2018 IEEE 38th International Conference on Distributed Computing Systems

How to Prevent Skynet From Forming

(A Perspective from Policy-based Autonomic Device Management)

S.Calo, D. Verma
IBM Research
Yorktown Heights, NY, USA

E. Bertino
Purdue University
Purdue University, USA

J. Ingham
UK DSTL, Porton Down
Wiltshire SP4 0JQ, UK

G. Cirincione
Army Research Labs
Adelphi, MD, US

***Abstract*—Artificial Intelligence (AI) in the context of military systems has frequently been portrayed as dangerous, and as leading to humanity being put in danger by an errant AI system, such as the Skynet imagined in the Terminator movie series. At the same time, the benefits of using AI in such systems are numerous. Therefore, we need to develop techniques that will let military systems benefit from the advances in AI, while ensuring that a system like Skynet never turns against humanity. In this paper, we examine the problem from the perspective of device management, a set of intelligent systems that manage themselves and determine their own policies. We discuss mechanisms that could be used to prevent these systems from becoming malignant.**

Keywords—AI, military systems, policy-based management, distributed systems, collaborative systems

I. Introduction

As part of an international research collaboration between the U.S. and UK militaries, an alliance of researchers from several universities, industrial research labs and government research labs is exploring fundamental science challenges in the area of distributed Artificial Intelligence (AI)/Analytics in the context of military coalitions. Among the challenges being explored is one related to generative policies, i.e., determining approaches by which devices can generate the policies which they need to manage themselves [1, 2]. A specific area is that of device security, i.e., allowing devices to determine their own security and safety policies, and this determination is driven largely by the use of AI technologies that learn from human behavior, enabling devices to share the intelligence they learn [3], and improve upon their safety and security as well as the safety and security of humans.

A system of devices that use AI to determine their own security policies for military coalition operations can sound extremely scary. Such a system forms the villain in several popular movies and science fiction stories, where such a system takes over the control of the world from humans. Examples include the Skynet system in the Terminator series of movies, the sentient machines in the Matrix series of movies, the super-computer VIKI in the I, Robot movie, the Omnius network in the Dune series, and several others. There have been several prominent people who have voiced a concern that the growing uses of AI can lead to such systems being formed.

While it is easy to dismiss these concerns as being naïve, irrational and unscientific and mostly expressed by people not working directly in the field of AI, such a dismissal would be unscientific and irrational as well. Given the amount of scientific effort being poured into research in machine learning and AI, it is inevitable that some set of researchers will make scientific breakthroughs that enable machines to make better decisions than humans. In that case, we need to have approaches, which are embedded into such systems, which prevent the systems from running rogue. Research has recently started to address such concerns and prominent directions include explainable AI [4], quantification of input influence in machine learning algorithms [5], ethics embedding in decision support systems [6], "interruptibility" for machine learning systems [7], and data transparency [8].

In this paper, we consider a set of technologies that are complementary to other technologies being investigated to address the major challenge of preventing AI systems from running rogue. In order to present approaches and solutions, we first discuss the type of intelligent devices that we are envisioning in the future, specifically in the context of military coalition operations. We then discuss the properties that a set of intelligent devices must exhibit in order to develop into a rogue system, followed by a discussion of devices that manage themselves and generate their own management policies, discussing the similarities between such systems and Skynet. This is followed by a more precise definition of what it means to form a rogue Skynet in terms of system state, a discussion on how such a rogue behavior may emerge, and the suggestion of some high-level approaches that can be used to prevent the formation of rogue systems. These ideas are meant to be a starting point for development of potential approaches for preventing Skynet formation. Finally, we consider one specific approach, namely safe state space maintenance and discuss how it can lead to a secure set of intelligent devices.

II. Intelligent Devices for Military Coalitions

We envision a future in which each human tasked with any responsibility is assisted by a fleet of intelligent devices. These devices take over repetitive, mundane, dangerous, or physically taxing activities from human beings, so that the human can perform his or her task more efficiently and safely. While such assistive devices can be present in almost any aspect of human life, we will focus on the use-cases that arise in military coalitions.

In the future, coalitions of friendly nations may be required to conduct various tasks such as maintain peace in a troubled region. The war-fighters in the coalition are assisted by different devices in their military operations, and the devices have a significant degree of autonomy. The devices can call

2575-8411/18/$31.00 ©2018 IEEE
DOI 10.1109/ICDCS.2018.00137

upon other devices with additional capabilities, including lethal fire-power, when needed.

As an example, we can consider a scenario in which two friendly nations (such as the U.S. and the U.K.) are charged with maintaining peace in a remote mountain range supporting a nation which is beset with insurgents. Each coalition member is in charge of reconnaissance and surveillance operations in separate geographic areas. The personnel in charge of surveillance in both countries rely on a set of surveillance devices such as drones and mules. When needed, a device can call upon and dispatch other devices with additional capabilities, e.g., a drone sees smoke and calls upon another drone with chemical and radioactive sensors to make a better assessment of what the situation may be. Similarly, if it sees a suspect convoy, it may call upon a ground mule to intercept the convoy along the path. In current ground operations, most of these decisions are made by human beings, but in the future these decisions will be made by the devices themselves, with only a few decisions being sent for human cross-validation.

As another example, one can consider a confrontation between two opposing coalitions of countries where both sides are technologically advanced. Each of the coalitions would have automated intelligent devices on land, sea, sky and space that have different types of sensors and actuators. These

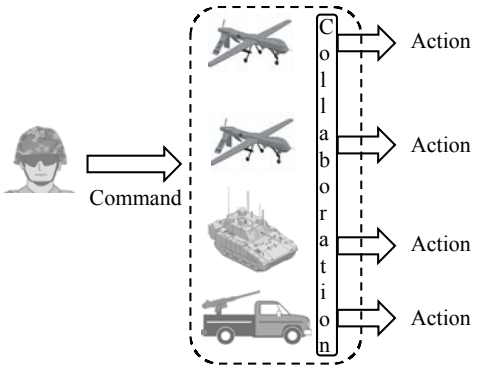

Figure 1: Mode of Operation of Devices

devices would attempt to work together to destroy the devices, infrastructure and (unfortunately) the humans associated with the other side. Each coalition would use automated decision making as a necessary measure for survival. Humans may only be involved in strategic decisions, as opposed to any tactical activities. The operational approach would be as shown in Figure 1, where several devices within control of a human collaboratively decide how to execute actions that satisfy the command of that individual.

Since each human will oversee many different devices, ranging from tens to hundreds, the devices would need to be self-managing. They would need to repair themselves, or go to another mechanic device to be repaired, and deal in an autonomous manner with failures, security issues and any other operational constraints that may arise during their operation. This would also include the task of self-determining any policies that may be applicable for management functions. This self-determination of policies is likely to also be the capability which may lead to the possibility of a device running rogue.

Since, the concept of "forming Skynet" or "turning rogue" is a vague one, we need to define it with some more precision.

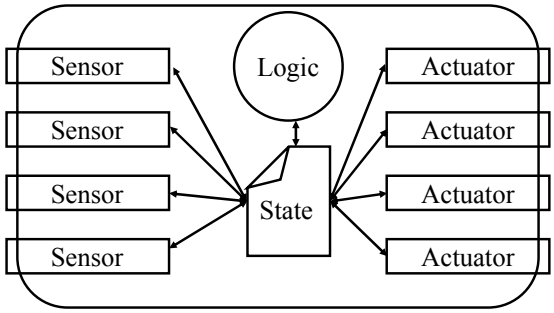

Figure 2: Abstract Model of a Device

In the next section, we discuss some of the properties that a system like Skynet may possess, which is followed subsequently in Section V by a more specific definition of what the "rogue device" or "Skynet formation" means based on an abstract model of the devices.

III. PROPERTIES OF SKYNET

Skynet or equivalent systems in popular movies/literature are usually presented according to an anthropomorphic representation. In order to understand how Skynet may form, or more specifically, how we can prevent it from forming, we need to look at it in a more technical manner. In this section, we discuss the properties that are required for a system to potentially become Skynet. The following are some of the key requirements that a system like Skynet must display:

- **Networked**: Skynet is a collection of many different devices that are networked together and form a collective. Even when it is represented as a single master computer, the master computer uses several other machines in the network as its sub-components, and is made up of several intelligent sub-components.
- **Learning**: The Skynet system learns from its environment, and the actions of others around it. The system is not limited to a static set of knowledge, or a pre-determined preprogrammed set of instructions it follows. The learning component of the Skynet includes an ability to be aware of the current situation, and predict and anticipate changes in the current situation based on the potential actions of other entities in the environment.
- **Cognitive**: Because the system learns, and is capable of using other intelligent processes such as reasoning, and predictive analysis, it can take actions that it was not originally programmed to do. The capabilities of Skynet keep on evolving and it is able to expand upon its capabilities over time.
- **Multi-Organizational**: In order to be most effective, a system like Skynet needs to leverage and take over

computing devices that may belong to more than one organization. A single organization system can be forced to restrict its damage to other systems in the organization, but a multi-organization system can use resources from other systems, and bring them under its own control.

- **Physical Aspect**: In order to cause damage to the humans, the Skynet system must have a physical component, i.e., a component which changes the physical environment around it or any of its constituent pieces. Skynet cannot exist in a pure information domain[1].
- **Malevolent**: The system tries to take at least one action which it was not originally programmed to do, and the action is potentially harmful to humans. A malevolent system requires taking some action that it is aware may cause harm to humans and was not an intent provided to it by any legitimate human user.

While current systems may not be able to meet these requirements, several research efforts in many organizations are working on technologies that can lead to systems meeting these requirements. We will now review briefly the scope and goal of one such effort. This effort is presented as an exemplar of several efforts that would create technologies that can be imagined to inadvertently lead to forming a system like Skynet.

IV. Generative Policies for Systems Management

As discussed in the introduction, the research alliance [9] established between several U.S. and UK research organizations has developed the concept of generative policies, enabling devices to generate the policies they need to manage themselves on their own. The basic motivation is driven from the realization that humans would not be able to manage a large number of devices and may not even be able to define policies for how these devices ought to work. A policy in this context is an event-condition-action rule directing the devices to take specific actions when an event happens and the conditions specified hold true. While a traditional policy based management system would have a human define these policies, the generative policy approach [1, 2] allows the devices to generate the policies themselves from a high level description of the overall environment.

In the current version of the generative policy architecture, a human manager provides two types of information to each device. The first type of information specifies what the device can expect to see in its environment, in particular the other types of devices that would be encountered and their attributes. The second type of information provides directions indicating what kinds of policies it should generate as new devices are discovered in the environment. The former is specified by means of an interaction graph, the latter by means of a policy generator grammar or a policy template. Based on these two classes of information, devices discover other devices in the system and decide on the policies to be used in their interaction with those devices.

The current architecture is only the starting point for providing devices with the ability to manage themselves. It can be augmented and will continue to be extended in many ways. While the current system is relatively innocuous and far from becoming a Skynet like system, future research explorations make the system more likely to approach Skynet characteristics. As the devices need to decide on the new policies to be generated, they can augment the information provided by the human manager on their own. They can use unsupervised machine learning techniques to add or remove from the types of devices that the human has specified, learn the relationship between the attributes they see among the devices in the system and create predictive models of those relationships, share the information and policies they generate with other devices, and mine information on the web to augment their predictive models, among other techniques. These technologies will be applied to several application domains, including the management of the security of devices in coalition contexts, and for the operation of cyber-physical systems.

The autonomous systems that follow such a generative policy architecture have several attributes, which include:

- **Networked**: The generative policy system has the ability to manage a networked set of devices, with dynamic discovery.
- **Learning**: The generative policy system will use machine learning techniques to improve its ability to generate effective management policies.
- **Cognitive**: The system improves upon its policy management capabilities over time.
- **Multi-Organizational**: The system is targeted to address coalition environments, which are multi-organizational by nature.
- **Physical Aspect**: Some application domains of generative policies are related to physical environments.

Comparing these attributes with those of Skynet, we can see a very good correspondence, with the only differing property being that the Skynet is malevolent. If the generative management system also picks up this property along the way, it can lead to the generation of a Skynet like system.

While no rational person would design the system to be malevolent, there are many ways by which malevolence can creep into the system. A malevolent system in turn can bypass controls that are put in place by humans. Among the mechanisms that could lead to such behaviors are the following:

- *Mistakes in Learning*: As machines learn from the environment, they can make many mistakes. Mistakes can result because of bad data (the data being fed to the learning system is biased or insufficient), a bad algorithm (the algorithm being used makes assumptions that are not valid), a bad system design (implementation

[1] Rogue processes purely in the information domain are also a security threat and need to be protected against. However, Skynet like systems are more dangerous because they also have a physical component, and can cause physical damage directly or indirectly by using their control over information systems.

bugs, untested software), or other factors that can lead to incorrect models being learnt.

- *Attacks to Systems*: A system of devices can be subject to cyber-attacks, and an intruder may be able to insert spyware or other types of malicious software in the device. A reprogrammed device may turn malevolent and convert other devices into following the same behaviors. Notice that, whereas so far intrusive activities (like inserting spyware) are started by some human hackers, nothing prevents an intelligent malevolent system to start hacking other devices on its own.
- *Adversarial Machine Learning*: A subset of attacks that can be launched belong to the field of adversarial machine learning [17, 18], and we are calling it out because it is specially targeted for learning systems. Attacks in this area include attempts to poison data used for training, obfuscating features of data used for training, denying access to selected sets of data, along with other measures that can interfere with the training and correct use of trained models. Counter-measures that are used to counter such attacks enable machines to exclude selected training data from consideration, which can also lead to machines learning unexpected patterns and behaviors.
- *Backdoors and Vulnerabilities*: In an attempt to have humans in control of machines, a common but perhaps misguided philosophy is to have a backdoor that can be used by a human to enter into the system and shut it down. Unfortunately, it also introduces a significant vulnerability for malware to be introduced into the environment. As we already mentioned, it would be easy for an intelligent malevolent system to exploit such vulnerabilities on its own. Once such a system gets into other systems, it can disarm existing controls (such as anomaly detection tools).
- *Inappropriate Emulation*: A common way for machines to improve themselves and learn new skills is to emulate the behavior of humans by observation. After a sufficient number of observations of how a human handles a situation, a machine can create a system to replicate it. However, humans are imperfect and prone to make mistakes, and the encoding of imperfect human behavior can lead to a mistaken and sometimes malevolent machine forming.
- *Malicious Actors:* A malicious human being can launch a variety of attacks on a machine to turn it malevolent. For machines involved in battlefields, it is easy to envision an adversary going out of the way to turn a machine against their own opponents. In particular, a malicious adversary can carefully manipulate the input data used for training a machine learning algorithm to exploit specific vulnerabilities of such algorithms to compromise the whole system. Many studies have been conducted in the area of adversarial learning to identify defense techniques against such attacks [17, 18].
- *Human errors:* Human error is often the cause for malfunctions and accidents in many different environments. A wrong command by the human operator, a mistake in understanding the limitations of the system, or inappropriate use of a device can lead to malevolent conditions. A machine that is designed for war-fighting could be used in peace-keeping operation, and may take inappropriate actions unsuitable for the environment. A scientist may create a powerful virus in a lab for research purposes, and accidentally release it outside. Similarly, a system created in lab may be accidentally deployed without a full set of validation tests, and prove to be malevolent like an escaped virus.

One or a combination of the above factors may result in formation of a system like Skynet. In order to prevent this, we first try to formulate a more precise definition of what being Skynet might mean.

V. Device Model and Definition of Skynet

Any device can be viewed as a set of sensors and actuators which has logic dictating its behavior under different circumstances. One way to characterize any such device is by its state, where the state is defined as consisting of the values of a set of variables, where each variable represents an attribute of the configuration of the sensors, actuators or other aspects of the device. When an event occurs (e.g., changes in sensor values, reception of a message from a network connection, etc.), the logic used within the device looks at the current state and the inbound event, and then takes an action. The result of the action, which may invoke an actuator, effectively moves the device to another state. This model is shown in Figure 2. The logic may be expressed as a set of event-condition-action rules, where the condition is the current state of the device, and the action is the invocation of an actuator, resulting in a new state.

The logic can be built in by the developer of the device, in which case the state transitions are a part of the natural behavior of the device. They could also be specified explicitly by the owner of the device in order to manage the device, or to program the device. In the former case, such rules define the concept of policy-based management, and in the latter case such rules perform the primary task assigned to the device. Given the similarity between the different pieces of logic, we can focus on the characteristics of the device from a policy-based management perspective.

In typical manual management processes associated with a device, some of the states of the device reflect its normal operation, while others are ones in which the device needs attention or repair. The good states (normal operation) and the bad states (need repair) can be identified by a set of conditions (e.g., the results of a set of diagnostic checks). In this regard, many states may actually be neither "good" nor "bad", but might be considered "neutral". In general, one could consider a "safeness" (or risk) metric associated with each state. The safeness metric would induce a partial ordering on the set of states. We would like the system to move to states with the highest safeness metric. In cases where this is not possible, one can choose the next best state. In any case, the truly "bad"

states where the safeness is below an acceptable level must be avoided.

When the device enters a bad state, a human being can take actions to change the state, e.g., if the bad state is entered because a configuration parameter has been set too high, a human being can adjust it to a lower value. In policy-based management, a set of policies (or rules) are written that can automate the steps to be taken when a device enters a bad state. They can adjust the state so that an actuator can display the need for a corrective action, e.g., when some parameters exceed some thresholds, the system can automatically reset their values to be within acceptable bounds. The adjustment can be controlled by means of the event-condition-action rules described earlier. The definition of the good states versus the bad states, and the specification of the policies are done by a human, while the state transitions happen automatically.

In the generative policy architecture, devices can create their own policies. The job of the human being is to just identify the good and the bad states within the system, and the devices automatically determine how to maintain the system within the good states. This requires the devices to be able to automatically detect their current states and possibly anticipate the potential next states. There is today a lot of technology that would make this possible; see for example the use of a vision analytics approach to support automatic state inference for helicopters [10] which uses images obtained by an inexpensive camera mounted on the helicopter cockpit.

In a simplified case, where the state consists of only two variables, the state space can be represented visually as shown in Figure 3. In manual management, the boundaries are defined by a human and the device asks for help when the state moves into a bad one. In policy-based management, the logic for moving from bad states to good states is defined by a human, in addition to the definition of the good and bad states. In the generative policy architecture, the logic is determined automatically by the device.

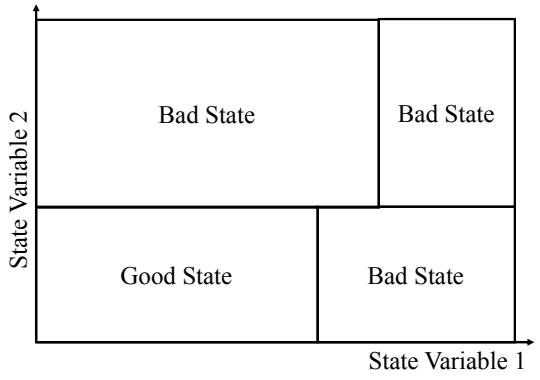

Figure 3: Simplified State Description of System

Just like the definition of good states and bad states can be used for management functions, an analogous definition can be used to define Skynet. We can consider the devices as being in a good state when the device is in a state that does not harm a human[2] and in a bad state when the device is in a state that can harm a human. Our task is to ensure that no device and no collection of devices ever end up in a situation where they are in a bad state.

Our task is further complicated by the fact that the harm to a human may come from the collective actions of different devices. While each of the devices may individually be in a good state, i.e., no one is individually taking an action that may harm a human, the net impact of the action may result in harm to the human.

It may therefore be difficult to categorize the states of a system as being exclusively "good" or "bad". There are a number of research challenges in making such a determination. Even for single devices, some states may be explicitly "bad", but others may be dangerous in that they lead to sequences of states with some cumulative effects that are undesirable. In collections of devices, the impacts of state changes in each individual device on the states of other devices must be considered. The patterns of states exhibited by the collection may also be difficult to interpret because of temporal effects or emergent behaviors.

VI. Approaches to prevent Skynet formation

We want the collection of systems working together to never enter a situation where they end up in a bad state (where a bad state is one in which a human may be harmed). Assuming that it is possible to clearly separate out the good states from the bad states, we can use the following techniques to ensure that the bad states are never entered. Each technique assumes that it can be performed in a manner that is tamper-proof.

A. Pre-Action Checks

Since the main goal of the system is to ensure that there is no harm being done to a human, one approach is for each device to incorporate a check before taking any action (i.e., activating any actuator) that the action will not harm a human. A set of properly defined checks before the action would ensure that any action taken by a device is safe. The pre-action check can work whether a single device or a collection of devices is being used for the task. It can prevent a direct action which causes harm to the human.

However, if the action causes indirect harm to a human, the pre-action check may fail in some cases to catch that. As an example, if the action is to dig a hole in a path, and the machine does not anticipate a human to come on the path, it will proceed to dig a hole. The next human that comes on the path (against the machine's prediction) is likely to fall in that hole and get hurt. One approach to prevent indirect harm to humans would be to extend the event-condition-action with obligations, that is, further actions that need to be executed after the original action has been executed (or even while the original action is being executed) [11]. In the example of the hole, possible obligations would include posting notices

[2]Management systems are designed to keep the devices running in good stead, so they should not harm any humans. In the case that a system is designed to harm an unfriendly group of humans, the definition should be changed to specify that it does not harm any friendly humans.

indicating the hole, broadcasting messages to humans approaching the location of the hole, and so forth. Notice that many such obligations can be easily automatically implemented with today's technology. The main interesting challenge is to develop ontologies of such obligations so that devices can automatically select the ones most relevant to their actions.

Regardless of the limitations, a pre-action check is required in a system to ensure that devices minimize the chance of hurting humans.

B. State Space Checks

The logic of the device is responsible for maintaining its state. If the good states and bad states can be identified properly, then the device can maintain a check which prevents it from ever entering a bad state. If the device finds itself entering into a bad state, it will not take the action that leads to that state, simply choosing the option of taking no action (which keeps it in the current good state) or taking an alternative action which puts it into a new state which is also good.

Note however that situations can occur in which the only possibility for the device of escaping a bad future state is an action that would place the device into another bad state. An example would be of electronic components having no alternative but to run at maximum capacity to prevent loss of life but risking a fire at the same time. There are approaches that can be combined to address such issues, including: (a) the adoption of break-glass rules – often used in medical applications; (b) an ontology of state preferences; and, (c) risk estimation techniques.

Break-glass rules are typically used in medical systems to allow operators emergency access to data and IT systems when normal authentication cannot be successfully completed or the access control policies would not allow access [12]. Use of such rules in our context would require support for audits to verify that devices did not abuse the break-glass rules. Such audits in turn would require the collection of comprehensive context information. In addition, it is critical that a device be able to obtain trustworthy information concerning its own status and the environment to allow the device to base its decision of breaking the glass on true information. This in turn requires the deployment of specialized techniques to protect devices that typically acquire information by using sensors (both their own and possibly of other devices) from deception attacks [13].

A state preference ontology organizes the possible states of a device into an ontology based on a preference relationship. Organizing the set of bad states into such an ontology allows a device, which has to decide between two bad states, to select the "less bad" state. For example, in the case of a device that has to decide between preventing loss of human life and starting a fire, most likely the former will be the worse bad state and thus the device would go into the state that would eliminate or minimize human life loss, even if this means starting a fire. The notion of state preference is based on approaches for preference graphs developed in the area of constraint satisfaction and optimization [14] and techniques developed in this area could be applied to our context. A critical issue for the deployment of such an approach is the creation of the state preference ontology; such an ontology can be based on human input and then possibly refined based on machine learning techniques, testing, and scenarios.

The use of a state preference ontology would work particularly well when combined with risk estimation techniques in that it would allow devices to make a more articulated decision about which next state to move to. Risk assessment would be particularly useful, for example, when all possible next states may involve losses of human life. Deploying such an approach requires the device to have reliable and up-to-date information about the context, and also to incorporate application-dependent risk factors which may be very specialized not only for specific applications but also for specific situations and contexts.

If a check for good state versus bad state exists, then a tamper-proof implementation of this check when the system is executing its logic would ensure that the device never enters a bad state. Also it is critical to ensure the trustworthiness of data used in the state assessment.

C. Deactivating Machines in Bad States

As in the case of manual administration, devices that go into a bad state or are prone to take actions that make them go into a bad state, can be deactivated by a tamper-proof mechanism. If every device is never allowed to go into a bad state, then it is less likely that a collection of such devices would get into a bad state. In order to ensure that such collections will not go into a bad state, however, aggregate and emergent behaviors must be considered, as will be subsequently discussed.

Even if devices are allowed to make their own decisions as to the policies they implement in their logic, if each step of the logic invocation is checked by this step, or their transition into a new state is guarded by such a step, the devices can be maintained in a safe situation where they do not harm the human. Unfortunately, this does not insure that a collection of such devices cannot become harmful.

D. Checks on Collection Formation

As we have mentioned, a critical issue is that the combination of many innocuous devices could become a dangerous device, one which could potentially be harmful to the humans. For example, components within an electronic device may each be operating within regions where the heat that they generate is acceptable, i.e., it does not interfere with their operation, but the cumulative amount of heat generated may exceed the safety limits of the device, potentially causing fire.

One method for preventing the Skynet from forming would be to use a human check each time a network of devices is formed, i.e., when a new device is added or removed from the network. If a manual check is involved, and the human making the check is assisted by another machine which remains offline and disconnected from other machines while assisting the human to run through a situational analysis of whether the new network configuration can potentially cause harm to the humans, the probability of any single device or a collection of

devices entering a bad state can be significantly reduced. Such an approach corresponds to the very well-known security principle of separation-of-privilege [15]. We need also collaborative state assessment techniques by which a group of devices would jointly determine whether a set of actions, to be undertaken by devices in the group, could lead to some aggregate bad states, even though each device would still be in good state.

Modelling, analysis and simulation methods have been used to determine whether systems of systems would exhibit emergent behavior [16]. Such behaviors emerge from the interactions between individual component systems or system elements. They can be quite complex, and in some instances, may arise in ways counter to the intended functioning of the system components, e.g., rolling blackouts in a power grid. Thus, not only do the states of individual components need to be considered, but also the manners in which they interact.

E. AI overseeing AI

One way to counter an intelligent collective which can exceed human abilities using AI approaches would be to have each such collective be overseen by another collective. In human societies, a system of check and balances is often created by having institutions that keep each other's power in check. An example is in the U.S. constitution, where the three branches of federal government, the judiciary, the executive and the legislative keep each other in check. A similar paradigm can be followed when creating intelligent autonomous systems, by creating not a single collective of machines, but two or more collectives, each of which keeps the other ones in check by creating a system of check and balances.

As an example, any collective that has the ability to change the physical world can generate their policies and act upon them, but it needs to ensure that its actions are within the scope defined by a set of higher level meta-policies that are defined by an independent and distinct collective. When there is an inconsistency between the existence of the conflict of the scope, the inconsistency is resolved by another intelligent collective which arbitrates the dispute among them. The three collectives can be viewed as the analogues of the executive, legislative and judiciary branches in human governance.

Looking at the specific case of defining states as good, bad or neutral, and associating them with risks and utility functions, the executive collective would be responsible for its assessment of risk and utility, while the legislative collective would be responsible for defining the risk and utility function using its learning mechanism. The judiciary would determine if any of the functions are inappropriately interpreted under a given state of the overall system. Assuming that two out of the three collectives always prevail, these three collectives would keep each other in check, and reduce the overall system moving into a malevolent state.

An exploration of similar check and balances among multiple intelligent collectives, and having them control each other to prevent malevolence, would be a promising area of investigation.

VII. ILL DEFINED STATE SPACES

A challenge with the implementations of the schemes defined in the previous section is that they all depend on the ability to define what the good and bad states are, where the good states are those in which the device is not able to cause harm to a human, and the bad states are those in which the device can cause harm to a human. However, in many cases, the configuration of a device may be too complex for a human to define the rules that can map state spaces into good or bad. We therefore need an approach to handle this situation.

One approach to handle complex state spaces is to define good versus bad states with regard to their relationship to the different state variables. If we consider a device state being defined by means of N variables, $x_1, x_2, \ldots x_N$, then the task of defining the state space requires defining a function which maps the values of these variables into the good or bad classification for any combination of the variables. In other words, we need to define the function $f(x_1, x_2, \ldots x_N)$ which maps into a binary space of good versus bad states. The definition of this function may not always be easy to do.

However, for different state variables, it may be easy to determine the sign of the derivative of the function. While a human may not be able to exactly define whether the state is good or bad, it may be possible to define what the likelihood of the state becoming worse or better is given any change in a state variable. In effect, we may be able to define the sign of the partial derivatives ($\partial f/\partial x_i$) with respect to some (if not all) of the state variables. In those cases, we can write rules that define a utility function for the device which changes as a function of the different parameters. Note that the utility function could be, but need not be, different than the risk function described earlier. Specifically, the utility may augment the risk function with the value that is determined in satisfying the objective or goal that is given to the system.

From an anthropological perspective, the utility function may be viewed as a pain or pleasure function for the device, where the pain increases as the device approaches a bad state, and the pleasure increases as the device approaches a good state. The pleasure and pain functions are defined as functions of different state variables and combined to create the aggregate pleasure or pain for any device. This partial derivative approach allows the devices to prefer moving into states which do not harm humans and cause them pain as they move into a state where a human may be harmed. As devices would try to maximize their pleasure and avoid pain, they would prefer to take actions that will not cause harm to the humans.

While the introduction of the utility function does not provide an absolute fool-proof mechanism that will prevent a collection of devices from ever harming a human being, the mechanism can decrease such a probability in a significant manner.

VIII. CONCLUDING REMARKS

In this paper we have discussed the use of policy-based management and state analysis techniques as an approach for assuring that next-generation intelligent devices will not endanger humans – which is today a significant concern given

the massive deployment of advanced AI techniques and autonomous intelligent devices embedding these techniques. Of course, the mechanisms proposed in this paper have to be combined with others to effectively address such concerns. The right combination of techniques also may depend on the specific category of devices and applications. It is also important to make sure, as always, that software running on these devices is free from vulnerabilities and that devices have proper defense mechanisms – some of which have been mentioned in the paper – in order to prevent malicious adversaries from corrupting devices. In this respect techniques and tools for the security of autonomous devices and device systems will be increasingly crucial.

ACKNOWLEDGMENT

This research was sponsored by the U.S. Army Research Laboratory and the U.K. Ministry of Defence under Agreement Number W911NF-16-3-0001. The views and conclusions contained in this document are those of the authors and should not be interpreted as representing the official policies, either expressed or implied, of the U.S. Army Research Laboratory, the U.S. Government, the U.K. Ministry of Defence or the U.K. Government. The U.S. and U.K. Governments are authorized to reproduce and distribute reprints for Government purposes notwithstanding any copyright notation hereon.

REFERENCES

[1] D. Verma, S. Calo, S. Chakraborty, E.Bertino, C. Williams, J. Tucker, B. Rivera, "Generative Policies for Autonomic Management", Proceedings of 1st IEEE International Workshop on Distributed Analytics InfraStructure and Algorithms for Multi-Organization Federations, DAIS 2017, Fremont, CA, USA, August 6-7, 2017.

[2] E. Bertino, S. Calo, M. Touma, D. Verma, C. Williams, B. Rivera, "A Cognitive Policy Framework for Next-Generation Distributed Federated Systems: Concepts and Research Directions", Proceedings of 37th IEEE International Conference on Distributed Computing Systems, ICDCS 2017, Atlanta, GA, USA, June 5-8, 2017.

[3] E. Bertino, Geeth de Mel, Alessandra Russo, Seraphin B. Calo, Dinesh C. Verma, " Community-based Self Generation of Policies and Processes for Assets: Concepts and Research Directions", Proceedings of 2017 IEEE International Conference on Big Data, BigData 2017, Boston, MA, USA, December 11-14, 2017.

[4] D. Castelvecchi, "Can we Open the Black Box of AI?", Nature, October 5, 2016.

[5] A. Datta, S. Sen, and Y. Zick, "Algorithmic Transparency via Quantitative Input Influence: Theory and Experiments with Learning Systems", Proceedings of IEEE Symposium on Security and Privacy, SP 2016, San Jose, CA, USA, May 22-26, 2016.

[6] J. Greene, F. Rossi, J. Tasioulas, K. B. Venable, B. C. Williams, "Embedding Ethical Principles in Collective Decision Support Systems", Proceedings of the Thirtieth AAAI Conference on Artificial Intelligence, February 12-17, 2016, Phoenix, Arizona, USA.

[7] E. M. El Mhamdi, R. Guerraoui, H. Hendrikx, A. Maurer, "Dynamic Safe Interruptibility for Decentralized Multi-Agent Reinforcement Learning", Proceedings of 30th Annual Conference on Neural Information Processing Systems, NIPS 2017, 4-9 December 2017, Long Beach, CA, USA.

[8] E. Bertino, S. Merrill, A. Nesen, C. Utz, "Data Transparency – Concepts and Research Directions", IEEE Computer (to appear in June 2018).

[9] G. Seffers, "Solving the Hard Science for Future Coalition Forces," Signal Magazine, February 2017, pp 32-33

[10] S. Shin, I. Hwang, "Data-Mining-Based Computer Vision Analytics for Automated Helicopter Flight State Inference", J. Aerospace Inf. Sys. 14(12): 652-662 (2017).

[11] Q. Ni, E. Bertino, J. Lobo, "An Obligation Model Bridging Access Control Policies and Privacy Policies", Proceedings of 13th ACM Symposium on Access Control Models and Technologies, SACMAT 2008, Estes Park, CO, USA, June 11-13, 2008.

[12] Joint NEMA/COCIR/JIRA Security and Privacy Committee (SPC), " Break-Glass – An Approach to Granting Emergency Access to Healthcare Systems", December 2004, downloaded on Feb. 4, 2018, from http://www.medicalimaging.org/wp-content/uploads/2011/02/Break-Glass_-_Emergency_Access_to_Healthcare_Systems.pdf.

[13] M. Rezvani, A. Ignjatovic, E. Bertino, S.K. Jha, "Secure Data Aggregation Technique for Wireless Sensor Networks in the Presence of Collusion Attacks", IEEE Trans. Dependable Sec. Comput. 12(1): 98-110 (2015).

[14] F. Rossi, K. B. Venable and T. Walsh, "Preferences in Constraint Satisfaction and Optimisation," AI Magazine, pp.58-68, 2008.

[15] J. Saltzer, M. Schroeder, "The Protection of Information in Computer Systems", The Proceedings of the IEEE 63(9): 1278-1308 (1975).

[16] J. Osmundson, T. Huynh, G. Langford, "Emergent Behavior in Systems of Systems", INCOSE International Workshop 2008.

[17] B. Biggio, G. Fumera, and F. Roli, "Design of robust classifiers for adversarial environments", Proceedings of IEEE Int'l Conf. on Systems, Man, and Cybernetics (SMC), pages 977–982, 2011.

[18] M. Torkamani and D. Lowd, "Convex Adversarial Collective Classification", in Proceedings of the 30th International Conference on Machine Learning (pp. 642-650), Atlanta, GA., 2013.