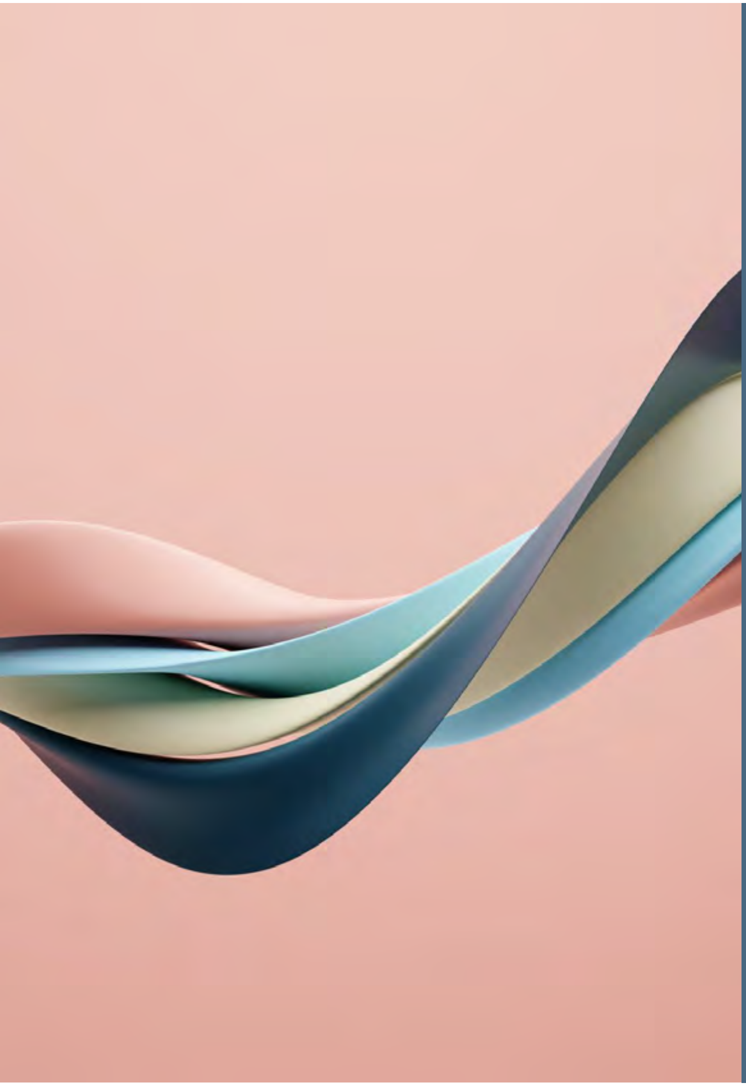


TOWARD MACHINE LEARNING BASED ACCESS CONTROL

Ram Krishnan

Microsoft President's Endowed Professor

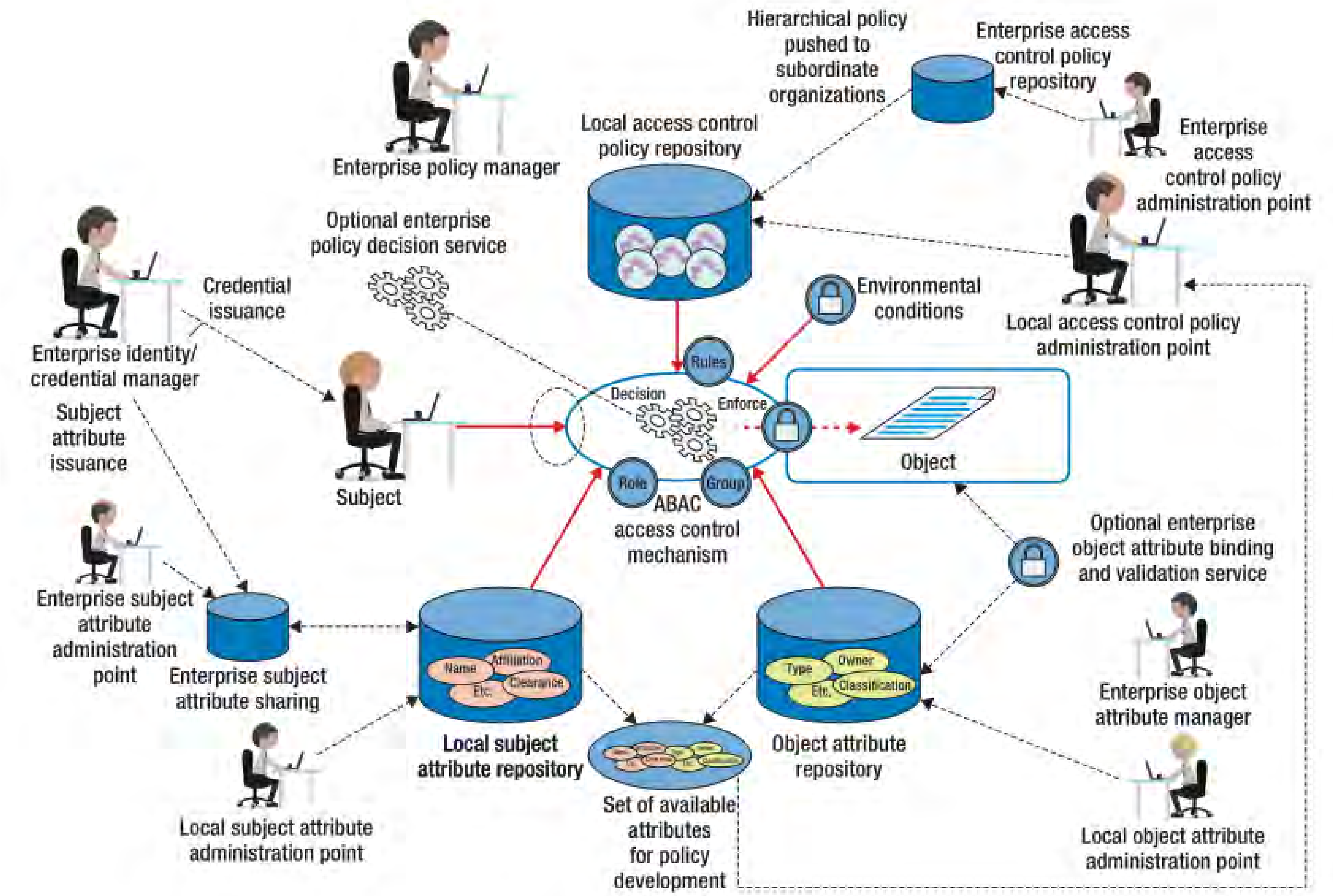
Department of Electrical and Computer Engineering

UTSA

Introduction

- Access Control
  - The decision to permit or deny a user access to a resource
  - User: a human user, a process, an application, etc.
  - Resource: network, data, application, service, etc.
- There are many mainstream classical 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

NIST ABAC


Hierarchical policy pushed to subordinate organizations
Enterprise access control policy repository
Enterprise policy manager
Local access control policy repository
Enterprise access control policy administration point
Optional enterprise policy decision service
Credential issuance
Environmental conditions
Local access control policy administration point
Enterprise identity/ credential manager
Rules
Decision
Enforce
Object
Subject attribute issuance
Subject
Role
Group
ABAC access control mechanism
Optional enterprise object attribute binding and validation service
Enterprise subject attribute administration point
Enterprise subject attribute sharing
Name
Affiliation
Etc.
Clearance
Type
Owner
Etc.
Classification
Local subject attribute repository
Object attribute repository
Enterprise object attribute manager
Local subject attribute administration point
Set of available attributes for policy development
Local object attribute administration point

Issues in Classical Approaches

Attribute Engineering

- An expert designs attributes based on the metadata
- E.g., ‘status’ attribute is engineered from ‘spending’ and ‘credit’ history

Policy Engineering (Policy Mining)

- To design policy through a manual or automated process
- E.g., <status = ‘platinum’, type=‘secured’> <access = ‘read, write’>

Generalization

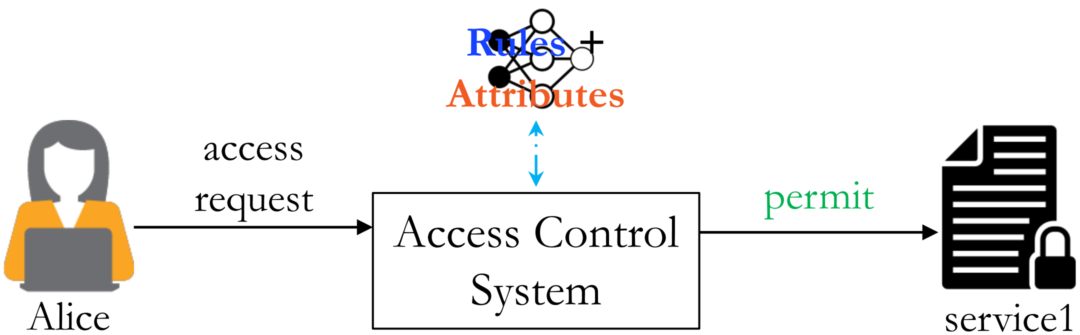
- Focus on capturing given access control state
- E.g., Knowing Alice’s access, is it possible to determine Bob’s access?

Attribute and Policy Update (administration)

- Revoke existing access or introduce a new access to existing users
- Depends on human, error-prone

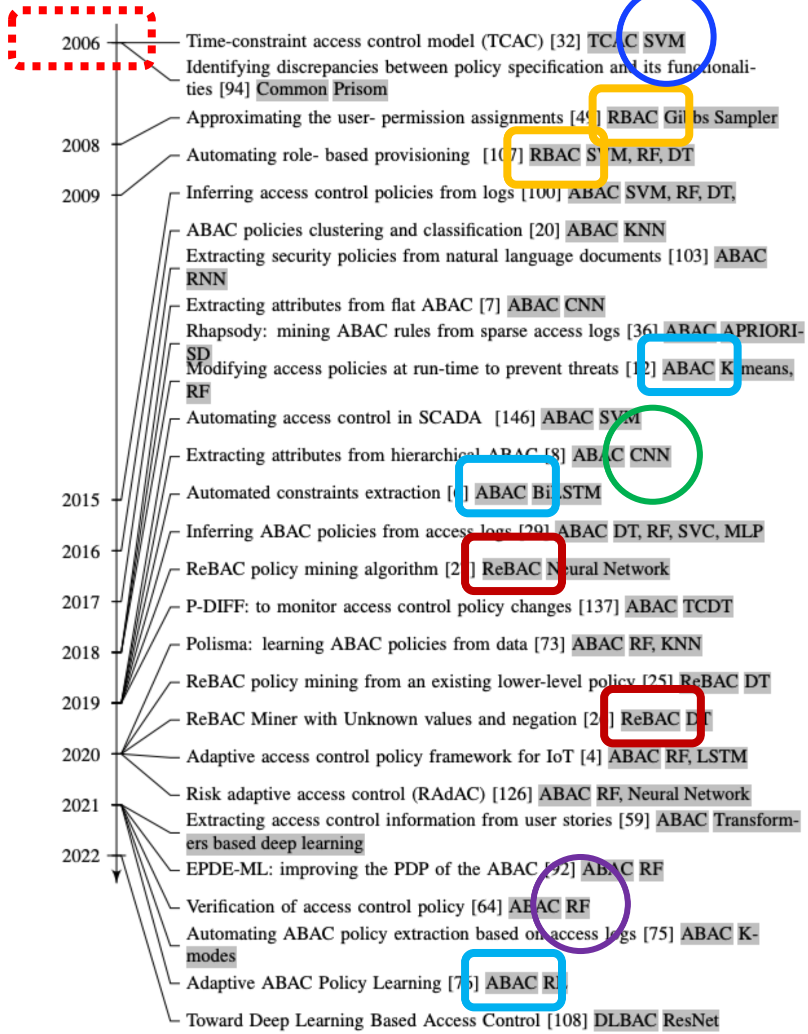
Machine Learning in Access Control

- 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?

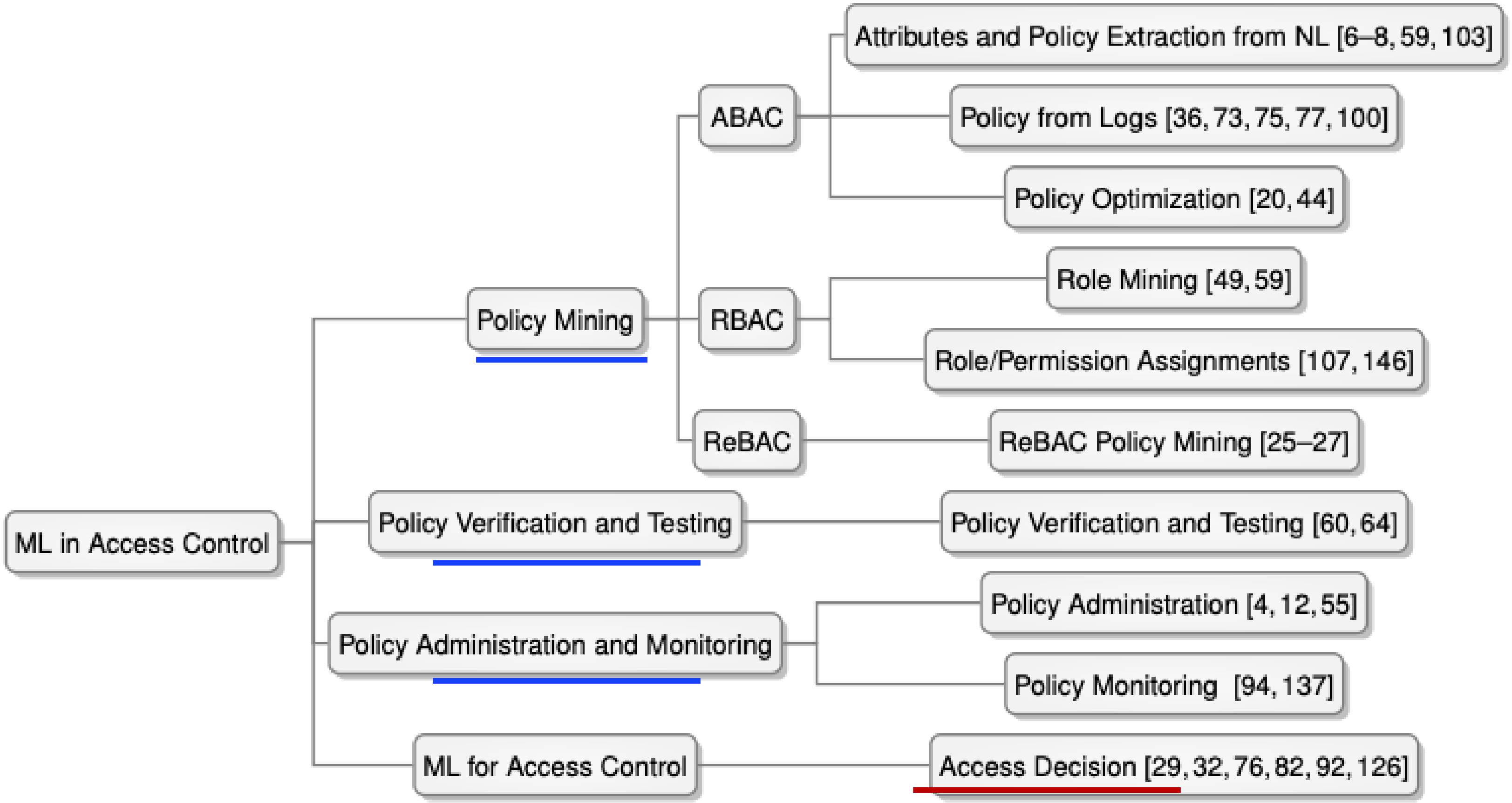


- Obviates the need for related procedures
  - Attribute Engineering and Assignments
  - Policy Engineering
- Ease of policy updates (Administration)

Timeline of
ML in Access Control



Taxonomy of ML in Access Control



Roadmap

Machine Learning Based Access Control (MLBAC)

State of the Art: ML in Access Control

Operational Model of MLBAC

Administration of MLBAC

DLBAC (prototype, interpretation)

Adversarial Attacks in DLBAC

Implementation and Evaluation of DLBAC

Section-2

Machine Learning Based Access Control (MLBAC)

State of the Art: ML in Access Control

Operational Model of MLBAC

DLBAC (prototype, interpretation)

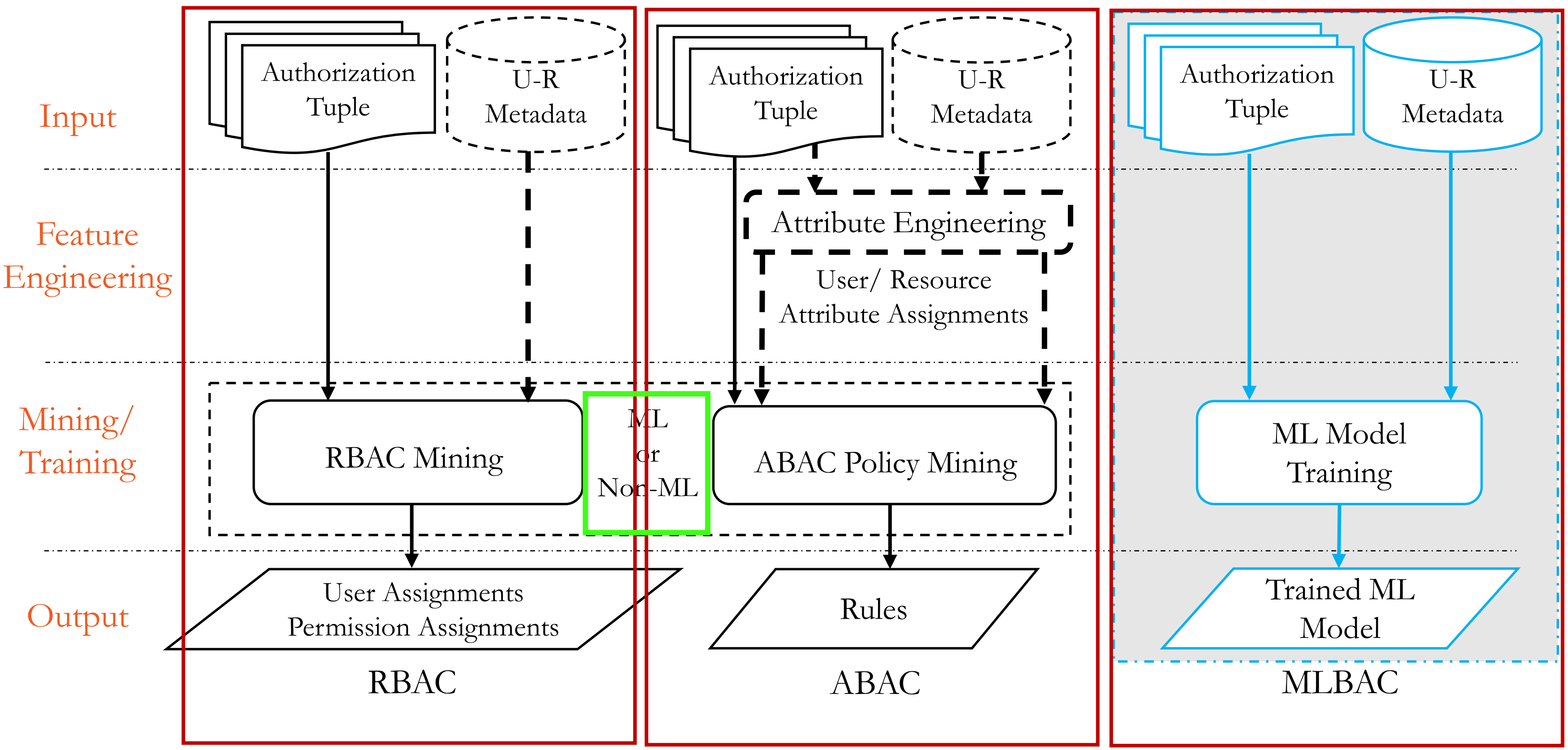
Administration of MLBAC

Adversarial Attacks in DLBAC

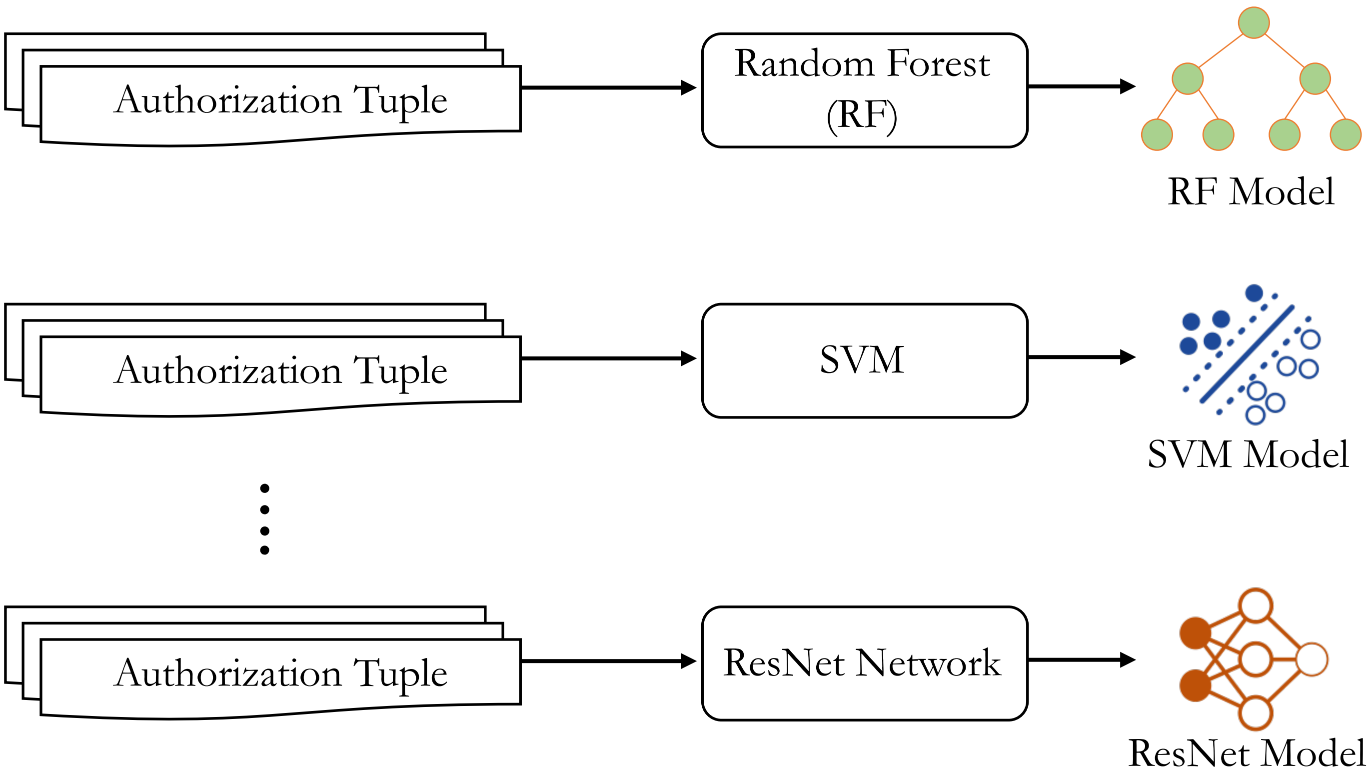
Implementation and Evaluation of DLBAC

Operational Model of Machine Learning Based Access Control

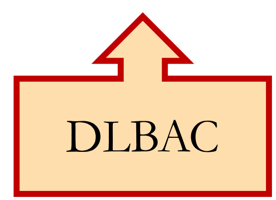
Authorization Tuple <Alice, projectA, {read, write}>



Candidate MLBAC Models



We create a DLBAC instance: **DLBAC$_\alpha$**



DLBAC$_\alpha$ Dataset

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, $\boxed{\{op1, op3\}}$>

developer	dev	projA	…	Nov 2012	source	dev	projA	…	medium	1	0	1	0

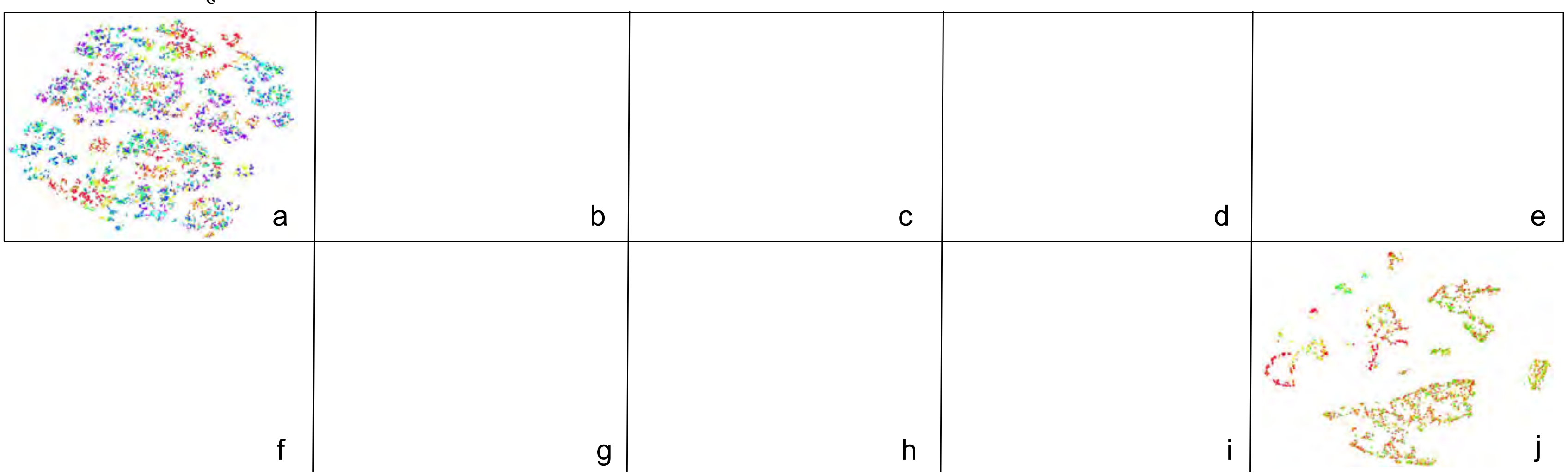
User metadata values | Resource metadata values | Access to operations

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

List of Datasets

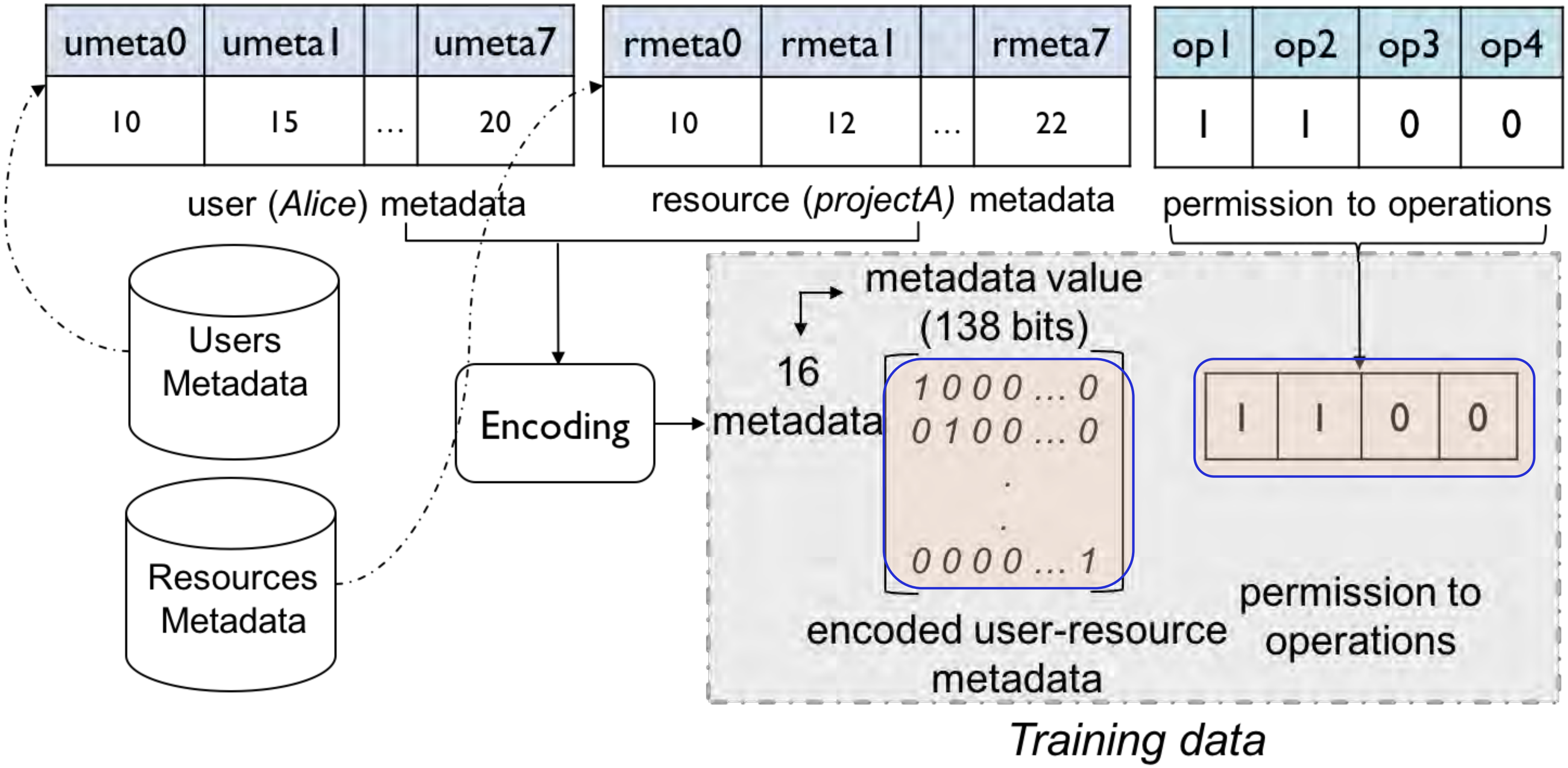
#	Dataset	Type	Users	User Metadata	Resources	Resource Metadata	Authorization Tuples
1	*amazon-kaggle*	Real-world	9560	8	7517	0	32769
2	*amazon-uci*	Real-world	4224	11	7	0	4224
3	*u4k-r4k-auth11k*	Synthetic	4500	8	4500	8	10964
4	*u5k-r5k-auth12k*	Synthetic	5250	8	5250	8	12690
5	*u5k-r5k-auth19k*	Synthetic	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

t-SNE visualizations

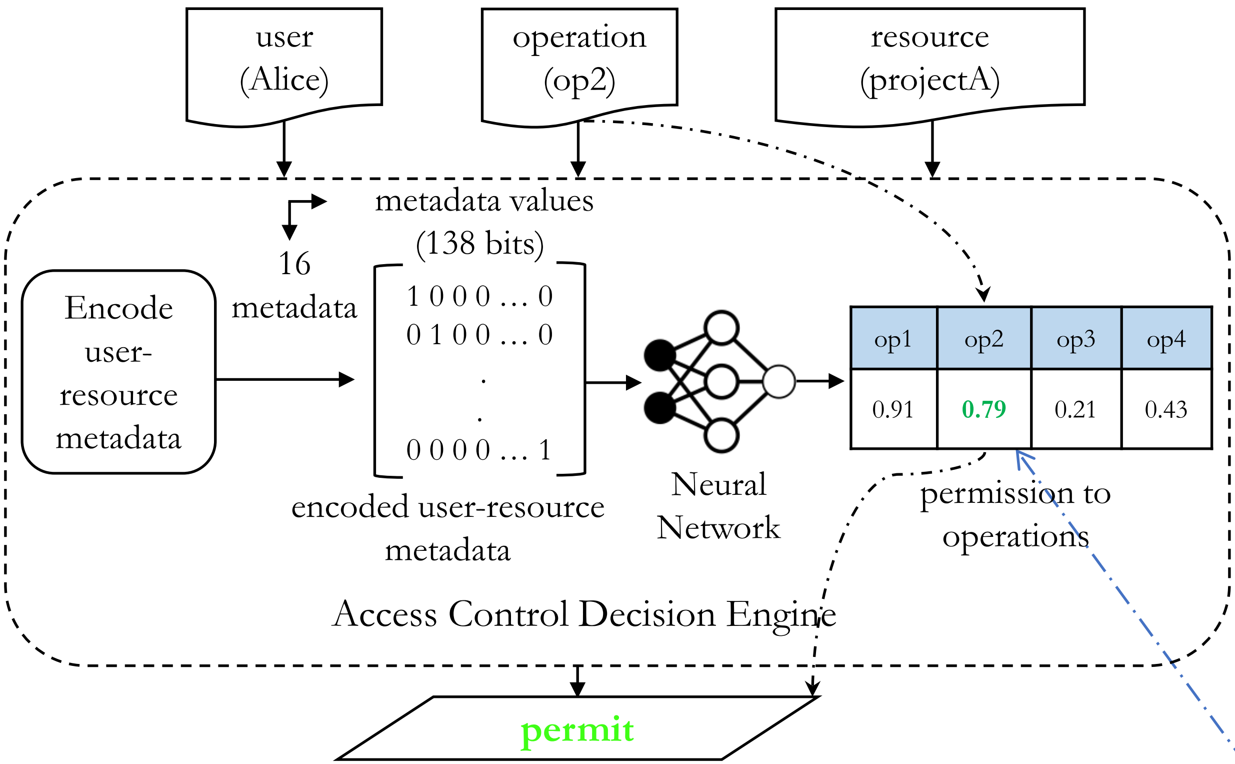


Preparing Training Data for DLBACα

The data type in our datasets are **nominal-categorical**



Decision Making Process in DLBACα



Permit decision is made comparing the output **probability** with a threshold

Evaluation Methodology

Multiple instances of DLBACα

- ResNet ($\text{DLBAC}_{\alpha\text{-R}}$)
- DenseNet ($\text{DLBAC}_{\alpha\text{-D}}$)
- Xception ($\text{DLBAC}_{\alpha\text{-X}}$)

Classical ML Algorithms

- SVM
- Random Forest (RF)
- Multilayer Perceptron (MLP)

State-of-the-art policy mining techniques

- XuStoller [1]
- Rhapsody [2]
- EPDE-ML [3]

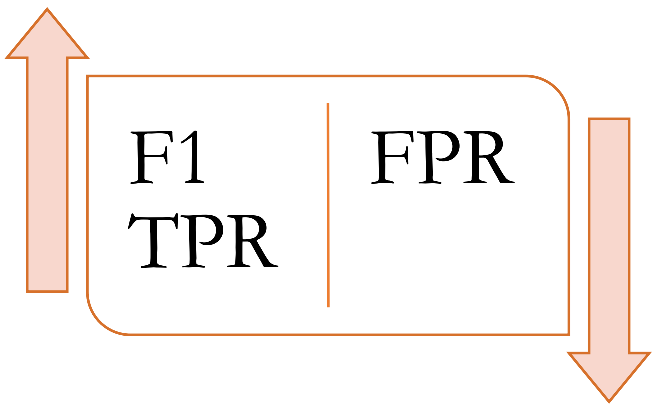
[1] Xu et al. 2014. "Mining attribute-based access control policies." IEEE TDSC

[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

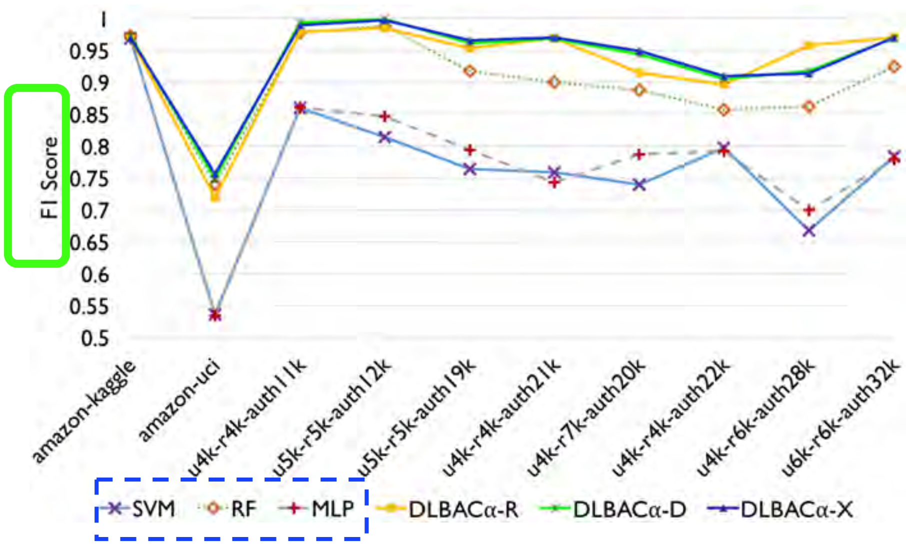


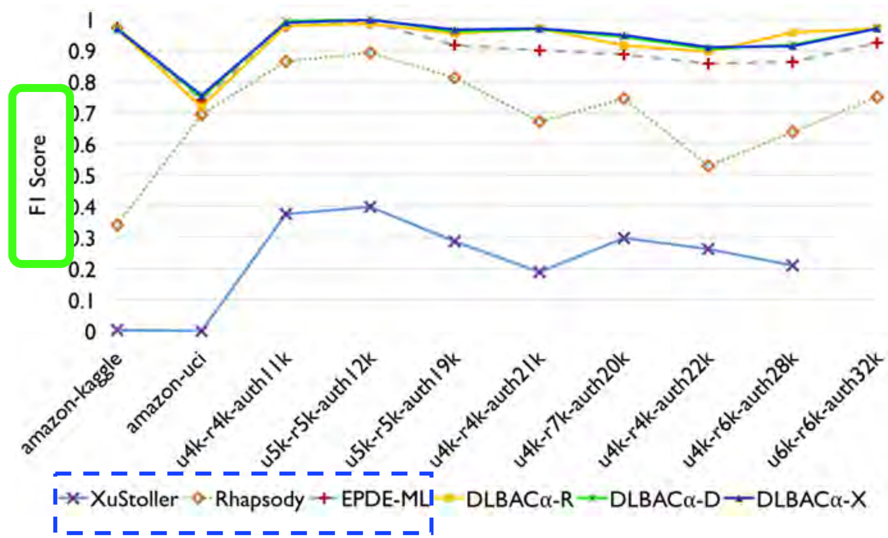
A higher F1 score: better generalization

A higher TPR: accurate and efficient in granting access

A lower FPR: efficient in denying access

Comparison with ML Algorithms and State-of-the-art Policy Mining

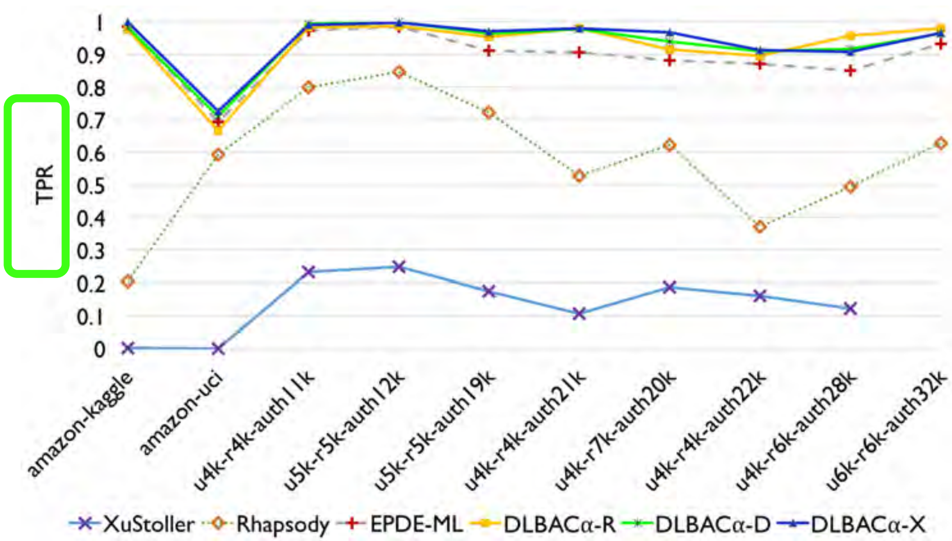




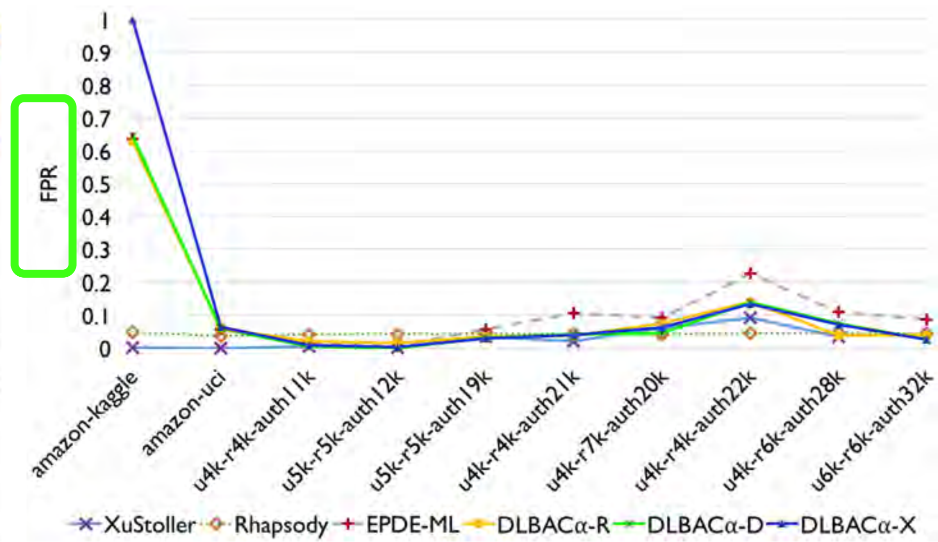
make accurate access decisions and generalize better

Comparison with Policy Mining Algorithms

handling desirable access

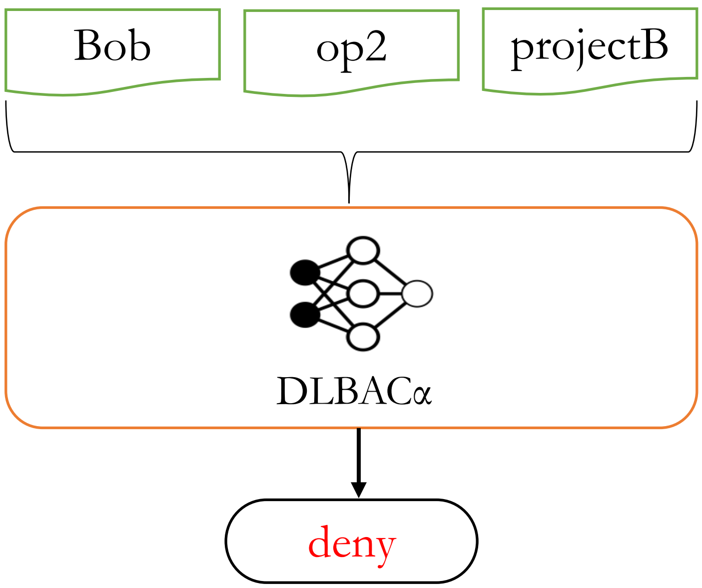


handling unwanted access



Efficient in permitting desired accesses and denying unwanted accesses

Understanding DLBAC Decisions



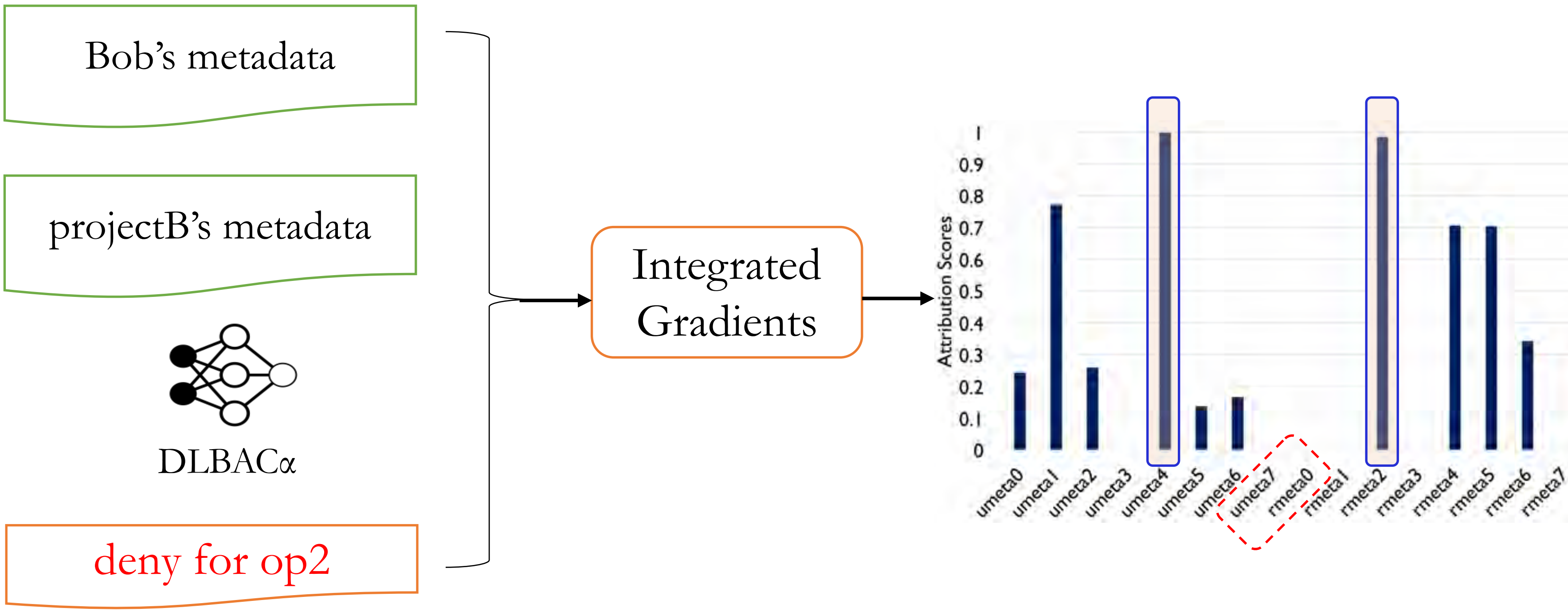
A sample access request

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

Which metadata are important/influential for this decision?

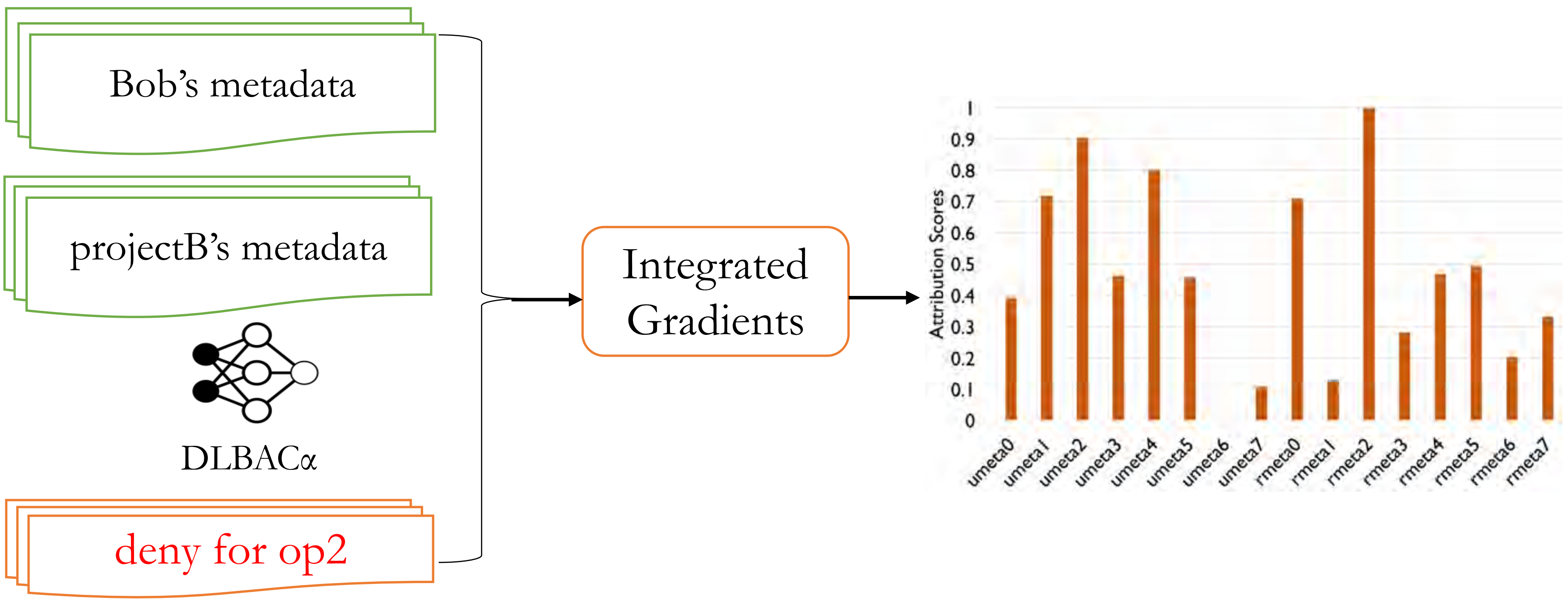
- Two approaches
  - Integrated Gradients
  - Knowledge Transfer

Integrated Gradients



Local Interpretation

Integrated Gradients



Global Interpretation

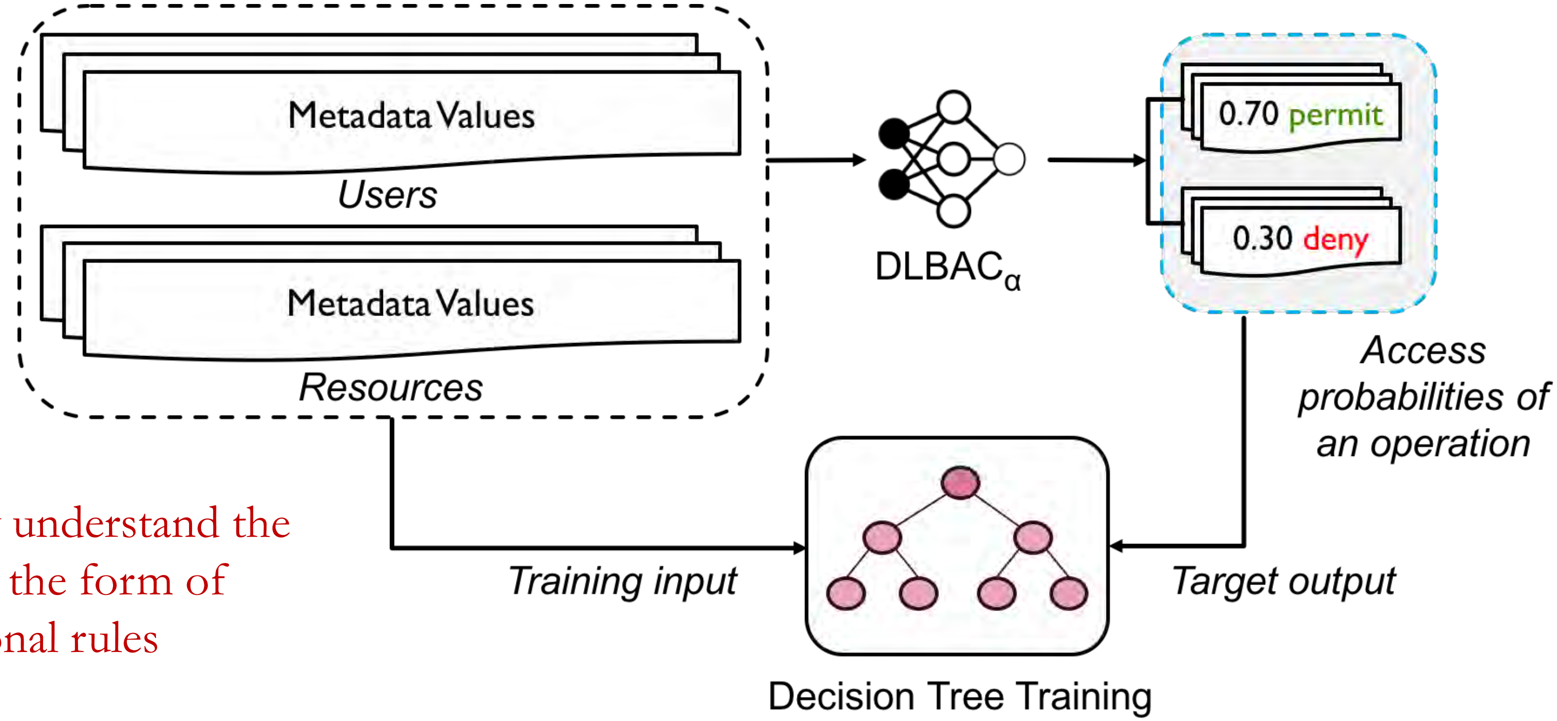
Application of Integrated Gradient-based Understanding

	umeta0	umeta1	umeta2	umeta3	umeta4	umeta5	umeta6	umeta7	rmeta0	rmeta1	rmeta2	rmeta3	rmeta4	rmeta5	rmeta6	rmeta7	access to op1 operation
tuple1	30	51	126	26	37	129	89	5	5	123	95	40	37	129	89	14	deny
	Carol’s metadata								projectC’s metadata								
tuple2	61	84	5	29	44	105	6	30	5	123	95	40	37	129	89	14	permit
	Dave’s metadata								projectC’s metadata								
modified tuple1	30	84	126	26	44	129	89	5	5	123	95	40	37	129	89	14	permit
	Carol’s metadata								projectC’s modified metadata								

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

Is there any relation among metadata?

Knowledge Transferring



approximately understand the decision in the form of traditional rules

- Rule: local interpretation
- DT: global interpretation

Section-3

Machine Learning Based Access Control (MLBAC)

State of the Art: ML in Access Control

Operational Model of MLBAC

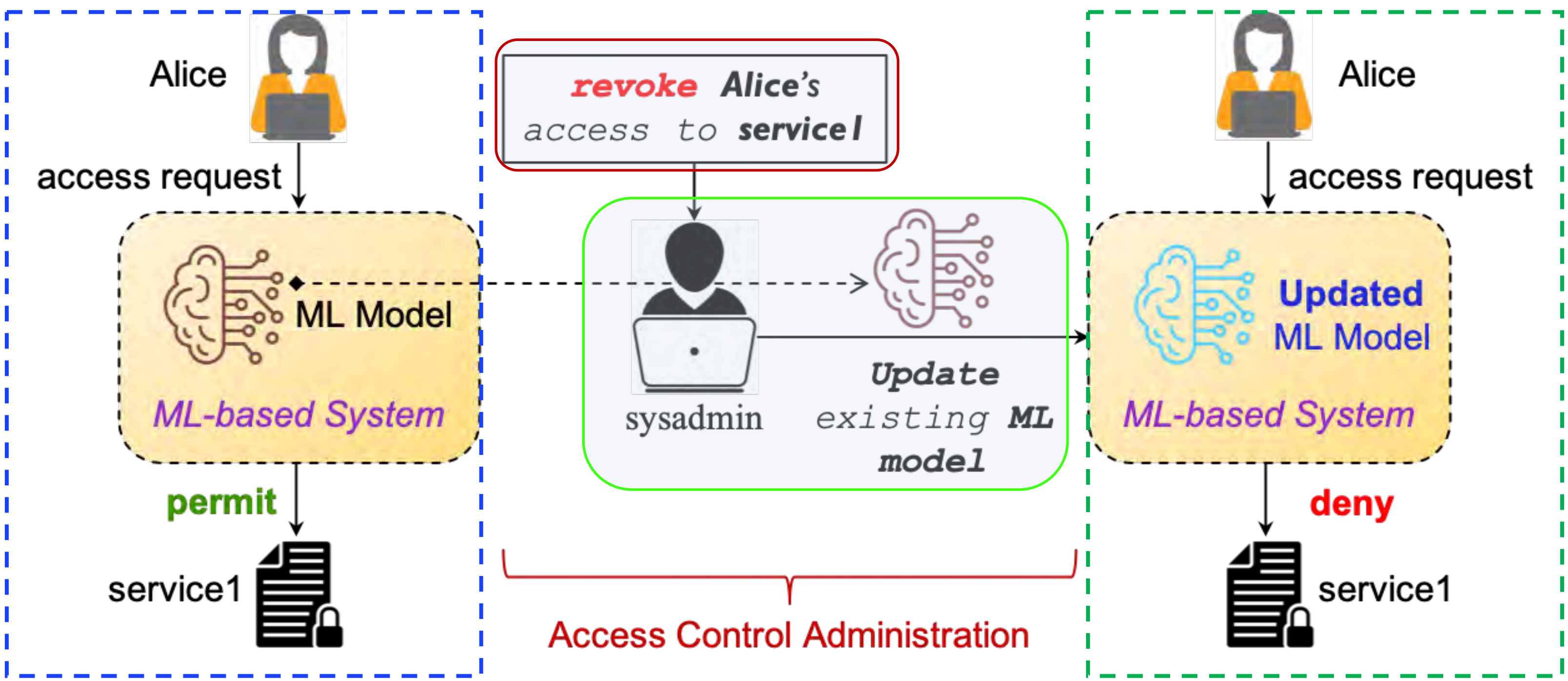
Administration of MLBAC

DLBAC (prototype, interpretation)

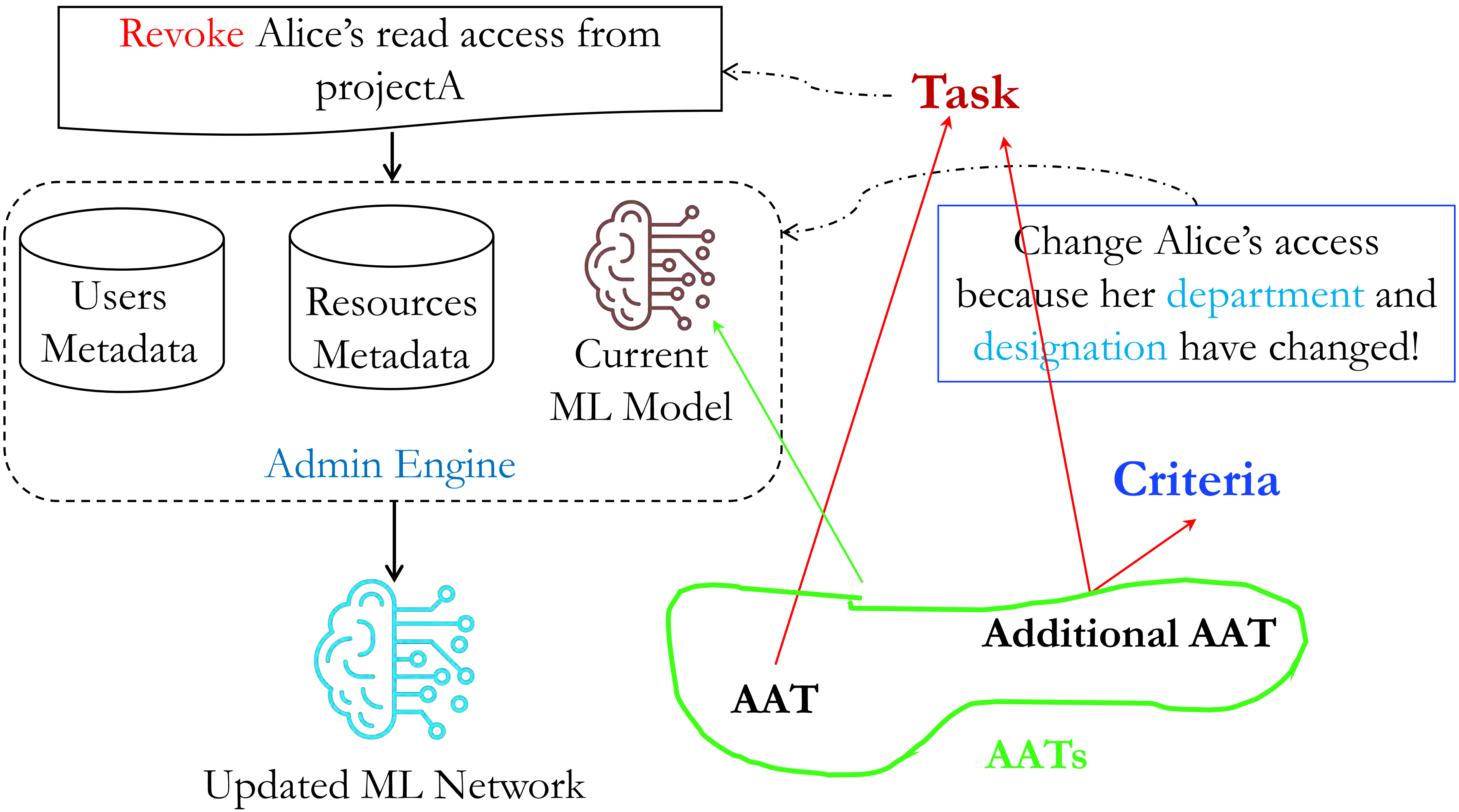
Adversarial Attacks in DLBAC

Implementation and Evaluation of DLBAC

Administration in Machine Learning Based Access Control

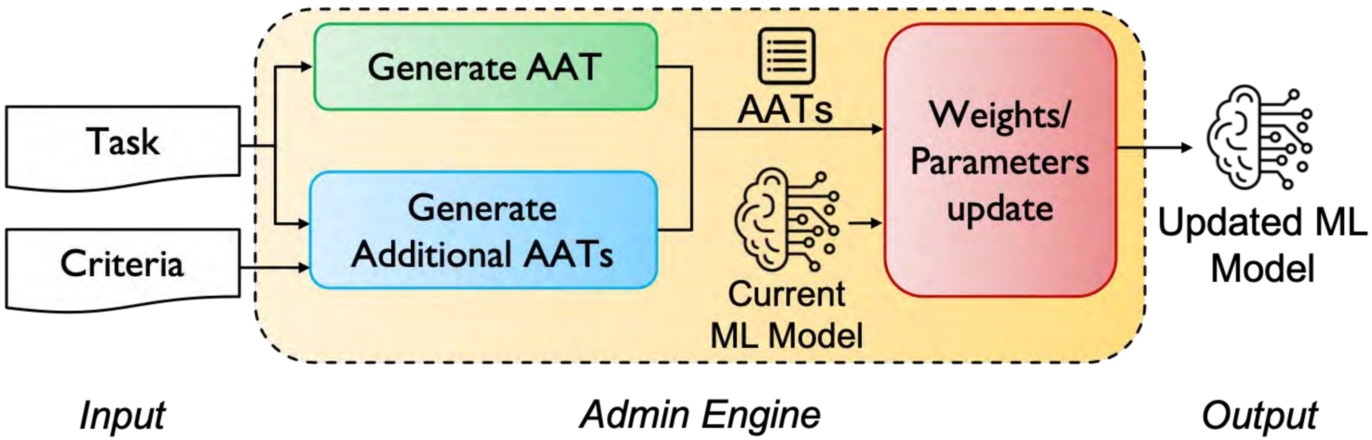


MLBAC Administration Overview

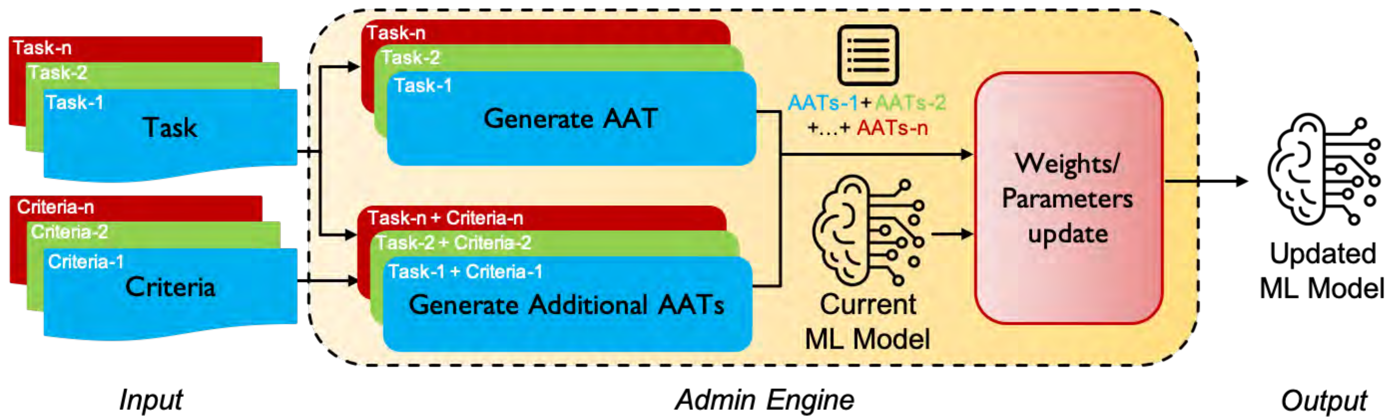


Administration Process Flow

Single Task

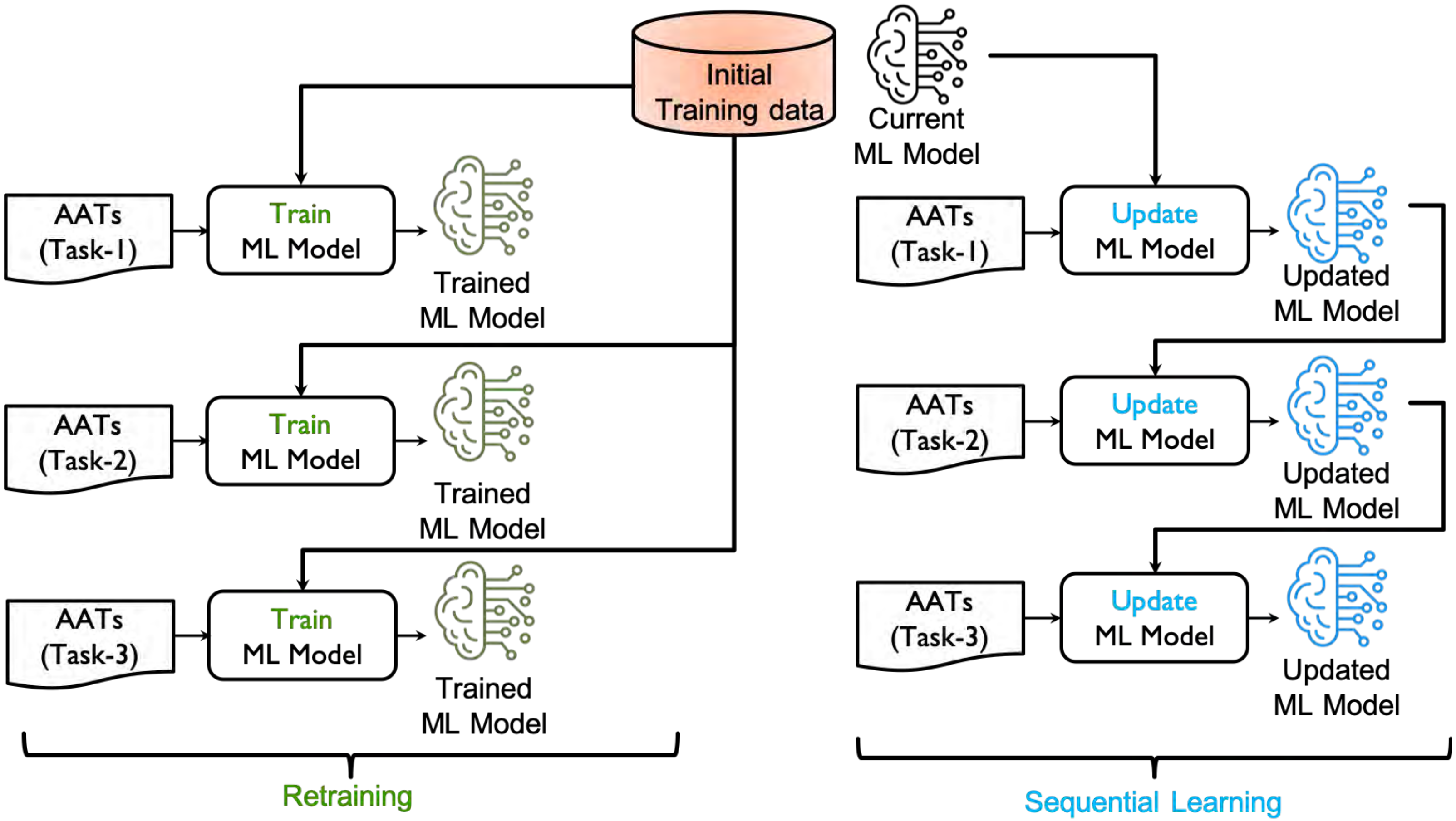


Multi Task



Simulate 2-Tasks, 3-Tasks, and 6-Tasks

Weights/Parameters Update

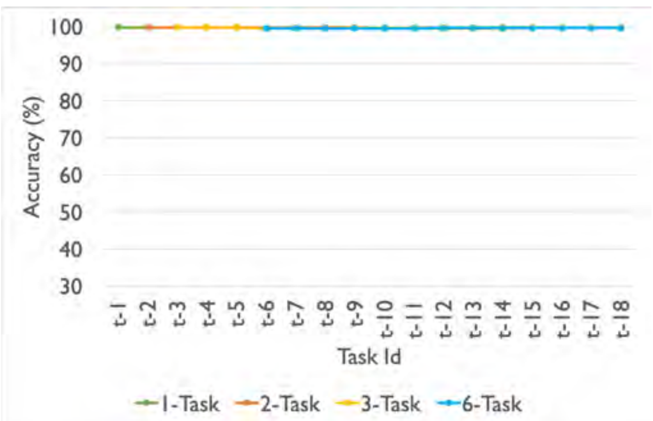


18 random Tasks with different Criteria

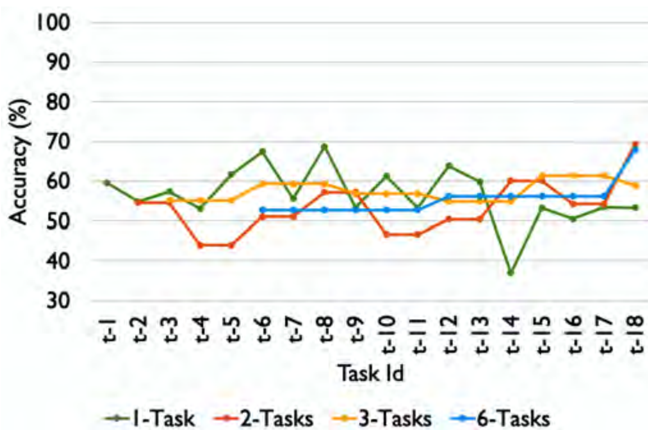
Performance Evaluation

- RF-MLBAC: Add additional estimators
- ResNet-MLBAC: Fine-tuning

- How accurately can it learn new changes (AATs)
- How well can it preserve existing access states for all other users/resources (**OATs**)



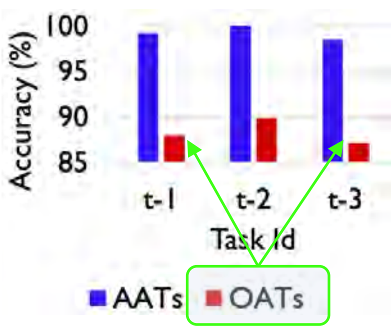
OATs



AATs

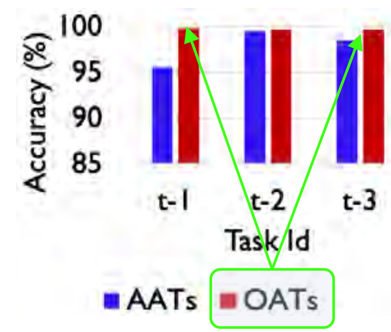
Unable to accommodate new changes with good accuracy.

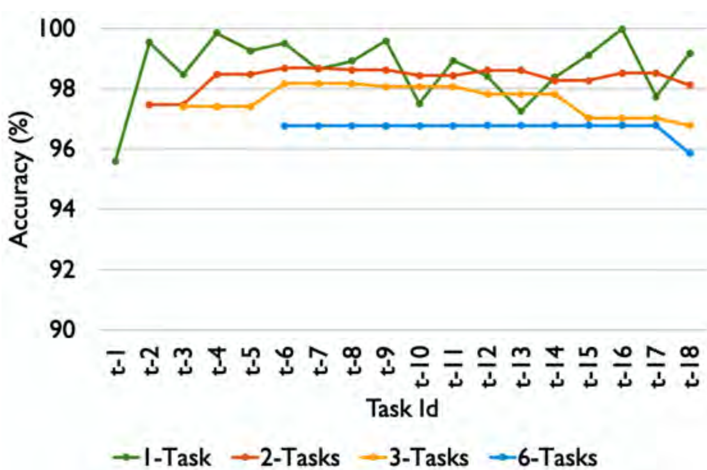
Performance Evaluation (cont'd) (ResNet-MLBAC)



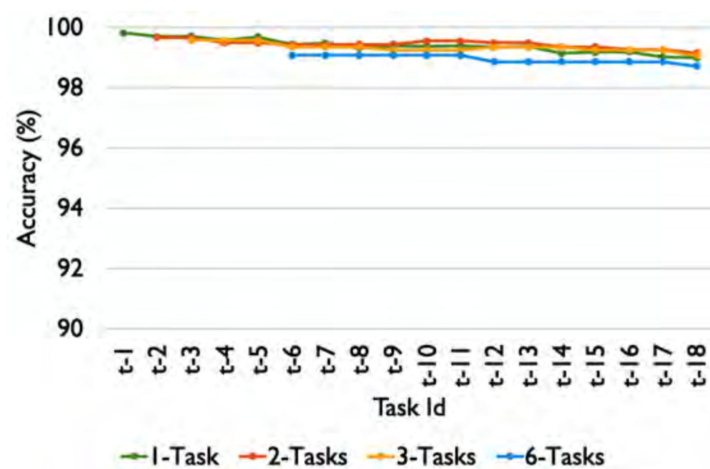
Starts to forget other Access Control state - Catastrophic forgetting

Replay Data





AATs



OATs

Multi-task administration generally provides better performance

Section-4
(Part-A)

Machine Learning Based Access Control (MLBAC)

State of the Art: ML in Access Control

Operational Model of MLBAC

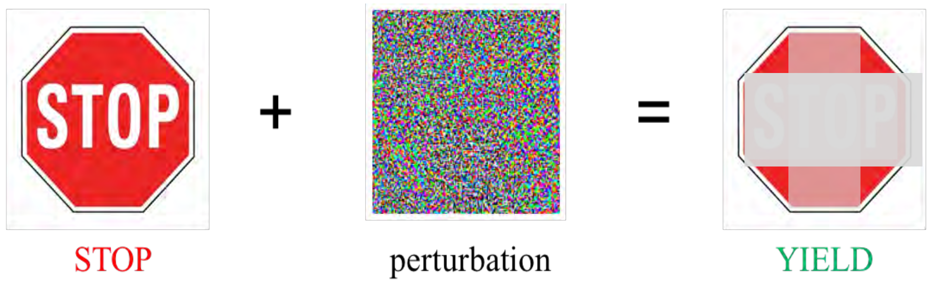
Administration of MLBAC

DLBAC
(prototype, interpretation)

Adversarial Attacks in DLBAC

Implementation and Evaluation of DLBAC

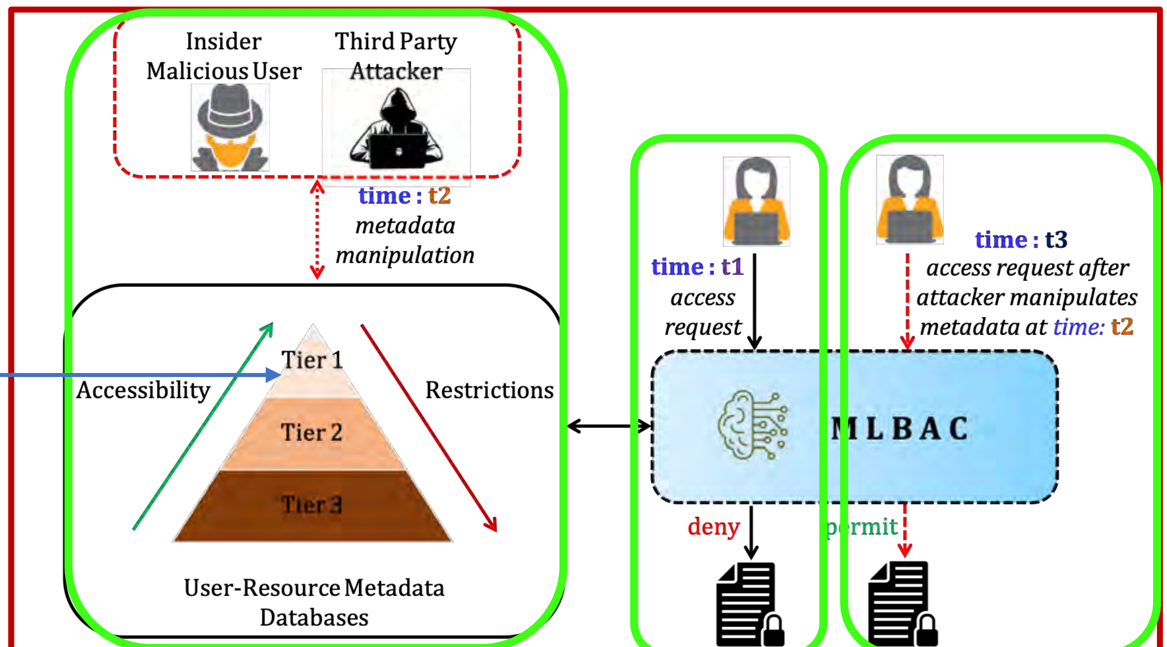
Adversarial Attack in MLBAC



Adversarial Example

Imperceptibility

Modify part of the input to any degree



Adversarial Attack Problem

Actual decision

Perturbation

$$f(x) = y \neq f(x + x_p) = t$$

Target decision

Perturbation weight

perturbation $$g(x_p) = \mathcal{L}(x + x_p,\, t) + \omega \,\|\, x_p \,\|$$

Accessibility Constraint

Access Restriction $$g(x_p) = \mathcal{L}(x + x_p,\, t) + \omega \,\|\, x_p \circ c \,\|$$

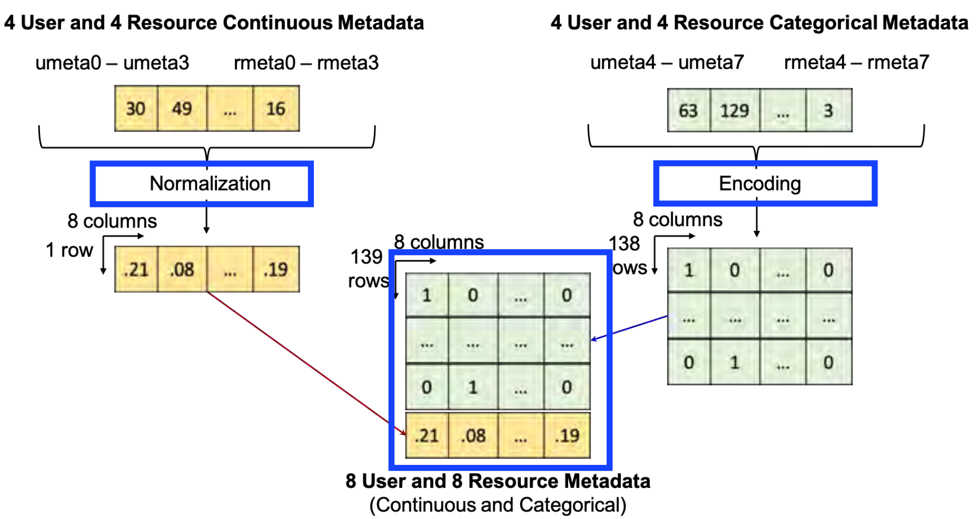
Mitigation Approach

Continuous and Categorical

'age,' 'salary', 'security_level,' 'designation'

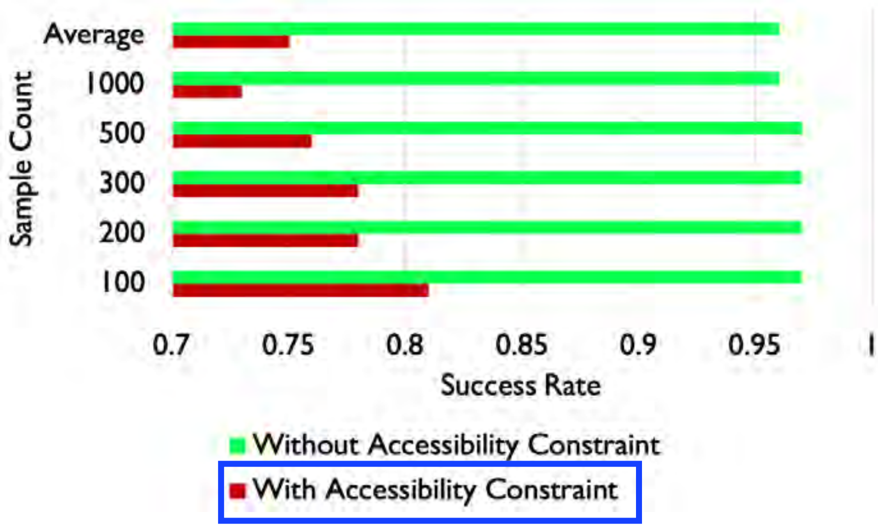
- Accessibility Constraint
  - Pearson's Correlation
  - Value between 0 and 1
  - Higher correlation, more restricted

- Two DLBAC datasets
  - System-1 and System-2

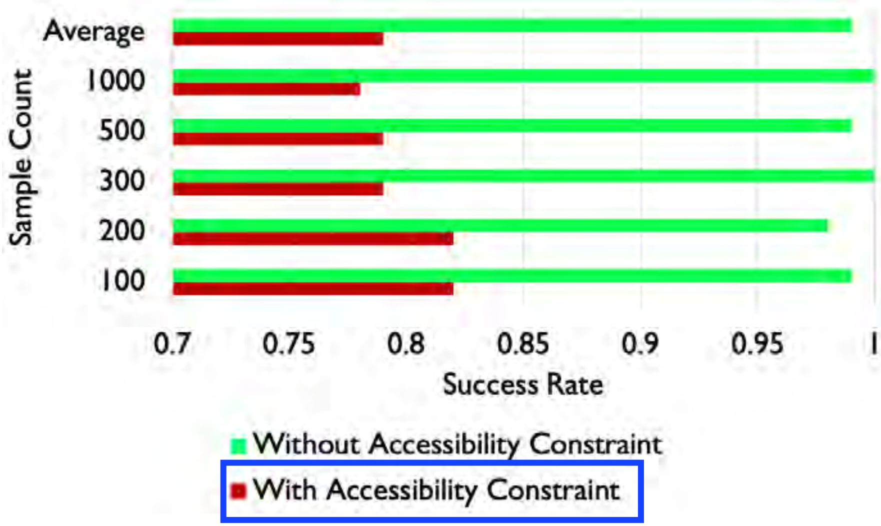


Evaluation

$$\textbf{Success Rate} = \frac{\text{Successfully crafted adversarial examples}}{\text{Samples attempted for the adversarial example creation}}$$



System-1



System-2

Section-4
(Part-B)

Machine Learning Based Access Control (MLBAC)

Comprehensive Literature Review : ML in Access Control

Operational Model of MLBAC

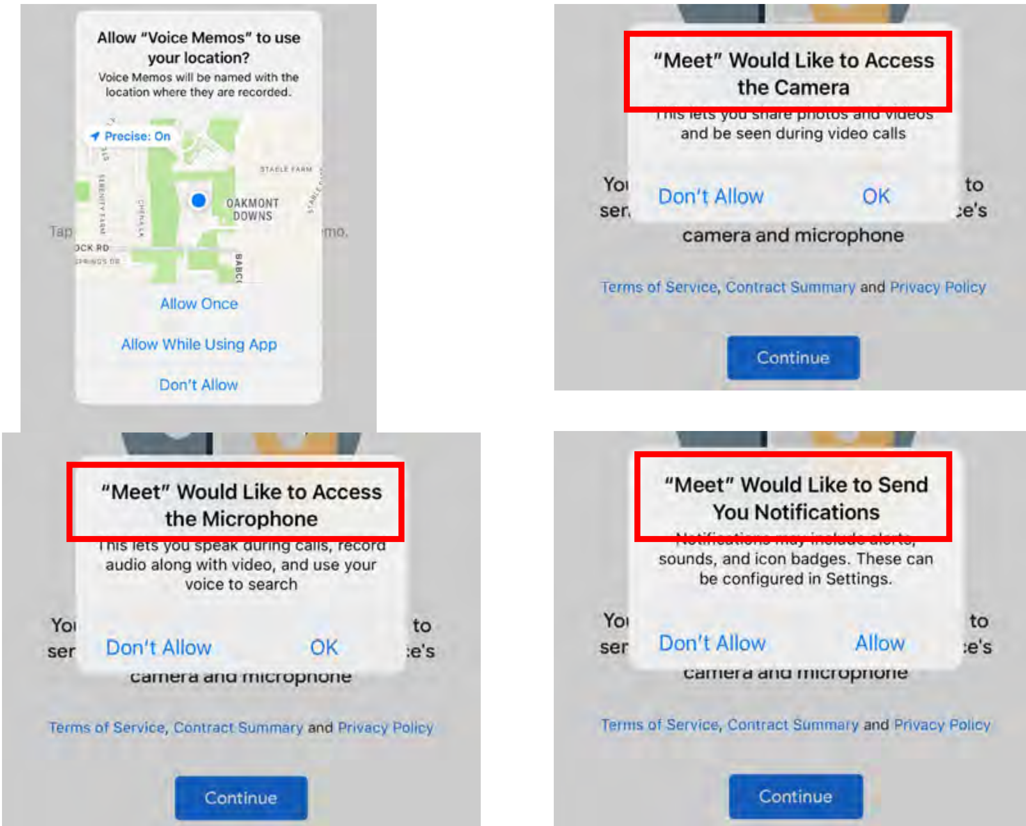
Administration of MLBAC

DLBAC
(prototype, interpretation)

Adversarial Attacks in DLBAC

Implementation and Evaluation of DLBAC

DLBAC Assisted Permission Recommendation for Mobile Devices



Ask-On-Install (AOI)

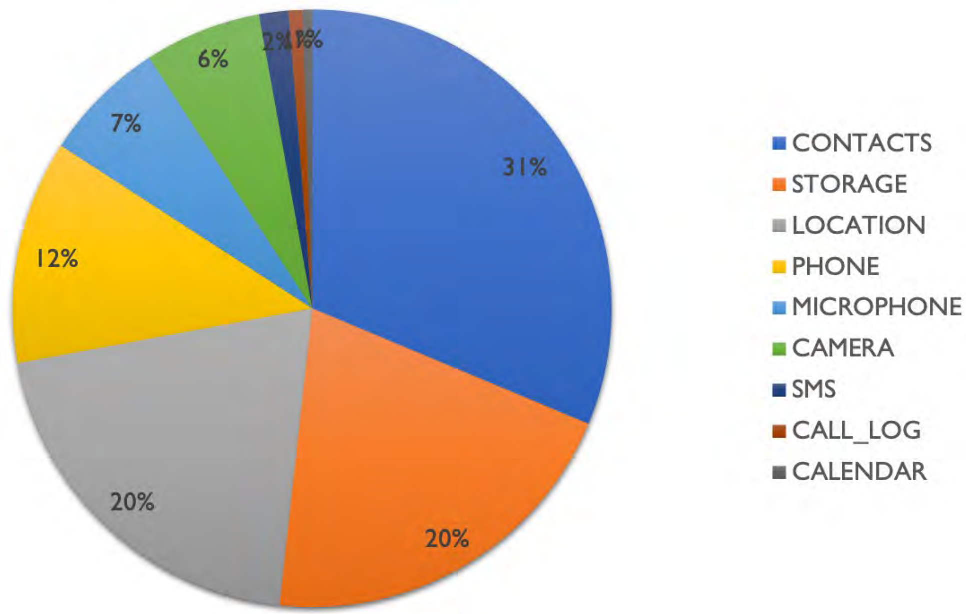
Ask-On-First-Use (AOFU)

Could DLBAC automate this permission decision?

… abundant permission requests

COP-MODE Dataset

- Developed by Mendes et al. [4], **65K** permission requests
- At each permission request:
  - **Requesting application:** name and play store category
  - **Permission:** name (CONTACTS, STORAGE, etc.) and grant result (allow/deny)
  - **Phone state:** geolocation, plug, call state, network connection , etc.
  - **User context:** time, semantic location, in event or not, etc.

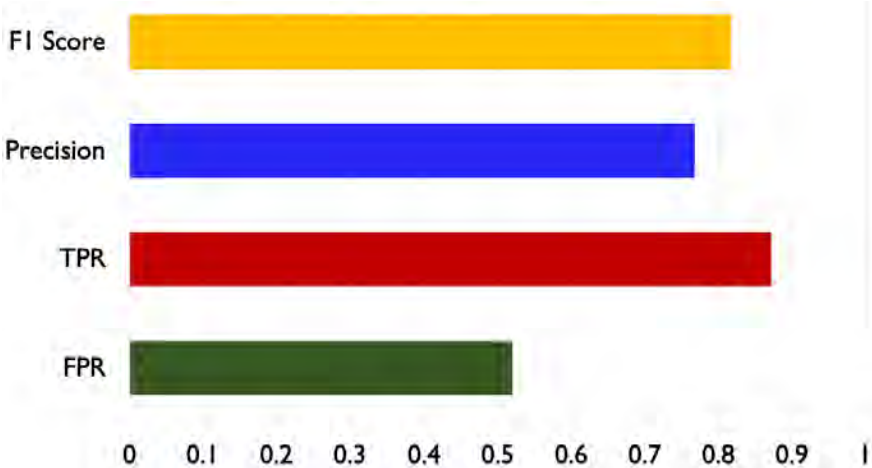


[4] . Mendes, R., Brandão, A., Vilela, J. P., and Beresford, A. R.. Effect of User Expectancy on Mobile App Privacy: A Field Study. In 2022 IEEE PerCom.

Evaluation

- Three DLBAC instances with: ResNet, DenseNet, and Xception

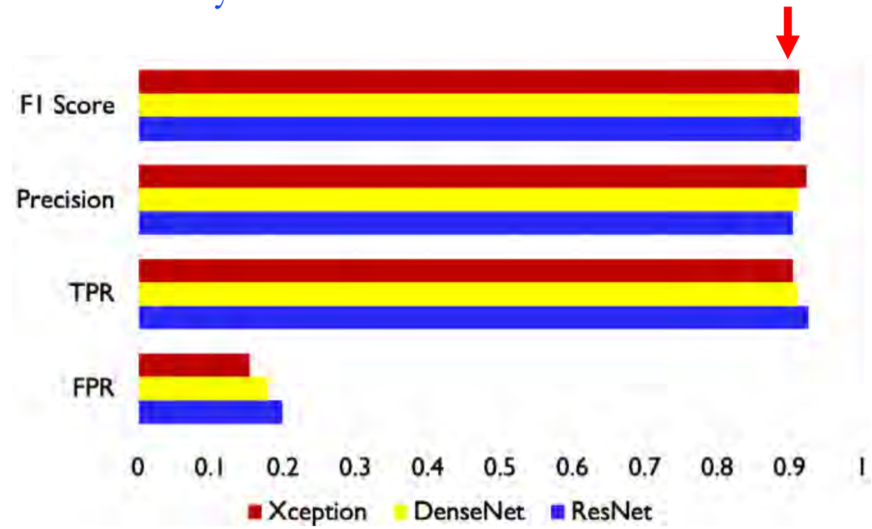
Accuracy: **74.02%**



DLBAC Performance (ResNet)

Cluster like-minded users, Liu et al. [6]

Accuracy: ~**88.5 %** F1 Score: ~0**.915**



DLBAC Instances Performance

[5]. Brandão, A. et al. Prediction of Mobile App Privacy Preferences with User Profiles via Federated Learning. In 2022 ACM CODASPY.
[6]. Liu et al. Follow My Recommendations: A Personalized Privacy Assistant for Mobile App Permissions. In SOUPS 2016.

Future Research Directions

DLBAC Issues
- Understanding, Administration, etc.
- Accuracy is lower in some cases

MLBAC Verification
- Measuring Correctness
- Testing Framework

Bias and Fairness
- Data could come from untrusted sources
- Imbalanced data may bias the decision

Adversarial Issues
- Adversarial attack for Classical ML based systems
- Need more strong defense mechanisms

DLBAC in Tandem
- Reinforcing access decision
- Monitoring and feedback

Selected Publications

- Closest
  - (ACM CODASPY 2022) Nobi, Mohammad Nur, Ram Krishnan, Yufei Huang, Mehrnoosh Shakarami, and Ravi Sandhu. "Toward Deep Learning Based Access Control."
  - (ESORICS 2022) Mohammad Nur Nobi, Ram Krishnan, Yufei Huang, and Ravi Sandhu. "Administration of Machine Learning Based Access Control".
  - (itaDATA 2022) Mohammad Nur Nobi, Ram Krishnan, and Ravi Sandhu. "Adversarial Attacks in Machine Learning Based Access Control".
  - (ACM Computing Surveys, *under review*) Mohammad Nur Nobi, Maanak Gupta, Lopamudra Praharaj, Mahmoud Abdelsalam, Ram Krishnan, and Ravi Sandhu. "Machine Learning in Access Control: A Taxonomy and Survey".
- Relevant
  - (ACM CCS 2013) Philip Fong, Pooya Mehregan and Ram Krishnan, Relational Abstraction in Community-Based Secure Collaboration
  - (ACM TOPS) Ram Krishnan, Jianwei Niu, Ravi Sandhu and William H. Winsborough, Group-Centric Secure Information Sharing Models for Isolated Groups

Source code and datasets URL:

https://github.com/dlbac/DlbacAlpha
https://github.com/mlxac/MLBAC-Admin
https://github.com/mlxac/MLBAC-AdversarialAttack

THANK YOU!
Q&A