2019 IEEE International Conference on Smart Computing (SMARTCOMP)

# Generative Policy Framework for AI Training Data Curation

V. Salapura[1], D. Wood[1], S.A. Witherspoon[1], K. Grueneberg[1], E. Bertino[2], A.A.Jabal[2], S. Calo[1]
[1]IBM T.J. Watson Research Center
Yorktown Heights, NY-10549
{salapura, dawood, Shonda.Adena.Witherspoon, kgruen, scalo}@us.ibm.com
[2]Purdue University, Computer Science Dept.
West Lafayette, IN-47907
{bertino,aabujaba}@purdue.edu

*Abstract*— **Policy-based mechanisms are used to implement desired autonomic behavior of a managed system in a distributed environment. For modern dynamically changing systems, policy-based mechanisms tend to be too rigid, and quickly lose their efficacy when conditions of the autonomous system change during its operation. In this paper, we propose a generative policy framework that can generate policies for an autonomous system when conditions change. For changed conditions, the policy generation manager dynamically generates new set of policies optimized for the new situation. As a use case, we demonstrate how our generative policy framework generates policies for selecting optimal data for an AI model training. The policies are dynamically generated based on the availability and trustworthiness of data in a coalition environment.**

***Keywords—generative policy framework, context aware policies, automatic policy generation, autonomous managed systems***

## I. Introduction

Policy-based management is an approach that can simplify the complex task of managing distributed systems. Under this approach, an administrator deploys a set of policies that govern the behavior of a distributed system. Policy-based management can increase significantly the self-managing aspects of any distributed system or network, leading to more autonomic behavior. For its efficacy in simplifying the management and operations of complex distributed environments, policy-based management found its use in a number of different domains, such as computer networks, storage area networks, industrial systems, enterprise access control, security and identity management, military sensor networks, and many other areas.

For modern dynamically changing systems, policy based mechanisms tend to be too rigid. When conditions of the operation of the autonomous managed system change – for example, by adding new participants to a computer network – the specific set of policies can become sub-optimal, inefficient, or incorrect. For dynamically changing environments, dynamic policy generation is needed to adapt the autonomous system and its ruling policies for new context.

An example of one such environment involves training an Artificial Intelligence (AI) model. As AI models are becoming ubiquitous in virtually all aspects of future technology, it is critical that the AI models are robust, reliable and adaptable to dynamic situations. Using trusted training data is the key requirement for training a high-quality AI model [1]. The robustness and accuracy of the AI model is highly dependent on the quality of the data used for training. It is advantageous to use the training data available from different sources to create a larger and more versatile training data set. Coalitions of various partners can be formed, resulting in larger more robust sets of data from different sources. The acquisition of training data is usually the most laborious and time-consuming part of the entire process for creating an AI model; this is where policies come in.

In this paper, we introduce a framework for generating context-aware policies. The proposed framework dynamically detects when training data is available, or has changed, and generates accordingly a new set of policies. We demonstrate the capabilities of our context aware policy generation manager on a data curator with policies for accepting trusted data, as introduced in [2]. We use our framework to accept training data for an AI model based on trust for a dynamically changing context scenario.

## II. Generative Policy Manager Architecture

Policy-based Management Systems typically use policies that are provided by administrators, or experts on the system's operations. Defined policies are usually static, and not altered, until new policies are provided to the system. Manual policy creation and modification is tedious and error prone. Recently, approaches for automatic policy generation have been proposed [3]. However, these approaches are focused on static policies, i.e., predefined policies that do not change over time.

The Generative Policy-based Model (GPM) [4], proposes the concept of generative policies i.e., policies that are self-generated by the managed system, or devices. This allows flexible policy definitions that make managed systems more autonomous. A generative policy model learns from an initial set of policies of the managed system, and from the analysis of the system's operations, resource usage, and previous policy decisions.

In this paper, we propose a framework for the Generative Policy Manager (GPM) system. The framework allows a managed system to generate a set of optimal policies for its autonomous operation dynamically, adapting to a new context. The architecture of our Generative Policy Manager (GPM) architecture is shown in Figure 1. The main components of the GPM are described below.

### *A. Management System*

The management system provides a plan for the autonomous managed system (AMS), a set of parameters to measure, and a

Work funded by U.S. Army Research Laboratory and the U.K. Ministry of Defense

978-1-7281-1689-1/19/$31.00 ©2019 IEEE
DOI 10.1109/SMARTCOMP.2019.00092

goal defining when the AMS objective is achieved. It also provides a set of policy templates (templates for the possible policies which will be generated).

The AMS system takes as inputs policy templates and attributes to evaluate for determining the status of the system, as well as an objective value. Policy templates are stored in the Policy repository.

### *B. AMS – Autonomous Managed System*

The AMS system implements the main part of the Generative Policy Manager. The context aware policy instantiation module takes a set of policy templates in the repository as input, and the current context, as captured in the context storage, and generates a set of current policies for the AMS. These policies are stored in the Policy Repository.

When the context of the AMS changes, as detected by monitoring the current state of the system, the new status is captured, and the policy instantiation module processes context updates through policy adaptation, and instantiates new policies to meet new business objectives. The new set of policies is deployed in the policy repository replacing the existing policies.

Another important component of our dynamic context aware framework is the context-aware policy template adaptation module. This module evaluates if the business goals are met, and modifies policy templates if necessary, in order to meet the goals.

Finally, the remaining component of the AMS is the control system itself, which applies the generated policies on the system that it is managing. It consists of a policy decision point (PDP) and a policy enforcement point (PEP) for the given set of policies. The PDPs select the policies, and decide which ones are applicable for the current data. The PEP is used to enforce the policy decisions. Many PEPs may be controlled by a single PDP, and a single Policy Repository may be used to distribute policies to many PDPs.

## III. Trust Based Policy Generation and Data Curation

We demonstrate the operation and versatility of our Context Aware Policy Generation Framework on selecting data for AI model training. In this use case, data for training the AI model originates from multiple countries of a coalition, and US is responsible for training the AI model. Not all coalition partners are trusted equally. Since there is not enough data from US alone, data from other partners are added for model training, with the goal of having a balanced set of examples for all classes. To express the level of the quality of data from various sources, we assign a trust value to each coalition country. For the AI model training, usage of data with a higher trust score is preferable. The trust level for some coalition partners might be too low, in which case we would not accept any of their data. The method to determine the trust level of a coalition partner is beyond the scope of this paper. Thus, we will select the data from coalition partners based on their trust level, consuming highest trust data first, then moving to a lower level, and so on, until we reach a sufficient amount of data for training the AI model, or we use up all data available, whichever comes first.

The training data is labeled for a set of classes. There is a different number of data available in each of the classes. For the purpose of this use case, our AI model will be trained to pattern match 10 different classes, labeled 1 to 10. The coalition has four participating countries, US, UK, Japan and Kish. The trust for the countries, from the highest to lowest is the following: US, UK, Japan, and Kish. All coalition partners are sufficiently trusted to be used for the AI model training. The number of data per class per country is illustrated in Figure 2.

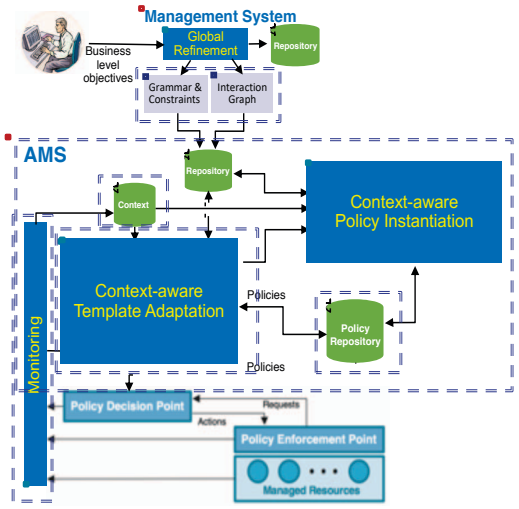


Fig. 1. Generative policy manager - architecture.

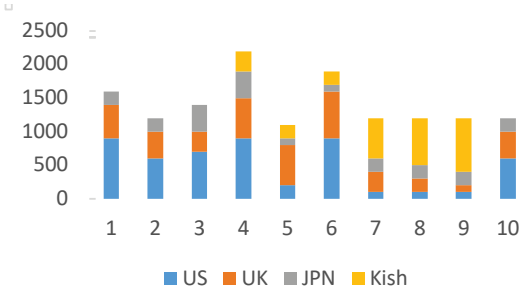


Fig. 2. Amount of data available from coalition partners.

We use two policy templates:

1) if data is from *{source country}* and the label is *{label}* then accept the data
2) if data is from *{source country}* then accept the data

The first policy template accepts data selectively depending on its label and country of origin; the second accepts all data from a country indiscriminately.

Initially, policies are generated for the current context, determined by data availability and trust level of the originating country. However, when the context changes, the system detects it and responds accordingly. Examples of context change are for example, when more countries are added to the coalition and publish data, or when the trust level of a country in the coalition changes, or the amount of data of a country changes. The framework automatically generates policies optimized for the

new context using the provided templates and stores them in the policy repository.

For our use case of trust-based policy generation and data curation, we run several scenarios, as described below:

### A. *Data available from US, UK and Kish only*

The policies are generated by using the policy template (2) for US to accept all available data from US, which has the highest trust level (as they are the source of curation). Similarly, we accept all UK data up to a maximum number of sample data needed. From Kish, which has the lowest trust level in this coalition, we use only data for the classes where more data is needed, as shown in table I.

TABLE I. INITIAL SET OF POLICIES

| *Country* | *Template* | *Accept class* |
|---|---|---|
| US, UK | 2 | All |
| Kish | 1 | 5, and 7 to 9 |

### B. *Japan publishes data*

When a new country publishes its data, the context changes and the framework generates new policies for the new context. For example, when Japan publishes its data, it has a higher trust level than Kish. Preferably, we would use data from Japan over Kish, when available, for the data curation system. In our scenario, we add classes 5, 7, 8 and 9 for Japan, and reduce accepted data from Kish to classes 7, 8 and 9, as shown in table II.

TABLE II. MODIFIED SET OF POLICIES

| *Country* | *Template* | *Accept class* |
|---|---|---|
| US, UK | 2 | All |
| JPN | 1 | 5, 7 to 9 |
| Kish | 1 | 7 to 9 |

### C. *UK publishes more data*

The context also changes if a country publishes more data. More data from a country with higher trust can cause a different set of policies to be generated. For example, if UK publishes additional data per class for all classes, this will trigger a change in the policies being generated, as the number of data to accept from all sources will change, as listed in table III.

TABLE III. POLICIES FOR DATA INVENTORY INCREASE

| *Country* | *Template* | *Accept class* |
|---|---|---|
| US, UK | 2 | all |
| JPN | 1 | 7 to 9 |
| Kish | 1 | 8 to 9 |

The resulting data mix for the AI model training is illustrated in Figure 3 for the three scenarios described.

## IV. CONCLUSION

We presented a generative policy framework that dynamically generates new set of policies for an autonomous system when operation conditions change. To compensate for changing environment conditions, a new set of policies are dynamically generated, which are optimized for the new situation, which we demonstrated can improve AI training environments. Future work would focus on generating policies for other dynamic environments and observing its impact on the system.

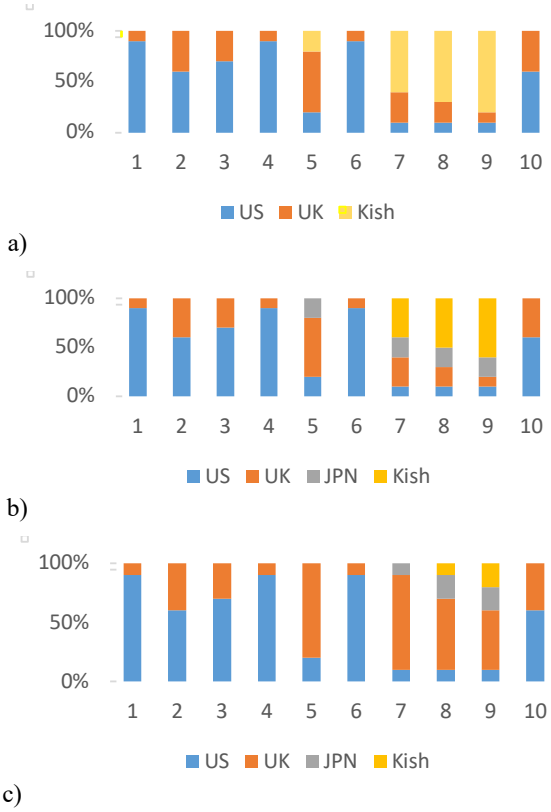


Fig. 3. Data mix generated by dynamically generated policies of the data curator as a response to context change: b) adding new partners, or c) new data with higher trust.


## 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] C. Cortes., L. D. Jackel, and W.-P. Chiang, "Limits on learning machine accuracy imposed by data quality," in Advances in Neural Information Processing Systems, 239-246 (1995).

[2] D. Verma, S. Calo, S. Witherspoon, I. Manotas, E. Bertino, A.A.Jabal, G.Cirincione, A.Swami, G.Pearson, G.de Mel. "Self Generating Policies for Machine Learning in Coalition Environments" 2018.

[3] A. Quiroz, M. Parashar, N. Gnanasambandam, N. Sharma, "Autonomic policy adaptation using decentralized online clustering." In Proceedings of the 7th International Conference on Autonomic Computing, ICAC, pages 151–160. ACM, 2010.

[4] D. Verma, S. Calo, S. Chakraborty, E. Bertino, C. Williams, J. Tucker, B. Rivera. "Generative policy model for autonomic management," 2017 IEEE Advanced & Trusted Computing, 2017.