Security Issues for Distributed Fusion in Coalition Environments

Greg Cirincione
CISD
Army Research Lab
Adelphi, MD, USA
gregory.h.cirincione.civ@mail.mil

Dinesh Verma
Distributed AI Dept
IBM TJ Watson Research Center
Yorktown Heights, NY, USA
dverma@us.ibm.com

Elisa Bertino
Dept of Computer Science
Purdue University
West Lafayette, IN, USA
bertino@purdue.edu

Ananthram Swami
CISD
Army Research Lab
Adelphi, MD, USA
ananthram.swami.civ@mail.mil

***Abstract*—When sensor fusion operations are conducted in coalition environments, security of the data and infrastructure used for model fusion are very important. AI enabled sensor fusion infrastructure can be attacked on many fronts, including attacks on the data used for sensor information fusion and disrupting the communication between devices and the fusion nodes, in addition to the traditional security attacks. As the infrastructure for sensor fusion becomes more automated with multiple intelligent assistants for data collection, different types of attacks are possible. AI enabled approaches can be used to improve the security and resiliency of federated networks, and the data that is shared across coalition problems. In this paper, we discuss the challenges associated with security of coalition infrastructures, and approaches to improve the security using AI and machine learning techniques**

Keywords—Distributed Learning, Security, Coalition Operations, Generative Policies

I. Introduction

During coalition operations, there are many scenarios where operational effectiveness can be improved by use of distributed data fusion, where coalition partners share and transform information using AI enabled fusion algorithms to interpret the information and support reasoning. However, due to differences in the trust level that exists among coalition members, such distributed fusion introduces new security vulnerabilities and requires proper safeguards. In this paper, we look at some of the security challenges that arise in sensor information fusion in coalition environments and explore ways to address them.

In Section II, we discuss our abstracted view of fusion in distributed coalition operations. The basic operational construct that we assume is that of a dynamic community of Interest (CoI) which consists of a group of people and assets drawn across different coalition members that come together for a specific purpose, perform their operations, and then disband. We focus specifically on CoIs that are created with the explicit purpose of AI enabled sensor fusion across coalitions. After describing the assumed environment, we discuss their security needs and requirements in Section III. Then, we discuss how we can use machine learning approaches to dynamically learn and generate the security policies that are required for specific CoIs. We present a preliminary analysis of the amount of training data required for the machine learning approach. The analysis shows that acquisition of training data is likely to be the most difficult challenge in building a solution, and we discuss approaches to reduce the need for training data to learn security policies.

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.

II. Dynamic Communities of Interest

A dynamic CoI brings together people and assets from multiple coalition partners to create a group with a focused activity. The small focused activity may be to complete a humanitarian mission, to perform a surveillance operation, or to build a suite of AI models (e.g. neural networks or decision trees) that can be used for subsequent coalition operations, or adapted for private use by each member. Dynamic CoIs can be enabled through a variety of approaches, e.g. via self-managing cells [1] or variations of software defined networking [2]. In this paper, we focus on the dynamic CoIs that are created with a focus on generating AI models based on fusion of information from different coalition partners.

We assume that each coalition member has its own computational and communication infrastructure which includes a variety of sensors, Intelligence, Surveillance, and Reconnaissance (ISR) equipment, information databases etc. When a dynamic CoI for creating an AI model is established, each coalition partner agrees to provide some of the information databases to be used for training the AI model. Training of the AI model also requires some computing power, which may only be available from a subset of the coalition partners. A dynamic CoI would require the assets from all the different organizations be used to create the shared AI model. However, not all members may trust the information flowing from other partners equally, and some may have concerns regarding the data being misleading, or that data being sent to others may be misused.

Different types of dynamic CoIs may need to be established depending on the nature of sensor fusion application that is being conducted. Some typical sensor fusion scenarios that may arise in the coalition context are described in the next two subsections. These two are only exemplars, and other sensor fusion scenarios with different types of topology can also arise in practice.

A. Training across Coalition Data Sets

Each coalition member may have collected some amount of data during its operations. Coalition members may desire to use the collected set of information in order to train an AI model that can be used by surveillance cameras. The trained model can then be used by all of the coalition partners.

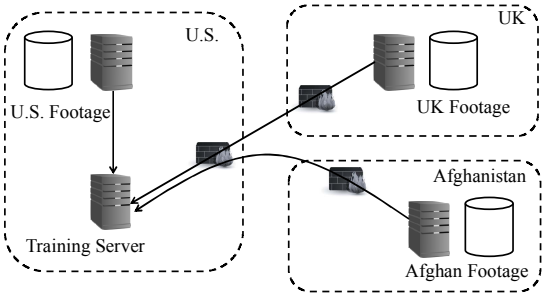

Figure 1: Model Training across Coalition Data Sets

A specific example is shown in Figure 1. The armies of U.S., UK and Afghanistan are conducting peace-keeping operations in remote mountains of Afghanistan. The three coalition members may have collected different surveillance footage of the vehicles and firearms carried by insurgents. They may want to use the combined footage to train a video recognition model to recognize such vehicles automatically and send alert notifications to soldiers when suspicious vehicles are observed. The CoI consists of people and machines that work together to create the video recognition model.

To enable the CoI, the machines that contain the surveillance footage and the machines that run the training algorithms need to be connected together for the period during which the training process runs. Let us assume that the U.S. is providing the machines that will perform the training using data from all three partners. The base-camp networks of the three countries need to be reconfigured temporarily so that the data stores containing the footage can connect to the machines running the training environment. Once the training is completed, and the models sent to partners from the U.S. machine, the need for the dynamic CoI disappears, and the ability of the machines to communicate together needs to be revoked. This requires setting up access control policies on the different components shown in Figure 1.

Assuming that the three countries can communicate over a shared common network, the firewalls in each country need to be set up to enable access to the machines holding the video footage to communicate to the server running the learning algorithm. Policies to convert data into a common format for training needs to be defined. In addition to this, the U.S. may put a lower credence on the footage from Afghan sources than those from UK sources. The U.S. may be worried about data poisoning attacks and need safeguards against them.

B. Surveillance across Partners

In another type of CoI, one of the coalition partners may be using AI enabled algorithms to conduct surveillance, e.g. trying to track a high value target using the surveillance assets available from all the coalition partners. Identification of high value targets requires fusion of a variety of information including distributed sensor inputs, intelligence reports, databases, and biometrics. The Afghan partner may have static cameras covering some parts of the local terrain, while the U.S. and UK may be providing UAVs to conduct the surveillance. The data from different partners needs to be processed at a location which is manned by U.S. personnel and the detection of the footage happens on U.S. provided computing systems. All of the assets are communicating over wireless links, and each asset needs to be able to get access to the right components in the system. Since the assets have a shared wireless link among them, they need to enforce the right security policies to interact with each other, without necessarily enjoying the benefit of a firewall. Note that this CoI may be formed after the CoI defined in A has been dissolved, and use the model trained from the previous CoI.

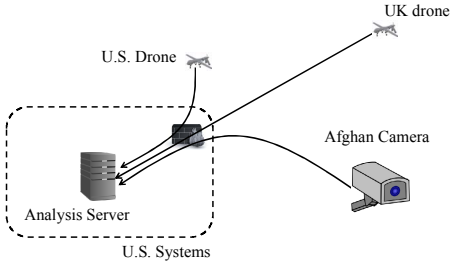

Figure 2: Surveillance across Partners

C. CoI Life Cycle

In order to support the exemplar use-cases, and other similar needs for sensor fusion in coalition environment, we need to establish the CoIs with the proper security and safety provisions in place. We assume that the establishment of dynamic CoIs follows a 3-stage life-cycle process [3] where the first stage consists of off-line planning by participants from different partners to decide which assets will be put into the CoI, the second stage consists of provisioning the environment to enable the CoI during operations, and the third stage would be dissolving the CoI infrastructure once the task of the CoI is over. The participants in each stage would be as described in [3].

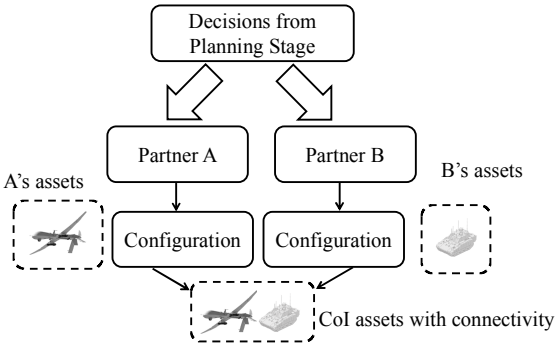

Figure 3: Process for configuring dynamic CoI

We assume that the provisioning process is done independently by each of the coalition members, where they configure their devices to enable communication with other assets during the life-cycle of the CoI.

The process for configuration of a CoI involving two partners is illustrated in Figure 3. During the planning stage, the

partners decide on the assets to be used by the CoI. Members of each partner receive, from their representative in the planning stage, the list of devices which are needed along with a description of the devices being provided by the other partners. They would also receive any shared credentials to be used for authenticating members of the CoI. Members of the partner involved in the operations stage configure their assets so that it can interact properly with the other assets during the operation of the CoI.

The exact set of assets and the type of configuration that needs to be done would depend on the physical topology that is required to satisfy the needs of the CoI. In some CoIs, e.g. in the surveillance scenario above, the assets may be interacting directly with each other using a wireless network. In other CoIs such as the training across coalition data sets, the assets may be machines in the infrastructure of each of the partners, and the configuration would require enabling access at some gateways. Regardless of the exact configuration, a set of access control policies need to be given to each of the assets so that the task of the CoI can be performed seamlessly. Another set of policies needs to be defined to deal with the issues of trust in coalition partner's data sources, their use of different communication protocols, or assets of varying capabilities.

D. Canonical Model of an Asset

In real-life there are many different types of assets. In order to consider the security issues and handling of assets in a unified way, we introduce a canonical model for any asset that is used in a CoI. This canonical model allows us to treat the different capabilities of an asset in a consistent manner.

We model each of the assets as being a collection of micro-services [4]. The micro-services architecture requires each system be developed as a set of independent services, where each service implemented in a fault-tolerant light-weight manner and accessed via light-weight efficient protocols. It is an architecture gaining popularity due to its ability to support complex systems in a modular manner.

A data set available at an asset will be represented by its own individual access micro-services. Each micro-service would typically be identified by means of a unique string label which typically would be a substring of the Uniform Resource Identifier (URI) used to access the micro-service. For access to a data set, the components of the URI representing an access request and the data set can be considered as the equivalent identifier string. Each of the assets in the dynamic CoI would also be assigned an address which will be unique in the networking context of the dynamic CoI. Like the URI, the network address would be a distinct and unique identifier that depends on the communication protocol being used. Thus, the access control matrix of the CoI can be viewed as granting or denying access to specific micro-service request coming from specific network addresses.

Assuming the right set of credentials are provided to each of the assets in the dynamic CoI and that, during the planning stage, all have agreed upon a common way to discover each other when operational, the assets belonging to the CoI can discover each other, and set up communication channels among each other. However, in coalition environments, where trust is not absolute among all partners, several security challenges still remain. In the next section, we discuss some of these security challenges.

III. SECURITY CHALLENGES IN COALITION ENVIRONMENTS

There are many security challenges that need to be addressed to enable dynamic CoIs in a manner described in the previous section. The first challenge of course is the proper authentication and discovery of the assets in the CoI. However, assuming that the credentials and the mechanism for discovering other devices have been agreed upon during the planning process, the appropriate solution for authenticating and discovering other assets can be deployed. We discuss the remaining challenges that need to be addressed using the canonical model.

A. Access Policy Determination

If all assets belonged to the same coalition partner and can be trusted equally, the system could set up access control simply by giving each of the devices access to the different micro-services that are running on other devices. However, coalition members may not trust each other completely, and thus may only be authorizing accesses to a subset of the all the micro-services running in the machine. Determining this subset is the next security challenge in supporting dynamic CoIs.

From the perspective of each asset in the CoI, setting the proper security consists of the specification of an allow/deny matrix where one dimension contains the set of micro-services running locally and the other dimension contains the network addresses of other assets in the dynamic CoI. The subset of micro-services that ought to be allowed access depends on several attributes - the type of local asset, the type and ownership of the remote asset, and the nature of the dynamic CoI being the most important ones. When the planning phase is over, the type of the dynamic CoI, the set of assets and their attributes such as type and ownership are known.

B. Micro Service Transformation

Another security challenge is the determination of the right procedures to be invoked on each of the assets before it is put into use by a dynamic CoI. Some assets may need to run in a degraded mode when they are used with another coalition partner. Some types of data may need to be deleted before the asset is allowed to participate in the dynamic CoI. In some cases, some micro-services may not be accessible directly from the coalition partners, even though they should be accessible directly by CoI participants of the coalition member owning the asset. Instead, another micro-service that provides secure access to that resource would be needed. The additional micro-service may also be needed to transform the data that is available before being sent to a coalition partner.

Using our canonical model, each asset is a set of micro-services. Before being used in the CoI, it needs to be transformed into another modified set of micro-services. An existing micro-service may be removed from the device completely, remain visible on the device, or be modified in order to participate in the dynamic CoI. New micro-services may also be introduced as an asset is required to participate in the CoI. A possible

transformation of an asset's set of micro-services as used in the CoI and as used internally is shown in Figure 4.

C. Data Governance for Training and Sensor Fusion

When data is obtained from different partners, the data may not be trusted completely. This lack of trust issue arises in both dynamic CoIs that deal with training of new models using data

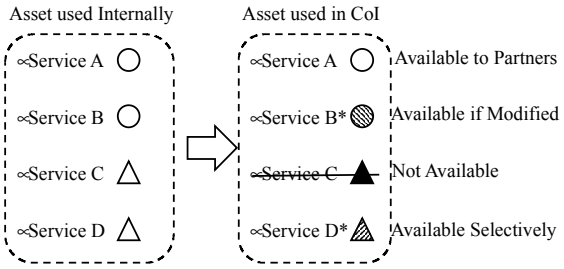

Figure 4: Transformation of Asset MicroServices

from coalition partners, e.g. the training across coalition data sets scenarios, as well as CoIs that deal primarily with sensor fusion applications, where they are using the information from coalition partners to make operational decision, e.g. in the joint surveillance scenario. The data that is not from a completely trusted source needs to be vetted properly before being used for any purpose.

Similarly, data that is going from a coalition partner to others may also need to be sanitized properly. A U.S. UAV may not want to send high resolution images to a partially trusted partner, and in some cases may not even want to reveal the fact that it has the capability to take high resolution images to such a partner. Data needs to be properly transformed in these cases.

Using the canonical micro-service representation, we can consider the data governance issues as a special case of service transformation, where both access to high-resolution and low-resolution images is provided through the same micro-service, but different instances of the micro-service are made available to different partners, depending on the context.

D. Self-Generation of Policies

Since the assets in a dynamic CoI belong to different countries, the communication permissions between them need to be set up by the network administrators of different countries. However, such expertise may be hard to obtain during operations, and connecting devices from different countries may require a long manual delay in the reconfiguration of the network to allow dynamic CoIs to be created. If the model training process takes 10 hours of computation, and the network reconfiguration process requires a 2-day wait, the CoIs may not be very dynamic. Therefore, we need to have an automated mechanism which can perform the task of setting up the network configuration as safely and securely as a human network administrator.

Similarly, the task of determining the white list (or equivalent black list) and the mapping of micro-services will typically be done by a human taking into account the requirements of the CoI and the nature of the assets. Given the lack of qualified personnel who can do it in the field, this task needs to be automated so that each device can learn the right policies to be used for the CoI on its own.

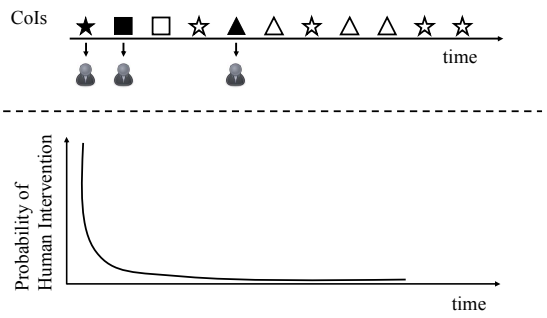

Figure 5: Automation Approach

IV. AUTOMATION OF SECURITY POLICIES

The approach we propose for automation is to use machine learning to enable assets to reduce the overhead on humans to define security policies that enable secure dynamic CoIs. These include determining access control policies and the right service transformations including those transformations required for data governance. The data to train the learning algorithm is obtained by observing the actions of humans in the field when they deal with similar CoI establishment requests. While the initial steps of configuring the CoIs are done by a human administrator, an automated agent learns how to do the equivalent task using machine learning approaches.

A. Automation Approach

The basic approach for automation is illustrated in Figure 5. It shows a time-line for an asset to be involved in different dynamic CoIs. The type of CoI is shown by the shape. When an asset is deployed for the first time in a dynamic CoI, it needs a manual configuration to define the appropriate security policies it needs. In all subsequent deployments, the asset tries to determine those policies on its own based on the policies which were used in previous deployments. In the figure, the CoIs where human intervention is needed are shown as dark shaded shapes while the light shapes depict ones where the asset can determine the policies on its own. As time passes on, an asset will be more capable of determining its own policies, and the probability for human intervention decreases. This can be viewed as a type of online learning where the training data comes from the involvement of the asset in different CoIs.

When an asset is needed for a CoI, it is provided (via a standard micro-service interface) a description of the CoI. This description should include the type of the CoI, the set of other participating assets, and information about each asset such as its owing country. From this description, the asset needs to determine two tables, one is a mapping table that determines how each of the micro-services available internally ought to be transformed so that it becomes usable in the CoI. The second is a mapping table that lists the set of access control policies (e.g. white list) of the transformed micro-services so that they can be accessed by the other assets in the dynamic CoI.

The first table (mapping table) can be viewed as consisting of four columns, the first listing the type of CoI, the second listing the micro-service used internally, the third listing the coalition partner, and the fourth as the new micro-service that will be visible to the partner. For services that will not be visible, a default name can be used.

The second table (access table) can be viewed as consisting of four columns, the first listing the coalition partner, the second listing the type of partner asset, the third listing the name of the transformed micro-service in the mapping table, and the fourth listing whether the access is allowed or not. Either a white list approach or black list approach for specifying the model can be used.

The experience data contains the resulting mapping tables and access tables for previous CoIs about which the asset has the knowledge. To learn the mapping table, we can consider only that subset which was used for the same type of dynamic CoI and the same coalition partner in the past from the experience data. For each of the micro-service that is available internally, the subset would show that micro-service mapping into one of four categories, either being exposed as-is, being hidden, being deleted, or being exposed as a transformed service.

For each micro-service, we formulate a machine learning problem where the training data consists of the attributes of the CoI in which the asset was involved, including features such as the other participating coalition members, the types of assets being brought by each of the coalition members etc. The output in each of these problems is a selection among one of the four categories described in the previous paragraph. The machine learning process can be viewed as the process of learning the mapping of the micro-service to one of the categories. If we consider different options for transformations as independent categories, the number of categories may increase slightly, but would remain in a small number.

If we consider the access table, the appropriate machine learning problem would be defined for each of the transformed services, and each participating asset type from each coalition nation. Each micro-service is mapped into one of two categories – allowed access, or denied access. The training data would consist of the attributes of the CoI, as before.

Let us consider the learning problem formulation for the two scenarios discussed in Section II. In the formulation, we assume that the coalition consists of 5 nations cooperating with each other.

B. Training across Coalition Data Sets

When training is used across multiple coalition partners, each asset is one of three types, a data store, a training computer, or a firewall. For each data store, the CoI is defined by a binary attribute for each of the other coalition members, as to whether that member is participating or not, a binary attribute for each training computer per coalition member, as to whether or not that member is providing a computer for the training task, and whether or not a firewall from the same coalition member is present in the environment. With 5 coalition members, each data store characterizes the CoI by means of 9 binary variables. For each training computer, the set is augmented by another binary variable indicating whether or not a data store from the coalition partner is used, resulting in 14 binary variables.

For each micro-service on the data store asset, the CoI results in transforming it into one of the four categories, exposed as-is, being hidden, being deleted, or being exposed as a transformed service. The same holds true for the training computer assets. The learning problem for the data store asset is to map a binary vector with 9 attributes into one of 4 categories, while that for the training data store is to map a binary vector with 14 attributes into one of 4 categories.

If we consider each of the machine learning problems that are involved above, we see that they are trying to partition a space into 4 categories. The different axes of the space happen to be restricted to be binary (presence or absence of specific coalition partners, presence of absence of specific types of assets from specific partners). The new CoI for which decision is to be made defines one specific combination of these values. The selection of the right choice of the category depends on what those combinations show for the training data.

C. Surveillance across Partners

For surveillance across partners, assets may be of different types. Depending on where the analysis server is, each asset may decide to expose a micro-service on the service to the analysis server or not. The other attributes that will determine the decision are the types of members that are participating and the types of assets that are being contributed by each of the coalition members.

Suppose there are 4 types of assets that are involved for each of the 5 coalition nations. For each asset, the decision consists of 5 binary values as to whether the analysis service is running in one of the coalition partners, and 20 (4 x 5) binary values as to whether or not the coalition partner is contributing a specific type of asset. This results in a CoI characterization of 25 binary variables, which are mapped into a binary decision to expose or not to expose a micro-service.

D. General Formulation

Extrapolating from the two exemplar scenarios, we can define the general problem of learning of policies in dynamic CoI as machine learning in a feature space consisting of N-dimensional binary vectors. The N-dimensional binary vector space which has some nice mathematical properties [5], specifically that it can be considered as a mathematical field. This allows for the definition of a cosine distance among binary vectors. Another unique property is that the binary vector space is a finite space, and any point in the space can be mapped onto a natural number less than 2^N. Working with finite spaces simplifies the problem of classifying and can lead to more efficient algorithms for machine learning.

Given the unique characteristics of this learning problem, the fact that the input features are a binary vector with a large number of dimensions, and the output is a very small number of classes, we can use these characteristics to develop efficient learning approaches to determine the category for a new dynamic CoI that is being formed. Specifically, we can model this machine learning problem as a function which takes in any

one of 2^N values, each corresponding to a binary vector in the feature space, and produces an output category, which is one among a handful of choices. This learning algorithm can be modeled as trying to learn a discrete relationship, and we can study the properties of discrete relationships to understand how well machine learning will work in such environments. Specifically, the performance of the machine learning algorithm depends on determining how much training data is needed before the asset can determine its own policies.

V. AMOUNT OF TRAINING DATA NEEDED

The performance of the machine learning algorithm depends upon the manner in which categories are structured along the feature space defined by the various binary vectors in the feature. In this section, we discuss how much training data would be required to get a reasonable accuracy in the prediction of the required category for the learning of policies based on previous human input. In order to do so, let us examine the model for machine learning process in general.

A. Model of Machine Learning Process

We can model the machine learning problem as that of learning a ground truth function given a set of training data. Both the ground truth and training data are mappings from a finite discrete set to another finite discrete set. The training data consists of a number of mappings that are defined as the asset gets involved in different CoIs.

The machine learning process implemented by each asset can be modeled as the sequence of steps shown in Figure 6. There is ground truth – which is the function that characterizes

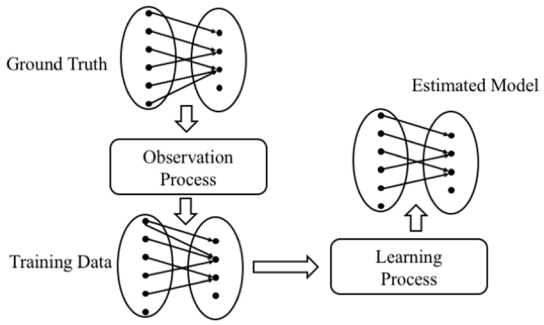

Figure 6. Abstract model of the machine learning process

the policies a human will specify. By observing the human behavior, each asset creates a set of training data. The training data creation process may be noisy, in that it may introduces edges that may not be present in the ground truth, or remove some edges that may be present in the ground truth. The noise may come because human administrators can make mistakes when specifying the required policies for a CoI. The asset uses the training data to learn an estimate of the ground truth.

The precision and recall metrics for effectiveness of machine learning algorithms can be determined in terms of the difference between the ground truth and the estimated model. If we define A as number of edges in ground truth, B as number of edges in the estimated model, and C as the number of edges present in both, then the precision of the model is C/B and the recall is C/A.

B. Noiseless Observation process

We analyze the relationship between training data and the precision/recall of the learnt model by making various assumptions about how the training data is constructed. One way is to assume that the observation process consists of selecting an edge from the ground truth randomly. As the system is observed, each observation consists of one edge in the ground truth. In this case, the training data is a collection of *K* edges from the ground truth in which the edges are randomly selected at each step with repetition. Note that with the noiseless observation process, the machine learning step can just pick up the edges from the training data to build the estimated model.

Let the domain have *D* distinct elements and the range have *R* distinct elements. In this case, the ground truth consists of one edge per element in the domain, i.e. some *D* edges out of a total number of possible *D.R* edges. As a starting point, let us consider the case where the observation process follows a uniform distribution in selection of the edges in the ground truth. In this case, each step in the generation of the training data set selects one of the *D* edges in the system with a probability of *1/D*.

In the absence of noise, all edges selected in the estimator will be the ones in ground truth, and the precision of the model would be 100%. The recall would be the expected value of the number of edges that are captured in a training data set of size *K*. For a 100% recall, a necessary condition is that the training data be larger than the size of the domain ($K > D$).

In order to get an estimate of the recall expected on a training set of size K, we note that each edge has a probability of $1/D$ of getting picked at any step of the observation process, so the probability that a given edge has not been selected after K steps is $(1 - 1/D)^K$. The expected number of missing edges would be $D(1 - 1/D)^K$. Therefore, the expected recall of a training data set of size K would be $1 - (1 - 1/D)^K$.

Another useful metric is to determine the size of training set which results in 100% recall. We note that the process for creating the training data is identical to a classical problem in combinatorics – namely the coupon collector problem [9]. Each edge in the ground truth can be viewed as a coupon which is collected with the probability of *1/D* on each turn. From the results on the coupon collector problem, the expected size of the test data set required to get a 100% recall of *D* distinct items would be approximately $D.ln(D) + \gamma D$ where γ is the Euler–Mascheroni constant with an approximate value of 0.5772156649.

Furthermore, it is known by the results to the coupon collector's problem that the probability that more than $\beta.D.ln(D)$ trials will be needed is bounded by $D^{1-\beta}$ (where $\beta > 1$). It follows that if we want to keep the probability of complete recall to be greater than 1-ε, then we need to have a training data size of *K* which satisfies:

$$K \geq Dln(D) - D\ln(\varepsilon)$$

Conversely, it follows that if we have a training data set of some fixed size *S* and we would like a 100%% recall, then the domain of the function that we are trying to model should not be more than *d*, where the equation $d.ln(d) + \gamma d < S$ holds true.

C. Noisy Observation Process

When the observation process is noisy, spurious edges that may not be in the ground truth may be introduced in the training data, and the precision of the model will no longer be 100%. The learning process will need to have a mechanism to decide whether an edge in the training data is spurious and reject them.

Let us assume that there is a probability η of introducing noise in the training data. One can model the creation of a training set with noisy data as a process of repeated selection where at each step, the process selects an edge randomly from the ground truth with a probability of $(1-\eta)$ and, with the probability of η, selects any of the possible $D.R$ edges that are possible between the domain and the range.

From the point of analyzing the recall of the model that results with noise in the observation, a training set with K edges from a model resulting from the noisy process can be considered as equivalent to a noiseless training data with $(1-\eta)K$ edges. Accordingly, it follows that if we want to keep the probability of complete recall to be greater than 1-ε, then we need to have a test data size K which satisfies:

$$(1-\eta)K \geq D\ln(D).-D\ln(\varepsilon))$$

Conversely, if we have a test data set of size S and we know that the training process was noisy, and the probability of noise in the training data is p, then the domain of the function that we are trying to model should not have more than d elements, where the equation $d.ln(d) + \gamma d < (1-\eta)\ S$ holds true. If we assume that η is less than 0.5, then we want to satisfy the minimum relationship that $d.ln(d) + \gamma d < S/2$.

With noisy training data, the precision is no longer 100%. The reduction in precision would depend on both how noise ends up introducing spurious edges, and also how the learning process is picking up the edges for the estimate.

For our specific problem, we can assume that each element of the domain has exactly one edge to an element in the range, and that each element in the range has a mapping from at least one element in the domain. The algorithm used during the learning process would select precisely one edge from each element in the domain. The most logical choice for choosing the edge would be to select the edge from each element that has the most frequent occurrence in the training data set.

With the noise process as introduced, let us consider the edges coming out of each node. On each trial of the observation process to create a new point in the training data set, an edge starting with that node would be drawn from a binomial distribution that picks either the right edge in the ground truth or picks a wrong edge. The probability of picking the right edge is $((1-\eta) + \eta/D)$, and the probability of picking any one of the wrong edge is $(\eta(D-1)/D)$. The assumption here is that the noise process has a chance that it may pick up the right edge as well. If the overall training process results in K edges, the expected number of edges that are picked up with a given element in the domain will be K/D. The probability that the learning process picks any wrong edge in the estimate is the probability that after the entire process finishes, one of the wrong edges ends up getting picked up more frequently than the right edge.

Let us define $B(Q,p,q)$ as the probability that in Q trials where an event $e1$ can occur with probability p, and the event $e2$ can occur with probability q, $e2$ occurs more often than $e1$. For any given edge in the ground truth, it may be incorrect in the learnt model if it is not the most frequently occurring one. This probability will be given by $pr_i = B(K, ((1-\eta) + \eta/D), (\eta(D-1)/D)$.

And the overall precision of model will be $(1- pr_i)^M$.

An approximate expression can be obtained for $B(Q,p,q)$ when Q is large and p, q are relatively small by using the Poisson approximation for Binomial distributions. In this case, $B(Q,p,q)$ is approximately $e^{-Q(p-q)}$. This leads to an approximate value for the precision of the model overall to be $1-D\ e^{-Q(1-2\eta)}$.

VI. REDUCTION OF TRAINING DATA

As discussed in the exemplars, the domain of the learning function in the examples is binary vectors with 10-20 dimensions. These would require training data based on experience of 1000+ CoI engagements. It is unlikely that a single asset may be engaged in such a large number of CoIs. Therefore, we need approaches by which we could reduce the requirement to have training data in order to make the approach practical. Some of the possible approaches are discussed below.

A. Sharing of Previous Experience

As an organization, each member of a coalition is likely involved in several dynamic CoIs than any individual asset is likely to encounter in its lifetime. Any coalition nation is likely to have hundreds of any type of assets (UAVs, sensors of different types, computing and networking equipment etc.), if not more, and each asset is likely to be used for only a few of those dynamic CoIs. If the experiences from each dynamic CoI were to be saved and be accessible to other assets from the same coalition member, each asset can learn from the previous experience and can use that for its learning experience.

In order for this approach to work, each asset involved in any dynamic CoI would need to publish a description of its engagement, including the type of CoI it was, the other participating nations, and security policies that were used for that particular engagement. If this information can be stored securely, and retrieved from the secure storage for access, the number of previous data points to use in the learning algorithm can be increased.

One approach to share this knowledge can be found in community knowledge sharing system for policies (e.g. [6]), although the focus on that work is on sharing details on how a procedure can be repeated across multiple assets. Nevertheless, the same approach can be used to share experiences from previous dynamic CoIs that different assets have been involved in, thereby leading to a larger training data set.

B. Types and Categorization

Another approach to increase the number of training data points available to each asset is to group different features into larger equivalence classes or groups. This can be viewed as a variation of the approaches that try to reduce the dimensions required for machine learning [7]. Instead of having several

binary feature vectors, one can consider several of those features as equivalent.

As an example, let us consider coalition operations that involve several countries of NATO. In different dynamic CoIs, different countries may participate, and the trust relationship among each country and a selected country (e.g. U.S.) may be different. However, several of the partner countries may be considered equivalent from a trust perspective, e.g. the U.S. asset may consider participating with a UK asset in a dynamic CoI as being equivalent to participation in a dynamic CoI with a French asset. When such equivalence classes can be defined, several of the features can be collapsed into a single one.

In the case of an asset learning from its prior experience, this means that an asset which has been involved in a mission with one NATO nation can use that to learn and determine its policies for a future dynamic CoI with another NATO nation.

C. *Learning Higher Abstractions*

An approach that can further simplify the task of collecting training data is to move the policy specification level up a level of abstraction. Instead of an asset trying to identify the access control and security policies associated with dynamic CoI, it can try to identify a higher-level abstraction associated with policy generation and refinement.

Current research in policy-based management is exploring the concept of generative policies [8], in which devices are allowed to generate their own policies by means of higher layer abstractions that are provided to them by a management system. The higher-level abstractions take the form if an interaction graph [8], which specifies the different roles and visible attributes of the devices in different roles, and a policy generator which is a template that enables devices to generate policies. The graph is used to discover and determine the attributes of different devices in the system and can represent several devices at the same time. Given these two abstractions, a device can generate its own policies for security and access control.

Since each CoI requires learning roles and relationships, i.e. whether a defined role or relationship exists in one type of device, the nature of the machine learning problem, mapping binary vector spaces to a domain which has 2 values (present/not present) remains unchanged. Therefore, the analysis of required training data points would still be applicable.

An attempt to learn the higher-level abstractions instead of the access control policies themselves has several advantages in reducing the training data size that is involved. This learning will need to be done at the level of the management tool as opposed to the asset level, so the learning can be applied by taking into account all the dynamic CoIs that a coalition member is involved in. This is more likely to result in obtaining a larger number of training data points. Furthermore, different number of devices from different partners are all aggregated into a single role. As a result, the number of additional dimensions in the typical vector space can be reduced significantly.

VII. CONCLUSIONS

In this paper, we have looked at the security challenges involved in creating dynamic infrastructure for sensor fusion in coalition environments. Given the dynamicity of CoIs that are needed, we have argued that the determination of security challenges be done by means of defining the appropriate security policies. We have proposed a machine learning based approach for identifying the security policies in the environment, where the machines learn from human defined policies, and gradually are able to determine their own policies. We have explored the amount of training data points that will be required to make the machine learning approach viable in practice and identified schemes by which the challenges associated with obtaining the minimum required amount of training data can be addressed.

Our next step in the investigation is to generate synthetic traces for dynamic CoIs that may be expected in coalition settings and understand the empirical performance of different machine learning algorithms on those synthetic traces.

REFERENCES

[1] E. Asmare, N. Dulay, E. Lupu, M. Sloman, S. Calo and J. Lobo, "Secure dynamic community establishment in coalitions," Proc. IEEE Military Communications Conference, IEEE MILCOM 2007, Orlando, FL, pp. 1-7, October 2007.

[2] V. Mishra, D. Verma, C. Williams and K. Marcus, "Comparng software defined architectures for coalition operations," Proc. IEEE Int. Conf. Military Communications and Information Systems, ICMCIS-2017, Oulu, Finland, pp 1-7, May 2017.

[3] V. Mishra, D. Verma and C. Williams, "Improving Security in Coalition Tactical Environments Using an SDN Approach", in Guide to Security in SDN and NFV, S. Zhu, S. Scott-Hayward, L. Jacquin and R. Hill, Eds, Springer, 2017, pp. 273-298.

[4] J. Thönes, "Microservices," IEEE Software vol 32, no. 1, pp. 113-116, January 2015.

[5] K. Arai and H. Okazaki, "N-Dimensional binary vector spaces." Formalized Mathematics vol 21 no 2, June 2013, pp 75-81.

[6] E. Bertino, G. de Mel, A. Russo, S. Calo and D. Verma, "Community-based self generation of policies and processes for assets: Concepts and research directions," IEEE International Conference on Big Data Boston, MA, 2017, pp. 2961-2969.

[7] I. Fodor, "A survey of dimension reduction techniques," Technical Report No. UCRL-ID-148494, Lawrence Livermore National Lab., CA (US), 2002.

[8] D. Verma, S. Calo., S. Chakraborty, E. Bertino, C. Williams, J. Tucker, & B. Rivera, “Generative Policy Model for Autonomic Management”. In 1st Int. Workshop Dist Analytics Infrastructure & Algorithms, IEEE Smartworld Congress, August 2017.

[9] A. Boneh and M. Hofri, “The coupon-collector problem revisited—a survey of engineering problems and computational methods,” Stochastic Models, vol. 13; no. 1, pp. 39-66. Jan 1997.