I·C·S
The Institute for Cyber Security

C·SPECC
Center for Security and Privacy Enhanced Cloud Computing

Toward Deep Learning Based Access Control

Mohammad Nur Nobi, Ram Krishnan, Yufei Huang*, Mehrnoosh Shakarami, Ravi Sandhu

University of Texas at San Antonio

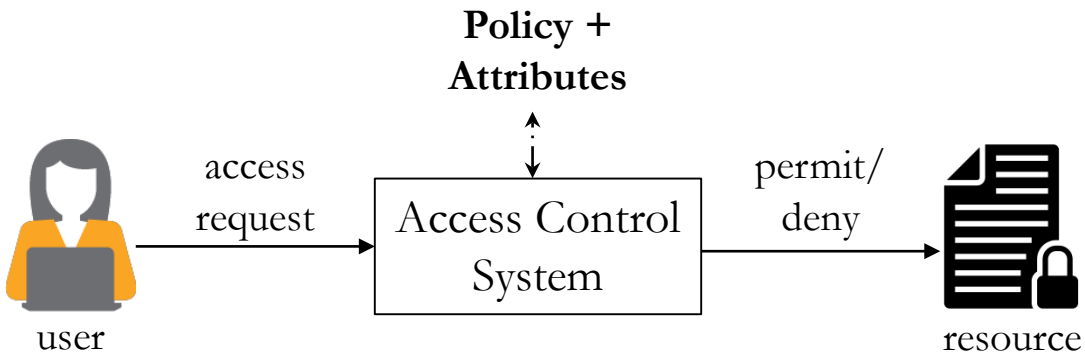
*University of Pittsburgh

April 26, 2022

UTSA Computer Science

Introduction

- There are many mainstream approaches for access control
  - Access Control Lists (ACLs), Role Based Access Control (RBAC), Attribute Based Access Control (ABAC), Relationship Based Access Control (ReBAC), etc.
- These approaches have their benefits and numerous advancements
- Skilled security administrators needed to engineer and manage accesses
  - **Over-provisioned** to ease administrative burden
  - **Under-provisioned** for the sake of tightened security



A Sample System

Access Control State

<Alice, service1, read>
<Alice, service2, {read, write}>
<Bob, service1, {read, write}>
<Bob, service2, {read}>

Authorization Tuple

<User, Resource, {Operations}>

User Metadata

User	spending	join_date	credit_score
Alice	high	Nov 2010	700
Bob	high	Jul 2015	500

Resource Metadata

Resource	limit	process_time
service1	10K	1 day
service2	5K	1 week

Phase-1: Attribute Engineering

User metadata

User	spending	join_date	credit_score
Alice	high	Nov 2010	700
Bob	high	Jul 2015	500

Resource metadata

Resource	limit	process_time
service1	10K	1 day
service2	5K	1 week

Attribute

User	spending	C_status
Alice	high	platinum
Bob	high	silver

User Attributes

Attributes Assignment

Resource	S_status
service1	premium
service2	regular

Resource Attributes

Phase-2: Policy Engineering

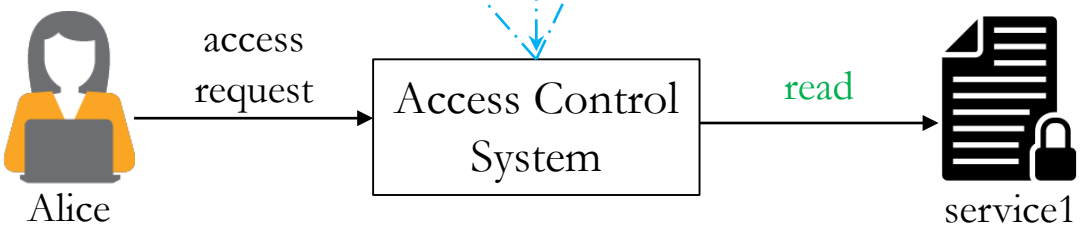
User	spending	C_status
Alice	high	platinum
Bob	high	silver

User Attributes

Resource	S_status
service1	premium
service2	regular

Resource Attributes

Rule: <spending = high, C_status = platinum, S_status = premium> = *read*

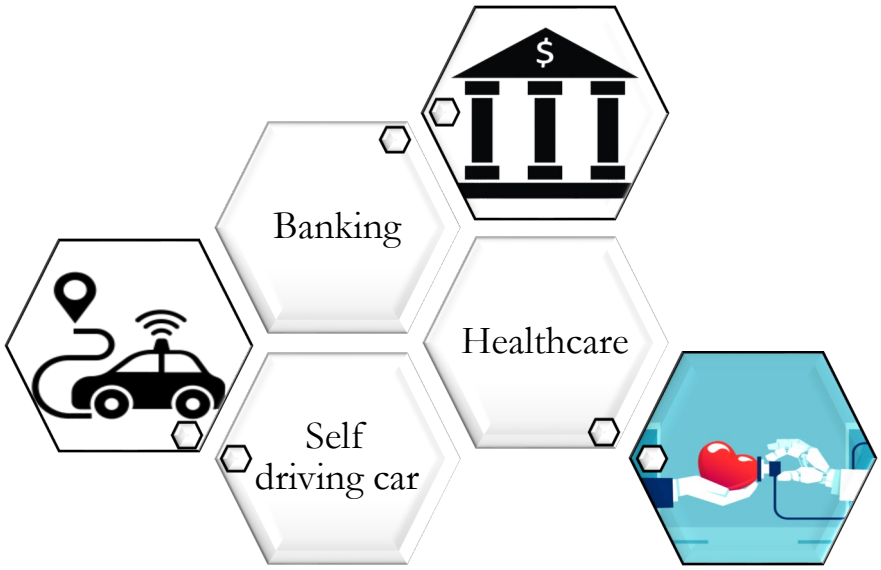


Access Control State

<Alice, service1, {read}>
<Alice, service2, {read, write}>
<Bob, service1, {read, write}>
<Bob, service2, {read}>

Generalization !

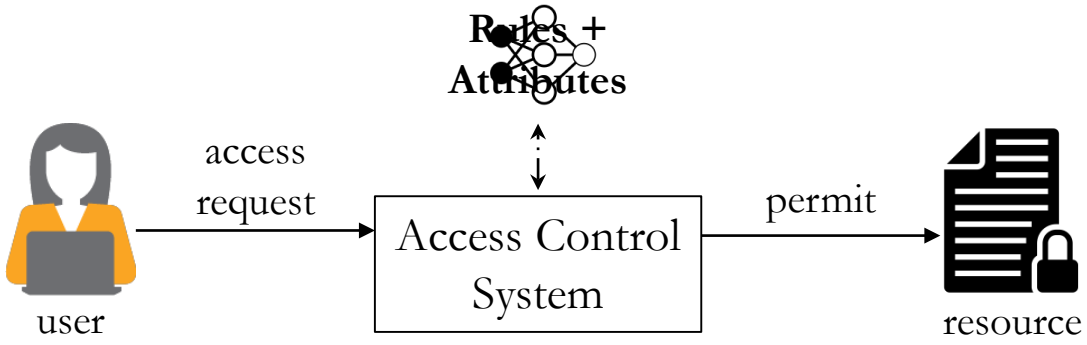
Deep Learning



- Learn by example
- Learn directly from the raw data (no feature extraction)
- Obtain an excellent accuracy

Deep Learning in Access Control

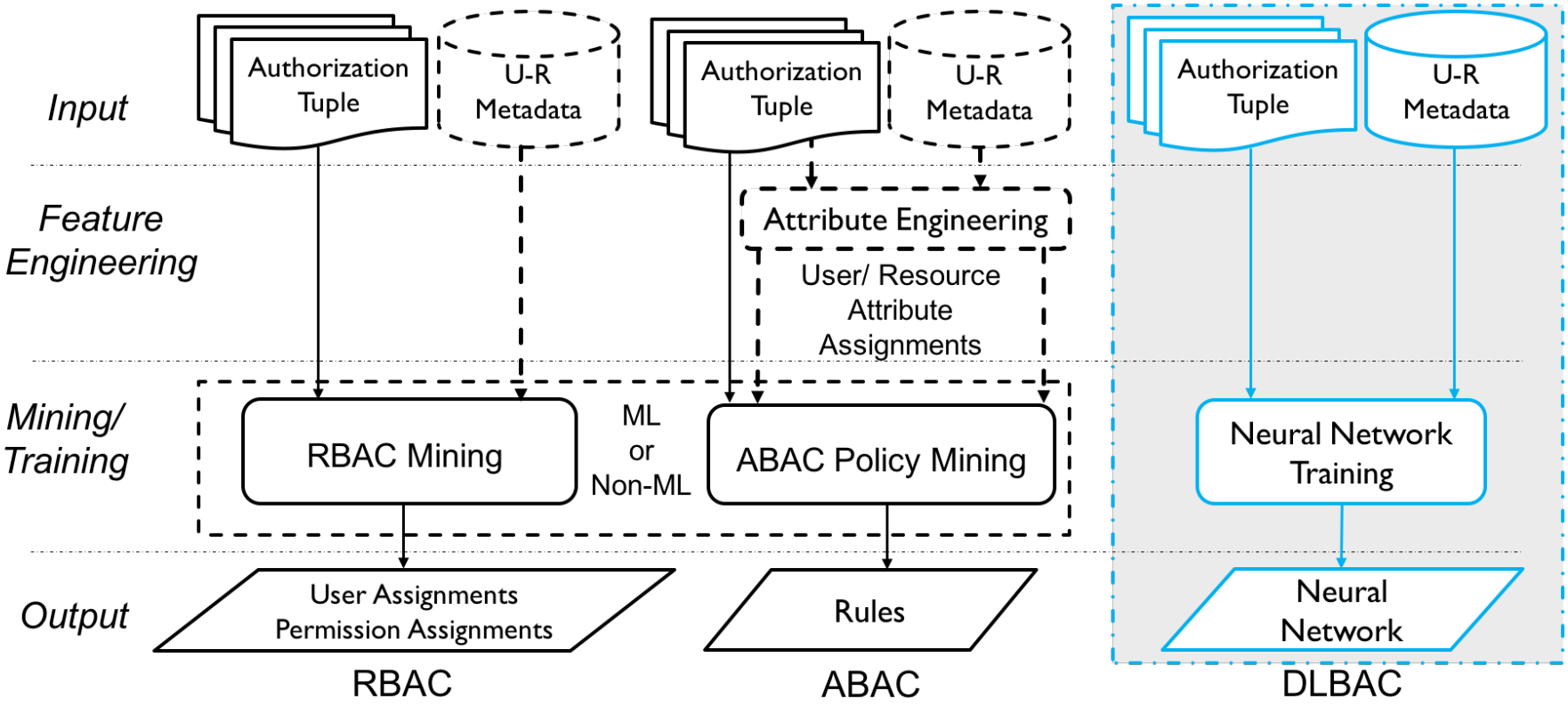
- Could it learn from existing access control state of the system?
- Could it learn directly from the metadata?
- Could it make access control decisions that are accurate and generalize better?



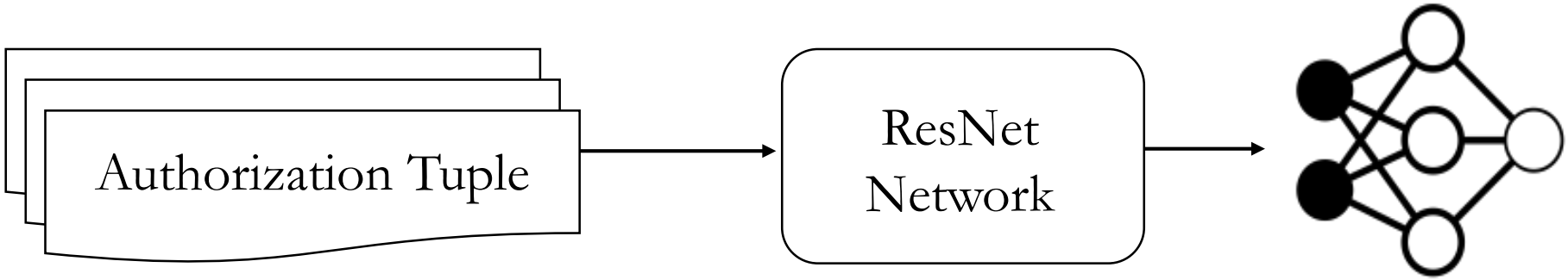
- Obviates the need for related processes
  - Attribute Engineering and Assignments
  - Policy Engineering

Deep Learning Based Access Control (DLBAC)

Policy Engineering in Classical Approaches vs. DLBAC



A Candidate DLBAC Model : $DLBAC_{\alpha}$



Authorization Tuple and Dataset in DLBAC$_\alpha$

User/ Resource metadata

User: Alice

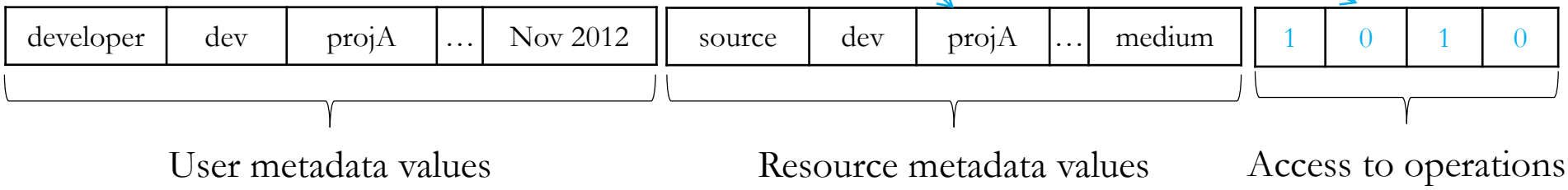
rank	team	project		join date
developer	dev	projA	…	Nov 2012

Resource: projectA

type	team	project		size
source	dev	projA	…	medium

Operations: op1, op2, op3, op4

Authorization Tuple: <Alice, projectA, {op1, op3}>

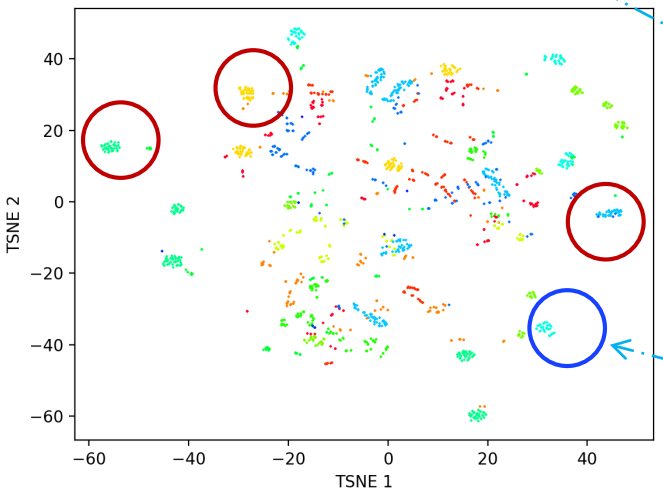


A **dataset** for DLBACα is the collection of such authorization tuples (samples)

Dataset Generation

Generate a synthetic dataset using Xu et al. [1]

<Alice, projectA, {op1, op3}>



t-SNE plot of a synthetic dataset

Each dot represents an authorization tuple

Each color indicates a unique combination of access operations

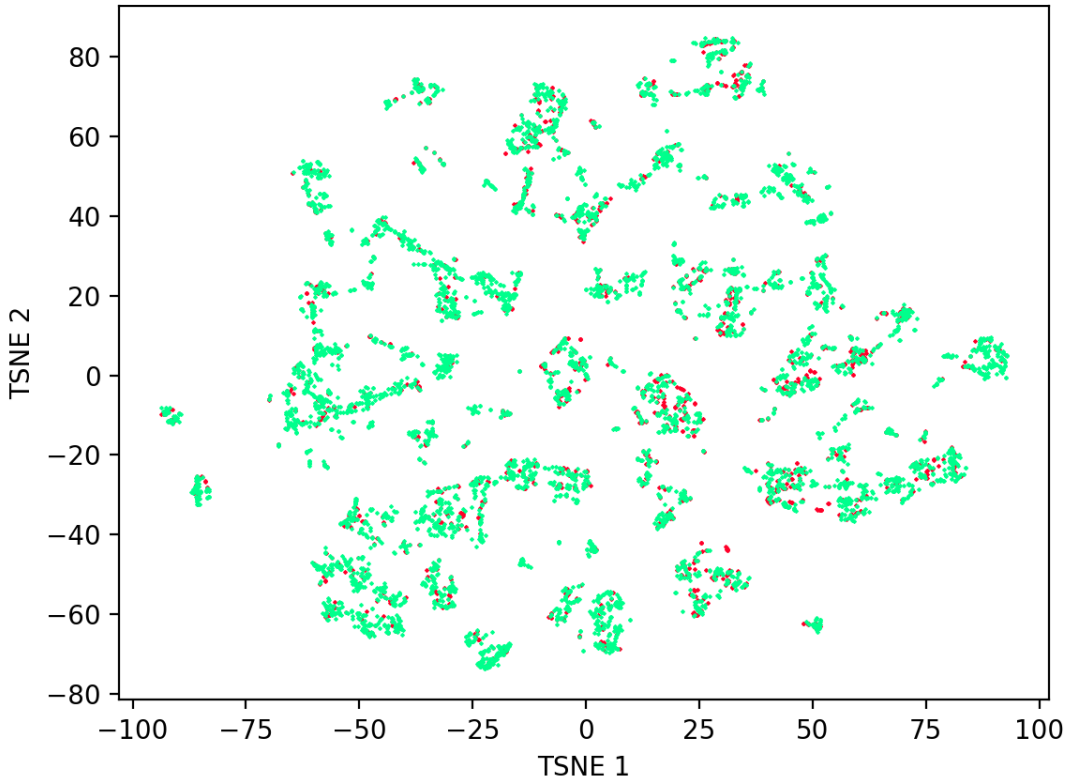
The position of a tuple is based on both user and resource metadata values

Two tuples are closed to each other, which indicates they have similar user-resource metadata values

multiple tuples of the same color indicate they have the same access

1. Xu et al. 2014. "Mining attribute-based access control policies." IEEE TDSC.

Real-world Systems Are Complex !

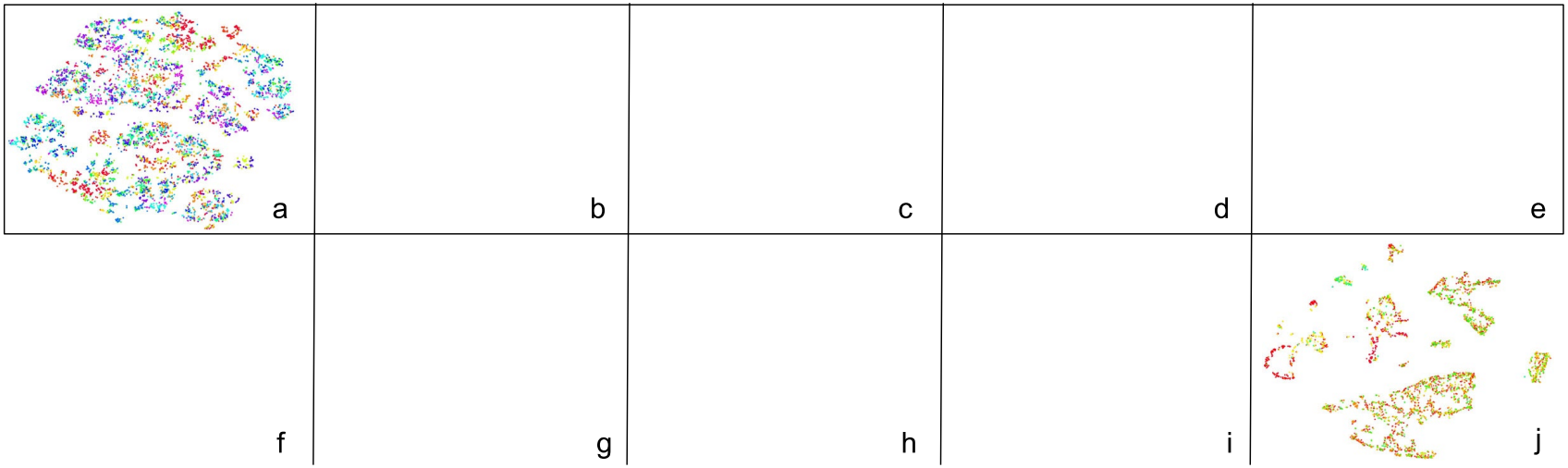


A dataset representing Amazon access control system

https://www.kaggle.com/c/amazon-employee-access-challenge/

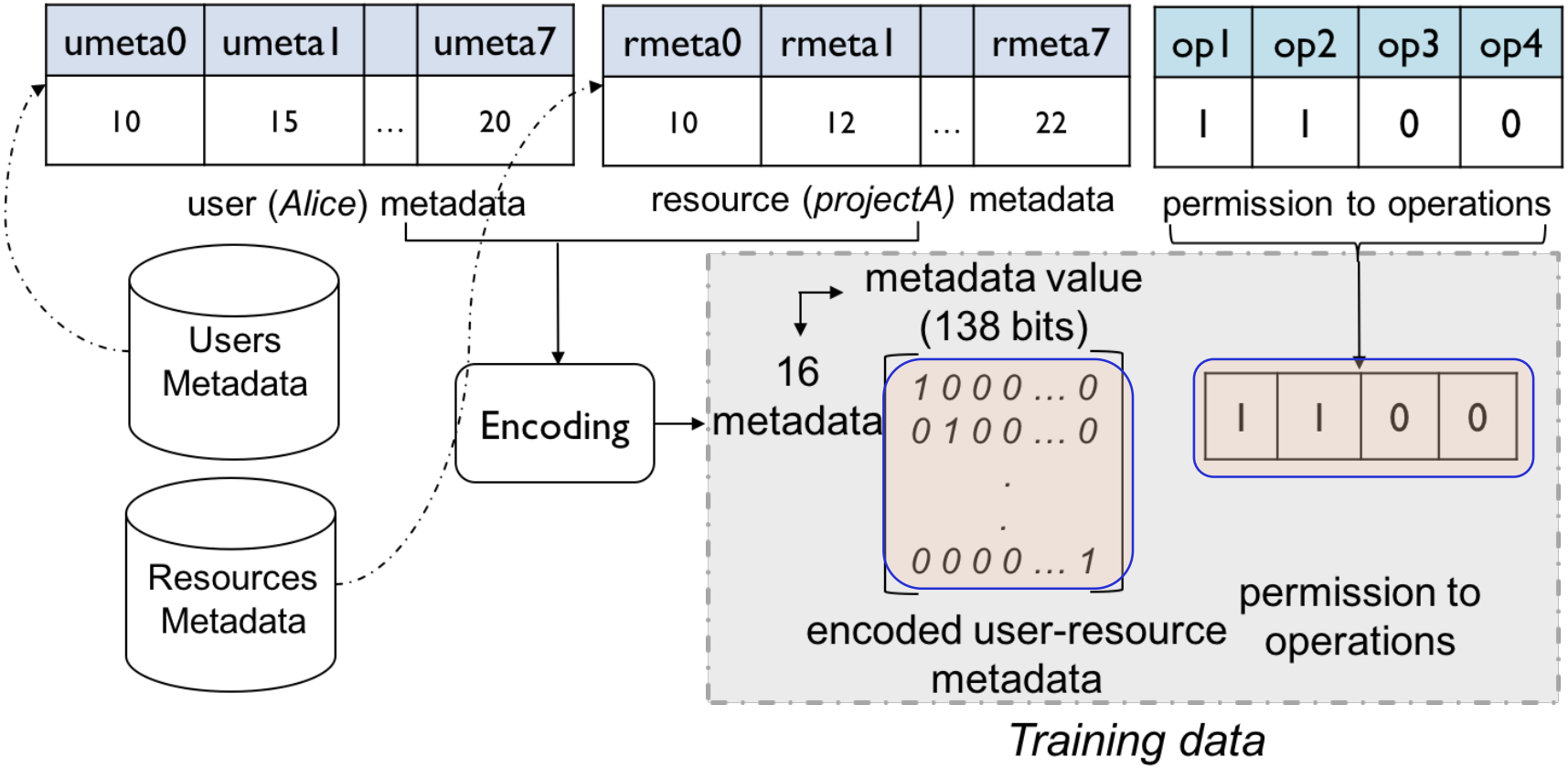
List of Datasets

#	Dataset	Type	Users	User Metadata	Resources	Resource Metadata	Authorization Tuples
1	*amazon-kaggle*	Real-world	9560	8	7517	0	32769
2	*amazon-uci*	Real-world	4224	11	7	0	4224
3	*u4k-r4k-auth11k*	Synthetic	4500	8	4500	8	10964
4	*u5k-r5k-auth12k*	Synthetic	5250	8	5250	8	12690
5	*u5k-r5k-auth19k*	Synthetic	5250	10	5250	10	19535
6	*u4k-r4k-auth21k*	Synthetic	4500	11	4500	11	20979
7	*u4k-r7k-auth20k*	Synthetic	4500	11	7194	11	20033
8	*u4k-r4k-auth22k*	Synthetic	4500	13	4500	13	22583
9	*u4k-r6k-auth28k*	Synthetic	4500	13	6738	13	28751
10	*u6k-r6k-auth32k*	Synthetic	6000	10	6000	10	32557

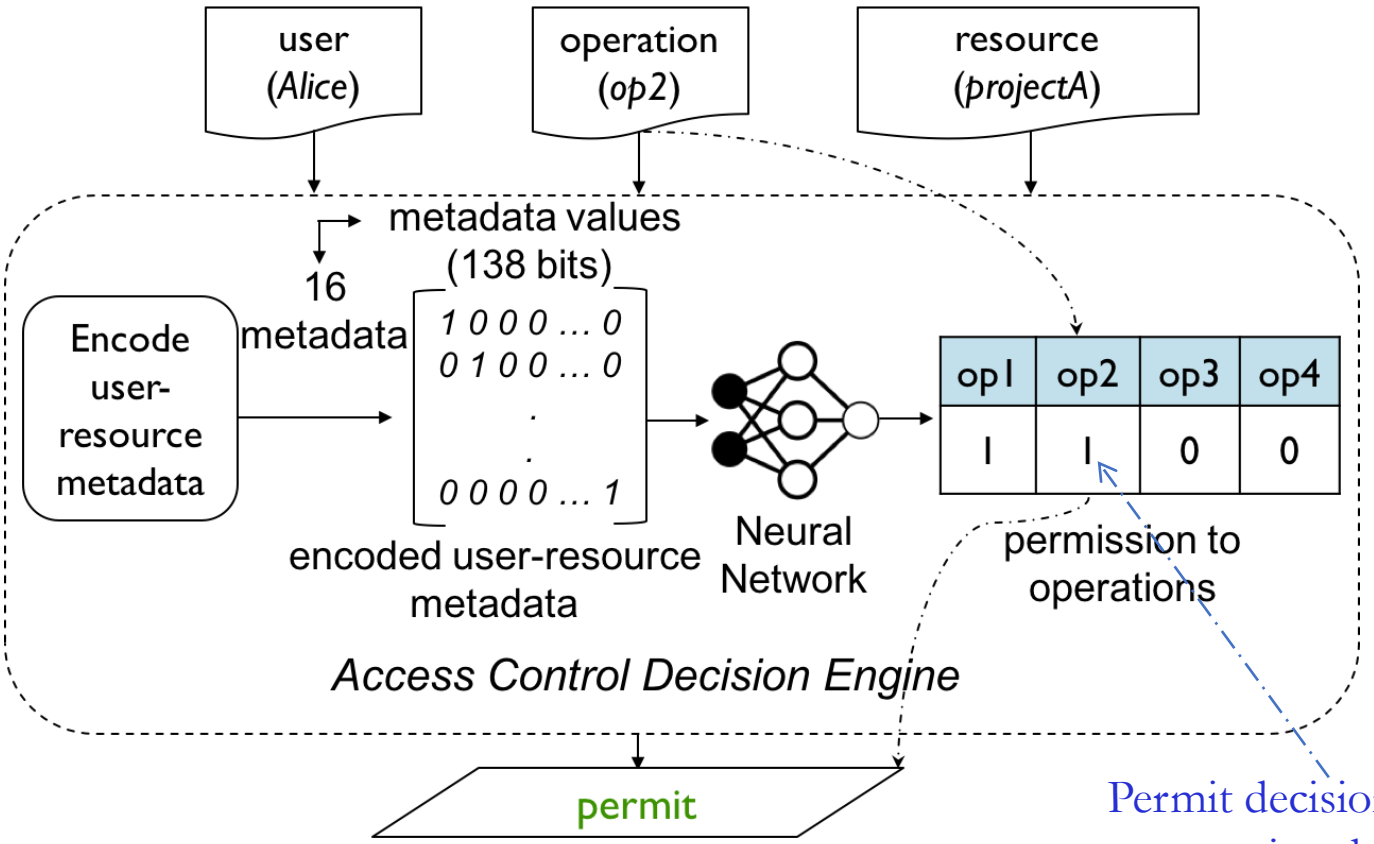


Preparing Training Data for DLBACα

We consider the data in our datasets are categorical



Decision Making Process in DLBACα



Permit decision is made comparing the output probability with a threshold

Evaluation Methodology

Multiple instances of DLBACα	• ResNet ($DLBAC_{\alpha\text{-}R}$) • DenseNet ($DLBAC_{\alpha\text{-}D}$) • Xception ($DLBAC_{\alpha\text{-}X}$)
Classical ML Algorithms	• SVM • Random Forest (RF) • Multilayer Perceptron (MLP)
State-of-the-art policy mining and ML-based techniques	• XuStoller [1] • Rhapsody [2] • EPDE-ML [3]

[1] Xu et al. 2014. "Mining attribute-based access control policies." IEEE TDSC
[2] Cotrini et al. 2018. Mining ABAC rules from sparse logs. In IEEE Euro S&P.
[3] Liu et al. 2021. Efficient Access Control Permission Decision Engine Based on Machine Learning. Security & Communication Networks.

Evaluation Metrics

80% samples for the training, and 20% testing

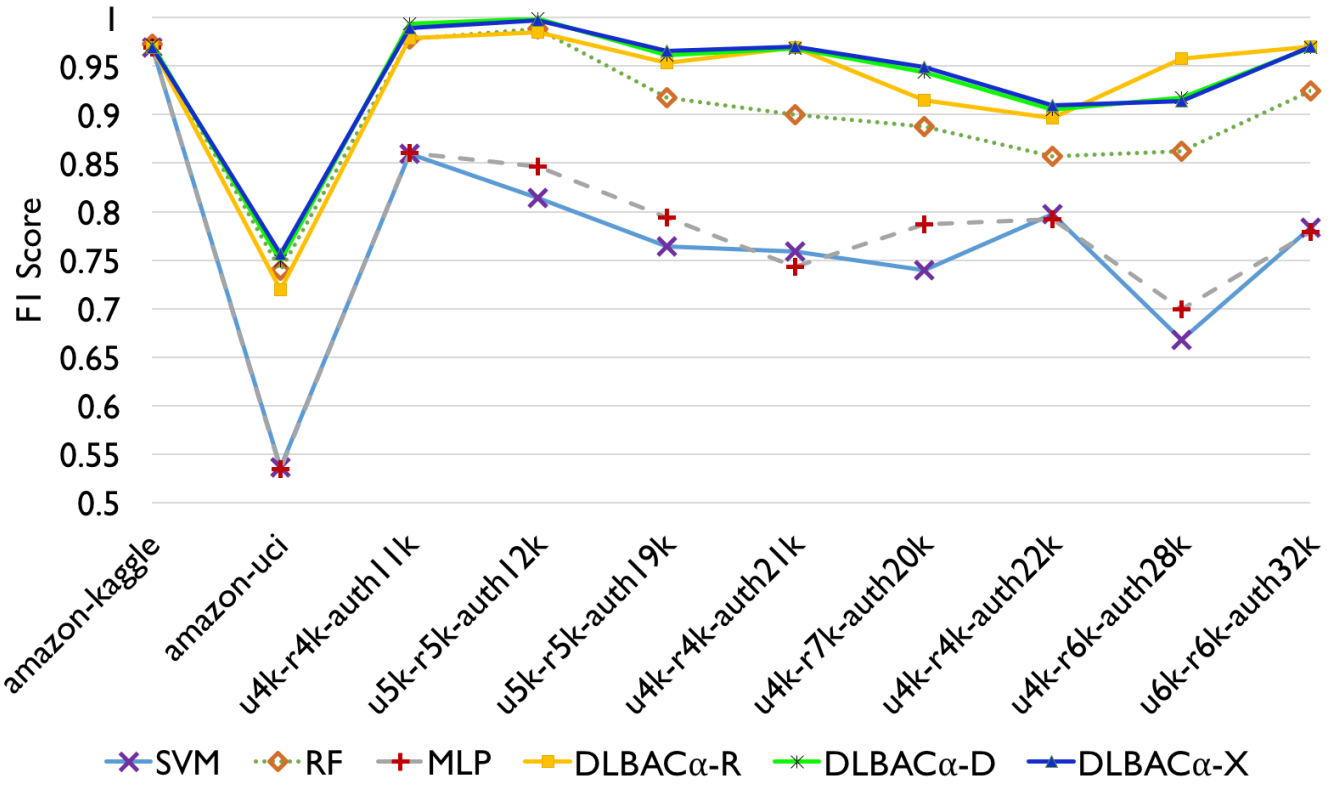
A higher F1 score: better generalization

A higher TPR: accurate and efficient in granting access

A lower FPR: efficient in denying access

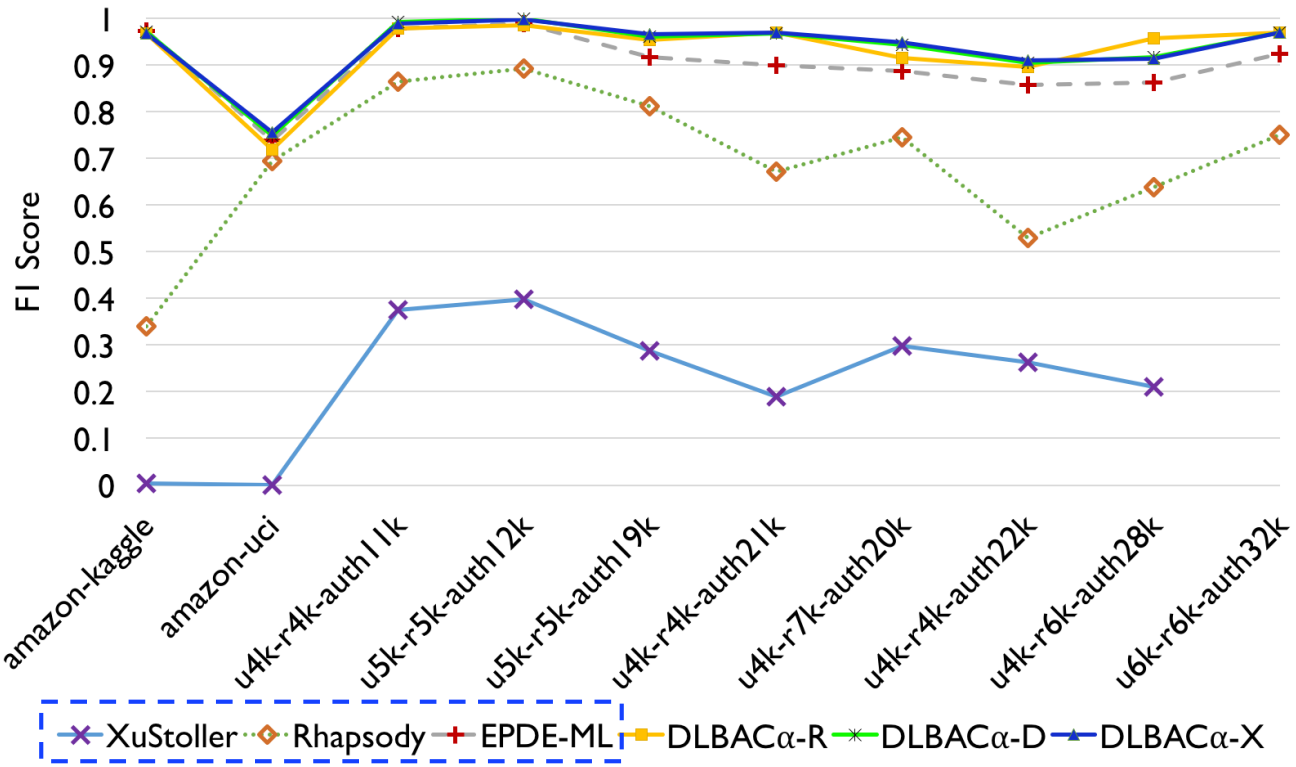
Under-Provision vs. Over-Provision

Comparison with ML Algorithms



DLBACα instances are more effective and accurate than classical ML approaches for making accurate access decisions

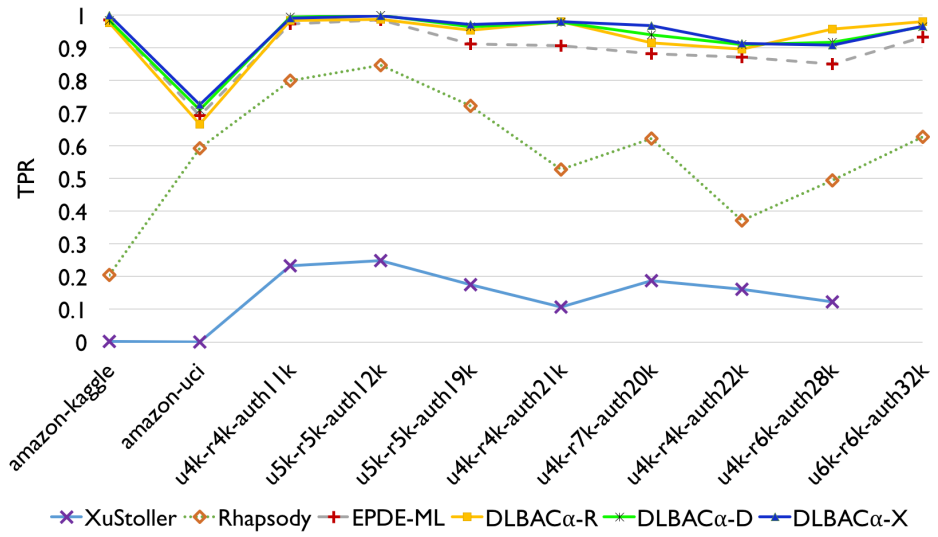
Comparison with Policy Mining Algorithms



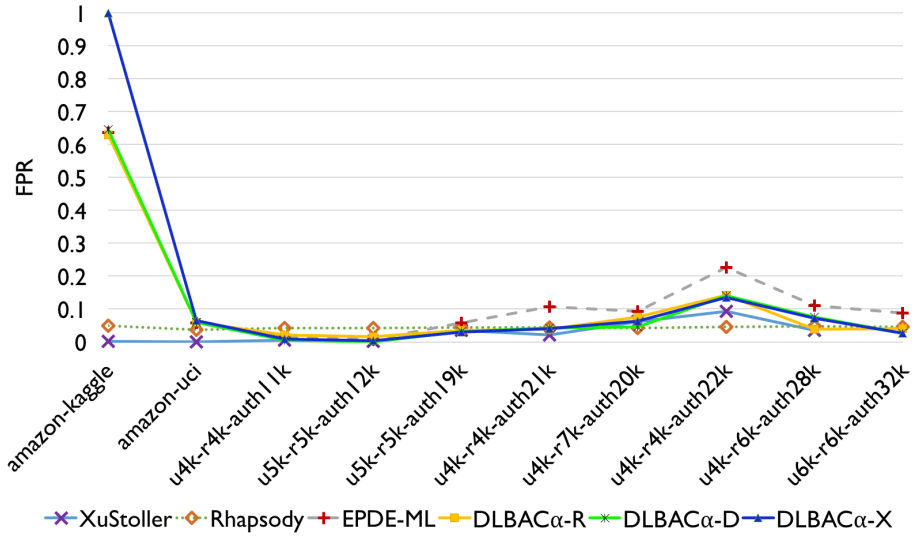
A DLBACα can make more accurate access control decisions and generalize better

Comparison with Policy Mining Algorithms (cont'd)

handling desired accesses

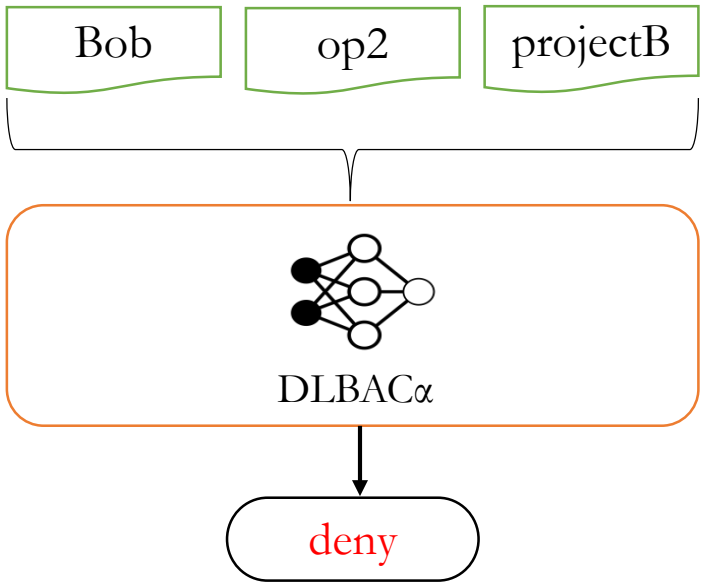


handling unwanted accesses



A DLBACα can balance both permitting desired accesses and denying unwanted accesses

Understanding DLBAC Decisions



A sample access request

Why does Bob's 'op2' access been denied for projectB resource?

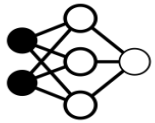
Which metadata are important/ influential for this decision?

- Propose two approaches
  - Integrated Gradients
  - Knowledge Transferring

Integrated Gradients

Bob's metadata

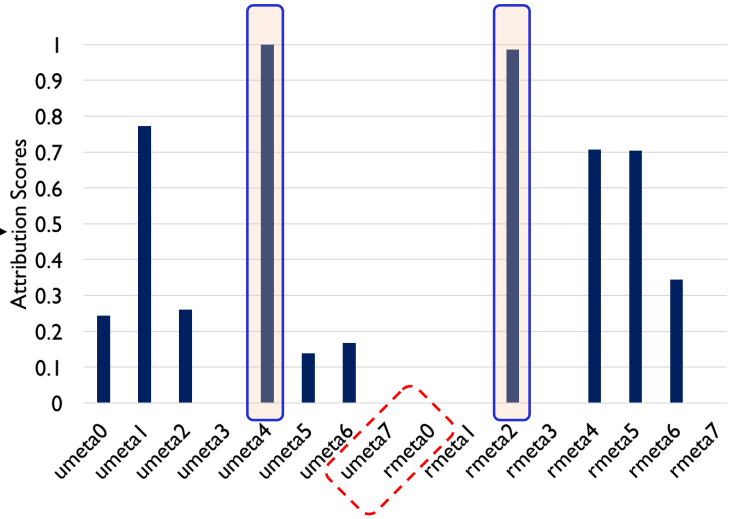
projectB's metadata



DLBACα

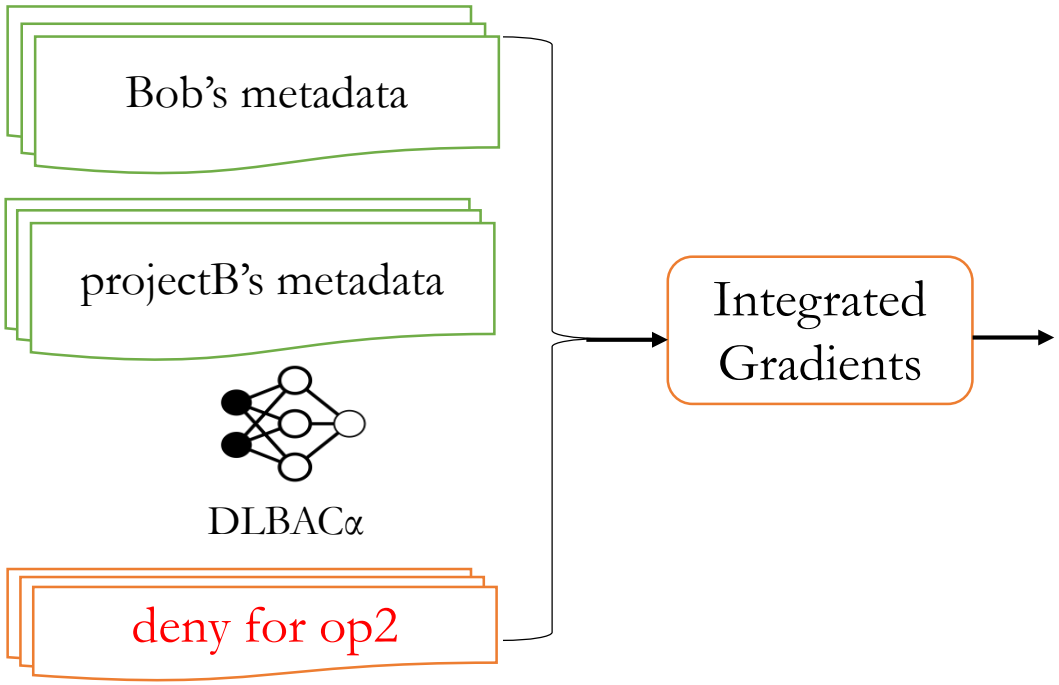
deny for op2

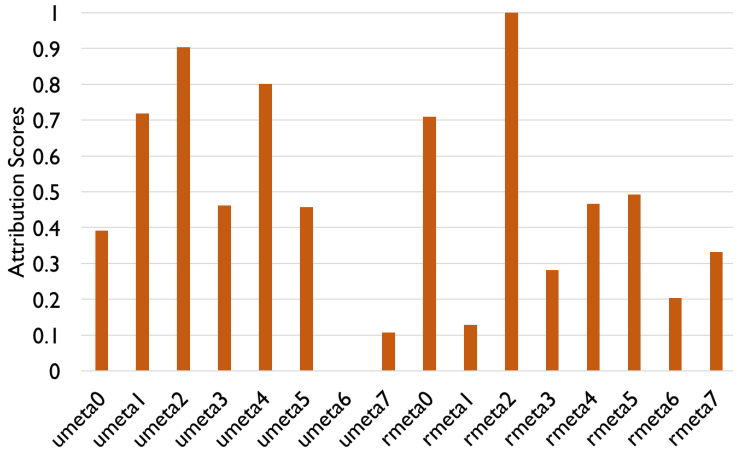
Integrated Gradients



Local Interpretation

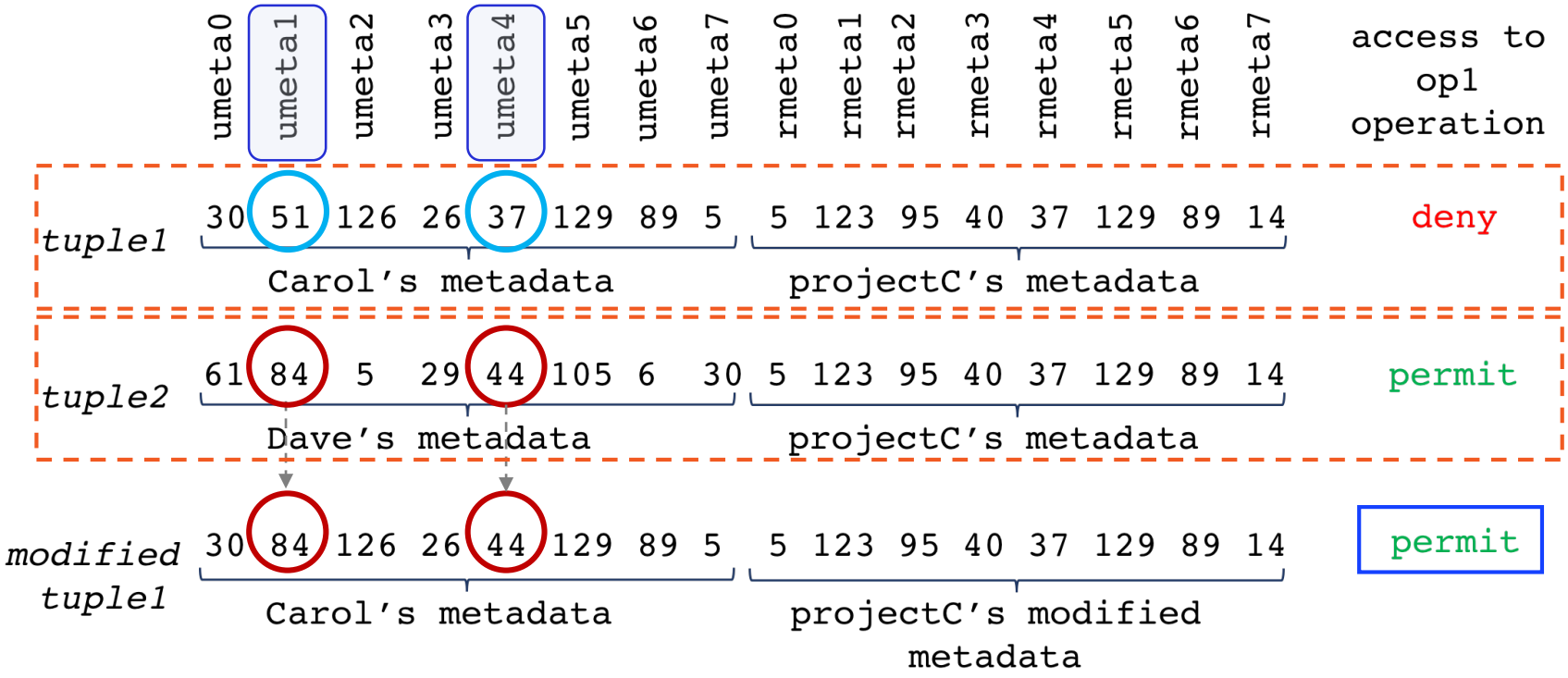
Integrated Gradients





Global Interpretation

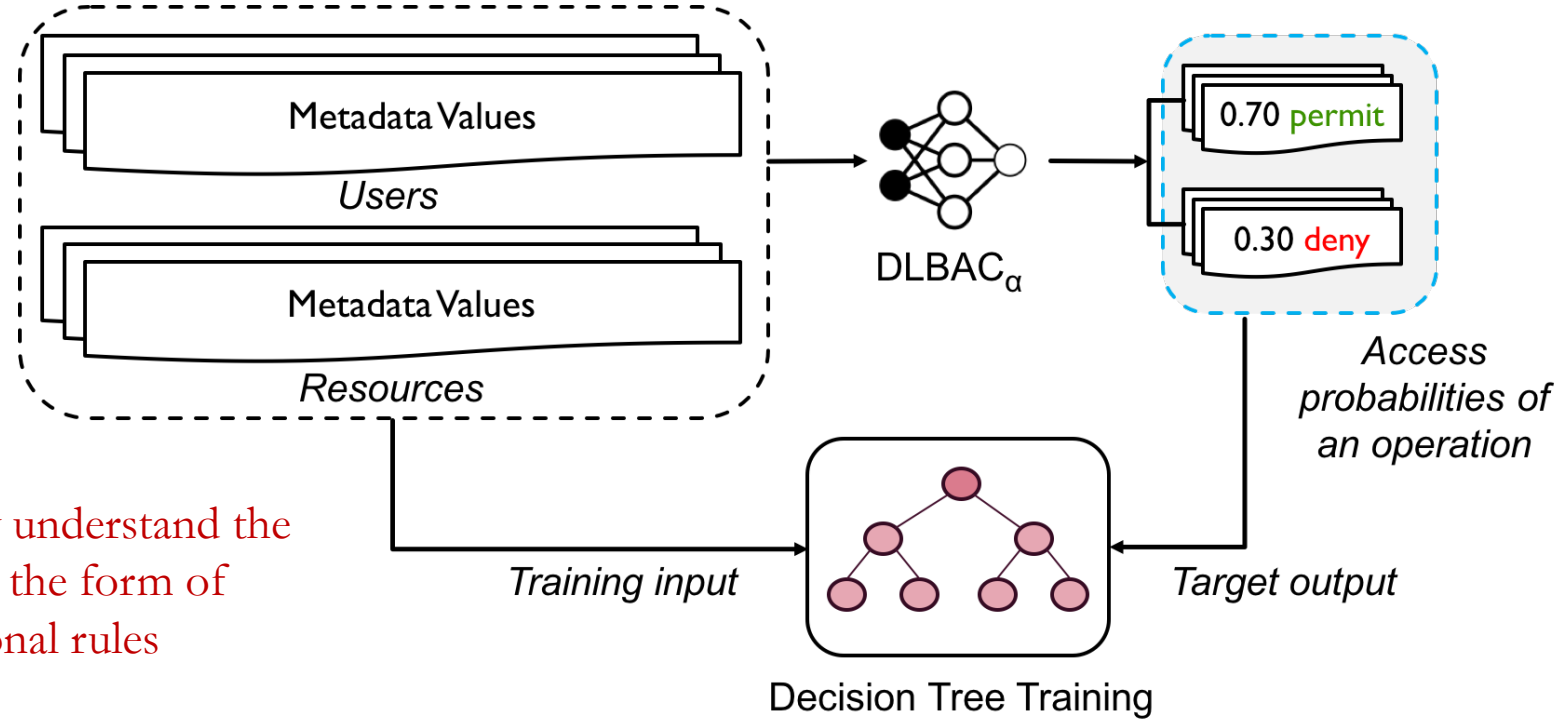
Application of Integrated Gradient-based Understanding



- Strengthen the effect of "influential metadata"
- Can be utilized in future access modification

doesn't establish the relationship among metadata

Knowledge Transferring



approximately understand the decision in the form of traditional rules

- Rule: local interpretation
- DT: global interpretation

Future Research Directions

- DLBAC Administration
- DLBAC in Tandem
- Adversarial Attacks from Access Control Perspectives
- Bias and Fairness

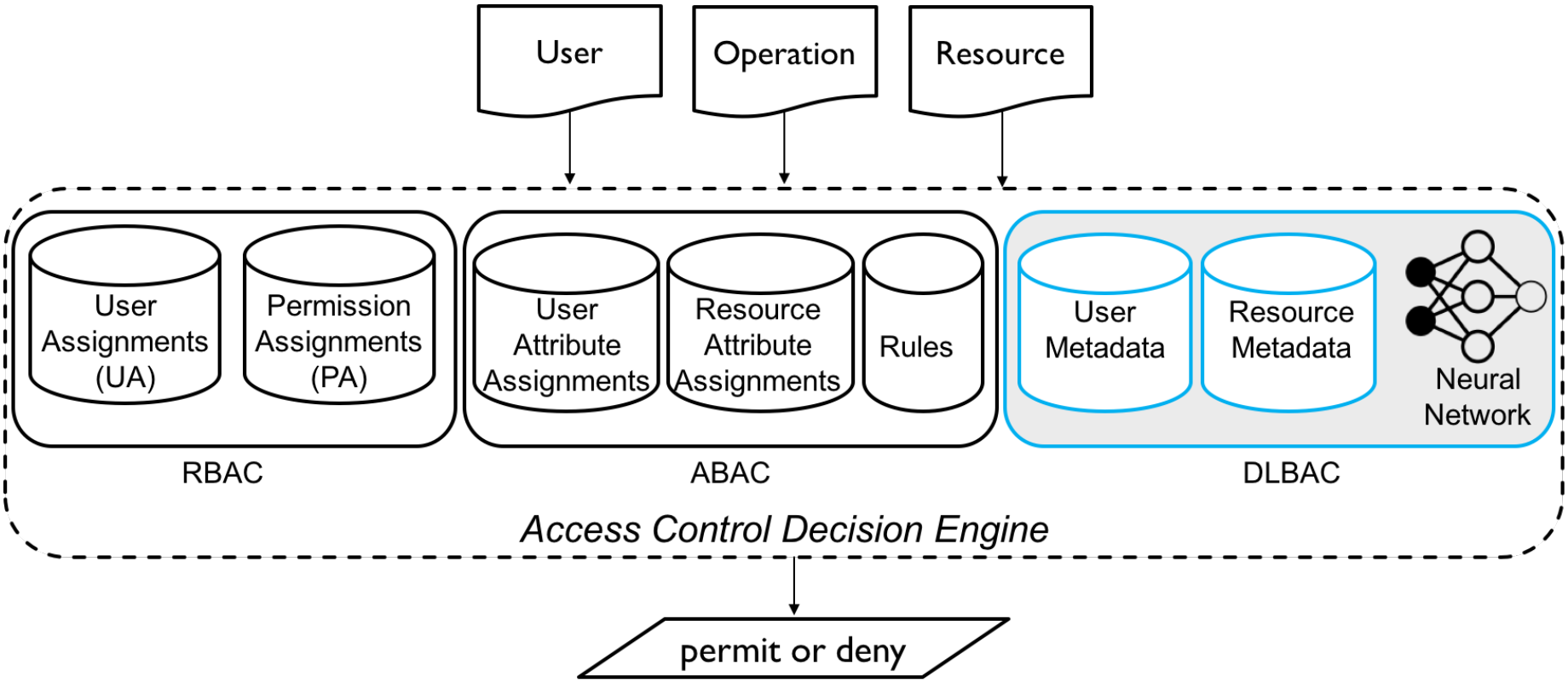
DLBAC Source code and datasets URL:

https://github.com/dlbac/DlbacAlpha

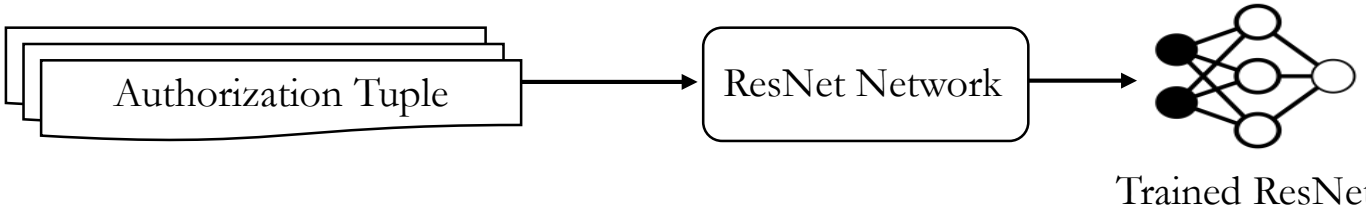
Thank You

Backup

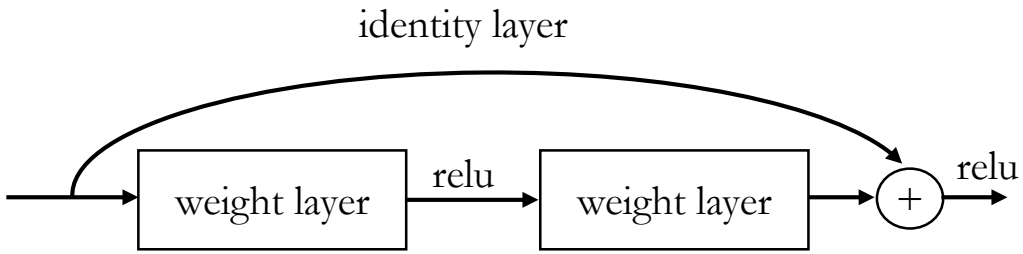
Decision Making in Classical Approaches vs. DLBAC



A Candidate DLBAC Model : $DLBAC_{\alpha}$



- ResNet dominates in different deep learning applications
- Reduces parameters and faster training
- Reducing the effect of vanishing gradient problem through identify layers



How about Authorization Tuples?

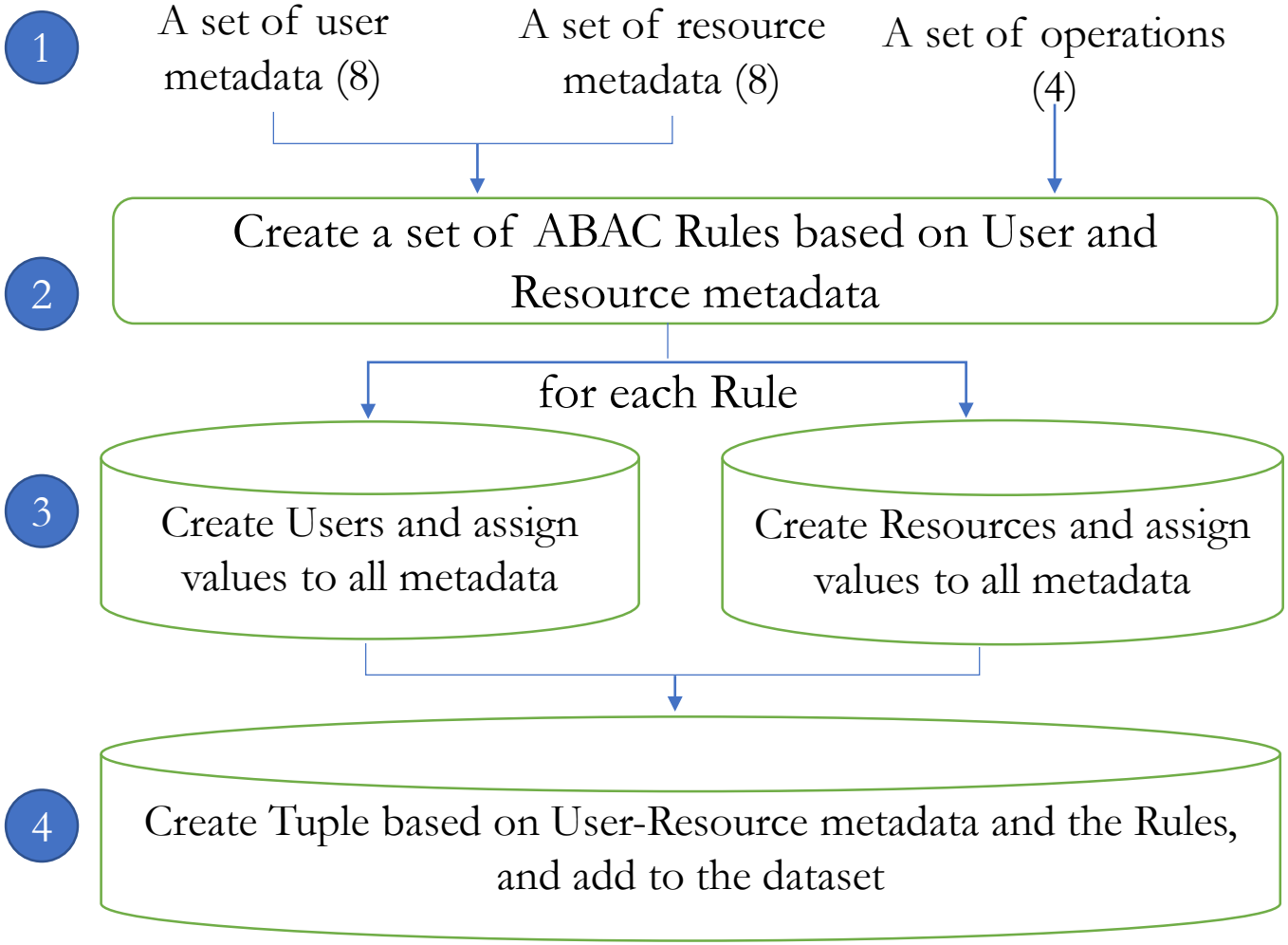
Network Architectures

For dataset 1-4: ResNet8
For dataset 5-10: ResNet50

Layers	Output Size	DenseNet-121
Convolution	112 × 112	
Pooling	56 × 56	
Dense Block (1)	56 × 56	[1 × 1 conv, 3 × 3 conv] × 6
Transition Layer (1)	56 × 56 28 × 28	
Dense Block (2)	28 × 28	[1 × 1 conv, 3 × 3 conv] × 12
Transition Layer (2)	28 × 28 14 × 14	
Dense Block (3)	14 × 14	[1 × 1 conv, 3 × 3 conv] × 24
Transition Layer (3)	14 × 14 7 × 7	
Dense Block (4)	7 × 7	[1 × 1 conv, 3 × 3 conv] × 16
Classification Layer	1 × 1	

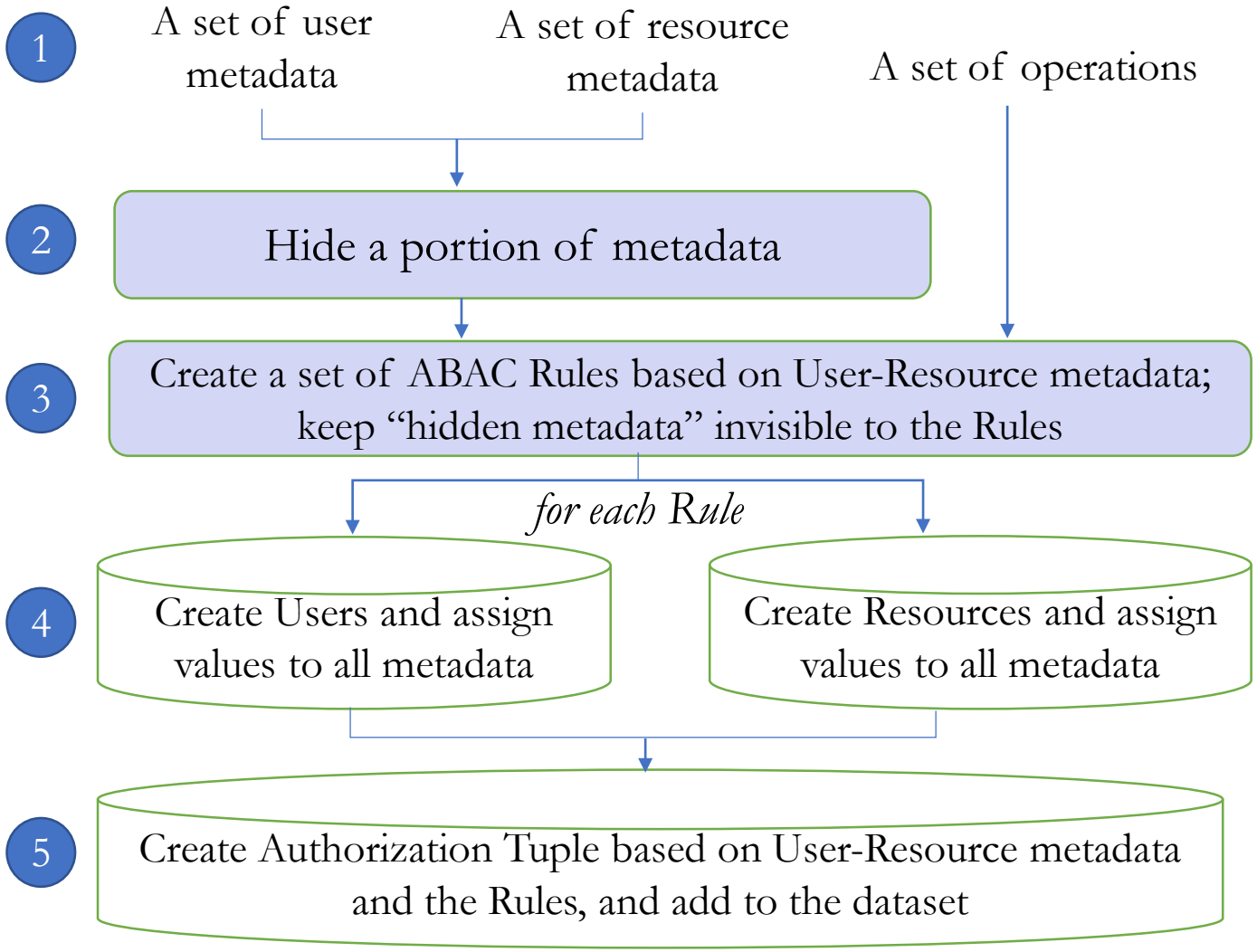
ResNet, DenseNet

Dataset Generation

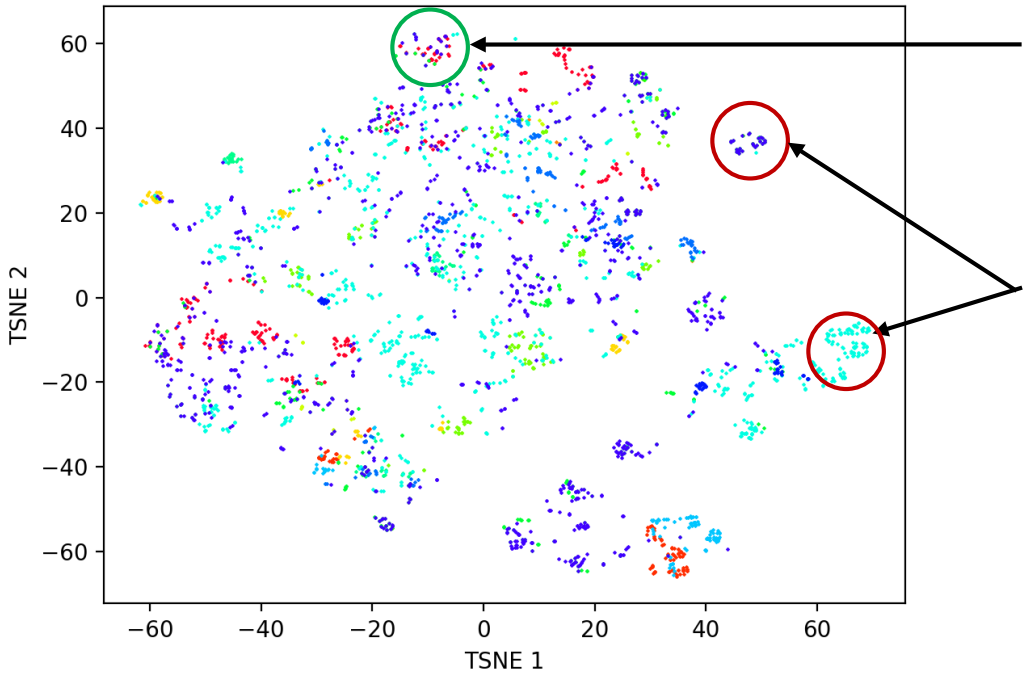


Proposed by Xu et al. , Mining attribute-based access control policies, TDSC'2014

Dataset Generation



Dataset Generation



Mixes authorization tuples of other colors

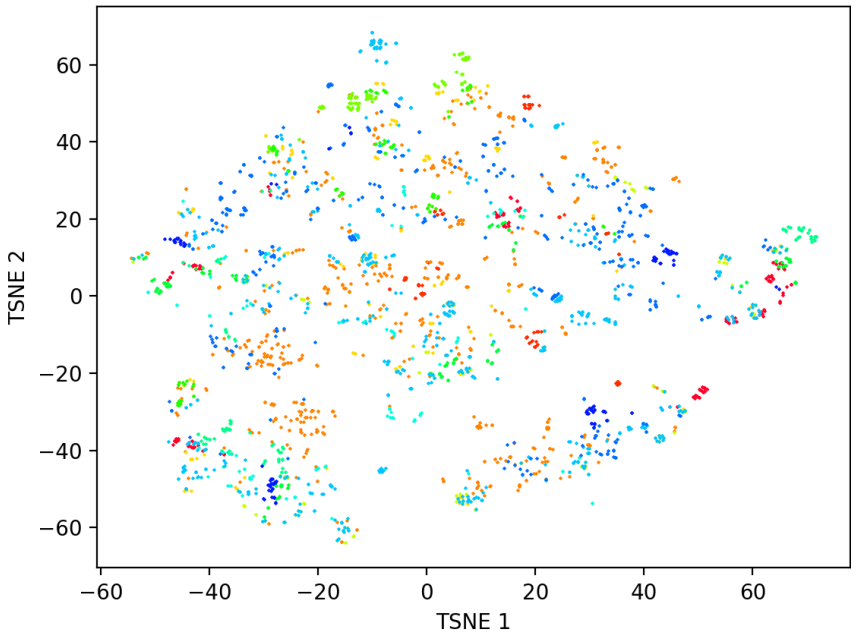
There are tuples of same color those are easily distinguishable

Metadata values are assigned based on a very sparse distribution

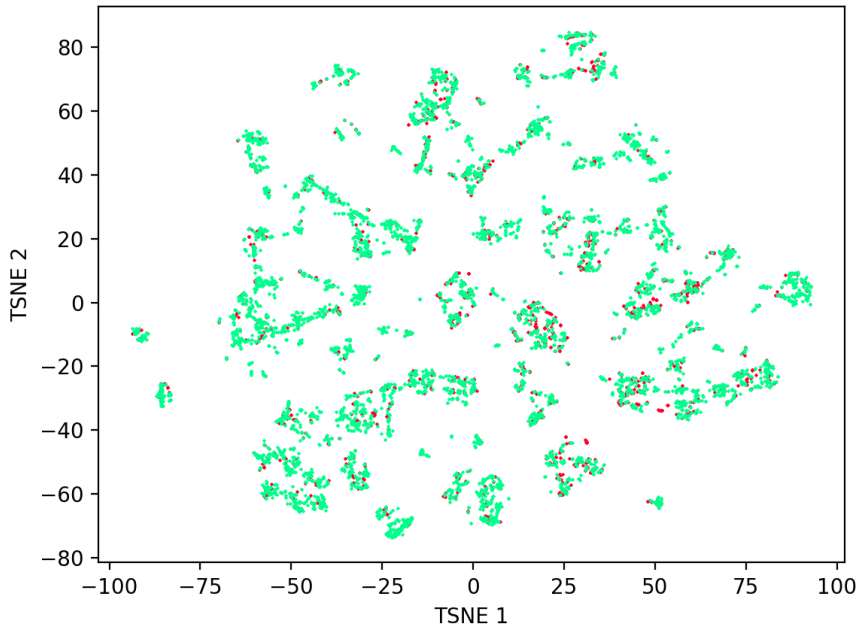
We determine a fixed set of values for each metadata

A dataset with 1000 users and 639 resources, **3 hidden** user/resource metadata.

Dataset Generation

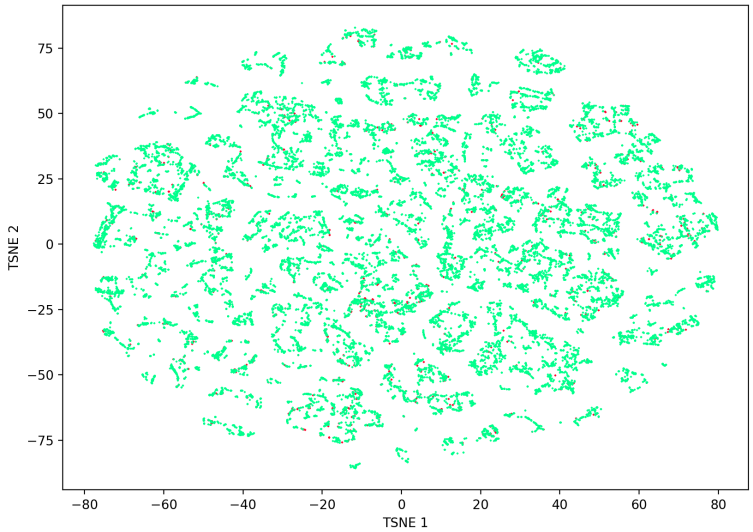


A dataset with 800 users and 665 resources, 3 hidden metadata, **fixed set of metadata values**.

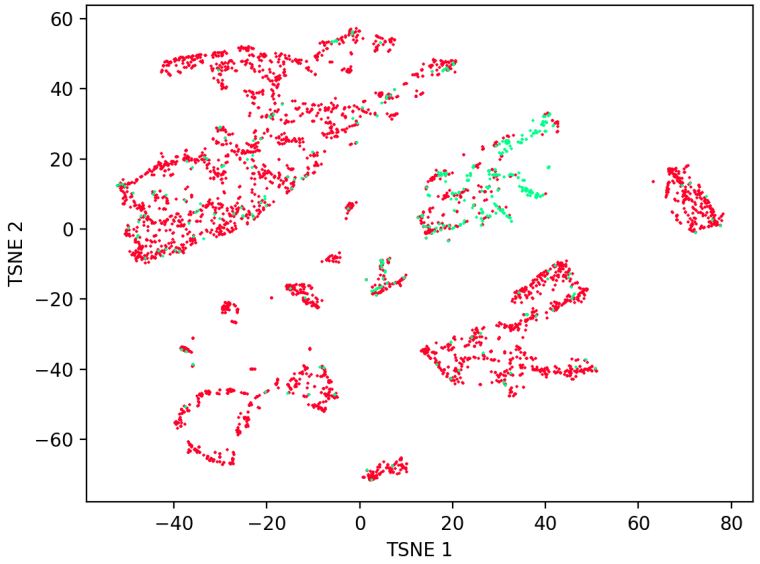


A real-world dataset from Amazon

Characteristics of AmazonKaggle and AmazonUCI Datasets



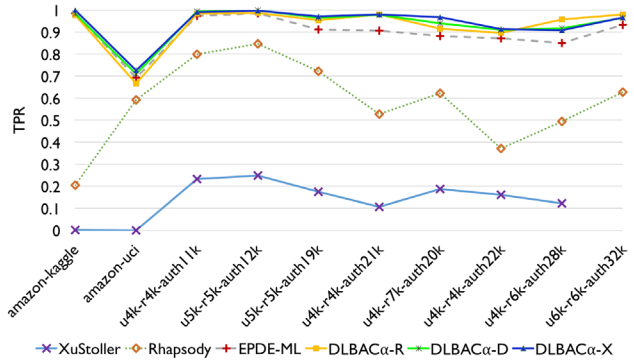
Amazon Kaggle Dataset

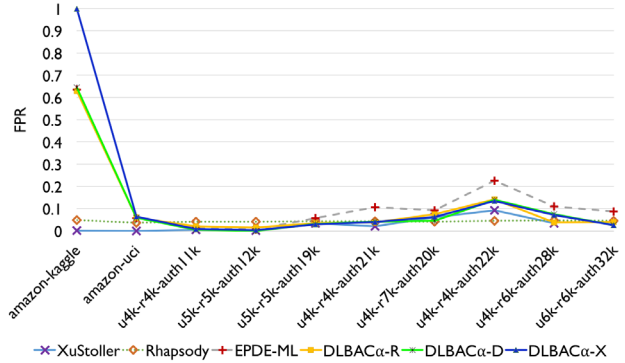


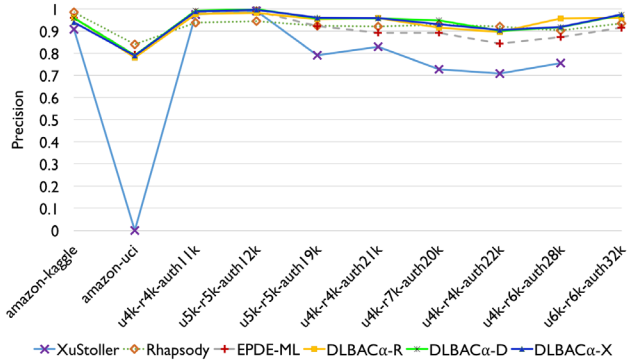
Amazon UCI Dataset

Highly imbalanced !

Performance Comparison with Policy Mining Algorithms

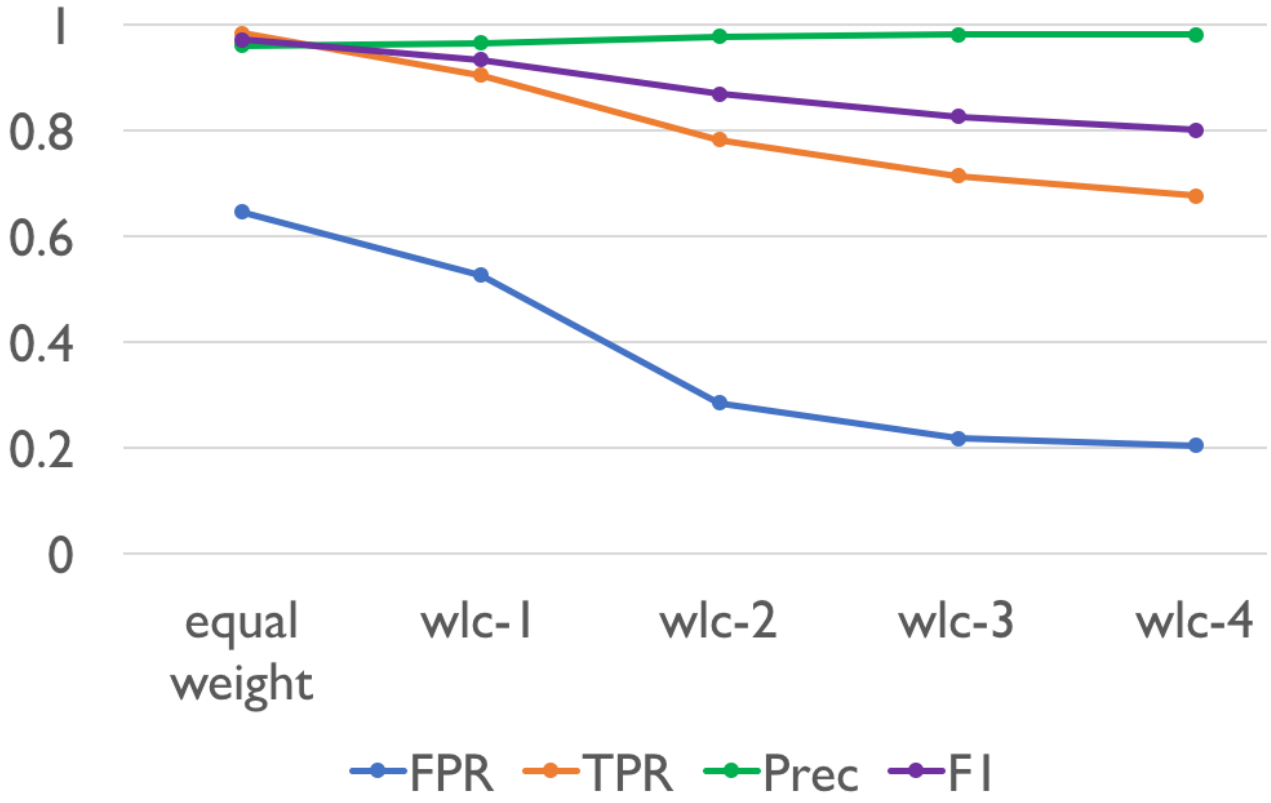




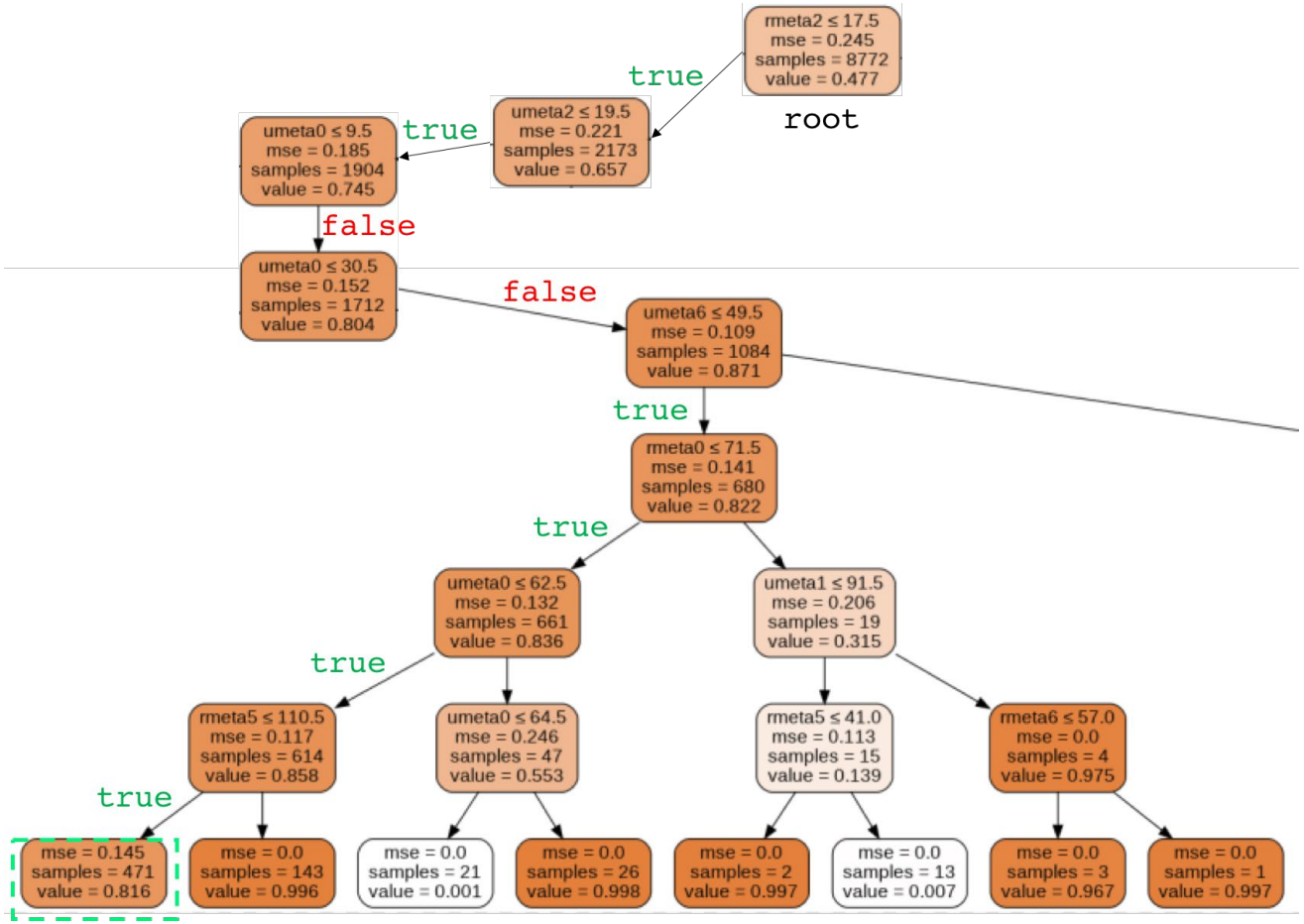


A deep learning based approach can properly balance both over-provision and under-provision

FPR Performance Improvement in DLBACα



Decision Tree Generated from KT in DLBACα



Summary

- Propose DLBAC framework as an automated access control system
- Experiment and evaluate the performance of DLBAC prototype using both synthetic and real-world access control data
- DLBAC Performance:
  - Make more accurate access control decisions and generalize better
  - Properly balance both permitting desired accesses and denying unwanted accesses
- Propose two methods for understanding DLBAC decisions in human terms