Self-organizing Map (SOM) NN is used for TCP connection level attacks.

Performance and learning of semi-supervised, supervised classifier and feature space and online learning algorithms have been compared for attack detection in smart grid environments (Ozay et al., 2016). Further development in Smart Grids, namely, Advanced Metering Infrastructure (AMI), is shown in Ali and Al-Shaer (2015), fourth level Markov Chains are used to predict behavior given limited state space requirements. A randomization model is also presented in regards to node behavior, as fixed behavior patters may aid hiding attempts.

Improvements to scaling and associated throughput for pattern matching for NIDS using Ternary Content Addressable Memory (TCAM) lookups (Folino and Sabatino, 2016), where flows are partitioned and inspected in parallel. *Negative Patterns* are introduced as not to miss malicious patterns across flows. TCAM lookups are reduced through *Pattern Matching*, where flows are sub-tabled and only needed patterns matched.

3.2.2. *Goal*

Collecting information among peers and reporting back to a cluster head is used within Le et al. (2016), whereby cluster heads determine malicious agents given alerts reported from agents. Agents conduct detection through operating conditions classed against optimal desired goal states. States are determined from Routing Protocol for Low power and Lossy network (RPL) characteristics and employed within agents. Cluster heads determine malicious nodes based on solicited node information.

Inter-device requests are monitored to detect deviation in system behavior interactions within (Lauf et al., 2007). Behavior analysis results are categorized based on statistical features, represented in a list as values. Behaviors are generated based on semi-random Chi-squared probability density functions. Local and global maxima are used to determine malicious agents, with lower tolerance levels yielding improvements.

Markov models are used for statistical anomaly detection within a wireless sensor environment (Paschalidis and Chen, 2010). Chains are used to model behavior in comparing activity to known anomaly free behavior with transitions and deviations. First considering node level events of interest, secondly based on connection structures through tree indexing and thirdly, hierarchical state relating to parent/child, also by an index tree.

3.2.3. *Utility*

Lauf et al. (2010) proposes a decentralized anomaly detection of node behavior, namely application level interactions. Within an ad-hoc environment, identification is achieved through two detection methods, *Maximal Detection System* (MDS), which is then used to calibrate the *Cross-Correlation Detection System* (CCDS). Behavior determination is achieved through a Probability Density Function (PDF) of local and global maxima to help determine the state of the network. Application requests among devices are recorded within a history table to an associated integer; entries are ranked based on most appropriate behavior against defined offline behavior set, with CCDS constructing average nodes scores based on probability density function (PDF). Variance thresholds based on average score results in identification, providing an associated utility regarding operation. Adjustments were made to allow scalable function and threshold calibration.

An autonomous Multi-Agent Immune System, *MAIS-ID,S* is presented in Seresht and Azmi (2014). The hybrid approach seeks to determine system settings and traffic classification through the use of Immune Network, Negative Selection (NS), Clonal Selection (CS) and Danger Theory (DTh). Implemented within a virtual environment, agents seek to improve their detection goals in collaboration based on an agent's fitness utility. A review of immunity based intruder detection and context mappings can be found at Liu et al. (2011).

Permissive IDS nodes monitor for Blackhole attacks, in which once a given threshold is reached, IDS nodes broadcast to block deviant nodes in Su (2011). Monitoring nodes employ an *Anti-Blackhole Mechanism* in estimating potential anomalies in Route Request (RREQ) and Route Reply (RREP) packets.

Agents seek to detect Sinkhole attacks and intruders within WSN through the use of Ant Colony Optimization Attack Detection (ACO-AD) (Sreelaja and Pai, 2014). Each node is running as an IDS while also performing sensor operations; determined rules help identify nodes whose link quality does not match. Nodes coordinate privately in voting, in which after alert counts are processed and determined intruders are removed from the network.

Liu et al. (2012) model detection in perhaps what can be considered a traditional format, yet detection is based on isolation rather than search via density, distance or clustering. As normal data are considered to exist often, anomalous is then regarded to be rare; a binary tree structure is used to help identify anomalous points through sub-sampling of data, creating an ensemble of trees. An anomaly score is then associated given path lengths.

In combating nodes deliberating dropping packets, Sánchez-Casado et al. (2015) first modeled legitimate process when packets may be dropped in MANET environments. An anomaly based detection method is proposed through windowing detection based on MAC and network layer features selection. Two models are presented where either nodes work alone, or collectively in a distributed manner; of the two, the distributed mode is favored. Monitoring nodes collect information permissively within a given area in determining rouge nodes.

3.2.4. *Learning*

Distributed agents work collectively to identify attacks and share rules are presented in Igbe et al. (2016). An Artificial Immune System (AIS) approach is used whereby GA is used to generate valid detectors for NS. Each agent through NS generates corresponding rules determining intrusions based on self and non-self; rules are shared between agents and are given a utility based on weight and existence measures. Intercommunication

Table 4 – Comparison: individual or distributed decision-making is introduced through the agent scale, yet Utility-based agents represent those that process environment adjustments collectively in the reviewed works. Based on these works, we identify that predominately Parallel, Reflex and State based IDS works either made the decision themselves, while Goal and Utility methods collectively.

Agent	Decision point			Decision process		Transferable practice
	Collective	Distributed	Individual	Self	Collective	
Parallel	√	√		√		Distributed computing, data collection
Reflex	√			√		Prioritized rules
State	√			√		Individual focus
Goal	√	√			√	Collective working
Utility		√	√	√	√	Autonomous self-healing
Learning	√		√	√		Autonomous, adaptive

allows rules to be dispersed between agents. Simulations of common attacks show valid detector (rules) propagation among nodes.

3.2.5. *Impact*

Distributed methodologies have a high impact upon an IoT context. Research is linked to perceptual layer devices where new challenges are arising. The conducted works furthermore contain applicable knowledge and practices. They operate on resource-constrained devices and limited networks, parallel processing, distributed learning opposed to single source, are adaptive, collaborative, and autonomous.

3.3. *Discussion of IoT challenges*

The development of smart IoT defensive agents faces multiple challenges in breaking away from traditional detection cycles, and also with the introduction of manufacturing challenges that contrast both Traditional and Distributed approaches. In this section we present an updated defense cycle, increasingly more relevant for the challenges ahead as a starting point for IoT agents. Elements of this cycle are mapped to the OWASP IoT vulnerabilities project (OWASP Foundation, 2017). Furthermore, we consider manufacturing issues through direct and indirect challenges that may pose a risk to best enabling secure agents. It should be noted that we exclude parallel deployments from our discussion (Table 4).

3.3.1. *Defense cycle*

We previously determined that two sides exist within a cycle in approaches to IDS agent deployments or research as seen in Fig. 4. The right side of the perforated red line, or sensing, represents a general approach, whereby monitoring of the network is followed by feature extraction of information, which in turn is aggregated to determine detection, finally repeating. This cycle can also been seen in Fig. 8, in the left topmost loop, identified through black and red lines. Actuating, seen on the left-hand side of Fig. 4, is where the introduction of autonomous processes based on security policy requirements in enacted after the sensing processes. This extended cycle from sensing is represented through the perforated blue lines in Fig. 8, as the outside right-hand loop. It can be seen then that a complete loop of Fig. 4 follows the steps of monitoring and detection, then the blue loop through the red loop, returning to

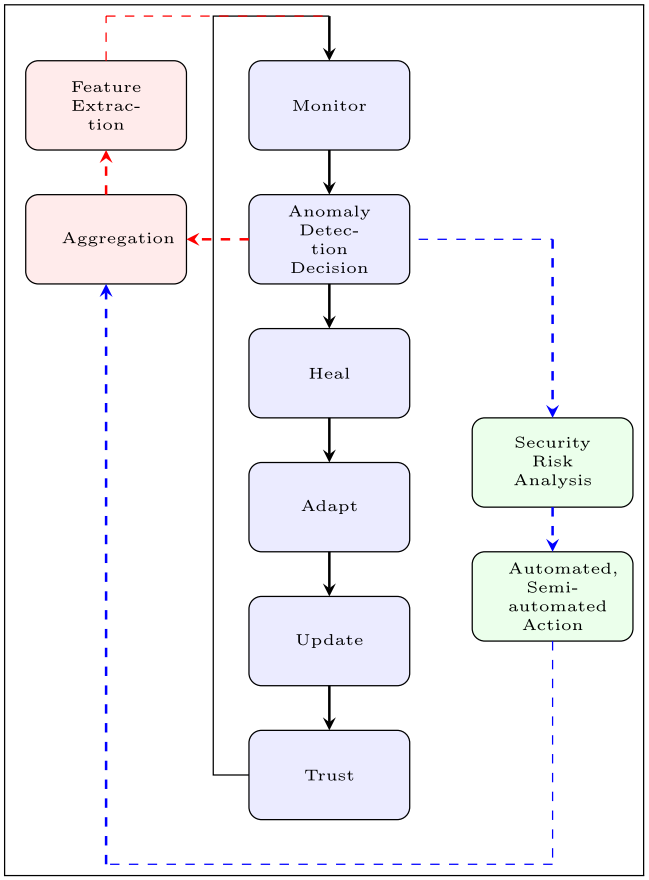

Fig. 8 – Agent defense cycle transitions: a suggested starting point for an appropriate defense cycle as expanded from Fig. 4 and the considerations of research approaches from previous surveys. I identifies sensing elements, while ii key actuating stages. These stages are drawn out to a new cycle presented for iii. This loop provides the necessary steps an agent should consider in maintaining a collective environment harmony and conduct actions more autonomously. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

Table 5 – Defense cycle elements: respective of their defense cycle position, the most appropriate OWASP elements were mapped.

Defense cycle	OWASP elements	
	Traditional	Distributed
Agents monitoring	Device Web, Interface Device, Network Service, Admin Interfaces, Cloud Web Interface, Mobile Application, Network Traffic	Ecosystem Communication, Device Physical Interfaces, Device Web Interface, Device Firmware, Device Network Service, Admin Interfaces, Cloud Web Interface, Mobile Application, Network Traffic
Agents healing	Expert help	Hardware, Device firmware, Ecosystem
Agent's trust	Expert determination	Ecosystem, access control, Update mechanism

monitoring. The inner black-lined loop is presented as the initial starting point as a defense cycle for IoT defensive agents. It encompasses the complete loop of Fig. 4, or the red outer top and blue lower left loop in Fig. 8, its consideration alone warrants the dismissal of the outside loop as it encompasses both. Elements are drawn out into new stages as for agents to seek to autonomously enact to enable a collective harmony and effective operational status. The formulation of this new cycle is representative of the desire to enable agents to conduct actions more autonomously.

The first requirement of an agent is Monitoring of network patterns, traffic and devices. Detection within an IoT environment could now be considered a collective initiative, as deployments can be assumed to dramatically increase the amount of network, and perception layer networks. Detection systems must focus not only on traffic, but also surrounding devices within a given network subset. Traditional monitoring of the environment has been conducted through a single point, which may be hierarchically positioned, while rules and signatures are used to achieve detection classification and consist of reflexive and state based means. Distributed employs multiple locations and combination of traffic and devices monitoring, where agents may be distributed among the environment or encapsulated within each node. Goal, utility, and learning functionality are seen within these types of deployments. Anomaly behavior is determined through comparison against defined profiles. Agent states keep track of behavioral patterns and not classification of traffic in the majority of cases. Both approaches are successful in monitoring their environments. If we consider the IoT monitoring attack elements as shown in Table 5, we see that distributed agents cast a wider range of elements to be monitored. Traditional agents are successful for attack areas that are relevant to their current format, but communication and internal specific items are greatly more supported through distributed agents.

As discussed, distinguishing between an attack or normal traffic is achieved traditionally through either known signatures, reflexive rules, or state training, while distributed approaches consist of behavioral comparison to profiles, while holding limited state inclusion for anomalous traffic. Once analysis has been conducted, agents must focus through a decision point and process. Traditional agents are both central and individual in their location, typically agents make their decisions generally close to a network gateway or even offline. Distributed agents work collectively and individually in reaching a decision, where the process is determined by maintaining goals of network operations/harmony. Goal-only based agents pass requested state memory relating to goals to other agents (base station), while utility and learning based agents reach self or collective determination through voting based on utility thresholds. The process leading to healing can been seen as a by-product of defined risk analysis in determining which defenses mechanisms are required for agents.

Healing of the system concerns removing nodes or devices that may have been compromised through attack or physical environment adjustments. Goal and utility policies allow agents to enact operations that maintain the health of the environment. Many variances have been suggested, particularly in relation for energy efficiency within IoT devices (Akgül and Canberk, 2016), but self-healing is the process whereby agents recover from detected faulty states (Schneider et al., 2015). Healing provides a way for the system to change in response to the environment, Traditional Agents will typically provide generation alerts, but refrain from intervention until expert intervention is conducted in regards to a new environment. There is no direct effect of an agent to heal itself or environment. Distributed decisions are those that have been determined from shared analysis, healing is conducted by removing any rouge nodes or physically damaged nodes from the system, and the overall system participation is updated. Utility agent enact decisions themselves, while goal based relate to the cluster base station, which in turn relays acceptable nodes in the environment after processing. Contrasting the two agent models with respect to OWASP attacks seen in Table 5, Distributed agents are not only able to counter hardware and firmware threats through monitoring them, but also to enable the system to heal its ecosystem by removing agents that are compromised via those methods. Traditional agents require expert intervention in retraining agents that act maliciously or that do not meet expectation. Expert determination is required for agents of Traditional methods, where security professionals undertake healing considerations based on altered intrusions or monitoring of devices.

This change to the system requires Adaptation of the environment in response; participating nodes redefine roles and ability, determining new constraints, goals, utility, etc. in Distributed, while expert intervention is required for an environmental change in Traditional. Distributed presents far greater functionality through cloning and reassigning of roles. Once the system and network have re-aligned, agents are required to Update knowledge, rules, goals, utility functions and information among the network. Traditional agents require the help of experts to retrain state and adjust rules, though maybe automatic, agents do not act autonomously in doing so.

Furthermore, Distributed shows instances of rules propagation based on fitness. Large amounts of data will be generated, such that refinements need to be distributed that may aid other agents. Again, Traditional agents are not tasked with these sort of requirements, but rely on security experts to conduct these steps as they are oblivious to any changes in the environment.

Once the previous steps are achieved, agents should agree on a state of Trust through consensus, and may begin to monitor the environment again. A trust consensus guarantees ecosystem access control vulnerabilities as collectively all nodes are agreeing on all other nodes and update mechanism processes are furthermore validated. Trust may be reputation based, threshold mechanisms or simply implicit after environmental changes. Table 5 shows that trust ultimately resides with security expert for Traditional agents. While update mechanisms are validated, ecosystem access is completed as the network returns to an optimized state for Distributed agents.

3.3.2. *Security manufacturing challenges*

Security challenges do not solely exist from bad actors, agents face direct and indirect security challenges from IoT device and system manufacturing. The OWASP IoT Security Challenges (OWASP Internet of Things Project) are shown and agents are contrasted in which direct and indirect challenges they could potentially counter.

I1: Insecure Web Interface
I2: Insufficient Authentication/Authorization
I3: Insecure Network Services
I4: Lack of Transport Encryption
I5: Privacy Concerns
I6: Insecure Cloud Interface
I7: Insecure Mobile Interface
I8: Insufficient Security Configurability
I9: Insecure Software/Firmware
I10: Poor Physical Security

As identified in Fig. 9, direct challenges that influence Traditional and Distributed agents are positioned within their respective region.

Indirect challenges are identified within the intersection of Traditional and Distributed regions. Indirect pose serious operating threats for agents in introducing unnecessary security risks that may allow further avenues of exploitation. Similarities between the two approaches with direct challenges shows agents are very much aligned, highlighting that an agent work is more influenced than influential between the manufacturing challenges.

3.3.3. *Defense case study comparison*

An agent's ability to detect and control an attack and its environment is reflective of its ability to complete its cycle of defense. The scenario we consider is a collection of perception layer nodes misbehaving and their interaction with a controling agent of a given network segment. The effect that compromise has on nodes results in draining power, increasing the computational output until processes or memory is spent and an encrypted connection is established to a remote host not defined within security policy.

Our previous household scenario suggested the use of smart devices like a mirror, kettle and a bed, lets consider these devices compromised and acting by control of an external host of malicious nature. Malicious code was previously embedded within news and weather information retrieved by the mirror. The collection of the compromised nodes begins to solicit new routing information within the network to gain control of the flow of traffic being routed. Routing and link costs are broadcast to nodes, which alter tables and environment dynamics. Once a new path has been established, an encrypted transmission is established from the mirror.

Compromised sensors embedded in the household devices are used to adjust the routing table to secure a path to the mirror and bypass the cluster head so the mirror is the entry and exit point for the attacker. This adjustment removes a segment detection head with the mirror directly connecting to the gateway. Account information for each device is collected, time periods of no occupants are aggregated, malicious code is distributed which may combine sensors to a larger botnet or ransom the nodes.

The initial detection of malicious code will bypass defense regardless if it traveled openly or securely in mirror transmissions as new attacks bypass rules and signatures. Once the nodes are compromised, router and firewall have an internal device establishing outbound connections, thus the return traffic will be permitted.

How do our agents handle the compromised network? Reflex and State agents are not afforded the ability to adjust the environment and allow the compromised nodes to continue until manual intervention is required. Goal, Utility and Learning agents can monitor the change in routing, the new communication commencing from the mirror, exclusion of the cluster head node. Action can proceed in determining

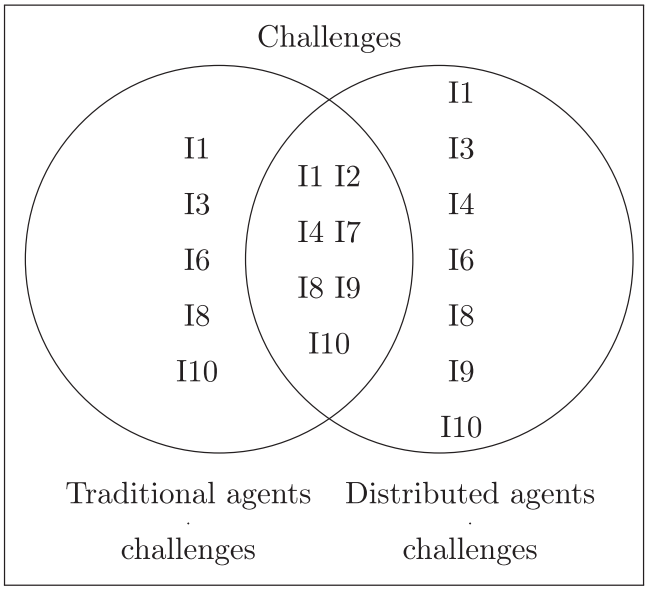

Fig. 9 – Manufacturing challenges: agent design and ability, not only IDS, are directly and indirectly influenced by other agents manufacturing. The union of each shows the shared indirect challenges faced by agents, while each holds their own direct manufacturing challenges.

which nodes are compromised through rational action, determine if the new connection is anomalous, removing malicious nodes and resetting the environment to a trusted state. Overall we can understand that agents working collectively between their own level and interacting with the others possess the ability to better determine intrusions than a single agent classifying attempts.

4. Discussion

For our discussion, we fist look at the current approaches that stand out against what we perceive to be challenges for the approaching IoT. Considerations to security policy, current challenges to state based agents, the data and attacks used in such approaches and future challenges are discussed.

4.1. *Intelligent approaches*

Agents are asked to handle significant tasks and challenges in helping secure our environments. Of these approaches, several methodologies exist that aid and help the development in reaching autonomous systems.

4.1.1. *Transferable attack defense*

Looking at the techniques used to determine detection, it could be suggested that more could be obtained from agents given the right operational configurations in coordination and sharing of knowledge.

4.1.1.1. Perception layer. From the collection of goal, utility and learning agents, those that reflect biology or real-world processes currently reflect intelligent approaches to agents (Igbe et al., 2016; Seresht and Azmi, 2014; Sreelaja and Pai, 2014). The network is regarded as itself a living system that requires cooperation in not only defending itself, but healing and updating. Specification agents show effectiveness to combat perception layer issues of routing, behavioral or system interference though AIS, GA, ACO and expert specification based algorithms created based on requirements.

4.1.1.2. Network and application. Agents comprising of state methods reflect SVM, *k*-NN, DT and ELM feature prominently, but ultimately, results rely on the afforded training and their ability to achieve correct classification. The goal of a state agent is to ultimately relieve experts of the requirements to define incidents of interest and strengthen network access without complex and time consuming processes. Training of machine states has employed a variety of ML techniques and optimization, with a large potion of the work concerned with clustering data to determine anomalies. Despite the benefits acquired through improved methodology of processing data for analysis, much still remains on the ability to update or alter state memory. EML is yielding a method in which processing time remains relatively low and with high accuracy (Creech and Jiang, 2012; Huang et al., 2016). The development and evolution of network characteristics are continually evolving in conjunction with attacking methodologies, state based representation needs to react in time with attacks however. Not mentioned within our comparison discussion, parallel reduced training yields application to distribute state baselines efficiently (Wang et al., 2016). Besides the other mentioned classifiers, ELM presents opportunity based on its ability to adjust baselines effectively and the time taken to train the classifier is considerably low.

As with real-world deployments, rule instances already currently hold weight with their ability to help mitigate bad actors. Rule optimization based on environmental requirements through ML and collaboration whereby tacking of devices (Chen et al., 2009; Sonchack et al., 2015), each individually optimized present promise.

4.1.2. *Key security policies*

Several considerations for not only agent development, but to the wider ecosystem (OWASP Internet of Things Project).

1. Assuming hostility and the worst: The quest for a perfect agent and foolproof security is likely to come undone and result in bad actors gaining control of devices, particularly edge devices that allow physical access. Agents are required to be combat malicious actors, events, and compromised systems. To achieve this, naturally inspired algorithms from the reviewed literature, the adaptation and healing qualities allow systems to adequately adapt in the face of these challenges.
2. Internet of lies: Requests and information coming from supporting applications and other process data need to be verified as systems may operate, and may not appear compromised. The ability to audit data and sources will help strengthen trust within the system as a whole. Furthermore, aggregation or distributed coordination in verifying situational occurrences may help to reduce falsehoods.
3. Life cycle: Design of systems, deployments and approaches to agents need to incorporate the adding and removing of components, while also providing a means to re-purpose those ones currently in use. This also requires cooperation from healing and adaptive processes in the detection cycle. These measures allow system to develop without hindrance.
4. Targeting weakness: Any perceivable weakness will be attacked which may compromise the entire system. All nodes are required to contribute to help keeping the environment state healthy. Policy and security risk analysis must be considered in association with agent development.
5. Hardening: Possible physical and software vulnerabilities need to be removed through system hardening. Closing of unused ports, services or any physical readers should be conducted.

4.2. *Limitations and future challenges*

4.2.1. *Application*

From the reviewed literature, we can see that much has been given to the quest for anomaly detection, yet still little remains functional within live environments due to their complexity and false alarm rate (Sommer and Paxson, 2010). However, a real-world implementation will not happen without the developments already conducted. It could be suggested that the

overall scope of a one-stop-shop of anomaly detection is not delivering what it is intended to do, and perhaps a more single-minded focus is required. Reflexive agents already benefit from selective rules, perhaps, the same should applied from state instance agents. A focus should shift to more local requirements based on expectations, employed services and critical infrastructure requirements. Improvements to classification flows are developing overtime (Jun et al., 2013), but the gap still remains for IDS systems. Furthermore, goal and utility agents are seen as perhaps a by-product of healing and management in wireless or mobile networks, these agents are afforded intelligence to counter attacks that disrupt the network, but may impact the environment also at the same time in disrupting application performance and the associated goals and utility, thus computational and power indifferences are introduced within the system, potentially degenerating overall output, agent life or node performance. Learning agents that undertake problem generation for development learning as defined within the formal model stand to receive little application in live environments. The idea of problem generation may be too costly for critical infrastructure where compromise is avoided at all cost. Though Igbe et al. (2016) make decisions based on utility and learning, they do not actively seek for defining new rules of mechanisms that may work based on problem generation.

Security methodologies and practices stand to benefit from thorough assertion of agent challenges at a given network segment. Currently agents compete with an overall risk cost a network may face, which unfairly positions IDS deployments as costly and time consuming deployments. Risk cost identification at a given network segment should reflect afforded agent intelligence positioned within the region. An example could be agents that seek to determine DoS attacks entering and exiting the perception layer. Their state based training and determined rules can be specialized with greater focus for increased accuracy of detection. Until a truly autonomous agent is developed, defense stands to benefit from specialized approaches respective of their segment.

4.2.2. *Attacks*

Ultimately agents are developed to detect attacks that are known or anomalous or do not fit the prescribed behavior. These attacks fall within one of three categories; those that exist within the research only due to belonging to a common dataset, or a given targeted attack of purpose, i.e., Al-Jarrah et al. (1796), Sreelaja and Pai (2014), or those that reflect potential challenges of a given environment. Of the three, attacks within common dataset are widely considered outdated and easily mitigated by today's standards, and actively do not reflect current and future challenges. Primarily these attacks are tested against state based agents whose aims are to counter and determine all possible threats. This is not to say that these challenges are not to be faced, but that agent development is behind the point where current attack evolution sits.

Perception level agents face threats surrounding routing, authentication, physical damage, DoS, and node capture; network layer agents challenges reflect those currently faced, sniffing, Man-in-the-Middle, viruses and compatibility issues; application layer threats furthermore surround permission, identity, assets and data confidentiality and software vulnerabilities (Gou et al., 2013; Roman et al., 2013; Zhao and Ge, 2013). Exacerbating these challenges will be the sheer increase of data and communication as agents are not only concerned with their respectively level, but also in conjunction with the layer above or below.

Zero-day, Social Engineering and Malware threats pose an increased security challenge; not only for increasing the amount of potentially compromised devices, but the risk to privacy, disruption or corruption of systems and the ability to launch widespread attacks posing as legitimate traffic (Atwell et al., 2016).

Device infection and compromise through malware pose significant additional challenges associated to time and cost. Potential ransomware attacks may render a large quantity of nodes useless unless financial or exhaustive manual intervention is conducted to overturn encrypted file systems. As with the evolution of network and application attacks, ransomeware has developed an ability to blend in with normal behavior of IoT communication, with various packages now making use of XMPP protocol to communicate with controling sources (Desai et al., 2015; Monika et al., 2016).

4.2.3. *State and data*

The prescribed internal state of an agent required and the dataset used in doing so go hand in hand in determining the results even before beginning. As discussed, an optimal state allows better classification (Bengio et al., 2013), with the training of classifiers walking a thin line, whereby stray too far from the profile through the introduction of noise within the data, and the detection results may suffer. Attempts have been made in developing a more systematic approach to the use of training and testing data, but several hindrance of data acquisitions exist, namely privacy, completeness and the variety of current attacks (Shiravi et al., 2012). Given these drawbacks, agent state representation will always be restricted unless other methodologies are developed.

Indeed, the current data used for state based training is outdated for current attack and network characteristics, the acquisition of training data for IoT modeling faces several challenges. No formal dataset actively reflects the potential challenges that maybe faced.

Defining a dataset reflective raises several important issues. Firstly how are the data acquired, and what can be concluded as an adequate sample? As each domain is unique, could the set be generalized in reflecting a blanket overview of all typical deployments, or are environmental factors too large to misrepresent within the data? Furthermore how can we adequately label attacks, do attacks propose challenges for one type of deployment yet not another? And how are new attacks identified and introduced? These issues pose great challenges combined already with current datasets.

4.2.4. *Goals, utility and learning*

Associating goals, utility and learning methods remain narrowly focused and simply operated to explicitly remove offending nodes. The impact may drastically reduce performance within the environment, driving up the utility required in not only achieving detection goals, but operational duties. Nodes that are excluded may be only selectively compromised, and retain some functionality that maybe harnessed for use. Furthermore, as discussed, when compared against our

formal model, the issues of problem generation provide a risk that might be considered too great. The sensitive nature of system, privacy, secrets or financial implications pose too great a threat to allow agent to trial and adapt acceptable risk however. Alternatively, after a utility decision, rouge nodes could be kept to induce learning to other nodes, participating within a quarantined zone in aiding learning development.

Intelligence should not only seek to heal, but to also learn and re-align purposeful goals and utility functions.

4.2.5. *Distributed data*

Considerations are required for the data generated by agents and the format that it can be shared between participating nodes. With devices coming in and out of the environment, mechanisms for granting and removing data are required so that they may be irrefutable and their integrity holds true. Access and permission considerations also need to allow agents to complete their required tasks, but not leak any privacy issues.

4.3. *Moving forward*

Agents should reflect the ideas of an IoT environment, sharing data, distributed processing, and coordinating collectively. The detection of one layer's issues is not unique to that layer only, but should concern supporting layers. Agents need to be able to identify, determine, mitigate and heal, adjust, distribute knowledge and ultimately trust one another. Trust within the system is pivotal for human acceptance and interaction; the afforded potential also means sacrifice of significant personal privacy that currently is already an area of concern in the current form of the Internet. Agents must autonomously develop to mitigate threats and attacks, yet provide a means of audit should compromise be detected of the entire system.

Models that employ reflexive techniques are benefited by rule optimization, either by collaboration or ML techniques. Security analysis should be conducted to shape rules appropriate to a given environment. State, Goal, Utility and Learning models for the time being until and complete agent exists, need to be individualized based on policy or environment requirements and need to incorporate elements of each model.

Of the approaches discussed aligned to the presented formal model, Distributed multi-agent approaches provided the closest real-world solution. In undertaking the next step to an Internet of Things, where the system is a collective intelligence, and should not fault at a single point, cooperative agents are envisioned for application by 2020, putting older methodologies behind us.

Overall, if we consider the application of data, with respect to attacks, a striking imbalance exists in achieving results, but also the level of agent functionality. If we imagine a drone with a loss of signal regarding coordinate positioning, several implications occur. It can be assumed that in reaction to this loss of signal, our drone may hover in the same location until a connection is re-established, it either tries to land or expends all its battery, let's consider this interruption has occurred through malicious intent. If we consider traditional, non-distributed approaches, the drone simply waits, expending its utility, wasting resources and potentially suffering damage; the agent's functionality has also been reduced, moving from a potential utility-based agent to a reflexive agent. If there is a series of drones working together in the same scenario, again, the effected drone, or drones are rendered useless, unable to improve their situation. Yet from a distributed perspective making use of coordination and aggregation, the drone may inquire with peers to the newly updated coordinates and come to realize through higher reasoning it has been compromised and to continue with its peer's information. This scenario makes a lot of assumptions, but the key point remains that, traditional approaches can fall prey to simple issues through isolation, a lack of coordination and setting a limit in its ability to reason. This is not to say that distributed agents are immuned, but rather a distributed model enables higher level reasoning through the aggregation of data within the network. If we reapply this scenario to our defensive agents, a standalone IDS system might not know anything is wrong, yet perhaps, a distributed IDS may have enough intelligence to reason that it is under attack. In this scenario we can see that data hold more promise for intelligent behavior from the statistics or patterns we can draw from it, its characteristics. Intrusion detection is concerned with forces on its outside, AI internally, the IoT, everywhere; the common link here are the data, and their characteristics mean something different to each element, the model that can best harness these data driven cyber security remains our distributed approach for the future of the IoT.

5. Conclusions

The challenges for intrusion detection are numerous not only in their current format, but future security challenges redefine what we associate with detection models. The inclusion of everyday objects connecting to network applications opens further vectors for attacks and exploitation. Considering the work conducted thus far, challenges still, and potentially will always exist in developing methods to secure networks.

The development of IoT systems requires flexible and adaptive agents to not only operate but defend. These systems need to be able to adjust, heal and promote trust through autonomous action.

Deployments still rely to reflexive rules and signature monitoring to help defend and alter against intrusion detection. To better harness the use of machine learning, *Reflex* and *State* may benefit from more individualized approach in rule construction and state training. Given the potential vast complexity associated with the IoT, optimized agents that are scattered and aligned to security analysis can focus on a given segment better and its security cost. *Goal*, *Utility* and *Learning* agents have seen to benefit from natural, nature inspired approaches, and are strengthened through security analysis in determining specification requirements.

Regardless of the approach chosen, it is evidential that defense is not layer specific, rather a collective effort in which coordination and knowledge sharing are required.

REFERENCES

Aburomman AA, Reaz MBI. A novel SVM-kNN-PSO ensemble method for intrusion detection system. Appl Soft Comput 2016;38:360–72.

Akgül OU, Canberk B. Self-organized things (SoT): an energy efficient next generation network management. Comput Commun 2016;74:52–62. Current and Future Architectures, Protocols, and Services for the Internet of Things.
Al-Jarrah OY, Alhussein O, Yoo PD, Muhaidat S, Taha K, Kim K. Data randomization and cluster-based partitioning for botnet intrusion detection. IEEE Trans Cybernet 1796;46(8):2016.
Ali MQ, Al-Shaer E. Randomization-based intrusion detection system for advanced metering infrastructure. ACM Trans Inf Syst Secur 2015;18(2):7:1–30.
Ali MQ, Al-Shaer E, Khan H, Khayam SA. Automated anomaly detector adaptation using adaptive threshold tuning. ACM Trans Inf Syst Secur 2013;15(4):17:1–30.
Alsmadi I, Xu D. Security of software defined networks: a survey. Comput Secur 2015;53:79–108.
Ambusaidi MA, He X, Nanda P, Tan Z. Building an intrusion detection system using a filter-based feature selection algorithm. IEEE Trans Comput 2016;65(10):2986–98.
Archibald C, Altman A, Shoham Y. A distributed agent for computational pool. IEEE Trans Comput Intell AI Games 2016;8(2):190–202.
Armanfard N, Reilly JP, Komeili M. Local feature selection for data classification. IEEE Trans Pattern Anal Mach Intell 2016;38(6):1217–27.
Atwell C, Blasi T, Hayajneh T. Reverse TCP and social engineering attacks in the era of big data. In: 2016 IEEE 2nd international conference on Big Data Security on Cloud (BigDataSecurity), IEEE international conference on High Performance and Smart Computing (HPSC), and IEEE international conference on Intelligent Data and Security (IDS). 2016. p. 90–5.
Balaji PG, Srinivasan D. Multi-agent system in urban traffic signal control. IEEE Comput Intell Mag 2010;5(4):43–51.
Bengio Y, Courville A, Vincent P. Representation learning: a review and new perspectives. IEEE Trans Pattern Anal Mach Intell 2013;35(8):1798–828.
Chen H, Summerville DH, Chen Y. Two-stage decomposition of snort rules towards efficient hardware implementation. In: Design of Reliable Communication Networks, 2009. DRCN 2009. 7th international workshop on. 2009. p. 359–66.
Creech G, Jiang F. The application of extreme learning machines to the network intrusion detection problem. In: Numerical analysis and applied mathematics ICNAAM 2012: International Conference of Numerical Analysis and Applied Mathematics, vol. 1479. AIP Publishing; 2012. p. 1506–11.
Desai P, Sheth A, Anantharam P. Semantic gateway as a service architecture for IoT interoperability. In: 2015 IEEE international conference on mobile services. 2015. p. 313–19.
Díaz-Morales R, Navia-Vázquez A. Efficient parallel implementation of kernel methods. Neurocomputing 2016;191:175–86.
Fang X, Liu L. Integrating artificial intelligence into Snort IDS. In: Intelligent Systems and Applications (ISA), 2011 3rd international workshop on. 2011. p. 1–4.
Folino G, Sabatino P. Ensemble based collaborative and distributed intrusion detection systems: a survey. J Netw Comput Appl 2016;66:1–16.
Galindo C, Fernandez-Madrigal JA, Gonzalez J. Improving efficiency in mobile robot task planning through world abstraction. IEEE Trans Robot 2004;20(4):677–90.
Gou Q, Yan L, Liu Y, Li Y. Construction and strategies in IoT security system. In: Green Computing and Communications (GreenCom), 2013 IEEE and Internet of Things (iThings/CPSCom), IEEE International conference on and IEEE Cyber, Physical and Social Computing. 2013. p. 1129–32.
Guo C, Ping Y, Liu N, Luo S-S. A two-level hybrid approach for intrusion detection. Neurocomputing 2016;214:391–400.
Haque MJ, Magld KW, Hundewale N. An intelligent approach for intrusion detection based on data mining techniques. In: Multimedia Computing and Systems (ICMCS), 2012 International Conference on. 2012. p. 12–16.
Hsu R-L, Abdel-Mottaleb M, Jain AK. Face detection in color images. IEEE Trans Pattern Anal Mach Intell 2002;24(5):696–706.
Huang S, Wang B, Qiu J, Yao J, Wang G, Yu G. Parallel ensemble of online sequential extreme learning machine based on MapReduce. Neurocomputing 2016;174(Pt A):352–67.
Igbe O, Darwish I, Saadawi T. Distributed network intrusion detection systems: an artificial immune system approach. In: 2016 IEEE First International Conference on Connected Health: Applications, Systems and Engineering Technologies (CHASE). 2016. p. 101–6.
Ji S-Y, Jeong B-K, Choi S, Jeong DH. A multi-level intrusion detection method for abnormal network behaviors. J Netw Comput Appl 2016;62:9–17.
Jun Z, Yang X, Yu W, Wanlei Z, Yong X, Yong G. Network traffic classification using correlation information. IEEE Trans Parallel Distrib Syst 2013;1:104.
Keung GY, Li B, Zhang Q. The intrusion detection in mobile sensor network. IEEE/ACM Trans Netw 2012;20(4):1152–61.
Khan R, Khan SU, Zaheer R, Khan S. Future internet: the internet of things architecture, possible applications and key challenges. In: 2012 10th international conference on frontiers of information technology. 2012. p. 257.
Kuang F, Xu W, Zhang S. A novel hybrid {KPCA} and {SVM} with {GA} model for intrusion detection. Appl Soft Comput 2014;18:178–84.
Lauf AP, Peters RA, Robinson WH. Embedded intelligent intrusion detection: a behavior-based approach. In: Advanced Information Networking and Applications Workshops, 2007, AINAW '07. 21st International Conference on, vol. 1. 2007. p. 816–21.
Lauf AP, Peters RA, Robinson WH. A distributed intrusion detection system for resource-constrained devices in ad-hoc networks. Ad Hoc Netw 2010;8(3):253–66.
Le A, Loo J, Chai KK, Aiash M. A specification-based IDS for detecting attacks on RPL-based network topology. Information 2016;7(2):25.
Liao H-J, Lin C-HR, Lin Y-C, Tung K-Y. Intrusion detection system: a comprehensive review. J Netw Comput Appl 2013;36(1):16–24.
Lin S-W, Ying K-C, Lee C-Y, Lee Z-J. An intelligent algorithm with feature selection and decision rules applied to anomaly intrusion detection. Appl Soft Comput 2012;12(10):3285–90.
Lin W-C, Ke S-W, Tsai C-F. CANN: an intrusion detection system based on combining cluster centers and nearest neighbors. Knowl Based Syst 2015;78:13–21.
Liu C, Yang J, Chen R, Zhang Y, Zeng J. Research on immunity-based intrusion detection technology for the internet of things. In: Natural Computation (ICNC), 2011 seventh international conference on, vol. 1. 2011. p. 212–16.
Liu FT, Ting KM, Zhou Z-H. Isolation-based anomaly detection. ACM Trans Knowl Discov Data 2012;6(1):3:1–39.
Lu WL, Wang YS, Lin WC. Chess evolution visualization. IEEE Trans Vis Comput Graph 2014;20(5):702–13.
Lugo-Cordero HM, Guha RK. What defines an intruder? An intelligent approach. In: Computational Intelligence in Cyber Security (CICS), 2013 IEEE symposium on. 2013. p. 31–6.
Macia-Perez F, Mora-Gimeno FJ, Marcos-Jorquera D, Gil-Martinez-Abarca JA, Ramos-Morillo H, Lorenzo-Fonseca I. Network intrusion detection system embedded on a smart sensor. IEEE Trans Indust Electron 2011;58(3):722–32.

Modi C, Patel D, Borisaniya B, Patel H, Patel A, Rajarajan M. A survey of intrusion detection techniques in cloud. J Netw Comput Appl 2013;36(1):42–57.
Mohapatra SK, Sahoo PK, Wu S-L. Big data analytic architecture for intruder detection in heterogeneous wireless sensor networks. J Netw Comput Appl 2016;66:236–49.
Monika MM, Zavarsky P, Lindskog D. Experimental analysis of ransomware on windows and Android platforms: evolution and characterization. Proc Comput Sci 2016;94:465–72. The 11th International Conference on Future Networks and Communications (FNC 2016)/The 13th International Conference on Mobile Systems and Pervasive Computing (MobiSPC 2016)/Affiliated Workshops.
Muja M, Lowe DG. Scalable nearest neighbor algorithms for high dimensional data. IEEE Trans Pattern Anal Mach Intell 2014;36(11):2227–40.
O'Reilly C, Gluhak A, Imran MA. Distributed anomaly detection using minimum volume elliptical principal component analysis. IEEE Trans Knowl Data Eng 2016;28(9):2320–33.
OWASP Foundation. OWASP Internet of Things Project; 2017. Available from: https://www.owasp.org/index.php/OWASP_Internet_of_Things_Project. [Accessed 25 October 2016].
Ozay M, Esnaola I, Vural FTY, Kulkarni SR, Poor HV. Machine learning methods for attack detection in the smart grid. IEEE Trans Neural Netw Learn Syst 2016;27(8):1773–86.
Paschalidis IC, Chen Y. Statistical anomaly detection with sensor networks. ACM Trans Sen Netw 2010;7(2):17:1–23.
Patel A, Taghavi M, Bakhtiyari K, Celestino J Jünior. An intrusion detection and prevention system in cloud computing: a systematic review. J Netw Comput Appl 2013;36(1):25–41.
Perera C, Zaslavsky A, Christen P, Georgakopoulos D. Context aware computing for the internet of things: a survey. IEEE Commun Surv Tutor 2014;16(1):414–54.
Rivera D, Cruz-Piris L, Lopez-Civera G, de la Hoz E, Marsa-Maestre I. Applying a unified access control for IoT-based intelligent agent systems. In: 2015 IEEE 8th international conference on Service-Oriented Computing and Applications (SOCA). 2015. p. 247–51.
Roman R, Zhou J, Lopez J. On the features and challenges of security and privacy in distributed internet of things. Comput Netw 2013;57(10):2266–79. Towards a Science of Cyber SecuritySecurity and Identity Architecture for the Future Internet.
Russell SJ, Norvig P, Canny JF, Malik JM, Edwards DD. Artificial intelligence: a modern approach, vol. 2. Upper Saddle River: Prentice Hall; 2003.
Sánchez-Casado L, Maciá-Fernández G, García-Teodoro P, Magán-Carrión R. A model of data forwarding in MANETs for lightweight detection of malicious packet dropping. Comput Netw 2015;87:44–58.
Sangkatsanee P, Wattanapongsakorn N, Charnsripinyo C. Practical real-time intrusion detection using machine learning approaches. Comput Commun 2011;34(18):2227–35.
Schneider C, Barker A, Dobson S. A survey of self-healing systems frameworks. Softw Pract Exp 2015;45(10):1375–98.
Seresht NA, Azmi R. MAIS-IDS: a distributed intrusion detection system using multi-agent {AIS} approach. Eng Appl Artif Int 2014;35:286–98.
Shiravi A, Shiravi H, Tavallaee M, Ghorbani AA. Toward developing a systematic approach to generate benchmark datasets for intrusion detection. Comput Secur 2012;31(3):357–74.
Singh R, Kumar H, Singla R. An intrusion detection system using network traffic profiling and online sequential extreme learning machine. Exp Syst Appl 2015;42(22):8609–24.
Sobh TS. Wired and wireless intrusion detection system: classifications, good characteristics and state-of-the-art. Comput Stand Interface 2006;28(6):670–94.
Sommer R, Paxson V. Outside the closed world: on using machine learning for network intrusion detection. In: Proceedings of the 2010 IEEE symposium on Security and Privacy, *SP '10*. Washington, DC, USA: IEEE Computer Society; 2010. p. 305–16.
Sonchack J, Aviv AJ, Smith JM. Cross-domain collaboration for improved {IDS} rule set selection. J Inform Secur Appl 2015;24:25–40.
Song J, Takakura H, Okabe Y, Nakao K. Toward a more practical unsupervised anomaly detection system. Inf Sci (Ny) 2013;231:4–14. Data Mining for Information Security.
Sreelaja N, Pai GV. Swarm intelligence based approach for sinkhole attack detection in wireless sensor networks. Appl Soft Comput 2014;19:68–79.
Stankovic JA. Research directions for the internet of things. IEEE Internet Things J 2014;1(1):3–9.
Stauffer C, Grimson WEL. Learning patterns of activity using real-time tracking. IEEE Trans Pattern Anal Mach Intell 2000;22(8):747–57.
Su M-Y. Prevention of selective black hole attacks on mobile ad hoc networks through intrusion detection systems. Comput Commun 2011;34(1):107–17.
Thornton C, Hutter F, Hoos HH, Leyton-Brown K. Auto-WEKA: automated selection and hyper-parameter optimization of classification algorithms. CoRR, abs/1208.3719, 2012.
Tsai C-F, Lin C-Y. A triangle area based nearest neighbors approach to intrusion detection. Pattern Recognit 2010;43(1):222–9.
Veeramachaneni K, Arnaldo I, Korrapati V, Bassias C, Li K. *ai*^{2}: training a big data machine to defend. In: 2016 IEEE 2nd international conference on Big Data Security on Cloud (BigDataSecurity), IEEE International Conference on High Performance and Smart Computing (HPSC), and IEEE International Conference on Intelligent Data and Security (IDS). 2016. p. 49–54.
Wan Q, Panetta K. A facial recognition system for matching computerized composite sketches to facial photos using human visual system algorithms. In: 2016 IEEE Symposium on Technologies for Homeland Security (HST). 2016. p. 1–6.
Wang Y, Dou Y, Liu X, Lei Y. PR-ELM: parallel regularized extreme learning machine based on cluster. Neurocomputing 2016;173(Pt 3):1073–81.
Wu SX, Banzhaf W. The use of computational intelligence in intrusion detection systems: a review. Appl Soft Comput 2010;10(1):1–35.
Xiang J, Westerlund M, Sovilj D, Pulkkis G. Using extreme learning machine for intrusion detection in a big data environment. In: Proceedings of the 2014 workshop on Artificial Intelligent and Security Workshop, AISec '14. New York, NY, USA: ACM; 2014. p. 73–82.
Xu LD, He W, Li S. Internet of things in industries: a survey. IEEE Transactions on Industrial Informatics 2014;10(4):2233–43.
Yang W, Tao J, Xi C, Ye Z. Sign language recognition system based on weighted hidden Markov model. In: 2015 8th International Symposium on Computational Intelligence and Design (ISCID), vol. 2. 2015. p. 449–52.
Zanella A, Bui N, Castellani A, Vangelista L, Zorzi M. Internet of Things for smart cities. IEEE Internet Things J 2014;1(1):22–32.
Zhang X, Wu G, Dong Z, Crawford C. Embedded feature-selection support vector machine for driving pattern recognition. J Frankl Inst 2015;352(2):669–85. Special Issue on Control and Estimation of Electrified vehicles.
Zhao K, Ge L. A survey on the internet of things security. In: Computational Intelligence and Security (CIS), 2013 9th International Conference on. 2013. p. 663–7.
Zhou Z-H, Wu J, Tang W. Ensembling neural networks: many could be better than all. Artif Int 2002;137(1–2):239–63. cited By 900.

Lei Pan works at Deakin University where he serves as the director of Master of Cyber Security course. He is mentoring and coaching security students in participating hackerthorns and ctf challenges. He is also an active educator at futurelern.com

Rory Coulter received first class honours in Information Technology from Deakin University, Melbourne, Australia. He is currently a computer science Ph.D. candidate in artificial intelligence and cyber security at Swinburne University of Technology, Melbourne, Australia. His research interest includes data driven cyber security, intrusion detection, malware detection, machine learning and visualization. He is very interested in knowledge discovery through applying new fields and challenges to deep learning, traditional machine learning and visualization. Any resources that may be shared through Data or Datasets, software and code repositories are very much appreciated for collaboration.