04 December 2025 / Privacy Aware Computing

# Seedless & Seed-based Deanonymization

# Membership Inference Attack on DisTilBert Model

S. Vignesh Kumar Pandian[1], Apu Kumar Chakroborti[1]

[1] Department of Computer Science, Georgia State University

# Outline

1. Individual Project - Vignesh
2. Individual Project – Apu Kumar Chakroborti
3. Group Project
4. Future Scope

We're doing it #thestateway

# Section 1

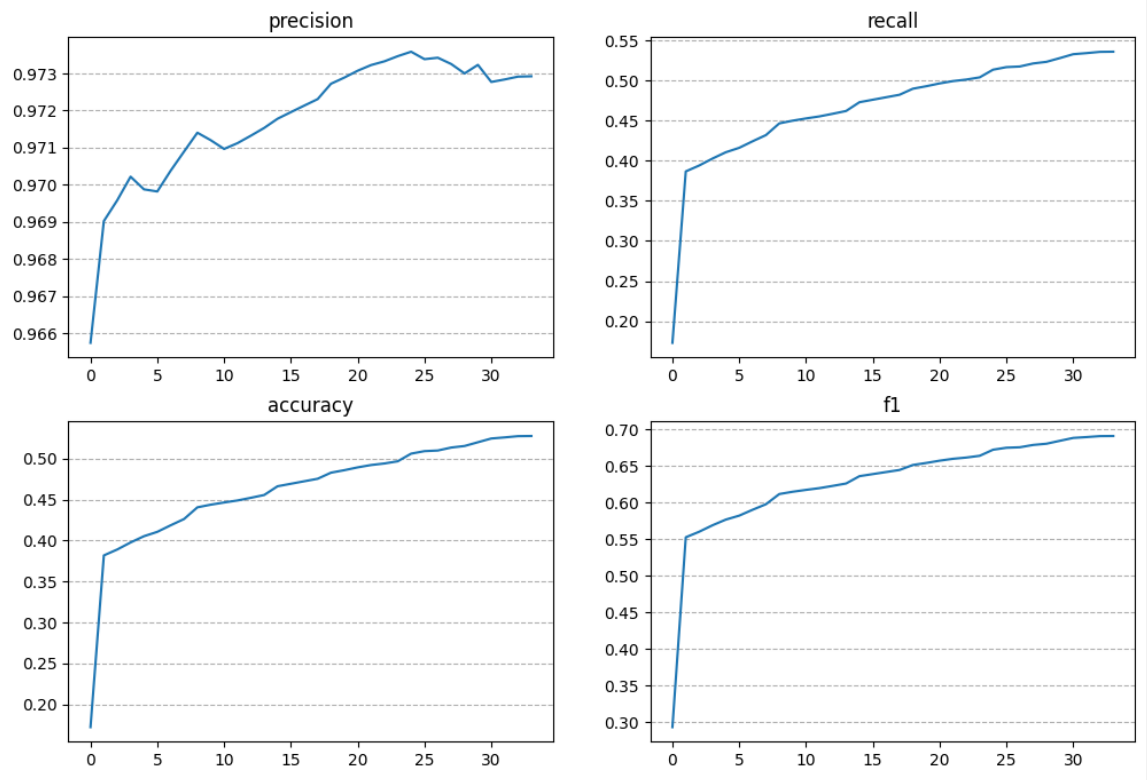
# Individual Project
# Vignesh Kumar

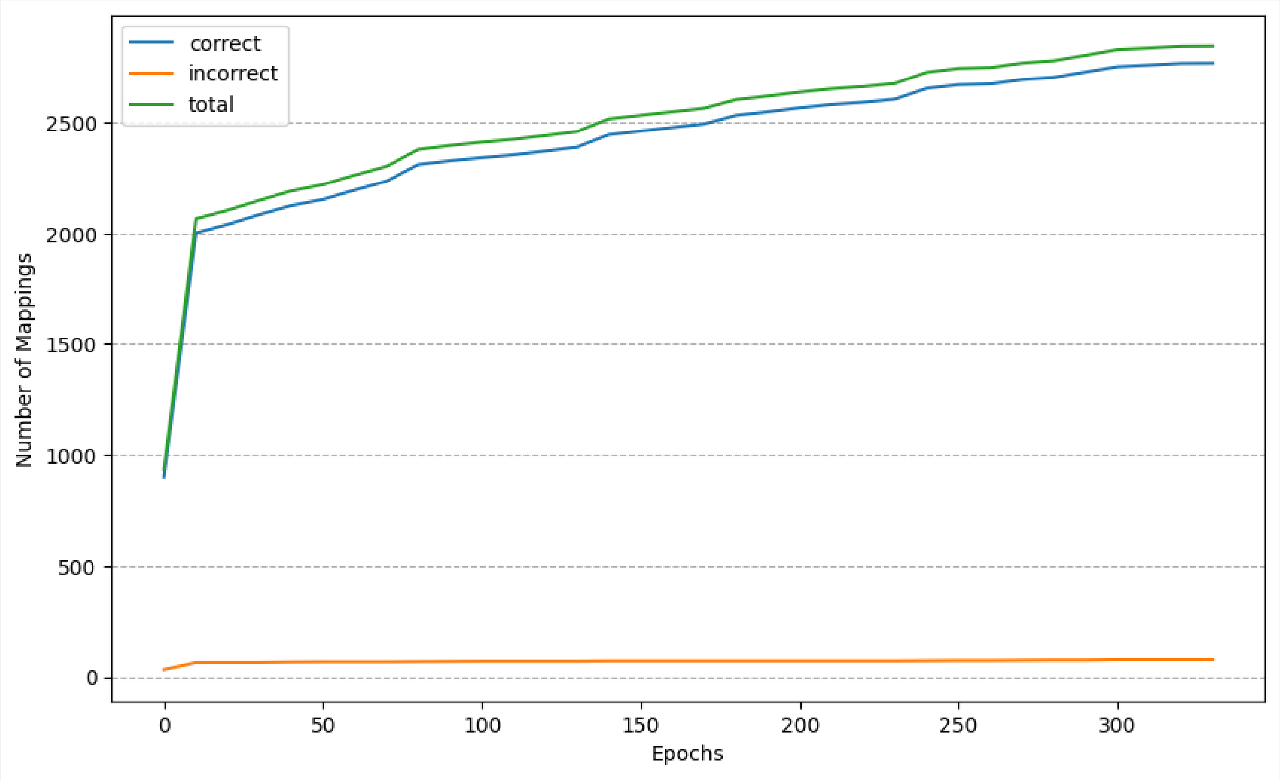
We're doing it #thestateway

# Seed-based Deanonymization

- Multi-hop neighborhood
  - Distance-based Weighting for hop importance
- Threshold Scheduling
- Quantile-based Thresholding
- Sparse Matrix computation
- Repeated Training (dropped)
- GPU-based Training
- Ablation
  - Network Hops
  - Eccentricity threshold
  - Quantile-based threshold

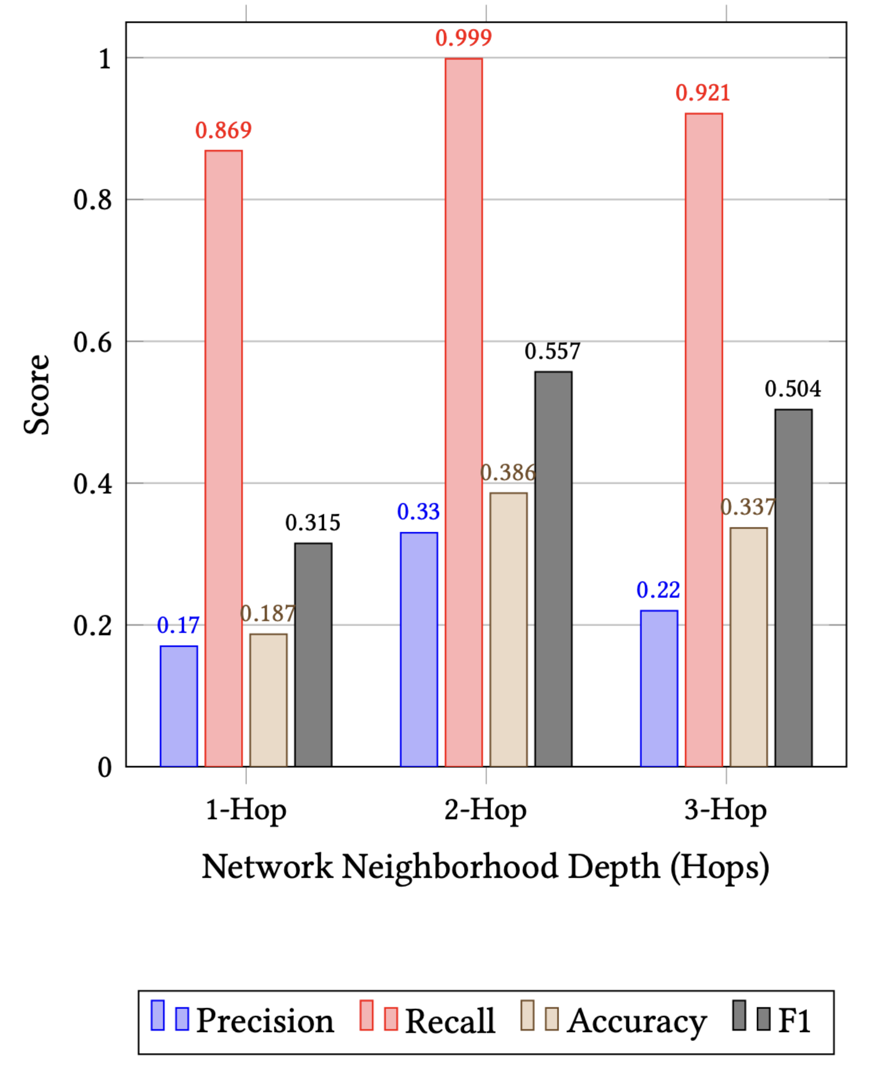
# Results on Validation

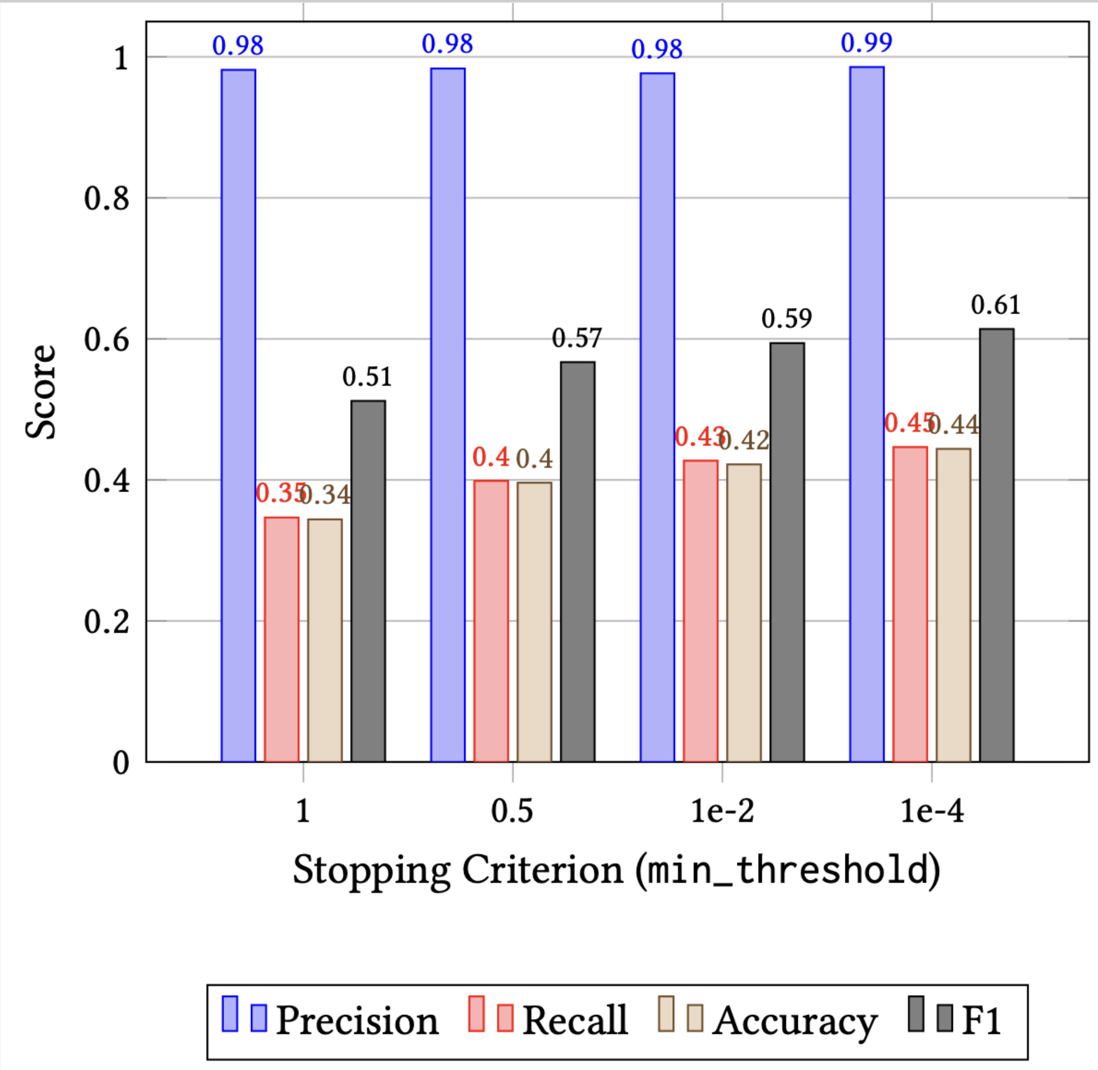



**We're doing it #thestateway**

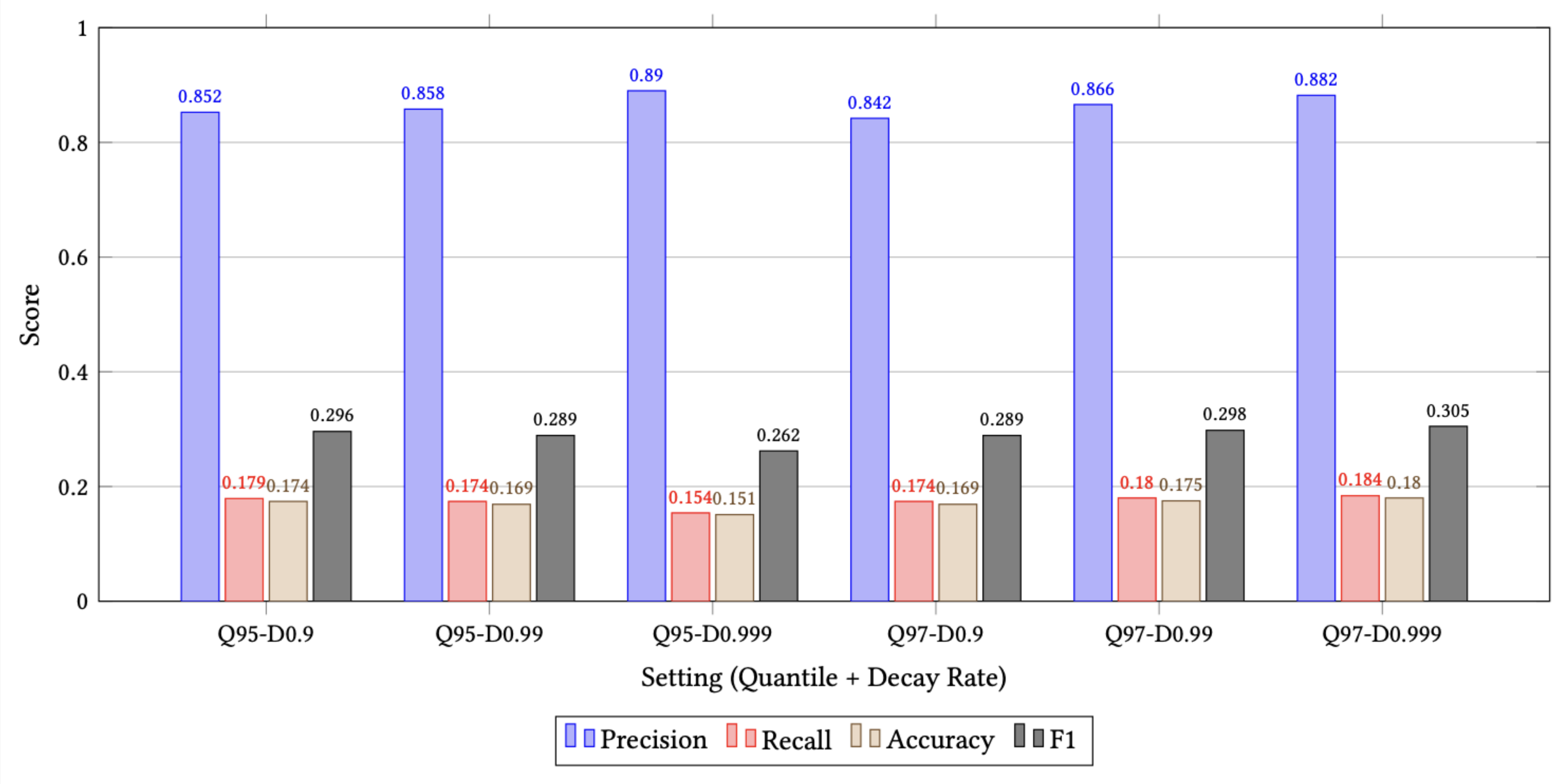
# Ablation




We're doing it #thestateway

# Ablation




## Best Model Performance

| | |
|---|---|
| Correct | :2767 |
| Incorrect | : 77 |
| Missing | : 2398 |
| Precision | : 0.97292 |
| Accuracy | : 0.52785 |
| F1 score | : 0.69097 |
| Recall | : 0.53572 |

## Configuration

2-Hop
Threshold 6
0.99 threshold decay

**We're doing it #thestateway**

# Seed-free Deanonymization

- Calculate Pseudo labels using heuristics
  - Degree
  - Two hop neighbours
  - Average neighbour degree
  - Clustering coefficient
- Use Seed-based method using Pseudo labels
- Threshold Scheduling
- Sparse Matrix computation
- GPU-based Training

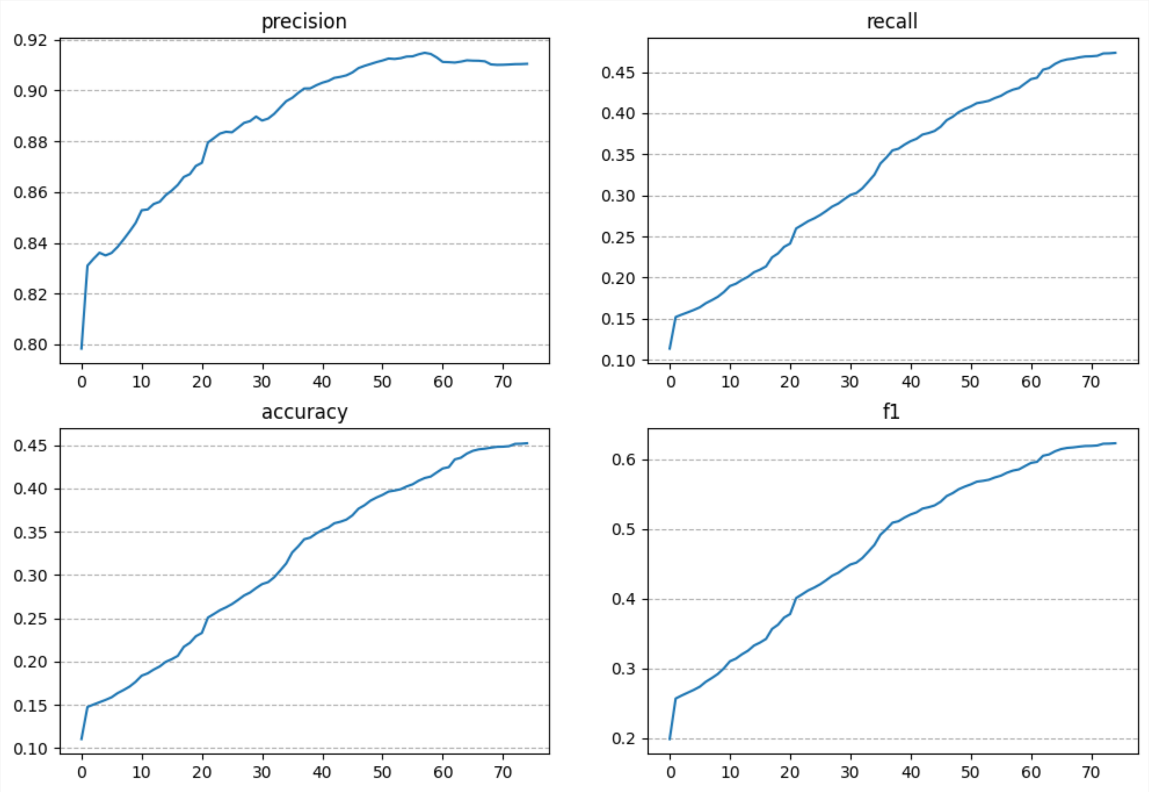
**Best Model Performance**

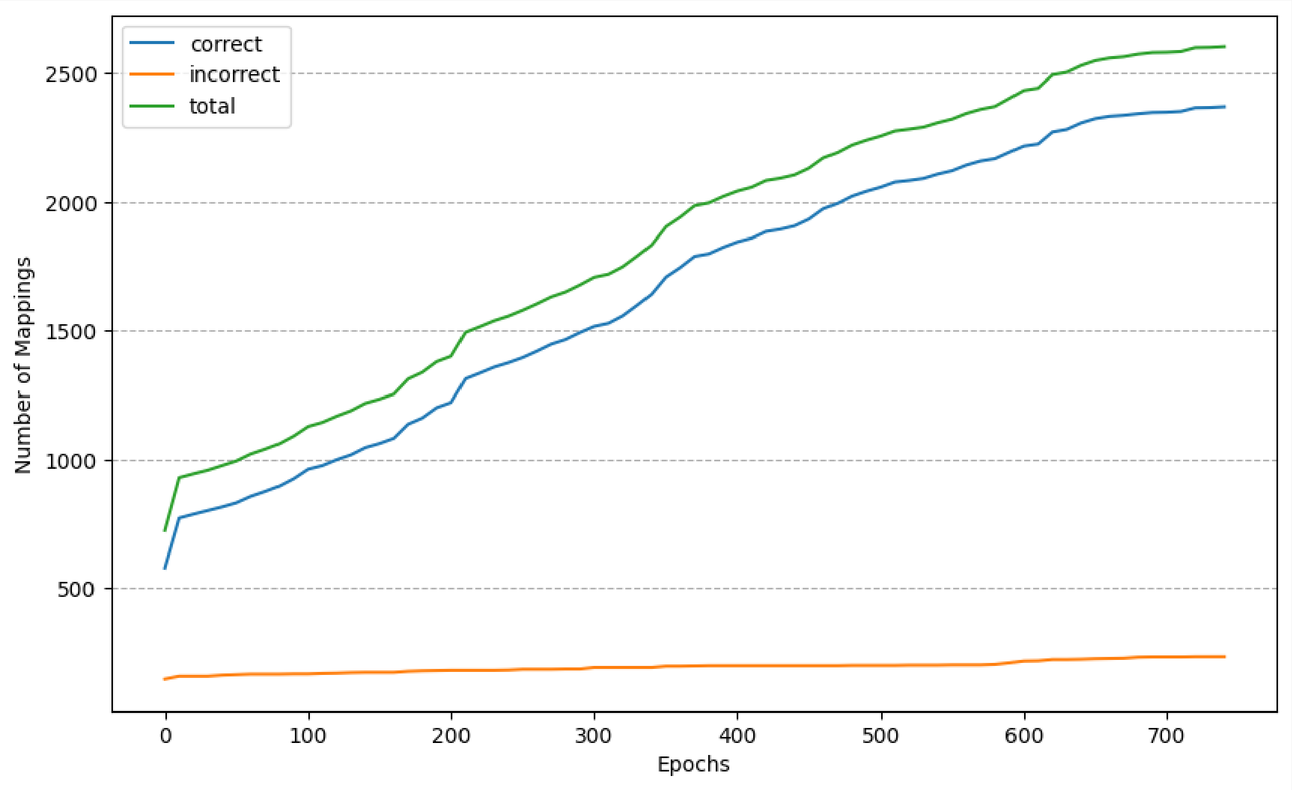
Correct :2376
Incorrect : 233
Missing : 2662

Precision : 0.91057
Accuracy : 0.46792
F1 score : 0.63870
Recall : 0.47866

**Configuration**
2-Hop
Threshold 10
0.99 threshold decay
Min_threshold 1e-2

# Results on Validation




We're doing it #thestateway

# Section 2

# Individual Project
# Apu Kumar Chakroborti

We're doing it #thestateway

# Introduction

- Many real-world datasets (**social networks**, **communication graphs**, **mobility networks**) are anonymized by replacing user identities with random labels.

- **Graph deanonymization** aims to recover the true correspondences between two graphs $\boldsymbol{G1}$ (original/**anonymized**) and $\boldsymbol{G2}$ (**auxiliary graph** with identity information) based on structural similarity.

- Two problem settings:
  - **Seed-based deanonymization**: a subset of node pairs $(ui, vi)$ is already known.
  - **Unseeded deanonymization**: **no initial mappings**; algorithm must infer initial matches automatically.
- Goal: produce a reliable full mapping $f{:}V(G1){\rightarrow}V(G2)$ using structural, neighborhood, and iterative propagation techniques.

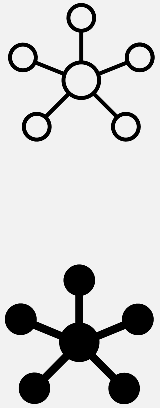
We're doing it #thestateway

# Seed based/free Deanonymization Method

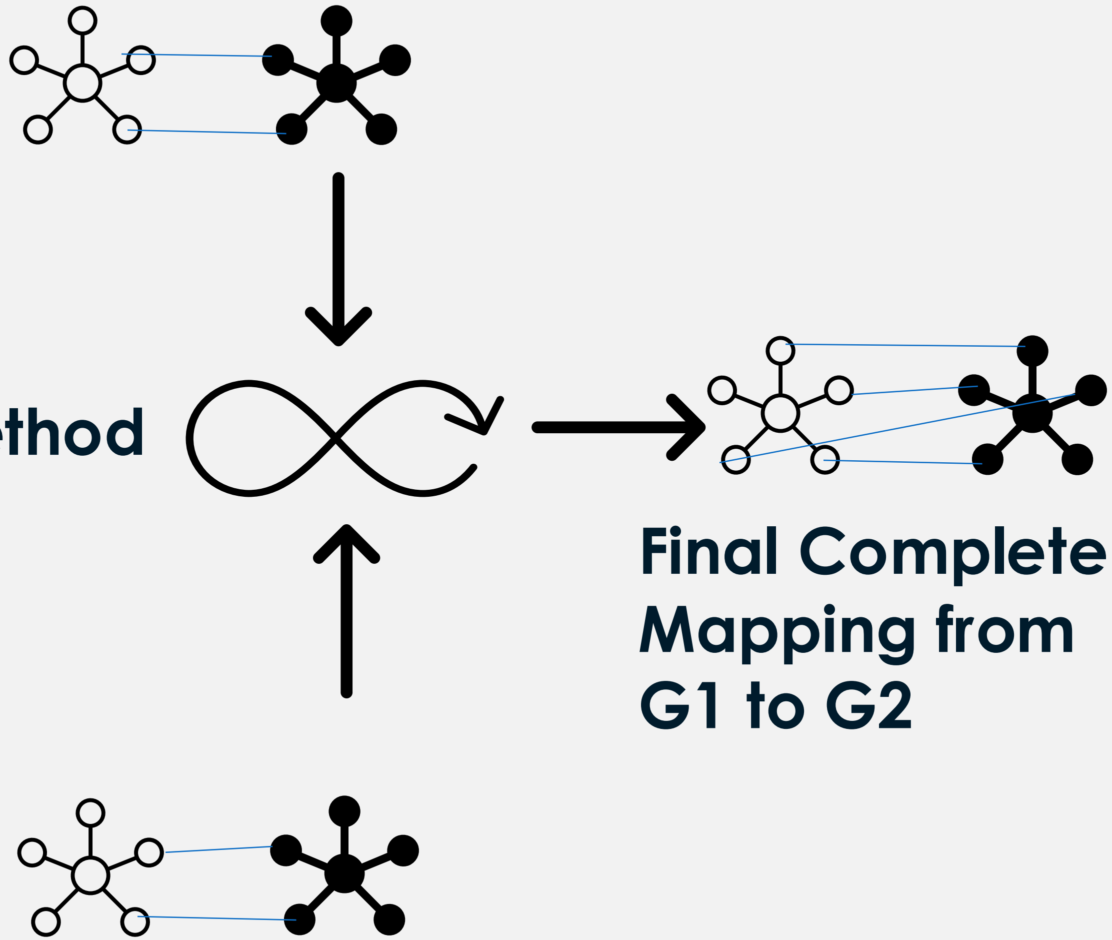
**Seed Based Method**:
Initial mapping

G1

G2



**Propagation Method**



**Final Complete Mapping from G1 to G2**

**Seed free Method**: structures: degree, local clustering coefficient, and page rank

**We're doing it #thestateway**

# Propagation Method(1)

```
function propagationStep(lgraph, rgraph, mapping)

  for lnode in lgraph.nodes:
    scores[lnode] = matchScores(lgraph, rgraph, mapping, lnode)
    if eccentricity(scores[lnode]) < theta: continue
    rnode = (pick node from right.nodes where
          scores[lnode][node] = max(scores[lnode]))

    scores[rnode] = matchScores(rgraph, lgraph, invert(mapping), rnode)
    if eccentricity(scores[rnode]) < theta: continue
    reverse_match = (pick node from lgraph.nodes where
          scores[rnode][node] = max(scores[rnode]))
    if reverse_match != lnode:
      continue

  mapping[lnode] = rnode
```

# Propagation Method(2)

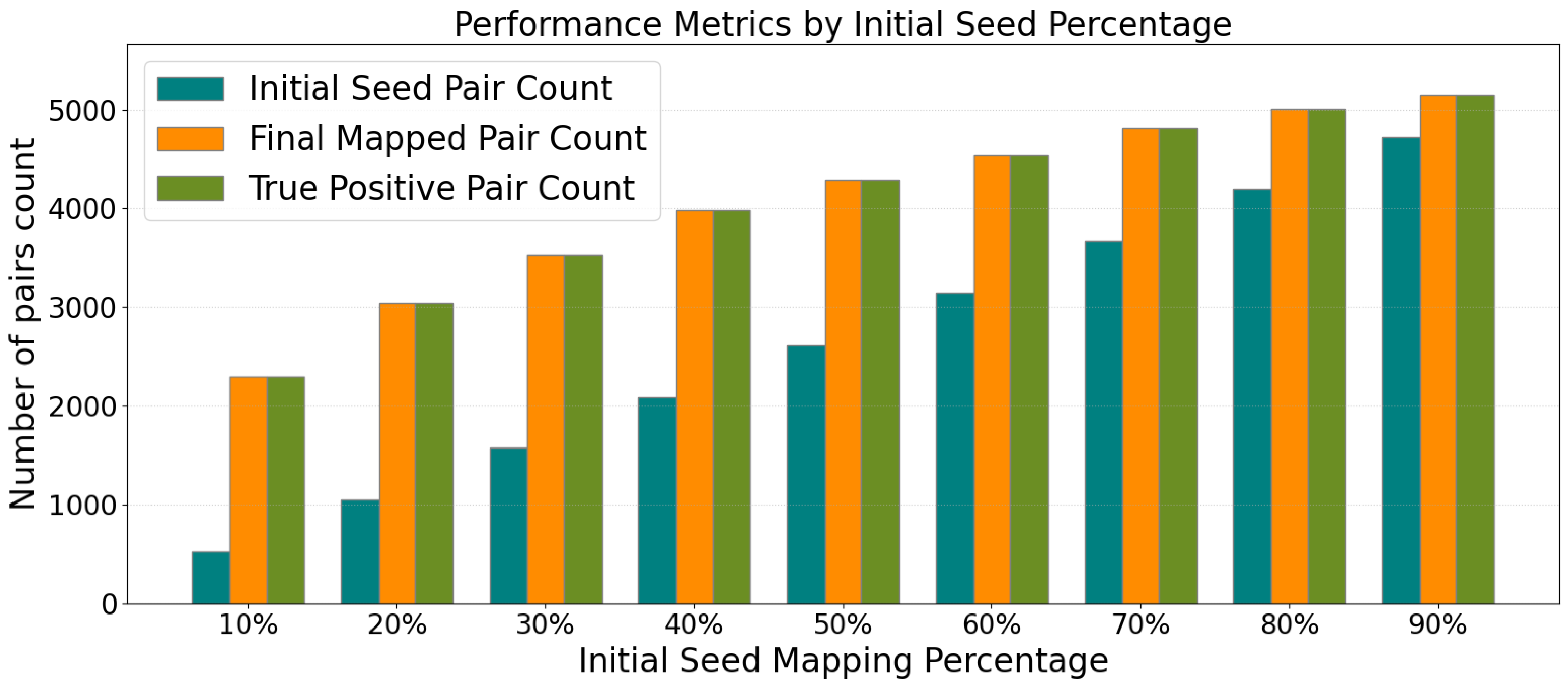
```
function matchScores(lgraph, rgraph, mapping, lnode)

  initialize scores = [0 for rnode in rgraph.nodes]

  for (lnbr, lnode) in lgraph.edges:
    if lnbr not in mapping: continue
    rnbr = mapping[lnbr]
    for (rnbr, rnode) in rgraph.edges:
      if rnode in mapping.image: continue
        scores[rnode] += 1 / rnode.in_degree ^ 0.5
```

# Propagation Method(3)

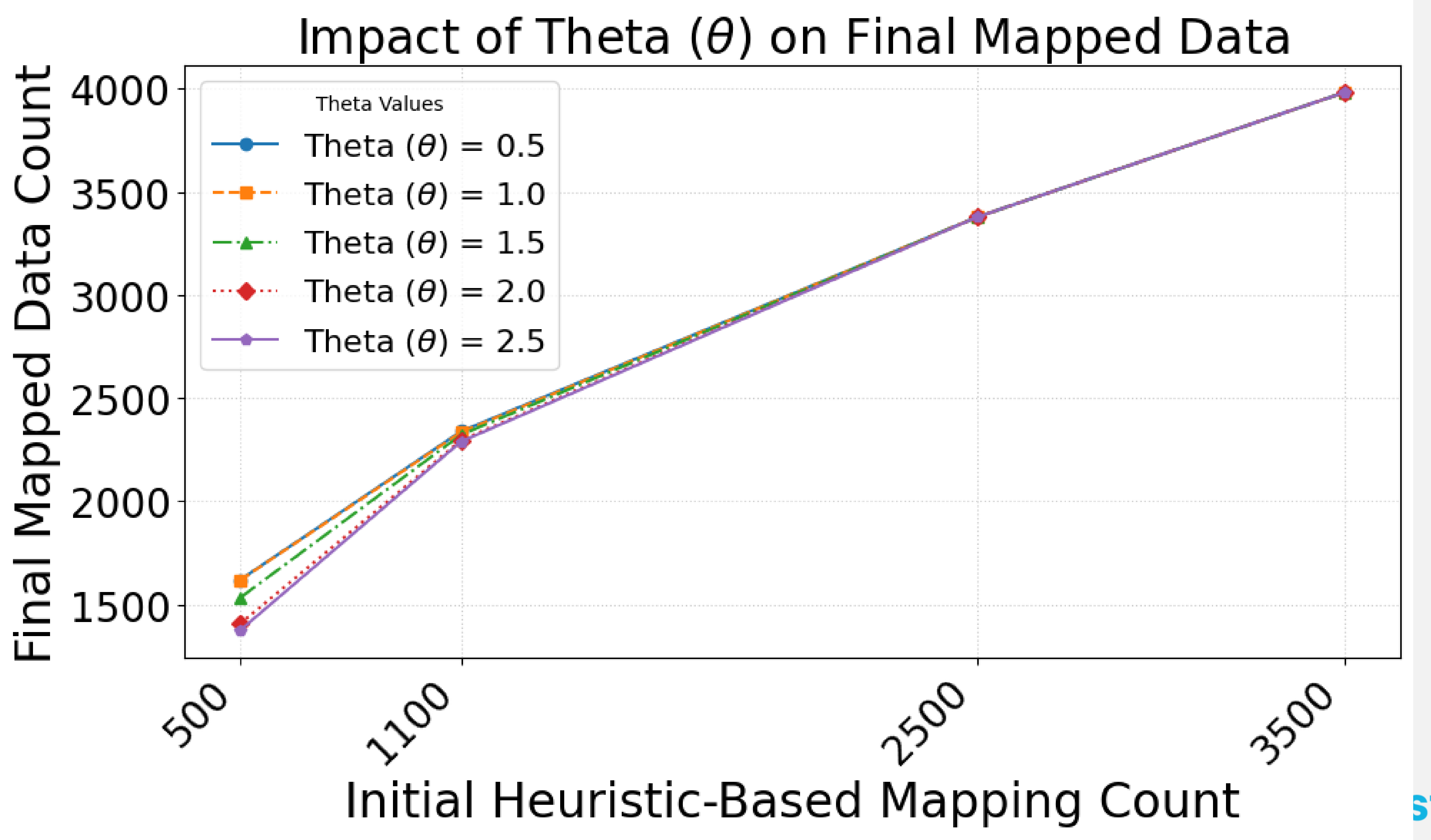
```
function eccentricity(items)

  return (max(items) - max2(items)) / std_dev(items)

until convergence do:
  propagationStep(lgraph, rgraph, seed_mapping)
```

We're doing it #thestateway

# Ablation Results on Validation Datasets(1)



# Ablation Results on Validation Datasets(2)




stateway

# Section 3

# Group Project
# Membership Inference Attack

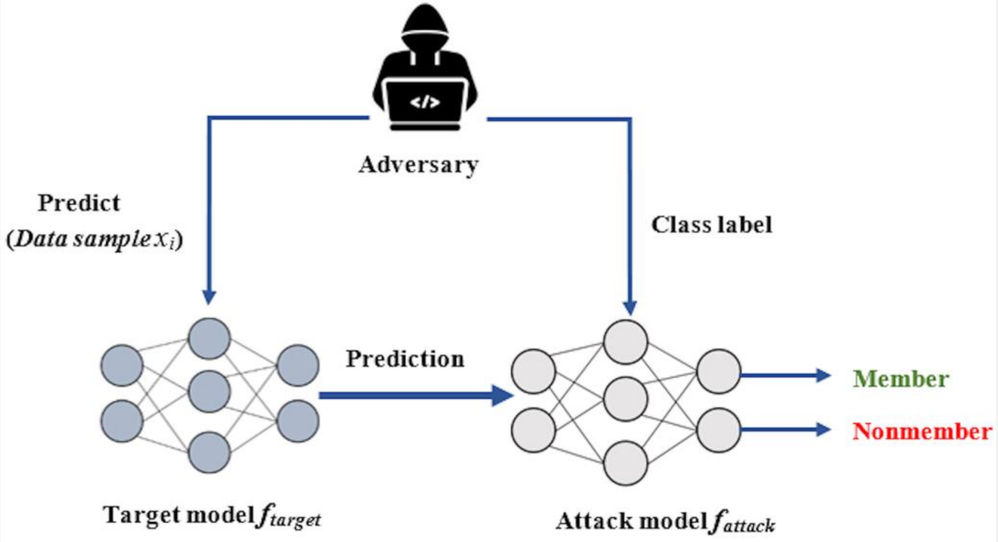
We're doing it #thestateway

# What is Membership Inference Attack?

- A privacy attack where the adversary predicts whether a sample was part of a model's training set

- Exploits the fact that models often behave differently on training vs non-training samples (overfitting, confidence differences, loss shape)

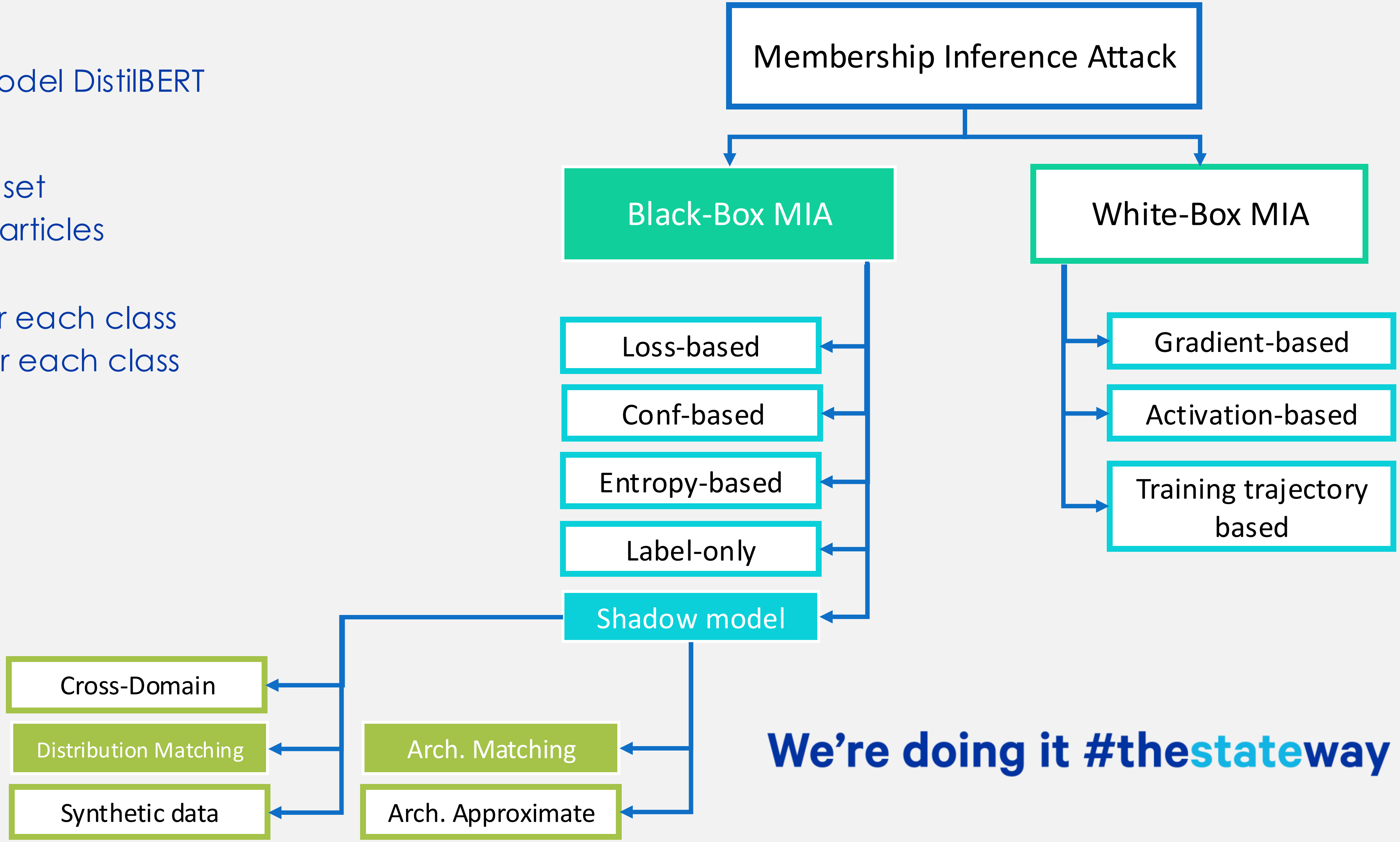
- Critical for privacy-sensitive domains: medical data, financial data, recommendation systems, NLP datasets




We're doing it #thestateway

# Problem Setting

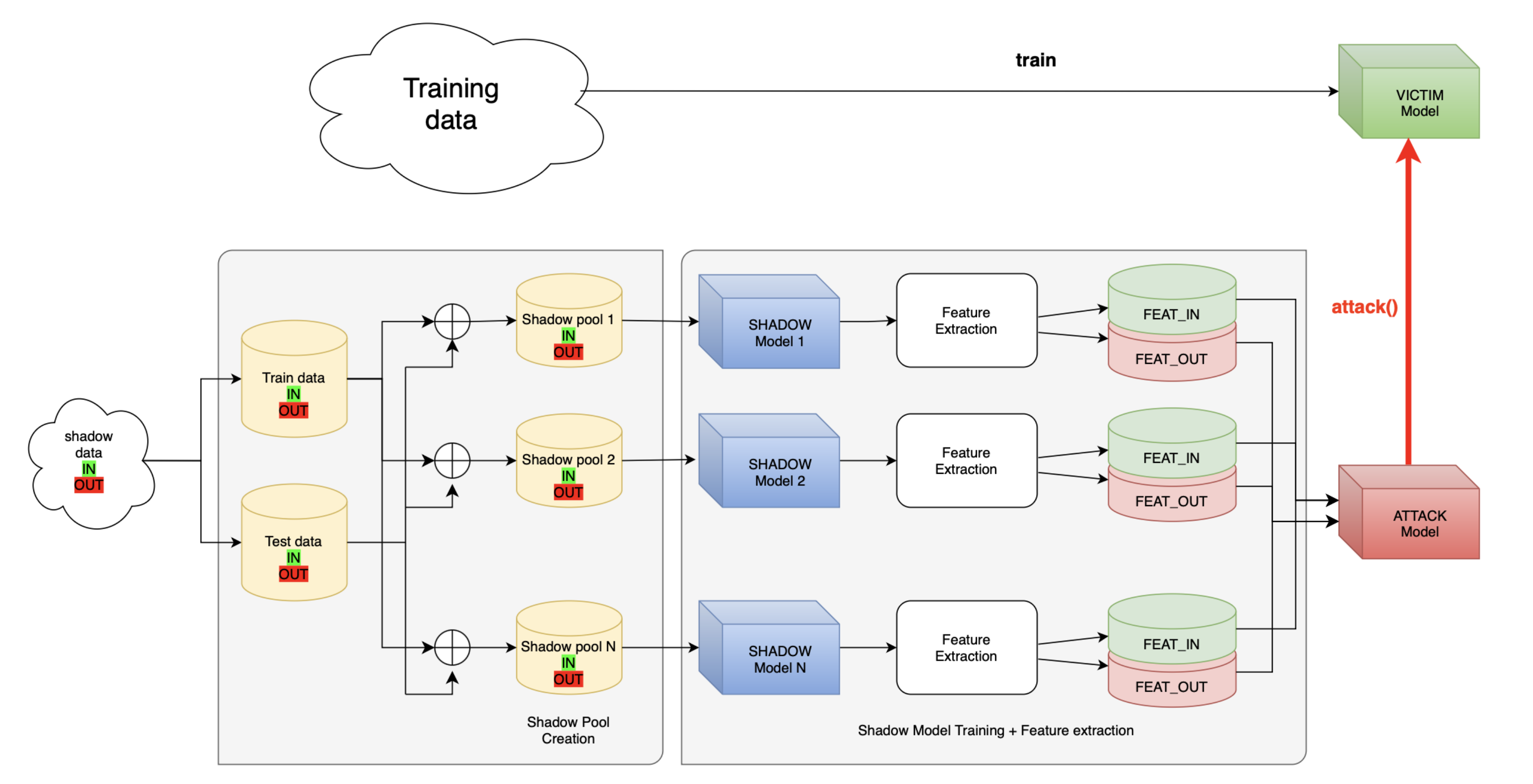
- Model
  - Transformer model DistilBERT
- Dataset
  - AG news dataset
  - 1 million news articles
  - 4 classes
  - 30K training for each class
  - 1900 testing for each class




We're doing it #thestateway

# Working Methodology

1. Shadow dataset creation
2. Shadow model training
    1. 5 shadows
    2. 5 epochs per shadow
3. Feature Extraction
    1. Using victim model
4. Attack dataset creation
5. Attack model training
6. Evaluate on victim model




We're doing it #thestateway

# Novelty

## Loss Criterion

- Noisy Label Cross-Entropy for Shadow training
  - Allows for misclassification
- Focal Loss for Shadow training (rejected)

## Multi-phase Optimization

- AdamW followed by LBFGS

## Model Selection

- Random Forest
- MLP (LayerNorm + LeakyRelu + Dropout)
- XGBoost

## Feature Extraction

- Confidence-based features
  - Confidence
  - Margin (Top-1 – Top-2)
  - Logit Margin
  - Logit Norm
  - correctness
- Loss-based features
  - Loss
- Distribution-based
  - Entropy
  - Class Norm

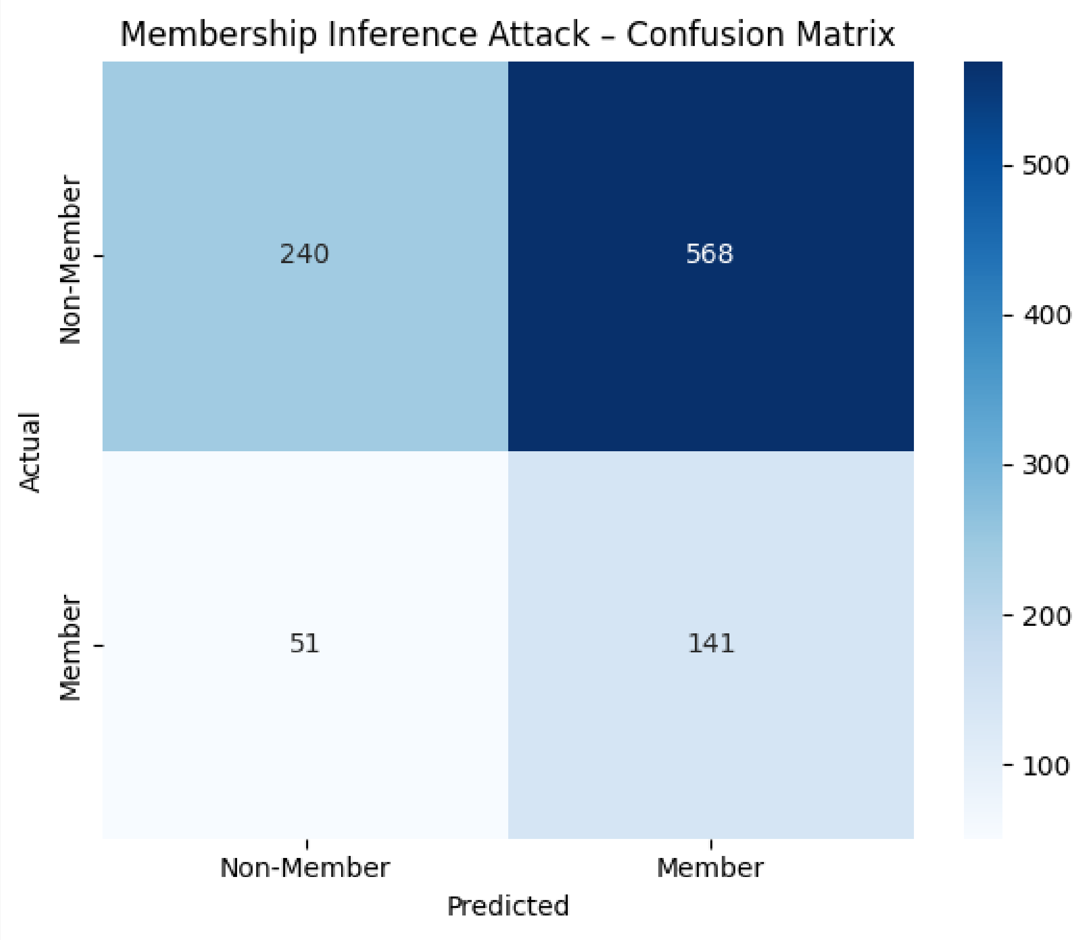
We're doing it #thestateway

# Results

## **Best Model Cofiguration**

- 5  shadow models
- Noisy Label CE (noise = 0.2)
- Attack MLP model
    - 3 layers
    - Layer Norm
    - Leaky ReLU
    - Dropout regularization

| Metric | Value |
| --- | --- |
| ROC_AUC | 0.5312 |
| Precision | 0.2188 |
| Recall | 0.734 |
| F1 | 0.3329 |
| Accuracy | 0.381 |




We're doing it #thestateway

# Section 4

# Future Plans

We're doing it #thestateway

# Future Plans

- Cross-attention based features
- GAN-based MIA
- No-data MIA

We’re doing it #thestateway

# Questions?

We're doing it #thestateway